

**Министерство образования и науки  
Донецкой Народной Республики  
ГОУВПО «Донецкий национальный технический университет»  
Институт компьютерных наук и технологий  
Факультет интеллектуальных систем и программирования  
Кафедра «Программная инженерия» им. Л.П. Фельдмана**

**Программная инженерия:  
методы и технологии разработки информационно-  
вычислительных систем**

**(ПИИВС-2022)**

**Сборник материалов IV Международной научно-практической  
конференции  
(студенческая секция)**

**г. Донецк  
29-30 ноября 2022 года**

**Донецк – 2022**

**Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2022):** сборник научных трудов IV научно-практической конференции (студенческая секция), Том 2, 29-30 ноября 2022 г. – Донецк, ГОУВПО «Донецкий национальный технический университет», 2022. – 200с.

Студенческая секция, проводимая в рамках научной конференции ПИИВС-2022, является одним из этапов совместной исследовательской деятельности преподавателей и студентов. Основная задача студенческой секции состояла в укреплении научного и педагогического сотрудничества среди студентов и научных руководителей, развитие у студентов навыков самостоятельной работы, способностей к анализу и обобщению изучаемого материала, умению формировать собственные выводы и заключения, излагать их письменно и в форме публичных выступлений.

Основные направления работы студенческой секции конференции:

- Современные языки и технологии программирования
- Информационные системы, базы данных, безопасность и защита данных
- Интеллектуальные системы, анализ данных, распознавание образов
- Компьютерное моделирование, системы виртуальной реальности, компьютерной графики и обработки изображений
- Проектирование программных и компьютерных систем, средства автоматизированного проектирования ПО и систем

На 5 подсекциях были представлены 38 докладов, в которых рассмотрен широкий спектр научных направлений работ студентов. Конференция организована Донецким национальным техническим университетом Министерства образования и науки ДНР. В организации конференции приняли участие: Министерство образования и науки ДНР, ГОУВПО «Донецкий национальный технический университет», ФГБОУ ВПО «Ульяновский государственный технический университет» (г.Ульяновск), ФГАОУ ВО «НИУ Московский институт электронной техники» (МИЭТ, г.Москва), Институт цифровой экономики и информационных технологий РЭУ им. Г.В.Плеханова (г.Москва).

Во втором томе сборника научных трудов представлены доклады студентов, которые учатся в бакалавриате и магистратуре Донецкого национального технического университета.

## ОГЛАВЛЕНИЕ

<b>СЕКЦИЯ 1. «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ПРОГРАММНЫЕ СИСТЕМЫ» .....</b>	<b>6</b>
<b>П.С. Похлёбин, А. В. Боднар</b> АНАЛИЗ ФРЕЙМВОРКОВ ДЛЯ РАЗРАБОТКИ ДЕСКТОПНОГО ПРИЛОЖЕНИЯ.....	6
<b>М.Е. Москаленко, Н.Е. Губенко</b> АНАЛИЗ И СОВЕРШЕНСТВОВАНИЕ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ КЛИЕНТА В СИСТЕМЕ “КРИПТОКОШЕЛЕК” .....	10
<b>К.К. Руднев, Т.Г. Дмитриук</b> СОЗДАНИЕ АВТОНОМНОЙ СИСТЕМЫ УПРАВЛЕНИЯ УМНЫМ ДОМОМ .....	16
<b>Я.В. Баляба, О.В. Рычка</b> ИССЛЕДОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ ИЗВЛЕЧЕНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ ИЗ АУДИО ФАЙЛОВ.....	21
<b>А.И. Машенко</b> ВЛИЯНИЕ ПРАВИЛ ЕСТЕСТВЕННОГО ЯЗЫКА НА КОНЦЕПЦИИ ПРОГРАММИРОВАНИЯ НА ПРИМЕРЕ WEB-РАЗРАБОТКИ .....	25
<b>А.В. Суворина, А.В. Боднар</b> ПОДХОДЫ К РАЗРАБОТКЕ 2D ПЛАТФОРМЕРА НА UNITY .....	29
<b>А.В. Боднар, М.Ю. Павлов</b> АНАЛИЗ ОСНОВНЫХ ТЕНДЕНЦИЙ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР .....	33
<b>Б.Ю. Егоров, И.Ю.</b> NODE.JS КАК ВАРИАНТ ВЫСОКОУРОВНЕВОГО ПРОГРАММИРОВАНИЯ .....	37
<b>А.А. Носаченко, С.В. Щедрин</b> ОСНОВНЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ МОБИЛЬНЫХ СИСТЕМ .....	41
<b>А.И. Боровиков, С.А. Зори</b> ОПТИМИЗАЦИЯ И ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ИГРОВЫХ ДВИЖКОВ ДЛЯ ANDROID.....	46
<b>Е.А. Закамаркина, Н.Е. Губенко</b> АНАЛИЗ ПРОЦЕССА ПЕДАГОГИЧЕСКОГО ДИЗАЙНА ПРИ СОЗДАНИИ ДИСТАНЦИОННОЙ СИСТЕМЫ ОБУЧЕНИЯ ИНОСТРАННОМУ ЯЗЫКУ .....	51
<b>В.И. Зензеров, Д.А. Куликов</b> АНАЛИЗ МЕТОДИК ПРОДВИЖЕНИЯ САЙТОВ .....	57
<b>СЕКЦИЯ 2. «ИНФОРМАЦИОННЫЕ СИСТЕМЫ, БАЗЫ ДАННЫХ, БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ» .....</b>	<b>62</b>
<b>В.В. Якубенко, Т.Г. Дмитриук</b> АНАЛИЗ ПОДХОДОВ К ПРОЕКТИРОВАНИЮ БАЗ ДАННЫХ.....	62

<b>К.Н. Ефименко, Д.В. Пауков, Н.С. Кострыкин, Е.Д. Калинин</b> АНАЛИЗ КРИПТОГРАФИЧЕСКИХ ПРОТОКОЛОВ УДАЛЕННОГО ДОСТУПА К ПК .....	68
<b>Е.Ю. Потребя, Н.Е. Губенко</b> ПРЕДОТВРАЩЕНИЕ УТЕЧЕК КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ ДИЗАЙН-СТУДИИ ТЕХНОЛОГИЯМИ DLP .....	74
<b>К.Н. Ефименко, С.А. Шлыков, Е.С. Исаев</b> РЕАЛИЗАЦИЯ БИЗНЕС-СИСТЕМЫ СЕРВИСНОГО ЦЕНТРА ПО ОБСЛУЖИВАНИЮ ТЕЛЕФОНОВ НА «1С: ПРЕДПРИЯТИЕ» .....	79
<b>О.В. Рычка, Р.В. Самойленко</b> РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ТЕСТИРОВАНИЯ И УЧЁТА УСПЕВАЕМОСТИ ОБУЧАЮЩИХСЯ .....	84
<b>А.О. Нанава</b> СРЕДСТВА И МЕТОДЫ РАЗРАБОТКИ БАЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ СУБД .....	88
<b>СЕКЦИЯ 3. «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, АНАЛИЗ ДАННЫХ И РАСПОЗНАВАНИЕ ОБРАЗОВ» .....</b>	<b>93</b>
<b>А.Д. Балабан, О.И. Федяев</b> МОДЕЛЬ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АНАЛИЗА ТОНАЛЬНОСТИ ОТЗЫВОВ К КИНОФИЛЬМАМ.....	93
<b>Е.В. Таций, И.Ю. Анохина</b> ПРИМЕНЕНИЕ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ ОБРАБОТКИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ.....	100
<b>А.В. Боднар, Е.С. Бондаренко</b> АНАЛИЗ ПЕРСПЕКТИВ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ RFID В ТОРГОВЛЕ.....	105
<b>А.С. Пильненький, О.И. Федяев</b> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ С МОДУЛЕМ РАСПОЗНАВАНИЯ РЕЧИ НА БАЗЕ ПЛАТФОРМЫ ANDROID.....	109
<b>Д.А. Присяжный, О.М. Копытова</b> 115 ИССЛЕДОВАНИЕ МЕТОДОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ И РАСПОЗНАВАНИЯ ОБРАЗОВ	115
<b>Я.Р. Парсаданян, А.В. Боднар</b> ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ПРИНЦИПА РАБОТЫ НЕЙРОННЫХ СЕТЕЙ .....	120
<b>К.Н. Ефименко, И.В. Бабич</b> УПРАВЛЕНИЕ ТРАНСПОРТНЫМ СРЕДСТВОМ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ .....	125
<b>Н.В. Мелешенко, О.И. Федяев</b> ПРИМЕНЕНИЯ МУЛЬТИАГЕНТНЫХ СИСТЕМ ДЛЯ УПРАВЛЕНИЯ ГРУППОЙ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ .....	129
<b>В.Р. Юрьева, И.Ю. Анохина</b> САЙТ КАК ЭЛЕМЕНТ ПРОДВИЖЕНИЯ ОБРАЗОВАТЕЛЬНЫХ УСЛУГ .....	135

<b>СЕКЦИЯ 4. «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, КОМПЬЮТЕРНОЙ ГРАФИКИ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ».....</b>	<b>140</b>
<b>О.Р. Семериков, С.А. Зори</b>	
ИССЛЕДОВАНИЕ ВЛИЯНИЯ ТЕХНОЛОГИЙ СГЛАЖИВАНИЯ НА РЕНДЕРИНГ ИЗОБРАЖЕНИЯ И ПРОИЗВОДИТЕЛЬНОСТЬ В ИГРАХ .....	140
<b>А.В. Гембицкий, О.И. Федяев</b>	
ПРОГНОЗИРОВАНИЕ НАПРАВЛЕНИЯ БУДУЩЕЙ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ УЧЕНИКОВ С ПОМОЩЬЮ МУЛЬТИАГЕНТНОЙ МОДЕЛИ СИСТЕМЫ ОБУЧЕНИЯ .....	145
<b>М.П. Руденко, Д.Е. Звягинцев</b>	
МЕТОДЫ ГЕНЕРАТИВНОГО МОДЕЛИРОВАНИЯ В ПРОМЫШЛЕННОМ ДИЗАЙНЕ.....	151
<b>Д.М. Зеленский, Л.А. Лазебная</b>	
ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПРОГРАММЫ ДЛЯ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ BLENDER .....	155
<b>С.А. Коломойцева, Д.Э. Баев</b>	
ОБЗОР ТЕХНОЛОГИЙ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ.....	162
<b>С.А. Ермоченко, Н.А. Ильина</b>	
БИБЛИОТЕКА ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ИЕРАРХИЧЕСКОЙ АГЛОМЕРАТИВНОЙ КЛАСТЕРИЗАЦИИ.....	162
<b>СЕКЦИЯ 5. «ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ, СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ» .....</b>	<b>166</b>
<b>В.В. Дорошенко, А.В. Григорьев</b>	
ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ГИС КАК СРЕДСТВ ГЕОЛОКАЦИИ.....	171
<b>В.К. Ремизов, А.В. Григорьев</b>	
ПРОБЛЕМАТИКА РАЗРАБОТКИ СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БЕСПИЛОТНЫХ АВТОМОБИЛЕЙ.....	177
<b>Е.А. Лазуренко, Я.А. Рудь, А.П. Ищенко, А.В. Григорьев</b>	
ТЕКСТОВЫЙ РЕДАКТОР С ВОЗМОЖНОСТЬЮ ОБРАБОТКИ ДОКУМЕНТОВ С НЕФОРМАТИРОВАННЫМ И ФОРМАТИРОВАННЫМ ТЕКСТОМ .....	182
<b>О.Р. Семериков, А.В. Григорьев</b>	
ИССЛЕДОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ CATIA V5 В ПРОЕКТИРОВАНИИ ЛИСТОВЫХ АЭРОКОСМИЧЕСКИХ ДЕТАЛЕЙ.....	186
<b>И.Р. Настеко, А.В. Григорьев</b>	
ОСОБЕННОСТИ ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ.....	190
<b>Р.В. Мальчева, Ю.Ю. Дьяченко</b>	
ОРГАНИЗАЦИЯ ПОИСКОВОГО ИНДЕКСА ПРИ ПРОЕКТИРОВАНИИ КОМПЬЮТЕРНОЙ БИБЛИОТЕЧНОЙ СИСТЕМЫ .....	197

# **СЕКЦИЯ 1. «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ПРОГРАММНЫЕ СИСТЕМЫ»**

УДК 004.42

## **Анализ Фреймворков для разработки десктопного приложения**

**П.С. Похлёбин<sup>\*1</sup>, А. В. Боднар<sup>\*2</sup>**

<sup>\*1</sup>студент, Донецкий национальный технический университет,  
xendri@list.ru

<sup>\*2</sup>к.э.н, доцент, Донецкий национальный технический университет,  
linabykova13@ya.ru, Orcid: 0000-0002-1800-0178, SPIN-код: 5068-7409

*Похлёбин П.С., Боднар А.В. JavaScript Фреймворк для разработки десктопного приложения. В статье представлен обзор существующих Фреймворков в JavaScript для разработки web-приложений. Проведен анализ фреймворков на основе их плюсов и минусов выбран лучший для дальнейшей разработки.*

**Ключевые слова:** Фреймворк, проектирование, Личный бухгалтер, React, Angular, Vue.js

### **Введение**

Создание веб-сайта является коммерческой необходимостью и условием выживания любой компании. При такой скорости развития Интернета ни одна компания, поставившая перед собой цель поступательного развития, не может обойтись без веб-сайта.

Интернет в настоящее время является наиболее популярным средством массовой информации, а создание веб-сайтов является серьезным инструментом решения маркетинговых стратегий, позволяющих эффективно и наиболее эффективно использовать достижения глобальной компьютерной сети с точки зрения финансовой привлекательности, технических возможностей и рекламных целей.

### **Анализ существующих Фреймворков. Определение Фреймворка программного обеспечения**

У JavaScript есть три популярных Фреймворка Angular, React и Vue.js

Чтобы выбрать Фреймворк необходимо разобраться что такое фреймворк необходимо дать ему краткое определение. Фреймворк – это программная платформа, которая упрощает разработку программного продукта, определяет структуру проекта и помогает удобно объединять в нём разные компоненты.

Для чего нужен фреймворк, какие функции они выполняет.

- Увеличивает скорость разработки.

- Упрощает работу.

- Обеспечивает безопасность

Фреймворки реализуют одну общую структуру Model-View-Controller (MVC):

- Модель(Model) – блок, отвечающий за доступ к файлам, находящимся в облаке или БД (базе данных).

- Отображение(View) – блок, отвечающий за отображение данных пользователю через веб-страницу или телефон.

- Контролер (Controller) – связывают два предыдущих блока для передачи данных из хранилища для отображения на устройстве.

### **React**

React – это open-source проект разработанный Facebook.

React сам по себе не позволит вам создать веб-приложение, поскольку он предназначен для создания представлений (отсюда и буква «V» в MVC). Что может сделать React, так это создать представления компонентов, для которых данные могут быть переданы в дочерние представления. Но для полноценной работы React необходим еще Flux данная архитектура в сочетании с React делает следующее:

- Пользователь нажимает на элемент React
- Происходит активация. После данное действие отправляется в store через библиотеку Dispatcher
- Хранилище наблюдает за состоянием приложения и методом извлечения данных. Также любое обновление состояния видно в представлениях (рис. 1).

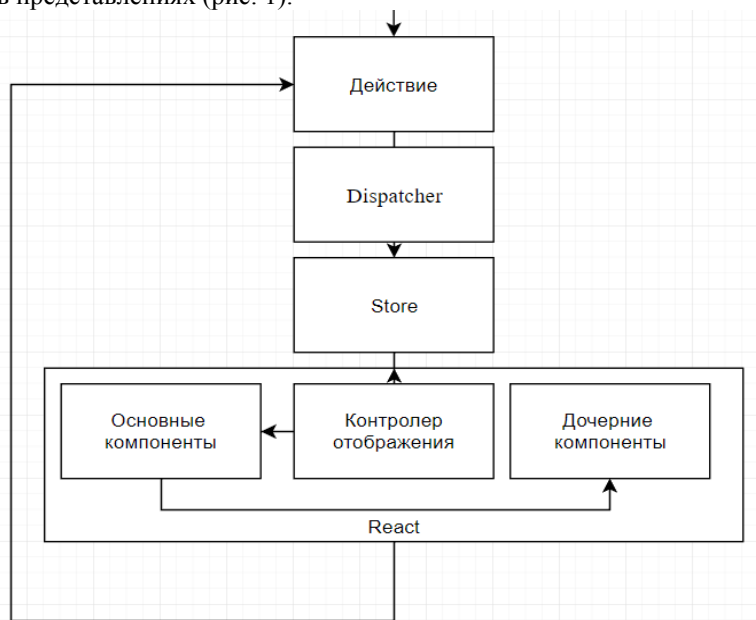


Рисунок 1 – Flux дополняющий React

Достоинством React является— неизменность компонентов и односторонняя привязка. Создатель React Джордан Вальке всегда был поклонником функциональной парадигмы, но никак не мог найти для себя подходящий фреймворк. Вот почему он написал этот трафик.

Приложения разрабатываются при помощи React Native, также кроссплатформенные (Android, iOS)

Основные преимущества этого Фреймворка:

- Гибкость приложения
- Использование DOM
- Поддержка высоких нагрузок
- Неизменность родительских данных
- Малый вес баз данных
- Простая миграция между версиями

Недостатки:

- Неупорядоченная документация
- Долго время для изучения минимальной информации

## Angular

AngularJS был прочно построен на основе шаблона MVC, который разделял приложение на три разных уровня. Комбинация модели, представления и контроллера, а также дополнительные сложности, связанные с освоением директив, фабрик, служб и других компонентов для создания одностраничного приложения.

В angular используется компонентный подход.

Компоненты - это самый базовый строительный блок пользовательского интерфейса в приложении Angular. Приложение Angular представляет собой дерево компонентов Angular.

В angular компоненты – это классы TypeScript над которыми помещен декоратор @Component. Иерархия компонентов в Angular разработана таким образом, что можно объединить структуру и функциональность в одном объекте. Высокоуровневый архитектурный обзор компонентов и их связь со всем остальным в Angular.

Основные преимущества Фреймворка Angular:

- Обширное разнообразие функций.
- Взаимосвязанность функций.
- Получение информации напрямую

- Малое количество ошибок.
- Недостатки:
  - Сложный язык
  - Ошибки во время миграции между версиями.

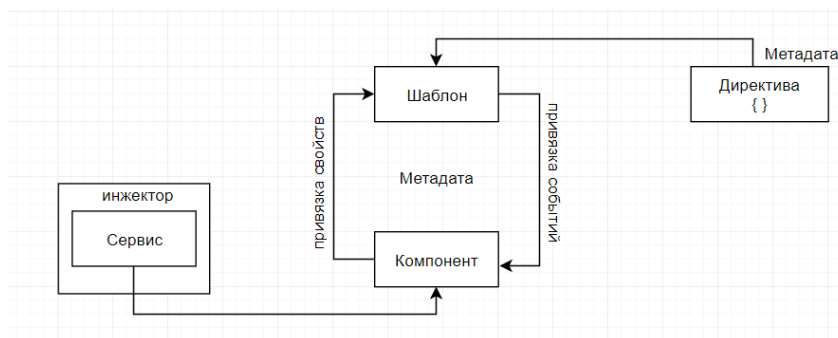


Рисунок 2 – Архитектура Angular

## Vue.js

Vue.js Фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Легко интегрируется в проекты с использованием других библиотек JavaScript. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле.

Отличительной особенностью является использование «на уровне представления», что позволяет интегрировать его с другими проектами и библиотеками.

Vue.js прост в освоении — основной целью команды разработчиков было сделать фреймворк, с которым могли бы полноценно начать работать веб-мастера разной квалификации. Средних знаний HTML, CSS, JavaScript будет достаточно, чтобы полноценно использовать Vue и создавать на нем мощные проекты.

Основными концепциями Vue являются: конструктор, компоненты, директивы, переходы.

Самые распространение функции Vue – реактивные интерфейсы, декларативный рендеринг, логика шаблонов

Преимущества Фреймворка Vue.js:

- Малый вес (17кб)
- Детальная и обширная документация.
- Лёгкость составления документации к проектам.
- Постоянное обновление компонентов.
- Продвинутый HTML
- Реактивность, скорость

Недостатки:

- Нехватка ресурсов.
- Нечеткие архитектурные шаблоны.
- Неоднозначный.

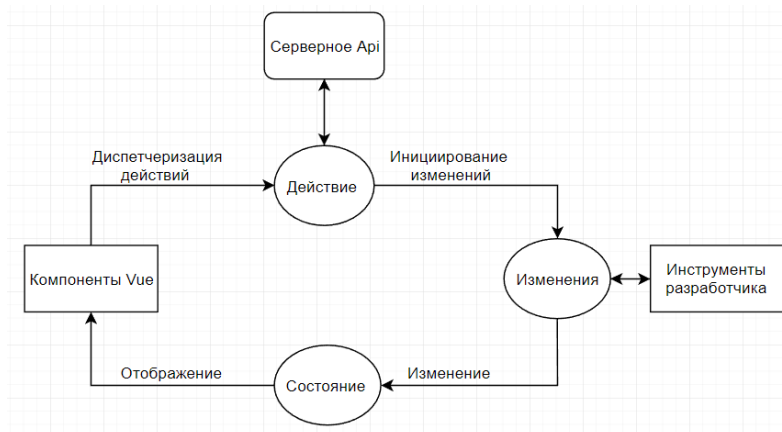


Рисунок 3 – Архитектура Vue.js



## **Вербальное описание учебного портала**

Основная цель реализации проекта – создание кроссплатформенного приложения для учета денежных средств с поддержкой возможности создания секций в которых будут указаны расходы на ту или иную потребность с категориями (возможно создание групповых секций). Кроссплатформенный личный менеджер представляет из себя приложение, которое поможет более рационально использовать свои деньги или вести электронный отчет по расходам за определенный период времени. Кроссплатформенный личный бухгалтер приложение, которое будет иметь платную версию с большим функционалом. Решение второстепенных задач (реклама) появляется возможность внедрения рекламы (довольно успешное) так как это приложение отвечает за деньги, следовательно, можно будет брать рекламу разных направлений.

## **Выбор архитектуры**

В связи с большим объемом и высокой сложностью разрабатываемого продукта, было принято решение не рассматривать Angular так как он не очень выдерживает большие объёмы данных также имеет обширное количество недочетов. Перспективным был выбран React, так как он позволяет работать с большими объёмами также вес баз данных очень мал по сравнению с Angular и Vue. React хорошо подходит для маленьких проектов но с большим количеством расчетов и преобразований.

## **Выводы**

Таким образом, в статье проведен анализ самых распространённых Фреймворков JavaScript. Был сформирован план построения структуры сайта также некий его функционал на основе Фреймворка React. Данный Фреймворк больше всего подходит для не объемных десктопных приложений с большим количеством математических операций.

## **Литература**

1. Vue vs React and Angular [Электронный ресурс] -Режим доступа: <https://proglib.io/p/vue-vs-react-vs-angular-kakoy-frontend-freymvork-vybrat-2020-08-03>
2. Руководство по React [Электронный ресурс] - Режим доступа: [https://habr.com/ru/company/ruvds/blog/471040/ng-book-2: The Complete Book on Angular 5 \[+ Code\] \(2018\) \[PDF\] \[En\] – 647с](https://habr.com/ru/company/ruvds/blog/471040/ng-book-2: The Complete Book on Angular 5 [+ Code] (2018) [PDF] [En] – 647с)
3. Angular и TypeScript. Сайтостроение для профессионалов - Файн Я., Моисеев А. [2021, PDF, en] – 434с [Электронный ресурс] - Режим доступа: [https://vk.com/wall-51126445\\_23593](https://vk.com/wall-51126445_23593)
4. Саймон Холмс "Стек MEAN. Mongo, Express, Angular, Node" [2021, PDF, en] – 644с [Электронный ресурс] - Режим доступа: [https://vk.com/wall-54530371\\_184655?lang=en](https://vk.com/wall-54530371_184655?lang=en)

**Похлёбин П.С., Боднар А.В. JavaScript Фреймворк для разработки десктопного приложения.** В статье представлен обзор существующих Фреймворков в JavaScript для разработки web-приложений. Проведен анализ фреймворков на основе их плюсов и минусов выбран лучший для дальнейшей разработки.

**Ключевые слова:** Фреймворк, проектирование, Личный бухгалтер, React, Angular, Vue.js

**Pokhlebin P.S., Bodnar A.V. JavaScript Framework for developing desktop applications.** The article provides an overview of existing JavaScript frameworks for developing web applications. Based on their pros and cons, the frameworks were analyzed and the best one was selected for further development.

**Keywords:** Framework, design, Accountant, React, Angular, Vue, js

## Анализ и совершенствование модели взаимодействия клиента в системе “Криптокошелек”

М.Е. Москаленко\*1, Н.Е. Губенко\*2

\*1 бакалавр, Донецкий национальный технический университет,  
korolevskiy.m@mail.ru

\*2 к.т.н., доцент, Донецкий национальный технический университет,  
negubenko@mail.ru

*Москаленко М.Е., Губенко Н.Е. Анализ и совершенствование модели взаимодействия клиента в системе “Криптокошелек”.* Статья посвящена одной из актуальных и свежих тем по совершенствованию взаимодействия в кошельках цифровой валюты. Рассматривает возможность улучшения безопасности хранения криптовалюты, анализирует существующие системы хранения.

**Ключевые слова:** продуктовый дизайн, криптокошельки, цифровизация, криптовалюта, криптография, безопасность.

### Криптовалюта: основные понятия

В настоящее время в связи с популярностью и развитием нового вида электронных денег - криптовалюты, актуальны задачи, связанные с их научным осмыслением и необходимостью для современного пользователя.

Несмотря на высокий рейтинг, сегодня не существует единого, признанного в мире определения криптовалют, которое бы в полной мере раскрывало их сущность. Это объясняется новизной данного инструмента и разнообразием технических решений, реализованных в системах электронных расчетов. Анализируя множество источников, криптовалюта - это цифровой или виртуальный актив, в котором для обеспечения безопасности используется криптография. Криптовалюты децентрализованы, что означает, что они не подконтрольны правительству или финансовым учреждениям.

Биткойн - первая и самая известная криптовалюта была создана в 2009 году. Криптовалюты часто торгуются на децентрализованных биржах, а также могут использоваться для покупки товаров и услуг. Например, биткойн можно использовать для бронирования отелей на Expedia, покупки мебели на Overstock и игр для Xbox. Но это лишь верхушка айсберга. Настоящая история - это технология блокчейн, на которой основаны криптовалюты.

Принцип функционирования криптовалют основан на сети узлов, поддерживающих блокчейн, что делает практически невозможной подделку данных без обнаружения другими участниками сети. Это происходит потому, что каждый узел или компьютер, подключенный к сети Bitcoin, получает копию блокчейна, которая загружается автоматически при присоединении к сети. Узлы помогают поддерживать безопасность сети, одобряя транзакции. Неодобренные транзакции не добавляются в блокчейн.

Майнеры биткойна - это узлы, которые обеспечивают безопасность блокчейна, одобряя транзакции. За свою работу они получают криптовалюту (подобно тому, как наша современная банковская система вознаграждает работников фиатной валютой). Майнеры используют специальное программное обеспечение для решения математических задач и получают определенное количество биткойнов в обмен за свою работу. Это обеспечивает разумный способ выпуска новой валюты и одновременно стимулирует большее количество людей к майнингу.

Следует также отметить, что курс любой криптовалюты характеризуется высокой волатильностью, что способствует к образованию восходящего рынка и нисходящего рынка. Пример показан на основе курса криптовалюты Биткойн за 2017–2019 г. (см. рис. 1).

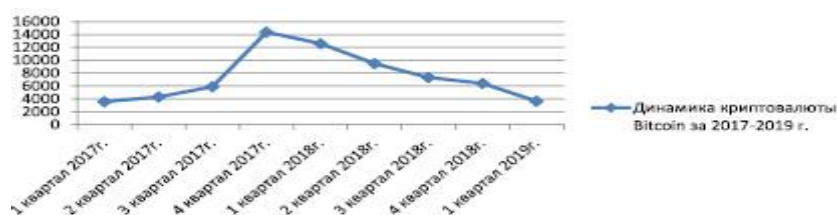


Рисунок 1 - Динамика курса Bitcoin поквартально за 2017–2019 г.

### Классификация криптокошельков, их безопасность

Существует четыре основных типа кошельков:

1. Онлайн-кошельки - это программные кошельки, которые хранятся в Интернете. Они удобны тем, что к ним можно получить доступ из любого места, но они также менее безопасны, поскольку подвержены взлому.
2. Автономные кошельки, также называемые холодными хранилищами, не подключены к Интернету. Они могут храниться на USB-накопителе или другом внешнем жестком диске. Они более безопасны, поскольку не подвержены взлому, но менее удобны, поскольку доступ к ним возможен только тогда, когда накопитель находится при вас.
3. Аппаратные кошельки - это физические устройства, которые хранят электронные монеты в автономном режиме. Они очень надежны, поскольку их невозможно взломать, но их можно потерять или украсть.
4. Бумажные кошельки - это автономные кошельки, которые распечатываются на бумаге. Они достаточно безопасны, поскольку их невозможно взломать, но их можно потерять или украсть.

Оптимальный выбор кошелька зависит от личных потребностей пользователя. Самым безопасным является офлайн-кошелек, удобным - онлайн-кошелек. Баланс между безопасностью и удобством выполняет аппаратный кошелек.

Как показывает практика, криптовалюты характеризуются высокой безопасностью, так как зашифрованы и децентрализованы, что затрудняет их взлом. Однако, произошедшие в последние годы громкие взломы криптовалютных бирж, говорят о том, что нет никакой гарантии при потере денег получить их обратно, так как криптовалюты не застрахованы в FDIC, как традиционная валюта. Поэтому безопасность является одним из главных условий использования электронных денег.

В связи с этим рассмотрим на примере самые известные кошельки для хранения крипто-активов и их безопасность (таблица 1):

Таблица 1 – Характеристика криптокошельков и их безопасности

Наименование	Краткое описание	Безопасность
Atomic Wallet	Холодный кошелек, поддерживает более 300 различных криптовалют, включая Bitcoin, Ethereum, Litecoin, Bitcoin Cash, Monero, Dash, др. Имеет интуитивно понятный интерфейс, позволяет покупать/продавать криптовалюты прямо из приложения с комиссией 0%. Предлагает мобильное приложение для управления своими криптоактивами на ходу. Пользователи имеют полный контроль над своими приватными ключами, которые обычно шифруются и хранятся на их устройствах, что дает им полное право распоряжаться своими средствами без прохождения верификации личности (KYC). Несмотря на то, что программный код кошелька недоступен для экспертной оценки (т.е. не является открытым исходным кодом), платформа представляет пользователям ресурсы о том, как защитить свои кошельки. Один из таких ресурсов подчеркивает важность доступа к кошельку с помощью виртуальной частной сети (VPN). VPN - это сеть, которая обеспечивает конфиденциальность и анонимность путем создания частной сети из публичной сети. Криптокошелек - один из лучших, когда речь идет о конфиденциальности и анонимности.	Кошелек завоевал репутацию в криптовалютной индустрии, поскольку не было ни одного случая о нарушении безопасности. Для авторизации любых транзакций Atomic Wallet требует от пользователей ввода пароля. Пользователи могут восстановить свои учетные записи с помощью резервной начальной фразы из 12 слов, которая генерируется случайным образом при первой настройке кошелька Atomic.

Продолжение таблицы 1 - Характеристика криптокошельков и их безопасности

Exodus	<p>Холодный кошелек, который поддерживает биткоин, Ethereum, Litecoin, Dash, OmiseGO, токены Augur REP, токены Golem GNT, токены Aragon ANT, токены Decentraland MANA, токены district0x DNT, токены Civic CVC и многие другие токены ERC20, созданные на блокчейне Ethereum. Exodus имеет интуитивно понятный интерфейс с живыми графиками и позволяет обменивать криптовалюты прямо из приложения с комиссией 0%. Exodus также предлагает мобильное приложение для управления криптоактивами на ходу. Кошелек Exodus твердо верит в конфиденциальность и неограниченные права своих пользователей. В связи с этим у него нет требований к верификации личности (KYC), что означает, что пользователям не требуется регистрироваться или предоставлять личную информацию. Все транзакции на платформе анонимны, и Exodus генерирует новый кошелек для каждой транзакции BTC. Преимуществом этого является то, что использование кошелька Exodus не требует разглашения какой-либо личной информации. У Exodus нет доступа к кошельку пользователя, и компания не собирает личные данные. Создание нового кошелька для каждой транзакции BTC добавляет уровень безопасности и увеличивает препятствия в случае, если злоумышленник захочет отследить ваши транзакции BTC. Однако пользователю необходимо сохранять возможность просматривать все адреса своих кошельков.</p>	<p>Ключевым фактором, который следует отметить в отношении безопасности, является то, что на данный момент Exodus не имеет доступа к активам пользователя или приватным ключам. Структура кошелька, не требующая хранения, означает, что все хранение и шифрование происходит на жестком диске пользователя, а не на платформе Exodus. После загрузки платформы пользователи получают доступ к своему счету по паролю. При необходимости восстановить учетную запись, пользователю будет предложено сделать это с помощью фразы восстановления из 12 слов.</p>
Blockchain Wallet	<p>Холодный десктопный и мобильный криптовалютный кошелек, который поддерживает более 20 цифровых активов, включая Bitcoin, Ethereum, Litecoin и Bitcoin Cash. Blockchain Wallet шифрует ваши приватные ключи и хранит их на вашем компьютере или мобильном устройстве в зашифрованном формате. Blockchain Wallet также предоставляет пользователю резервную копию вашей начальной фразы, чтобы можно было восстановить свои средства, если доступ к компьютеру или мобильному устройству потерян. Blockchain Wallet можно скачать и использовать бесплатно.</p>	<p>Кошелек имеет несколько уровней безопасности для защиты средств пользователей от любых возможных злоумышленников, включая саму компанию. Как и другие цифровые сервисы, счета Blockchain Wallet требуют паролей для защиты пользователей. Однако компания Blockchain не хранит пароли пользователей и не может сбросить их в случае утери. Эта мера предотвращает возможность кражи криптовалют инсайдерами компании. Если пользователь забывает или теряет пароль, восстановить учетную запись можно только с помощью мнемонических фраз. Для снижения опасности фишинга, Blockchain Wallet позволяет пользователям использовать двухфакторную аутентификацию или белые списки IP-адресов для предотвращения входа в систему с незнакомых устройств. Также можно заблокировать доступ через сеть Tor, тем самым, не позволяя потенциальным хакерам маскировать свои IP-адреса.</p>

Таким образом, проанализировав особенности криптокошельков, можно выделить их преимущества и недостатки.

Плюсами использования крипто-активов является:

1. Криптовалюты зашифрованы и децентрализованы, что затрудняет их взлом.
2. Криптовалюты не подлежат контролю со стороны правительства или финансовых учреждений.
3. Криптовалюты можно использовать анонимно, что дает пользователям больше конфиденциальности, чем традиционные банковские операции.
4. Комиссии за транзакции с криптовалютами обычно ниже, чем комиссии за традиционные банковские операции.
5. Стоимость криптовалют не подвержена инфляции, поскольку предложение ограничено.

Недостатками таких систем выступают:

1. Несколько громких случаев взломов криптовалютных бирж за последнее время.
2. Криптовалюты не застрахованы FDIC, как традиционная валюта.
3. Стоимость криптовалют нестабильна и может быстро колебаться.
4. Криптовалюты еще не получили широкого признания в качестве формы оплаты.
5. В настоящее время криптовалюты используются в основном только инвесторами и спекулянтами.

В заключении, исследуя данный вопрос, следует отметить: идентификация и верификация личности для хранения криптовалюты – нецелесообразно, так как собственные средства пользователя становятся централизованными и ими может управлять биржа. Оптимальным вариантом является использование кастодиального кошелька, который можно восстанавливать систему при помощи 12 мнемонических фраз. К кастодиальному (холодному) кошельку никто не имеет доступа, кроме пользователя. Для шифрования используется некоторый символьный пароль пользователя, который не является устойчивым к прямому перебору или атакам по словарю и для предотвращения этих атак используется алгоритм scrypt. Алгоритм проектируется таким образом, чтобы тратить значительные ресурсы памяти и вычислений для значительного усложнения перебора по словарю или грубой силой (см. рис. 2).

```
{
  "address": "99a3ef9e714dc69b02c14b0f24be09fabd2ac183",
  "crypto": {
    "cipher": "aes-128-ctr",
    "ciphertext": "abad78d578512bf8cc49e433f9eaced55348caed9704231d9c58c8affbec29df",
    "cipherparams": {
      "iv": "b3a53514eadbb2e6a312fdad69be071e"
    },
    "kdf": "scrypt",
    "kdfparams": {
      "dklen": 32,
      "n": 262144,
      "p": 1,
      "r": 8,
      "salt": "b916fe052c6c079c406a7dafdc5305c7eb999e8510a9d75c1a7a97b7f7821b66"
    },
    "mac": "b1f98340578ae32e86a9bdf50c3c48d2e3f30444d013c377fc5e599778040f1"
  },
  "id": "dfb00c15-bf1f-449c-8703-1b2b41f3ebf8",
  "version": 3
}
```

Рисунок 2 - Файл хранения ключей

Сохранность частных ключей в кошельках базируется на устойчивости используемых криптографических функций и защищенности аппаратных компонентов. Использование незащищенных чипов приводит к возможности прямого считывания паролей и частных ключей из памяти устройства посредством специальных устройств и ПО. В связи с этим считаю, что в кошельках следует начать развивать систему 24 фраз, так как это усложнит возможность взлома для хакеров.

### **Дизайн электронных денег: анализ и выявление недостатков**

Брендинг и дизайн криптовалюты не имеют физического воплощения, поэтому они рассматриваются пользователями исключительно визуально-виртуально. Изучая логотипы электронных денег и отзывы пользователей, напрашивается вывод: дизайн криптовалют разработан слабо, с большим количеством графических и стилистических ошибок, которые накладывают отпечаток на осознанное или неосознанное мнение о проекте. Самые распространенные ошибки в дизайне популярных криптовалют, которые лежат на поверхности и бросаются в глаза даже непрофессионалам рассмотрим в таблице 2.

Таблица 2 – Логотипы криптовалюты с позиции дизайна

Криптовалюта	Сильные стороны	Слабые стороны
Bitcoin	<ol style="list-style-type: none"> <li>1. Визуальный акцент на знак доллара отражает революционную идею "новая мировая валюта"</li> <li>2. Активный цвет, побуждающий к движению в контрасте с черным - цветовой акцент на революционности</li> </ol>	<ol style="list-style-type: none"> <li>1. Диссонанс шрифта заглавной В подкашивает обещание бренда</li> <li>2. Несбалансированные пропорции логотипа к фирменному знаку.</li> </ol>
Litecoin	<ol style="list-style-type: none"> <li>1. Простой шрифт, серебристый оттенок, говорящий о «цифровом серебре»</li> </ol>	<ol style="list-style-type: none"> <li>1. Диссонанс шрифта «L» в знаке с шрифтом логотипа</li> <li>2. Несоответствие пропорций логотипа и знака</li> <li>3. Отсутствие ясного месседжа, выраженного через лого</li> <li>4. Серый цвет не подчеркивается общим дизайном и может смотреться слабо на фоне конкурентов</li> </ol>
Verge	<ol style="list-style-type: none"> <li>1. Аккуратный шрифт, заглавные буквы придают стабильность</li> <li>2. Красивое звучание и значение слова</li> </ol>	<ol style="list-style-type: none"> <li>1. Стрела вниз может интерпретироваться как “движение” проекта вниз</li> <li>2. Более несбалансированные пропорции логотипа к фирменному знаку</li> </ol>
Dash	<ol style="list-style-type: none"> <li>1. Уникальное решение для "D" в знаке, которая несет прямой акцент на открытости</li> <li>2. Наклон и черточка усиливают ощущение скорости, что важно в транзакциях</li> <li>3. Краткость</li> </ol>	<ol style="list-style-type: none"> <li>1. Стилистика шрифта отправляет в 80-е со всеми плюсами и минусами</li> <li>2. Гарнитура шрифта немного тяжеловата для цифровой тематики и ближе к теме стритрейсинга и ралли</li> <li>3. Классический оттенок</li> </ol>

Таким образом, когда дело касается инвестиций в «электронные деньги», стоит большое внимание уделить разработке дизайна логотипа и брендинга, так как два фактора определяют будущий успех: идея и упаковка.

### **Заключение**

Криптовалюта - это инновационная разработка, которая возросла до огромных масштабов, а децентрализованные сети получили большую популярность. Исследовав данную тему, можно сделать следующие выводы и внести некоторые предложения:

1. Для хранения электронных денег в кошельке для долгосрочного роста следует выбирать кошелек, который даст контроль над закрытыми ключами (кошелек без права хранения), а именно аппаратные и бумажные криптокошельки.

2. Для регулярного пользования средствами целесообразен вариант использования кошелька с хранилищем, например, Atomic Wallet или Exodus. Они просты в использовании, но не дают вам контроля над закрытыми ключами, поэтому они могут не подойти для больших объемов криптовалюты.

3. Необходимо использовать несколько форм хранения важных закрытых ключей и никогда не хранить их на устройствах, подключенных к Интернету.
4. Дизайн современной криптовалюты и кошельков небезупречен. Слабый брендинг может негативно сказаться на дальнейшем успехе в реализации идей.
5. При разработке дизайна для криптовалюты и кошельков для их хранения стоит обратить особое внимание на визуальные аспекты системы идентификации, которая начинается именно с логотипа.
6. Для повышения надежности в системе идентификации и верификации личности в кошельках следует начать развивать систему 24 фраз, так как это усложнит возможность взлома для хакеров.

### **Литература:**

1. Crypto-Currency Market Capitalizations [Электронный ресурс] - Режим доступа -URL: <http://coinmarketcap.com/> (дата обращения 11.10.2022).
2. Новости криптовалют: [Электронный ресурс] — Режим доступа. — URL: <http://cryptomart.ru/kriptovalyuta/> (Дата обращения 07.10.2022).
3. Вести и экономика [Электронный ресурс] — Режим доступа — URL: <http://www.vestifinance.ru/articles/31342> (Дата обращения: 12.10.2022)
4. Риск криптовалют, [Электронный ресурс], URL: <https://www.optiv.com/blog/the-risk-of-cryptocurrencies> (дата обращения: 12.10.2022).
5. Global charts | CoinMarketCap. [Электронный ресурс]. URL: <https://coinmarketcap.com/charts/> (дата обращения: 12.10.2022).
6. Market Share Statistics for Internet Technologies, [Электронный ресурс], URL: <https://netmarketshare.com/> (дата обращения: 12.10.2022).

**Москаленко М.Е., Губенко Н.Е. Анализ и совершенствование модели взаимодействия клиента в системе “Криптокошелек”.** Статья посвящена одной из актуальных и свежих тем по совершенствованию взаимодействия в кошельках цифровой валюты. Рассматривает возможность улучшения безопасности хранения криптовалюты, анализирует существующие системы хранения.

**Ключевые слова:** продуктовый дизайн, криптокошельки, цифровизация, криптовалюта, криптография, безопасность.

**Moskalenko M.E., Gubenko N.E. Analysis and improvement of the client interaction model in the Crypto Wallet system.** The article is devoted to one of the topical and fresh topics on the improvement of interaction in digital currency wallets. It considers the possibility of improving the security of cryptocurrency storage, analyzes the existing storage systems.

**Keywords:** product design, cryptocurrency wallets, digitalization, cryptocurrency, cryptography, security.

## Создание автономной системы управления умным домом

К.К. Руднев<sup>\*1</sup>, Т.Г. Дмитриук<sup>\*2</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
thebooom@yandex.ru

<sup>\*2</sup> ассистент, Донецкий национальный технический университет,  
tnauka@lenta.ru, OrcID: 0000-0003-4803-5555, SPIN-код: 9249-2928

*Руднев К.К., Дмитриук Т.Г. Создание автономной системы управления умным домом. Статья посвящена разработке проекта аппаратно-программного комплекса для автономной системы управления умным домом на базе платформы Arduino. Разработанная система позволит в автоматизированном режиме управлять экосистемой помещения.*

**Ключевые слова:** умный дом, сервер, автоматизация, схема, плата.

### Введение

В настоящее время всё больше востребованной является система автоматизации контроля домашних устройств, которые принято называть «умными домами». Такая система, в зависимости от задач и масштабов, может использоваться в помещениях практически любых типов: дом, квартира, офис или гаражное помещение. В каждом индивидуальном случае найти готовый макет, разработать свой или нанять для этих целей специалистов – не составляет труда. Проекты, предложенные в публикациях, включают в себя как схемы отдельных частей, так и полностью готовое к установке и настройке программное обеспечение.

Авторами статьи поставлена следующая задача: разработка базового набора макета экосистемы «умный дом», который возможно беспрепятственно настроить и дополнить новыми элементами, устройствами, а также задачами.

Разрабатываемый автоматизированный комплекс ограничен по количеству используемых приборов. Допустим наше помещение – небольшая квартира и все используемые компоненты должны быть максимально автономны от действий человека. К достоинствам использования предложенного прототипа можно отнести следующие:

- низкая стоимость;
- гибкость настройки для нужд потребителя;
- повышение условий энергобезопасности в помещении;
- возможность удаленного контроля;
- возможность получения статистических показателей работы экосистемы;
- возможность взаимодействия и управления АСУ физически и удаленно (например, через веб-страницу и Telegram-канал).

### Серверный модуль предложенной разработки

Для получения возможности удаленного управления умным домом требуется разработать программное обеспечение для получения и отправки данных. Среди всех возможных вариантов ПО для АСУ данного рода стоит остановиться на использовании программной платформы для создания быстрых и масштабируемых веб-приложений «Node.js». Преимуществом является тот факт, что в «Node.js» применяется неблокирующая модель управления событиями ввода-вывода, что представляет платформу одним из лучших вариантов [1].

Задачи разработки данного ПО состоят в следующем:

- создание API для передачи информации из Arduino,
- настройка базы данных для хранения и обработки всей получаемой сервером информации.

Вывод статистических показателей и анализируемой информации должен происходить на главной странице пользователя. Также должен быть подключен Telegram-бот для отправки в сообщениях показателей выборки в определённое время суток, а также оповещений при отклонениях показателей от установленной нормы в течение дня для быстрого реагирования.

Размещается приложение на любом гаджете (ПК, смартфон и пр.) при условии постоянно включенного устройства или на арендованном удаленном VDS-сервере.

Использование сервера позволит создавать различные пользовательские сценарии (например, включение/отключение элементов по расписанию), выводить срочную информацию (например, утечка газа),



представлять ежемесячные отчёты со статистическими данными (например, с показаниями приборов учёта для оплаты коммунальных услуг) или отслеживать запросы (например, включение/выключение электроэнергии или газа в помещении).

### **Основная платформа системы управления умным домом**

Arduino - это платформа для разработки устройств на базе микроконтроллера, на простом и понятном языке программирования в интегрированной среде Arduino IDE [2]. Низкая стоимость, высокое качество компонентов и простота в использовании делает эту марку плат лидером на рынке. Основную плату можно выбрать из элементов таблицы 1.

Таблица 1 – Список контроллеров

Контроллер	Чип	Память	Порты и интерфейсы
Due	32-битный Cortex-M3 ARM SAM3U4E, 84 МГц	64+36Кб/256+256Кб	54 цифровых ИО (некоторые со спец. функциями) 12 аналоговых входов (с АЦП) 2 аналоговых входа
Uno	8-бит ATmega328 или ATmega8U2 (в новой версии), 16 МГц	2Кб/32Кб+1Кб EEPROM	14 цифровых ИО (6 могут использоваться как ШИМ) 6 аналоговых входов с АЦП, UART TTL, USB, разъемы для модулей расширения
Leonardo	8-бит ATmega32u4, 16МГц	2.5кб/32Кб+1Кб EEPROM	20 цифровых ИО (7 с функциями ШИМ) 12 аналоговых входов (могут работать как цифровые), USB, расширения
Mini	8-бит ATmega168, 16МГц	1Кб/16Кб+512б EEPROM	14 цифровых ИО (6 могут использоваться как ШИМ) 6 аналоговых входов с АЦП
Nano	8-бит ATmega168 или ATmega328, 16МГц	1Кб/16Кб+512б EEPROM	-
Mega	8-бит ATmega1280, 16МГц	8Кб/128Кб+4Кб EEPROM	54 цифровых ИО (14 с функциями ШИМ), 16 аналоговых входов с АЦП, 4 порта UART, USB

При выборе плат стоит учитывать тот факт, что не все они поддерживают интернет-соединение и в таком случае придется докупать модуль (например, W5500 Ethernet).

Нагрузка и функционал работы умного дома напрямую зависит от первоначального выбора контроллера, от чего и начинает варьироваться начальная стоимость АСУ.

Линии связи с контроллером могут быть двух типов: проводная и беспроводная. В качестве примера будем рассматривать использование линий передачи беспроводной связи. К преимуществам последней относятся:

- минимальное количество проводов;
- компактность;
- простота модификации и взаимодействия компонентов.

Из недостатков можно выделить:

- высокая стоимость;
- необходимость дополнительной защиты от хакеров;
- невысокая надежность соединения.

В конце сборки пользователю может быть предложена примерная схема (один из возможных вариантов) подключения элементов умного дома (рис. 1). Схема состоит из следующих элементов:

- климат-контроль (элементы контроля качества воздуха и управления вентиляцией);
- видеонаблюдение (камеры видеонаблюдения);
- охрана помещения (датчики движения у входной двери);
- автоматические жалюзи (жалюзи с шаговым двигателем);
- аудиосистема;
- автоматическое освещение (освещение с дистанционным управлением);
- умные розетки (розетки с возможностью работы по сценарию);
- контроллер (главный связующий контроллер Arduino);

- сервер (удаленный от контроллера хост);
- база данных (для записи и хранения информации о работе экосистемы).

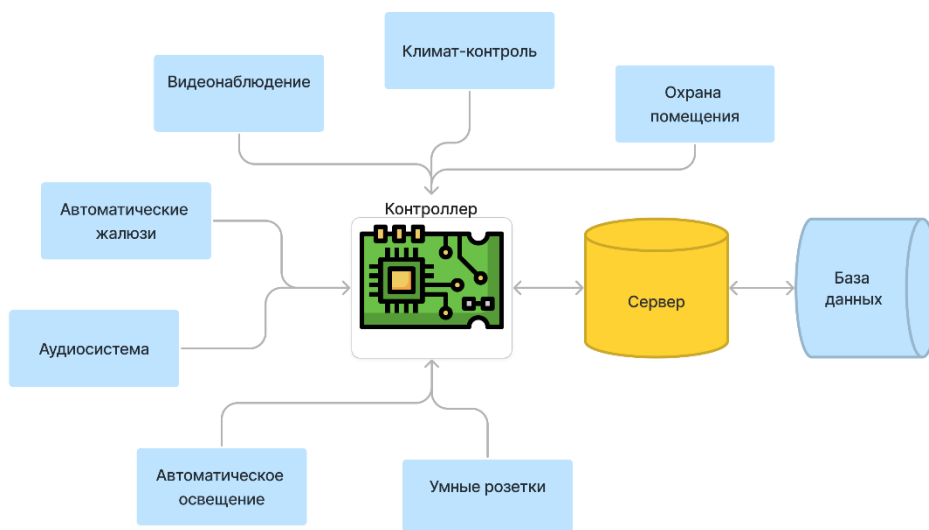


Рисунок 1 – Общая схема подключения элементов экосистемы «умный дом»

Такая структура схемы изначально позволяет в будущем лучше распределять нагрузку и проще отлаживать систему. Одной из вариаций модификаций этой схемы будет связка нескольких контроллеров под каждую пару элементов. Все контроллеры также будут соединены с сервером, но не между собой. Это позволит оптимизировать общую работу умного дома во время расширения. Стоит подчеркнуть, что явным недостатком будет значительное повышение стоимости и сложности настройки работы контроллеров.

### **Выбор и разработка компонентов экосистемы**

Компоненты экосистемы — модули для Arduino, настроенные таким образом, чтобы исполнять ту или иную поставленную задачу. Не стоит забывать, что умный дом — это условный конструктор. Самая главная проблема — правильное написание прошивки для главного контроллера. В нашем случае, прошивка должна:

- быть «гибкой» — удобно дополняться/изменяться/подстраиваться;
- быть понятной для будущих корректировок;
- быть максимально оптимизированной ввиду малого объема памяти и вычислительной мощности микроконтроллера.

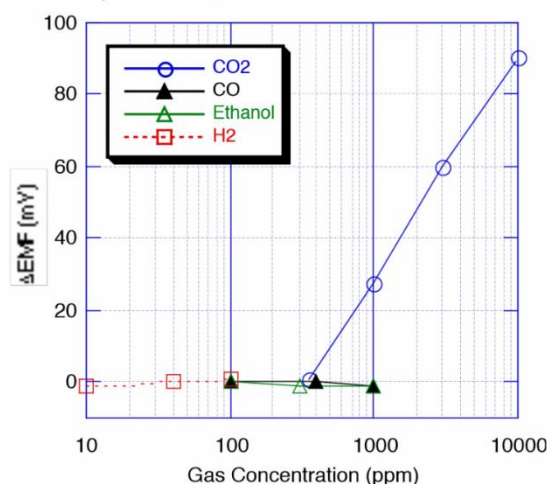
Код для Arduino пишется на языке программирования, базирующемся на C/C++, и использует библиотеку AVR-Libc 2.1.0. Для начала работы пользователю достаточно иметь базовые знания синтаксиса языка. Благодаря наличию тысяч элементов для контроллера и возможности взаимодействия с ними через прошивку открывается возможность добавления в наш умный дом любой идеи. Рассмотрим возможные сценарии.

1) **Охрана помещения.** Используя датчики движения, при условии дополнительной логической разработки схемы управления, АСУ сможет отслеживать и уведомлять о непрошенных гостях. В ванной комнате можно установить датчики утечки воды для предотвращения затопления и экстренного устранения прорыва труб (при желании можно соединить этот модуль с автоматическим перекрытием воды).

2) **Анализ и очистка воздуха.** Датчик температуры и влажности DHT-22 и датчик углекислого газа TGS4161 можно подключить и настроить таким образом, чтобы, например, в случае увеличения концентрации углекислого газа в помещении автоматически включалось проветривание (самый дешевый вариант - это поставить в вентиляции компьютерные вентиляторы). Все датчики, несмотря на их низкую стоимость, показывают вполне точные данные (см. рис. 2) [3].

3) **Автоматическое освещение.** На основе датчиков движения или датчиков освещенности можно предусмотреть автоматическую подсветку нужных зон (например, в коридоре или на балконе). Для жилых комнат такой вариант будет непрактичным из-за отключения датчиков во время бездействия. В последних лучше использовать другие варианты: включение/выключение освещения через веб-интерфейс либо пульт, а также автономная работа включения/выключения осветительных приборов при входе/выходе человека.

**Sensitivity Characteristics:**



**Humidity Dependency:**

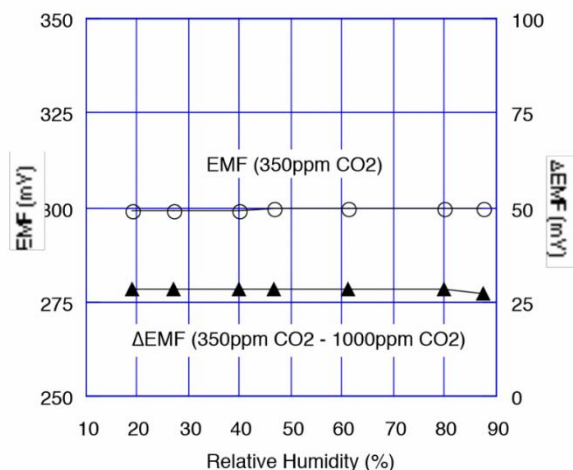


Рисунок 2 – Технические характеристики датчика TGS4161

4) **Автоматические жалюзи.** Подключив ко всем жалюзи на окна шаговый двигатель 28BYJ-48 и датчик освещенности, можно больше ни о чем не беспокоиться [4]. Утром, в зависимости от времени восхода солнца, они будут сами открываться, а вечером — закрываться.

5) **Видеонаблюдение.** Хорошим дополнением ко всему будет установка пары видеокамер в помещении для удаленного наблюдения за обстановкой. Новые модели видеокамер имеют встроенное подключение по WiFi, а поэтому и вывод прямой трансляции на сайт через сервер не составит труда.

6) **Аудиосистема.** Подключение аудиосистемы к умному дому позволит использовать пользовательские сценарии работы с плейлистами. Можно предусмотреть автоматическое включение спокойной, расслабляющей музыки в конце дня или совместить аудиосистему с плавным открытием жалюзи перед утренним пробуждением, чтобы начинать день можно было максимально комфортно.

7) **Умные розетки.** Можно управлять всеми розетками в помещении дистанционно. Создание умной розетки на базе Arduino выходит в разы дешевле, чем готовые аналоги (однако, спайка и настройка их в помещении займет уйму времени).

Учитывая изложенное выше, представляется возможность предложить пользователю приблизительную общую схему взаимосвязи компонентов АСУ (рисунок 3).

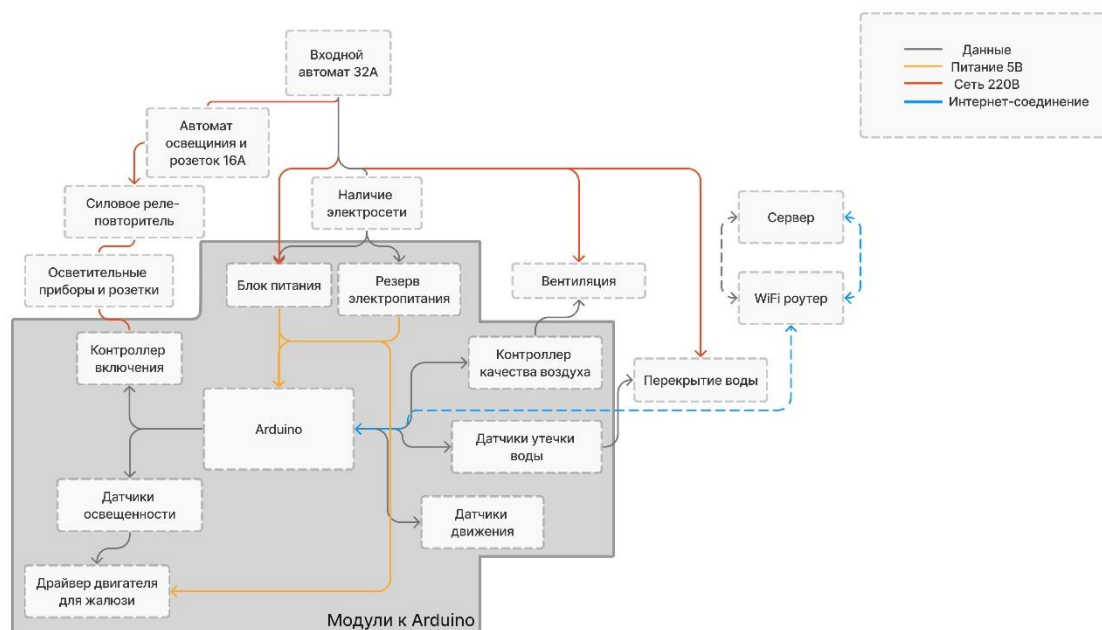


Рисунок 3 – Схема взаимодействия компонентов умного дома

Прямоугольными блоками на схеме изображены модули и компоненты предложенного проекта умного дома. Линиями и стрелками соответствующих цветов представлены предполагаемые каналы и направления потоков статистических данных, электроэнергии и обмена протоколами данных посредством компьютерной сети

Интернет. Данная разработка может быть внедрена как в единичное производство, так и выполняться в условиях домашнего конструирования как базовая модель, рассчитанная на пользователей разных возрастных категорий, с возможной подстройкой под потребности клиента.

## **Выводы**

В рамках данной статьи выполнена разработка проекта умного дома. Были проанализированы преимущества и недостатки существующих автоматизированных систем управления приборами домашней экосистемы, в том числе рассмотрены варианты использования контроллеров АСУ и её компонентов. Авторами предложены варианты решений возникающих проблем при разработке, сборке и автономном подключении умного дома.

Предложенный в данной статье проект умного дома способен конкурировать с другими дорогостоящими аналогами по функционалу. Самостоятельная разработка и сборка комплекса может осуществляться инженерами-любителями, что исключает необходимость дополнительной платы мастеру за модификацию экосистемы при сравнении с существующими аналогами или её модернизацию.

Возможность расширения списка компонентов управления умным домом, а также техническая и программная реализация идеи будут рассмотрены в последующих работах.

## **Литература**

1. Общие сведения о Node.js [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/windows/dev-environment/javascript/nodejs-overview>. - Загл. с экрана.
2. Bloom J.W. Изучаем Arduino: инструменты и методы технического волшебства. Под ред. Сависте Е.А.: Санкт-Петербург, 2015. – 331 с.
3. Автономная охранная система на Arduino + GSM [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/493514/>. - Загл. с экрана.
4. Умные шторы своими руками [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/374285/>. - Загл. с экрана.

***Руднев К.К., Дмитриук Т.Г. Создание автономной системы управления умным домом.***  
*Статья посвящена разработке проекта аппаратно-программного комплекса для автономной системы управления умным домом на базе платформы Arduino. Разработанная система позволит в автоматизированном режиме управлять экосистемой помещения.*

***Ключевые слова:*** умный дом, сервер, автоматизация, схема, плата.

***Rudnev K.K., Dmitriuk T.G. Creation of an autonomous smart home management system.***  
*The article is devoted to the development of a hardware and software complex project for an autonomous smart home control system based on the Arduino platform. The developed system will allow you to automatically manage the ecosystem of the room.*

***Keywords:*** smart home, server, automation, circuit, board.

## **Исследование программных средств извлечения текстовой информации из аудио файлов**

**Баляба Я.В. <sup>\*1</sup>, Рычка О.В. <sup>\*2</sup>**

<sup>\*1</sup>магистрант, Донецкий национальный технический университет,  
y.baliaba22@gmail.com

<sup>\*2</sup>к.т.н., Донецкий национальный технический университет,  
olga\_rychka@mail.ru

***Баляба Я.В., Рычка О.В. Исследование программных средств извлечения текстовой информации из аудио файлов. Рассмотрена актуальность разработки. Определены назначение и задачи. Проводится сравнительная характеристика существующего программного обеспечения. Рассмотрены основные достоинства и недостатки существующих систем. Сделаны выводы.***

***Ключевые слова:** транскрибация, аудио файлы, текст, информация, извлечение текста.*

### **Введение**

Просмотреть по диагонали текст и понять, есть ли там что-то полезное и интересное, часто быстрее, чем послушать пятиминутное аудио или просмотреть видео.

Многие из нас предпочитают тексты во многих ситуациях: когда находятся в общественных местах без наушников и на работе.

Владельцам сайтов, которые постоянно выкладывают видео и аудиоконтент, стоит помнить о тех, кто по тем или иным причинам просмотреть или прослушать информацию не может. Зато может ее прочитать.

Расшифровка аудио и видеофайлов — занятие нудное и не самое приятное. Но мы живем в век технологий, а, значит, к нашим услугам множество возможностей, чтобы транскрибировать аудио и видеофайлы.

В статье рассматривается сравнение существующих программ для извлечения текста из аудио файлов.

Целью работы является, проведение исследования методов извлечения текстовой информации из аудио файлов.

### **Назначение и задачи средств извлечения текстовой информации из аудио файлов**

Транскрибация — это процесс прослушивания аудиозаписи и расшифровки услышанного вручную или автоматически. Предположим, есть аудио или видеофайл с объяснениями. Нужно прослушать и перевести аудио и видео в текст, сохранив смысл и авторскую подачу. Можно долго слушать, запоминать, делать паузы в аудио или видео и печатать текст на слух или использовать специальную программу.

Её задача — грамотно перевести и отредактировать аудио и видео в другой формат: убрать слова-паразиты, неправильные фразы, паузы и шум. Готовый текст должен состоять из логичных фраз, построенных по правилам русского языка [1].

### **Анализ существующих систем для извлечения текстовой информации из аудио файлов**

В настоящее время, на рынке информационных предложений представлен большой выбор готовых решений по извлечению текста из аудио файлов. Рассмотрим основные из них.

#### **1. Онлайн-конвертеры**

Для преобразования видео или аудио в текст можно использовать онлайн-сервисы. Качество расшифровки у платных сервисов обычно на порядок выше, чем у бесплатных. Но если качество записи аудио изначально высокое, то и бесплатные конвертеры могут хорошо справиться с задачей преобразования видео в текст онлайн и расшифровкой аудио [2].

#### **Google Документы**

Использование сервиса Google Документы — самый простой и доступный любому способ перевести видео в текст или расшифровать аудио.

Чтобы перевести видео в текст онлайн, необходимо открыть и запустить функцию «Голосовой ввод» во вкладке «Инструменты». Если одновременно включить воспроизведение видео или аудио рядом с микрофоном, система начнет сама набирать текст (см.рис.1).

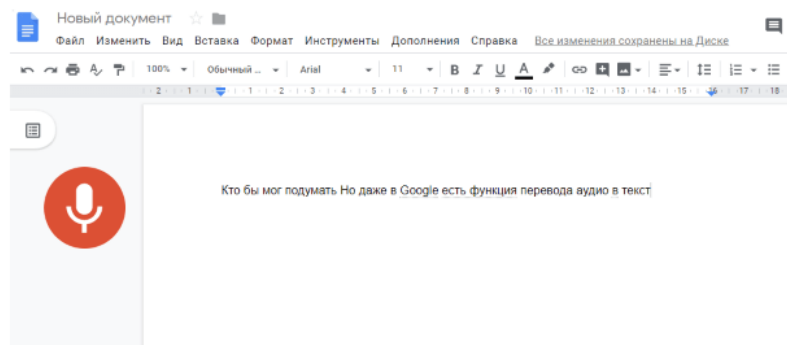


Рисунок 1 - Интерфейс Google Документ

Увы, но сервис очень плохо справляется с тихой и шумной диктофонной записью, но отлично с диктовкой в микрофон. Главный недостаток – работает только в активном окне Google Docs, то есть наговорить что-то с другой вкладки или включить запись на компьютере не получится [3].

Достоинства:

- бесплатный продукт;
- автоматическое сохранение текста;
- возможность сразу же отредактировать материал.

Недостатки:

- медленная расшифровка;
- распознает не все слова: требуется хорошая диктовка, чтобы получить качественный текст;
- не сможет распознать запись из другой вкладки браузера или плеера.

Speechpad

Speechpad – бесплатный онлайн-инструмент, с помощью которого можно расшифровать голосовую запись. Работает исключительно со звуком из микрофона, поэтому следует позаботиться о качестве оборудования.

Поддерживается Google Chrome, но есть приложения для iOS и Android. Также возможна интеграция в Windows, Mac и Linux, чтобы обеспечить голосовой ввод в любом текстовом поле. Четкий звук понимает достаточно хорошо, плохой – крайне посредственно. Интерфейс представлен на рисунке ниже (см. рис.2).

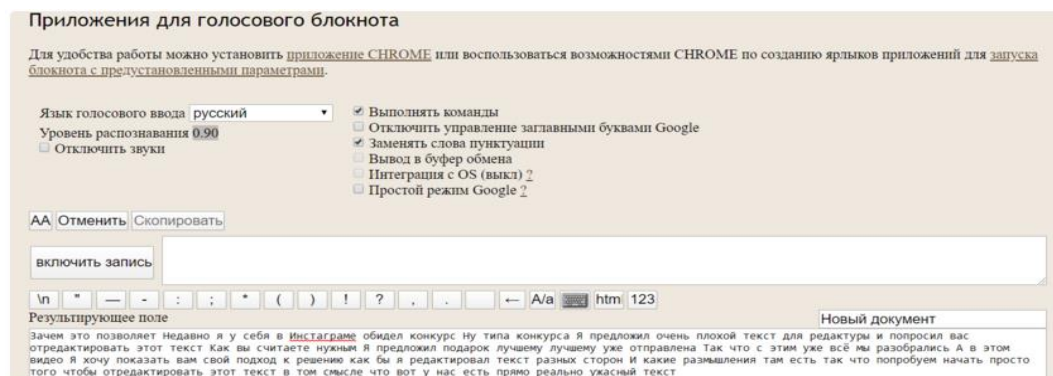


Рисунок 2 - Интерфейс программы Speechpad

Достоинства:

- бесплатная программа на русском языке;
- можно отредактировать текст;
- есть инструкции по работе с сервисом;
- воспринимает звуки с соседних вкладок браузера – можно работать с одного устройства.

Недостатки:

– транскрибация из файла требует хорошего качества звука, в ином случае расшифровка будет неполной.

RealSpeaker

RealSpeaker – платный сервис для перевода аудиофайлов в текстовый контент. Работает исключительно с готовыми файлами, поэтому использовать микрофон, как в предыдущих случаях, не получится.

Перевод аудио в текст бесплатен только в том случае, если запись длится не более 1.5 минут. Далее – 8 руб. за минуту, максимальная продолжительность аудио – 180 мин. Поддерживается более 40 языков, включая русский.

Для работы с сервисом достаточно выбрать язык озвучки, загрузить файл, рассчитать время расшифровки и оплатить услугу (см.рис.3).

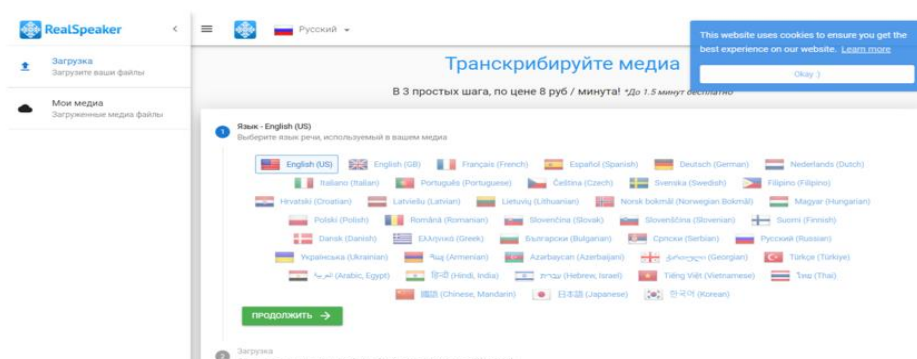


Рисунок 3 - Интерфейс сервиса RealSpeaker

Достоинства:

- есть возможность работать с файлами;
- простой в использовании сервис.

Недостатки:

- платный сервис;
- не позволяет надиктовать текст в микрофон;
- в течение 24 часов файл доступен всем, и его нельзя скрыть.

## 2. Программное обеспечение для десктопных устройств

Существует множество платных и бесплатных версий ПО, которые позволяют конвертировать видео в текст. И аудио тоже. Принцип работы и функционал такого программного обеспечения схож с онлайн-сервисами. Но если онлайн-конвертеры требуют подключения к интернету, ПО можно использовать всегда, когда необходимо перевести звук из видео в текст или преобразовать в текст аудио.

Voco

Эта отечественная программа для перевода видео в текст работает на ОС Windows. Она станет помощником для тех пользователей, которые не хотят расшифровывать аудио файлы самостоятельно. Текст можно наговорить в микрофон, взять готовую аудиозапись или звуковую дорожку из видео. Программа работает со всеми популярными аудио форматами. Программа осуществит и перевод видео в текст. Качество расшифровки — довольно высокое. Единственный и главный недостаток — программа платная [4]. Интерфейс программы представлен ниже на рисунке (см.рис.4).

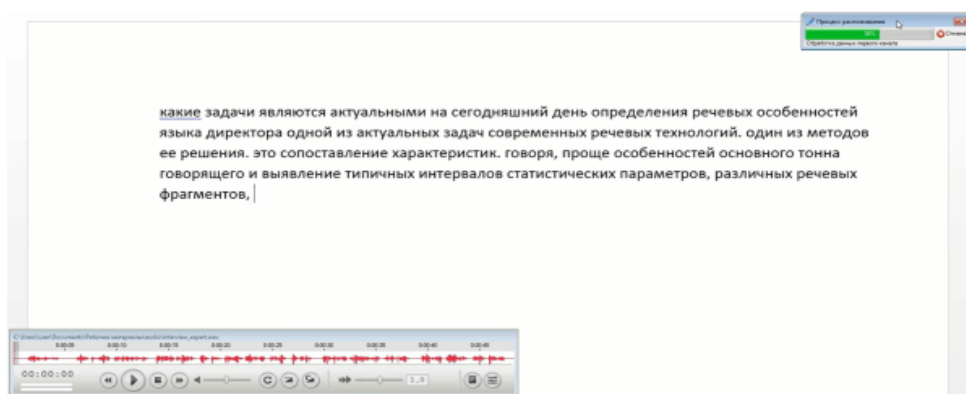


Рисунок 4 - Интерфейс программы Voco

## Выводы



На сегодняшний момент существует потребность в создании новых общедоступных систем извлечения текста. С помощью которых можно будет получать текст бесплатно и в кратчайшие сроки, а также редактировать его. Самое главное сделать эту программу оффлайн для того, чтобы если нет доступа к интернету пользователи смогли воспользоваться ей. Также очень важно чтобы чтение текста программой было максимально точным даже если исходный файл плохого качества.

Стоит уделить внимание интерфейсу, он должен быть максимально понятным и простым, чтобы у пользователя не было проблем с пониманием как работать в данной программе.

В ходе исследования были рассмотрены достоинства и недостатки организации методов извлечения текстовой информации из аудио файлов.

### **Литература**

1. Перевод речи в текст [Электронный ресурс] – Режим доступа: <https://www.compgramotnost.ru/programmy/perevod-rechi-v-tekst> – Загл. с экрана.
2. Разработка приложения для распознавания речи [Электронный ресурс] – Режим доступа: <https://medium.com/nuances-of-programming/-1863d143552a> – Загл. с экрана.
3. Google Документы [Электронный ресурс] – Режим доступа: <https://texterra.ru/blog/polnoe-rukovodstvo-po-google-docs.html> – Загл. с экрана.
4. VOCO. Windows-приложение для преобразования речи в текст [Электронный ресурс] – Режим доступа: <https://www.speechpro.ru/product/programmy-dlya-raspoznavaniya-rechi-v-tekst/voco> – Загл. с экрана.

**Баляба Я.В., Рычка О.В. Исследование программных средств извлечения текстовой информации из аудио файлов.** Рассмотрена актуальность разработки. Определены назначение и задачи. Проводится сравнительная характеристика существующего программного обеспечения. Рассмотрены основные достоинства и недостатки существующих систем. Сделаны выводы.

**Ключевые слова:** транскрибация, аудио файлы, текст, информация, извлечение текста.

**Balyaba Ya. V., Rychka O. V. Study of software tools for extracting textual information from audio files.** The relevance of the development is considered. Purpose and tasks are defined. The comparative characteristic of the existing software is carried out. The main advantages and disadvantages of existing systems are considered. Conclusions are drawn.

**Keywords:** transcription, audio files, text, information, text extraction.



## Влияние правил естественного языка на концепции программирования на примере WEB-разработки

А.И. Мащенко

студент, Технологический институт (филиал) ДГТУ в г. Азове,  
atidstu.ru

*Мащенко А.И. Влияние правил естественного языка на концепции программирования на примере WEB-разработки. В статье производится анализ влияния правил естественного языка на концепции и парадигмы языков программирования. Для сравнения выбрана сфера WEB-разработки, в технологиях и программных языках которой, приводится схожесть в конструкциях и структуре с естественными языками. Приводятся примеры, когда сравнение двух видов языков может привести к ошибкам в понимании работы программного языка. Перспектива развития изучения данного вопроса, возможно, поможет определить будущий вектор развития языков программирования.*

**Ключевые слова:** язык программирования, естественный язык, WEB-разработка, декларативность, императивность, структуры данных, части речи, операторы, ключевые слова, асинхронность.

### Введение

Нотация, определяющая семантику и синтаксис любого языка программирования, имеет своё начало в естественном языке, на котором человек коммуницирует ежедневно. Чем более язык программирования высокоуровневый, тем более в нём смысловых конструкций (описывающих структуры данных и операции над ними) и тем виднее влияние правил построения предложений и текста естественного языка. Это помогает программисту лучше понимать программы и правила их построения.

Язык программирования является искусственным языком, который спроектирован для специфических задач. В отличие от естественного, он применяется только для управления информационными узлами. В нём нет средств для выражения чувств и эмоций, а его команды имеют лишь одну интерпретацию. Однако, в упрощённом виде, программный язык принял в себя множество других концепций естественного языка. Ключевые слова и построению, и по смыслу относятся к различным частям речи; обращение к структурам данных производится по правилам декларативных предложений; используются различные временные модели и т.д.

Анализ влияния правил естественного языка на концепции и парадигмы программирования и является целью данной научной работы.

В качестве основы для исследования, выбрано направление WEB-разработки, которое является одним из самых востребованных. А, так как его технологический стек состоит из языков программирования, основанных на английском синтаксисе и английских словах, то сравнение будет производиться относительно этого языка.

### Части речи и ключевые слова

Английский, как и любые другие языки, имеет понятие частей речи. Однако, в контексте его использования в WEB-разработке, остаются лишь основные: имя существительное, имя прилагательное, глагол, местоимение, предлог и союз.

Классический стек технологий для создания сайтов и WEB-приложений [1] выглядит так:

- HTML (HyperText Markup Language) – язык разметки;
- CSS (Cascading Style Sheets) – язык стиля;
- JavaScript – язык программирования.

В HTML имена тегов относятся к имени существительному; например, <head>, <body>, <img>, <table>. Как в естественных языках имена существительные указывают на предметы, вещи, лица и так далее, так и в программном языке разметки они указывают на объект страницы.

CSS описывает стили разметки в виде ссылки на конкретный объект с некоторым количеством свойств типа ключ-значение; например: `body {background: black;}`, где сначала указывается объект к которому применяется стиль, а затем ключ регулирующий параметры фона со значением чёрного цвета. Ключ выражает существительное, а его значение – прилагательное.

Язык разметки и стилей почти полностью состоят из содержательных слов, а именно существительных и прилагательных, и они указывают на объект и выражают его состояние, без внимания к тому, как он будет

преобразован в него. Поэтому, эти языки относят к декларативной парадигме программирования, в которой описывается желаемый результат, а логика его получения скрыта.

JavaScript выделяется тем, что в нём есть как содержательные так и функциональные слова. Поэтому, он является противоположным им, то есть императивным. Для его работы и решения определённых задач, требуется точное описание последовательности шагов. Хотя в его состав также входят содержательные слова, в основном глаголы, однако, он содержит большое количество функциональных слов. Именно они требуются для логической связи программных блоков и операторов.

Классифицировать ключевые слова языка JavaScript можно следующим образом:

- существительные – function, var, class и другие;
- глаголы – import, export, extends, return, break, continue, delete, switch, try, catch, throw, yield и другие;
- предлоги – for, in, else;
- союзы – if, while;
- местоимения – this;

Как сказано выше, помимо множества функциональных слов, для JavaScript (как и любых других императивных языков программирования) свойственно большое количество глаголов. Это позволяет манипулировать блоками данных и их сущностями.

### ***Проблема понимания работы оператора «this» в контексте английского языка***

В английском языке местоимение в предложениях понимается в контексте предшествующего текста. Например, в предложениях: «John fixed the bike. He's been riding it all day», понятно, что местоимение «he» указывает на Джона, а «it» на велосипед.

Однако, в JavaScript часто невозможно оператором this указать на то, что будет выглядеть естественно относительно кода программы, так для this существуют различные правила, указывающие на разные контексты. Потому, аналогия с естественным языком не очень хорошо работает.

Например, программист Кайл Симпсон [2] указывает на «одно распространенное ошибочное представление — что this для функции ссылается на саму функцию». Если бы this ссылался на область видимости, которая статична и состоит всегда из одного и того же количества переменных, то этот оператор был бы ближе к своему аналогу из естественного языка, то есть, местоимению. Но, так как он относится к контексту выполнения, который динамичен и изменяется в зависимости от того, как вызывается функция, то this не может являться фиксированной ссылкой на функцию, как местоимение является ссылкой на конкретный субъект.

### ***Типы предложений и структура блоков операторов***

Каждое предложение в тексте на английском языке имеет свою структуру и относится к тому, или иному виду. Так как повествовательные предложения, вопросительные, повелительные и восклицательные имеют определённые эмоции, правильнее было бы рассмотреть простейшие декларативные предложения на английском языке.

Структура субъект-предикат-объект, достаточно часто применяется в программировании: child.go(school), где субъект соответствует модели данных (класс или объект), предикат соответствует методу, а объект – параметру метода.

Но, важно понимать, что программный язык имеет две очень похожие формы - утверждение (присвоение) и выражение:

– school.color = 'white' – «школа белая», это утверждение, и оно не имеет возвращаемого значения.

Происходит присвоение значения одному из полей объекта;

– school.color === 'white' – «школа белая?», это выражение вернёт true или false;

Выражение чаще всего применяется в условиях, например: if (school.color === 'white') paintRed(school), где при возвращении значения true вызовется функция paintRed.

Можно заметить, что простейшие декларативные предложения на естественном языке вполне соответствуют построению ключевых слов и операторов в программном коде. Однако, выражения в программировании имеют логическое значение, а значит требуют к себе особого внимания.

### ***Времена естественного языка выражающие асинхронные операции***

Очень важной концепцией в программировании является асинхронность, которая обычно используется для обмена данными между информационными узлами [3]. Например, после заполнения какой-то формы на веб-странице, пользователь нажимает на кнопку и данные отправляются на сервер. Так как интернет соединение не стабильно, то на отправку и получение ответа потребуется какое-то время. Из-за этого нарушается стандартный – синхронный порядок выполнения программы.

В синхронном коде особого разнообразия во временах нет: в основном, используется настоящее простое время для наименования функций и переменных, потому процесс выполнения вполне понятен. Однако, с

асинхронностью, естественный порядок хода программы нарушается, потому что мы доходим до значений, которых у нас может ещё не быть, так как время получения ответа неизвестно. Поэтому необходимы более сложные временные конструкции.

Для работы с асинхронными вызовами, обычно, используют объекты Promise [4]. Это «обещание» позволяет контролировать состояние выполняемой асинхронной операции. Завершается (исполняется) оно вызовом одной из двух функций: resolve или reject.

То есть, объект Promise имеет три состояния:

- pending – начальное состояние ожидания выполнения асинхронного запроса. Объявляется при инициализации обещания, до вызова управляющих функций;
- resolved – состояние при вызове функции resolve. Ответ получен;
- rejected – состояние при вызове функции reject. Ошибка.

Для перемещения между состояниями используются глаголы resolve и reject. Слово pending, выражает стартовое состояние и имеет форму настоящего непрерывного времени (present continuous), а потому очень подходит для выражения ожидания ответа от асинхронной операции. А вот, слова resolved и rejected имеют форму настоящего совершенного вида, и описывают уже завершившиеся к настоящему моменту события, а именно принятый ответ или произошедшую ошибку. Таким образом, временные модели естественного языка очень органично перенесены в концепцию программы.

### **Заключение**

Исходя из выше сказанного, действительно можно обнаружить множество схожих концепций и влияние естественного языка на программный:

- ключевые слова программного языка можно отнести к тем или иным частям речи;
- декларативные языки программирования состоят из содержательных слов, а императивные имеют также и функциональные для манипуляций над структурами данных;
- структура обращения к данным схожа с построением декларативных предложений на английском языке;
- для ключевых слов, обрабатывающих асинхронные операции, используются различные формы настоящего времени.

Конечно, далеко не во всех концепциях программный и естественный язык схожи. Но, взгляд на программирование с позиции гуманитарных наук, может помочь в понимании основополагающих принципов и концепций программного языка. Ведь человеку легче учиться, когда он сталкивается с немного изменёнными, но привычными для него понятиями из естественного языка. Также, такие исследования помогут определить будущий вектор развития языков программирования.

### **Литература**

1. Cal Henderson, Building Scalable Web Sites // O'Reilly Media, Sebastopol 2006. 1-4, 6-14 p.
2. Кайл Симпсон, Познакомьтесь, JavaScript. 2-е международное издание [2022] // Издательский дом Питер, Санкт-Петербург. 2022. 128 с.
3. Eloquent JavaScript: Asynchronous Programming [Электронный ресурс] URL: [https://eloquentjavascript.net/11\\_async.html](https://eloquentjavascript.net/11_async.html)
4. Learn JavaScript: Promise [Электронный ресурс] URL: <https://learn.javascript.ru/promise>

**Мащенко А.И. Влияние правил естественного языка на концепции программирования на примере WEB-разработки.** В статье производится анализ влияния правил естественного языка на концепции и парадигмы языков программирования. Для сравнения выбрана сфера WEB-разработки, в технологиях и программных языках которой, приводится схожесть в конструкциях и структуре с естественными языками. Приводятся примеры, когда сравнение двух видов языков может привести к ошибкам в понимании работы программного языка. Перспектива развития изучения данного вопроса, возможно, поможет определить будущий вектор развития языков программирования.

**Ключевые слова:** язык программирования, естественный язык, WEB-разработка, декларативность, императивность, структуры данных, части речи, операторы, ключевые слова, асинхронность.

**Mashchenko A.I. The influence of natural language rules on programming concepts on the example of WEB development.** The article analyzes the influence of natural language rules on the concepts and paradigms of programming languages. For comparison, the sphere of WEB development is chosen, in

*the technologies and programming languages of which, similarity in designs and structure with natural languages is given. Examples are given when comparing two types of languages can lead to errors in understanding the operation of a programming language. The prospect of developing the study of this issue may help determine the future vector of development of programming languages.*

**Keywords:** *programming language, natural language, WEB development, declarativeness, imperativeness, data structures, parts of speech, operators, keywords, asynchrony.*

## Подходы к разработке 2D платформера на Unity

А.В. Суворина<sup>\*1</sup>, А.В. Боднар<sup>\*2</sup>

<sup>\*1</sup> бакалавр, Донецкий национальный технический университет,  
anya.suvorina@yandex.ru

<sup>\*2</sup> к.э.н, доцент, Донецкий национальный технический университет,  
linabykova13@ya.ru, , Orcid: 0000-0002-1800-0178, SPIN-код: 5068-7409

*Суворина А.В., Боднар А.В. Подходы к разработке 2D платформера на Unity. В статье проведен анализ среды разработки Unity, её преимуществ и недостатков, рассмотрены особенности жанра платформер, его появления и развития, главные представители во все периоды. Раскрыты основы создания 2D платформера на Unity.*

**Ключевые слова:** Unity, игра, 2D, C#, платформер, движок.

### Введение

Компьютерные игры актуальны уже более 30 лет, и с каждым годом их распространенность возрастает. Самыми популярными жанрами являются экшены, шутеры, головоломки и платформеры. Платформер – жанр компьютерных игр, в которых основу игрового процесса составляет передвижение по платформам, преодоление препятствий, сбор предметов, необходимых для завершения уровня.

### Среда разработки игр Unity

В игровой индустрии существует множество разных движков. Одни разработаны под конкретную игру, на других можно делать игры конкретного жанра (шутеры от первого лица, платформеры, гонки), а есть универсальные, вроде Unity, которые открывают разработчикам больше возможностей.

Unity – межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на почти 30 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. В нем есть разные компоненты для работы с графикой, анимацией, физикой объектов, звуком, шаблонами и скриптами. Это удобный бесплатный инструмент для начинающих разработчиков.

У Unity есть два основных преимущества перед другими ведущими инструментами разработки игр: производительный визуальный рабочий процесс и мощная межплатформенная поддержка. [1] В нём заложено использование гибкой модульной системы при создании сцен и персонажей в игре. К недостаткам же относят сложности при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

Unity сразу идёт в комплекте со средой разработки. Это одновременно и редактор кода, и работа с графикой, и логика поведения предметов в игровой сцене. Поведение игровых объектов контролируется с помощью компонентов, которые присоединяются к ним. Unity позволяет создавать свои компоненты, используя скрипты. Они позволяют активировать игровые события, изменять параметры компонентов, и отвечать на ввод со стороны пользователя любым способом. Unity изначально поддерживает язык программирования C#. Взаимодействие предметов в игре между собой и с окружающей средой называется физикой. В Unity есть уже готовая физика и шаблоны для создания своих правил.

В отличие от механизмов наследования, объекты в Unity создаются посредством объединения функциональных блоков, а не помещения в узлы дерева наследования. Такой подход облегчает создание прототипов, что актуально при разработке игр. В редакторе имеется система наследования объектов; дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта.

### Появление жанра платформер

Платформеры – это жанр игр, в которых основным геймплейным элементом является путешествие по уровням, которые разбиты на множество различных платформ. Их наполнение может различаться – в каких-то играх игроков ждут только всевозможные препятствия и ловушки, а в других – еще и различные противники. Существуют как двухмерные платформеры, где игроки видят все "сбоку", так и трехмерные, где приходится гораздо тщательнее рассчитывать каждый свой шаг, чтобы не упасть и не погибнуть.

Наиболее распространенные варианты движения в этом жанре – ходьба, бег, прыжки, атака и лазание. Многие схватки с боссами требуют решения головоломки некоторого вида, когда игрок должен изучить характер

реакций и атак босса и определить последовательность действий, учитывающих этот характер и ведущих к победе. [2] Многие платформеры также имеют ямы или другие особенности игрового мира, которые убивают персонажа, если он упадет в них.

Платформерные игры возникли в конце 70-х – начале 80х, когда игровые консоли не были достаточно мощными. «Space Panic» 1980 года часто называют первым платформером в истории, но в ней отсутствует один из важнейших параметров жанра – прыжки. В ней впервые появились лестницы, позволявшие перемещаться по уровню. Действие игры не выходило за пределы одного экрана. Персонаж лазал вверх и вниз по лестницам или прыгал с платформы на платформу, часто сражаясь с противниками и преодолевая препятствия. Большим шагом вперед для жанра стало изобретение боковой прокрутки, которая позволяла игровому персонажу всегда оставаться в пределах видимости пользователя, перемещая локацию по горизонтали влево или вправо, имитируя полноценное ощущение движения. Самым ярким представителем платформера того времени до сих пор считается видеоигра Super Mario Bros., выпущенная в 1985 году фирмой Nintendo. Игра была наполнена большими и сложными уровнями, и стала примером для последующих создателей игр.

### **Развитие платформеров**

К тому моменту, когда даже минимальные свежие идеи в индустрии стали заканчиваться, освоение 3D-технологий разом изменило вектор развития всех актуальных жанров конца XX века.

Термин «трехмерный платформер» может обозначать или геймплей, включающий все три измерения, или использование трехмерных полигонов в реальном времени для отрисовки уровней и героев, или и то, и другое. Появление трехмерных платформеров принесло изменение конечных целей некоторых платформеров. В большинстве двумерных платформеров нужно было достичь на уровне только одной цели, однако во многих трехмерных играх уровень необходимо исследовать, собирая предметы. Это дало возможность эффективного использования трехмерных областей и вознаграждало игрока за исследование уровня. Но не все трехмерные игры были такими, примером является Crash Bandicoot. Эта игра оставалась верной традиции двумерных платформеров и в ней использовались плоские уровни, в конце которых располагалась игровая цель. [3]

Также появились изометрические платформеры, которые, вероятно, являются поджанром двух- и трехмерных платформеров. Они отображают трехмерную сцену с помощью двумерной графики, которая показывает мир с четко ориентированной камерой без учета перспективы. Ранними примерами изометрических платформеров являются игра 1983 года Congo Bongo для аркадных автоматов и 3D Ant Attack для ZX Spectrum.

Однако к XXI веку платформеры снова ушли в затяжной кризис. Придумать что-то новое в узких рамках жанра, истерзанного технически обновленным подходом к разработке проектов, было сложно. Постепенно он раскололся на широкий ряд подвидов и одновременно начал тонуть в других геймплейных направлениях.

### **Платформеры в наше время**

Если классические игры предлагали однообразный игровой процесс, то современные обладают множеством свежих дополнительных механик. Игры платформеры продолжают развиваться, предлагая пользователям всё новый и новый игровой опыт.

Inside – игра от создателей Limbo, такая же мрачная и загадочная. Игроки берут на себя управление маленьким мальчиком, который стремится вырваться из некоего тоталитарного мира. Герою необходимо преодолеть препятствия, столкнуться с опасностями и рискнуть жизнью, чтобы прийти к неожиданному финалу. Секретная концовка игры запутывает игрока, что в сочетании с приятной картинкой, затягивающим геймплеем и необычным сеттингом делает его одним из лучших представителей жанра платформер.

Ori and the Blind Forest – красивый и сложный платформер, посвященный приключениям сказочного существа по имени Ори, которое должно вылечить умирающий лес и победить зло. Игра отличается высокой сложностью, но это только побуждает проходить игру с новыми силами: она тяжелая не из-за плохого геймдизайна или сильных противников, а благодаря грамотно продуманным уровням и проработанной акробатике, которые поддаются не каждому игроку.

Terraria – 2D аналог Minecraft, действия которого разворачиваются в открытом мире, разделенном на биомы и подземные царства. Игра, разработанная с использованием набора инструментов Microsoft XNA, даёт в распоряжение игрока процедурно генерируемый и изменяемый двумерный мир. Помимо добычи ресурсов и строительства сооружений, в игровом процессе Terraria также уделяется внимание исследованию мира, поиску сокровищ и сражениям с противниками. В игре предусмотрен кооператив.

Hollow Knight – инди-платформер, который отправляет игрока в мир насекомых: в роли главного героя здесь выступает жук, путешествующий по королевству и сражающийся с монстрами. В игре представлен большой открытый мир, в часть локаций которого невозможно попасть без определенных инструментов и способностей, что побуждает более тщательно исследовать уровни и возвращаться в уже посещенные места. В остальном игра представляет собой экшен-платформер с интересным сюжетом и боевой системой.

Самый последний популярный платформер – It Takes Two стал одной из лучших кооперативных игр. Он представляет собой приключение двух кукол, в которых превратилась ругающая семейная пара. Ключевое

отличие от других игр – огромное количество механик. Игра будет и типичным платформером, и шутером, и аркадной гонкой. Игру отлично дополняют визуальный стиль и увлекательное повествование.

### Создание 2D платформера на Unity

Первым шагом в создании 2D платформера будет установка Unity с официального сайта. Для разработки первой игры можно установить бесплатную версию Unity Personal. В качестве редактора кода можно использовать Visual Studio Code, Notepad++ или Sublime Text.

Запускаем Unity Hub и создаём новый проект по кнопке «New». В параметрах выбираем Template – 2D, имя проекта и папку для хранения проекта. Нажимаем кнопку «Create», после чего видим главное окно. Рассмотрим видимые панели. Toolbar (панель инструментов) – содержит элементы управления объектами в игре. Панель предварительного просмотра игры содержит кнопки «Play», «Pause» и «Step». Hierarchy (дерево объектов) – панель, содержащая объекты, расположенные на сцене. По умолчанию игровые объекты в панели выстраиваются в порядке добавления на сцену. Project (браузер проекта) – панель для управления ресурсами проекта. Inspector (свойства объекта) – все игры на Unity состоят из игровых объектов (GameObject), к которым подключаются различные компоненты. У каждого объекта есть свои свойства (размер, положение, скрипты). Через данное окно и происходит управление ими. Scene (сцена) – окно управления расстановкой объектов в игровом мире. Game (игра) – окно предварительного просмотра самой игры.

Начнем с создания платформ. Перетащим изображение платформы в папку Assets, а из неё – на сцену. Можно изменить её размеры и положение в компоненте Transform, а также вручную. Для взаимодействия с окружающим миром необходимы компоненты из вкладки Physics2d. Добавим спрайту платформы Polygon Collider. Для того чтобы не добавлять одинаковые компоненты на каждом уровне, в Unity есть механика под названием Prefabs. Она позволяет создать из игрового объекта префаб, который можно использовать много раз и менять, при этом изменятся все его экземпляры. Для создания префаба достаточно перенести объект из иерархии во вкладку Assets.

Для создания персонажа так же перетащим изображение персонажа в папку Assets. Добавим на верхнюю часть персонажа Box Collider для столкновений с препятствиями, а Circle Collider расположим на уровне ног для движения по наклонным поверхностям. Для того чтобы на героя действовала гравитация, добавим ему в меню Physics2D компонент Rigidbody2D (рис. 1). В добавленном компоненте есть параметры: Mass – масса объекта; Linear Drag и Angular Drag – линейное и угловое сопротивление; Gravity Scale – коэффициент гравитации объекта; Collision Detection – метод, используемый физическим движком для проверки столкновения двух объектов; Sleeping Mode – состояние сна, в котором первоначально будет находиться объект; Interpolate – физическая интерполяция, используемая между обновлениями.

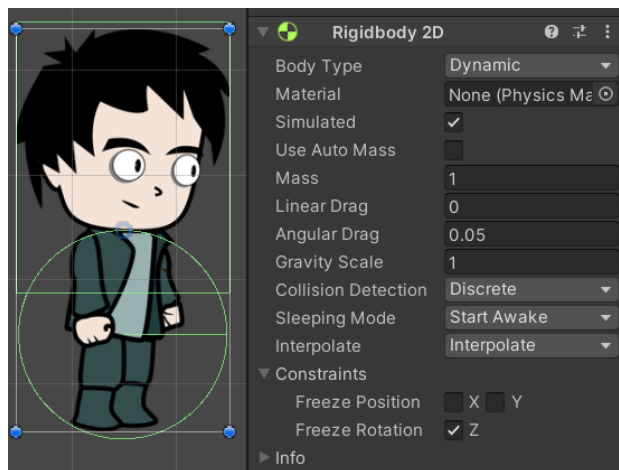


Рисунок 1 – Пример параметров объекта персонажа

Следующим шагом научим персонажа двигаться. Создадим скрипт на объекте персонажа. Объявим в нем переменные, отвечающие за свойства объекта: его скорость, силу прыжка, сторону, в которую он повернут, и положение относительно земли. Если в проекте нет необходимости изменять переменную на панели Inspector или обращаться к ней из других сценариев, то такой переменной можно задать модификатор private. [4] В скрипте автоматически создаются функции Start() и Update(). Метод Start() выполняется, когда игра запускается, а Update() выполняется каждый кадр. В методе Update() при нажатии на кнопки «Вправо» и «Влево» будем двигать объект при помощи векторов. Проверка стороны, в которую смотрит персонаж, будет поворачивать его спрайт с помощью Vector3. Vector3 – это структура, в которой хранятся значения X, Y и Z, Vector3 может использоваться для передачи положения, поворота или масштаба игрового объекта. [5] При нажатии на кнопку «Вверх» и условии, что объект находится на земле, он будет делать прыжок при помощи вектора Vector2.

Для создания врагов так же необходимо перетащить изображение в папку Assets. Добавим в меню Physics2D компонент Rigidbody2D, чтобы на врага действовала физика. Создадим скрипт врагу, в котором объявим его скорость и направление. В функции Update() будем двигать врага с помощью векторов Vector2, а в функции OnTriggerEnter2D() будем менять его направление с помощью вектора Vector3 при столкновении с невидимыми коллайдерами с двух сторон от него. Добавим врагу тег «Enemy», а в скрипт персонажа добавим проверку на столкновение с объектом с данным тегом, при которой уровень будет начинаться заново.

Необходимо сделать обработку падения персонажа с платформы. На местах, где можно упасть с платформы, разместим невидимые элементы Box Collider и поставим во вкладке Assets галочку напротив Is Trigger (рис. 2). Создадим скрипт персонажа, где в функции OnTriggerEnter2D() при столкновении с коллайдером уровень будет начинаться заново.

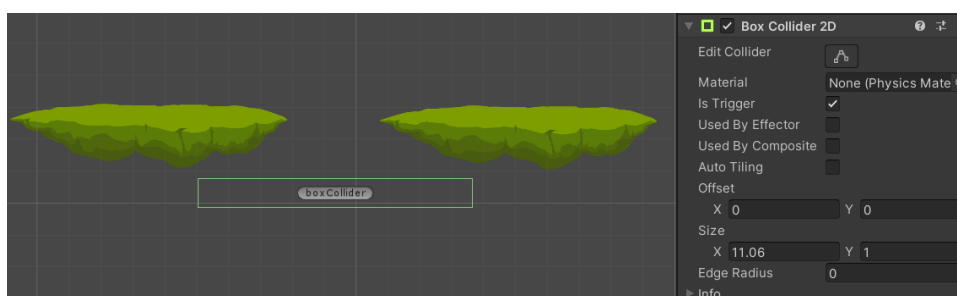


Рисунок 2 – Параметры Box Collider

Чтобы камера во время игры не стояла статично, а плавно двигалась за персонажем, создадим на ней скрипт. Main camera, на которую повешен этот скрипт, следует за объектом, который указан в публичной переменной target при помощи векторов и положения объекта. Укажем в переменной target персонажа.

## Выводы

Таким образом, в статье проведен анализ среды разработки Unity, ее достоинств и недостатков. Также проведен анализ особенностей жанра платформер, его возникновения и его примеров во все периоды существования. Описанные подходы к разработке 2D платформера на Unity отвечают всем требованиям к данному жанру.

## Литература

1. Хокинг, Дж. Unity в действии. Мультиплатформенная разработка на C# / Пер. с англ. И. Рузмайкиной. — СПб.: Питер, 2016. — 336 с.
2. Бонд, Дж. Unity и C#. Геймдев от идеи до реализации / Пер. с англ. А. Киселева. — СПб.: Питер, 2022. — 928 с.
3. Платформер // Виртуальная лаборатория [Электронный ресурс]. — Режим доступа: <https://vlab.fandom.com/ru/wiki/Платформер>
4. Харрисон Ф. Изучаем C# через разработку игр на Unity / Библиотека программиста — СПб.: Питер, 2022. — 400 с.
5. Введение в скрипты Unity // Jwinters [Электронный ресурс]. — Режим доступа: <https://jwinters.ru/unity3d/scripting-1/>

**Суворина А.В., Боднар А.В. Подходы к разработке 2D платформера на Unity.** В статье проведен анализ среды разработки Unity, её преимуществ и недостатков, рассмотрены особенности жанра платформер, его появления и развития, главные представители во все периоды. Раскрыты основы создания 2D платформера на Unity.

**Ключевые слова:** Unity, игра, 2D, C#, платформер, движок.

**Suvorina Anna, Bodnar Alina. 2D platformer Unity development approaches.** The article analyzes the Unity development environment, its advantages and disadvantages, discusses the features of the platformer genre, its appearance and development, and the main representatives in all periods. The basics of creating a 2D platformer on Unity are revealed.

**Keywords:** Unity, game, 2D, C#, platformer, engine.



## Анализ основных тенденций разработки компьютерных игр

Боднар А.В.<sup>\*1</sup>, Павлов М.Ю.<sup>\*2</sup>.

<sup>\*1</sup> к.э.н, доцент, Донецкий национальный технический университет,  
linabykova13@ya.ru

<sup>\*2</sup> бакалавр, Донецкий национальный технический университет,  
vigototheroads@mail.ru

**Боднар А.В., Павлов М.Ю. Анализ основных тенденций разработки компьютерных игр.**  
*В работе рассматриваются основные тенденции разработки современных компьютерных игр, что поможет сделать выводы о том, с чего нужно начинать разработку игр, как современный рынок влияет на создание игр, как были сформулированы тенденции компьютерных игр и их влияние на всю индустрию.*

**Ключевые слова:** Разработка, тенденция, механика, игра, Obsidian, Краудфандинг, влияние рынка

### Введение

Исследование основано на личном анализе с учетом персонального игрового опыта и на изучении игровой индустрии, что включает в себя критические обзоры, статьи, а также книги о внутренней разработке игр. При условиях постоянно развивающейся индустрии подход требует не одностороннего рассмотрения проблемы, а комплексного, с учетом внешних факторов.

Методы, которые использовались в исследовании, позволяют взглянуть на проблему со многих сторон, в первую очередь с позиции разработчика, который старался над выходом своей игры, переживал краш и уделял проекту всего себя; и немаловажно то, как проблема разработки влияла на пользователей, которые и будут проводить своё время в игре и которые её ждали, в некоторых случаях вкладывались в разработку посредством краудфандинга. Перед исследователем стояла задача – изучить путь, который прошло множество игр от этапа идеи до выхода в свет; те проблемы, с которыми сталкивались разработчики и те, которые относились к внутренней среде и рынку и которые усложняли и без того многогранный процесс разработки.

### Рождение тенденций при разработке игр

Разработка игры – это словно снимать кино, только перед каждым новым фильмом, вам потребуется вновь собрать совершенно новую камеру или, делать игру – как строить дом во время землетрясения, – именно так высказывался о разработке владелец студии игр Obsidian, Фергюс Уркхарт – человек, работавший над знаменитой игровой франшизой Fallout и проделавший огромный путь от тестера до главы студии. По мнению Фергюса[1], такие игры как Fallout, Mass Effect, The Witcher 3 внесли вклад в тренды игровой индустрии, в особенности из-за проработанности мира и особенностей сеттинга.

Игра Fallout, разработанная студией Black Isle, которой владела компания Interplay, является одной из первых игр, в полной мере использующих нелинейность повествования. Данный пример цепляет своим постапокалиптическим сеттингом и, что немаловажно, доказывает, что игра может быть большой и сложной, а прохождение – неоднобразным. Каждую проблему можно решить разными способами в зависимости от имеющегося оборудования, степени изученности мира и прокачки персонажа. История создания этой игры очень сложная: сначала над игрой трудился лишь один программист, Тимоти Кейн. За 6-ть месяцев он написал игровой движок и адаптировал правила игры под систему GRUPS – на то время популярную ролевою игру за авторством Стива Джексона[2]. Вооружившись энтузиазмом, он продолжил работать и убеждать компанию в том, что игра того стоит. Через год разработки Кейну выделили 15 сотрудников, и процесс действительно пошёл, набирая обороты[2]. К проекту приложили руки люди, которые до сих пор ценятся в индустрии, такие как Крис Тейлор, Фергюс Уркхарт, Леонард Боярский. В будущем это позволило выпустить легендарную серию, которая живет и по сей день.

Fallout является идейным продолжением Wasteland, славится своим подходом к чёрному юмору и системой S.P.E.C.I.A.L., которая складывается из заглавных букв характеристик персонажа (силы, восприятия, выносливости, обаяния, интеллекта, ловкости и удачи) и позволяет из сеанса в сеанс создавать уникальных героев. На фоне игр с DnD-системой и фэнтези-сеттингом игра стала считаться инновационной. Именно Fallout

породил множество различных RPG, став одним из основных столпов жанра и показав, что попытки создания чего-то нового вопреки рискам могут привести к удивительным результатам. К числу идейных наследников относится одна из лучших RPG 2015 года, *Pillars of Eternity*, разработанная студией Obsidian под руководством Джона Сойера и Адама Бреннеке.

*Pillars of Eternity* проделала огромный путь от публикации идеи на Kickstarter в 2012 году и до выхода полноценной игры 26 мая 2015 года [3]. Само решение обратиться к краудфандинговой платформе носило репутационные риски. При этом ранее, 12 мая 2012 года, студия пережила крах: компания Microsoft прекратила финансирование важной игры *Stormlands* и тем самым отменила её выход в свет [4]. Из компании уволили те 26 человек, которые и работали над игрой. Вскоре работа в компании превратилась в ад: несмотря на то, что она предоставила свои услуги почти всем известным игровым студиям, всё было бесперспективно, ведь сотни проектов получили отказ. Дело дошло до того, что Соьер и Бреннеке повели главу студии, Уркхарта, в его любимое кафе и буквально уговаривали попробовать обратиться за помощью на Kickstarter, в противном случае грозясь уволиться из компании. И они получили разрешение.

На решение Уркхарта так же повлияло то, что ранее Крис Авеллон, один из создателей *Fallout New Vegas*, публично похвалил Kickstarter и провел опросы в команде Obsidian на тему того, какой бы проект они поддержали. Получив согласие главы, Бреннеке на несколько дней заперся в своем офисе, чтобы создать идеальный проект для Kickstarter [4]. По самым позитивным оценкам, игра не должна была собрать более 2 миллионов долларов [4], а минимальным денежным порогом для выхода игры в свет была объявлена сумма в 1,1 млн долларов. Вся студия с замиранием ждала нажатия «красной» кнопки 14 сентября 2012 года [3], и вот процесс запустился. Страницу тут же перезагрузили, а на счету уже было 800 долларов. Бреннеке обновил страницу ещё раз, и сумма возросла до 2 700 долларов [4]. Через минуты счет перевалил за пятизначное число. Всего через неделю была собрана минимальная стоимость разработки, и амбиции игры резко взлетели. С увеличением денежного фонда на странице игры на Kickstarter стали появляться новые цели и обещания новых игровых элементы, которые должны были разрабатываться по степени сбора средств. Одной из таких целей стало создание второго огромного города, с которым возникли проблемы. Невозможно было просчитать и стоимость левел-дизайна, по крайней мере, до начала разработки. Поэтому Бреннеке и Соьер не смогли вложиться в первоначальный бюджет и сроки.

В планах было выпустить вертикальный срез игры к 2013 году [3]. В этот промежуток решались многочисленные творческие вопросы о том, что в игре стоит оставить, а что – убрать, чтобы сократить издержки и ускорить выпуск игры. Ко всему прочему, требовалось устранить уже существовавшие к этому моменту баги, но главной проблемой оказался сам сюжет. Им занимался талантливый сценарист Эрик Фенстермейкер, которому параллельно поручили писать сюжет для игры *South Park: The Stick of Truth*. Хотя он и успел набросать несколько глобальных сюжетных концепций и прописать большую часть мира, без сторонней помощи дописать всё не представлялось возможным, поэтому студия приняла решение пригласить полноценного сценариста – Кэрри Пэйтель [4]. В конце 2013 года вышел тизер [3], на который возложили роль поддержки ожиданий спонсоров. И несмотря на то, что на тот момент проект был далёк от завершения, разработчикам удалось реализовать многие основные моменты игры.

Помимо этого, для данной игры использовали новую программу для работы с 3D-графикой: вместо всем привычной Infinity Engine команда перешла на Maya, что также замедляло процесс [4]. Но финальный продукт выглядел в разы перспективней и лучше, чем его предшественники. В тизере продемонстрировали основные механики и отдали дань прошлому, в том числе – *Baldur's Gate* и другой классике, разработанной на движке Infinity Engine.

Как итог, разработчики сделали ставку на краудфандинговую платформу, превращавшую обычных поклонников и всех заинтересованных лиц в сообщество инвесторов, но общую привлекательность данной идеи омрачали финансовые расчёты. Проблема предварительной оценки стоимости разработки, как выяснилось, является особо острой именно для индустрии компьютерных игр, а не какой бы то ни было другой.

К концу Kickstarter-компания, 16 октября 2012 года, Obsidian подсчитали всю собранную за это время сумму. На счету проекта было 3 986 794 доллара [3], а с учетом дополнительных поступлений через собственный сайт и PayPal общая касса разработки игры достигла отметки в 5.3 миллиона долларов [4]. Всего за полгода компания переломила линию судьбы своих неудач. Теперь компания не балансирует на грани краха и может финансово, за счёт оформившейся франшизы, позволить создавать проекты мечты.

Близился релиз игры и выставка E3, но проблемы оставались нерешенными. Первый город, Бухта Непокорности, был замечательным, однако очень дорогим с точки зрения разработки. Несмотря на это, разработчики дали обещание и на второй такой город – необязательный, но уже ожидаемый элемент игры. Разработка ещё сильнее замедлилась, и игра не смогла выйти в срок.

Студия приняла гениальное решение. Так, на выставке E3 в игру могли сыграть все желающие при соблюдении важного условия: управлять персонажем будут не они, а разработчики, тогда как игрок будет отдавать им команды. Это было сделано с целью затушевать баги и подводные камни тестовой версии игры. Одним из таких камней был двойной клик по экипировке, вызывавший сброс характеристик персонажа до первоначальных значений. Разработчики обратились к главе и сделали запрос на внеплановое финансирование

из внутренних резервов компании, на что Уркхарт дал четкие указания: если игра не выйдет в мае 2015 года, то Бреннеке и Соьер – уволены. Вопреки недоверию, выделенные средства окупились с лихвой, и фанаты дождались финальной версии игры и получили продукт, удовлетворявший их представления. [4].

Pillars of Eternity – уникальный проект, который позволил студии Obsidian выбраться из финансовой ямы и показать, что риском и борьбой до победного конца можно выкарабкаться из людей сложной ситуации. Данный опыт доказывает важность планирования и примерного предварительного представления о работе проекта, а также обучения персонала современным технологиям и освоения ими новых инструментов.

Краудфандинг стал оправданным риском, и даже несмотря на то, что компания была не первой в деле по сбору средств, она выделилась среди других кандидатов, когда сумела показать, как можно эффективно использовать данный инструмент в кризисной ситуации. Obsidian, большая и известная студия, не предавшая ожиданий поклонников, также запустила тренд краудфандинга, выставив свой проект в 2012 году. Это можно увидеть на следующей диаграмме (см.рис 1):



Рисунок 1 – Результат сборов видеоигровых проектов на площадке Kickstarter

### ***Влияние современного рынка на процесс разработки игр***

Индустрия компьютерных игр развивается огромными темпами: за 2020-й год рынок вырос на 23.1%, согласно отчету Google[7]. Этот отчет содержит прогноз на 2024 год: рынок разрастется до 218,7 млрд долларов, что существенно привлечет инвесторов.

Также нужно учитывать, что игры не боятся эксплуатировать механику pay-to-win, что в прямом переводе означает «плати и победи». Данная механика запустила тренд и стала частью множества современных игр. Однако, финансовый успех таких проектов, как правило, сопровождается отрицательным резонансом в игровых сообществах. В пример можно привести Diablo Immortal, ставшую финансово успешной, но потерявшую все доверие игроков при оценке в 0.3 балла из 100 на Metacritic[8]. И этому тренду следует и другая игра от Blizzard Activision, Overwatch 2. Вторая часть является прямым идейным продолжением первой. Она перенимает все основные игровые аспекты предшественницы и немного дополняет их, однако при этом игроки лишаются возможности сыграть в первую часть. К тому же, хоть сиквел и стал бесплатным, в нём появилась механика pay-to-win. Она влияет на получение новой героини Кирико, которая хоть и достаётся игроку вместе с бесплатным BattlePass, но с некоторыми оговорками. Так, в платной версии пропуска Кирико дается сразу, а в бесплатном вам для этого потребуется играть две недели без перерыва.

Таким образом, механика pay-to-win позволяет облегчить себе игровой опыт, тогда как её игнорирование, на которое имеет право каждый игрок, заставляет вас «работать». Это одна из проблем, которую породил рынок: игра стала походить не на своего легендарного предшественника, Overwatch, а превратилась в его аналог с тонной багов. Именно поэтому аудитория очень плохо отзывается об Overwatch 2. Этому свидетельствует оценка 1.5 баллов из 100 на Metacritic[9]. Еще пару лет назад было сложно представить, что такой легендарный проект как Overwatch сможет превратиться в подобную игру.

### ***Инновации, породившие тенденции***

Инновации в игровой индустрии задают тенденции: так, например, до 1996 года 3D-графика в играх не применялась. Открытие 3D-технологий для сферы компьютерных игр отозвалось всплеском новых проектов, прародителем которых считается Quake. Но тенденции запускаются не только новыми технологиями.

Так, игра Р.Т за авторством Хидео Кодзимы поразила сердца миллионов игроков, и это позволило есть создать целый поджанр в играх жанра hotdog, который и был назван в её честь. Казалось бы, Р.Т., недорогая в разработке игра, состоящая из одной зацикленной локации, не привнесла в технологический портфель индустрии

чего-то существенно нового, но виртуозное исполнение на простых принципах и механиках породило инновационную концепцию. Игра умело эксплуатирует местность, постепенно окружая тебя теплом и комфортом, а после планомерно растворяет дружескую атмосферу, когда заставляет ощутить на себе ужас клаустрофобии. Соавтором игры выступал Гильермо Дель Торо, мастер ужаса[10]. Повторяющиеся действия вводят игрока в состояние гипноза, делая его уязвимым, – и это и есть тот самый новый концептуальный приём, лежащий за гранью механик и проникающий в область психологии. В нём игрока принуждают последовательно повторять одни и те же действия, намереваясь «зациклить» его, чтобы затем выбить человека из колеи.

Другой пример свидетельствует о том, как, казалось, совсем маленькое изменение полностью перевернуло понимание экшен игр. До 2005 года все игры, в которых упор делался на стрельбу, закрепляли камеру на игроке или же фиксировали её в определенной точке, пропорционально изменяя угол обзора при передвижении героя. Этим пользовались популярные франшизы, к числу которых относится DOOM, Resident Evil, Silent Hill и т. д. Но Resident Evil 4 пошёл по иному пути: он закрепил камеру над героем, а в процессе прицеливания приближал её сбоку.

## Заключение

Таким образом, в статье проведен теоретический анализ влияния рынка на разработку игры, влияния краудфандинга на процесс разработки, изменения тенденций под влиянием рынка и то, как разные факторы порождают тенденции разработки.

## Литература

1. The future of RPGs, according to Obsidian. Режим доступа: <https://www.pcgamer.com/obsidians-feargus-urquhart-on-the-future-of-rpgs/>
2. Fallout: A Postmortem- Режим доступа [https://ubm-twvideo01.s3.amazonaws.com/o1/vault/gdc2012/slides/Design%20Track/Cain\\_Timothy\\_Classic\\_Game\\_Postmortem\\_Fallout.pdf](https://ubm-twvideo01.s3.amazonaws.com/o1/vault/gdc2012/slides/Design%20Track/Cain_Timothy_Classic_Game_Postmortem_Fallout.pdf)
3. Pillars of Eternity (formerly Project Eternity) – Kickstarter Режим доступа: <https://www.kickstarter.com/projects/obsidian/project-eternity>
4. Шрейер.Д “ Blood, Sweat, and Pixels: The Triumphant, Turbulent Stories Behind How Video Games Are Made” // Шрейер.Д перевод с англ. Александра «Альфина» Голубева,. Издательство “БОМБОРА” Москва 2022 г – 368 стр.
5. Статья Skill Box media - Краудфандинг видеоигр. Режим доступа: <https://skillbox.ru/media/gamedev/kraudfanding-videoigr/>
6. Герасимова Е.И.Студент,Финансовый университет при Правительстве РФ ИНВЕСТИЦИОННЫЙ ПОТЕНЦИАЛ РЫНКА ВИДЕОИГР. Режим доступа: <https://cyberleninka.ru/article/n/investitsionnyy-potentsial-rynka-videoigr>
7. Google Beyond 2021:Where does gaming go next? Режим доступа:<https://games.withgoogle.com/reports/beyondreport/>
8. Metacritic - Diablo Immortal. Режим доступа: <https://www.metacritic.com/game/pc/diablo-immortal>
9. Metacritic - Overwatch 2 Режим доступа:<https://www.metacritic.com/game/pc/overwatch-2>
10. Hideo Kojima and Guillermo del Toro are making Silent Hills Режим доступа:[https://www.eurogamer.net/hideo-kojima-and-guillermo-del-toro-are-making-silent-hills?utm\\_source=twitter&utm\\_medium=social&utm\\_campaign=socialoomph](https://www.eurogamer.net/hideo-kojima-and-guillermo-del-toro-are-making-silent-hills?utm_source=twitter&utm_medium=social&utm_campaign=socialoomph)

**Боднар А.В. , Павлов М.Ю. Анализ основных тенденций разработки компьютерных игр**  
В работе рассматриваются основные тенденции разработки современных компьютерных игр, что поможет сделать выводы о том, с чего нужно начинать разработку игр, как современный рынок влияет на создание игр, как были сформулированы тенденции компьютерных игр и их влияние на всю индустрию

**Ключевые слова:** Разработка, тенденция, механика, игра, Obsidian, Краудфандинг, влияние рынка

**Bodnar A.V. Pavlov M.Y, Analysis of the main trends in video game development.** The article examines the main development trends in modern video games and helps to draw conclusions about where to start in game development, how the modern market influences game production, how trends in video games have been shaped and their impact on the industry as a whole

**Keywords:** Development, trend, mechanics, game, Obsidian, Crowdfunding, market influence

## Node.js как вариант высокоуровневого программирования

Егоров Б. Ю.<sup>\*1</sup>, Анохина И. Ю.<sup>\*2</sup>

<sup>\*1</sup> бакалавр, Донецкий национальный технический университет, bodya18x@mail.ru,

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
OrcID: 0000-0003-1073-6333, SPIN-код: 3555-6335  
ingatula@mail.ru

*Егоров Б.Ю. Анохина И.Ю. Node.js как вариант высокоуровневого программирования. В статье описаны подходы асинхронного и многопоточного написания серверного приложения. Представлена суть разработки с использованием архитектуры REST и рассказано о функциях обратного вызова.*

**Ключевые слова:** REST, node, асинхронность, поток, json, xml

### Введение

Node.js - это очень интересный новый способ для создания серверных приложений, веб-сайтов, серверов и сетевых клиентов. Данный язык спроектирован так, чтобы создать для пользователя очень удобную масштабируемость приложений, за счет умного использования функций JavaScript, сочетания асинхронного ввода-вывода, ну и использования его самого на стороне сервера.

Модель Node.js отлична от популярных, на текущий момент, языков для построения серверных приложений. Благодаря функциональной архитектуре у Node упрощается стиль программирования, повышается пропускная способность и что немало важно снижается потребление оперативной памяти. В данный момент Node развивается с большими темпами. Многие считают ее довольно удобной и эстетичной альтернативой к подходу в разработке серверов - на уже зарекомендовавших себя Python, PHP и т. п.

Основой Node является движок V8 от гугл, создающий из JavaScript модель с приоритетом на разработку серверных приложений. Node.js нет смысла прямо или косвенно сравнивать с ЯП (языками программирования), которые, в основном, используются для создания веб приложений (Python, PHP, Ruby on Rails, Java и т.д.). В то же время я считаю, что в будущем она вполне может заменить традиционные наборы для разработки веб приложений.

Платформа Node.js – это одна из популярных сущностей для построения масштабируемых REST API. А это значит, что её можно использовать не только для веб, но и для андроид, ios и везде где используется протокол rest.

### Архитектура REST и SOAP

REST – это стиль архитектуры приложения, когда обмен данными с фронтендом (визуальной частью) и бекендом (серверной, в нашем случае node.js) реализуется по определенному url адресу.

Каждая единица информации определяется URL – и, следовательно, что URL является идентификатором для единицы данных. Например, третья вещь, со шкафа будет иметь вид /item/3. Отсюда и получается строго заданный формат. Причем без разницы, в каком формате находятся данные по ГКД – это может быть и HTML, и отсканированная копия в виде jpeg-файла, и документ Microsoft Word.

Вот как это будет выглядеть на примере:

- GET /item/ — получить список всех вещей
- GET /item/5/ — получить вещь номер 5
- PUT /item/5 — изменить вещь (данные в теле запроса)
- POST /item — добавить вещь (данные в теле запроса)
- DELETE /item/5 — удалить вещь

SOAP расшифровывается как Simple Object Access Protocol, то есть простой протокол доступа к объектам. Протокол основан на XML, а точнее является расширением XML-RPC (eXtensible Markup Language Remote Procedure Call) для кодирования сообщений и в качестве транспорта – HTTP, HTTPS, FTP, SMTP.

Он применяется, когда одной программе необходим доступ к функциональности другой программы, находящейся где-то в сети на другом сервере. При этом программы могут быть написаны на разных языках и иметь чужеродную друг другу логику работы.

Первоначально был разработан XML-RPC компанией UserLans Software в сотрудничестве с Microsoft в 1998 году, его функциональность начала расширяться и через несколько циклов возник новый формат SOAP.

Как видно, в архитектура REST очень проста в плане использования. По виду пришедшего запроса сразу можно определить, что он делает, не разбираясь в форматах (в отличие от SOAP, XML-RPC). Данные передаются без применения дополнительных слоев, поэтому REST считается менее ресурсоемким, поскольку не надо парсить запрос чтоб понять, что он должен сделать и не надо переводить данные из одного формата в другой.

Приложения, созданные следуя клиент-серверной архитектуре, работают следующим образом — клиент запрашивает по нужному url данные у сервера, а он отправляет их в ответ. В данном случае сервер, вернув ответ, закрывает соединение.

### **Поточная работа приложения**

Клиент-серверная архитектура эффективна так как все запросы к серверу требуют определенное количество ресурсов (оу, процессорное время...). Для обработки каждого следующего запроса от клиента, сервер должен завершить обработку предыдущего. Когда серверу приходит новый запрос, он выделяет ему отдельный поток для его дальнейшей обработки.

Поток, простым языком, это ресурс, который CPU выделяет на выполнение какого-то блока инструкций. Сервер может обрабатывать определенное количество запросов одновременно, и их ограничение это потоки. Такая модель так же называется поток на запрос.

Для обработки N-го количества запросов серверу нужно N-ое количество потоков. Если сервер получит N+1 запрос, тогда он ожидает, пока один из потоков не освободится.

На рисунке 1 (Поточная работа сервера), сервер может обрабатывать до 4 запросов одновременно и когда он получает следующие 3 запроса, эти запросы должны ждать пока любой из этих 4 потоков не станет доступным.

Один из способов решения данной проблемы — это добавить больше ресурсов (памяти, ядер)

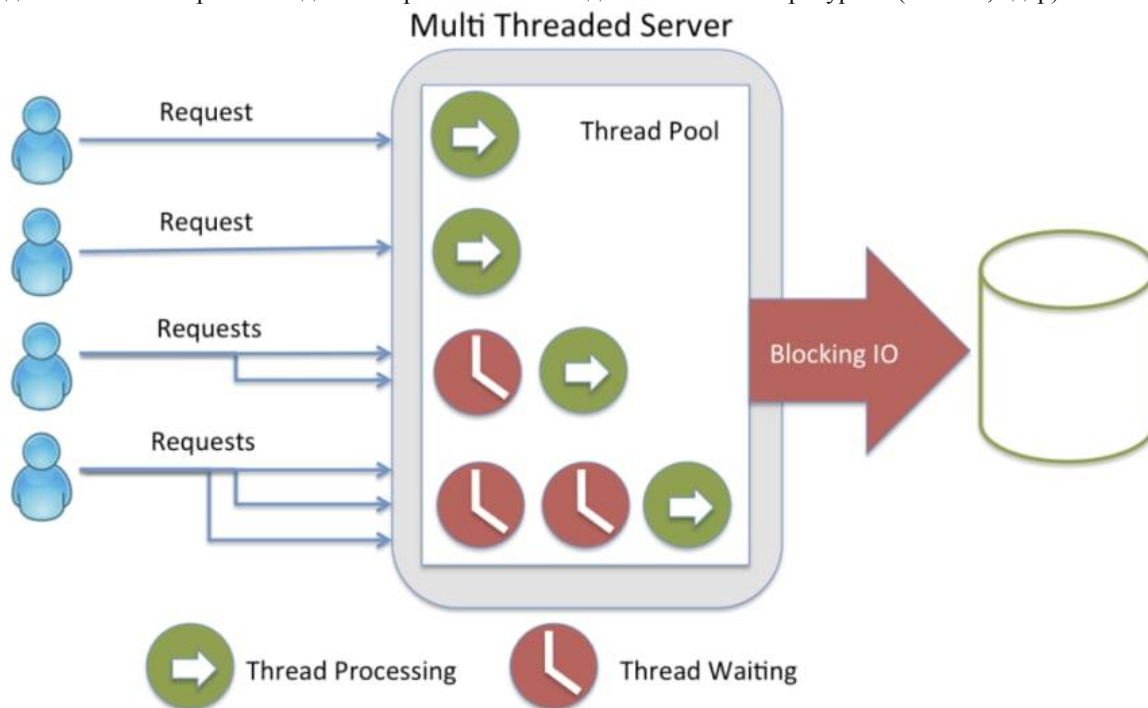


Рисунок 1 – Поточная работа сервера

### **Блокирующий ввод/вывод**

Число потоков это не единственная проблема. Но у меня был вопрос: почему один поток не может обработать несколько запросов одновременно? И ответ был в блокировке операций ввода-вывода.

К примеру, мы разрабатываем сайт-дропшипинг и нам нужна страница где пользователь может посмотреть список всех товаров. Реализация достаточно простая.

Пользователь переходит на страницу <http://my.dropshipping.com/> и сервер отправляет HTML файл со всеми товарами.

Но, что же происходит на сервере?

Когда пользователь отправляет запрос, метод выполняется, для обработки запроса. Код анализирует URL-адрес и переходит в метод или функцию. Поток работает.

Теперь метод выполняться.

Далее мы обращаемся к базе и получаем все продукты, но это блокирующая операция ввода/вывода. Поток ждёт.

Мы получили список продуктов.

Далее требуется отрендерить шаблон для будущей страницы, но сначала нам нужно прочитать все шаблоны. Поток ждёт.

Движок рендеринга шлёт ответ клиенту. Поток работает снова.

Поток освобожден.

Являются ли медленными операции ввода-вывода? Это зависит от конкретной задачи, см. таблицу 1.

Операции чтения с диска слишком медленные. В итоге операции ввода-вывода заставляют ждать и тратить ресурсы впустую.

Таблица 1 – Количество тактов на операцию

Операция	Количество CPU тактов
CPU Registers	3 такта
L1 Cache	8 тактов
L2 Cache	12 тактов
RAM	150 тактов
Disk	30 000 000 тактов
Network	250 000 000 тактов

### Функции обратного вызова

Функция обратного вызова — это функция, которая передается в качестве аргумента в другую функцию, а затем выполняется после завершения выполнения другой функции. Мы используем обратный вызов, чтобы убедиться, что код выполняется только после завершения асинхронной операции.

В течение длительного времени обратный вызов был самым распространенным механизмом написания асинхронного кода, но сегодня он считается устаревшим, поскольку затрудняет чтение кода. На этом шаге вы напишите пример асинхронного кода, используя обратные вызовы, чтобы использовать его в качестве исходной точки и продемонстрировать рост эффективности при использовании других стратегий.

Существует множество способов, позволяющих использовать обратные вызовы функции внутри другой функции. Обычно они имеют следующую структуру:

```
function asynchronousFunction([ Function Arguments ], [ Callback Function ]) {  
    [ Action ]  
}
```

Хотя синтаксис JavaScript или Node.js не требует, чтобы функция обратного вызова использовалась в качестве последнего аргумента внешней функции, это стандартная практика, которая упрощает определение обратных вызовов. Также, как правило, разработчики JavaScript используют **анонимную функцию** в качестве обратного вызова. Анонимные функции — это функции, которые не имеют имени. Обычно код читается гораздо лучше, когда функция определяется в конце списка аргументов.

#### Пример 1

```
const request = require('request');  
request('https://ghibliapi.herokuapp.com/films', (error, response, body) => {/*Работа функции*/});
```

При использовании функции request() мы передадим в нее два параметра:

- URL-адрес сайта, который мы запрашиваем
- Функция обратного вызова, которая обрабатывает любые ошибки или успешные ответы на выполненные запросы

Наша функция обратного вызова имеет три аргумента: `error`, `response` и `body`. Когда запрос HTTP выполняется, аргументам автоматически присваиваются значения в зависимости от вывода. Если не удалось выполнить запрос, тогда `error` будет содержать объект, а `response` и `body` будут иметь значение `null`. Если запрос будет выполнен успешно, HTTP-ответ сохраняется в качестве значения аргумента `response`.

Если наш HTTP-ответ возвращает данные (в данном примере мы получим JSON), данные сохраняются в качестве значения аргумента `body`.

#### Пример 2

```
var myCallback = function(data) {  
  console.log('got data: '+data);  
};  
var usingItNow = function(callback) {  
  callback('get it?');  
};  
usingItNow(myCallback);
```

В данном случае мы видим, что функцию обратного вызова мы можем не только в контексте примера 1 использовать. А и повсеместно когда функция принимает другую функцию.

### **Вывод**

Node.js это серверная платформа, работающая на движке Google Chrome V8, умеющего компилировать JavaScript код в машинный код. Node.js использует **событийную** модель и **неблокирующий ввод-вывод**, что делает его эффективным. Это не фреймворк, и не библиотека, это способ использования JavaScript.

Node.js использует неблокирующие ввод-вывод операции и это значит, что:

- Главный поток не блокируется операциями ввода-вывода.
- Сервер продолжает обслуживать запросы.
- Мы работаем с асинхронным кодом.

Цикл событий — на самом деле один поток. **Libuv** — библиотека реализует этот паттерн и является частью ядра Node.js. В event loop присутствует 6 фаз (так называемых тиков). В момент, когда циклу событий требуется выполнить операцию ввода-вывода он использует поток ОС с thread pool.

### **Литература**

1. The definitive Node.js handbook – Learn Node for beginners. Учебное пособие. [Электронный ресурс] – Режим доступа: <https://www.freecodecamp.org/news/the-definitive-node-js-handbook-6912378afc6e>.
2. Libuv: is a multi-platform [Электронный ресурс] – Режим доступа: <https://github.com/libuv/libuv>.
3. Everything you need to know about Node.js [Электронный ресурс] – Режим доступа: [https://dev.to/jorge\\_rockr/everything-you-need-to-know-about-node-js-lnc#theproblemwithcpuintensivetasks](https://dev.to/jorge_rockr/everything-you-need-to-know-about-node-js-lnc#theproblemwithcpuintensivetasks)
4. Асинхронный JS [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/JavaScript/Asynchronous>

***Егоров Б.Ю. Анохина И.Ю. Node.js как вариант высокоуровневого программирования. В статье описаны подходы асинхронного и многопоточного написания серверного приложения. Представлена суть разработки с использованием архитектуры REST и рассказано о функциях обратного вызова.***

**Ключевые слова:** REST, node, асинхронность, поток, json, xml

***Egorov B.Y. Anokhina I.Yu. Node.js as a variant of high-level programming. The article describes the approaches of asynchronous and multithreaded writing of a server application. The essence of development using the REST architecture is presented and the callback functions are described.***

**Keywords:** REST, node, asynchrony, threads, json, xml



## Основные технологии программирования мобильных систем

А.А. Носаченко<sup>\*1</sup>, С.В. Щедрин<sup>\*2</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
KerononS.learn@gmail.com

<sup>\*2</sup> ст. преп., Донецкий национальный технический университет,  
do010575ssv@gmail.com

*Носаченко А.А., Щедрин С.В. Основные технологии программирования мобильных систем. В докладе приведён обзор современных тенденций применения различных технологий программирования мобильных систем. Представлена краткая характеристика рассмотренных технологий. Текущие технологии создания программного обеспечения продолжают развиваться, разработка мобильных систем изменяется с большой скоростью, что требует регулярного исследования её технологий.*

*Ключевые слова:* доклад, программирование, кроссплатформенная разработка, мобильная система, Android, iOS, нативное приложение.

### **Введение. Предпосылки к исследованию технологий мобильного программирования**

Использование мобильных устройств растет уже много лет. Мобильные устройства все чаще используются во всем мире вместо настольных систем. Эта тенденция применима как для проектов, рассчитанных на отдельных пользователей, так и при разработке программных продуктов для предприятий.

Каждому пользователю мобильных устройств современного мира очевиден факт, что среди всех мобильных операционных систем подавляющее большинство устройств управляются именно системами Android и iOS. Обычно новые приложения разрабатываются для обеих систем для охвата наибольшего количества возможных пользователей. Однако Android и iOS сильно отличаются как с точки зрения пользователя, так и с точки зрения разработчика. Облегчает процесс создания приложений для разных операционных систем кроссплатформенное программирование.

Именно выбор между технологиями, основанными на кроссплатформенной разработке, стоит перед всеми начинающими разработчиками мобильных систем и является основанием для написания данного доклада.

### **Последние публикации. Развитие технологий разработки мобильных приложений**

Мобильная разработка как производственное направление – относительно новое и активно развивающееся явление. Открытые источники содержат малое количество статей, комплексно описывающих текущие основные технологии мобильной разработки, а сами публикации перестают быть актуальными за короткий промежуток времени. Потому на момент своего представления доклад имеет высокие степени важности и актуальности.

### **Рассматриваемая проблема. Важность изучения тенденций мобильной разработки**

Технологии разработки мобильных приложений – сложная тема. Широкий спектр доступных решений, инструментов и структурная сложность процесса программирования легко способны сбить с толку неподготовленного человека, решившего более сильно углубиться в мобильную разработку. По этой причине есть необходимость в изучении и представлении основных тенденций в выборе технологий программирования мобильных систем на данный момент.

### **Основная часть. Технические особенности реализации мобильных приложений**

Рассмотрение технологий мобильной разработки следует начать с разделения приложений на их основные типы с технической точки зрения:

- нативные приложения;
- веб-приложения;
- гибридные приложения.

Нативные приложения (англ. native app(lication)s) — это прикладные программы, которые были разработаны для использования на определённой платформе или на определённом устройстве [1]. Преимущество

нативных приложений в том, что они оптимизированы для соответствующей операционной системы. Пользовательский интерфейс полностью соответствует спецификациям платформы и совместим с системной средой. Еще одним важным преимуществом является то, что нативные приложения не имеют ограничений на доступ к аппаратному обеспечению конкретного устройства смартфона или планшета. Ко всем датчикам этих устройств можно обращаться напрямую через системные API. Нативные приложения обычно развертываются через магазины приложений. Если установлено собственное приложение, его можно запустить в автономном режиме без доступа к Интернету.

С другой стороны, есть веб-приложения. Это специальные приложения, адаптированные для использования на различных мобильных устройствах через сетевые сервисы. Однако доступ к системному оборудованию ограничен. Можно использовать простые функции, такие как служба определения местоположения. Отображение происходит в системном браузере и может лишь частично учитывать особенности отдельной платформы. Веб-приложения нельзя развертывать через магазины приложений. Они работают на сервере и поэтому требуют постоянного подключения к Интернету. Использование в автономном режиме стало возможно с прогрессивными веб-приложениями (PWA).

Гибридные приложения занимают позицию между веб-приложениями и нативными приложениями. Они являются нативными по своей основной технологии, но основная пользовательская область такого приложения представляет из себя браузерный интерфейс. Это делает их похожими на родное приложение для операционной системы, предоставляя доступ к её функционалу, сохраняя гибкость пользовательского интерфейса веб-приложений. Однако это усложняет требования к интерфейсу, который должен быть достаточно универсальным, чтобы справиться со всеми поддерживаемыми системами. Преимущество гибридных приложений в том, что их можно распространять через магазины приложений.

В случае с веб-приложениями, разработка сводится к навыкам разработки интерактивных сайтов, интеграции в них расширенного функционала и адаптации сайтов к мобильным устройствам. Всё это хотя и касается мобильных устройств, все же представляет собой основы веб-разработки. Потому мы сосредоточимся на рассмотрении технологий программирования нативных кроссплатформенных мобильных приложений.

Поскольку под мобильными приложениями обычно подразумеваются именно приложения для Android и iOS, кроссплатформенные подходы, ориентированные на эти две системы, зачастую более специализированы, чем подходы, которые также позволяют создавать приложения для настольных систем. В первую очередь это связано с тем, что аппаратные требования мобильного устройства (сенсорный ввод, маленький экран и т.д.) полностью отличаются от требований настольного компьютера.

### **Xamarin**

Рассмотрение технологий мобильной кроссплатформенной разработки начнём с фреймворка Xamarin. Он создан компанией Microsoft и основан на кроссплатформенной среде .Net. Благодаря этому, любое устройство, на котором установлены пакеты .Net Core, способны запускать приложения, написанные с помощью Xamarin, а значит так можно создавать и мобильные приложения для Android и iOS.

Фреймворк поддерживает два основных способа разработки интерфейса приложений:

- Традиционный (собственный): пользовательский интерфейс создается с использованием технологий конкретной платформы, а внутренняя-логика программируется на базе общего кода на C#.
- Xamarin.Forms: пользовательский интерфейс можно настроить с помощью собственных библиотек Xamarin. Формы создаются декларативно на диалекте XAML. Xamarin переводит это общее описание пользовательского интерфейса в конкретный вид для конкретной платформы [2].

Кроме того, Xamarin предоставляет так называемые классы-оболочки, чтобы иметь возможность использовать и другие функции конкретных платформ. Таким образом можно обращаться к специальному аппаратному обеспечению систем. Например, вы можете получить доступ к функциям датчика отпечатков пальцев iPhone из C#. Компилятор Xamarin независимо создает собственные приложения из исходного кода C# и базы кода .NET Core для различных систем исключительно вовремя релиз-сборки проекта.

Как разработчику, вам будет важно, чтобы в фреймворке было как можно меньше ошибок взаимодействия платформ, потому что команда Xamarin необходимо своевременно обновлять функции платформ Android и iOS, чтобы разработчики всегда имели доступ к последним изменениям. Однако не стоит переживать, ведь благодаря поддержке большой и успешной родительской компании, на данный момент фреймворк работает очень хорошо, и обновления для библиотек Xamarin становятся доступны через относительно короткие промежутки времени после изменений целевых платформ.

### **RAD Studio с FireMonkey**

Приложения для Android и iOS можно создавать с помощью среды разработки RAD Studio [3]. В документации RAD Studio содержится информация о конкретных требованиях, которые необходимо выполнить для создания приложений для Android. Кроссплатформенная графическая среда FireMonkey делает упрощает данный процесс. Приложения для мобильных систем являются результатом так называемого

кроссплатформенного программирования. Помимо приложений для мобильных систем, приложения для Windows, macOS и Linux также могут быть созданы из одного исходного кода. Важным преимуществом создания приложений с помощью RAD Studio является то, что пользовательский интерфейс полностью создается в графическом дизайнере.

Постобработка в коде не нужна. Важные свойства элементов управления также настраиваются с помощью графической конфигурации. Еще во время разработки можно оценить, как будут выглядеть экраны с помощью функции предварительного просмотра.

Среда разработки также помогает удобно выполнять конфигурации для конкретных платформ через диалоговые окна. С помощью всего нескольких нажатий вы можете изменить целевую платформу и сгенерировать соответствующий пакет приложения. При кроссплатформенной разработке создается функционально идентичное приложение для разных систем.

Также стоит упомянуть так называемый LiveBinding. Это позволяет связать свойства одних визуальных компонентов со структурами данных или свойствами других компонентов. Таким образом, потоки данных к пользовательскому интерфейсу могут быть реализованы через конфигурацию, другими словами, для управления приложением не обязательно нужно писать программный код. С помощью Bindings Expert можно быстро выбрать источник и назначение такого визуального подключения к данным и настроить их соответствующим образом. Таким образом, объем исходного кода значительно сокращается. Стандартных компонентов, доступных в палитре, в большинстве случаев должно быть достаточно для реализации требований пользовательского интерфейса. Тем не менее, есть возможность установить и использовать дополнительные внешние компоненты.

Внешний вид приложения на основе FireMonkey можно быстро адаптировать к желаемому дизайну с помощью предустановленных стилей. RAD Studio сама по себе является приложением для Windows, но вы также можете запустить ее на виртуальной машине macOS, чтобы использовать только одну машину для разработки.

## **React Native**

Фреймворк React Native существует уже некоторое время и до сих пор очень популярен. Он был создан Facebook, поэтому очевидно, что гигант социальных сетей использует эту технологию. Правда, не во всей платформе, потому что React Native для него слабоват, но для Instagram и Ads Manager его точно хватит. Также стоит отметить, что React Native выбрали такие компании, как UberEats, Pinterest, Walmart, Bloomberg и другие известные и любимые бренды.

Благодаря React Native вы можете создавать качественные мобильные приложения. Он позволяет писать пользовательские интерфейсы приложений на языке JavaScript. Затем продукт может быть запущен на Android, iOS и даже на Windows. Довольно приятный вариант. Но значит ли это, что мы получаем 2 или даже 3 приложения по цене одного? Нет. Один базовый код — это только начало, а реализация приложения на конкретной платформе — совсем другое. React Native компилируется в нативные элементы, а это значит, что вам все равно придется реализовывать внутренний функционал приложений отдельно на каждой платформе.

Реальное преимущество кроссплатформенных фреймворков заключается в том, что вам нужна только одна команда для создания приложения для нескольких систем. Это обеспечивает экономию на управлении проектами, быстрое внедрение и гораздо более быстрый путь с нуля, например, до приложений MVP. Использование кроссплатформенных фреймворков снизит цену приложения примерно на 20-30% или позволит вам создать больше функций для большего количества пользователей в рамках вашего бюджета. Последнее может быть определяющим моментом при выборе технологий разработки. В конце концов, цель разработчиков — это предоставить пользователям наилучший продукт, который они могут загрузить из Google Play и App Store.

## **Flutter**

Еще один фреймворк кроссплатформенной разработки Flutter. И хотя в последние 3-4 года спрос на React Native был очень высоким, но Flutter какое-то время лидировал в создании кроссплатформенных приложений. Конечно, поскольку у нас есть много экспертов по фреймворку от Facebook, мы по-прежнему создаем продукты в React Native. Если клиент хочет использовать эту технологию и нет противопоказаний, например, необходимости создания расширенного функционала, React Native по-прежнему является отличным инструментом для создания кроссплатформенных решений. По мнению наших специалистов, React Native в ближайшие годы продолжит уступать место Flutter, хотя и останется значимым игроком в кроссплатформенной группе. Ведь второе место — это все-таки подиум.

Flutter создан компанией Google и предлагает новый подход к созданию нативных приложений для iOS и Android [4]. Flutter не использует компоненты пользовательского интерфейса целевых платформ, а работает с собственным механизмом рендеринга Skia для отрисовки компонентов. В нем были воссозданы элементы пользовательского интерфейса с нуля. Пользовательский интерфейс здесь создается непосредственно в коде, графического редактора нет. Flutter основан на языке программирования Dart от Google. Синтаксис Dart основан на синтаксисе C. Dart строго типизирован и объектно-ориентирован. Для разработки требуется Flutter SDK. Например, в качестве среды разработки можно использовать Android Studio или IntelliJ IDEA. Также можно

использовать Visual Studio Code. Производительность разрабатываемых приложений очень хорошая. Материальный дизайн пользовательского интерфейса соответствует Android и может быть легко настроен. Это упрощает создание приложений, которые вы хотите согласовать, например, со спецификациями существующего корпоративного дизайна.

### **Swift**

Если говорить о специализированных мобильных решениях для разработки, то в последние несколько лет в сфере производства iOS-приложений классический Objective-C был заменен языком Swift. Технологии, инструменты и решения Swift проложили путь к разработке современных мобильных приложений для самых требовательных пользователей. Нативные реализации iOS ограничены только возможностями мобильного устройства и вашим воображением.

Если вы хотите предлагать инновационные решения, сочетающие в себе красоту, качество и максимально возможную производительность, нативное приложение для iOS будет вашим лучшим решением. Если ваше приложение должно стать очередным хитом, самым скачиваемым приложением из магазина — вы не сможете равнодушно пройти мимо нативной iOS. Однако для такой разработки необходимо устройство под управлением операционных систем компании Apple.

### **Kotlin**

В случае же с Android, помимо инноваций, когда речь идет о качестве, скорости и производительности, настоящие приложения превосходят гибридные, кроссплатформенные приложения или приложения PWA. А в данной сфере на Google I/O 2019 было объявлено, что язык программирования Kotlin стал приоритетным в разработке под Android, с помощью которого вы можете создать приложение самого высокого уровня. Kotlin (Кóтлин) — статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine и разрабатываемый компанией JetBrains. Синтаксис языка преимущественно комбинирует наследство из двух языковых ветвей: Си/C++/Java и ML (Meta Language — по словам создателей, через Scala) [5]. Работая на всех типах оборудования, независимо от производителя, разрешения экрана, версии Android, мобильные приложения, написанные на Kotlin, обеспечивают лучший пользовательский опыт по сравнению с другими языками программирования.

### **Среда разработки и системы**

Ещё немного о технических особенностях реализации мобильных приложений. В отличие от разработки классических настольных приложений, разработка приложений для мобильных систем требует немного больше усилий с точки зрения системы и среды разработки. Android основан на ядре Linux и в основном является открытой системой, другими словами, разработка возможна на настольных операционных системах Windows, Linux и macOS. Эмуляторы также доступны для всех операционных систем. Иная ситуация с iOS. Для создания окончательных пакетов приложений для развертывания требуется Mac с Xcode и iOS. Симулятор iOS также можно запустить только на macOS. Чтобы протестировать приложения, вы можете использовать эмулятор/симулятор в качестве первого шага. Однако надо учитывать элементы, которые не заметны в эмуляторах:

- отображение расхода батареи;
- как работает камера мобильного устройства;
- прерывания от входящих звонков и смс;
- последствия интенсивного использования памяти.

Этот небольшой список уже показывает, что всестороннее тестирование на реальных устройствах необходимо на продвинутой стадии разработки. На практике, однако, у разработчика есть только ограниченный набор оборудования. В частности, для Android существует бесчисленное множество типов устройств, классов производительности и версий операционной системы. Однако неоднородность также значительно увеличилась в iOS. С помощью облачных решений вы можете протестировать свое приложение на самых разных устройствах. Примером может служить Центр приложений Visual Studio [6]. Это дает вам доступ к бесчисленным мобильным устройствам для Android и iOS. На продвинутых этапах проекта это хорошее дополнение для противодействия растущему разнообразию устройств. Дизайн системной среды также может варьироваться в зависимости от подхода.

### **Заключение. Возможности дальнейших исследований**

Приведенная информация показывает, что кроссплатформенные подходы обеспечивают хорошее соотношение затрат и выгод с технической и экономической точек зрения. Но какой подход правильный? Однозначно ответить на данный вопрос невозможно. Соответствующий подход, используемый язык программирования, способ построения пользовательского интерфейса и цикл разработки сильно различаются от

случая к случаю. Однако это типично для кроссплатформенного программирования в целом, потому что в этом случае мы хотим обслуживать и Android, и iOS из общего исходного кода. Из этого можно сделать вывод: если к архитектуре приложения или оборудованию предъявляются особые требования, то возможность их реализации должна быть предварительно оценена. В противном случае вы рискуете не заметить, пока проект не перейдет в продвинутую стадию разработки, что выбранный путь слишком тернист или вообще ведет в тупик, а весь созданный проект придется закрывать и начинать сначала. Поэтому важно заранее тщательно взвесить и принять решение о выборе технологий разработки в случаях, если необходима поддержка различных платформ, а специалистов работы с конкретным фреймворком в команде нет и будет необходимо изучение инструментов с нуля.

В докладе были рассмотрены различные варианты создания нативного приложения для мобильных систем. Хорошая новость для вступающих на путь мобильной разработки: спектр возможностей разработки широк. Это же является плохой новостью. Тем не менее, в начале данного пути следует принять решение, с какими технологиями вы готовы работать. Это возможно только в том случае, если вы сначала поэкспериментируете. Как далеко вы сможете продвинуться с одним подходом? Что получается хорошо, а что не работает? На данный момент имеются и новые, удобные, прогрессивные решения разработки кроссплатформенных приложений, и проверенные временем, поддерживаемые своим сообществом и разработчиками, требующие новых специалистов технологии для работы над уже существующими продуктами.

Постоянное же развитие возможностей для создания новых приложений требует и постоянного исследования появляющихся, развивающихся и исчезающих технологий для облегчения начала разработки новых специалистов и обеспечения полного понимания ситуации в технологической сфере для действующих исполнителей и заказчиков.

### **Источники и литература**

1. Википедия. Нативные приложения [Электронный ресурс] – 2022 – Режим доступа: [https://ru.wikipedia.org/wiki/Нативные\\_приложения](https://ru.wikipedia.org/wiki/Нативные_приложения)
2. Is Xamarin.Forms Making Traditional Xamarin Obsolete? [Электронный ресурс] / Adam – 2017 – Режим доступа: <https://xamarinhelp.com/xamarin-forms-making-traditional-xamarin-obsolete/>
3. RAD Studio 11.2 [Электронный ресурс] – Режим доступа: <https://www.embarcadero.com/products/rad-studio>
4. Flutter [Электронный ресурс] – Режим доступа: <https://flutter.dev>
5. Википедия. Kotlin [Электронный ресурс] – 2022 – Режим доступа: <https://ru.wikipedia.org/wiki/Kotlin>
6. Welcome to App Center [Электронный ресурс] – Режим доступа: <https://appcenter.ms/>

**Носаченко А.А., Щедрин С.В. Основные технологии программирования мобильных систем.** В докладе приведён обзор современных тенденций применения различных технологий программирования мобильных систем. Представлена краткая характеристика рассмотренных технологий. Текущие технологии создания программного обеспечения продолжают развиваться, разработка мобильных систем изменяется с большой скоростью, что требует регулярного исследования её технологий.

**Ключевые слова:** доклад, программирование, кроссплатформенная разработка, мобильная система, Android, iOS, нативное приложение.

**Nosachenko A.A., Schedrin S.V. Basic technologies for programming mobile systems.** The report provides an overview of current tendencies in the use of various technologies for programming mobile systems. The considered technologies are briefly described. Current software development technologies continue to evolve, mobile systems development is changing at a high rate, requiring regular research into its technologies.

**Keywords:** report, programming, cross-platform development, mobile system, Android, iOS, native application.

## ОПТИМИЗАЦИЯ И ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ИГРОВЫХ ДВИЖКОВ ДЛЯ ANDROID

Боровиков А.И.<sup>\*1</sup>, Зори С. А.<sup>\*2</sup>

<sup>\*1</sup> магистр кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,

aleksey.borovikov.00@mail.ru, SPIN-код: 2472-1006

<sup>\*2</sup> д.т.н., проф. кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,

ik.ivt.rec@mail.ru, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

**Зори С. А., Боровиков А. И. Оптимизация и повышение производительности игровых движков для android.** В статье представлен анализ влияния производительности игры на «комфорт» для пользователей, проведён анализ возможных проблем оптимизации движка, тонкостей языка программирования java, которые могут повлиять на производительность и особенности написания игр под мобильные приложения. В дальнейших исследованиях планируется проверить на практике приведённые методы и представить метрики в виде графиков и диаграмм с целью их усовершенствования.

**Ключевые слова:** Java, CPU, GPU, fps, время простоя процессора, циклы, потоки, память, рендер, Vulkan, тепловая нагрузка, энергопотребление, OpenGL, libGDX, JVM GC.

### Введение

Производительность является важным компонентом любой игры, от неё зависит популярность игры, комфорт пользователя и общее впечатление. Под производительностью обычно подразумевают средний уровень FPS в игре за определённый промежуток времени. FPS (Frame Per Second) - число кадров в секунду, кадры - это изображения, которые обрабатывает графический адаптер по команде центрального процессора. Чем больше кадров демонстрируется в секунду, тем плавнее и комфортнее игра [1].

Вопреки ожиданиям немногие игроки требуют высоких показателей fps, для большинства достаточно средних показателей в 30 единиц, а для некоторых даже меньше. Исследования, проделанные компанией Lesta Studio [2], показывают, что для всей совокупности пользователей связь между производительностью и комфортной игрой достаточно слабая (рис.1).

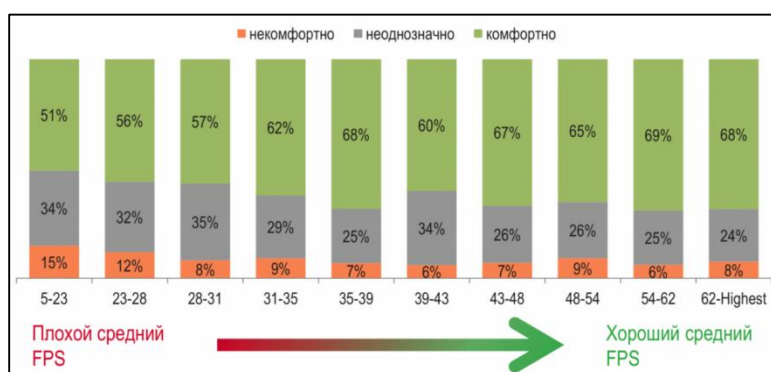


Рисунок 1 – Диаграмма зависимости между восприятием комфорта и средним FPS

Однако уровень комфорта зависит так же от жанра игры и от самого пользователя. Например, многие консольные игры работают на частоте 30fps, но с такими показателями картинка недостаточно плавная, а отзыв на манипуляции игрока запаздывает, что мешает полному погружению в игру. Для игроков в казуально-аркадные игры будет достаточно 40+fps, для шутеров от первого лица, стратегий, ритм-игр, МОБА понадобится около 60-100fps. Показатель Fps выше 100 потребуется игрокам соревновательных шутеров или киберспортсменов.

Таким образом, если не брать во внимание предпочтения в fps профессиональных игроков, то для большинства будет комфортным средний показатель в 60fps, этот показатель считается комфортным минимумом.

### **Влияние на производительность**

На производительность игры влияет следующее:

1. Клиентское ПО, включая ОС. Если система загружена фоновыми процессами или запущены программы, обращающиеся к диску во время игры, то ресурсов для самой игры может оказаться недостаточно.
2. Конфигурация ПК. От комплектующих компьютера напрямую зависит производительность.
3. Настройки графики. Графику настраивают пользователи обычно сами или используют пресеты, однако ни тот ни другой вариант не являются гарантией того, что настройки будут выставлены максимально оптимально, из-за чего пользователь теряет производительность.
4. Сетевые проблемы. Клиент-серверное взаимодействие не всегда устойчиво, нередко у пользователей возникают проблемы с сетью, что влияет на производительность сетевых игр. Причиной этому могут послужить проблемы на стороне провайдера, большое количество беспроводных клиентов неоптимальный сетевой маршрут доставки трафика, некачественное сетевое оборудование.
5. Модификации. Модификации оригинальной игры сторонними разработчиками отрицательно влияют на производительность. Они не только снижают средний fps, но и создают резкие провалы в тех случаях, где чистый клиент работает стабильно.
6. Скорость работы приложения. Существует огромное количество разных игровых движков со своими плюсами и минусами, все они потребляют разное количество ресурсов, в зависимости от качества архитектуры и поддерживаемых эффектов. Влияет так же и язык программирования, на котором написан движок, т.к. в разных языках ресурсы распределяются поразному. Об этом и пойдёт речь дальше.

### **Способы оптимизации кода**

Для того, чтоб движок выдавал максимальное количество fps необходимо не только уметь с ним эффективно работать, но и знать тонкости разработки на языке, на котором написан движок, и принципы работы движка для лучшего понимания процесса распределения ресурсов [3].

Существует много разных факторов, влияющих на скорость обработки кадров, рассмотрим общие методы оптимизации кода. Которые могут подойти практически в любой ситуации:

1. Минимизация влияния объектов за пределами экрана. Минимизация объёма вычислений при помощи оптимизации отрисовки объектов чрезвычайно важна. Зачастую этим занимаются движки, или же графический процессор. Для реализации необходимо разделить объект на 2 слоя - первый будет графическим представлением объекта, а второй - данными и функциями. Таким образом, если объект находится за пределами экрана - нам больше не нужно его отрисовывать.
2. Независимость от обновления кадров. Обычно в игровых движках большинство объектов обновляются в каждом кадре, это сильно нагружает процессор и понижает производительность. Необходимо, по возможности, избавиться от обновления в каждом кадре. Для этого необходимо отделить функцию рендеринга и вызывать её только при изменении состояния объекта.
3. Непосредственные вычисления и поиск значений. Неплохую прибавку к производительности может дать хеширование тригонометрических функций, т.к. эффективнее хранить большую таблицу и брать данные из неё, а не выполнять расчёты на лету.
4. Использование времени простоя процессора. Необходимо выделить функции, которые не зависят от времени, например, погодные условия и вычислять их, когда пользователь отошёл, занят чтением или другими не затратными ресурсами вещами. Время, которое освобождается, когда пользователь занят вещами, не нагружающими процессор, можно использовать для вычисления множества других событий.

### **Оптимизация движка на основе языка java**

Так же не лишним будет упомянуть оптимизацию кода, опираясь на особенности языка программирования java и особенности работы с движками на его основе [4];

1. Циклы. Если вам необходимо обойти большой массив данных, например, список прямоугольников (для рендеринга), врагов или любых других тяжелых объектов и вы собираетесь для этого использовать цикл for, то лучше пользоваться обратным for. Обратный for даёт большую эффективность за счёт того, что ему не нужно каждый раз проверять размер массива, а также это происходит, потому что сравнение с нулем является нормальным, в то время как сравнение с целым числом требует вычитания. Но если нет необходимости использовать цикл for, то более эффективным будет while.
2. Потoki. Необходимо разделять тяжелые действия на разные потоки и не запускать сложную логику в потоке отрисовки. При профилировании ЦП вы увидите поток с именем GLThread. Большинство движков используют GL 2.0/3.0 для рендеринга, и это поток, который содержит контекст GL. Это означает, что каждое

изменение пользовательского интерфейса должно производиться через этот поток, иначе случаются плохие вещи — текстуры не загружаются, элементы пользовательского интерфейса меняются одновременно из разных мест, это может быть беспорядок. Проблема в том, что использование этого потока оставляет пользователя с зависшим приложением.

3. Память. Не стоит забывать очищать компоненты, которые уже не нужны. Java имеет свой сборщик мусора, который чистит ненужные классы, но некоторые движки, например libGDX, этого не делают. Это связано с тем, что память OpenGL не управляется JVM GC, и, таким образом, если вы создаете свои собственные текстуры, вам придется удалять их вручную, иначе вы рискуете трудно отлаживаемой утечкой памяти.

4. Пакеты. Вызов начала и конца пакета спрайтов и ShapeRenderer занимает много памяти. Итак, старайтесь изо всех сил вызывать начало и конец только один раз за кадр. Попробуйте визуализировать все свои спрайты, а затем визуализировать фигуры. Вместо того, чтобы рендерить ваши спрайты, затем формы, а затем повторно открывая ваш spriteBatch.

5. Избегайте некоторых методов: Есть некоторые методы, которые требуют много ресурсов (или больше, чем другие), когда они используются, например, пересечения класса Rectangle. Например, лучшее, что вы можете сделать для просмотра массива прямоугольников, — это сначала проверить, близки ли оба прямоугольника, и если да, то вызвать для них метод Intersects. Вместо вызова Intersects со всеми прямоугольниками в массиве.

6. Визуализация. Метод рисования — это еще один элемент, который требует большого количества ресурсов, если вам нужно изменить состояние элемента, даже если он находится за пределами экрана, вы можете каждый раз пытаться обновлять его логику/переменные, но вызывать метод рисования только если он находится внутри координат экрана.

7. Переменные. объявление переменных или объектов в плохих местах может вызвать зависание, избегайте объявления переменных в цикле рендеринга и, в основном, не объявляйте их внутри чего-то вроде цикла for, вместо этого объявляйте их вне цикла и обновляйте его значения внутри. Хороший момент в этом, заключается в том, что это не имеет значения, когда мы говорим о примитивных типах данных, поэтому, если переменная имеет тип int, boolean или float, ее объявление внутри цикла не окажет большого влияния на производительность.

8. Паттерны. Паттерны значительно упрощают разработку и эффективность приложения, а также поддержку кода. Если вы хотите иметь один элемент, который может взаимодействовать и находится в любом классе вашей программы, что-то вездесущее, например, класс игрока, вы можете сделать его классом Singleton (это очень просто и даст вам более удобный код). Если вы не хотите, чтобы в вашей программе были бесполезные экземпляры, которые, возможно, вы никогда не будете использовать, вы можете использовать шаблон проектирования Factory.

### **Оптимизация игр для смартфонов [5]**

1. Скорость загрузки. Игроки хотят как можно быстрее погрузиться в действие вашей игры, поэтому важно максимально сократить время загрузки вашей игры. Следующие меры обычно помогают сократить время загрузки:

- Выполнить ленивую загрузку. Если вы используете одни и те же assets в последовательных сценах или уровнях игры, загружайте эти assets только один раз.

- Уменьшите размер своих assets. Таким образом, вы можете связать несжатые версии этих ресурсов с APK вашей игры.

- Используйте эффективный для диска метод сжатия. Примером такого метода является zlib.

2. Необходимо держать потоки с большим объемом памяти на одном процессоре. На многих мобильных устройствах кэши L1 находятся на определенных ЦП, а кэши L2 — на наборе ЦП, которые имеют общие часы. Чтобы максимизировать попадание в кэш L1, как правило, лучше всего, чтобы основной поток вашей игры, наряду с любыми другими потоками с большим объемом памяти, выполнялся на одном процессоре.

3. Необходимо откладывать кратковременную работу на процессоры с меньшей мощностью. Большинство игровых движков умеют откладывать операции рабочего потока на другой ЦП по сравнению с основным потоком вашей игры. Однако движок не знает о конкретной архитектуре устройства и не может предвидеть рабочую нагрузку вашей игры так же хорошо, как вы. Большинство устройств типа «система на кристалле» имеют как минимум 2 общих тактовых генератора. один для быстрых процессоров устройства и один для медленных процессоров устройства. Следствием этой архитектуры является то, что если одному быстрому ЦП необходимо работать на максимальной скорости, все остальные быстрые ЦП также работают на максимальной скорости. В примере отчета, показанном на рис. 2, показана игра, использующая преимущества быстрых процессоров [6]. Однако, этот высокий уровень активности быстро генерирует большое количество энергии и тепла.





## Выводы

Существует огромное количество игровых движков, которые предоставляют программисту много различных функций и позволяют создавать качественные игры, однако если она рассчитана и на пользователей со слабым устройством, то необходимо использовать знания языка программирования, архитектуры движка и операционной системы, чтоб добиться удовлетворительных показателей производительности программы. Для создания подобных проектов необходим большой опыт и немалое количество навыков.

В дальнейших исследованиях планируется проверить на практике приведённые выше методы и представить метрики в виде графиков и диаграмм с целью их усовершенствования.

## Литература

1. Каким должен быть комфортный FPS в играх [Электронный ресурс] – Режим доступа: <https://andpro.ru/blog/pc/kakim-dolzhen-byt-komfortnyy-fps-v-igrakh/> - Загл. с экрана.
2. Почему производительность в играх — это не просто средний FPS [Электронный ресурс] – Режим доступа: [https://habr.com/ru/company/lesta\\_studio/blog/265487/](https://habr.com/ru/company/lesta_studio/blog/265487/) - Загл. с экрана.
3. Основы оптимизации кода игр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/358176/> - Загл. с экрана.
4. Optimizing your Game [Электронный ресурс] – Режим доступа: [https://www.reddit.com/r/libgdx/comments/7cw4va/optimizing\\_your\\_game/](https://www.reddit.com/r/libgdx/comments/7cw4va/optimizing_your_game/) - Загл. с экрана.
5. The LibGDX performance guide [Электронный ресурс] – Режим доступа: <https://yairm210.medium.com/the-libgdx-performance-guide-1d068a84e181> - Загл. с экрана.
6. Best practices for Android game optimization [Электронный ресурс] – Режим доступа: <https://developer.android.com/games/optimize> - Загл. с экрана.

*Зори С. А., Боровиков А. И. Оптимизация и повышение производительности игровых движков для android. В статье представлен анализ влияния производительности игры на «комфорт» для пользователей, проведён анализ возможных проблем оптимизации движка, тонкостей языка программирования java, которые могут повлиять на производительность и особенности написания игр под мобильные приложения. В дальнейших исследованиях планируется проверить на практике приведённые методы и представить метрики в виде графиков и диаграмм с целью их усовершенствования.*

**Ключевые слова:** Java, CPU, GPU, fps, время простоя процессора, циклы, потоки, память, рендер, Vulkan, тепловая нагрузка, энергопотребление, OpenGL, libGDX, JVM GC.

### Annotation:

**Zori S. A., Borovikov A. I. Optimization and performance improvement of game engines for android.** The article presents an analysis of the impact of game performance on "comfort" for users, an analysis of possible engine optimization problems, the subtleties of the java programming language, which can affect the performance and features of writing games for mobile applications. In further research, it is planned to test the above methods in practice and present the metrics in the form of graphs and charts in order to improve them.

**Keywords:** Java, CPU, GPU, fps, cpu idle time, cycles, threads, memory, render, Vulkan, thermal load, power consumption, OpenGL, libGDX, JVM GC.

## **Анализ процесса педагогического дизайна при создании дистанционной системы обучения иностранному языку**

**Е.А. Закамаркина<sup>\*1</sup>, Н.Е. Губенко<sup>\*2</sup>**

<sup>\*1</sup> магистрант, Донецкий национальный технический университет,  
zakamarkina@mail.ru

<sup>\*2</sup> к.т.н., доцент, Донецкий национальный технический университет,  
negubenko@mail.ru

*Закамаркина Е.А., Губенко Н.Е. Анализ процесса педагогического дизайна при создании дистанционной системы обучения иностранному языку. Статья посвящена одной из актуальных тем модернизации образования. Рассматривает возможность реализации модели педагогического дизайна в условиях цифровизации обучения, формулирует ряд задач, решение которых является необходимым для практической реализации данной модели.*

**Ключевые слова:** образовательная среда, средства обучения, педагогический дизайн, цифровизация

### **Специфика дистанционного обучения иностранным языкам**

Образовательный процесс в настоящее время сложно представить без использования online обучения. Стремительный рост числа массовых онлайн курсов вызывает интерес к ним со стороны образовательных организаций. Современное поколение живет в виртуальном пространстве, поэтому онлайн обучение — это отличный способ выучить иностранный язык или улучшить свои знания.

Основными достоинствами онлайн-обучения стоит считать:

- потенциал к созданию в данной системе персонального (индивидуального) образовательного направления для каждого студента. Персональная программа обучения плодотворно влияет на создание у обучающихся интереса к познанию, мотивирует, формирует желание приобретать новые знания и навыки, реализовать профессиональное самоопределение, подготовить необходимую основу для будущего карьерного развития;

- создание условий для постоянного доступа к профессиональному образованию, вне зависимости от страны или региона проживания, для всех студентов.

Дистанционный формат обучения обладает широким спектром возможностей для применения. В виду того, что система организации обучения иностранным языкам связана с применением различных методик и практик преподавания, особое внимание уделяется интерактивности процесса и обратной связи.

Сегодня дистанционный формат занятий успешно применяется в педагогическом процессе при изучении иностранных языков в большинстве высших учебных. Внедрение дистанционного формата в образовательную систему дало возможность студентам приобретать знания и навыки без отрыва от основной деятельности, выбирать способы изучения материала в различных условиях, что существенно увеличило уровень знаний. Сейчас студентам особенно интересен тот преподаватель, который готов, знает и умеет применять в процессе обучения дистанционные технологии, создавать виртуальную образовательную среду для полноценного взаимодействия с обучающимися. Навык взаимодействия с информацией очень высоко ценится, в виду чего особо значима реализация педагогического потенциала и дистанционных методик относительно изучения иностранных языков в высших учебных заведениях.

Теоретический материал можно вполне доступно преподавать в онлайн-формате в виде презентаций с возможностью задавать вопросы, а также в формате файлов и коротких видео. Элементы индивидуальной работы, такие как письменные упражнения и обратная связь, проверка домашнего чтения также можно перенести в онлайн.

Социализация в процессе обучения является очень важным аспектом. Анализ полученного опыта даёт понять, что применяемые методики, не могут в полной мере заменить «прямое» аудиторное взаимодействие со студентами. При этом методики онлайн обучения являются достаточно эффективными при изучении отдельных аспектов языка, в частности, грамматики и перевода. [1]

Задачами ближайшего будущего можно бесспорно считать определение оптимального соотношения объемов рассматриваемых методик, выявление нюансов их использования при обучении различным языковым аспектам, акцентное использование тех или иных методик в группах различного уровня языковых компетенций,

разработка их учебно-методического обеспечения и создание новых «смешанных» учебников. В данный момент для изучения иностранных языков существует большое количество ресурсов, которые существенно увеличивают продуктивность усвоения материала. Они представляют собой порталы, платформы, веб-сайты и другие приложения для взаимодействия студентов и преподавателей, выполнения заданий и упражнений, участия в конференциях и вебинарах, распространения различного контента, и так далее. Более подробно можно рассмотреть востребованные ресурсы для изучения английского языка в таблице 1:

Таблица 1 - Ресурсы для изучения иностранных языков

Ресурс	Описание
Macmillan English Campus	Macmillan English Campus – платформа для изучения английского языка для подростков, молодых взрослых и профессионалов, предоставляющая студентам языковую практику по запросу, а преподавателям – гибкие материалы и курсы, доступные в любое время и в любом месте.
Vocabulary Exercises for Academic Word List	Этот веб-сайт содержит более 750 упражнений для заполнения пробелов, чтобы изучить и просмотреть более 2000 элементов общей лексики и академической лексики на английском языке. Данные упражнения – это отличный способ попрактиковать словарный запас в различных контекстах. Ресурс может быть использован для расширения учащимся словарного запаса.
UEfAP Academic Writing Files	Использование английского языка в академических целях: информация и рекомендации для студентов высших учебных заведений.
Ted talks	TED – некоммерческая организация, занимающаяся распространением идей, как правило, в форме коротких бесед. Ресурс стартовал в 1984 году как конференция, на которой сходились технологии, развлечения и дизайна, и сегодня охватывает почти все сферы – от науки до бизнеса и глобальных проблем на более чем 110 языках.
ESL Brains – Unique video-based teaching materials for adults	Программы для обучения взрослых, которые затрагивают различные аспекты современной жизни.
BBC news	Последние новости, спорт, телевидение, радио и многое другое. Ресурс информирует, обучает и развлекает независимо от вашего возраста
The Economist – World News, Politics, etc.	Международная еженедельная газета, печатаемая в журнальном формате и издаваемая в цифровом формате, которая фокусируется на текущих делах, международном бизнесе, политике и технологиях.
The New Yorker magazine	The New Yorker – американский еженедельный журнал.
Quizlet: Learning tools & flashcards	Обучающий ресурс, который позволяет изучать любое направление.

Онлайн-формат занятий стал практически повсеместно неотъемлемой частью курса обучения иностранным языкам, при этом он не заменяет полноценное очное взаимодействие преподавателя и студента.

Внедрение инновационного подхода к изучению иностранных языков с применением современных информационных технологий остаётся приоритетным направлением в сфере образования.

### **Организация дистанционного обучения иностранному языку на базе компьютерных телекоммуникаций**

Телекоммуникационная обучающая среда предполагает использование широкого спектра новых информационно-коммуникационных технологий, составляющую системы средств дистанционного обучения. Актуализируется разработка вариативных моделей педагогического дизайна телекоммуникационных сред обучения, основанных на методологии проектирования.

Педагогический дизайн — это область науки и творческой практико-ориентированной деятельности, опирающаяся на совокупность теорий познания (бихевиоризм, когнитивистика, коннективизм, позитивизм и конструктивизм), элементы инженерного и художественного конструирования и принципы наиболее успешных педагогических практик, занимающаяся вопросами рационализации процесса обучения, создания комфортной образовательной среды, стилистики, структуры и содержания современных учебных материалов. Исходя из общей установки, что средство — это «все то, что использует субъект в процессе движения к цели», под средствами педагогического дизайна понимается педагогически обоснованные, этически выверенные и эстетически оформленные педагогические средства, созданные с опорой на принципы педагогического дизайна.

На основе ряда научных работ были сформулированы следующие принципы педагогического дизайна:

– эффективное целеполагание: четкое представление о том, что обучающиеся должны усвоить в результате занятий и как можно измерить произошедшие изменения;

– индивидуализация и дифференциация обучения: учет психологических особенностей и потребностей обучающихся и соответствующая коррекция программы, деление обучающихся на группы по их уровням, индивидуальные задания, способствующие формированию и развитию личности;

– открытость и нелинейность процесса обучения, создание условий для конструирования собственного опыта: актуальность и сложность (многогранность) тем, а также их связь с окружающей действительностью, модульная организация программы обучения, атмосфера свободного и неформального диалога обучающегося и обучающего, возможность построения индивидуальной траектории и конструирования собственного опыта для каждого обучающегося без готовых знаний, моделей, алгоритмов и способов решения задач, условия для самовыражения, самообучения, самосовершенствования и самореализации обучающихся;

– комплексное воздействие на все репрезентативные системы: эффективность преподавания обеспечивается благодаря апелляции к различным стилям мышления, разному культурному наследию, а также воздействию на все репрезентативные системы или сенсорные модальности (зрительные, слуховые, вкусовые, тактильные и обонятельные ощущения), на основе которых формируется опыт;

– рефлексивность и постоянная обратная связь как неотъемлемый компонент обучения: оценка обучающимися приобретенных знаний, умений и навыков, а также изменений в духовном мире, обратная связь от обучающихся и других преподавателей для оценки эффективности проведенных занятий или программы;

– эстетизация образовательной среды средствами педагогического дизайна. Помимо опоры на комплекс перечисленных нами принципов, отличительной особенностью средств педагогического дизайна является их внешнее представление, то есть преобразование собственно педагогических средств, средствами дизайна с целью улучшения восприятия, запоминания и усвоения учебной информации. Условно такие «преобразованные» средства называют визуальными, аудиальными, аудиовизуальными и предметно-пространственными.

Полагается, что средства педагогического дизайна играют особую роль в создании такого пространства взаимодействия, которое оказывает влияние на ценностные ориентации, идеалы, установки, что обусловлено активным воздействием предметной среды на человеческое сознание. Имея представление о психологических особенностях восприятия, системе психических структур и духовных ценностей обучающихся, в процессе разработки средств педагогического дизайна можно запрограммировать их эмоционально-психическое и нравственное воздействие на обучающихся. [2]

Итак, в определении термина «педагогический дизайн» подчеркивается не только научная, но и творческая его составляющая. Знание технологий проектирования задает только формальную схему разработки электронных учебных материалов, а этого недостаточно для создания методически грамотного продукта. Требуется творческий подход в создании собственных цифровых образовательных ресурсов с помощью соответствующих технологий и инструментальных средств. Средства педагогического дизайна рассматриваются как симбиоз конвенциональных педагогических средств и средств дизайна с опорой на определенные принципы, результатом которого является повышение привлекательности и результативности учебных материалов. Перспективными дальнейшими направлениями работы считаются изучение влияния звукового сопровождения уроков, речи преподавателя, личности педагога и ресурсного обеспечения изучаемой дисциплины в режиме самостоятельной работы.

### ***Знание основ педагогического дизайна***

Из выше сказанного следует, что педагогический дизайн весьма актуален как для преподавателей, может рассматриваться как одна из важнейших составляющих подготовки и переобразования, так и для студентов, изучающих подготовки педагогических кадров.

Педагогический дизайн это – разработка и развитие социально-образовательной среды в направлении передачи все большего числа механических функций от учителя техническим средствам и программным агентам. Ученые рассматривают педагогический дизайн как науку и практику. Как наука педагогический дизайн изучает эффективность учебных материалов и средств, которые создают определенную образовательную среду. В этой связи важными категориями педагогического дизайна являются: образовательная среда, образовательный ресурс, образовательное средство, учебный материал (контент). Образовательная среда в этом случае рассматривается как система влияний и условий формирования личности по заданному образцу, а также создание возможностей для ее развития. То есть ученик, погруженный в образовательную среду постоянно занят работой с информацией: читает книгу, анализирует текст, решает задачи и тому подобное. Поэтому на первый план педагогического дизайна выходит создание учебного материала, который проведет обучаемого от незнания к знанию, от неумения к умениям и навыкам, формированию ключевых компетенций. Когда учебное средство создается как решение возникшего противоречия, способ решения определенной образовательной цели, а не как пересказ уже известных фактов, его можно назвать эффективным.

Знаниевая парадигма образования предполагала ретрансляцию готового знания, репродуктивное усвоение учебного материала, главенствующую роль предметного знания. В качестве результата

образовательного процесса в рамках данной парадигмы рассматривается степень загруженности обучаемого учебной информацией. [3]

Практико-ориентированная (компетентностная модель) образовательного процесса подразумевает усиление самостоятельной работы студентов, интерактивный характер обучения, индивидуализацию обучения. При этом предметное знание теряет свою главенствующую роль, степень загруженности информацией учащегося отходит на второй план. В качестве результата образовательного процесса рассматривается формирование у обучаемого системы определенных компетенций.

Основными этапами педагогического дизайна являются:

1. Анализ: определение цели учения, средств, условий будущей учебной работы.
2. Проектирование: подготовка планов, разработка прототипов, выбор основных решений, составление сценариев.
3. Разработка: превращение планов, сценариев, прототипов в набор учебных материалов.
4. Применение: учебные материалы используются в учебном процессе.
5. Оценка: результаты учебной работы оцениваются, данные оценки используются для корректировки учебных материалов.

При этом этапы педагогического дизайна представляют собой циклическую деятельность, в результате которой они повторяются для осуществления корректировки, совершенствования разрабатываемого учебного материала в течение практической апробации.

Прежде чем создавать проект сетевого урока, необходимо провести следующую подготовительную работу:

- определить цель создания, область применения, а также аудиторию, для которой он предназначен, так как все эти факторы в дальнейшем будут влиять на уровень изложения и представления материала;
- собрать и проанализировать информацию по разрабатываемой теме – она должна отражать действительность, быть актуальной, достаточной, не должна содержать неопределенностей, изложение должно соответствовать возрастной аудитории (лаконичное и грамотное);
- подобрать соответствующий иллюстративный материал – графики, чертежи, рисунки, видео-аудио ряды;
- продумать сценарий работы пользователя с проектом, исходя из этого выбрать его структуру и навигацию.

### ***Разработка технологической карты сетевого урока***

Таким образом, технологическая карта проектирует учебный процесс в зависимости от типа урока. В технологической карте определяют планируемый уровень результатов обучения: личностный, предметный и метапредметный. Задача состоит в том, чтобы на каждом этапе урока предоставить возможность ученику стать реальным субъектом деятельности, то есть сделать так, чтобы он не только выполнял определённые действия по намеченному плану, но и принимал участие в следующем:

- постановке цели и задач данного урока на основе границы собственного знания и незнания;
- планировании способов достижения намеченной цели;
- осуществлении контроля и оценки полученных результатов;
- осуществлении необходимой коррекции учебной деятельности на основе собственных затруднений;
- рефлексии деятельности как по итогам урока, так и на каждом структурном элементе урока;
- выборе задания из предложенных системой с учётом индивидуальных возможностей.

Алгоритм составления электронной технологической карты:

1. Выделить все виды работ для режима электронный учебник (преподаватель) плюс новый материал, показ приемов самостоятельной работы, показ приемов самоконтроля и взаимоконтроля.
2. Спланировать самостоятельную работу обучающихся, которая включает выполнение практических работ.
3. Сформировать блоки - задания по каждому виду самостоятельной работы.
4. Продумать вариант представления нового учебного материала.

При обучении в сетевой среде и на базе социальной сети, которой учеба рассматривается как формирование этой сети. Такой подход к обучению опирается, прежде всего, на акторно-сетевую теорию. Согласно этой теории, сеть выглядит как совокупность агентов (людей и вещей), которые взаимодействуют в едином пространстве и деятельность которых направлена на решение общей задачи.

Процесс обучения может быть представлен в форме раскручивающейся спирали: вообрази, создай, играй, поделись, обсуди и вновь вообрази. Процесс сетевого обучения предполагает, что участники совершают следующие действия:

1. Воображают и представляют, что именно они хотят сделать и получить в результате.
2. Создают проект, основанный на своих представлениях.
3. Играют с результатами своей деятельности.
4. Делятся результатами своей деятельности с другими людьми.

5. Обдумывают и обсуждают свои результаты.

6. Обсуждение и обдумывание приводит к новым представлениям и новым проектам.

Особое место в дистанционном обучении занимают элективные курсы, углубляющие, развивающие основные темы, изучаемые на занятиях. Их следует специально разрабатывать как автономные курсы, связанные, однако, по тематике с базовым материалом. [4]

Обратимся к сетевой модели, когда весь курс размещается в сетях. Это, собственно, дистанционное обучение для тех учащихся, которые не имеют возможности учиться очно. Это могут быть автономные курсы или информационно-образовательная среда (см.рис.1).



Рисунок 1 - Модель сетевого курса дистанционного обучения

При организации сетевого урока общая стратегия организации совместной деятельности состоит в том, что она поддерживает конструктивную деятельность, не навязывая участникам формы поведения. Это поддерживает у участников ощущение развивающегося сообщества, в котором постоянно происходит создание новых цифровых объектов, идей, проектов.

Педагогические возможности социальных сетей связаны с их использованием для сетевого обучения, целью которого является установление и укрепление связей между учителями, учениками, цифровыми объектами и орудиями деятельности. Педагогический дизайн сетевого урока поддерживает возникновение и укрепление связей и отношений в ходе совместной деятельности, результатом которой является коллективный проект.

Технологическая карта структурирована в логике системно-деятельностного подхода, то есть отражает деятельность педагога, деятельность обучающихся и планируемые результаты образовательного процесса. Сама структура карты нацеливает преподавателя на осознанный подход к планированию процесса формирования образовательных результатов.

### Список литературы

1. Роль и принципы дистанционного обучения иностранным языкам [Электронный ресурс]. Режим доступа: <https://moluch.ru/archive/88/17038/>
2. Организация дистанционного обучения [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/1727083/>
3. Ахаян А.А., Лаптев В.В. Перспективные и текущие задачи дистанционного сопровождения общего и профессионального педагогического образования // в сб. «Актуальные проблемы дистанционного педагогического образования». - СПб.: РГПУ, 1998. - С. 3-5.
4. Универсальная технологическая карта урока [Электронный ресурс]. Режим доступа: <https://multiurok.ru/files/tiekhnologhichieskaia-karta-kak-sovriemiennaia-for.html>

**Закамаркина Е.А., Губенко Н.Е. Анализ процесса педагогического дизайна при создании дистанционной системы обучения иностранному языку.** Статья посвящена одной из актуальных тем модернизации образования. Рассматривает возможность реализации модели педагогического дизайна в условиях цифровизации обучения, формулирует ряд задач, решение которых является необходимым для практической реализации данной модели.

**Ключевые слова:** образовательная среда, средства обучения, педагогический дизайн, цифровизация

***Zakamarkina E.A., Gubenko N.E. Analysis of the pedagogical design process when creating a distance learning system for a foreign language. The article is devoted to one of the topical issues of modernization of education. Considers the possibility of implementing a model of pedagogical design in the context of digitalization of learning, formulates a number of tasks, the solution of which is necessary for the practical implementation of this model.***

***Keywords:*** educational environment, learning tools, pedagogical design, digitalization.



## Анализ методик продвижения сайтов

В.И. Зензеров<sup>\*1</sup>, Д.А. Куликов<sup>\*2</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
mpvik56@mail.ru

<sup>\*2</sup> студент, Донецкий национальный технический университет,  
danilbastard4@gmail.com

*Зензеров В.И., Куликов Д.А. Анализ методик продвижения сайтов. Внедрение поисковой оптимизации (SEO) является важной задачей. Необходимо оптимизировать веб-сайт, чтобы успешно конкурировать с другими подобными веб-сайтами. SERP и SEO являются правильным выбором, используя методы SEO на странице для оптимизации сайта изнутри и оптимизации выдачи веб-сайта, может помочь конкурировать с другими аналогичными веб-сайтами в Интернете и улучшить результаты (SERP), чтобы заранее определить целевую аудиторию веб-сайта, его отображение на верхней странице поисковой выдачи нужно улучшить рейтинг веб-сайта и количество посетителей и выйти на более широкий рынок.*

**Ключевые слова:** поисковая оптимизация (SEO), Search Engine Results Page (SERP), электронная коммерция.

### Введение

IT-направления регулярно требуют улучшений, внесения изменений в главные составляющие области, включая SEO. Технологии развиваются стремительными темпами, что влечет за собой постоянную актуализацию, запуск инновационных технологий. Сфера поисковой оптимизации не исключение. Поэтому, стоит постоянно мониторить тренды в области SEO, чтобы разработать более эффективную и “работающую” стратегию продвижения.

Search Engine Optimization - это комплекс действий, направленных на улучшение сайта и продвижение его позиции в поисковой выдаче. Основная задача - привлечь клиентов в маркетинговую воронку. Благодаря качественной стратегии SEO -продвижения, на сайте увеличивается органический трафик.

В 2022 году ожидается несколько обновлений, касающиеся алгоритмов Google, улучшений пользовательского опыта. Как результат, большая часть изменений заденет и SEO область. Важно ставить поисковую оптимизацию в приоритет, так как инвестиции в данную область гарантируют хорошую производительность и общий успех рекламной кампании. Чтобы оставаться в ТОП по популярности запросов, необходимо следовать тенденциями в области SEO, которые меняются из года в год, факторам, необходимым для того, чтобы предлагать пользователям сети богатый и качественный контент. Если следовать новым трендам SEO на 2022 год, есть возможность идти в ногу с активным развитием технологий и получать высокие позиции в SEO-оптимизации, улучшая при этом пользовательский опыт [1].

Сегодня наряду с продолжающимся развитием интернет-технологий, так же развивается простота доступа к информации для более широкого пласта населения, это так же влияет на количество самих интернет-пользователей.

Согласно исследованию проекта WEB-Index, в феврале-ноябре 2020 года интернетом в России хотя бы раз в месяц пользовались в среднем 95,6 млн человек или 78,1% населения всей страны старше 12 лет. В среднем за день в интернет выходили 87,1 млн человек или 71,1% населения России [2].

Наблюдая за профессиональным ростом пользователей, не исключено, что предприниматель начнет искать способы, чтобы продукты или услуги, которые они предлагают, могли быть замечены и доступны интернет-пользователям. Но есть препятствия в достижении этих целей, потому что созданный веб-сайт является новым, позиция или рейтинг веб-сайта находятся в невыгодном положении. Достижение высокого рейтинга в результатах поиска очень важно для привлечения трафика на веб-сайт и является основной движущей силой маркетинговых усилий в поисковых системах. Однако достичь высокого рейтинга очень сложно, потому что поисковые системы не публикуют свои алгоритмы ранжирования публично и не раскрывают информацию о факторах, используемых при ранжировании.

62% интернет-пользователей больше заинтересованы в выборе 10 или 20 вебсайтов в результатах SERP, что означает, что они находятся на 1 и 2 страницах поиска, в то время как остальные 38% увидят веб-сайт, который находится на следующих страницах, поэтому необходимо принять меры по оптимизации, чтобы вебсайт мог

конкурировать с аналогичными вебсайтами. Выполняя SEO-оптимизацию, веб-сайт получит высокую ранжированность в результатах поиска, а поисковые системы также сделают контент веб-сайта более доступным. Согласно методу, использование соответствующих ключевых слов на веб-страницах очень важно для индексации поиска и ранжирования или релевантности, которые приписываются веб-страницам поисковыми системами. Страницы веб-сайтов с ключевыми словами, содержащимися в трех полях метатэгов (поля заголовка, поля темы и поля описания), будут иметь лучшую позицию для повышения рейтинга веб-сайта [3].

Целью данного исследования является повышение эффективности поисковой системы (SERP) на веб-сайте [sotdill 111 .wixsite.com/website-3](http://sotdill111.wixsite.com/website-3), чтобы повысить рейтинг веб-сайта в результатах поиска поисковых систем. Задачей работы – анализ методик продвижения сайтов.

### **Главные тренды SEO**

Можно выделить главные тренды SEO на 2022 год [4]:

1. Искусственный интеллект. Это новый способ взаимодействия с аудиторией с помощью онлайн-контента, который в корне меняет устаревшую модель. Интеллектуальный алгоритм показывает пользователю ту информацию, которая интересует его больше всего, облегчая поиск в сети. На данный момент, поисковые системы не распространяют свои технологии и достижения. Однако, считается, что это улучшит пользовательский опыт во время поиска. Могут быть включены факторы, в зависимости от рейтинга кликов, а также до времени, проведенного на странице. Для этого нужно увлечь читателей полезным и хорошо структурированным контентом, что позволит пользователям легче находить необходимую информацию или конкретный бренд на любой платформе.

2. Голосовой поиск. Голосовые помощники Google Assistant, Siri, Alexa уже используются в мобильных гаджетах, но в 2022 году процент пользователей увеличится. Emarketer крупная компания, занимающаяся исследованиями американского рынка, проводили анализ, в результате чего в 2019 году 33% населения использовали в повседневной жизни голосового помощника. Данный инструмент обеспечивает легкость и комфорт в сравнении с набором текста вручную. Согласитесь, проще задать запрос в поисковом поле, просто проговорив ключевые слова. Данная технология будет только прогрессировать и станет трендом SEO в 2022 году.

Чтобы улучшить SEO-оптимизацию для собственного бренда в этой конкретной области поиска, следует сосредоточиться на ключевых словах и фразах с дополнительной формулировкой. Голосовой помощник лучше работает с более детальными запросами. При наборе текста вручную пользователи обычно сокращают слова. Готовясь к SEO 2022 года, важно правильно оптимизировать контент, чтобы он отвечал на голосовой запрос, и соответствующим образом подбирать ключевые слова.

3. SEO с видео. Инновационный тренд SEO 2022, при этом активно развивающийся последние годы. На платформе Youtube уже более миллиарда пользователей. Instagram и Tiktok также насчитывают многомиллионные аудитории. По результатам опроса, опубликованного Statista, 27% людей тратят на видео не менее 10 часов в неделю, в то время как 15,4% тратят 7-10 часов, а 18,3% — 4-7 часов. Можно сделать вывод, что пользователи любят смотреть видео. По этой причине более эффективное SEO 2022 с помощью таких платформ, как Youtube, Instagram, TikTok, стало больше необходимостью, а не приоритетом. Внедрение видеоконтента в поисковую оптимизацию требует внимания не только основным ключам, но и второстепенным. Google больше не просто смотрит на строки слов. Теперь это анализ контекста по запросу и попытка понять основную цель поиска пользователя. Это означает, что важно предоставлять больше релевантной информации с помощью основных и дополнительных ключевых слов.

4. Адаптация контента под мобильные устройства. Сегодня большинство пользователей предпочитают серфить в интернете с помощью мобильного гаджета или планшета. В 2022 году важно иметь сайт, адаптированный под мобильные устройства для того, чтобы занимать более высокие позиции в поисковых системах. В 2019 году компания Google объявила, что любой веб-сайт без такой поддержки не представляют большой ценности для системы, учитывая индексацию мобильных устройств.

Внедрение тренда SEO 2022 включает несколько главных аспектов.

- нужно оптимизировать дизайн веб-сайта для автоматической адаптации под размер любого экрана;
- должен быть подходящий дисплей, автоматически интегрирующийся с любым разрешением, независимо от планшета, модели телефона;
- предоставить приложение для загрузки и установки на систему iOS и Android;
- выбрать дизайн для сайта или приложения, который упростит мобильный доступ и навигацию;
- предварительно протестировать для мобильных устройств через поиск Google. Таким образом, вы можете получить доступ к отчету об удобстве использования мобильных устройств в Google Search Console.

5. Google EAT. Сервис Google всегда указывает, что при поисковом запросе большое значение уделяется качеству контента. Это напрямую будет влиять на позицию в выдаче. К основным факторам, определяющим уровень качества контента, относятся экспертность, авторитетность и надежность (с англ. - Expertise, Authoritativeness, and Trustworthiness, сокращенно EAT). Подобный принцип имеет решающее значение, влияет на успех SEO и ранжирования. С помощью EAT Google может отличить сайт предлагающий некачественную или

неподходящую информацию, от веб-страницы, предоставляющей богатое и надежное содержание. Поэтому, тренд SEO 2022 - это создание контента в соответствии с требованиями Google.

Чтобы написать качественно, необходимо охватить свою целевую аудиторию, определить и понять, что они хотят. Полноценный собранный образ позволит изучить интересы, впоследствии представить соответствующий контент. Обязательно следует использовать ссылки на официальные источники, что значительно повысит уровень доверия. Быть в поисковом списке первым - значит сконцентрироваться на принципе EAT Google, соблюдая указанные факторы.

6. Избранные фрагменты (Featured Snippets). Google разработал специальные блоки с важной информацией. Это функция предоставляет пользователям самую быструю и актуальную информацию простым и доступным способом. Основная цель - как можно быстрее отвечать на вопросы и повышать удобство работы пользователей.

Таким образом, за счет сокращения затрачиваемого времени, улучшается взаимодействие. С точки зрения SEO-оптимизации, в первую очередь, данная опция считается «полезной для пользователя». Также, это увеличит процент органических показов, можно войти в список авторитетных источников в 2022 году.

7. Длинный и качественный контент. Еще один тренд SEO 2022, к которому пришли только со временем. Контенту уделяется больше внимания, выдвигаются особые требования. Длинный, но вовлекающий и интересный текст генерирует больше трафика. Важно сосредоточиться на подготовке актуальной и качественной информации, но максимально увлекательной. Обязательно следует использовать заголовки, разделять на подзаголовки, что упрощает ботам Google индексацию и сканирование статьи.

8. Доступность. Число людей в сети интернет с каждым днем увеличивается, включая людей с нарушениями зрения или другими ограниченными возможностями. Поэтому, ключевым фактором будет применение инструментов для того, чтобы быть увиденным и найденным каждым пользователем. Обеспечение всеобщей доступности контента, услуг, продуктов и прочего не только поможет продвигаться в рейтинге, но и улучшит сервис обслуживания клиентов.

9. Пользовательский опыт. Качество обслуживания — тренд SEO 2022, идущий параллельно с требованиями к удобству взаимодействия со страницей, которые Google упомянул в последнем обновлении. Google будет принимать во внимание пользовательский опыт сайта, чтобы должным образом ранжировать его. К ним относятся факторы, влияющие на время загрузки страницы, количество перенаправлений, изображений и длина страниц. Google будет проверять основные параметры страницы, а также удобство использования и навигацию сайта как на компьютере, так и на мобильных устройствах, размещение кнопок, отображение динамических изображений (всплывающих окна). Новое обновление также означает, что использование формата AMP больше не является обязательным условием.

10. Оптимизация и скорость загрузки. Данный тренд SEO 2022 стоит вынести в отдельный пункт. Почти все составляющие качественного продвижения в будущем году направлены на оптимизацию страницы, сокращение загрузки информации, изображений до минимума. Сайт должен загружаться быстро, не теряя при этом качество содержимого контента. Пользователи становятся все более избирательными и нетерпеливыми, поэтому важно адаптироваться под изменения в поведении человека.

### **Исследование эффективности методов поиска**

В этом исследовании используется метод внутреннего SEO, чтобы рейтинг [sotdill 111 .wixsite.com/website-3](https://sotdill111.wixsite.com/website-3) мог подняться, для получения преимущества в виде увеличения трафика и возможности конкурировать с другими аналогичными сайтами, которые уже существуют в Интернете. Это исследование проведено в несколько этапов, для поддержки реализации SEO включающее в себя выявление проблем. На этом этапе объектом является веб-сайт [sotdill 111.wixsite.com/website-3](https://sotdill111.wixsite.com/website-3), целью этого этапа является оптимизация сайта для того, чтобы он мог быть проиндексирован поисковыми системами. Далее был проведен сбор данных, при этом было использовано несколько способов, такие как изучения литературы, интервью и документации, чтобы получить максимальный объем данных [4, 5].

Следующий этап – разработка сайта, на этом этапе разрабатывается дизайн сайта с использованием конструктора сайтов Wix. В процессе разработки также уделяют внимание оптимизации для мобильных устройств. После создания сайта следующим шагом к внедрению SEO является исследование ключевых слов, на котором ведется подбор правильных ключевых слов. Поскольку использование ключевых слов является главной и основной частью внедрения SEO, определение правильных ключевых слов играет очень важную роль в определении успеха внедрения SEO.

Следующим шагом после исследования ключевых слов является предварительное тестирование страницы результатов поисковой системы (SERP). Этот этап выполняется в качестве основного эталона для сравнения результатов до и после оптимизации сайта. После того, как результаты предварительного тестирования получены, следующим этапом является внедрение SEO на вебсайт, в этом внедрении используются методы SEO на странице. Последним этапом этого исследования является повторное тестирование или посттестирование результатов поисковой после этого этапа тестирование проводится снова чтобы определить результаты после оптимизации и сравнить их с результатами до оптимизации [6].

### **Исследование ключевых слов**

На этапе исследования ключевых слов используется сайт Яндекс wordstat. При проведении этого исследования ключевых слов предлагаемый веб-сайт используется как инструмент для поиска целевых ключевых слов, как показано на рис. 1.

игровые клавиатуры

По словам

По регионам

История запросов

Все регионы

Подобрать

Все

Десктопы

Мобильные

Только телефоны

Только планшеты

Последнее обновление: 11.11.2022

Что искали со словом «игровые клавиатуры» — 64 132 показа в месяц

Статистика по словам

Показов в месяц

<a href="#">игровая клавиатура</a>	64 132
<a href="#">купить игровую клавиатуру</a>	7 371
<a href="#">клавиатура игровая механическая</a>	7 112
<a href="#">игровая клавиатура +с подсветкой</a>	5 486
<a href="#">лучшие игровые клавиатуры</a>	3 107
<a href="#">клавиатура компьютера игровая</a>	2 966
<a href="#">беспроводная игровая клавиатура</a>	2 626
<a href="#">проводная игровая клавиатура</a>	2 559
<a href="#">игровая клавиатура red</a>	2 463
<a href="#">игровая клавиатура +и мышь</a>	2 109
<a href="#">игровая клавиатура logitech</a>	1 982
<a href="#">игровая клавиатура rgb</a>	1 926
<a href="#">игровая клавиатура usb</a>	1 888
<a href="#">топ игровых клавиатур</a>	1 739
<a href="#">игровая клавиатура tkl</a>	1 726
<a href="#">игровая клавиатура red square</a>	1 674

Запросы, похожие на «игровые клавиатуры»

Статистика по словам

Показов в месяц

<a href="#">blood компьютерная игра</a>	1 039
<a href="#">game keyboard</a>	10 173
<a href="#">компьютерные игры для девочек</a>	2 143
<a href="#">gamekeyboard</a>	1 307
<a href="#">игры с геймпадом</a>	19 299
<a href="#">лучшие компьютерные игры для девочек</a>	390
<a href="#">игры только мышкой</a>	2 083
<a href="#">gaming mouse</a>	13 339
<a href="#">компьютерные игры для мальчиков</a>	1 240
<a href="#">game keyboard скачать</a>	801
<a href="#">aaa компьютерные игры</a>	129
<a href="#">геймерский браузер</a>	327
<a href="#">геймерский ноутбук</a>	866
<a href="#">gaming keyboard</a>	10 173
<a href="#">геймерский компьютер</a>	2 013
<a href="#">геймерская мышка</a>	990

Рисунок 1 – Подбор ключевых слов

После того, как подобраны необходимые ключевые слова и фразы, составим семантическое ядро, схема которого представлена на рис. 2.

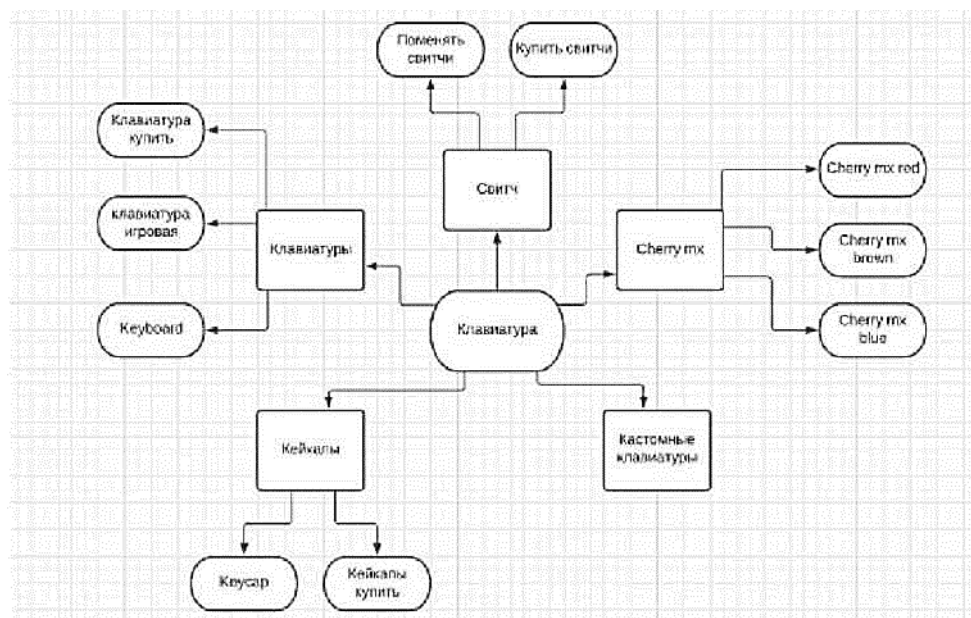


Рисунок 2 – Семантическое ядро

На основании результатов исследования по внедрению поисковой оптимизации (SEO) с использованием методов on page SEO, можно сделать вывод, что она может быть использована для улучшения рейтинга страниц результатов поисковых систем (SERP) на сайте [sotdill 111 .wixsite.com/website-3](http://sotdill111.wixsite.com/website-3) с заранее определенными целевыми ключевыми словами. Благодаря эффекту улучшения ранжирования сайт, который изначально занимал 40-е место, может постепенно подняться до 6-го места. SERP в течение примерно двух месяцев. Исходя из проведенной оценки, есть некоторые предложения, которые необходимо рассмотреть, а именно: максимизация методов вне страничного SEO путем увеличения количества качественных обратных ссылок и частого

социального обмена, чтобы рейтинг сайта стал более надежным и стабильным. Кроме того, вновь созданный веб-сайт также нуждается в дальнейшем исследовании осведомленности о бренде, чтобы пользователи Интернета могли более внимательно относиться к создаваемому веб-сайту.

## **Выводы**

Актуальные тренды SEO 2022 позволят принять и соответствовать новым тенденциям. Обновления помогут оставаться на вершине, попадать в ТОП лидеров при поисковом запросе, увеличить рост и развитие собственного бизнеса. Знание трендов SEO в 2022 году может принести множество преимуществ сайту и бренду, не считая достижения более высоких рейтингов. Это возможность заранее подготовиться и успешно реализовать разработанные принципы в маркетинговой стратегии на следующий год. В работе выполнен анализ методик продвижения сайтов, результаты которого позволяют повысить эффективность поисковой системы (SERP) на веб-сайте [sotdill111.wixsite.com/website-3](http://sotdill111.wixsite.com/website-3), чтобы повысить рейтинг веб-сайта. Выполнены анализ главных трендов SEO на 2022 год и исследования эффективности выбора методов поиска и ключевых слов.

## **Литература**

1. Петракова, Н. В. Информационные технологии дистанционного обучения / Н. В. Петракова // Сборник научных трудов института энергетики и природопользования. – Брянск: БГАУ, 2017. – С. 171-174.
2. Петракова, Н. В. Актуальность использования информационных технологий в системе профессионального образования / Н. В. Петракова // Информационные технологии в образовании и аграрном производстве: Сборник материалов III Международной научно-практической конференции. – Брянск: БГАУ, 2020. – С. 637-642.
3. Лысенкова, С. Н. Обзор методик продвижения услуг через интернет / С. Н. Лысенкова, А. И. Иноземцева // Вестник образовательного консорциума Среднерусский университет. Серия: Информационные технологии. – Брянск: БГАУ, 2019. – № 2 (14). – С. 14-17.
4. Милютин, Е. М. SEO оптимизация – основа продвижения сайта / Е. М. Милютин, Л. И. Бишутна, К. В. Исаев // Вестник образовательного консорциума Среднерусский университет. Серия: Информационные технологии. – Брянск: БГАУ, 2020. – № 2 (16). – С. 7-10.
5. Сержанова, И. В. Основы разработки сайта / И. В. Сержанова, Л. И. Бишутна // Информационные технологии в образовании и аграрном производстве: Сборник материалов III Международной научно-практической конференции. – Брянск: БГАУ, 2020. – С. 372-378.
6. Мишин, И. Н. Создание и использование информационной системы сельскохозяйственного предприятия на основе веб-сайта с интегрированной базой данных / И. Н. Мишин // Цифровые технологии – основа современного развития АПК: Сборник материалов Международной научной конференции. – Смоленск: Смоленская государственная сельскохозяйственная академия, 2020. – С. 421-426.

***Зензоров В.И., Куликов Д.А. Анализ методик продвижения сайтов. Внедрение поисковой оптимизации (SEO) является важной задачей. Необходимо оптимизировать веб-сайт, чтобы успешно конкурировать с другими подобными веб-сайтами. Используя методы SEO на странице для оптимизации сайта изнутри и оптимизации выдачи веб-сайта, может помочь конкурировать с другими аналогичными веб-сайтами в Интернете и улучшить результаты (SERP), чтобы заранее определить целевую аудиторию веб-сайта, его отображение на верхней странице поисковой выдачи нужно улучшить рейтинг веб-сайта и количество посетителей и выйти на более широкий рынок.***

**Ключевые слова:** поисковая оптимизация (SEO), Search Engine Results Page (SERP), электронная коммерция.

***Zenzerov Vladimir, Kulikov Daniel Analysis of methods of website promotion. The implementation of search engine optimization (SEO) is an important task. It is necessary to optimize the website in order to compete successfully with other similar websites. Using on-page SEO techniques to optimize the site from the inside and optimize the output of the website, can help compete with other similar websites on the Internet and improve the results (SERP), in order to determine the target audience of the website in advance, its display on the top page of the search results need to improve the ranking the website and the number of visitors and reach a wider market.***

**Key words:** Search Engine Optimization (SEO), Search Engine Results Page (SERP), e-commerce.

## **СЕКЦИЯ 2. «ИНФОРМАЦИОННЫЕ СИСТЕМЫ, БАЗЫ ДАННЫХ, БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»**

УДК 004.65:005

### **Анализ подходов к проектированию баз данных**

**В.В. Якубенко<sup>\*1</sup>, Т.Г. Дмитриук<sup>\*2</sup>**

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
vasya.ivanov2004.ivanov@mail.ru

<sup>\*2</sup> ассистент, Донецкий национальный технический университет,  
tnauka@lenta.ru, OrcID: 0000-0003-4803-5555, SPIN-код: 9249-2928

*Якубенко В.В., Дмитриук Т.Г. Анализ подходов к проектированию баз данных. Целью работы является обзор методов и средств проектирования баз данных, выделение преимуществ и недостатков их проектирования и использования, а также проведение сравнительного анализа существующих систем управления базами данных.*

*Ключевые слова:* база данных, методы, средства, СУБД, проектирование.

#### **Введение**

База данных, как и многие другие программные продукты, имеют жизненный цикл. В концепцию их функционирования, согласно системным и пользовательским требованиям, включено проектирование БД. Основное внимание пользователей уделяется их логическому проектированию, которое выражается в формулировке структуры представления данных и их связей, создании запросов и отчётов БД, а также разработке алгоритмов работы с данными. Выполнение задач проектирования базы данных на логическом уровне подчиняется особенностям заданной предметной области, для которой БД предназначена [1].

Цель настоящей работы – рассмотреть и проанализировать существующие методы и инструменты проектирования баз данных, согласно предложенным критериям выделить сильные и слабые стороны существующих наиболее популярных СУБД.

#### **Концепция проектирования БД**

При проектировании базы данных принято придерживаться принципов трёхуровневой системы организации базы данных, представленной на рисунке 1 [2].



Рисунок 1 — Трёхуровневая система организации БД

В модели, представленной на рисунке, описан алгоритм процесса проектирования БД, включающий три уровня: физический, концептуальный и логический.

На внешнем уровне разработчиками определяется, каким образом информация будет представлена пользователям. Данные в наиболее распространенных типах современных баз данных хранятся в виде строк и столбцов матрицы данных. Современные информационные технологии позволяют легко управлять данными, изменять, обновлять, контролировать и упорядочивать их [3]. В настоящее время известно много систем, выполняющих функции управления базами данных, наиболее востребованы пользователями — Oracle, MySQL, продукт Microsoft SQL Server, MongoDB и PostgreSQL (СУБД с открытым кодом и свободным распространением).

На концептуальном уровне осуществляется переход между внешним и внутренним уровнями системы организации БД. Концептуальная модель является результатом системного анализа предметной области и описывает механизм представления информации для пользователей. Пользователей БД, в свою очередь, условно делят на следующие категории: администраторы, обслуживающий персонал (разработчики, аналитики, программисты) и непосредственные пользователи БД. Обработка информации в БД производится при помощи ввода данных, указания взаимосвязей между ними, их сортировки и фильтрации; взаимодействие БД с пользователем осуществляется посредством запросов данных, для чего используется язык структурированных запросов (SQL). Современные СУБД в результате эволюции превратились в интегрированные системы, предназначенные не только для манипулирования данными, но и для разработки ориентированных на работу с базами данных приложений (прежде всего обеспечивающих дружественный интерфейс с пользователем).

После построения концептуальной модели происходит представление данных в виде структуры данных и осуществляется переход на физический уровень. Для представления данных на внутреннем уровне учитывается способ организации и представления данных в компьютерных системах и описания во внутренних схемах.

### **Методы проектирования баз данных**

Главной трудностью при проектировании баз данных считается принятие решений разработчиком о том, из каких отношений (связей) будет состоять разрабатываемая БД и какие атрибуты ей присущи. Метод проектирования БД является совокупностью механизмов представления моделей, в которых отражены особенности отношений разрабатываемой системы с применением определенных, чётко сформулированных нотаций. Ниже представлены самые известные модели и методы проектирования БД [1, 4].

**1. Метод пошаговой декомпозиции** состоит в итерационной последовательной декомпозиции (разложении на элементы) исходной и промежуточных схем отношений до того шага, на котором результаты станут соответствовать предъявляемым свойствам или требованиям заказчика.

**2. Метод синтеза** состоит в синтезе декомпозиционных элементов схем из отобранного набора атрибутов конкретной предметной области через множество функциональных зависимостей, заданное для связи атрибутов.

**3. Модель «сущность-связь»** рекомендуется к использованию для концептуального моделирования предметной области. Подход анализа предметной области, который продуман в модели «сущность-связь», стал одним из источников для методов объектно-ориентированного анализа.

**4. Метод IDEF1x (метод ER-диаграмм)** позволяет наглядно представить статический «снимок» сущностей и связей между ними в заданной предметной области. Сущность идентифицируется как действительный либо абстрактный объект исследования конкретной предметной области, обладающий свойствами-атрибутами. Как правило, экземпляр одной сущности, именуемый родительской сущностью, связан с произвольным (и нулевым в том числе) количеством экземпляров второй сущности, которая именуется сущностью-потомком, а любые экземпляры сущности-потомка ассоциированы исключительно с одним экземпляром сущности-родителя.

Метод IDEF1x является результатом синергии описанных выше моделей и методов, интегрированные и усовершенствованные идеи и принципы которых легли в его основу.

### **Обзор средств проектирования баз данных**

Результатами хозяйственно-экономической деятельности всех организаций, от мала до велика, являются ежегодные колоссальные объёмы данных. IT-менеджеры находят все больше и больше способов использовать их для улучшения работы предприятий, более отчётливого понимания клиентов, более быстрой доставки товаров и снижения расходов, а также для других целей. Все это стимулирует огромные инвестиции в инструменты и технологии работы с большими данными. В своем отчете за август 2021 года компания IDC, которая занимается исследованием рынка, оценила ожидаемые мировые расходы на системы больших данных и аналитики в \$215,7 млрд. в 2021 году, что на 10,1% больше, чем в 2020 году. Она также прогнозировала, что таковые расходы будут расти на 12,8% в год до 2025 года. Из этого следует, что мир больших данных развивается колоссальными темпами.

Существует множество инструментов больших данных с открытым исходным кодом, некоторые из которых также предлагаются в коммерческих версиях или как часть платформ больших данных и управляемых



услуг. Ниже представлен обзор пяти популярных инструментов и технологий с открытым исходным кодом для управления и анализа больших данных с кратким описанием их ключевых функций и возможностей [5].

### 1. Oracle

*Oracle RDBMS* (она же Oracle Database) занимает первое место среди СУБД. Система популярна у разработчиков, проста в использовании, у нее понятная документация, поддержка длинных наименований, JSON, улучшенный тег списка и Oracle Cloud. К основным достоинствам Oracle можно отнести поддержку баз данных колоссального объема (до 64 Гбайт), мощные средства разработки и администрирования, поддержку многопроцессорности и двух языковых сред, а также интеграцию с Web. Вместе с этим программа предъявляет серьезные аппаратные требования и высокую цену.

### 2. MySQL

MySQL поддерживается Linux, Windows, OSX, FreeBSD и Solaris. Можно начать работать на бесплатной версии, а затем перейти на коммерческую. Лицензия GPL с открытым исходным кодом открывает возможность модифицирования ПО MySQL.

Эта система управления базами данных использует стандартную форму SQL. Утилиты для проектирования таблиц имеют интуитивно понятный интерфейс. MySQL поддерживает до 50 миллионов строк в таблице. Предельный размер файла для таблицы по умолчанию 4 ГБ, однако он может быть увеличен. Поддерживает секционирование и репликацию, а также Xpath и хранимые процедуры, триггеры и представления.

### 3. Microsoft SQL Server

Наиболее популярная коммерческая СУБД. Она привязана к Windows, однако это плюс, если используются продукты Microsoft. Функционал SQL Server зависит от платформы. И графический интерфейс, и программное обеспечение основаны на командах. Основной язык запросов — *Transact-SQL*, создан совместно Microsoft и Sybase. Transact-SQL - это реализация стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Применяется для работы с базами данных размером от персональных до больших баз данных уровня предприятия; создаёт конкуренцию другим СУБД в этом сегменте рынка. Поддерживает SQL, непроцедурные, нечувствительные к регистру и общие языки баз данных.

### 4. PostgreSQL

Масштабируемая объектно-реляционная база данных, которая работает на Linux, Windows, OSX и в нескольких других системах. В PostgreSQL 10 есть такие функции, как логическая репликация, декларативное разбиение таблиц, улучшенные параллельные запросы, более безопасная аутентификация по паролю на основе SCRAM-SHA-256. Она основана на открытом коде, а это означает, что систему базы данных постоянно улучшает профильное сообщество из более чем 600 участников. Применяется в таких крупных компаниях, как Apple, Cisco, Fujitsu, Skype, and IMDb.

### 5. MongoDB

Самая популярная NoSQL система управления базами данных. Больше всего подходит для динамических запросов и определения индексов. Гибкая структура, которую можно модифицировать и расширять. Поддерживает Linux, OSX и Windows, но размер БД ограничен 2,5 ГБ в 32-битных системах. Использует платформы хранения MMAPv1 и WiredTiger.

Представим технические характеристики средств проектирования БД в таблице 1.

Таблица 1 – Основные данные о СУБД

Наименование СУБД	Oracle	Microsoft SQL Server	MySQL	PostgreSQL	MongoDB
Разработчик	Oracle Corporation	Oracle Corporation	Microsoft Corporation	PostgreSQL Global Development Group	MongoDB Inc. (2007)
Язык программирования	Assembly, C, C++	C, C++	C, C++	C	C++
Последняя версия (на ноябрь 2022 г.)	21c	15.0	8.0.31	15.1	6.0.1
Блог	Oracle NoSQL	Microsoft SQL Server Blog	SQL Server Blog	PostgreSQL	MongoDB
Скачивание с сервера	Oracle NoSQL	MySql	Microsoft SQL Server	PostgreSQL	MongoDB

Все перечисленные СУБД регулярно обновляются и качественно поддерживаются своими разработчиками.



## Обзор и сравнительный анализ существующих СУБД

В настоящее время в мире используется достаточно большое количество универсальных промышленных СУБД, рейтинг их развития представлен на рис. 2 [5].

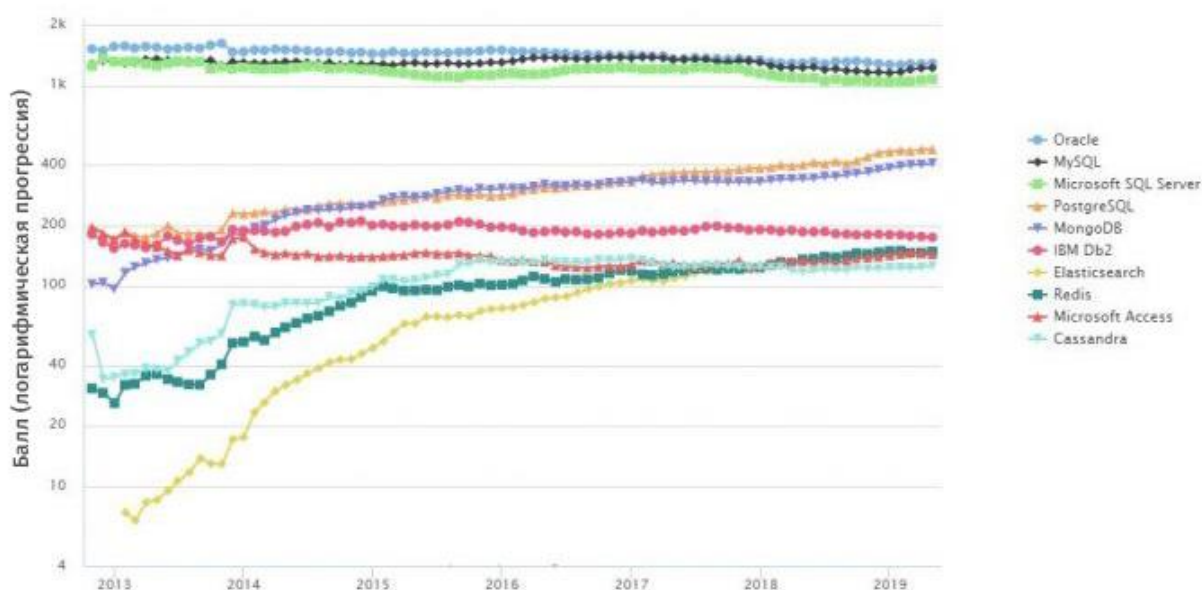


Рисунок 2 – Рейтинг СУБД

Следует отметить, что такие СУБД, как Oracle, Microsoft SQL Server, MySQL, PostgreSQL, MongoDB относят к БД высшего уровня проектирования. Для осуществления количественного анализа описанных СУБД представим их рейтинговые оценки, согласно [6], в таблице 2.

Таблица 2 — Рейтинг СУБД

Рейтинг			СУБД	Модель базы данных	Балл		
Ноябрь 2022	Октябрь 2022	Ноябрь 2021			Ноябрь 2022	Октябрь 2022	Ноябрь 2021
1	1	1	Oracle	Реляционная СУБД	1241.69	+5.32	-31.04
2	2	2	MySQL	Реляционная СУБД	1205.54	+0.17	-5.98
3	3	3	Microsoft SQL Server	Реляционная СУБД	912.51	-12.17	-41.78
4	4	4	PostgreSQL	Реляционная СУБД	623.16	+0.44	+25.88
5	5	5	MongoDB	Документная СУБД	477.90	-8.33	-9.45

Рейтинговые оценки популярности СУБД выставлялись по следующим критериям:

- количество упоминаний систем на веб-сайтах,
- общий интерес к системе,
- частота технических обсуждений системы,
- количество предложений о работе, в которых упоминалась система,
- количество профилей в профессиональных сетях (на примере LinkedIn),
- актуальность в соцсетях.

По результатам приведенной в таблице информации можно заявить, что *Oracle* является абсолютным лидером на рынке популярных систем (на протяжении 2021-2022 гг.), догоняет его по рейтинговой таблице — *MySQL*.

Для выполнения качественно анализа представленных СУБД определим их особенности и выделим сильные стороны каждой из рассматриваемых СУБД.

### 1. Особенности *Oracle*:

- работает с большими данными;
- поддерживает SQL, к нему можно получить доступ из реляционных БД Oracle;
- Oracle NoSQL Database с Java/C API используются для чтения и записи данных.

2. Особенности MySQL:
    - масштабируемость;
    - лёгкость использования;
    - безопасность;
    - поддержка Novell Cluster;
    - скорость;
    - поддержка многих операционных систем.
  3. Особенности *Microsoft SQL Server*:
    - высокая производительность;
    - зависимость от платформы;
    - возможность установить разные версии на одном компьютере;
    - генерация скриптов для перемещения данных.
  4. Особенности *PostgreSQL*:
    - поддержка табличных пространств, а также хранимых процедур, объединений, представлений и триггеров;
    - восстановление на момент времени (PITR);
    - асинхронная репликация;
    - NoSQL-базы данных.
  5. Особенности *MongoDB*:
    - высокая производительность;
    - автоматическая фрагментация;
    - работа на нескольких серверах;
    - поддержка репликации Master-Slave;
    - данные хранятся в форме документов JSON;
    - возможность индексировать все поля в документе;
    - поддержка поиска по регулярным выражениям.
- Производительность** СУБД оценивается:
- длительностью выполнения запросов;
  - скоростью поиска данных в неиндексированных полях;
  - временем выполнения операций импортирования базы данных из других форматов;
  - скоростью индексирования и выполнения таких объёмных операций, как обновление, вставка, удаление данных;
  - наибольшим числом параллельных запросов к данным в многопользовательском режиме;
  - временем формирования отчета из сгенерированных данных.

На производительность СУБД влияют два фактора:

- СУБД, которые следят за целостностью данных, несут дополнительную нагрузку, которую не испытывают другие программы;
- производительность собственных прикладных программ ощутимо зависит от проектирования и построения базы данных для конкретной предметной области.

Исходя из представленных особенностей каждой системы управления базами данных, мы можем сделать вывод, что самой высокой производительностью обладают: *MongoDB* и *Microsoft SQL Server*.

**Целостность данных.** Самой лучшей способностью сохранения целостности данных обладает *PostgreSQL*, так как данная СУБД имеет точку восстановления перед моментом падения базы данных, что крайне важно для всех пользователей. Уровень высокой **безопасности данных** хорошо реализован в СУБД MySQL. Администратор имеет возможность назначать системе различные права доступа на уровне файла, поля, а также организовать автоматическое шифрование данных. Все перечисленные СУБД предназначены для работы в многопользовательских средах.

**Доступ к данным посредством языка запросов SQL** (Structured Query Language) предоставлен во многих известных СУБД [7] как базовый либо как альтернативный. Язык SQL является международным стандартом языка запросов и широко используется как специалистами в области обработки данных, так и непосредственно пользователями.

Приведенные характеристики функционирования СУБД отражают специфику проектирования БД, а также определяют популярность их использования организациями, в т.ч. и их собственную конъюнктуру на мировом рынке.

## **Заключение**

В данной работе проведен системный анализ существующих методов и средств проектирования БД. Выполненный сравнительный анализ характеристик СУБД даёт возможность выделить лидеров по следующим критериям.

1. Надежностью сохранения целостности данных на уровне базы данных характеризуются СУБД *PostgreSQL*.

2. Высокий уровень безопасности обеспечивается в *MySQL*.

3. При сравнении СУБД по критерию доступа к данным посредством языка SQL хорошие характеристики имеют: *MS SQL Server* и *Oracle*.

Выбор СУБД пользователями зависит лишь от конкретной задачи, поставленной заказчиком проекта перед разработчиками, однако бесспорным лидером в рейтинге лучших систем признан *Oracle*.

## **Список литературы**

1. Халимон, В.И. Базы данных: учебное пособие / В.И. Халимон, Г.А. Мамаева, А.Ю. Рогов, В.Н. Чепикова. – С-Пб.: СПбГТИ(ТУ), 2017. – 118 с.

2. Базы данных: учебное пособие для обучающихся по направлению подготовки 09.03.03 «Прикладная информатика» / Сост.: Т.Ж. Базаржапова, О.А. Гармаева, А.Ю. Хаптахаяев. – Улан-Удэ: ФГБОУ ВО БГСХА, 2022. – 84 с.

3. Что такое база данных? | Oracle СНГ [Электронный ресурс] - Режим доступа: <https://www.oracle.com/cis/database/what-is-database/>. - Загл. с экрана.

4. Методы проектирования БД [Электронный ресурс] - Режим доступа: <https://helpiks.org/8-66618.html> - Загл. с экрана.

5. ТОП-10 систем управления базами данных [Электронный ресурс] - Режим доступа: <https://proglab.io/p/databases-2019> - Загл. с экрана.

6. DB-Engines Ranking [Электронный ресурс] - Режим доступа: <https://db-engines.com/en/ranking> - Загл. с экрана.

7. Почему лучше выбрать MS SQL Server кратко [Электронный ресурс] - Режим доступа: <https://midbox.com.ua/instrukciya/sql/pochemu-luchshe-vybrat-ms-sql-server-kratko> - Загл. с экрана.

*Якубенко В.В., Дмитриук Т.Г. Анализ подходов к проектированию баз данных. Целью работы является обзор методов и средств проектирования баз данных, выделение преимуществ и недостатков их проектирования и использования, а также проведение сравнительного анализа существующих систем управления базами данных.*

**Ключевые слова:** база данных, методы, средства, СУБД, проектирование.

*Yakubenko V.V., Dmitriuk T.G. Analysis of approaches to database design. The purpose of the work is to review the methods and tools of database design, highlight the advantages and disadvantages of their design and use, as well as conduct a comparative analysis of existing database management systems.*

**Key words:** database, methods, tools, DBMS, designing.

## Анализ криптографических протоколов удаленного доступа к ПК

К.Н. Ефименко<sup>\*1</sup>, Д.В. Пауков<sup>\*2</sup>, Н.С. Кострыкин<sup>\*3</sup>, Е.Д. Калинин<sup>\*3</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
KN\_Efimenko@mail.ru

<sup>\*2</sup> магистрант, Донецкий национальный технический университет,  
denshik591@gmail.com

<sup>\*3</sup> бакалавр, Донецкий национальный технический университет,  
kostrykin2007@gmail.com

*Ефименко К.Н., Пауков Д.В., Кострыкин Н.С., Калинин Е.Д. Анализ криптографических протоколов удаленного доступа к ПК. Выполнен краткий сравнительный анализ основных характеристик, особенностей использования и преимуществ современных криптографических протоколов, обеспечивающих защищенную передачу данных между узлами в сети Интернет и при удаленном доступе к ПК. TLS/SSL – протокол для шифрования передаваемой информации и аутентификации. IPSec – набор протоколов для безопасного обмена информацией по сетевому протоколу IP. SSH – протокол для удаленного управления операционными системами и туннелирования TCP-соединения. Описаны этапы установления соединения.*

**Ключевые слова:** анализ, криптографический протокол, TLS/SSL, IPSec, SSH.

### Введение

Сетевой протокол – это комплекс установок, благодаря которым определяется и регулируется процесс информационного обмена между компьютерами, подключенными к Интернету.

Функционирование сети основывается на работе сразу нескольких протоколов, располагаемых на разных уровнях. В настоящее время распространены две сетевых модели передачи данных.

TCP/IP – сетевая модель передачи данных, которая описывает способ передачи данных от источника информации к получателю. Модель предусматривает прохождение информации через четыре уровня, каждый из которых описывается протоколом передачи данных, на которых базируется Интернет.

OSI/ISO – сетевая модель стека сетевых протоколов, с помощью которой различные сетевые устройства взаимодействуют между собой. Модель предусматривает семь уровней взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

Наиболее часто используемыми криптографическими протоколами, обеспечивающими защищенную передачу данных между узлами в сети Интернет, являются: TLS/SSL, IPSec и SSH-соединения. Каждый из перечисленных протоколов имеет свои особенности использования, достоинства и недостатки, но все они были приняты как Интернет-стандарт.

Целью данной работы является выполнение краткого анализа наиболее распространенных современных криптографических протоколов удаленного доступа к ПК, выбор и углубленное изучение которых должны помочь будущим специалистам в области прикладной математики стать востребованными на рынке труда.

### Криптографические протоколы удаленного доступа

Проанализируем особенности использования и преимущества трех современных криптографических протоколов удаленного доступа.

#### 1. Протокол TLS/SSL.

В настоящее время все более актуальным становится использование цифровых сертификатов. Особенностью данной технологии является использование протоколов TLS/SSL. На рынке IT-услуг появились компании, которые бесплатно предоставляют цифровые сертификаты всем желающим, чтобы гарантировать шифрование трафика между посещаемыми сайтами и браузером клиента.

Сетевые протоколы SSL и TLS являются криптографическими протоколами, обеспечивающими аутентификацию и защиту от несанкционированного доступа, нарушения целостности передаваемых данных. Протоколы SSL/TLS предназначены для исключения подмены идентификатора на клиентской или серверной стороне, раскрытия или искажения данных. Для этих целей используется надежный метод аутентификации, применяются шифрование канала связи и коды целостности сообщений.

Протокол TLS/SSL изначально был разработан компанией Netscape для защиты данных между сервисными и транспортными протоколами. Первая обнародованная версия была выпущена в 1995 году. Широко используется для VoIP-приложений, сервисов обмена мгновенными сообщениями. TLS/SSL представляет собой безопасный частный аутентифицированный канал. При транспортировке сообщений осуществляется проверка целостности с использованием MAC. Протокол TLS/SSL использует как симметричный, так и асимметричный ключи [1].

Протокол TLS/SSL обеспечивает решение двух задач – шифрование передаваемой информации и передача информации именно туда, куда требуется (аутентификация). Основное назначение протокола – предоставление надежного способа обмена данными между приложениями. Реализация SSL выполнена в виде многослойной среды, которая используется для безопасной передачи информации посредством незащищенных каналов связи.

Установка соединения обеспечивается в несколько этапов [2]:

1) Клиент устанавливает соединение с сервером и запрашивает защищенное подключение. Это может обеспечиваться либо установлением соединения на порт, который изначально предназначен для работы с SSL/TLS, например, 443, либо дополнительным запросом клиентом установки защищенного соединения после установки обычного.

2) При установке соединения клиент предоставляет список алгоритмов шифрования, которые он «знает». Сервер сверяет полученный список со списком алгоритмов, которые «знает» сам сервер, и выбирает наиболее надежный алгоритм, после чего сообщает клиенту, какой алгоритм использовать

3) Сервер отправляет клиенту свой цифровой сертификат, подписанный удостоверяющим центром, и открытый ключ сервера.

4) Клиент может связаться с сервером доверенного центра сертификации, который подписал сертификат сервера, и проверить, валиден ли сертификат сервера. Но может и не связываться. В операционной системе обычно уже установлены корневые сертификаты центров сертификации, с которыми сверяют подписи серверных сертификатов, например, браузеры.

5) Генерируется сеансовый ключ для защищенного соединения. Это делается следующим образом:

- клиент генерирует случайную цифровую последовательность;
- клиент шифрует ее открытым ключом сервера и посылает результат на сервер;
- сервер расшифровывает полученную последовательность при помощи закрытого ключа.

Учитывая, что алгоритм шифрования является асимметричным, расшифровать последовательность может только сервер. При использовании асимметричного шифрования используется два ключа – приватный и публичный. Публичным отправляемое сообщение шифруется, а приватным расшифровывается. Расшифровать сообщение, имея публичный, ключ нельзя.

6) Таким образом, устанавливается зашифрованное соединение. Данные, передаваемые по нему, шифруются и расшифровываются до тех пор, пока соединение не будет разорвано.

TLS/SSL имеет множество мер безопасности [3]:

- защита от понижения версии протокола к предыдущей (менее защищенной) версии или менее надежному алгоритму шифрования;
- нумерация последовательных записей приложения и использование порядкового номера в коде аутентификации сообщения (MAC);
- использование ключа в идентификаторе сообщения (только владелец ключа может сгенерировать код аутентификации сообщения). Алгоритм вычисления кода аутентификации (HMAC), используемый во многих сессиях TLS, определен в RFC 2104;
- сообщение, которым заканчивается подтверждение связи («Finished»), используется для подтверждения аутентичности ранее переданных сообщений и, таким образом, выбранных параметров TLS-соединения.

В общем случае применение криптографии в протоколах TLS/SSL значительно снижает производительность приложений, зато обеспечивает надежную защиту передачи данных. Протоколы не требуют практически никаких настроек с клиентской стороны, считаются самыми распространенными протоколами защиты в сети интернет. Последними версиями протокола являются TLS 1.2, опубликованная в 2008 году и TLS 1.3 – 2018 года.

## 2. Протокол IPsec.

Протоколы прикладного уровня, поддерживающие шифрование (например, HTTPS) не могут охватить все возможные сценарии использования. Поэтому реализация шифрования на сетевом уровне обеспечивает большую гибкость.

IPsec (Internet Protocol Security) – набор (стек) протоколов для безопасного обмена информацией по сетевому протоколу IP. Позволяет осуществлять подтверждение подлинности (аутентификацию), проверку целостности и/или шифрование IP-пакетов. IPsec также включает в себя протоколы для защищенного обмена ключами в сети Интернет. IPsec является набором стандартов Интернета и своего рода «настройкой» над IP-протоколом. В основном, применяется для организации VPN-соединений [3].

Базовой особенностью протокола является понятие SA (Security Association) – это набор параметров о том, как стороны будут в дальнейшем использовать те или иные свойства протоколов из состава IPsec.

Набор служб безопасности, которые может предоставлять IPSec, включает в себя контроль доступа, целостность без установления соединения, аутентификацию источника данных, отказ от повторных пакетов (форма частичной целостности последовательности), конфиденциальность (шифрование) и ограниченность конфиденциальности трафика. Поскольку эти услуги предоставляются на уровне IP, они могут использоваться любым протоколом более высокого уровня, например TCP, UDP и т.д. [4].

IPSec использует два протокола для обеспечения безопасности трафика:

- AH (Authentication Header) обеспечивает целостность без установления соединения, аутентификацию источника данных и дополнительную службу защиты от повтора. AH использует хэш-алгоритм для вычисления значения хэша, как для полезной нагрузки, так и для заголовка пакета, обеспечивая целостность пакета. Однако это вызывает очень специфическую проблему. AH не будет работать через NAT-устройство. NAT изменяет IP-заголовок пакета во время перевода, но значение хэша не изменяется. Таким образом, принимающее устройство будет полагать, что пакет был изменен при передаче и произойдет отклонение пакета;

- ESP (Encapsulating Security Payload) обеспечивает как полную, так и ограниченную конфиденциальность (шифрование) трафика, а также может обеспечивать подключение. Протокол обеспечивает целостность без установления соединения, аутентификацию источника данных и службу защиты от повтора. Таким образом, ESP выполняет шифрование и по своей сути более безопасен, чем AH. ESP вводит в пакет как дополнительный заголовок, так и трейлер. ESP также использует алгоритм хеширования для целостности данных. Однако хэш не включает IP-заголовок пакета, и, таким образом, ESP будет (обычно) работать через NAT-устройство.

Оба протокола AH и ESP являются транспортными средствами для контроля доступа на основе распределения криптографических ключей и управления потоками трафика по отношению к этим протоколам безопасности. Эти протоколы могут применяться отдельно или в сочетании друг с другом для обеспечения требуемого набора служб безопасности [4].

Каждый протокол поддерживает два режима: транспортный и режим туннеля. В транспортном режиме протоколы обеспечивают защиту в основном для протоколов верхнего уровня; в туннельном режиме протоколы применяются к туннелированным IP-пакетам. IPSec позволяет пользователю (или системному администратору) контролировать степень детализации, в которой предлагается служба безопасности. Например, можно создать один зашифрованный туннель для переноса всего трафика между двумя шлюзами безопасности или отдельный зашифрованный туннель, который может быть создан для каждого TCP-соединения между каждой парой хостов, взаимодействующих через эти шлюзы. Руководство IPSec должно включать средства для указания:

- какие службы безопасности использовать и в каких комбинациях;
- гранулярность, при которой должна применяться данная защита;
- алгоритмы, используемые для обеспечения криптографической безопасности [4].

Поскольку эти службы безопасности используют общие секретные значения (криптографические ключи), IPSec опирается на отдельный набор механизмов для размещения этих ключей (ключи используются для служб аутентификации/целостности и шифрования). Этот документ требует поддержки как ручного, так и автоматического распределения ключей. Он определяет конкретный подход на основе открытого ключа для автоматического управления ключами, но могут использоваться другие автоматизированные методы распространения ключей.

Каждый протокол IPSec (AH или ESP) может работать в двух режимах – транспортном и туннельном [4].

Транспортный режим – используется для шифрования поля данных IP пакета, содержащего протоколы транспортного уровня (TCP, UDP, ICMP). Исходные IP-заголовки остаются нетронутыми (рис. 1). Используется при обеспечении связи с одного устройства на другое одно устройство.

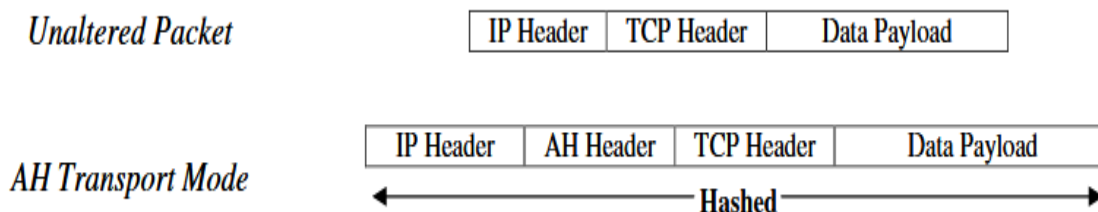


Рисунок 1 – Изменение IP-пакета протоколом AH

Туннельный режим – предполагает шифрование всего пакета, включая заголовок сетевого уровня. Во время транзита к пакету применяется временный IP-заголовок (рис. 2). Туннельный режим применяется в случае необходимости скрытия информационного обмена организации с внешним миром.

Взаимодействие двух узлов начинается с установления SA. Точнее с двух согласований – для протокола AH и ESP причем в одну и в другую стороны. SA начинается с аутентификации и затем стороны согласовывают будущие параметры сессии [5]:

1) для протокола AH – используемый алгоритм аутентификации, ключи, время жизни ключей и другие параметры;

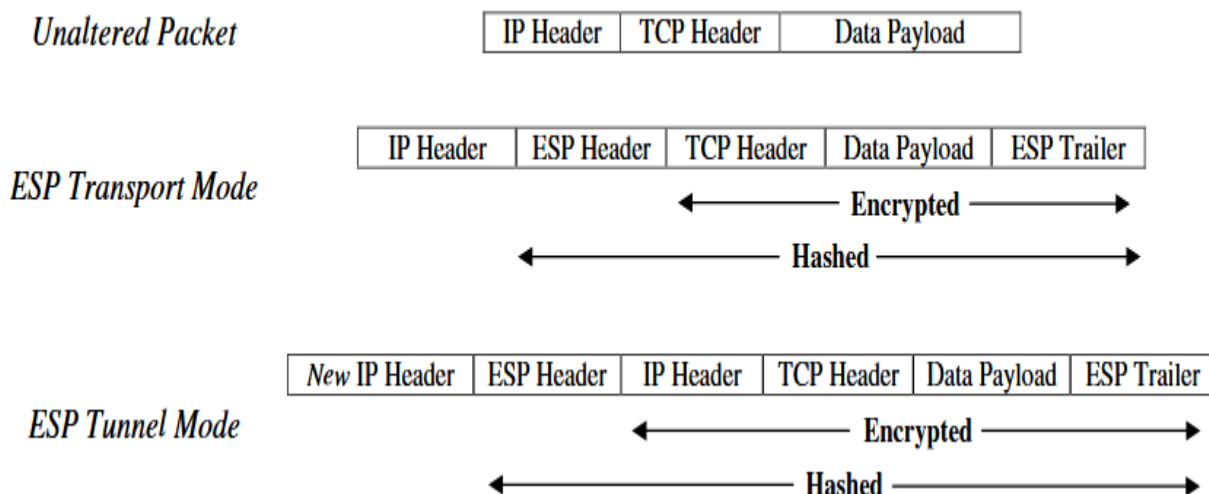


Рисунок 2 – Изменение IP-пакета протоколом ESP

2) для протокола ESP – алгоритмы шифрования и аутентификации, ключи, параметры инициализации, время жизни ключей и другие параметры.

При этом стороны договариваются о туннельном или транспортном режиме работы IPSec.

Установка соединения проходит две фазы [5]:

На фазе 1 происходит установление SA первой фазы. В первой фазе стороны договариваются о методе идентификации, алгоритме шифрования, алгоритме хэширования и группе Диффи-Хеллмана. Эта фаза может пройти путем обмена тремя нешифрованными пакетами (агрессивный режим) или шестью нешифрованными пакетами – стандартный режим. Если все прошло успешно, то создается SA фазы 1 под названием IKE SA (IKE – протокол, связывающий все компоненты IPSec в работающее целое) и осуществляется переход ко второй фазе.

На фазе 2 стороны договариваются о политике и создаются сами ключи. Эта фаза, в отличие от первой полностью шифруется и она наступает только в случае успешного окончания первой фазы. В связи с тем, что трафик этой фазы полностью шифрован, становится сложно осуществлять поиск неполадок, однако если все прошло успешно, то создается SA фазы 2 под названием IPsec SA. В этот момент можно сказать, что туннель установлен.

Согласованные на двух фазах ключи должны работать оговоренное политикой время. Это означает, что сторонам возможно предстоит пережить процедуру смены ключей (rekeying), а иначе согласованные SA распадутся. Как было сказано выше, у сторон есть ключи в рамках процесса фазы 1 (IKE) и фазы 2 (IPsec). Процедуры их смены различны, как и таймеры, которые за это отвечают. Для того, чтобы не было перерыва связи в процессе смены ключей стороны сначала согласовывают параметры новой SA и лишь после этой успешной процедуры уничтожают старую SA [5].

Важным аспектом является способность IPsec устанавливать соединение заново. Для этого существуют настройки в `ispsec.conf`, однако детали этих настроек могут отличаться от версии ПО.

### 3. Протокол SSH.

SSH (Secure Shell) – это протокол удаленного администрирования, разработанный для осуществления удаленного управления операционными системами и туннелирования TCP-соединения. Использование этого протокола допускает использование разных алгоритмов шифрования, что позволяет безопасно работать практически в любой незащищенной среде: работать с ПК через командную оболочку, передавать по зашифрованному каналу любой тип данных [6]. Протокол относится к прикладному уровню сетевой модели передачи данных.

Протокол SSH организует безопасное соединение над небезопасными каналами передачи данных. Особенностью протокола является шифрование по выбранному алгоритму всего передаваемого трафика, в том числе и паролей.

Первый релиз протокола состоялся в 1995 г, а уже в 1996 г была представлена усовершенствованная его версия, которая и стала основой для дальнейшего развития продукта. Протокол SSH-1 имел некоторые ошибки в схеме обеспечения безопасности. Он был средством безопасной аутентификации. Протокол второй версии, то

есть SSH-2, отвечает всем современным требованиям к шифрованию и обеспечивает невозможность подмены трафика путем пересылки контрольных сумм [7].

SSH – это коммерческий продукт и предоставляется на платной основе. Присутствует и бесплатная версия – OpenSSH, которую использует большинство программистов. Обе версии схожи между собой командами. Релиз последнего обновления протокола OpenSSH 8.6 за апрель 2021 г. можно найти по ссылке <http://www.openssh.com/txt/release-8.6>.

Сегодня для всех сетевых ОС доступны SSH сервер и SSH клиент, а сам протокол SSH является одним из самых популярных решений для удаленного управления системами и передачи важной информации.

SSH-сервер обычно прослушивает соединения на TCP-порту 22. Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов электронно-цифровой подписи RSA или DSA, но допускается также аутентификация при помощи и даже IP-адреса [3].

Аутентификация по паролю наиболее распространена. При каждом подключении подобно HTTPS вырабатывается общий секретный ключ для шифрования трафика.

При аутентификации по ключевой паре предварительно генерируется пара открытого и закрытого ключей для определённого пользователя. На машине, с которой требуется произвести подключение, хранится закрытый ключ, а на удалённой машине – открытый. Эти файлы не передаются при аутентификации, система лишь проверяет, что владелец открытого ключа также владеет и закрытым. При данном подходе, как правило, настраивается автоматический вход от имени конкретного пользователя в ОС.

Аутентификация по IP-адресу небезопасна, эту возможность чаще всего отключают.

Для создания общего сеансового ключа используется алгоритм Диффи-Хеллмана. Для шифрования передаваемых данных используется симметричное шифрование, алгоритмы AES, Blowfish или 3DES. Целостность передачи данных проверяется с помощью CRC32 в SSH1 или HMAC-MD5 в SSH2.

Для сжатия шифруемых данных может использоваться алгоритм LempelZiv (LZ77), который обеспечивает такой же уровень сжатия, что и архиватор ZIP. Сжатие SSH включается лишь по запросу клиента, и на практике используется редко [3].

Одной из основных особенностей протокола является туннелирование. SSH-туннель – это туннель, создаваемый посредством SSH-соединения и используемый для шифрования туннелированных данных. Используется для того, чтобы обезопасить передачу данных в Интернете (аналогичное назначение имеет IPsec). При пересылке через SSH-туннель незашифрованный трафик любого протокола шифруется на одном конце SSH-соединения и расшифровывается на другом.

Основным недостатком протокола SSH является то, что он не имеет средств защиты от действий злоумышленника, получившего root-доступ. Одной из мер предосторожности является запрет на удалённый root-доступ.

Один из наиболее популярных SSH-клиентов – программа Putty. Она используется для обеспечения удаленного доступа. Клиент FireSSH реализован как расширение браузера Firefox. Такой способ делает возможным использование программы в разных ОС. Для браузера Google Chrome есть другой официальный клиент – Secure Shell. Для использования в UNIX подходит программа OpenSSH. Для ее применения желателен опыт работы с командной оболочкой SSH. Для Windows подходит эмулятор терминалов Xshell, который позволяет отправлять команды нескольким серверам одновременно [6].

Алгоритм установления соединения в протоколе SSH можно разделить на три уровня, каждый из которых располагается над предыдущим: транспортный (открытие защищённого канала), аутентификация, подключение. В качестве предварительного уровня можно добавить уровень установки сетевого соединения, хотя официально этот уровень находится ниже SSH.

## **Заключение**

При использовании SSL/TLS одним из основных методов является метод MITM (Man In The Middle – человек посередине). Этот метод основывается на использовании серверного сертификата и ключа на каком-то узле, который будет прослушивать трафик и расшифровывать информацию, которой обмениваются сервер и клиент. Для организации прослушивания можно использовать, например, программу sslsniff. Поэтому корневой сертификат и ключ обычно желательно хранить на машине, которая не подключена к сети, для подписания приносить запросы на подпись на флэшке, подписывать и так же уносить.

Использование IPsec протокола имеет как положительные стороны: высокая криптоустойчивость; возможность использования L2TP внутри IPsec для аутентификации по имени пользователя и паролю, так и отрицательные: сложен для настройки и поиска неисправностей; большие накладные расходы на передачу трафика в канале за счет заголовков.

Протокол IPsec используется, в основном, для организации VPN-туннелей. В этом случае протоколы ESP и AH работают в режиме туннелирования.

Использование SSH подключения имеет ряд преимуществ: безопасная работа на удаленном ПК с использованием командной оболочки; использование разных алгоритмов шифрования (симметричного, асимметричного и хеширования); возможность безопасного использования любого сетевого протокола, что



позволяет передавать по защищенному каналу файлы любого размера [6]. Таким образом, SSH – один из самых безопасных протоколов для реализации удаленного доступа к ПК, что делает его самым популярным вариантом для удаленного администрирования компьютеров и безопасной передачи данных.

Область применения протокола SSH практически не ограничена. Исходя из его основной функции – удаленного входа в операционную систему, протокол используют: системные администраторы для удаленной настройки компьютеров локальной сети; для настройки почтовых служб; для скрытого обмена внутри сети массивными файлами; для Интернет-игр.

Таким образом, без использования криптографических протоколов при удаленном доступе не возможна работа ни одной компании. Удаленный доступ требуется повсеместно как для сотрудников, так и для технического персонала.

## **Литература**

1. Starlink [Электронный ресурс] / Режим доступа: <https://www.starlink.ru/articles/ssl-tls>. – Загл. с экрана.
2. Блог про Linux, Bash и другие информационные технологии [Электронный ресурс] / Режим доступа: <https://mnorin.com/tls-ssl-neobhodimy-j-minimum-znaniy.html>. – Загл. с экрана.
3. Свободная энциклопедия Википедия [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org>. – Загл. с экрана.
4. Сайт lanmarket.ua [Электронный ресурс] / Режим доступа: <https://lanmarket.ua/entsiklopediya/telekommunikatsionnye-tehnologii/ipsec.html>. – Загл. с экрана.
5. VPNKI [Электронный ресурс] / Режим доступа: <https://ua.vpnki.ru/questions/technologies/ipsec>. – Загл. с экрана.
6. Хостинг Freehost [Электронный ресурс] / Режим доступа: <https://freehost.com.ua/faq/wiki/chto-takoe-ssh/>. – Загл. с экрана.
7. Сила технологий [Электронный ресурс] / Режим доступа: <https://ipipe.ru/info>. – Загл. с экрана.

**Ефименко К.Н., Пауков Д.В., Кострыкин Н.С., Калинин Е.Д. Анализ криптографических протоколов удаленного доступа к ПК.** Выполнен краткий сравнительный анализ основных характеристик, особенностей использования и преимуществ современных криптографических протоколов, обеспечивающих защищенную передачу данных между узлами в сети Интернет и при удаленном доступе к ПК. TLS/SSL – протокол для шифрования передаваемой информации и аутентификации. IPSec – набор протоколов для безопасного обмена информацией по сетевому протоколу IP. SSH – протокол для удаленного управления операционными системами и туннелирования. TCP-соединения. Описаны этапы установления соединения.

**Ключевые слова:** анализ, криптографический протокол, TLS/SSL, IPSec, SSH.

**Efimenko K.N., Paukov D.V., Kostykin N.S., Kalinin E.D. Analysis of cryptographic protocols for remote PC access.** Performed a brief comparative analysis of the main characteristics, uses and advantages of modern cryptographic protocols that provide secure transmission of data between nodes on the Internet and with remote access to the PC. TLS/SSL – protocol for encrypting transmitted information and authentication. IPSec – a set of protocols for the secure exchange of information over a network protocol IP. SSH – protocol for remote control of operating systems and tunneling. The steps to establish a connection are described.

**Keywords:** analysis, cryptographic protocol, TLS/SSL, IPSec, SSH.

## Предотвращение утечек конфиденциальной информации дизайн-студии технологиями DLP

Е.Ю. Потребя<sup>\*1</sup>, Н.Е. Губенко<sup>\*2</sup>

<sup>\*1</sup> магистрант, Донецкий национальный технический университет,  
potrebart@gmail.com

<sup>\*2</sup> к.т.н., доцент, Донецкий национальный технический университет,  
negubenko@mail.ru

**Потребя Е.Ю., Губенко Н.Е. Предотвращение утечек конфиденциальной информации дизайн-студии технологиями DLP.** В статье описывается проблематика возникновения инцидентов информационной безопасности. Актуализируется проблема утечки конфиденциальной информации. Приводится модель инфологической схемы для визуализации потоков обмена данными дизайн-студии. Рассматривается процесс внедрения технологий Data Leak Prevention в систему управления компанией. Анализируется функционирование системы для разграничения доступа в зависимости от занимаемых должностей сотрудников дизайн-студии. Затрагивается вопрос необходимости использования криптографической защиты данных в системе управления.

**Ключевые слова:** информационная безопасность, конфиденциальность, утечка данных, DLP-система, дизайн-студия, криптография

### Введение

Сложившаяся экономическая ситуация значительно усилила конкурентную борьбу компаний за позиции на рынке, а иногда и за выживание. Некоторые компании, разрабатывая антикризисные стратегии, значительно сократили расходную часть по некоторым статьям своего бизнеса. Нередко, под сокращение расходов попадает информационная безопасность. Однако, экономия чревата утечками персональных данных и конфиденциальной информации, что может навредить бизнесу в целом.

Аналитики компании Falcongaze, утверждают, что половину рисков, с которыми может столкнуться компания, составляют внутренние угрозы. В период кризиса, показатели внутренних угроз могут подниматься до 60%. Всё дело в том, что во времена экономической нестабильности происходят случаи ротации и даже увольнения кадров. По этой причине недовольные сотрудники пытаются либо перепродать важную информацию конкурентам, либо уничтожить базы данных, считая их своей собственной наработкой. [1]

Стоит отметить, что деятельность дизайн-студий всегда была связана с обработкой и хранением большого количества конфиденциальных данных. Рабочий процесс дизайн-студии осуществляется посредством системы, в которой обрабатывается и хранится информация о текущих и реализованных проектах, о сотрудниках и их занятости, о персональных данных заказчиков.

Для предотвращения инцидентов информационной безопасности критически важно внедрить в систему управления дизайн-студией DLP-технологии (Data Leak Prevention), которые позволят анализировать перемещение данных, как внутри корпоративной сети, так и за её пределы, предотвращая утечку важной информации, согласно установленным правилам и политикам.

Целью статьи является актуализация вопросов безопасности конфиденциальных данных в информационных системах и внедрение технологий для минимизации соответствующих инцидентов.

### Система управления дизайн-студией

Дизайн-студия — это организация, занимающаяся созданием веб-сайтов и фирменных стилей, отличительными особенностями которой является одновременное ведение нескольких проектов и постоянный контакт с различными заказчиками для уточнения, согласования и расчетов по заказанным услугам.

В состав организационной структуры рассматриваемой дизайн-студии входят сотрудники следующих должностей: арт-директор, проджект-менеджер, маркетолог, программист, дизайнер, бухгалтер, юрист, системный администратор. Предприятие оперирует огромными объемами данных: как результатами работы сотрудников, так и отчетностью и статистикой.

Программным образом система реализуется языком разработки веб-приложений PHP, веб-фреймворком Laravel, реляционной системой управления базами данных MySQL и программным стеком WAMP.

Для визуализации потоков обмена данными внутри системы спроектирована инфологическая модель. Она

представляет собой ориентированную на человека и не зависимую от типа СУБД модель предметной области, определяющую совокупности информационных объектов, их атрибутов и отношений между объектами, динамику изменений, а также характер информационных потребностей. [2]

Инфологическая схема дизайн-студии приведена на рис. 1:

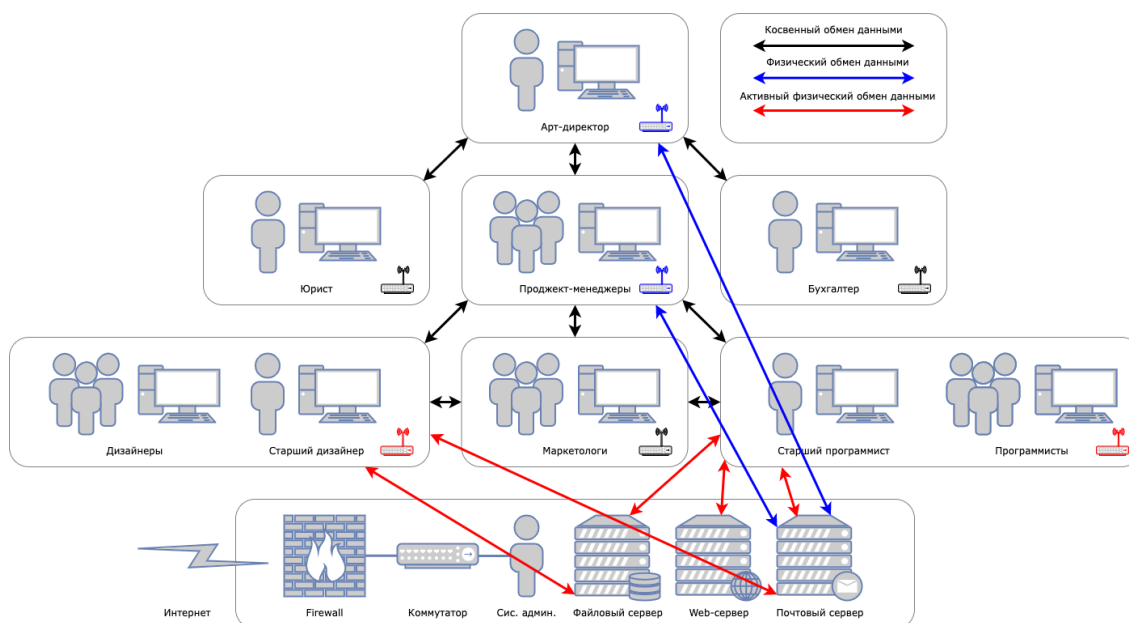


Рисунок 1 – Инфологическая схема дизайн-студии

На представленной инфологической схеме показаны потоки информации между участниками информационных процессов. На физическом уровне веб-сервер, файловый и почтовый сервер соединены с Интернетом через защищенный фаерволлом канал связи. Далее с помощью беспроводных точек доступа по локальной сети разные отделы обращаются к серверам.

Большое количество сотрудников и объем связей разных типов обмена данных повышают вероятность появления угроз конфиденциальности информационных ресурсов.

### **Угрозы конфиденциальности информационных ресурсов**

Источниками угрозы сохранности конфиденциальных данных могут являться как компании-конкуренты и злоумышленники, так и сотрудники вместе с органами управления компанией. Цель любой угрозы заключается в том, чтобы повлиять на целостность, полноту и доступность данных.

Угрозы бывают внешними и внутренними. Внешние угрозы представляют собой попытки получить доступ к данным извне и сопровождаются взломом серверов, сетей, аккаунтов работников и считыванием информации из технических каналов утечки (акустическое считывание с помощью жучков, камер, наводки на аппаратные средства, получение виброакустической информации из окон и архитектурных конструкций).

В свою очередь, внутренние угрозы подразумевают неправомерные действия персонала, рабочего отдела или управления фирмы. В результате пользователь системы, который работает с конфиденциальной информацией, может выдать информацию посторонним. На практике такая угроза встречается чаще остальных. Работник может годами предоставлять конкурентам секретные данные. Это легко реализуется, ведь действия авторизованного пользователя администратор безопасности не квалифицирует как угрозу.

Поскольку внутренние ИБ-угрозы связаны с человеческим фактором, отслеживать их и управлять ими сложнее. Предупреждать инциденты можно с помощью деления сотрудников на группы риска.

Попытка несанкционированного доступа может происходить несколькими путями:

через сотрудников, которые могут передавать конфиденциальные данные посторонним, забирать физические носители или получать доступ к охраняемой информации через печатные документы;

через сотрудников, которые могут передавать конфиденциальные данные посторонним, забирать физические носители или получать доступ к охраняемой информации через печатные документы;

с помощью программного обеспечения злоумышленники осуществляют атаки, которые направлены на кражу пар «логин-пароль», перехват криптографических ключей для расшифровки данных, несанкционированного копирования информации.

с помощью аппаратных компонентов автоматизированной системы, например, внедрение прослушивающих устройств или применение аппаратных технологий считывания информации на расстоянии

(вне контролируемой зоны). [3]

Для предотвращения кражи, изменения и распространения конфиденциальной информации настоятельно рекомендуется использование соответствующих комплексных программных решений.

### **Технологии предотвращения утечки данных**

Технологии защиты от утечек информации базируются на выявлении, предотвращении, регистрации и устранении последствий инцидентов информационной безопасности или событий, нарушающих регламентированные процедуры защиты ИБ.

В рамках обеспечения информационной безопасности дизайн-студии особое внимание должно обращать на защиту конфиденциальных данных от внутренних угроз. Таким образом вокруг системы управления компанией должен быть создан защищенный цифровой «периметр», который будет анализировать всю исходящую, а в ряде случаев и входящую информацию.

Контролируемой информацией должен быть не только интернет-трафик, но и ряд других информационных потоков: документы, которые выносятся за пределы защищаемого контура безопасности на внешних носителях, распечатываемые на принтере, отправляемые на мобильные носители через Bluetooth и т.д.

Поскольку система Data Leak Prevention должна препятствовать утечкам конфиденциальной информации, то она в обязательном порядке имеет встроенные механизмы определения степени конфиденциальности документа, обнаруженного в перехваченном трафике. Как правило, наиболее распространены два способа: путём анализа специальных маркеров документа и путём анализа содержимого документа. В настоящее время более распространён второй вариант, поскольку он устойчив перед модификациями, вносимыми в документ перед его отправкой, а также позволяет легко расширять число конфиденциальных документов, с которыми может работать система.

Принцип работы DLP-систем заключается в анализе всего трафика, который находится в пределах защищаемой корпоративной сети. Внедрение DLP-системы помогает контролировать входящие и исходящие потоки данных и блокировать попытки несанкционированной передачи важных корпоративных данных.

DLP работает по принципу data-centric security. Он подразумевает не защиту серверов, программного обеспечения или сетей, а контроль безопасности данных, которые обрабатываются в системе. Согласно этому принципу, все потоки информации разделяют на три категории:

Data-in-use – вся информация, с которой работают пользователи (создание и редактирование документов, медиа-контента).

Data-at-rest – информация, которая статично хранится на конечных устройствах пользователей и в местах общего доступа.

Data-in-motion – данные в процессе движения, передаваемые информационные потоки (транзакции, информация об авторизации, запросы «сервер-клиент» и другие).

Помимо своей основной задачи, связанной с предотвращением утечек информации, DLP-системы также хорошо подходят для решения ряда других задач, связанных с контролем действий персонала.

Наиболее часто DLP-системы применяются для решения следующих неосновных для себя задач:

- контроль использования рабочего времени и рабочих ресурсов сотрудниками;
- мониторинг общения сотрудников с целью выявления «подковерной» борьбы, которая может навредить организации;
- контроль правомерности действий сотрудников (печать поддельных документов и пр.);
- выявление сотрудников, рассылающих резюме, для оперативного поиска специалистов на освободившуюся должность. [4]

При внедрении DLP-системы важно придерживаться не только принципов защиты информации, но и норм законодательства. Контроль за соблюдением правил работы с конфиденциальной информацией не должен нарушать личные права пользователей, поэтому стоит отказаться от действий, которые могут быть расценены как слежка. Дополнительно стоит предусмотреть механизмы контроля администраторов системы, у которых есть доступ ко всем типам данных. Чтобы избежать недовольства и возмущения в коллективе, в общие сведения о работе системы рекомендуется включить пункты, где четко обозначить цели внедрения DLP-контроля и описать, как использование системы защиты информации способствует финансовому благополучию компании. Отдельно стоит подчеркнуть, что руководитель дизайн-студии имеет право на защиту коммерческой тайны организации, а компьютеры и другая техника, которую предоставляет работнику, являются собственностью компании, и для защиты собственности может применяться любая система защиты. [5]

Для обеспечения максимально возможной защиты информации в процессе внедрения DLP следует выполнять все рекомендации и использовать сразу несколько блоков защиты. Это позволит создать экономически выгодный, рабочий защитный контур. Внедрение DLP-системы должно выполняться поэтапно от подготовки до проектирования и настройки компонентов для работы под нагрузкой в компании.

### **Внедрение DLP-технологий в систему управления дизайн-студией**

Перед процессом внедрения DLP важно провести подготовительные процедуры. Процесс подготовки компании к установке системы защиты состоит из аудита защищенности информации, оценки рисков и урегулирования юридических вопросов. Аудит подразумевает оценку реальной степени защиты информации. На этом подготовительном отрезке идет поиск возможных каналов уязвимостей в данной экосистеме.

Помимо этого, в рамках внедрения DLP-технологий в систему управления дизайн-студией необходимо провести обследование информационных потоков, которое включает:

1. Оценку уровня безопасности при работе с внутренними документами компании.
2. Детальное изучение всех технических ресурсов компании, от серверов до сетевых потоков.
3. Создание перечня данных, которые относятся к группе информации с ограниченным доступом.
4. Разработка правил разграничения доступа.
5. Изучение процессов обработки, создания, передачи и хранения информации в рамках компании.

Оценка риска и создание правил разграничения доступа – обязательные шаги на этапе внедрения экономически эффективной DLP-системы. Риски оцениваются наряду с обследованием потенциальных каналов утечки. В зависимости от вероятного ущерба принимается решение о необходимости защиты канала утечки. [6]

Таким образом, опишем функционирование системы управления дизайн студией для создания правил разграничения доступа – набора прав, которые получает пользователь системы в зависимости от занимаемой должности. На рис. 2 изображена диаграмма вариантов использования разрабатываемой системы, которая включает в себя функции доступные администратору, сотрудникам и клиентам дизайн-студии:

Функции, доступные администратору: добавление сотрудников, назначение сотрудников, изменение данных о клиенте, изменение данных о сотруднике, изменение этапа разработки, просмотр данных о заказе.

Функции, доступные клиенту: изменение данных о клиенте, заполнение брифа, добавление проекта.

Функции, доступные сотруднику: изменение данных о клиенте, изменение данных о сотруднике, изменение этапа разработки, просмотр данных о заказе.

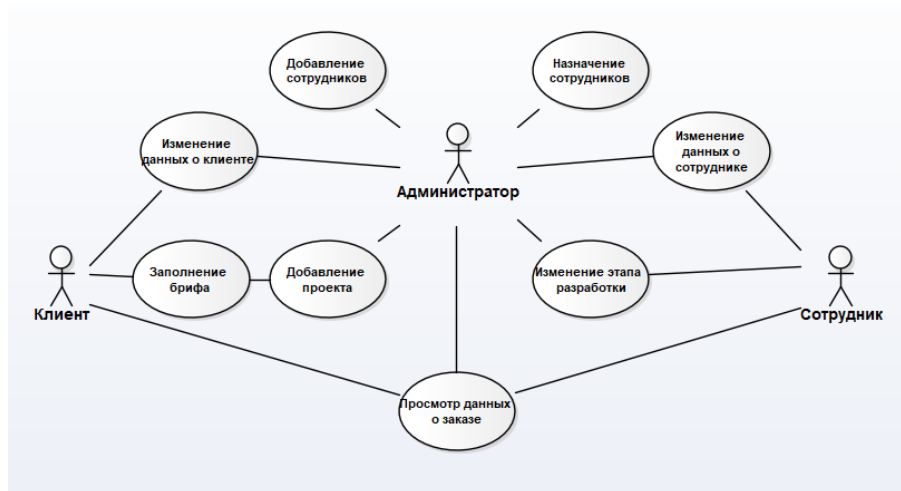


Рисунок 2 – Диаграмма вариантов использования

Чтобы предотвратить возможность несанкционированного доступа к конфиденциальной информации, решено использовать отдельные личные кабинеты для реализации разделения прав пользователей системы.

В свою очередь, для обеспечения безопасности всей системы, особое внимание необходимо уделить вопросу надежного хранения и шифрования аутентификационных данных от созданных личных кабинетов пользователей. Для предотвращения рисков, связанных с утечкой данных и обеспечения информационной безопасности общества, прежде всего, применяются методы криптографической защиты посредством шифрования данных.

В рамках защиты конфиденциальной информации дизайн-студии должен использоваться криптостойкий алгоритм с высокой скоростью шифрации и дешифрации (за счет генерации таблиц замены).

Стоит отметить, что для дизайн-студии, где безопасность данных входит в число бизнес-приоритетов, внедрение DLP – оптимальный выбор. Успешная интеграция DLP позволит контролировать все потоки информации, а также вовремя выявлять и устранять угрозы безопасности.

## Выводы

В статье описана проблематика возникновения инцидентов информационной безопасности. Актуализирована проблема утечки конфиденциальной информации. Приведена разработанная инфологическая схема для визуализации потоков обмена данными дизайн-студии. Рассматривается процесс внедрения технологий Data Leak Prevention в систему управления компанией. Проанализировано функционирование системы для разграничения доступа в зависимости от занимаемых должностей сотрудников дизайн-студии. Затронут вопрос необходимости использования криптографической защиты персональных данных в системе управления. Подчеркнута важность проведения дальнейших работ для комплексного обеспечения информационной безопасности дизайн-студии.

## Литература

1. DLP-системы — элемент информационной безопасности [Электронный ресурс]. Режим доступа: <https://www.azone-it.ru/dlp-sistemy-vazhnaya-sostavlyayushchaya-informacionnoy-bezopasnosti-predpriyatiya> – Загл. с экрана.
2. Инфологическая модель предметной области [Электронный ресурс]. Режим доступа: [https://studbooks.net/2278354/informatika/infologicheskaya\\_model\\_predmetnoy\\_oblasti](https://studbooks.net/2278354/informatika/infologicheskaya_model_predmetnoy_oblasti) – Загл. с экрана.
3. Основы информационной безопасности [Электронный ресурс]. Режим доступа: <https://цбис.рф/osnovy-informatsionnoj-bezopasnosti> – Загл. с экрана.
4. Информационная безопасность. Защита данных с помощью DLP-системы [Электронный ресурс]. Режим доступа: <https://searchinform.ru/informatsionnaya-bezopasnost/dlp-sistemy> – Загл. с экрана.
5. Внедрение DLP-системы на предприятии [Электронный ресурс]. Режим доступа: [https://spravochnik.ru/informatika/vnedrenie\\_dlp-sistemy\\_na\\_predpriyatii](https://spravochnik.ru/informatika/vnedrenie_dlp-sistemy_na_predpriyatii) – Загл. с экрана.
6. Информационная безопасность. Внедрение DLP-систем [Электронный ресурс]. Режим доступа: <https://searchinform.ru/informatsionnaya-bezopasnost/dlp-sistemy/vnedrenie-dlp-sistem> – Загл. с экрана.

**Потреба Е.Ю., Губенко Н.Е. Предотвращение утечек конфиденциальной информации дизайн-студии технологиями DLP.** В статье описывается проблематика возникновения инцидентов информационной безопасности. Актуализируется проблема утечки конфиденциальной информации. Приводится модель инфологической схемы для визуализации потоков обмена данными дизайн-студии. Рассматривается процесс внедрения технологий Data Leak Prevention в систему управления компанией. Анализируется функционирование системы для разграничения доступа в зависимости от занимаемых должностей сотрудников дизайн-студии. Затрагивается вопрос необходимости использования криптографической защиты данных в системе управления.

**Ключевые слова:** информационная безопасность, конфиденциальность, утечка данных, DLP-система, дизайн-студия, криптография

**Potreba Efim, Gubenko Natalia Prevention of leaks of confidential information of a design studio by DLP technologies.** The article describes the problems of information security incidents. The problem of leakage of confidential information is being updated. The model of the infological scheme for visualization of data exchange flows of the design studio is given. The process of implementing Data Leak Prevention technologies into the company's management system is considered. The functioning of the system for access differentiation is analyzed depending on the positions held by design studio employees. The issue of cryptographic protection of personal data in the management system is touched.

**Keywords:** information security, confidentiality, data leakage, DLP system, design studio, cryptography

## Реализация бизнес-системы сервисного центра по обслуживанию телефонов на «1С: Предприятие»

К.Н. Ефименко<sup>\*1</sup>, С.А. Шлыков<sup>\*2</sup>, Е.С. Исаев<sup>\*2</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
KN\_Efimenko@mail.ru

<sup>\*2</sup> бакалавр, Донецкий национальный технический университет,  
sersh2002@gmail.com

*Ефименко К.Н., Шлыков С.А., Исаев Е.С. Реализация бизнес-системы сервисного центра по обслуживанию телефонов на «1С: Предприятие». Выполнено описание и анализ применения программного продукта 1С: Предприятие, предназначенного для автоматизации деятельности предприятия. Рассмотрена актуальность использования программного продукта на современных предприятиях. Обосновано применение данного программного продукта для автоматизации бухгалтерского и управленческого учётов бизнес-системы сервисного центра по обслуживанию телефонов.*

**Ключевые слова:** 1С: Предприятие, архитектура, конфигурация, сервисный центр.

### Введение

Задачи учета и управления могут существенно отличаться в зависимости от рода деятельности предприятия, отрасли, специфики продукции или оказываемых услуг, размера и структуры предприятия, требуемого уровня автоматизации. Трудно себе представить одну программу, предназначенную для массового использования и удовлетворяющую при этом потребностям большинства предприятий. При этом руководителю, с одной стороны, необходимо решение, соответствующее специфике именно его предприятия, но, с другой стороны, он понимает преимущества применения массового проверенного продукта. Сочетание этих потребностей и обеспечивает «1С: Предприятие» как система программ.

Система программ «1С: Предприятие» предназначена для решения широкого спектра задач автоматизации учета и управления, стоящих перед динамично развивающимися современными предприятиями.

«1С: Предприятие» представляет собой систему прикладных решений, построенных по единым принципам и на единой технологической платформе. Руководитель может выбрать решение, которое соответствует актуальным потребностям предприятия и будет в дальнейшем развиваться по мере роста предприятия или расширения задач автоматизации [1].

Целью данной работы является выполнение краткого обзора современной системы программ «1С: Предприятие» для обоснования ее выбора в качестве средства реализации бухгалтерской и управленческой систем сервисного центра по обслуживанию телефонов.

### Описание и анализ применения «1С: Предприятие»

Система программ «1С: Предприятие» состоит из технологической платформы (ядра) и разработанных на ее основе прикладных решений («конфигураций»). Такая архитектура системы принесла ей высокую популярность, поскольку обеспечивает открытость прикладных решений, их функциональность и гибкость, короткие сроки внедрения, высокую производительность, масштабируемость от одного до десятков тысяч рабочих мест, работу в режиме «облачного» сервиса и на мобильных устройствах [2].

### 1. Архитектура

Технологическая платформа «1С: Предприятие» представляет собой программную оболочку над базой данных. Используются базы на основе DBF-файлов в 7.7, собственный формат 1CD с версии 8.0 или СУБД Microsoft SQL Server на любой из этих версий. Кроме того, с версии 8.1 хранение данных возможно в PostgreSQL и IBM DB2, а с версии 8.2 добавилась и Oracle. Платформа имеет свой внутренний язык программирования, обеспечивающий, помимо доступа к данным, возможность взаимодействия с другими программами посредством OLE и DDE, в версиях 7.7, 8.0 и 8.1 — с помощью COM-соединения.

Клиентская часть платформы функционирует в среде Microsoft Windows, а начиная с версии 8.3, также в среде Linux и Mac OS X. Начиная с версии 8.1, серверная часть платформы в клиент-серверном варианте работы «1С: Предприятия» может функционировать на ОС Microsoft Windows и Linux.

Существуют специальные версии среды исполнения 1С для ноутбуков и PDA, ПО создания веб-приложений, взаимодействующих с базой данных «1С: Предприятие».

## **2. История создания**

Программное обеспечение 1С было впервые представлено в 1991 году и с тех пор динамично развивается и совершенствуется в плане качества функционала с учетом изменений как в законодательной области так в технологиях. За прошедшие более тридцати лет было выпущено несколько поколений программного продукта начиная от версии 1.x (для ОС MS DOS) и заканчивая версиями 8.x (для ОС MS Windows, Linux и MacOS). Последней версией на данный момент является 8.3.21.1393 (дата релиза 19.06.2022 г.).

В настоящее время наиболее распространенными являются версии 8.x.

### **2.1. Версии 8.x**

Все «компоненты» программной оболочки включены в базовую поставку (платформа теперь не делится на «компоненты»: бухгалтерский учёт, расчёт, оперативный учёт, управление распределёнными базами данных, как это было с версиями 7.x). Что дает преимущество для пользователя, который приобретая любую конфигурацию (к примеру, «Управление торговлей»), получает доступ ко всем возможностям платформы, например, использовать бухгалтерские счета и периодические расчёты. Серверная «компонента» 1С 8.1 может работать в операционной системе Linux и использовать PostgreSQL.

В сравнении с 7.x существенные изменения претерпел встроенный язык программирования, который не совместим с 6.x и 7.x. «Новый» язык предлагает к использованию достаточно большое количество разнообразных фиксированных объектов.

В версии 8.x существует два основных режима: режим предприятия и режим конфигуратора (с встроенным режимом отладки).

Платформа «1С: Предприятие» 8.x допускает конфигурирование, одновременное использование одной базы несколькими пользователями при наличии необходимого количества дополнительных лицензий.

При использовании со специализированными «базовыми» конфигурациями обеспечивает работу в «базовом» режиме с ограничением возможностей: не допускает конфигурирования, не поддерживает режим внешнего соединения, не имеет многопользовательского режима. Для работы в «базовом» режиме используется облегчённый вариант лицензирования, не требуются аппаратные ключи защиты, а стоимость такого продукта существенно ниже.

Кроме того, выпускается учебная версия платформы 1С: Предприятие 8. Она входит, например, в состав продуктов «1С: Бухгалтерия 8. Учебная версия», «1С: Предприятие 8. Версия для обучения программированию» и ряд других.

В комплекте с лицензиями поставляются ключи защиты Aladdin HASP (USB), локальные H1M1 для одной лицензии и сетевые NetHASP для пяти и более (до 100), есть также ключи Aladdin HardLock на 300 и 500 лицензий. Использование одиночной (локальной) лицензии возможно только на том компьютере, на котором установлен ключ, причем только на системной консоли, использовать его через сеть или на удалённом рабочем столе невозможно. На один компьютер можно установить только один ключ каждой серии (H1M1, NetHASP или HardLock).

1С: Предприятие 8 может использоваться в файловом режиме, с хранением базы в виде файла, или в клиент-серверном режиме с хранением баз на сервере SQL. Клиент-серверный режим требует дополнительного приобретения лицензии на «сервер 1С: Предприятия 8».

Для создания веб-порталов используется «веб-расширение», но платформа – ASP.NET и рекомендуемый внутренний язык C#, так как большая часть объектов доступа к данным уже предоставляется именно на нём. Соответственно, платформа тоже Windows NT (W2k, XP, Win2003) и сервер IIS (версий 5, 5.1, 6).

Фирма 1С прекратила поддержку 1С: Предприятия 8.0 на версии 8.0.18.2, уведомила о прекращении поддержки 1С: Предприятия 8.1 с апреля 2011 года (последняя версия 8.1.15.14, выпущена в октябре 2009). Текущие версии платформы 1С: Предприятие 8.2 обеспечивают обратную совместимость с версиями 8.0 и 8.1.

### **2.2. Версия 8.2**

Главной отличительной особенностью является режим «управляемое приложение», при котором пользовательский интерфейс описывается декларативно, а его вид зависит от типа клиентского ПО:

- «толстый» клиент («старое» клиентское приложение версий 8.0 и 8.1);
- «тонкий» клиент;
- веб-клиент (включает клиентскую и серверную части).

Клиент-серверное взаимодействие реализовано с использованием технологий AJAX (DHTML), XMLHttpRequest и JavaScript.

Также отличается более «строгой» реализацией клиент-серверной архитектуры, в частности, требует переноса выполнения всей бизнес-логики в серверный код, реализована динамическая балансировка нагрузки в кластера серверов, добавлена поддержка СУБД Oracle. 25 сентября 2009 года выпущена окончательная версия 8.2.9.

В новой версии платформы реализована отказоустойчивость кластера серверов «1С: Предприятия» — основной «компоненты», обеспечивающей взаимодействие между пользователем и СУБД.

В выпущенном обновлении платформы 8.2.11 проведена оптимизация внутренних механизмов и



реализованы некоторые новые возможности: работа с криптозащитой; экспорт документов в форматы Microsoft Office; отладка веб-клиента; форматированный документ; поддержка браузеров Google Chrome и Safari.

### **2.3. Версия 8.3**

В данной версии произведены следующие «крупные» изменения:

- заметно снизилась и продолжает снижаться производительность программы, в связи с увеличением системных требований;
  - предоставление пользователям нативных 64-битных клиентов под Linux и MacOS (клиентские приложения существуют только для Mac OS X 10.8 и выше, и выпускаются для целей бета-тестирования);
  - 64-битный клиент и Конфигуратор для Windows;
  - полноценная мобильная платформа для iOS, Android и Windows Phone;
  - переработка механизма расположения элементов в формах;
  - изменения в интерфейсных механизмах.
- Разработчики-пользователи получили большое количество изменений:
- возможность создавать расширения конфигурации, позволяющие изменять конфигурацию без снятия её с поддержки;
  - улучшение механизмов хранилища конфигурации, при этом сравнения объектов перестаёт работать корректно и выдаёт ложные отличия в одинаковых объектах;
  - механизм рефакторинга кода;
  - механизм автоматизированного тестирования интерфейса;
  - выгрузка конфигурации в файлы текстового формата, в том числе частичная.

С этой версии, на разработчиков легла обязанность разделять код по доступности «на клиенте» или «на сервере», что усложнило процесс разработки [3].

### **3. Достоинства и недостатки системы**

С момента выхода программы «1С: Предприятие» на рынок, ее лидерство было определено на долгие годы.

«1С: Предприятие» это не готовая программа для автоматизации бизнеса, а программная оболочка. Для работы нужна конфигурация базы данных. Конфигурации, как правило, поставляются в составе комплекта, и при установке 1С на компьютере предлагается сразу выбрать и установить конфигурацию.

Конфигурация определяет основной функционал: электронные формы для ввода данных, и программы на языке 1С, выполняющие все расчеты, печатные формы, отчеты, обработки для анализа данных, внешний вид (меню и кнопки основного окна) и т.д.

Исходя из этого 1С обладает целым рядом преимуществ, определивших ее лидерство на рынке программного обеспечения для бизнеса.

1. Конфигурация «открыта», что позволяет в действующей конфигурации в любой момент внести корректировки, доработки по улучшению ее работы, по учету особенностей бизнеса, учесть пожелания пользователя, расширить функции. Другие программы обычно недоступны для корректировок.

2. Любой программист 1С может прочитать программный код, разобраться в работе имеющейся конфигурации и внести соответствующие доработки. Если в программе есть ошибки, то программист может их найти и исправить.

3. Модель, выбранная создателями 1С настолько удачна для решения именно проблем учета (хотя она и не идеальна), что перевод учета из другой системы управления базами данных в 1С может стоить дешевле, чем корректировки в исходной базе данных.

4. В 1С имеется возможность использовать «внешние отчеты и обработки» – отдельные файлы, содержащие программный код, интерфейс и печатные формы, которые «запускаются» в среде 1С и могут выполнять практически любые функции.

К недостаткам «1С: Предприятия» можно отнести следующие особенности.

1. Работа над конфигурацией в среде 1С отличается от разработки программ на таких обычных языках высокого уровня.

2. 1С относится к СУБД – системам управления базами данных, только в отличие от двух наиболее распространенных СУБД, 1С более специализирована. Многие задачи, на которые в других (неспециализированных) СУБД можно потратить дни, в 1С решаются за часы или даже минуты.

3. Ряд функций в 1С невозможно реализовать. В первую очередь работу с графикой. Можно описать изделие каким-то перечнем параметров (длина, высота, ширина и т.д.), но нарисовать это изделие программа не сможет, хотя в большинстве случаев это и не требуется [4].

### **4. Актуальность применения системы**

Представители современного бизнеса должны учитывать новые технологии и использовать их в своей работе. Это касается крупных организаций, средних и небольших компаний. Внедрение 1С является одним из важных мероприятий, необходимых для повышения эффективности бизнеса.

Внедрение системы 1С в организации обеспечивает повышение эффективности ведения бухгалтерского учета. Система обладает отличным функционалом. Повышается не только скорость выполнения работы специалистами организации, но и качество результатов.

Комплекс 1С – достаточно гибкое решение для бизнеса. С помощью IT-специалистов можно вносить определенные изменения в структуру форм отчетности, улучшать функционал системы. Наряду с внедрением 1С в организации желательно заключить договор на поддержку системы. Это обеспечит необходимую помощь со стороны квалифицированных специалистов.

Внедрение 1С на предприятии осуществляется поэтапно. При этом можно использовать определенные конфигурации системы, в зависимости от специфики организации («Бухгалтерия», «Торговля», «Управление производством» и т.д.). После установки и настройки системы проводится комплексное тестирование. Персонал компании проходит соответствующее обучение работе с 1С.

Внедрение «1С: Предприятие» обеспечивает оптимизацию бизнес-процессов и повышение эффективности работы специалистов финансового, бухгалтерского, производственного отделов. Благодаря использованию системы улучшается качество работы организации в целом. Применение современных инструментов способствует успешному развитию вашего бизнеса [5].

### **Проект сервисного центра по обслуживанию телефонов на «1С: Предприятие»**

В современном мире сервисные центры по ремонту различной техники являются малыми и средними предприятиями, в которых так или иначе необходимо вести бухгалтерский учет. В данном случае таким предприятием является сервисный центр по обслуживанию телефонов. Данный сервисный центр достаточно востребован в нашем обществе, так как у каждого человека есть хотя бы одно мобильное устройство.

Данный сервисный центр будет иметь отдел приема, осмотра и продажи, и отдел ремонта и обслуживания. В основные функции данного предприятия будут входить ремонт, замена прошивки, продажа аксессуаров и заказ мобильных устройств.

Система «1С: Предприятие» была выбрана для реализации данного проекта так как есть несколько причин, а именно:

- открытый код – дает возможность другим программистам реализовать новые программные модули или доработать имеющиеся;
- простота использования – по нажатию одной кнопки программа способна распечатать несколько документов;
- удобное представление данных – данные представлены в виде таблиц, строки в которых можно не только сортировать, но и редактировать;
- запросы – в 1С: Предприятие существует возможность создавать запросы, упрощающие поиск данных в таблицах и многое другое.

Именно поэтому для реализации данного проекта было принято решение использовать именно 1С: Предприятие. При этом проект будет иметь следующую структуру (рис.1).

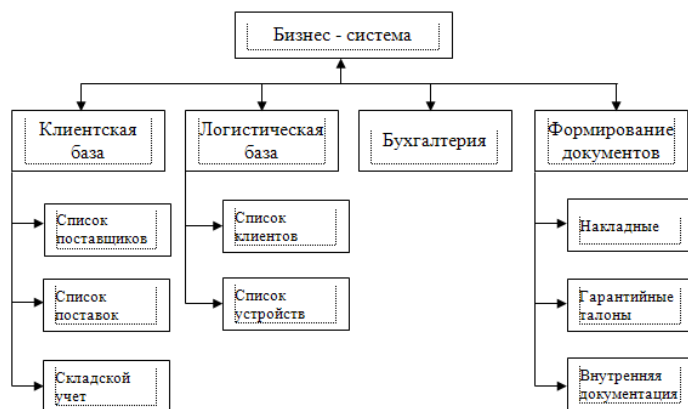


Рисунок 1 – Основные программные модули сервисного центра

### **Заключение**

Несмотря на то, что система «1:С Предприятие» является далеко не единственной на рынке программных продуктов аналогичного назначения, она, безусловно, относится к наиболее популярным. Сам факт умения работать в данной программе намного повышает значимость любого специалиста. Изучение принципов системы, освоение ее функциональных возможностей во многом способствует грамотной постановке учетных процессов на предприятии.

Возможности системы, позволяющие использовать одновременно несколько различных валют, а также несколько разных планов счетов, во многом способствуют решению актуальных задач управленческого учета. Например, параллельно с планом счетов бухгалтерского учета, действующим на территории Российской Федерации, можно использовать Международный план счетов GAAB.

Решению задач консолидированного учета во многом способствует возможность интеграции системы «1С: Предприятие» с другими системами (в частности, поддержка нескольких Интернет-протоколов, прием и отправка электронной почты, поддержка DBF- и XML-форматов).

### **Литература**

1. Портал 1С.ru [Электронный ресурс] / Режим доступа: <https://1c.ru/rus/products/1c/predpr/why-predpr.htm> – Загл. с экрана.
2. 1С:Предприятие 8 [Электронный ресурс] / Режим доступа: <https://v8.1c.ru/tekhnologii/overview/> – Загл. с экрана.
3. Свободная энциклопедия Википедия [Электронный ресурс] / Режим доступа: [https://ru.wikipedia.org/wiki/1С:Предприятие#Версия\\_8.4](https://ru.wikipedia.org/wiki/1С:Предприятие#Версия_8.4) – Загл. с экрана.
4. Allbets [Электронный ресурс] / Режим доступа: [https://revolution.allbest.ru/audit/00640778\\_0.html](https://revolution.allbest.ru/audit/00640778_0.html) – Загл. с экрана.
5. 1CLenta.ru [Электронный ресурс] / Режим доступа: <https://1clenta.ru/pattern/104> – Загл. с экрана.

***Ефименко К.Н., Шлыков С.А., Исаев Е.С. Реализация бизнес-системы сервисного центра по обслуживанию телефонов на «1С: Предприятие». Выполнено описание и анализ применения программного продукта 1С: Предприятие, предназначенного для автоматизации деятельности предприятия. Рассмотрена актуальность использования программного продукта на современных предприятиях. Обосновано применение данного программного продукта для автоматизации бухгалтерского и управленческого учётов бизнес-системы сервисного центра по обслуживанию телефонов.***

***Ключевые слова:*** 1С: Предприятие, архитектура, конфигурация, сервисный центр.

***Efimenko K.N., Shlykov S.A., Isaev E.S. Implementation of the business system of the telephone service center at 1C: Enterprise. The description and analysis of the application of the software product 1C: Enterprise, designed to automate the activities of the enterprise. The relevance of using a software product in modern enterprises is considered. The application of this software product for automation of accounting and management accounting of the business system of the telephone service center is justified.***

***Keywords:*** 1C: Enterprise, architecture, configuration, service center.

## Разработка информационной системы тестирования и учёта успеваемости обучающихся

О.В. Рычка<sup>\*1</sup>, Р.В. Самойленко<sup>\*2</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
olga\_rychka@mail.ru

<sup>\*2</sup> студент, Донецкий национальный технический университет,  
seltia0203@gmail.com

**Рычка О.В., Самойленко Р.В. Разработка информационной системы тестирования и учёта успеваемости обучающихся.** В статье представлен краткий обзор современных информационных систем тестирования и контроля успеваемости обучающихся. Представлен функционал разрабатываемой системы. Рассмотрены аналоги разрабатываемой системы тестирования и успеваемости обучающихся, указаны преимущества и недостатки этих систем.

**Ключевые слова:** информационная система, тестирование, C#, Telegram-бот, Python. вопрос теста, аналоги разрабатываемой системы.

### Введение

Сегодня тяжело представить себе мир без информационных технологий. Именно благодаря им, человечество смогло совершить много важных прорывов в науке.

Информатика – наука относительно молодая. Её формирование началось с середины XIX века. Во второй половине XX века информатика стала развиваться очень быстрыми темпами.

Для развития человеческого общества необходимы материальные, инструментальные, энергетические и другие ресурсы, в том числе и информационные. Настоящее время характеризуется небывалым ростом объёма информационных потоков. Это относится практически к любой сфере деятельности человека. Наибольший рост объёма информации наблюдается в промышленности, торговле, финансово-банковской и образовательной сферах.

В последние годы в сфере образования наблюдается стремительное увеличение интереса к автоматизации контроля результатов обучения.

В современном мире большую роль играет контроль знаний учащихся школ, работников, студентов вузов и т.д. Как известно, под контролем знаний понимается система научно обоснованной проверки результатов обучения. Более узкое определение гласит, что контроль - это выявление, измерение и оценка знаний, умений и навыков.

Существует множество форм контроля полученных знаний - экзамен, зачёт, аттестация, контрольная, опрос. Но наиболее популярным и корректным средством измерения знаний на сегодняшний день является тестирование, основанное на диалоге вычислительной системы и пользователя.

Тест - это совокупность специальным образом подготовленных и подобранных заданий, позволяющая провести выявление требуемых характеристик процесса обучения. Одно из главных преимуществ тестов состоит в том, что они позволяют опросить всех участников по всем вопросам нужного материала в одинаковых условиях, применяя при этом ко всем без исключения одну и ту же, заранее разработанную шкалу оценок [2].

Это значительно повышает объективность, обоснованность и правильность оценки, по сравнению, скажем, с экзаменом. Тесты и тестовые задания получили своё распространение в сферах, где нужно точно определить знания по всему курсу обучения, где большое количество сдающих, где нужен жёсткий отбор кандидатов, например, вступительные экзамены в учебные заведения, экзамены в ГИБДД, аттестация работников предприятий, приём специалистов на работу, аттестация и контроль знаний студентов, дистанционное образование и т.д.[1]

В эпоху информатизации общества, глобальной компьютеризации, развития технологии Интернет и передачи данных все более актуальными становятся разнообразные компьютерные системы тестирования, способные дополнить или заменить традиционные методы контроля и методики преподавания. Благодаря компьютерным системам тестирования стало намного удобнее проводить тестирование во всех сферах, где применялись и применяются обычные тесты. Например, дистанционное образование, которое стало распространённым способом получения знаний, особенно в последнее время.

Рассмотрим описание разрабатываемой системы тестирования и контроля успеваемости обучающихся.

Данная система будет иметь несколько ролей, а именно: преподаватель (учитель), студент (ученик), администратор.

Теперь рассмотрим функционал каждой роли:

- При разработке данной системы будет использоваться язык гипертекстовой разметки HTML, язык таблиц стилей CSS, язык программирования C#, язык программирования Python. Системой управления базами данных (СУБД) была выбрана PostgreSQL [7].

Преподаватель (учитель) благодаря этому Telegram–боту может просмотреть дисциплины, которые он ведёт, группы (классы), в которых он преподаёт, а также успеваемость группы (класса) по определённому предмету и учащемуся.

Системой предусмотрено несколько видов тестирования:

- В системе будет реализовано несколько видов вопросов для тестов:

- одиночный выбор – пользователь выбирает единственный правильный ответ;
- множественный выбор – пользователь выбирает несколько (два и более) правильных ответа;
- свободный ввод – пользователь вводит ответ в поле для ввода;
- на соответствие – пользователь должен сопоставить между собой варианты ответов [6].

Рассмотрим аналоги разрабатываемой системы тестирования и контроля успеваемости:

Информационная система тестирования знаний студентов «Таймлайн» – это сервис для проведения автоматизированной проверки знаний студентов в режиме реального времени. После авторизации пользователь попадает на страницу со всеми имеющимися тестами (см.рис.1).

Рисунок 1 – Страница с созданными тестами в информационной системе «Таймлайн»

Преимущества данной системы:

- интуитивно понятный и удобный интерфейс;
- быстрая обработка результатов тестирования;
- на страницах сервиса отсутствует информация, которая отвлекала бы студента от прохождения теста;
- возможность замены текстового вопроса видео – вопросом;
- система может построить график результатов для теста;
- предусмотрены различные виды тестовых вопросов;
- высокая система безопасности.

Недостатки данной системы:

- предсказуемость стандартизированного тестирования;
- в случае отключения интернета результат прохождения теста не сохраняется.

## 2. Информационный сервис "Мастер-тест"

Сервис "Мастер-тест" – это конструктор on-line тестов, который может быть использован при дистанционном обучении в школах, лицеях, техникумах, ВУЗах и так далее. В этом сервисе очень простая система создания тестовых вопросов (см.рис.2).

Добавить вопрос Изменить титул Проверить тест Результат

Тест

Самая высокая...

Заголовок: Самая высокая гора мира

Вопроса: .....

☐ Дополнительно

Тип вопроса: Однозначный Ответ

Ответ

☐ Эльбрус

☐ Акконкагуа

☐ Монблан

☒ Эверест

Вес Вопроса: 1

← Предыдущее | Далее →

Сохранить

Рисунок 2 – Создание тестового вопроса в сервисе "Мастер-тест"

Преимущества данной системы:

- бесплатный интернет сервис;
- интуитивно понятный и удобный интерфейс;
- возможность прохождения тестов без подключения к интернету, не устанавливая на компьютер дополнительных программ;
- возможность вставить готовый тест в свой сайт или блог с помощью html-кода;
- накопленные результаты тестирований можно проанализировать;
- проведение всевозможных интернет олимпиад на дистанционном уровне;

Недостатки данной системы:

- проблемы при публикации ссылки на онлайн тест, эта ссылка не всегда корректно работает;
- в процессе работы иногда случаются сбои;
- трудности повторного применения.

## Заключение

Тестирование является очень важным этапом оценки знаний учащихся и проводится с помощью специальных тестов, состоящих из заданного набора тестовых заданий. Тестовое задание – это чётко и ясно сформулированный вопрос, требующий однозначного ответа или выполнения определённого алгоритма

действий. Таким образом, сам тест является набором взаимосвязанных тестовых заданий, позволяющих оценить соответствие знаний ученика экспертной модели знаний в конкретной предметной области.

Использование тестовой формы контроля успеваемости в образовательных учреждениях обеспечивает:

- высокий уровень учебных достижений по предметам начальной ступени образования;
- воспитывает ценностное отношение к учебной деятельности;
- развивает память, логическое мышление учащихся;
- позволяет сделать правильный выбор;
- снижает уровень тревожности;
- помогает преподавателю осуществлять своевременно коррекцию полученных знаний.

Важной целью тестирования контроля успеваемости является мотивирование регулярной, напряжённой и целенаправленной работы учащихся.

Контроль и самоконтроль – важнейшие компоненты учебной деятельности. Осуществляя его систематически по отдельным темам, разделам учитель приучает учащихся к осознанному выполнению учебных задач на всех этапах усвоения знаний и умений.

С помощью тестов, в отличие от обычных проверочных работ, удобно проводить контроль, потому что обычная контрольная работа оценивает конечный результат, а тест позволяет установить его причину, благодаря поэтапному выполнению заданий, проверяющих те знания и умения, из которых складывается этот результат [5].

Информационная система тестирования успеваемости предоставляет учащимся возможность проявить самостоятельность, индивидуальность. Таким образом, эта система позволяет определить не только “проблемную зону”, но и конкретную “болевую точку”, даёт возможность установить причину итоговой неудачи, сделать выводы и построить соответственно коррекционную работу по устранению пробелов знаний.

Внедрение информационных систем тестирования позволяет экономить время преподавателей и обучающихся, а также более объективно оценить знания учащихся [3].

## **Литература**

1. Аванесов, В. С. Научные проблемы тестового контроля знаний / Аванесов, В. С- 3-е изд. – Москва: Центр тестирования, 2002. – 239 с.
2. Талызина Н.Ф. Методика составления обучающих программ / Москва: МГУ, 1990. – 46 с.
3. Бочкин А.И. О надёжности оценки доли знаний методом тестов с выбором варианта ответа / Санкт-Петербург : Питер, 2002. – 60 с.
4. Банокин П.И. Методы и средства проектирования информационных систем и технологий: учебное пособие / П.И. Банокин – Томск: Изд-во Томского политехнического университета, 2015. – 92 с.
5. Боброва Л.Н., Емельянова И.Н, Никулова Г.А. Когнитивные аспекты компьютерного контроля знаний // Информационные технологии в процессе подготовки современного специалиста. Межвузовский сборник. – 2001. – №4. – С.22-27
6. Аванесов, В. С. Композиция тестовых заданий / Аванесов, В. С- 1-е изд. – Москва: Центр тестирования, 1996. – 191 с.
7. Моргунов, Е. П. PostgreSQL. Основы языка SQL: учеб. пособие/. П. Моргунов, Е. В. Рогова, П. В. Лузанова. Санкт-Петербург : БХВ-Петербург, 2018. – 336 с.

**Рычка О.В., Самойленко Р.В. Разработка информационной системы тестирования и учёта успеваемости обучающихся.** В статье представлен краткий обзор современных информационных систем тестирования и контроля успеваемости обучающихся. Представлен функционал разрабатываемой системы. Рассмотрены аналоги разрабатываемой системы тестирования и успеваемости обучающихся, указаны преимущества и недостатки этих систем.

**Ключевые слова:** информационная система, тестирование, C#, Telegram-бот, Python. вопрос теста, аналоги разрабатываемой системы.

**Rychka Olga, Samoilenko Ruslan. Development of an information system for testing and accounting of students' progress.** The article presents a brief overview of modern information systems for testing and monitoring students' progress. The functionality of the system under development is presented. The analogues of the developed system of testing and student performance are considered, the advantages and disadvantages of these systems are indicated.

**Key words:** information system, testing, C#, Telegram-bot, Python. test question, analogues of the system being developed.

## Средства и методы разработки баз данных с использованием СУБД

Нанава А.О.\*<sup>1</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
sashananava1@mail.ru

*Нанава А.О. Средства и методы разработки баз данных с использованием СУБД. В данной статье кратко рассматривает что такое СУБД, его виды и типы архитектур баз данных. Рассмотрены методы и средства создания баз данных. Приведен анализ существующих СУБД.*

**Ключевые слова:** СУБД, РСУБД, PostgreSQL, базы данных, средства разработки БД.

### Введение

Базы данных это одно из самых распространенных направлений в программировании за 2022 год. Базы данных широко используются в множестве сфер от веб-технологий до компьютерных или мобильных игр. База данных – это система данных – баз данных, а также технических, программных, языковых и организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных. С каждым годом роль баз данных растет и появляется множество СУБД.

В данной статье в качестве СУБД будет рассмотрен PostgreSQL и его преимущества над другими. Будут рассмотрены основные методы и средства создания баз данных, так же, создана первая модель БД для работы склада.

### СУБД

СУБД – это система управления базами данных, она предназначена для создания, изменения, редактирования и организации баз данных. Базы данных используются практически везде. Огромное количество интернет ресурсов и компаний просто не способны функционировать без баз данных. СУБД были созданы для упрощения работы с базами данных[1].

Обычно основными элементами СУБД являются: ядро, процессор или компилятор и подсистемы времени исполнения.

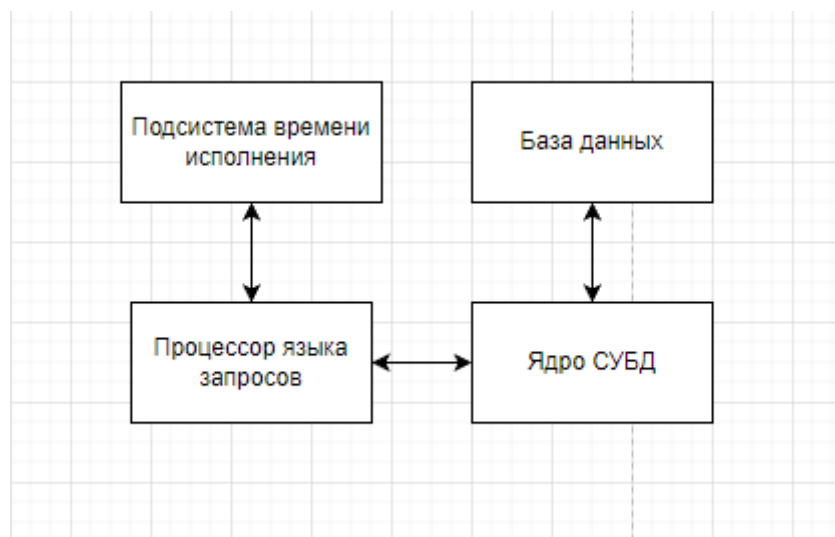


Рисунок 1 – Основные элементы СУБД

Основными задачами СУБД является:

- Управление базами данных их коррекция и программирование;



- Защита данных;
- Организация удобного хранилища данных;
- Управление запросами к базе данных;
- Резервное копирование данных и их возможно восстановление;

### **Виды СУБД**

СУБД подразделяются на разные виды по типу взаимодействия с клиентом и обработке запросов. Их можно разделить на:

- Клиент-серверные – в данном случае база данных расположена на каком-то удаленном сервере и доступ к ней организован с помощью запросов. Данный подход используется, например, в работе интернет-магазинов[2].
- Встраиваемые – СУБД, которые встроены в приложения и являются его частью. Они применяются для работы только в данном приложении.
- Файл-сервер – архитектурный подход, при котором база данных расположена на каком-то удаленном сервере, как и в случае с клиент-серверным БД, но при этом СУБД располагается на всех компьютерах-клиентах. Они получают данные с помощью запросов к БД через СУБД.

### **Архитектура БД**

Основной целью любой СУБД является возможность предложить обычному пользователю базы данных при этом, чтобы пользователю не приходилось задумываться над особенностями хранимых данных и управления ими. СУБД представляет абстрактную модель данных. Важнейшим моментом в проектировании БД является модель данных.

Модель данных – это группа понятий для определения данных, связей между данными и ограничений, которые накладываются на эти данные. Для грамотного проектирования базы данных – необходимо определить основные компоненты БД[3]:

- Структуру данных;
- Типы данных и операции над ними;
- Ограничения для корректной работы системы;

Можно определить несколько основных возможных моделей данных, которые подходят под 3-х уровневую архитектуру

- Иерархическая – древовидно представление данных в БД. Разделяет данные по уровням. Основой иерархической структуры является – узел дерева. Все элементы, расположенные выше – прародители, ниже – предки. Примером данной модели может выступать каталог файлов системы. Характерной чертой является то, что у элемента может быть только один родитель. Иерархическая модель является частным случаем сетевой модели, но при этом она появилась раньше. Основными преимуществами данной модели является её простота и малые затраты ресурсов(памяти). Главным недостатком подобной модели – это сложные логические связи.

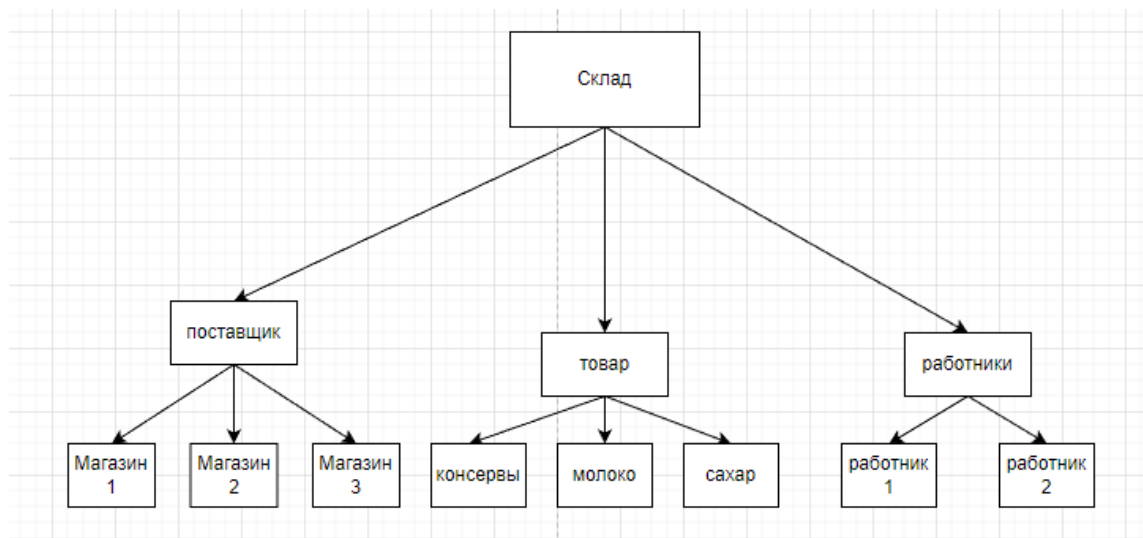


Рисунок 2 – Иерархическая модель

- Сетевая модель – это модель данных, которая состоит из записей, которые могут иметь явную ссылку на другую группу записей. Набор связей может быть произвольным, в отличие от иерархической модели. Проще говоря, это граф, в котором узлы могут иметь произвольное количество связей и нет ограничения на связь между уровнями, так как, все данные расположены на одном уровне. Основным плюсом является скорость, она в несколько раз быстрее чем иерархическая. К минусам относится сложная структура памяти и проблемы с обеспечением безопасности данных[4].

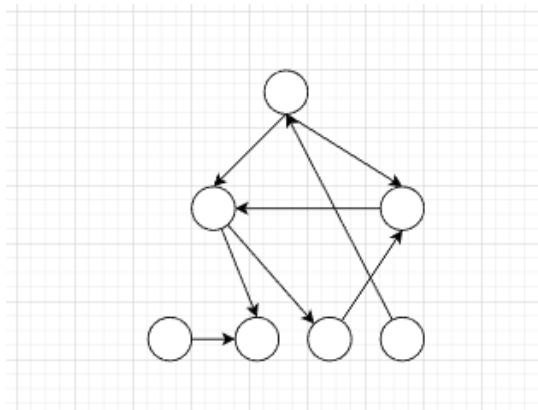


Рисунок 3 – Сетевая модель

- Реляционная – модель, которая предполагает наличие таблиц, которые состоят из строк и столбцов и имеют связи с другими таблицами. Свойства объектов определяются атрибутами полей. Из названия следует, что это конечный набор отношений между таблицами, каждое значение обладает ключом. Основные плюсы – это простота и гибкость. Минусами – являются отсутствие основных средств для определения отдельных записей таблицы, низкая скорость и использование больших объемов памяти.

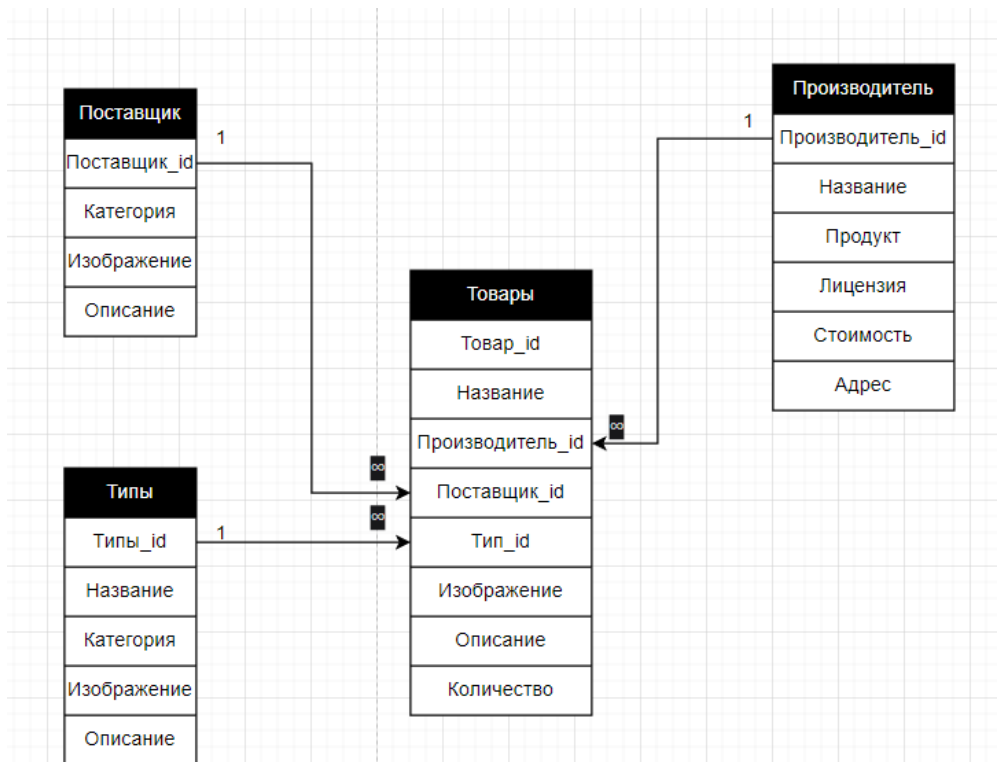


Рисунок 4 – Реляционная модель

### Анализ предметной области

В последнее время базы данных очень расширились и стали намного функциональней и производительней и это привело к тому, что люди начали их использовать во всевозможных сферах. Данная технология может быть

применена для работы высоконагруженных систем. Предприятие, которое хранит огромное количество товаров и их учет со временем становится не посильной задачей без использования компьютерных технологий. Базы данных позволят сэкономить множеству предприятий огромные деньги и упростит бюрократию.

### **Выбор СУБД**

Существует огромное количество СУБД, правильный её выбор может определить то на сколько будет быстрым и оптимальным решение. В статье были рассмотрены основные концепции и атрибуты СУБД. На основании выше изложенного можно выбрать оптимальную СУБД. Для решения поставленной задачи необходима система, которая будет проста в использовании и достаточно быстра. По данным критериям подходит реляционная модель построения баз данных.

По доступу к записям и схеме работы БД можно определить, что она будет клиент-серверная. По данным критериям можно выделить несколько СУБД - MS SQL, PostgreSQL и MySQL.

**MS SQL** – PCСУБД, созданная Microsoft, для работы с базами данных. Работает она по схеме – открытие сетевого порта, получение команд и вывод результата работы.

**Плюсы:** Масштабирование системы – может работать, как с большими данными на производительном компьютере, так и с маленькими объемами данных на слабых компьютерах; Динамическая блокировка транзакций; удобное хранение сложных данных.

**Минусы:** Может работать только под системами Microsoft; Высокая цена.

**PostgreSQL** – Это объектно-реляционная СУБД с открытым исходным кодом. Она очень популярная в русскоязычном сегменте. Является самой продвинутой базой данных и направленная на расширяемость системы[5].

**Плюсы:** Поддержка всех систем; Гибкость разработки; Полная SQL-совместимость; Объектно-ориентированность; открытый исходный код; Сильные инструменты разработки; Бесплатность.

**Минусы:** Маленькая популярность во всем мире и проблемы со скоростью

**MySQL** - одна из самых распространенных систем управления базами данных во всем мире. По принципу работы похожа на PostgreSQL.

**Плюсы:** Безопасность; Скорость; Распределенные операции; Кастомные решения; Бесплатность; Популярность.

**Минусы:** Недостаток функций; SQL-совместимость: Низкая скорость разработки.

### **NoSQL СУБД**

MongoDB – это очень хороший пример СУБД, которая является не реляционной (NoSQL). Огромным плюсом данной СУБД является то, что она способна хранить в себе данные без определенной структуры.

NoSQL подход стоит использовать в случаях когда:

- Необходимо хранить данные в разном формате: JSON, ключ-значение и другие;
- Записи не имеют общего формата
- Нет жестких требований к структуре
- Хорошая масштабируемость
- Огромные объемы данных

### **Жизненный цикл БД**

База данных – это достаточно обширный программный продукт и очевидно, что он нуждается в комплексном планировании и тестировании. Каждый программный продукт обладает жизненным циклом не зависимо от его сложности. При планировании базы данных большое внимание уделяется именно организации правильного жизненного цикла. У баз данных достаточно длинный жизненный цикл начиная с зарождением идеи и заканчивая заменой[6].

Жизненный цикл БД в более обширной форме подразделяется на несколько шагов:

- Планирование;
- Определение основных требований;
- Анализ предметной области;
- Проектирование базы данных;
- Разработка приложения
- Организация данных;
- Тестирование;
- Поддержка проекта;

## **Вывод**

Разработка баз данных – это очень комплексное и сложное занятие, но при этом не лишенное увлекательности и интересных моментов. Проектирование баз данных относится к той деятельности, от которой буквально зависит успех множества проектов. Хорошо построенная база данных сможет служить десятки лет без как-либо накладок. Базы данных используются в огромном количестве сфер без, которых сложно представить нашу жизнь – интернет, банки, магазины и многое другое.

В данной статье был проведен обзор существующих средств и методов создания баз данных. Проведен анализ средств разработки, а, так же, рассмотрены основные приемы и виды организации СУБД и баз данных. Был выполнен анализ предметной области.

## **Литература**

1. Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design - Michael J Hernandez – 673 с
2. Designing Data-Intensive Applications - Martin Kleppmann – 400 с
3. The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems - Ralph Kimball – 600 с
4. Refactoring Databases: Evolutionary Database Design – 370 с – Scott W Ambler, Pramod J Sadalage
5. PostgreSQL изнутри – 661 с - Е. Рогов
6. Mastering PostgreSQL 13 – 459 с - Ханс-Юрген Шёниг

**Нанава А.О. Разработка базы данных для работы склада с применением средств PostgreSQL.** В данной статье кратко рассматривает что такое СУБД, его виды и типы архитектур баз данных. Рассмотрены методы и средства создания баз данных. Приведен анализ существующих СУБД.

**Ключевые слова:** СУБД, РСУБД, PostgreSQL, базы данных, средства разработки БД.

**Nanova A.O. Tools and methods for developing databases using a DBMS.** This article briefly discusses what a DBMS is, its types and types of database architectures. The methods and means of creating databases are considered. The analysis of existing DBMS is given.

**Key words:** DBMS, RDBMS, PostgreSQL, databases, database development tools.

## **СЕКЦИЯ 3. «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ, АНАЛИЗ ДАННЫХ И РАСПОЗНАВАНИЕ ОБРАЗОВ»**

УДК 004.85

### **Модель машинного обучения для анализа тональности отзывов к кинофильмам**

**А.Д. Балабан<sup>\*1</sup>, О.И.Федяев<sup>\*2</sup>**

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
aleksey.balaban@mail.ru

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
fedyaev@donntu.ru, OrcID: 0000-0001-6822-7306

*Балабан А.Д., Федяев О.И. Модель машинного обучения для анализа тональности отзывов к кинофильмам. Статья посвящена автоматизации определения тональности отзывов к кинофильмам. Для решения поставленной задачи предложен метод, основанный на использовании нейронной сети LSTM. В рамках статьи показаны все этапы конвейера NLP: подготовка, обработка данных, построение, обучение и применение модели.*

**Ключевые слова:** Python, машинное обучение, Keras, LSTM, fastText, парсинг.

#### **Введение**

Анализ тональности – это область компьютерной лингвистики, которая занимается изучением мнений и эмоций в текстовых документах. Цель анализа тональности состоит в том, чтобы автоматически распознавать и классифицировать мнения, выраженные в тексте, для определения общего настроения.

Анализ тональности является одним из методов обработки естественного языка. Обработка естественного языка(NLP) – это область исследований в компьютерных науках и искусственном интеллекте (ИИ), занимающаяся работой с естественными языками. В частном случае, обработка включает в себя перевод естественного языка в данные (числа), с помощью которых компьютер может получить информацию об окружающем мире.

В статье демонстрируется автоматический анализ тональности рецензий к фильмам на сайте Кинопоиск с целью построения модели машинного обучения, которая затем сможет классифицировать отзывы как положительные, так и отрицательные. В рамках статьи показаны все этапы конвейера NLP(см.рис.1).

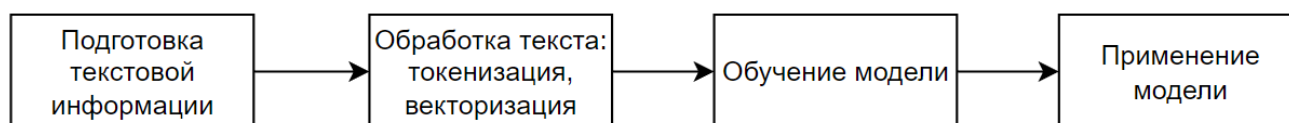


Рисунок 1 – Этапы конвейера NLP

Статья разбита на следующие разделы:

1. Подготовка данных: парсинг сайта Кинопоиск для получения положительных и отрицательных отзывов к фильмам, составление корпуса с отзывами на русском языке.
2. Обработка данных: токенизация, векторизация данных, разбиение данных на множества для обучения и проверки.
3. Построение модели машинного обучения. В этом случае используется нейронная сеть LSTM, с целью классификации отзывов к фильмам.
4. Проверка модели машинного обучения: использование различных метрик для того, чтобы убедиться, что модель хорошо обучена и может быть легко обобщена на другие наборы данных.

## Подготовка данных

Для решения задачи определения тональности отзывов к кинофильмам был создан корпус с отзывами на русском языке, взятых с сайта Кинопоиск. Для того чтобы автоматизировать этот процесс, на языке программирования Python с использованием библиотек BeautifulSoup и requests был написан веб парсер.

Функция `get_response_texts` очищает переданный ей список отзывов от html разметки(см.рис.2).

```
def get_response_texts(responses):  
    response_texts = []  
    for response in responses:  
        response = str(response)  
        response_text = BeautifulSoup(response, 'lxml').find('span', attrs={'class': '_reachbanner_',  
                                                                            'itemprop': 'reviewBody'})  
        title = BeautifulSoup(response, 'lxml').find('p',  
                                                    attrs={'class': 'sub_title'})  
        response_texts.append((title.text.replace('\xa0', ' ').replace('\x97', '-'),  
                               response_text.text.replace('\xa0', ' ').replace('\x97', '-')))  
    return response_texts
```

Рисунок 2 – Функция `get_response_texts`

Основная логика содержится в функции `parse_kinopoisk`(см.рис.3). Для каждого фильма из указанного списка составляется два списка – с хорошими и плохими отзывами. Затем эти списки сохраняются в файлах формата .json. Название файла – id фильма на сайте Кинопоиск.

```
def parse_kinopoisk(film_ids):  
    for film_id in film_ids:  
        page_idx = 1  
        good_response_texts = []  
        bad_response_texts = []  
        while True:  
            url = (f'https://www.kinopoisk.ru/film/{film_id}'  
                  f'/reviews/ord/date/status/all/perpage/{RESPONSE_PER_PAGE}/page/{page_idx}')  
            r = requests.get(url, headers=headers)  
            text = r.text.encode('utf-8').decode('utf-8')  
            soup = BeautifulSoup(text, 'lxml')  
            good_responses = soup.find_all('div', attrs={'class': 'response good'})  
            good_response_texts.extend(get_response_texts(good_responses))  
            bad_responses = soup.find_all('div', attrs={'class': 'response bad'})  
            bad_response_texts.extend(get_response_texts(bad_responses))  
            page_idx += 1  
            time.sleep(14 + random.random() * 6)  
            if not (len(good_responses) + len(bad_responses)):  
                break  
        json.dump(good_response_texts, open(fr'dataset\good\{film_id}.json', 'w', encoding='utf-8'), ensure_ascii=False)  
        json.dump(bad_response_texts, open(fr'dataset\bad\{film_id}.json', 'w', encoding='utf-8'), ensure_ascii=False)
```

Рисунок 3 – Функция `parse_kinopoisk`

## Обработка данных

Для загрузки данных была написана функция `load_dataset`(см.рис.4), которая возвращает список текстов всех отзывов. Каждый отзыв получает метку 0 (отрицательная тональность) или 1 (положительная тональность).

```
def load_dataset(path):  
    dataset = []  
    good_path = os.path.join(path, 'good')  
    bad_path = os.path.join(path, 'bad')  
    for filename in glob.glob(os.path.join(good_path, '*.json')):  
        with open(filename, 'r', encoding='utf-8') as f:  
            dataset.extend([(x[1], 1) for x in json.load(f)])  
    for filename in glob.glob(os.path.join(bad_path, '*.json')):  
        with open(filename, 'r', encoding='utf-8') as f:  
            dataset.extend([(x[1], 0) for x in json.load(f)])  
    np.random.shuffle(dataset)  
    return dataset  
  
dataset = np.array(load_dataset('dataset'))
```

Рисунок 4 – Функция load\_dataset

Следующий этап – токенизация и векторизация данных.

Токенизация – это процесс разделения предложений на слова-компоненты(токены). Векторизация(эмбединг) – процесс преобразования языковой сущности(слова, предложения, параграфа или целого текста) в набор чисел – числовой вектор.

Для токенизации текста был использован токенизатор RegexpTokenizer из библиотеки nltk, для векторизации – модель fastText[1]. Функция токенизации и векторизации представлена на рисунке 5.

```
def tokenize_and_vectorize(dataset):
    vectorized_data = []
    tokenizer = RegexpTokenizer(r'\w+|[\^\w\s]+' )
    for sample in dataset:
        tokens = tokenizer.tokenize(sample.lower())
        sample_vec = []
        for token in tokens:
            try:
                sample_vec.append(ft[token])
            except KeyError:
                pass
        vectorized_data.append(sample_vec)
    return vectorized_data
```

Рисунок 5 – Функция токенизации и векторизации

При помощи функции unique из библиотеки NumPy можно посмотреть количество отзывов каждого класса(см.рис.6).

```
for label, count in zip(*np.unique(dataset[:, 1], return_counts=True)):
    print(f'{"Положительных" if label == "1" else "Отрицательных"} отзывов: {count}.')
```

Отрицательных отзывов: 3494.  
Положительных отзывов: 12066.

Рисунок 6 – Количество положительных и отрицательных отзывов

Доли объектов разных классов существенно различаются – классы являются несбалансированными.

Далее подготовленные данные необходимо разбить на тренировочный и тестовый наборы в соотношении 80/20. Для этого можно воспользоваться функцией train\_test\_split из библиотеки sklearn. Так как присутствует несбалансированность классов, необходимо воспользоваться параметром stratify(см.рис.7).

```
x_train, x_test, y_train, y_test = train_test_split(vectorized_data, dataset[:, 1], test_size=.2,
                                                    stratify=dataset[:, 1])
```

Рисунок 7 – Разбиение данных на тренировочный и тестовый наборы

Затем задаётся большинство гиперпараметров сети. Переменная max\_len содержит максимально возможную длину отзыва. Поскольку все входные сигналы нейронной сети должны быть одинаковой длины, необходимо усечь все примеры длиннее 550(немного больше, чем средняя длина отзыва) токенов и дополнить нулями более короткие до 550 токенов(см.рис.8).

```
print(f'Средняя длина отзыва: {np.average(list(map(len, vectorized_data)):.1f}.')
```

Средняя длина отзыва: 491.8.

```
max_len = 550
batch_size = 32
embedding_dims = 100
epochs = 24
```

```
def pad_trunc(data, max_len, embedding_dims):
    new_data = []
    for sample in data:
        if len(sample) >= max_len:
            new_data.append(sample[:max_len])
        else:
            vector = np.zeros((max_len, embedding_dims))
            vector[max_len - len(sample):] = sample
            new_data.append(vector)
    return np.array(new_data)
```

```
x_train = pad_trunc(x_train, max_len, embedding_dims)
x_test = pad_trunc(x_test, max_len, embedding_dims)
```

Рисунок 8 – Объявление гиперпараметров, усечение данных

На этом обработка данных завершена.

### Построение нейросетевой модели

Создание многослойной нейронной сети в пакете Keras происходит при помощи класса Sequential.

В начале необходимо добавить несколько слоёв LSTM. Долгая краткосрочная память (Long short-term memory; LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям[2, 3].

LSTM вводит понятие состояния для каждого из слоёв рекуррентной сети. Состояние играет роль памяти. В LSTM правила, определяющие хранение информации в состоянии (памяти)(см.рис.9), – это сама обученная нейронная сеть. Её можно обучить тому, что надо запоминать, в то время как остальная рекуррентная нейронная сеть учится предсказывать целевую метку. С появлением памяти и состояния можно выучивать зависимости, простирающиеся на весь пример данных, а не только на 1–2 токена. Благодаря таким «долгим» зависимостям можно выйти за рамки отдельных слов и больше узнать о языке.

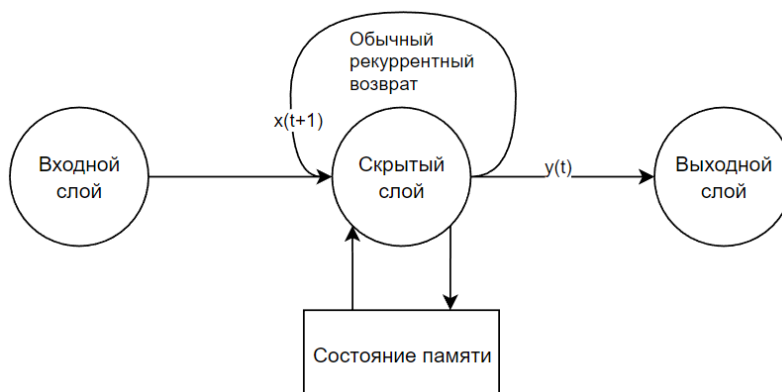


Рисунок 9 – Общая структура рекуррентной нейронной сети LSTM

В задачах, где доступны все временные шаги входной последовательности, можно использовать двунаправленные LSTM. Основная идея состоит в размещении двух LSTM рядом друг с другом и передаче одного и того же входного сигнала в одну обычным образом, а во вторую – в обратном порядке. Это может обеспечить дополнительный контекст для сети и привести к более быстрому и даже более полному изучению проблемы. В Keras есть слой-адаптер Bidirectional для автоматического поворота нужных входных и выходных сигналов в другую сторону и автоматического создания двунаправленной LSTM.

Для предотвращения переобучения необходимо добавить слои дропаута. Дропаут – это специальная методика предотвращения переобучения в нейронных сетях. Идея дропаута заключается в «выключении» определенной части передаваемого на следующий слой входного сигнала, выбираемой при каждом проходе случайным образом. В результате модель скорее усвоит не особенности именно тренировочного набора данных (и переобучится), а детальные представления закономерностей данных и, следовательно, окажется способна на обобщения и точные предсказания на основе совершенно новых для неё данных[4].



В конце необходимо добавить классификатор. В данном случае есть два класса: «Да – положительная тональность – 1» и «Нет – отрицательная тональность – 0». Поэтому необходимо добавить слой из одного нейрона (Dense(1)) и сигма-функцию активации. Слой Dense требует в качестве входного сигнала вектор из n элементов. Однако на выходе LSTM данные представляют собой тензор второго ранга. Для преобразования тензора второго ранга в вектор необходимо воспользоваться вспомогательным слоем Flatten().

Модель полностью описана. Осталось её скомпилировать(см.рис.10).

```
model = Sequential()
model.add(Bidirectional(LSTM(192, return_sequences=True), input_shape=(max_len, embedding_dims)))
model.add(Dropout(.2))
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(Dropout(.2))
model.add(Bidirectional(LSTM(128)))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile('rmsprop', 'binary_crossentropy', metrics=['accuracy', tf.keras.metrics.AUC(name='auc')])
```

Рисунок 10 – Компиляция модели

Параметр loss определяет, какую функцию сеть будет пытаться минимизировать. В данном случае используется функция потерь 'binary\_crossentropy', так как есть только два класса.

Параметр optimizer может принимать значение, соответствующее любой из перечня стратегий оптимизации сети во время обучения, например, RMSProp.

Параметр metrics позволяет отслеживать выбранные метрики во время обучения. Так как классы несбалансированны, была добавлена метрика AUC.

Общую информацию о модели можно посмотреть при помощи метода summary()(см.рис.11).

model.summary()		
Model: "sequential_4"		
Layer (type)	Output Shape	Param #
=====		
bidirectional_12 (Bidirectional)	(None, 550, 384)	450048
dropout_8 (Dropout)	(None, 550, 384)	0
bidirectional_13 (Bidirectional)	(None, 550, 256)	525312
dropout_9 (Dropout)	(None, 550, 256)	0
bidirectional_14 (Bidirectional)	(None, 256)	394240
flatten_4 (Flatten)	(None, 256)	0
dense_4 (Dense)	(None, 1)	257
=====		
Total params: 1,369,857		
Trainable params: 1,369,857		
Non-trainable params: 0		

Рисунок 11 – Общая информация о модели

Обучение модели осуществляется при помощи метода fit(). В зависимости от аппаратного обеспечения и размеров модели и данных этот процесс может занимать от пары секунд до нескольких месяцев. В большинстве случаев использование GPU может сильно сократить время обучения. Изменение метрики AUC во время обучения представлено на рисунке 12.

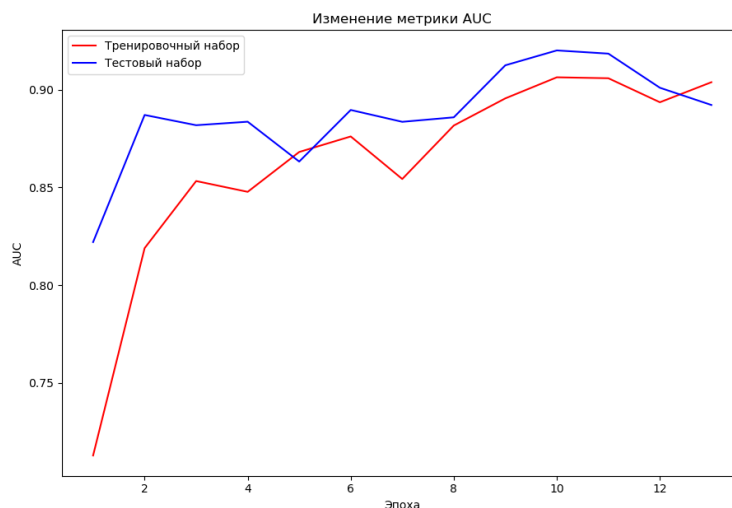


Рисунок 12 – Изменение метрики AUC во время обучения

Последний шаг – сохранение модели. Чтобы каждый раз не обучать модель заново, её веса можно сохранить в файл с расширением .h5. Структуру модели можно сохранить в формате .json.

### Проверка модели

При обучении модели после каждой эпохи выводятся значения метрик относительно тестовых данных. Наилучшие показатели разработанной модели: точность 87% и AUC 0.91.

После обучения модели можно передать ей новые отзывы для определения их тональности. Примеры прогнозов для отзывов с явно отрицательной и положительной тональностью приведены на рисунке 13. Прогноз модели: первый отзыв – отрицательный, второй – положительный.

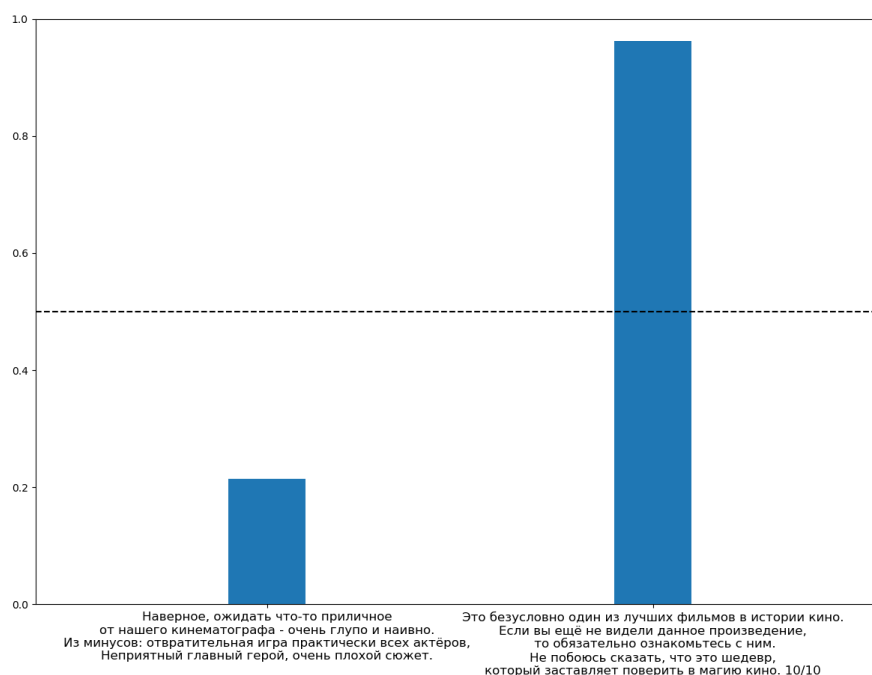


Рисунок 13 – Результаты определения тональности отзывов при помощи модели

### Заключение

В статье рассмотрена задача автоматической классификации текстов по тональности, которая относится к актуальным задачам интеллектуального анализа текстовых данных. Для решения поставленной задачи предложен метод, основанный на использовании нейронной сети LSTM. Метод реализован в виде программной модели на языке Python. Проведена оценка качества разработанной модели. Следует отметить, что задача определения тональности является весьма сложной, кроме того, её решение зависит от естественного языка, на котором написаны сообщения.

## **Литература**

1. fastText [Электронный ресурс]. – Режим доступа: <https://fasttext.cc/>
2. Long Short-Term Memory [Электронный ресурс]. – Режим доступа: <https://www.gwern.net/docs/ai/nn/rnn/1997-hochreiter.pdf>
3. Learning Precise Timing with LSTM Recurrent Networks [Электронный ресурс]. – Режим доступа: <http://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf>
4. Обработка естественного языка в действии / Хобсон Лейн, Ханнес Хапке, Коул Ховард - 1-е изд. – СПб.: Питер, 2020. – 576с.

**Балабан А.Д., Федяев О.И. Модель машинного обучения для анализа тональности отзывов к кинофильмам.** Статья посвящена автоматизации определения тональности отзывов к кинофильмам. Для решения поставленной задачи предложен метод, основанный на использовании нейронной сети LSTM. В рамках статьи показаны все этапы конвейера NLP: подготовка, обработка данных, построение, обучение и применение модели.

**Ключевые слова:** Python, машинное обучение, Keras, LSTM, fastText, парсинг.

**Balaban Aleksey, Fedyaev Oleg. Machine Learning Model for Sentiment Analysis of Movie Reviews.** The article is devoted to the automatic movie reviews sentiment analysis. To solve the problem, a method based on the use of the LSTM neural network is proposed. The article shows all stages of the NLP pipeline: preparation, data processing, building, training and application of the model.

**Key words:** Python, machine learning, Keras, LSTM, fastText, parsing.

## Применение программных средств для обработки текстов на естественном языке

Таций Е. В. <sup>\*1</sup>, Анохина И. Ю. <sup>\*2</sup>

<sup>\*1</sup> бакалавр, Донецкий национальный технический университет, TatsiyEV@yandex.ru,

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
OrcID: 0000-0003-1073-6333, SPIN-код: 3555-6335  
ingatula@mail.ru

*Таций Е.В. Анохина И.Ю. Применение программных средств для обработки текстов на естественном языке. В статье описаны подходы представления смысла слова в цифровом виде. Проведён сравнительный анализ библиотек языка программирования python для получения вектора из слова, по которому можно оценить семантическую близость слов.*

**Ключевые слова:** NLP, python, word embedding, word2vec, обработка естественного языка

### Введение

Информационно-коммуникационные технологии и области науки, связанные с ними, являются наиболее быстро развивающимися сферами жизни современного человека. Они же стимулируют появление новых наук. Например, на стыке кибернетики и лингвистики появилась новая наука, вначале получившая название – математическая лингвистика, затем название было заменено на «структурная лингвистика». Сейчас это компьютерная лингвистика.

Для лингвистов современные компьютеры – мощное и точное средство автоматизации обработки текстов, позволяющее осуществлять статистическую обработку текстов, автоматизировать создание, использование и редактировании словарей, лексических карточек и пр.

Компьютерную лингвистику определяют как направление, ориентированное на использование компьютерных инструментов – программ, компьютерных технологий организации и обработки данных – для моделирования функционирования языка в тех или иных условиях, ситуациях, проблемных сферах и т.д. [1].

Выделяют следующие основные направления компьютерной лингвистики: информационный поиск, извлечение информации, машинный перевод, корпусная лингвистика, построение экспертных и вопросно-ответных систем, распознавание и синтез речи, обучение языку с помощью компьютерных программ, создание электронных словарей, тезаурусов и пр.

В 1954г. корпорацией IBM совместно с Джорджаунским университетом была проведена демонстрация возможностей машинного перевода, в ходе которой был осуществлен полностью автоматический перевод более шестидесяти русских предложений на английский язык.

Бурное развитие сети Интернет привело к резкому росту объемов текстовой информации в электронном виде, что обусловило новые задачи информационного поиска. Если рассматривать информацию, выдаваемую при запросах на поиск, как огромный поток неструктурированных данных, то становится понятным, насколько важно развить оптимальные концепции работы с данными такого типа.

Использование статистических методов, например, разбиение текста на кластеры, поиск общих фрагментов, вычисление определенных показателей, дающих информацию о тексте, позволило решить ряд задач. В настоящий момент исследования все больше фокусируются на неконтролируемых и полууправляемых алгоритмах обучения. Такие алгоритмы могут учиться на данных, которые не были вручную аннотированы желаемыми ответами, или использовать комбинацию аннотированных и неаннотированных данных.

С развитием машинного обучения и, в частности, нейронных сетей, темпы исследований в области обработки естественного языка (или NLP от англ. Natural Language Processing) значительно увеличились. Появились общедоступные библиотеки, позволяющие использовать NLP в различных задачах, в том числе для оптимизации бизнес-процессов. NLP используется для создания систем распознавания речи, обобщения документов, машинного перевода, выявления спама, ответов на вопросы. Как один из наиболее общеизвестных примеров использования NLP – смартфоны с распознаванием речи.

### Виды обработки естественного языка

К основным этапам обработки текста относятся:

- графематический анализ – первичный, предварительный анализ текста (разделение текста на слова, выделение фамилий, инициалов, электронных адресов, абзацев, заголовков, примечаний и пр. );
- морфологический анализ – определение нормальной формы, т.е. анализ слова в именительном падеже и единственном числе (в качестве единицы выступает слово);
- фрагментационный анализ – выделение синтаксических фрагментов (в качестве единицы рассматриваются фразы, части предложения), один из главных этапов машинного перевода;
- синтаксический анализ осуществляется на уровне предложений и устанавливает связи между элементами предложения;
- семантический анализ – завершающий этап обработки текста, заключающийся в смысловом анализе текста.

Графематический анализ включает лексический анализ. Его задача заключается в выделении в тексте слов, чисел, знаков препинания, формул. Выделенные части называют токенами. Токенизация – один из главных шагов в конвейере НЛП. Токенизатор разбивает неструктурированные данные и текст на естественном (человеческом) языке на фрагменты информации, которые можно рассматривать как отдельные элементы. Вхождения токена в документе можно использовать непосредственно как вектор, представляющий этот документ. Это преобразует неструктурированный текстовый документ в числовую структуру данных, пригодную для машинного обучения.

При разбиении на слова, ориентиром могут быть пробелы, знаки препинания. В некоторых случаях таких условий недостаточно и для токенизации применяется сложный набор правил, уникальных для каждого языка [2].

Морфологический анализ, как правило, состоит из четырех разделов:

1. стемматизация (от англ. *stemming* – вытекающий, проистекающий) – процесс нахождения основы слова для заданного исходного слова. Например, для словоформ красный, краснеть, краснота, красно в результате стемматизации будет получена псевдооснова красн;
2. лемматизация – словоформа преобразуется к лемме, т.е. рассматривается слово в единственном числе, именительном падеже, т.е. так, как оно заносится в словарь;
3. приписывание словоформе множества наборов граммем. Граммема (грамматическая характеристика) – морфологический описатель, например, мр, жр, ср - мужской, женский, средний род; ед, мн - единственное, множественное число; нст, прш, буд - настоящее, прошедшее, будущее время;
4. получение парадигм – образование всех форм слова, например, это может быть существительное во всех падежах и числах (стул, стула, стулу, стулом, стуле, стулья, стульев, стульям, стульях, стульями) [3].

### **Компьютерное представление слова**

Слово, как изобретение, исключительно человеческое, и предназначенное для взаимодействия между людьми, не может быть достоверно представлено в цифровом устройстве. В процессе многолетних исследований в области NLP, было предложено несколько подходов к представлению смысла слов в цифровом виде, т.е. ставится задача преобразования категориальных значений в числовые.

**Label encoding.** Кодирование меток.

Суть подхода — закодировать слова цифрами по порядку следования в словаре. Например, рассматриваем города Москва, Архангельск, Тула. Архангельск будет закодирован нулем, Москва – 1, Тула -2. Но возникает проблема с использованием чисел, т.к. вводится отношение/сравнение между объектами. Имея число, можно подумать, что Архангельск имеет более высокий приоритет, чем Москва и Тула. Получается, что данные имеют какую-то иерархию/порядок.

Кроме этого при добавлении слова придется частично изменять нумерацию слов. Например, при добавлении в рассматриваемое множество города Орел, нумерация Тулы изменится с цифры 2 на 3.

Еще одной проблемой является несоблюдение так называемой семантической близости. Например, слова собака и щенок будут стоять достаточно далеко друг от друга.

**Embeddings.** Вложения.

Чтобы получить возможность представить семантическую близость, было предложено использовать *embedding*, то есть сопоставить слову вектор, отображающий его значение с учетом смысла. Это достигается путем токенизации каждого слова в последовательности (или предложении) и преобразования их в векторное пространство. Встраивание слов направлено на то, чтобы зафиксировать семантическое значение слов в последовательности текста. При этом присваиваются аналогичные числовые представления словам, которые имеют схожие значения.

Благодаря такому представлению можно сравнивать тексты с учетом различных метрик, как это принято, например, при кластерном анализе, в котором применяются различные виды метрик: евклидово расстояние, метод ближайшего соседа, манхэттенское расстояние, расстояние Чебышёва и пр.

**One-hot encoding.** Метод горячего кодирования.

Метод преобразования категориальных переменных в несколько двоичных столбцов, где 1 указывает на наличие этой строки, принадлежащей этой категории. Самый простой способ получить вектор из слова. Это далеко не лучший выбор для кодирования категориальных переменных с точки зрения машинного обучения. Метод имеет существенный недостаток. Например, если бы мы ввели столбец, кодирующий породы собак, то схема прямого кодирования привела бы к появлению такого количества дополнительных измерений, сколько пород мы собираемся кодировать. Кроме этого метод не обладает свойствами семантической близости.

**Bag of words.** Мешок слов.

Метод моделирования текста, основанный на обработке естественного, разговорного языка. В методе оценивается количество слов и игнорируются грамматические детали и порядок слов. Он называется «мешком» слов, потому что любая информация о порядке или структуре слов в документе отбрасывается. Модель касается только того, встречаются ли известные слова в документе, а не где и как именно в документе.

Пример. Пусть даны два предложения.

Sentence 1: Welcome to Great Learning, Now start learning.

Sentence 2: Learning is a good practice.

Метод заключается в подсчете присутствия каждого слова и отметке 0 в случае его отсутствия.

Результат будет выглядеть следующим образом:

Предложение	Welcome	to	Great	Learning	Now	,	start	learning	is	a	good	practice
Предложение 1	1	1	1	1	1	1	1	1	0	0	0	0
Предложение 2	0	0	0	0	0	0	0	1	1	1	1	1

В предложенном примере используются все слова из словаря для формирования вектора. Это не является ни практичным, ни лучшим способом реализации модели. На практике для формирования вектора используются только несколько слов из словаря, причем предпочтительно использовать самые распространенные слова.

**TF-IDF.** Частотно-обратная частота документа

**TF-IDF** – **T**erm **F**requency – **I**nverse **D**ocument **F**requency, что является модификацией матрицы term-document. Эта мера используется для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции.

TF (term frequency – частота слова) – отношение числа вхождений слова  $n(i, j)$  к общему числу слов документа, характеризует важность слова в пределах рассматриваемого документа.

$$TF(i, j) = \frac{n(i, j)}{\sum n(i, j)}$$

где  $\sum n(i, j)$  – число слов в документе.

IDF (inverse document frequency – обратная частота документа) – инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

$$IDF(t, D) = \log \frac{|D|}{DF(t, D)}$$

где D – число документов в коллекции;

DF(t, D) – число документов из коллекции DD, в которых встречается t.

Вес слова в пределах всего документа определяется

$$Vt, d = F(t, d) \times IDF(t, D)$$

Большой вес в TF-IDF получают слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

Пример. Документ содержит 100 слов. Слово «яблоко» встречается в нем пять раз. Частота слова «яблоко» в документе  $TF = 5/100 = 0.05$ .

Всего в коллекции 100000 документов. В 1000 документов встречается слово «яблоко». Обратная частота документа  $IDF = \log 100 = 2$ . Вес слова «яблоко» в выбранном документе  $Vt, d = TF \cdot IDF = 0,05 \cdot 2 = 0,1$ .

Функция TF-IDF с одной стороны демонстрирует, что чем чаще встречается в документе слово, тем лучше оно отражает содержание текста, а с другой стороны, чем в большем количестве документов оно встречается, тем менее значимым является для классификации[3–4].

**word2vec.**

Word2Vec — это недавний прорыв в мире НЛП. Томас Миколов, чешский ученый-компьютерщик из Чешского института информатики, робототехники и кибернетики, был одним из ведущих участников исследований и внедрения word2vec. Метод основан на гипотезе, что слова, которые встречаются в одинаковых

окружениях, имеют близкие значения. Word2Vec имеет множество приложений, таких как сходство текста, системы рекомендаций, анализ настроений и т. д.

Эффективность Word2Vec обусловлена его способностью группировать векторы похожих слов. Имея достаточно большой набор данных, Word2Vec может сделать точную оценку значения слова на основе его появления в тексте. Эти оценки дают ассоциации слов с другими словами в тексте. Например, такие слова, как «Король» и «Королева», будут похожи друг на друга. При выполнении алгебраических операций над вложениями слов можно найти близкое приближение сходства слов.

$$\text{Король} - \text{Мужчина} + \text{Женщина} = \text{Королева}$$

В модели предсказывается вероятность слова по его окружению (контексту). Иными словами вероятность слова близка к вероятности встретить это слово в этом окружении в реальном тексте.

$$P(\omega_0|\omega_c) = \frac{e^{s(\omega_0, \omega_c)}}{\sum_{\omega_1 \in V} e^{s(\omega_1, \omega_c)}}$$

где

$\omega_0$  — вектор целевого слова;

$\omega_c$  — это некоторый вектор контекста, вычисленный например, путем усреднения из векторов окружающих нужное слово других слов;

$s(\omega_1, \omega_2)$  — функция, которая двум векторам сопоставляет одно число [5-6].

При реализации алгоритма рассматривают последовательно  $(2k+1)$  слов, причем слово в центре является тем словом, которое должно быть предсказано. Окружающие слова являются контекстом длиной  $k$  с каждой стороны. Каждому слову в модели сопоставляется уникальный вектор, который меняется в процессе обучения нашей модели. Этот подход называется CBOW — Continuous Bag Of Words, continuous.

Миколовым был предложен другой подход — skip-gram (словосочетание с пропуском). Здесь текущее слово рассматривается в качестве входного данного и ставится задача предсказания слова до и после этого входного слова

### Сравнительный анализ библиотек

Для языка программирования python существует две модели показывающие лучшие результаты в NLP для русского языка с возможностью запуска на CPU: `naves` и `rusvectors`.

`Rusvectors` - набор из десятков моделей, обученных на открытых наборах данных. Самая лёгкая модель имеет размер 300 Мб.

`Naves` - 2 модели, обученные подражать гораздо большей и требовательной модели. Обучены на художественной литературе и новостных текстах. Размер модели, обученной на новостях - 25 Мб. Такой размер существенно сокращает время запуска алгоритма и позволяет использовать больше параллельно работающих приложений.

Для сравнения библиотек выполним анализ семантической близости на одинаковых множествах слов: зима, снег, солнце, мороз, море, чайка, океан, метель. Занесём полученные значения в таблицы, см. табл. 1-2.

Таблица 1 – Расстояния между векторами слов, рассчитанные библиотекой `naves`

	зима	снег	солнце	мороз	море	чайка	океан	метель
зима	-	0,38885	0,31871	0,28188	0,09290	0,02491	0,11922	0,26932
снег	0,38885	-	0,34271	0,40338	0,19340	0,00168	0,12911	0,61071
солнце	0,31871	0,34271	-	0,21426	0,30926	0,05040	0,16462	0,15493
мороз	0,28188	0,40338	0,21426	-	0,03080	0,16149	0,13404	0,36865
море	0,09290	0,19340	0,30926	0,03080	-	0,04871	0,39010	0,06797
чайка	0,02491	-0,00168	0,05040	0,16149	0,04871	-	0,10446	0,04035
океан	0,11922	0,12911	0,16462	0,13404	0,39010	0,104462	-	0,06281
метель	0,26932	0,610712	0,154931	0,368650	0,067975	0,040359	0,062817	-

Таблица 2 – Расстояния между векторами слов, рассчитанные библиотекой `rusvectors`

	зима	снег	солнце	мороз	море	чайка	океан	метель
--	------	------	--------	-------	------	-------	-------	--------

зима	-	0,56060	0,35628	0,63425	0,34983	0,23412	0,26963	0,4979
снег	0,5606	-	0,52109	0,62991	0,38580	0,26231	0,34403	0,6610
солнце	0,3562	0,52109	-	0,47778	0,44771	0,31987	0,39144	0,4474
мороз	0,6342	0,62991	0,47778	-	0,34325	0,18039	0,28474	0,6569
море	0,3498	0,38580	0,44771	0,34325	-	0,32724	0,72195	0,3786
чайка	0,2341	0,26231	0,31987	0,18039	0,32724	-	0,26055	0,2333
океан	0,2696	0,34403	0,39144	0,28474	0,72195	0,26055	-	0,3232
метель	0,4979	0,66103	0,44745	0,65691	0,37861	0,23333	0,32326	-

### **Вывод**

В результатах работы обеих моделей можно заметить определённое сходство. Все пары слов, имеющие прямую связь друг с другом, в реальном мире имеют высокие значения. Порядок следования наиболее похожих слов совпадает в обеих моделях.

Рассматривая менее похожие слова, заметны различия в их порядке. Так, например, при использовании библиотеки Navес можно отметить, что со словом «чайка» менее всего связано слово «снег», а Rusvectors - «мороз». Однако на практике слабо связанные слова не используются и этими различиями библиотек можно пренебречь.

При выполнении сравнительного анализа, опытным путём обнаружено, что на загрузку модели и вычисления Navес тратит в 8 раз меньше времени, чем при использовании библиотеки Rusvectors.

Подытоживая результаты, можно сделать вывод, что для задач обработки естественного языка обе модели хорошо себя проявляют. Но учитывая скорость загрузки моделей, для использования можно рекомендовать модель Navес.

### **Литература**

1. Боярский К. К. Введение в компьютерную лингвистику. Учебное пособие. – СПб: НИУ ИТМО, 2013. – 72с.
2. Razdel: Rule-based token, sentence segmentation for Russian language [Электронный ресурс] – Режим доступа: <https://github.com/natasha/razdel#evaluation>.
3. Батура, Т. В. Математическая лингвистика и автоматическая обработка текстов: учеб. пособие / Т. В. Батура ; Новосиб. гос. ун-т. – Новосибирск : РИЦ НГУ, 2016. – 166 с.
4. Great Learning. An Introduction to Bag of Words (BoW) | What is Bag of Words? [Электронный ресурс] – Режим доступа: <https://www.mygreatlearning.com/blog/bag-of-words/#ed6>.
5. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781,
6. Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119, 2013.

**Тацкий Е.В. Анохина И.Ю. Применение программных средств для обработки текстов на естественном языке. В статье описаны подходы представления смысла слова в цифровом виде. Проведён сравнительный анализ библиотек языка программирования python для получения вектора из слова, по которому можно оценить семантическую близость слов.**

**Ключевые слова:** NLP, python, word embedding, word2vec, обработка естественного языка

**Tatsiy E.V. Anokhina I.Y. Application of software tools for text processing in natural language. The article describes approaches to the representation of the meaning of the word in digital form. A comparative analysis of the libraries of the python programming language was carried out to obtain a vector from a word, which can be used to evaluate the semantic proximity of words.**

**Keywords:** NLP, python, word embedding, word2vec, natural language processing



## Анализ перспектив применения технологии RFID в торговле

А.В. Боднар<sup>\*1</sup>, Е.С. Бондаренко<sup>\*2</sup>

<sup>\*1</sup> к.э.н, доцент, Донецкий национальный технический университет,  
linabykova13@ya.ru, Orcid: 0000-0002-1800-0178, SPIN-код: 5068-7409

<sup>\*2</sup> магистрант, Донецкий национальный технический университет,  
kate.bond777@gmail.com

*Боднар А.В., Бондаренко Е.С. Анализ перспектив применения технологии RFID в торговле. В статье исследована актуальность и значимость использования технологии RFID в секторе торговли. Проанализированы преимущества применения и проблемы внедрения технологии RFID в секторе торговли. Выявлены способы преодоления препятствий на пути внедрения RFID в розничной торговле. Выделены направления развития технологии RFID для повышения эффективности продаж.*

**Ключевые слова:** технология RFID, внедрение, розничная торговля, управление запасами, системы управления, цепочка поставок.

### Введение

Технология RFID (Radio Frequency Identification) имеет несколько применений для оптимизации процессов и улучшения цепочки поставок во всех секторах и отраслях промышленности. Сектор розничной торговли был одним из первых секторов, увидевших потенциал технологии, которая особенно эффективна в деятельности, связанной с перемещением большого количества товаров каждый короткий промежуток времени, такой как массовое потребление. Идентификация и локализация каждого перемещения товара за считанные секунды облегчает и ускоряет управление запасами в магазинах, поступление и выбытие товаров на заводах, и снабжение в распределительных центрах. Но в настоящее время технология RFID повышает свою актуальность в экономике замкнутого цикла. Это чрезвычайно полезная технология для брендов, которые разрабатывают планы устойчивого развития и хотят быть прозрачными со своими клиентами.

Цель работы – исследование перспектив внедрения технологии RFID в торговлю. Для достижения цели поставлены следующие задачи:

- исследование актуальности и значимости использования технологии RFID;
- анализ преимущества применения и проблем внедрения технологии RFID в секторе торговли;
- выявление способов преодоления препятствий на пути внедрения RFID в розничной торговле;
- выделение направлений развития технологии RFID для повышения эффективности продаж.

### Популярность RFID

Первый патент на радиочастотную идентификацию, более известный как RFID-системы, был подан в 1983 году. Тем не менее, эти мощные маленькие метки по-прежнему недооцениваются и находятся на подъеме во всей мировой экономике. Ожидается, что к 2025 году объем мирового рынка RFID достигнет 40,5 миллиардов долларов, а совокупный годовой темп роста составит около 14,7% [1].

Внедрение приносит рентабельность инвестиций и поддерживает многоканальные возможности. По общему внедрению Северная Америка продолжает оставаться лидером. И когда мы смотрим на полное внедрение, во всех трех регионах Северной Америки, Европы и Азиатско-Тихоокеанского региона наблюдается огромный рост RFID с 2018 года (рис. 1, 2).



Рисунок 1 – Внедрение RFID в Северной Америке и Азиатско-Тихоокеанским регионом [1]



Рисунок 2 – Внедрение RFID в Европе [1]

Внедрение RFID увеличилось во всем мире с 2018 года, при этом в Европе наблюдается десятикратный рост. Это говорит о том, что технология RFID имеет большой потенциал и уже сейчас быстрыми темпами внедряется в разные сферы промышленности и экономики [1].

### **Преимущества RFID**

Среди множества своих применений RFID сыграл значительную роль в развитии и расширении сектора розничной торговли модной одеждой, особенно в течение последнего десятилетия, учитывая важность логистики в бизнес-модели. Технология RFID представляется как решение для решения новых задач, в данном случае связанных с устойчивостью и прозрачностью, поскольку она обеспечивает полную прослеживаемость в цепочке поставок, предоставляя всю информацию в режиме реального времени. Начиная с электронной коммерции и заканчивая физическими магазинами, они внедрились эту технологию в сотрудничестве с производителями текстиля и компаниями. Многие компании уже внедряют эту технологию с самого начала производства продукции.

Поскольку RFID обеспечивает точность в реальном времени почти на 100%, неудивительно, что эта технология используется для улучшения качества обслуживания клиентов при предоставлении различных услуг. Поэтому технология RFID является эффективной и имеет преимущества в использовании ее в сфере торговли.

RFID также развивается в управлении цепочками поставок и логистике.

Управление цепочками поставок и логистика – наиболее плодородная область с точки зрения применения RFID [2]. Рассмотрим все элементы управления цепочками поставок один за другим.

### **Управление запасами**

Управление запасами является важным элементом управления цепочками поставок. Она включает в себя различные аспекты, такие как мониторинг, администрирование, контроль, хранение и, в конечном счете, использование материалов для продажи продукта [3]. Неточности в управлении запасами неизбежны и широко распространены во многих отраслях промышленности. Неточность – это не что иное, как несоответствие между инвентарными записями и фактическим количеством товара, доступного для продажи. Технология RFID может обеспечить многочисленные преимущества для улучшения системы управления запасами. RFID-метки обладают способностью считывать содержимое насквозь. Более того, человек может сканировать несколько предметов одновременно. Эти свойства RFID помогают ускорить процесс управления запасами и сокращают количество человеческих ошибок, тем самым обеспечивая высокую точность учета запасов.

## **Управление складом**

RFID появилась как технология, поддерживающая системы управления складом для упрощения цепочки поставок и повышения разборчивости продукта. С помощью технологии RFID вы можете автоматизировать важные задачи, которые выполняются в процессе получения и отгрузки. RFID также повышает эффективность идентификации и валидации наряду с уменьшением человеческих ошибок. RFID обеспечивает мгновенную идентификацию товаров и больший контроль над товарами на складе. Благодаря этому цепочки поставок теперь имеют больший информационный поток по сравнению с материальным потоком. Это еще больше снижает стоимость валюты.

## **Сектор розничной торговли**

Розничные торговцы могут повысить ценность потребительских покупок за счет улучшения услуг, для которых RFID предоставляет несколько возможностей. Технология RFID уже начала революционизировать сектор розничной торговли [1]. RFID повышает видимость товара на складе розничной торговли, что помогает улучшить контроль запасов и качество обслуживания клиентов. Это очень актуально в крупных магазинах, где покупатели могут осуществлять поиск выбранных ими товаров онлайн, которые в магазине имеются в наличии на текущий момент для продажи. Во-вторых, RFID обеспечивает улучшенную идентификацию продукта за счет сохранения отличительного идентификационного номера. И, наконец, это сокращает время оформления заказа, поскольку RFID позволяет покупателю сканировать все содержимое корзины, даже не беря ни одного товара. Кроме того, это помогает в динамическом ценообразовании, снижении краж и отслеживании сотрудников.

RFID добавляет гибкости и интеллектуальности процессу для повышения уровня обслуживания. Встроенный автоматизированный прием PCP на производстве позволяет вам продолжать работу, не дожидаясь получения материала [3]. Это позволяет контролировать дату истечения срока годности, автоматическое заполнение недостающих файлов, контроль запасов и возвратов, а также ускоренное оформление заказа. Позволяет вам проверять полки, коробки и поддоны сверху без какого-либо зрительного контакта. Сокращает общее время цикла заказа до момента доставки товара. Уменьшает количество ошибок, допускаемых при доставке заказов клиентов. Позволяет считывать несколько тегов и, следовательно, увеличивает скорость чтения вместо обработки одного штрих-кода за раз. Легкий мониторинг всех логистических операций наряду с повышением безопасности. Повышенная скорость и маневренность при поиске материалов. Помогает избежать фальсификации записи уникальных кодов. Позволяет легко управлять продуктами и материалами с меньшими затратами рабочей силы.

Бирки можно встраивать в товары и карты лояльности, а считыватели можно применять при выезде. Тележки для покупок также могут быть снабжены бирками и даже считывателями. Маркированные корзины для покупок могут собирать данные о местоположении магазина и поведении покупателей и упрощать планирование POS-дисплеев и расположения полок [4]. Метки, размещенные на картах лояльности, могут помочь розничным торговцам отслеживать покупательское поведение клиентов и демографические характеристики и помочь розничным торговцам лучше адаптировать свои рыночные предложения.

Таким образом, технология RFID имеет множество преимуществ использования в сфере торговли. Тем не менее, имеются и проблемы внедрения технологии RFID в разные сферы.

## **Проблемы при внедрении RFID**

Прежде всего, нужно отметить, что RFID - это дорогостоящее дело. RFID требует дорогостоящего оборудования — будь то программное или аппаратное обеспечение. Более того, разговоры о тегах, будь то активные, пассивные или полупассивные, являются дорогостоящим делом и могут отбросить бизнес назад. Цены на RFID-метки снизились с 1970-х годов, но по-прежнему многие компании неохотно внедряют их из-за высоких цен.

Проблемы с металлами и жидкостями. RFID плохо сочетается с металлами и жидкостями, поскольку и то, и другое затрудняет получение правильных показаний на активах [5]. В металлах радиоволны отражаются повсюду. Аналогичным образом, жидкость может поглощать сигналы от RFID-меток.

Трудная для понимания технология. Трудно разобраться в различных тегах и частотах. Менеджеры должны хорошо понимать технологию, чтобы они могли обучать своих сотрудников ее работе.

Чтобы RFID был эффективным, он должен поддерживаться во всей организации розничной торговли — от поставщиков до магазинов и всех членов команды.

## **Розничные торговцы могут преодолеть препятствия на пути внедрения RFID.**

Количественная оценка ценности инвестиций и инициатив в области RFID. Розничные торговцы должны разработать подробные бизнес-кейсы [1]. Существует множество данных для количественной оценки потенциальной отдачи от быстрого управления и реализации. Определение правильных поставщиков/партнеров.

Для масштабирования RFID розничные торговцы могут рассмотреть возможность партнерства с более широкой экосистемой технологических партнеров, поставщиков и консультантов, которые могут способствовать быстрому продвижению.

Общение с сотрудниками и их обучение. RFID требует изменений в операциях хранения, и эти усилия нельзя недооценивать. Наличие надежной стратегии изменений имеет решающее значение для получения поддержки при внедрении RFID.

Чтобы RFID был эффективным, он должен поддерживаться во всей организации розничной торговли — от поставщиков до магазинов и всех членов команды. Руководство может подготовить почву для внедрения, поделившись историями успеха со своими сотрудниками и объяснив путь к созданию ценности с помощью RFID.

Финансовые выгоды RFID на уровне фирмы в первую очередь обусловлены ростом производительности производства, складирования и розничной торговли, а также улучшением качества обслуживания клиентов.

## **Вывод**

В статье рассмотрены ключевые вопросы, касающиеся технологии RFID, а также ее внедрения, применения, преимуществ и проблем. RFID обладает огромным потенциалом, и в ближайшие годы мы увидим его применение в различных отраслях экономики и промышленности.

Т.к. эта технология только начинает активно внедряться в разные сферы жизни, то эта тема для исследования будет оставаться актуальной в дальнейшем.

## **Литература**

1. Accenture Covid-19 Consumer Pulse Research November 26, 2021 to December 10, 2021. URL: <https://www.accenture.com/ca-en/about/newsroom/company-news-release-covid-consumer-research-2021> (дата обращения: 30.10.2022).
2. Зиборов И. А. Применения RFID технологий в деятельности различных субъектах хозяйствования / И. А. Зиборов. — Текст : непосредственный // Молодой ученый. — 2009. — № 12 (12). — С. 17-22.
3. Цифровая экономика: 2022 : краткий статистический сборник / Г. И. Абдрахманова, Ц75 С.А. Васильковский, К.О. Вишневский и др.; Нац. исслед. ун-т «Высшая школа экономики». — М.: НИУ ВШЭ, 2022. — 124 с.
4. Баркова Н. Ю., Бородин О. А. ТЕХНОЛОГИЯ РАДИОЧАСТОТНОЙ ИДЕНТИФИКАЦИИ ДАННЫХ: ПОТЕНЦИАЛ ИСПОЛЬЗОВАНИЯ В ЦЕПЯХ ПОСТАВОК ИНДУСТРИИ МОДЫ // Вестник ГУУ. 2020. №8.
5. Michael Keenan. What Is RFID Technology and How Can I Use It? / Inventory Management. — 2021. — URL: <https://www.shopify.com/retail/rfid-technology#1> (дата обращения: 02.11.2022).

**Боднар А.В., Бондаренко Е.С. Анализ перспектив применения технологии RFID в торговле.** В статье исследована актуальность и значимость использования технологии RFID в секторе торговли. Проанализированы преимущества применения и проблемы внедрения технологии RFID в секторе торговли. Выявлены способы преодоления препятствий на пути внедрения RFID в розничной торговле. Выделены направления развития технологии RFID для повышения эффективности продаж.

**Ключевые слова:** технология RFID, внедрение, розничная торговля, управление запасами, системы управления, цепочка поставок.

**Bodnar A.V., Bondarenko E.S. Analysis of prospects for the use of RFID technology in trade.** The article examines the relevance and significance of the use of RFID technology in the trade sector. The advantages of using and problems of implementing RFID technology in the trade sector are analyzed. The ways of overcoming obstacles to the introduction of RFID in retail trade have been identified. The directions of development of RFID technology to increase sales efficiency are highlighted.

**Keywords:** RFID technology, implementation, retail, inventory management, management systems, supply chain.

## Мобильное приложение с модулем распознавания речи на базе платформы Android

А.С. Пильненький<sup>\*1</sup>, О.И. Федяев<sup>\*2</sup>

<sup>\*1</sup> магистрант, Донецкий национальный технический университет,  
pi.16b.alex@gmail.com

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
fedyaev@donntu.ru, OrcID: 0000-0001-6822-7306

*Пильненький А.С., Федяев О.И. Мобильное приложение с модулем распознавания речи на базе платформы Android. В работе рассматриваются основные подходы к автоматическому распознаванию речи и алгоритмы дешифрования речи. В процессе исследования были найдены алгоритмы и инструментальные средства, необходимые для разработки модуля распознавания речи в среде ОС Android. Созданное приложение VoiceHelper позволяет реализовать удобное речевое управление телефонными звонками и отправками СМС на смартфоне.*

**Ключевые слова:** автоматическое распознавание речи акустическая и языковая модель, декодер, голосовое управление смартфоном, спектр Фурье.

### Введение

История автоматического распознавания человеческой речи началась ранее, чем принято думать: в большинстве случаев принято вести отсчёт с 1952 года, но на самом деле одним из первых устройств, реагировавшим на голосовые команды, был робот Televox. Созданный в 1927 году в США робот Герберт Телевокс представлял собой несложное устройство, в котором различные реле реагировали на звуки разной частоты. В роботе было три камертона, каждый из которых отвечал за свою тональность. В зависимости от того, какой камертон срабатывал, включалось то или иное реле. Следующим ключевым изобретением, открывшим путь к настоящему распознаванию человеческой речи, стала машина Audrey, разработанная в 1952 году в Bell Labs. Вся её функциональность сводилась к распознаванию произнесённых цифр от нуля до девяти, что является хорошим результатом для аналоговой машины. Принцип работы заключался в следующем: диктор проговаривал в микрофон цифры, делая между словами интервалы не менее 350 мс, Audrey переводила услышанные звуки в электрические сигналы и сравнивала их с записанными в аналоговой памяти образцами. По результатам сравнения машина подсвечивала цифру на приборной панели [1,2].

К 1976 году университет Карнеги - Меллона представил систему Harpu, способную оперировать словарём из 1011 слов. Harpu не слыхала целиком услышанные слова с образцами, а разделяла их на аллофоны (образец звучания фонемы в зависимости от окружающих её букв). Это был очередной успех, подтвердивший, что будущее за распознаванием отдельных фонем, а не цельных слов. По-настоящему революционным событием стало внедрение разработанной в 1980-х скрытой марковской модели - статистической модели, строившей точные предположения о неизвестных элементах на основе вытекающих из них известных. Проще говоря, с помощью всего нескольких распознанных фонем в одном слове скрытая марковская модель очень точно подбирает недостающие фонемы, тем самым значительно повышая точность распознавания речи [3,4].

В наше время лучшим средством для создания движка распознавания речи стала рекуррентная нейросеть (RNN), на которой построены все современные сервисы распознавания голоса, музыки, изображений, лиц, объектов, текста. RNN позволяет с высочайшей точностью понимать слова, а также предсказывать наиболее вероятное слово в рамках контекста, если оно не было распознано. Нейросетевая темпоральная классификация модели CTC выделяет в записанном аудиопотоке (слове, фразе) отдельные фонемы и расставляет их в том порядке, в котором они были произнесены. После многократного анализа CTC очень чётко выделяет определённые фонемы, а их текстовая запись сравнивается с базой слов нейросети и затем превращается в распознанное слово [6]. В 1996 году появилась первая коммерческая программа, способная различать не отдельные слова, а непрерывный поток естественной речи - IBM MedSpeak/Radiology [3]. Продукт IBM был специализированным, т.к. он использовался в медицине для стенографирования описания результатов рентгенограммой, произносимых врачом в ходе исследования.

Основной целью данной работы является создание и оценка эффективности программной системы управления смартфоном на базе ОС Android с помощью голосовых команд. Объектом исследования является система автоматического распознавания речи, встроенная в смартфон. Предмет исследования – методы и алгоритмы автоматического распознавания речи, технологии и инструментальные средства разработки систем распознавания и синтеза речи на базе ОС Android.

### **Выбор акустической модели систем распознавания речи**

Для работы создаваемого голосового управления достаточно команд «имя\_помощника», «позвони», «напиши», «имя\_адресата», например: «Денис, позвони Ивану Ивановичу», после чего голосовой помощник по названному имени Денис разберёт текст на отдельные слова и по их вхождению определит дальнейшие действия, и выполнит поставленную ему задачу. Распознавание отдельных слов выполняет акустическая модель системы.

Акустическая модель - это функция, принимающая на вход признаки на небольшом участке акустического сигнала (фрейме) и выдающая распределение вероятностей различных фонем на этом фрейме. Таким образом, акустическая модель дает возможность по звуку восстановить, что было произнесено - с той или иной степенью уверенности.

Самой популярной реализацией акустической модели является Скрытая Марковская Модель (СММ), в которой скрытыми состояниями являются фонемы, а наблюдениями - распределения вероятностей признаков на фрейме. Пример акустической модели для короткого слова «six» представлен на рисунке 1.

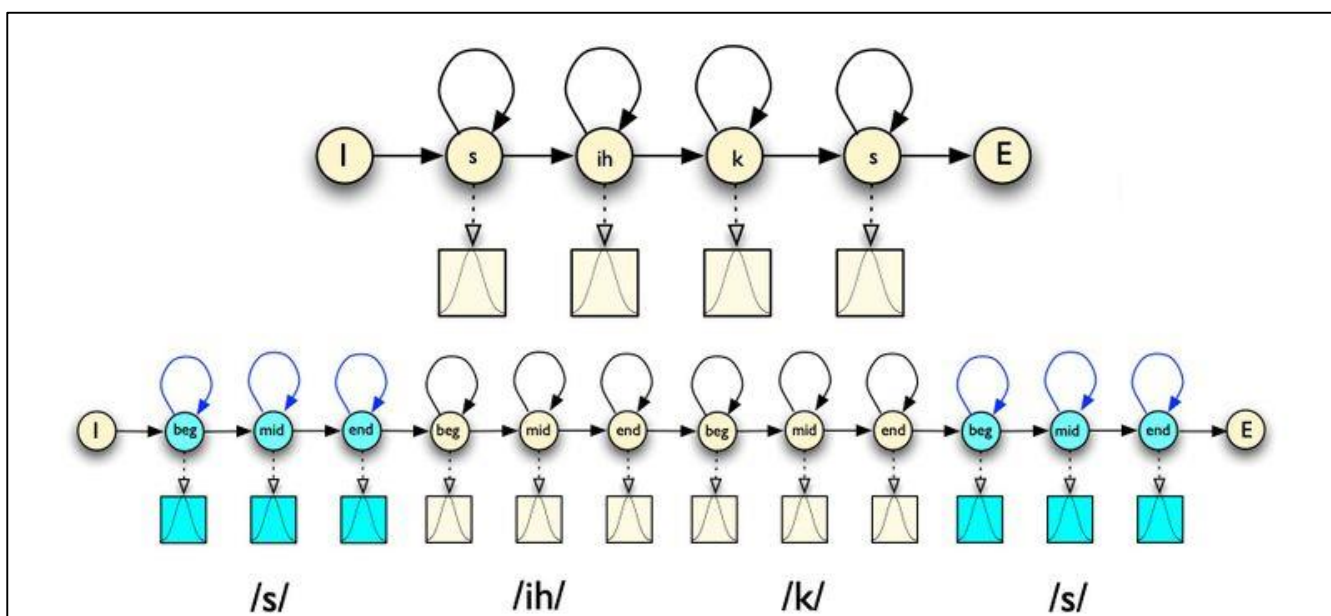


Рисунок 1 – Акустическая модель для слова «six»

В круглых (скрытых) состояниях изображены фонемы, а в квадратных (наблюдениях) - распределения вероятностей признаков (для упрощения, на рисунке изображено одномерное распределение). Фонемы часто разбивают на 3 части - начало, середину и конец, - потому что фонема может звучать по-разному в зависимости от момента времени её произнесения. Каждое скрытое состояние содержит переход само в себя, так как время произнесения одной фонемы может занять несколько фреймов. Вероятности перехода между фонемами в СММ являются обучаемыми параметрами, и для их настройки используют алгоритм Баума-Велша. Последовательность фонем по набору распределений на фреймах восстанавливают по алгоритму Витерсби [7].

В качестве функции распределения вероятностей признаков часто выбирают смешанную гауссову модель (англ. Gaussian Mixture Model, GMM): дело в том, что одна и та же фонема может звучать по-разному, например, в зависимости от акцента. Так как эта функция является, по сути, суммой нескольких нормальных распределений, она позволяет учесть различные звучания одной и той же фонемы.

### **Анализ современных языковых моделей**

Языковая модель позволяет узнать, какие последовательности слов в языке более вероятны, а какие менее. Здесь в самом простом случае требуется предсказать следующее слово по известным предыдущим словам. В традиционных системах применялись модели типа N-грамм, в которых на основе большого количества текстов

оценивались распределения вероятности появления слова в зависимости от  $N$  предшествующих слов [4,5]. Для получения надежных оценок распределений параметр  $N$  должен быть достаточно мал: одно, два или три слова – это модели униграмм, биграмм или триграмм соответственно. Внедрение языковой модели в систему распознавания речи позволило значительно повысить качество распознавания за счёт учёта контекста.

Нейронные сети – это другой класс моделей обработки языка (или языковые модели с непрерывным пространством). Они используют непрерывные представления или вложения слов, чтобы делать свои прогнозы. Эти модели используют рекуррентные нейронные сети. Встраивание в непрерывное пространство помогает уменьшить проклятие размерности в языковом моделировании: по мере того, как языковые модели обучаются на всё больших и больших текстах, количество уникальных слов (словарный запас) увеличивается. Количество возможных последовательностей слов экспоненциально увеличивается с размером словаря, вызывая проблему разреженности данных из-за экспоненциально большого количества последовательностей.

Таким образом, для правильной оценки вероятностей необходима статистика. Нейронные сети избегают этой проблемы, представляя распределение слов, как нелинейные комбинации весов в нейронной сети. Альтернативное описание состоит в том, что нейронная сеть аппроксимирует языковую функцию. Архитектура нейронной сети может быть прямой или рекуррентной, и хотя первая проще, вторая встречается чаще. Как правило, языковые модели нейронных сетей строятся и обучаются как вероятностные классификаторы, которые учатся предсказывать распределение вероятностей [7].

В будущем такие нейросетевые модели смогут решать больше задач, связанных с текстом. Например, даже нейросеть GPT-3 пока не умеет отслеживать источники и предоставлять пользователю доказательства своих ответов. Но в перспективе её можно научить сохранять и оценивать сайты, с которых она берет информацию. Сейчас с помощью нейросетевых моделей можно проследить, как меняются числовые данные (например, стоимость акций на бирже) и сделать прогноз на будущее. В новостях такие последовательности и изменения пока отслеживать трудно.

Кроме того, с помощью языковых моделей обработки языка класса NLP можно будет решать всё более сложные проблемы: например, они смогут давать юридические консультации, опираясь на загруженные в них законы и судебные практики. Однако для точных заключений и для задач, требующих воображения, всё равно всегда будет нужен человек – как и для составления и обучения моделей.

В разрабатываемой нами программной системе последовательность слов чётко определена (см. пример в предыдущем разделе), что упрощает разработку языковой модели.

### Декодер

В ходе работы системы автоматического распознавания речи задача распознавания сводится к определению наиболее вероятной последовательности слов, соответствующих содержанию речевого сигнала. Наиболее вероятный кандидат должен определяться с учётом как акустической, так и лингвистической информации. Это означает, что необходимо производить эффективный поиск среди возможных кандидатов с учётом различной вероятностной информации. При распознавании слитной речи число таких кандидатов огромно, и даже использование самых простых моделей приводит к серьезным проблемам, связанным с быстродействием и памятью систем. Как результат, эта задача выносится в отдельный модуль системы автоматического распознавания речи, называемый декодером [5]. Декодер должен определять наиболее грамматически вероятную гипотезу для неизвестного высказывания, т.е. определять наиболее вероятный путь по сети распознавания, состоящей из моделей слов (которые, в свою очередь, формируются из моделей отдельных фонов), с помощью аргумента максимизации  $\arg\max$ . Правдоподобие гипотезы определяется двумя факторами, а именно, вероятностями последовательности фонов, приписываемыми акустической моделью, и вероятностями следования слов друг за другом, определяемыми моделью языка. Математическая основа декодеров представлена в следующей формуле

$$W = \operatorname{argmax}[P(W)P(X|W)]$$

В данной формуле используются такие переменные:  $X = x_1^T = x_1, \dots, x_N$  – последовательность векторов признаков входного сигнала;  $W = w_1^n = w_1, \dots, w_n$  – последовательность слов, принадлежащих словарю размером  $N_W$ . Первый множитель  $P(W)$  описывает вклад лингвистического модуля, второй  $P(X|W)$  – лексического, фонетического и акустического источников знаний. В соответствии с концепцией марковских цепей, второй множитель представляет собой сумму вероятностей всех возможных последовательностей состояний, что приводит к уравнению:

$$W = \operatorname{argmax}[P(W) \sum_{s_1^T} P(x_1^T, s_1^T | w_1^N)] \quad .$$

В данной формуле используются следующие переменные:  $S_1^T$  – одна из последовательностей состояний, порождаемых последовательностью слов  $W_1^n$ . Обычно применяется критерий Витерсби – поиск последовательности состояний, дающих максимальный вклад в сумму

$$W = \operatorname{argmax}[P(W)^a \operatorname{Max}[P(x_1^T, s_1^T | w_1^N)]] \quad .$$

Различают систему раннего и систему позднего предсказания. В первой выполняется предсказание для акустической и языковой модели независимо, а затем оба предсказания поступают в декодер. При позднем предсказании, вычисленные признаки речи в акустической и языковой моделях без предсказания поступают в декодер, и уже на основе их совместного декодирования выполняется предсказание.

Были реализованы следующие этапы распознавания:

1. Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.
2. Результат оценки поступает в модуль акустической адаптации, который управляет модулем расчёта параметров речи, необходимых для распознавания.
3. В сигнале выделяются участки, содержащие речь, и происходит оценка параметров речи. Происходит выделение фонетических и просодических вероятностных характеристик для синтаксического, семантического и прагматического анализа (оценка информации о части речи, форме слова и статистические связи между словами)
4. Далее параметры речи поступают в основной блок системы распознавания - декодер. Это компонент, который сопоставляет входной речевой поток с информацией, хранящейся в акустических и языковых моделях, и определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания.

### **Речевое управление телефонными звонками и отправка СМС на смартфоне**

В разработанном приложении на базе ОС Android реализован следующий функционал: 1) звонки выбранному абоненту (звонок происходит сразу); 2) подготовка и отправка СМС выбранному абоненту (перед отправкой будет открыт стандартный редактор СМС-сообщений для редактирования в случае возможной опечатки или неправильно распознанного слова). В приложении предусмотрена одна основная форма (см. рис.2).

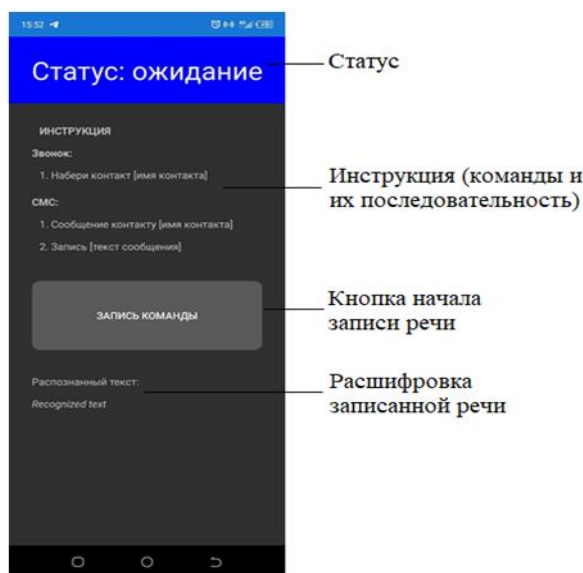


Рисунок 2 – Экранная форма приложения

Форма содержит индикатор статуса (Звонок или СМС), инструкцию для осуществления звонков и отправки СМС, кнопку начала записи речи и распознанный текст ниже кнопки.

Входные данные представляют собой непрерывную осциллограмму звуковой волны. В задачах распознавания речи эту осциллограмму разбивают на фреймы - фрагменты звукового потока длительностью около 20 мс и шагом 10 мс. Такой размер соответствует скорости человеческой речи: если человек говорит по 3 слова в секунду, каждое из которых состоит примерно из 4 звуков и каждый звук разбивается на 3 части, то на



одну часть выходит около 28 мс. Каждый фрейм независимо трансформируется и подвергается извлечению признаков, тем самым образуя векторизованный набор данных для задачи машинного обучения. Формирование фреймов представлено на рисунке 3:

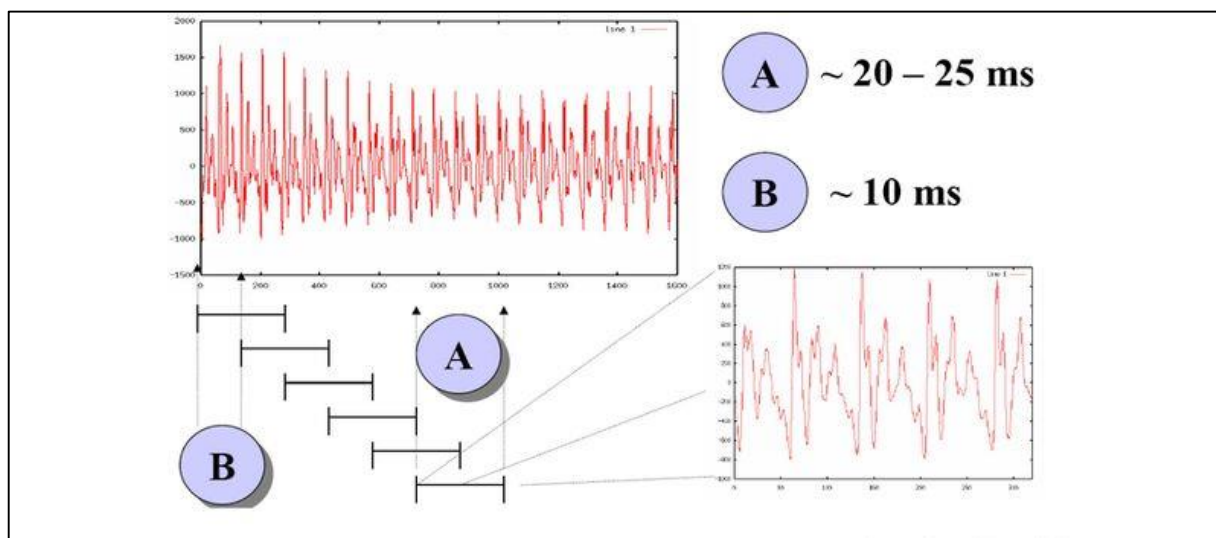


Рисунок 3 – Формирование фреймов из звукового потока

Спектр Фурье получают, используя алгоритм Быстрого Преобразования Фурье (БПФ) с длиной окна равной 2-4 периода основного тона, что составляет около 20 мс. При частоте квантования 10-16 кГц выбирается окно размером 256 отсчётов.

Для ослабления искажений сигнала, вызванных применением к непрерывному сигналу конечного окна анализа, чаще всего используется окно Хэмминга по формуле:  $S(n) = [0.54 - 0.46 \cdot \cos(2\pi n / (N-1))] \cdot S(n)$ , где  $n = 1..N$ ;  $N$  – размерность окна;  $S(n)$  – отсчёты речевого сигнала.

К каждому кадру, полученного Фурье спектра применяется блок мел-фильтров - треугольных пересекающихся фильтров, расположенных наиболее плотно в области нижних частот. Количество фильтров - 26. Для расчёта фильтров выбирается верхняя и нижняя частота. Затем осуществляется переход от частотной шкалы к мел-шкале по формуле:  $M(f) = 1127 \cdot \ln(1 + f/700)$ . На мел-шкале выбираются линейно расположенные точки (28 точек для 26 фильтров), после чего производится обратный переход в частотную область.

Описанный функционал приложения реализован в виде проекта VoiceHelper [8] по объектно-ориентированной технологии, содержащий следующие классы:

- класс Распознаватель речи;
- класс Автоматизатор телефонных звонков;
- класс Формирователь СМС-сообщений;
- класс Обработчик исключений.

### **Заключение**

В работе проанализированы основные подходы к организации голосового управления программно-аппаратными системами. Рассмотрены наиболее известные технологии компьютерного распознавания речи и перевода речи в текст. Выбраны современные способы автоматического распознавания речи на основе модельного подхода, который основан на построении языковой и акустической моделей. Для реализации акустической модели использовалась скрытая марковская модель (СММ). Разработан алгоритм автоматического распознавания речи с использованием разработанных моделей описания речи. Для распознавания слитной речи был разработан декодер, который определяет наиболее грамматически вероятную гипотезу для неизвестного высказывания. Дано математическое описание процесса декодирования речи.

Разработан комплекс алгоритмов и программных средств, позволяющих распознавать речь с переводом в текст. Созданное приложение VoiceHelper позволяет реализовать удобное речевое управление телефонными звонками и отправками СМС на смартфоне.

### **Литература**

1. Аргументы максимизации и минимизации – Режим доступа: [u.wikipedia.org/wiki/Аргументы\\_максимизации\\_и\\_минимизации#:~:text=Аргумент%20максимизации%20\(argmax%20или%20arg,быть%20расширен%20до%20набора%20решений](http://u.wikipedia.org/wiki/Аргументы_максимизации_и_минимизации#:~:text=Аргумент%20максимизации%20(argmax%20или%20arg,быть%20расширен%20до%20набора%20решений) - Загл. с экрана;
2. Распознавание речи – Режим доступа: [ru.wikipedia.org/wiki/Распознавание\\_речи#Методы\\_и\\_алгоритмы\\_распознавания\\_речи](http://ru.wikipedia.org/wiki/Распознавание_речи#Методы_и_алгоритмы_распознавания_речи) - Загл. с экрана;
3. Синтез и распознавание речи. Современные решения – Режим доступа: [frolov-lib.ru/books/hi/ch05.html?ysclid=I9zddr80mp325767398](http://frolov-lib.ru/books/hi/ch05.html?ysclid=I9zddr80mp325767398) - Загл. с экрана;
4. Распознавание речи: классический подход – Режим доступа: [machinelearning.ru/wiki/images/c/c3/Digital\\_Signal\\_Processing%2C\\_lecture\\_6.pdf](http://machinelearning.ru/wiki/images/c/c3/Digital_Signal_Processing%2C_lecture_6.pdf) - Загл. с экрана;
5. Создание системы распознавания речи начального уровня – Режим доступа: [soundchange.ucoz.ru/Books/Попов.pdf](http://soundchange.ucoz.ru/Books/Попов.pdf) - Загл. с экрана;
6. Методы распознавания речи – Режим доступа: [moluch.ru/archive/130/36213/?ysclid=I9zdi257xw226547077](http://moluch.ru/archive/130/36213/?ysclid=I9zdi257xw226547077) - Загл. с экрана;
7. Доступно о машинном обучении: распознавание речи – Режим доступа: [tproger.ru/articles/ai-in-speech-recognition/?ysclid=I9zdi257xw226547077](http://tproger.ru/articles/ai-in-speech-recognition/?ysclid=I9zdi257xw226547077) - Загл. с экрана;
8. Пильненький, А.С. Технология построения речевого командного управления программами на базе платформы Android. / А.С. Пильненький, О.И. Федяев // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2021) / Материалы VII Международной научно-технической конференции. - Донецк, ДонНТУ – 2021. С. 200-207.

*Пильненький А.С., Федяев О.И. Мобильное приложение с модулем распознавания речи на базе платформы Android. В работе рассматриваются основные подходы к автоматическому распознаванию речи и алгоритмы дешифрования речи. В процессе исследования были найдены алгоритмы и инструментальные средства, необходимые для разработки модуля распознавания речи в среде ОС Android. Созданное приложение VoiceHelper позволяет реализовать удобное речевое управление телефонными звонками и отправками СМС на смартфоне.*

**Ключевые слова:** автоматическое распознавание речи акустическая и языковая модель, декодер, голосовое управление смартфоном, спектр Фурье.

*Pilnenky A.S., Fedyaev O.I. A mobile application with a speech recognition module based on the Android platform. The paper discusses the main approaches to automatic speech recognition and speech decryption algorithms. In the course of the research, algorithms and tools necessary for the development of a speech recognition module in the Android OS environment were found. The created Voice Helper application allows you to implement convenient voice control of phone calls and SMS messages on your smartphone.*

**Key words:** automatic speech recognition, acoustic and language model, decoder, smartphone voice control, Fourier spectrum.

## Исследование методов обработки изображений и распознавания образов

Д.А. Присяжный<sup>\*1</sup>, О.М. Копытова<sup>\*2</sup>

<sup>\*1</sup> магистрант, Донецкий национальный технический университет,  
judaspriest999@mail.ru

<sup>\*2</sup> к.ф.-м.н, доцент, Донецкий национальный технический университет,  
omkop@list.ru

**Присяжный Д.А., Копытова О.М. Исследование методов обработки изображений и распознавания образов.** В работе рассмотрены базовые методы обработки и распознавания изображений; описаны общие задачи, решаемые данными методами.

**Ключевые слова:** распознавание образов, обработка изображений, нейросеть.

### Введение

Впервые методы для обработки изображений были разработаны в 60х годах 20го века в Лаборатории реактивного движения, Массачусетском технологическом институте, Лабораториях Белла, Мэрилендском университете и других исследовательских центрах. Они использовались для обработки снимков со спутников, для распознавания примитивных символов вроде крестиков и ноликов, а также для улучшения качества фотографий.

В настоящее время все большую актуальность приобретает процесс обработки изображений нейронными сетями, которые позволяют не только обработать изображение, но также выделить на нем различные объекты и определить их. Методы обработки изображений широко используются в поисковых системах, которые позволяют найти различные объекты, изображенные на фото (например, логотип бренда, похожий автомобиль). Современные методы обработки изображений позволяют решать разнообразные задачи: в каких-то ситуациях требуется найти на изображении определенный товар, в некоторых случаях нужно распознать определенный текст и перевести его на другой язык, возможен также специфический запрос – найти человека по лицу.

Количество информации в мире с каждым годом увеличивается в геометрической прогрессии, и люди часто сталкиваются с проблемой поиска нужно (релевантной) информации. Естественно, в глобальной сети, количество изображений будет постоянно расти, также будет расти количество запросов на распознавание объектов в режиме реального времени. В данном случае мы сталкиваемся с рядом проблем – это, конечно же, большая нагрузка на серверную часть и вычислительные мощности, а также релевантность тех или иных значений, выдаваемых поисковыми системами, системами распознавания объектов и др.

По причине постоянно увеличивающегося числа изображений растет потребность в их тщательном анализе для того, чтобы свести возможность получения нерелевантных результатов к минимуму. Для этого применяются разные подходы, причём чаще всего комбинируется несколько подходов: вначале могут применяться разного рода простые методы для удаления шумов, выравнивания изображения, улучшения качества, а далее обработанная картинка подается на вход нейронной сети, которая уже выдает финальный результат. Это довольно сложный алгоритм обработки изображений.

Целью работы является обзор базовых методов обработки изображений, а также методов распознавания объектов на них.

### Методы обработки изображений

Компьютерное зрение (Computer Vision, CV) – это область искусственного интеллекта, связанная с анализом изображений и видео [1]. Она включает в себя набор методов, которые наделяют компьютер способностью «видеть» и извлекать информацию из увиденного. Системы состоят из фото- или видекамеры и специализированного программного обеспечения, которое идентифицирует и классифицирует объекты. Они способны анализировать образы (фотографии, картинки, видео, штрих-коды), а также лица и эмоции. Чтобы научить компьютер «видеть», используются технологии машинного обучения. Собирается множество данных,

которые позволяют выделить признаки и комбинации признаков для дальнейшей идентификации похожих объектов.

Задачи, решаемые компьютерным зрением, можно разделить на «простые» и «сложные». «Сложные» определяют, что за объект на картинке, а также позволяют отнести его к тому или иному классу объектов при решении задач классификации. Они позволяют определить, к какому классу относятся объекты одного типа (напр. автомобиль – седан, кабриолет, хетчбек). В таких задачах чаще всего используются методы машинного обучения.

«Простые» же выполняют операции с пикселями, которые не используют машинное обучение. Они чаще всего предназначены для предварительной обработки «сложных» задач и отвечают за начальную обработку, преобразование изображений, выравнивание цветов, определение границ и т.п.

Одним из наиболее широко используемых и достаточно простых методов обработки изображений является линейная фильтрация. Она позволяет:

- скомбинировать пиксели входного изображения в выходное;
- выделить углы;
- убрать шум;
- изменить размер изображения.

Для линейной фильтрации используется формула скользящего среднего. Это один из методов свертки.

Линейная фильтрация изображения выполняется через названную свертку операции. Свертка является операцией окружения, в которой каждый выходной пиксель является взвешенной суммой соседних входных пикселей. Матрица весов называется ядром свертки, также известным как фильтр.

Чтобы посчитать среднее для изображения с окном (ядром свертки) 3 на 3 пикселя (см. рис. 1), можно воспользоваться формулой (1).

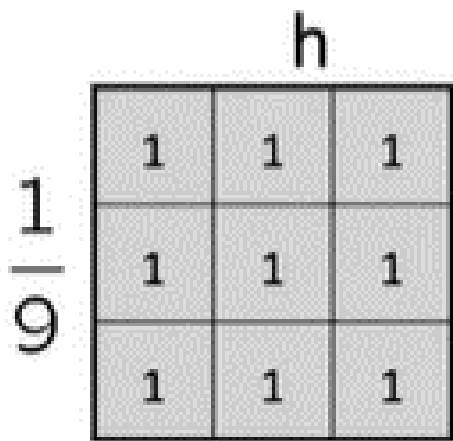


Рисунок 1 – Матрица 3 на 3

$$\frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k, l] = \frac{1}{9} \sum_{k=-1}^1 \sum_{l=-1}^1 f[n-k, m-l] \quad (1)$$

Таким образом, получается выражение для свертки, где  $f$  – изображение, двумерная функция, которая характеризует картинку,  $k$  и  $l$  – координаты пикселя,  $n$ ,  $m$  – индексы строк и столбцов матрицы.

Допустим, есть изображение размером 10 на 10, и нужно его пропустить через усредняющий фильтр размером 3 на 3.

Начиная с крайнего левого угла изображения, необходимо поместить туда фильтр размером 3 на 3 и заменить центральный элемент на среднее по всем девяти элементам (сумма элементов, попавших в фильтр, деленная на 9). Это будет только первым этапом. Далее нужно сдвинуть фильтр на один шаг вправо и повторить эту процедуру (см. рис 2).

После применения такой матрицы изображение получается более размытым.

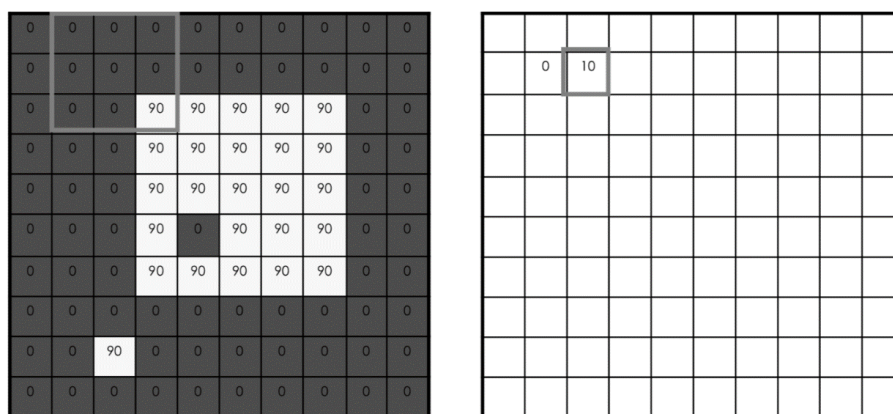


Рисунок 2 – Применение фильтра 3 на 3 к изображению размером 10 на 10

### Распознавание образов

Возможно использование различных подходов для распознавания образов или объектов. В последнее время методы машинного и глубокого обучения стали популярными подходами к проблемам распознавания объектов. Обе технологии учатся распознавать объекты на изображениях, но они различаются по своему исполнению.

Методы глубокого обучения стали популярным методом распознавания объектов. Модели глубокого обучения, такие как сверточные нейронные сети (или CNN), используются для автоматического изучения присущих объекту свойств, чтобы идентифицировать этот объект. Например, CNN может научиться определять различия между кошками и собаками, анализируя тысячи изображений и изучая характеристики, различающие кошек и собак.

Существует два подхода к распознаванию объектов с использованием глубокого обучения [2].

#### 1. Обучение модели с нуля.

Чтобы обучить глубокую сеть с нуля, необходимо собрать очень большой размеченный набор данных и разработать архитектуру сети, которая будет изучать характеристики и строить модель. Результаты могут быть впечатляющими, но этот подход требует большого количества обучающих данных, а также необходимо настраивать уровни и веса в CNN.

#### 2. Использование предварительно обученной модели глубокого обучения.

Большинство приложений глубокого обучения используют подход трансферного обучения – процесс, который включает в себя точную настройку предварительно обученной модели. Для начала берется существующая сеть, такая как AlexNet или GoogLeNet, и вводятся новые данные, содержащие ранее неизвестные классы. Этот метод требует меньше времени и может обеспечить более быстрый результат, поскольку модель уже обучена на тысячах или миллионах изображений. Глубокое обучение предлагает высокий уровень точности, но требует большого количества данных для точных прогнозов (см. рис 3).

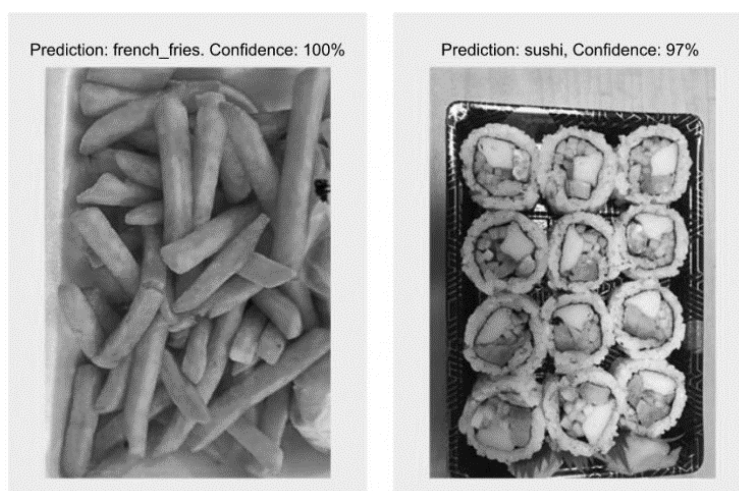


Рисунок 3 – Приложение глубокого обучения, демонстрирующее распознавание объектов ресторанной еды.

Задача поиска изображения по образцу является частью (подзадачей) более общей задачи распознавания образов. При несистематизированном и ненаправленном поиске «схожих» объектов из множества объектов, их можно перечислять бесконечно долго и не прийти к завершению с заданной вероятностью. В частных случаях объекты характеризуются такими идентификационными параметрами (признаками), как форма, цвет, положение, подвижность, по отличительным особенностям, их комбинации и т.п. В зависимости от этих факторов объекты подвергаются классификации. Часто стоит не глобальная задача классификации всех окружающих объектов, а необходимость выделить в поступающем видеопотоке объекты определенного рода. Рассмотрим далее наиболее распространенные классификационные признаки.

Когда встает задача выделить объекты по форме, должны быть заданы классификационные примитивы. В большинстве методов поиска в качестве примитивов используются круглые, эллиптические, прямоугольные или прямолинейные объекты.

Универсальным способом поиска по форме признан метод вписывания шаблона. Шаблон, имеющий форму, объекты которой необходимо выделить, перемещается по изображению, рассчитывается характеристика положения, и там, где показатель этой характеристики превышает некоторый порог – может находиться объект искомой формы.

Методы линейной фильтрации, один из которых был рассмотрен выше, также могут применяться для решения задачи нахождения объектов на картинке. Примером такого метода линейной фильтрации является корреляционный метод [3]. Названная корреляция операции тесно связана со сверткой. В корреляции значение выходного пикселя также вычисляется как взвешенная сумма соседних пикселей. Этот метод очень похож на свертку, но записывается в другом виде (см. формулу (2)). Корреляция применяется для того, чтобы охарактеризовать меру схожести двух изображений (см. рис. 4) [4].

$$r_{fg}[k, l] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f[n, m] g * [n - k, m - l] =$$

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} f[n + k, m + l] g * [n, m], \quad k, l \in Z \quad ((2))$$

Здесь  $f$  – изображение, двумерная функция, которая характеризует картинку,  $n$  и  $m$  – координаты пикселя,  $g$  – второе изображение с координатами пикселей  $k$  и  $l$ .  $r_{fg}$  – показатель взаимной корреляции  $f$  и  $g$ .

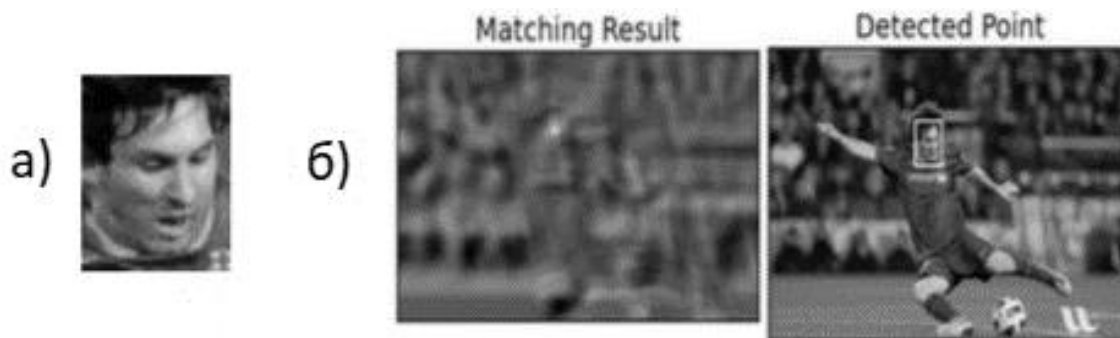


Рисунок 4 – Корреляция изображений:  
а) первое изображение, б) результат корреляции

На рисунке 4(б) слева показан результат применения корреляции для поиска лица. Белое пятно – это место, где оно найдено. Также корреляцию можно вычислять с использованием методов преобразования Фурье, которые тоже относятся к методам свертки изображений.

Если имеется точная копия изображения, то корреляция — это удобный способ поиска объектов на изображениях.

Можно выделить следующие методы распознавания образов [5].

1. Метод перебора. В этом случае производится сравнение с базой данных, где для каждого вида объектов представлены всевозможные модификации отображения. Например, для оптического распознавания образов можно применить метод перебора вида объекта под различными углами, масштабами, смещениями,

деформациями и т. д. Для букв нужно перебирать шрифт, свойства шрифта и т. д. В случае распознавания звуковых образов, соответственно, происходит сравнение с некоторыми известными шаблонами (например, слово, произнесенное несколькими людьми).

2. Более глубокий анализ характеристик образа. В случае оптического распознавания это может быть определение различных геометрических характеристик. Звуковой образец в этом случае подвергается частотному, амплитудному анализу и т. д.

3. Использование искусственных нейронных сетей (ИНС). Этот метод требует либо большого количества примеров задачи распознавания при обучении, либо специальной структуры нейронной сети, учитывающей специфику данной задачи. Тем не менее, его отличает более высокая эффективность и производительность.

4. Экспертный метод, основанный на непрерывном обучении экспертной системы в процессе эксплуатации.

## **Выводы**

В работе описаны общие подходы к обработке и распознаванию изображений. Описана линейная фильтрация как один из наиболее распространенных методов обработки изображений, позволяющий без особых вычислительных затрат преобразовать или обработать картинку в зависимости от заданной матрицы. Этот метод легко позволяет добавить размытие на изображение или убрать излишние шумы. Для линейной фильтрации приведены иллюстрации, схематически показывающие последовательность действий. Также в работе описан базовый корреляционный метод, позволяющий определить меру схожести двух изображений. Он прост в реализации и может использоваться для сравнения изображения по образцу с небольшой базой. В работе также перечислены существующие методы распознавания образов.

## **Литература**

1. Что такое компьютерное зрение и где его применяют [Электронный ресурс]. – Режим доступа: <https://trends.rbc.ru/trends/industry/5f1f007e9a794756fafbfa83>
2. Попова, Л.П. Обзор существующих методов распознавания образов [Электронный ресурс] / Л.П. Попова, И.О. Датев // Сайт <http://textarchive.ru>. – Режим доступа: [http://matlab.exponenta.ru/fuzzylogic/book5/1\\_3.php](http://matlab.exponenta.ru/fuzzylogic/book5/1_3.php)<http://textarchive.ru/c-2235514.html>
3. Корреляционное сопоставление изображений [Электронный ресурс]. – Режим доступа: [http://wiki.technicalvision.ru/index.php/Корреляционное\\_сопоставление\\_изображений](http://wiki.technicalvision.ru/index.php/Корреляционное_сопоставление_изображений)
4. Корреляционное сопоставление изображений [Электронный ресурс]. – Режим доступа: [http://wiki.technicalvision.ru/index.php/Корреляционное\\_сопоставление\\_изображений](http://wiki.technicalvision.ru/index.php/Корреляционное_сопоставление_изображений)
5. Бишоп, К. Распознавание образов и машинное обучение / Кристовер Бишоп. – М.: «Диалектика», 2020 – 480 с.

*Присяжный Д.А., Копытова О.М. Исследование методов обработки изображений и распознавания образов. В работе рассмотрены базовые методы обработки и распознавания изображений; описаны общие задачи, решаемые данными методами.*

**Ключевые слова:** распознавание образов, обработка изображений, нейросеть.

*Prisazhnyi D.A., Kopytova O.M. Investigation of methods of image processing and pattern recognition. The paper considers the basic methods of image processing and recognition; the general tasks solved by these methods are described.*

**Key words:** image recognition, image processing, neural network.

## ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ПРИНЦИПА РАБОТЫ НЕЙРОННЫХ СЕТЕЙ

Я.Р. Парсаданян<sup>\*1</sup>, А.В. Боднар<sup>\*2</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
ypars00@mail.ru

<sup>\*2</sup> доцент, Донецкий национальный технический университет,  
linabykova13@ya.ru

**Я.Р. Парсаданян, А.В. Боднар Теоретический анализ принципа работы нейронных сетей.**  
*В статье рассмотрен принцип работы нейронной сети. Изучены актуальные виды архитектур (полносвязная прямого распространения, сверточная, рекурсивная), проанализированы их особенности, преимущества и недостатки. На основе данного анализа сформулирован вывод о дальнейших исследованиях.*

**Ключевые слова:** Нейронная сеть, нейрон, архитектура, вес, информация, рекурсивная нейронная сеть, сверточная нейронная сеть, полносвязная нейросеть.

### Введение

Искусственная нейронная сеть – математическая модель, программная и аппаратная её реализация, которая стремится воссоздать организацию и функционирование биологической нейронной сети. Входящие данные обрабатываются по принципу процессов, которые происходят в нейронах живых существ.

Актуальность применения нейронных сетей заключается в необходимости решения плохо формализованных задач. Стоит обратить внимание на актуальность различных типов архитектур нейронных сетей. Ведь от запрашиваемой задачи и параметров более рационально использовать определенную структуру.

Потенциальные области применения искусственных нейронных сетей являются – те, где человек малоэффективен, а привычные расчеты трудоемки. Основные области применения нейронных сетей - автоматизация: классификации, прогнозирования, распознавания, принятия решений и другие [1].

Актуальность выбранной темы связана с активным развитием данной технологии в сфере информационных технологий. Нейронные сети – перспективное направление и решение большого количества задач человека. Чтобы правильно их реализовывать и использовать, нужно понимать принцип их работы.

Целью статьи является теоретический анализ принципов работы нейронной сети.

### История развития нейронных сетей

Человек всегда стремился познать строение мозга, принципы его работы и организации. В 1943 году была создана первая математическая модель нейрона. Ученый Уоррен Маккаллоу и его ученик У. Питтс сформулировали основные положения теории деятельности головного мозга:

разработана модель нейрона как простейшего процессорного элемента, которая вычисляет значение переходной функции от скалярного произведения вектора входных сигналов и вектора весовых коэффициентов; предложена конструкция сети таких узлов для осуществления логических и арифметических операций; такая сеть способна обучаться, распознавать образы, обобщать полученную информацию.

В наше время теория нейронных сетей ушла вперед, но большинство положений работы У. Маккаллоу остаются актуальными в работах современников.

Исследования Д.Хэбба также оставили неизгладимый след в области нейронных сетей. Он в 1949 году предложил закон обучения, который стал основой для будущих алгоритмов обучения нейронных сетей.

Далее к развитию технологии приложила руку монография нейрофизиолога Ф.Розенблатта «Принципы нейродинамики». В данной работе детально изложено устройство перцептрона (устройство, которое моделирует процесс восприятия информации мозгом). Описанные идеи нашли применение в дальнейших исследованиях. Таким образом в 1985-1986 годах теория нейронных сетей получила толчок в развитии. Также причиной этого толчка стала возможность моделировать сети на персональных компьютерах. Конечно мощность и размер значительно отличаются от современных, однако основополагающие принципы активно применялись [2].

Нейронные сети продолжают активно развиваться, находить себе применение. По оценкам специалистов, в скором времени ожидается технологический рост в области проектирования нейронных сетей. За последние



годы уже открыто немало новых возможностей нейронных сетей, а работы в данной области вносят существенный вклад в промышленность, науку и технологии, а также имеют большое экономическое значение.

### Основные понятия архитектуры искусственных нейронных сетей

Нейрон (в биологии) – клетка нервной системы, служит для приёма, обработки, хранения, передачи и вывода информации, которая поступает извне. Он состоит из тела, которое включает в себя ядро (содержит информацию), аксона и дендритов (см. рис. 1).

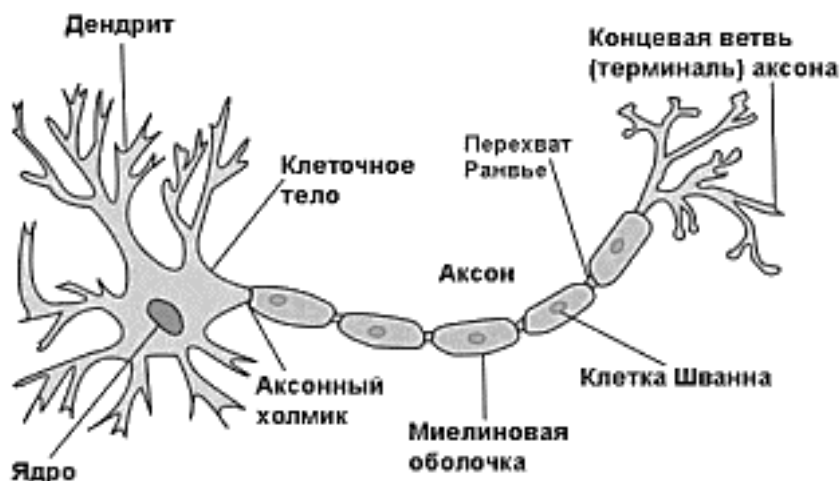


Рисунок 1 – Строения нейрона (биологический) [3]

Принцип работы следующий:

нейрон получает импульсы от других нейронов через дендриты;

информация переходит в тело и обрабатывается;

сигнал передается через аксон, который разветвляется на синапсы.

Синапс является связующим звеном между двумя нейронами.

Теперь рассмотрим строение формального нейрона искусственной сети (см. рис. 2). Рассмотрим принцип работы формального нейрона, зная строение биологического.

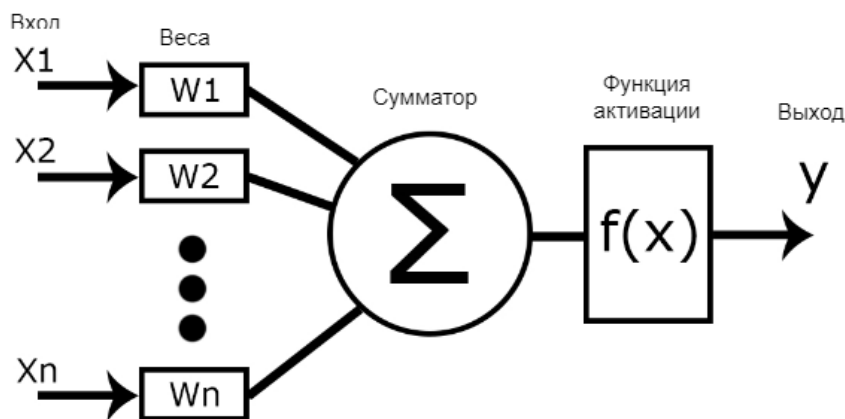


Рисунок 2 – Строения нейрона (формальный) [4]

Процесс вычисления выходного значения формального нейрона представляет собой движение потока данных и их преобразование.

Принцип работы следующий:

нейрон получает на вход сигналы за счёт нескольких входных каналов (дендриты);

данные преобразуются с помощью математических действий функции активации (тело нейрона);

результат передаётся на выход через синапсы для обработки другими нейронами.

Как видим формальный нейрон реализован по подобию биологического. Стоит отметить особенность вычисления суммы за счёт синапсов, определенными весовыми коэффициентами. Если они положительные, то они оказывают на возбуждающее действие, если отрицательные – тормозящее [5]. За счёт весов (приобретаются во время обучения) на выходе получаем вероятность того или иного варианта решения задачи.

Искусственная нейронная сеть – это совокупность формальных нейронов. Формальные нейроны бывают: входные, скрытые и выходные. Таким образом они могут сформировать целую сеть (см. рис. 3).

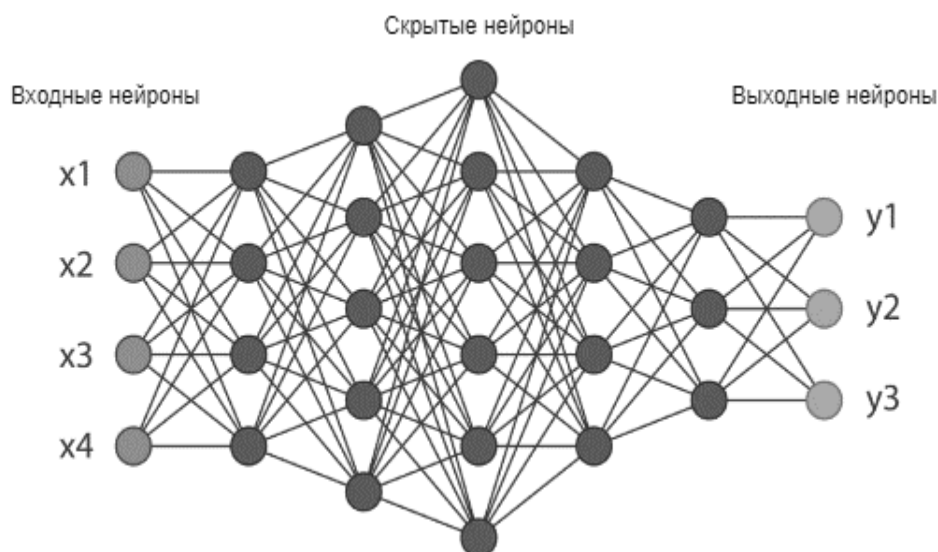


Рисунок 3 – Строение искусственной нейронной сети [6]

### **Классификация архитектур нейронных сетей**

В зависимости от задачи, требований и сферы применения нейронной сети можно выбрать различные её архитектуры. Рассмотрим самые распространенные виды архитектур и их особенности.

Первая классическая архитектура – полносвязные нейросети прямого распространения (см. рис. 3). Многослойный перцептрон состоит из 3 или более слоев. Все нейроны слоя связаны со всеми нейронами следующего слоя, чем и обеспечивается полная связность.

Основное преимущество - отличное решение задач классификации. Однако есть недостатки: большое количество параметров и затухающие градиенты.

Перейдем к более нестандартным видам архитектур.

Сверточная нейронная сеть содержит один или более объединенных (соединенных сверточных) слоев (см. рис. 4). Такой тип сетей использует различные варианты многослойного перцептрона. Объединенные слои выполняют свёртку для входных импульсов и передают информацию в следующий слой. Эта операция позволяет сети быть глубже с меньшим количеством параметров [7].

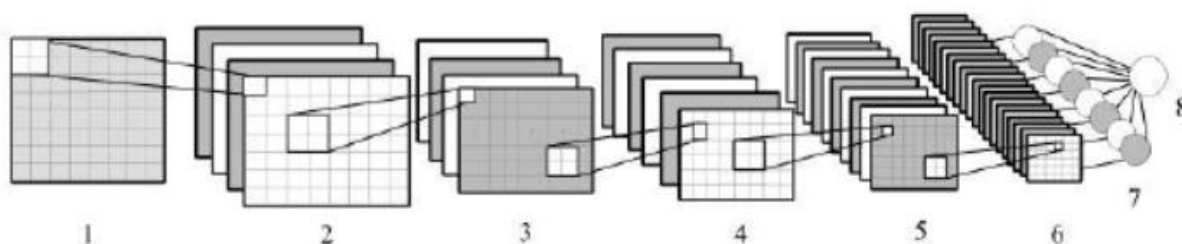


Рисунок 4 – Сверточная нейронная сеть: 1 – вход; 2, 4, 6 – свёрточные слои; 3, 5 – подвыборочные слои; 7 – слой нейронов; 8 – выход [8]

Сверточные нейросети справляются со следующими типами задач: классификация, детекция, сегментация. Данный тип также справляется с генерацией изображения [7].

Из преимуществ описанной архитектуры можно выделить значительно меньшее количество параметров относительно полносвязной нейросети прямого распространения. Как следствие более простое обучение. Из недостатков можно выделить долгое время обучения, большое количество необходимых образцов.

Следующая архитектура – рекурсивная нейронная сеть.

Рекурсивная нейронная сеть – тип сети, который сформирован за счёт обращения одних и тех же наборов весов рекурсивно над структурой (см. рис. 5). Данный тип применяется, чтобы создать скалярное или структурированное прогнозы по входным данным. Данные модели хорошо подходят для генерации текста.

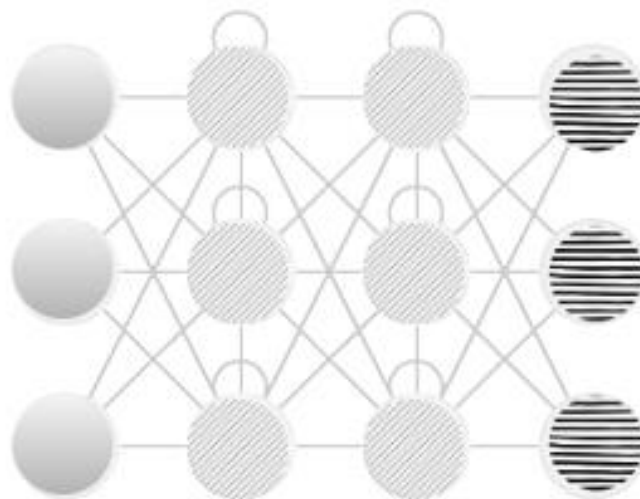


Рисунок 5 – Рекурсивная нейронная сеть [6]

Можно выделить такие плюсы архитектуры: возможность реализовать любую вычислимую функцию, обладание «памятью». Самый значимый минус – сложность обучения [6].

Выделяют следующие группы рекурсивных нейронных сетей:

рекуррентные;

LSTM;

генеративно-состязательные.

Рекуррентная нейронная сеть – вариант рекурсивной сети, связи между узлами в которой являются направленными циклами. Следовательно, что выход зависит не только от входа, а также от состояний нейрона на предыдущем шаге [7].

В отличие от классического вида архитектуры глубокого обучения, где используются на каждом слое различные весовые коэффициенты, рекуррентные сети используют одни и те же параметры. Суть в том, что выполняется одна и та же работа, но с разными входными данными.

LSTM (Long Short-Term Memory) включает в себя данные вне потока рекуррентной сети в закрытой ячейке. Информация хранится в нейроне, записывается или считывается. Узел решает: что хранить, и когда разрешить манипуляции с данными. LSTM созданы, чтобы избежать долговременной зависимости [9].

У генеративно-состязательной сети результат строится на состязании двух подсетей. Одна (генератор) создает данные, стараясь подать его как оригинальный и качественный, вторая (дискриминатор) определяет как хорошо справляется первая.

Таким образом генератор получает переменные на вход, а данные на вход. Дискриминатор является двоичным классификатором, выводящий: 1 - для реальных данных; 0 - для поддельных данных.

## Выводы

В результате исследования рассмотрен принцип работы нейрона, а также искусственной нейронной сети в целом. Изучены основные типы архитектур нейронных сетей, проанализированы их особенности, преимущества и недостатки.

Большинство нейронных сетей подразумевает многослойность, большое количество данных, поэтому необходимо понимать в каком случае какую архитектуру выбрать для реализации сети.

В ходе изучения для дальнейшего исследования темы нейронных сетей для генерации текста необходимо более детально обратить внимание на рекурсивные нейронные сети, так как им свойственна эта задача, они имеют «память», также важное преимущество над другими - зависимость конечного результата не только от входных данных, но и промежуточных.

## Литература

1. Фаустова К.И. Нейронные сети: применение сегодня и развитие завтра // Территория науки. - 2017 - №4.
2. Демкин В. И. История и перспективы развития нейронных сетей / В.И. Демкин, Д. К.Луков // Вестник современных исследований. - 2018. - № 6.1 (21).
3. Нейрон – строение, классификация, функции и свойства [Электронный ресурс] – Режим доступа: <https://nauka.club/anatomy/nejron.html>. - Загл. с экрана.

4. Сравнение мозга с нейронной сетью [Электронный ресурс] – Режим доступа: [https://ai-news.ru/2020/06/sravnenie\\_mozga\\_s\\_nejronnoj\\_setu.html](https://ai-news.ru/2020/06/sravnenie_mozga_s_nejronnoj_setu.html). - Загл. с экрана.
5. Клевцов Д.В. Перспективы использования нейронных сетей в современной экономике // Международный журнал прикладных наук и технологий «Integral», 2019, №1.
6. Введение в архитектуру нейронных сетей [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/oleg-bunin/blog/340184/>. - Загл. с экрана.
7. 7 архитектур нейронных сетей для решения задач NLP [Электронный ресурс] – Режим доступа: [https://ai-news.ru/2018/10/7\\_arhitektur\\_nejronnyh\\_setej\\_dlya\\_resheniya\\_zadach\\_nlp.html](https://ai-news.ru/2018/10/7_arhitektur_nejronnyh_setej_dlya_resheniya_zadach_nlp.html). - Загл. с экрана.
8. Маршалко Д.А. Архитектура свёрточных нейронных сетей / Маршалко Д.А., Кубанских О.В. // Ученые записки Брянского государственного университета. – 2019. - №4.
9. Гринин И.Л. Работа модели генерации текста с помощью нейронных сетей как составной системы: модульный анализ. Модуль третий. Нейронные сети - структура и машинное обучение // «Инновации и инвестиции», 2020, №9.
10. Никифорова К.В. Обзор типов нейронных сетей для архитектурной оптимизации // Сборник статей XLIX Международной научно-практической конференции WORLD SCIENCE: PROBLEMS AND INNOVATIONS. – 2020, г. Пенза.

***Я.Р. Парсаданян, А.В. Боднар Теоретический анализ принципа работы нейронных сетей.***

*В статье рассмотрен принцип работы нейронной сети. Изучены актуальные виды архитектур (полносвязная прямого распространения, сверточная, рекурсивная), проанализированы их особенности, преимущества и недостатки. На основе данного анализа сформулирован вывод о дальнейших исследованиях.*

***Ключевые слова:*** Нейронная сеть, нейрон, архитектура, вес, информация, рекурсивная нейронная сеть, сверточная нейронная сеть, полносвязная нейросеть.

***Y.R. Parsadanyan, A.V. Bodnar Theoretical analysis of the principle of neural networks. The article discusses the principle of operation of a neural network. The actual types of architectures (fully connected direct propagation, convolutional, recursive) are studied, their features, advantages and disadvantages are analyzed. Based on this analysis, a conclusion about further research is formulated.***

***Keywords:*** Neural network, neuron, architecture, VS, information, recursive neural network, convolutional neural network, fully connected neural network.

## Управление транспортным средством с помощью нейронных сетей

К.Н. Ефименко<sup>\*1</sup>, И.В. Бабич<sup>\*2</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
kn\_efimenko@mail.ru

<sup>\*2</sup> бакалавр, Донецкий национальный технический университет,  
babichivanvictorovich@yandex.ru

*Ефименко К.Н., Бабич И.В. Управление транспортным средством с помощью нейронных сетей. Выполнено описание и анализ программного продукта: Нейронная сеть предназначена для автоматизации процесса управления транспортным средством. Обоснован выбор подхода к машинному обучению, а также входные и выходные данные.*

**Ключевые слова:** нейронная сеть, нейрон, синапс, машинное обучение, автомобиль.

### Введение

Нейросеть в машинном обучении – это математическая модель, которая работает по принципу нейронной сети живого организма. В отличие от нейросети животного, которая передаёт сигнал от мозга к другим органам и полностью регулирует жизнедеятельность организма, компьютерная нейросеть учится решать только ту задачу, которую ей ставит человек.

Суть работы нейронных сетей – смоделировать способ решения задачи, присущий людям. Использование нейронных сетей опирается на накопленный опыт в виде данных и подходит для решения задач, с которыми человечество уже знакомо.

Рассматриваемая идея состоит в том, чтобы получить автомобиль, который сам по себе избегает препятствий виртуального мира. Каждое мгновение автомобиль самостоятельно решает, как изменить свою скорость и направление движения исходя из окружающей обстановки. Для достижения большей реалистичности автомобиль должен воспринимать только ту информацию, которую воспринимал бы на его месте человек. При наличии реальных входных данных искусственный интеллект мог бы быть использован в реальном транспортном средстве и работать так же хорошо [1].

Целью данной работы является выполнение краткого описания принципа работы нейронных сетей в контексте управления транспортным средством.

### Описание нейронной сети

Каждое животное на Земле имеет нервную систему, обеспечивающую регуляцию деятельности всех систем организма и реакцию на изменение условий внутренней и внешней среды. Нервная система человека функционирует как единое целое с органами чувств и управляется головным мозгом. Человеческий мозг, одним из свойств которого является способность хранить информацию, состоит из биологических нейронных сетей. Органы чувств передают информацию о раздражителе нейронным сетям, а те в свою очередь обрабатывают ее, благодаря чему мы чувствуем тепло и холод, ветер, влагу, можем распознать людей, вещи, запомнить информацию и так до бесконечности.

Искусственный нейрон – это тот же биологический нейрон, но только сильно упрощенный. Ему необходим лишь алгоритм работы биологического нейрона, чтобы осуществлять задуманное – самообучение компьютерных систем.

### Принцип работы нейронной сети

Алгоритм работы нейронной сети в точности повторяет алгоритм работы биологических нейронов. Каждый нейрон из сети умеет обрабатывать сигнал, а поскольку их миллиарды, то входящая информация на входе преобразуется в правильный сигнал на выходе. Важно учитывать то, что алгоритм работы каждого нейрона одинаков, поэтому возникает вопрос: если все нейроны работают по одинаковому примеру, то почему нейросети могут выдавать разные ответы, а не один и тот же постоянно? Дело в том, что помимо самих нейронов существуют еще и синапсы. Синапс – это место соединения выхода одного нейрона и входа другого, при этом сигнал при прохождении через синапс либо усиливается, либо ослабляется. Каждый из биологических нейронов

обладает тысячами входов для других нейронов, а значит и синапсов также тысячи [1]. Например, у человека около 85 млрд. нейронов в головном мозге.

Каждый синапс индивидуален, он либо усиливает, либо ослабляет сигнал нейрона, проходящий через него. В течение жизни синапсы меняются, а значит, и изменяются проходящие через них сигналы. Именно совместная работа нейрона и синапса преобразуют входящие сигналы в верные решения на выходе [1].

С точки зрения кибернетики, синапс — это связь между двумя нейронами. У синапсов есть 1 параметр — вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть 3 нейрона, которые передают информацию следующему. Тогда у нас есть 3 веса, соответствующие каждому из этих нейронов. У того нейрона, у которого вес будет больше, та информация и будет доминирующей в следующем нейроне. На самом деле, совокупность весов нейронной сети или матрица весов – это своеобразный мозг всей системы. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат. Как выглядит математический вид нейрона показано на рисунке 1.

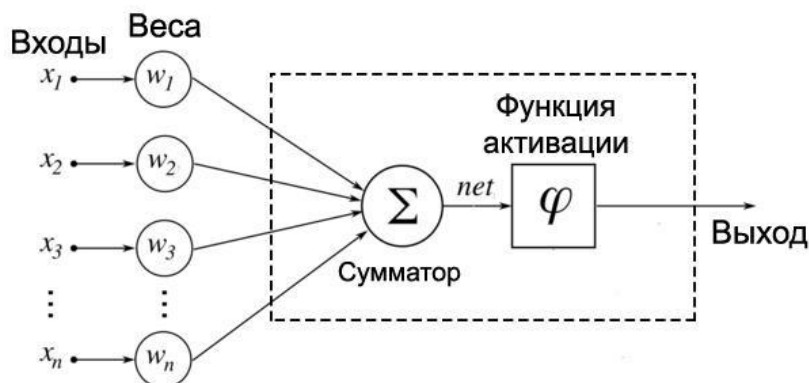


Рисунок 1 – Математический вид нейрона

Нейрон состоит из входов, на которые подаются числа, весов (те же синапсы) и сумматора. Поступающие на входы числа умножаются на соответствующие веса, после чего произведения суммируются. Сумма всех результатов не является окончательным решением и для выдачи готового результата используется функция активации. Функция активации вычисляет выходной сигнал сумматора в правильное решение [1].

Нейрон – это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной (на рисунке 2 обозначен синим цветом), скрытый (на рисунке 2 обозначен красным цветом) и выходной (на рисунке 2 обозначен зеленым цветом).

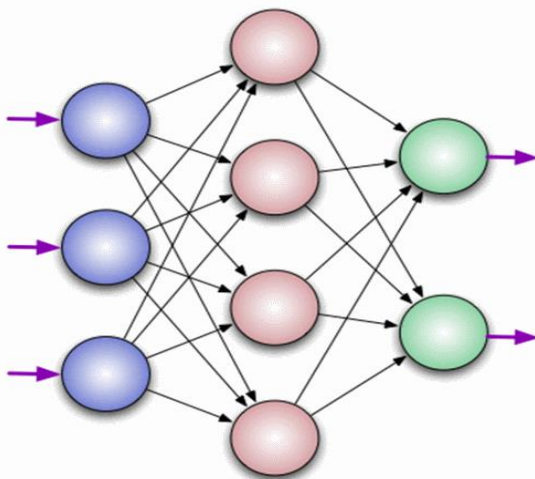


Рисунок 2 – Базовая нейронная сеть

В том случае, когда нейросеть состоит из большого количества нейронов, вводят термин слоя. Соответственно, есть входной слой, который получает информацию, какое-то количество скрытых слоев, которые ее обрабатывают и выходной слой, который выводит результат. У каждого из нейронов есть 2 основных параметра: входные данные и выходные данные.

## **Типы нейронных сетей**

Рассмотрим классификацию нейронных сетей и их основные типы.

Каждая сеть имеет первый слой нейронов, который называется входным. Он не выполняет никаких вычислений и преобразований, его задача состоит только в одном: принять и распределить по остальным нейронам входные сигналы. Это единственный слой, который является общим для всех типов нейронных сетей, дальнейшая их структура и является критерием для основного деления [1].

Однослойная нейронная сеть. Это структура взаимодействия нейронов, при которой после попадания входных данных в первый входной слой они сразу передаются в слой выхода конечного результата. При этом первый входной слой не считается, так как он не выполняет никаких действий, кроме приёма и распределения. Второй слой производит все нужные вычисления и обработки и сразу выдаёт конечный результат. Входные нейроны объединены с основным слоем синапсами, имеющими различный весовой коэффициент, обеспечивающий качество связей.

Многослойная нейронная сеть. Этот вид нейронных сетей помимо входного и выходного слоёв имеет ещё и промежуточные слои. Их количество зависит от степени сложности самой сети. Она в большей степени напоминает структуру биологической нейронной сети. Такие виды сетей были разработаны совсем недавно, до этого все процессы были реализованы с помощью однослойных сетей. Соответственно подобное решение имеет намного больше возможностей, чем её предшественник. В процессе обработки информации каждый промежуточный слой представляет собой промежуточный этап обработки и распределения информации.

В зависимости от направления распределения информации по синапсам от одного нейрона к другому, можно также классифицировать сети на две категории [1].

Сети прямого распространения или однонаправленная, то есть структура, в которой сигнал движется строго от входного слоя к выходному. Движение сигнала в обратном направлении невозможно. Подобные разработки достаточно широко распространены и в настоящий момент с успехом решают такие задачи, как распознавание, прогнозы или кластеризация.

Сети с обратными связями или рекуррентная. Подобные сети позволяют сигналу двигаться не только в прямом, но и в обратном направлении. В таких сетях результат выхода может возвращаться на вход исходя из этого, выход нейрона определяется весами и сигналами входа, и дополняется предыдущими выходами, которые снова вернулись на вход. Таким сетям свойственна функция кратковременной памяти, на основании которой сигналы восстанавливаются и дополняются в процессе обработки.

Обобщая сказанное можно сделать вывод, что каждая сеть имеет первый слой нейронов, который называется входным. Он не выполняет никаких вычислений и преобразований, его задача состоит только в одном: принять и распределить по остальным нейронам входные сигналы. Это единственный слой, который является общим для всех типов нейронных сетей, дальнейшая их структура и является критерием для основного деления [2].

В настоящее время известны следующие области применения нейронных сетей:

- распознавание образов и классификация;
- принятие решений и управление;
- кластеризация;
- прогнозирование;
- аппроксимация;
- сжатие данных и ассоциативная память;
- анализ данных;
- оптимизация.

В общем случае решение этих задач включает следующие этапы:

- сбор данных для обучения;
- выбор топологии сети;
- экспериментальный подбор характеристик сети;
- экспериментальный подбор параметров обучения;
- обучение сети;
- проверка адекватности обучения.

## **Проект нейронной сети, обучающейся управлению транспортным средством**

В современном мире искусственные нейронные сети широко используются при решении самых разнообразных задач особенно там, где обычные алгоритмические решения оказываются неэффективными или вовсе невозможными.

Для данного проекта была выбрана модель машинного обучения «Обучение без учителя», когда машина сама должна найти среди хаотичных данных верное решение (рис. 3). Потому что мы имеем дело с компьютерной симуляцией, в которой неверные действия нейронной сети не принесут вреда, поэтому у нейросети есть множество попыток для обучения и для них не нужна обширная база данных с входной информацией.

Для ориентирования в пространстве нейронной сети будут доступны несколько векторов, которые формируют угол обзора водителя, и расстояние, на котором вектор пересекается с другим объектом (препятствием), как входные данные.

Выходные данные представляют собой реакцию нейросети на возникновение препятствий, т.е. увеличение или снижение скорости и поворот влево или вправо.



Рисунок 3 – Схема обучения нейронной сети без учителя

### **Заключение**

Искусственные нейронные сети получили большое распространение при решении многих задач. Вместе с тем изучение большинства их возможностей находится в экспериментальной стадии. Технологии нейронных сетей не должны рассматриваться как универсальное средство решения всех интеллектуальных задач. Их применение оправдано в тех областях, в которых существует значительное число однотипных примеров, отражающих скрытые взаимосвязи [3].

Принципиальное отличие искусственных нейросетей от обычных программных систем, состоит в том, что они не требуют программирования. Они сами настраиваются, т.е. обучаются тому, что требуется пользователю.

В данной работе было рассмотрено и описано устройство нейронной сети, нейрона, а также типы нейронных сетей. Выбран и аргументирован подход машинного обучения для дальнейшей разработки системы управления транспортным средством.

### **Литература**

1. Портал habr.com [Электронный ресурс] / Режим доступа: <https://habr.com/ru/post/101789/>. – Загл. с экрана.
2. Большая российская энциклопедия [Электронный ресурс] / Режим доступа: [https://bigenc.ru/technology\\_and\\_technique/text/4114009](https://bigenc.ru/technology_and_technique/text/4114009). – Загл. с экрана.
3. РБК [Электронный ресурс] / Режим доступа: <https://trends.rbc.ru/trends/industry/60c85c599a7947f5776ad409>. – Загл. с экрана.

**Ефименко К.Н., Бабич И.В. Управление транспортным средством с помощью нейронных сетей.** Выполнено описание и анализ программного продукта: Нейронная сеть предназначена для автоматизации процесса управления транспортным средством. Обоснован выбор подхода к машинному обучению, а также входные и выходные данные.

**Ключевые слова:** нейронная сеть, нейрон, синапс, машинное обучение, автомобиль.

**Efimenko K.N., Babich I.V. Vehicle control using neural networks.** The description and analysis of the software product is carried out: The neural network is designed to automate the process of driving a vehicle. The choice of an approach to machine learning, as well as input and output data, is justified.

**Keywords:** neural network, neuron, synapse, machine learning, car.



## Применения мультиагентных систем для управления группой беспилотных летательных аппаратов

Н.В. Мелешенко<sup>\*1</sup>, О.И. Федяев<sup>\*2</sup>

<sup>\*1</sup> магистрант, Донецкий национальный технический университет,  
meleshenko.nikolay@gmail.com

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
fedyaev@donntu.ru, OrcID: 0000-0001-6822-7306

*Мелешенко Н.В., Федяев О.И. Применения мультиагентных систем для управления группой беспилотных летательных аппаратов. В статье рассмотрена технология мультиагентных систем, ее текущее состояние и перспективы. Исследована целесообразность применения мультиагентного подхода для управления группой беспилотных летательных аппаратов. В качестве практического примера рассмотрено применение мультиагентного подхода для управления группой беспилотных летательных аппаратов при поиске цели на заданной территории.*

**Ключевые слова:** мультиагентная система, роевой интеллект, распределённый искусственный интеллект, группа беспилотных летательных аппаратов, система управления.

### Введение

В последнее время роль беспилотных летательных аппаратов (БПЛА) все больше возрастает как в военной, так и в гражданской области. Также является очевидным, что групповое применение БПЛА является более эффективным по сравнению с одиночным, так как уменьшается шанс ошибки и расширяется зона применения. Но в тоже время такой подход требует решение проблемы управления группой БПЛА как единым целым. Такая группа, в силу специфики, должна обладать такими ключевыми свойствами как автономность и самоорганизация, чтобы координировать свои действия и адаптироваться к любой воздушной обстановке даже если связь с оператором потеряна. Таким образом, создание полностью или частично автономной системы управления группой БПЛА является актуальной научной задачей. Целью работы является обоснование мультиагентного подхода при реализации системы управления группой легких БПЛА для поиска цели на заданной территории. Для достижения данной цели необходимо исследовать особенности мультиагентной технологии, проанализировать опыт практического применения, выявить перспективные направления для применения и спроектировать систему управления при помощи данной технологии.

### Особенности МАС

Прежде всего необходимо определить главные понятия, затрагиваемые в данной работе. В основе МАС лежит понятие агента. Проанализировав работы [1-4] можно сформировать понятия «агента» и «мультиагентной системы».

Агентом принято называть автономную программную сущность, которая может анализировать внешнюю среду, в которой она находится, и воздействовать на неё. Агенты также могут общаться между собой путём обмена сообщениями. Важной способностью агента является целенаправленное поведение, которое может являться следствием его знаний и/или ментального состояния, то есть его убеждений, желаний и намерений.

В работе [5] приводятся главные отличия объектов в объектно-ориентированном подходе от агентов. Агент прежде всего обладает свойством автономности, то есть он может взять на себя какие-либо обязательства или отказаться от выполнения задания из-за наличия другой работы или недостатка знаний. Также агент может порождать других агентов или заменять их. Объект не может анализировать свою связь с другими объектами или своё к ним отношение.

Мультиагентная система с позиции программной инженерии представляет собой программный комплекс, который включает в себя множество агентов и среду, в которой эти агенты взаимодействуют. У агентов в МАС могут быть различные типы, различные цели и различные программные реализации, также агенты могут находиться на различных физических устройствах, соединённых компьютерной сетью.

Главной идеей МАС является получение качественно новых вычислительных способностей на основе взаимодействия агентов [1,6]. Взаимодействие агентов регулируется при помощи протоколов взаимодействия.

Цели взаимодействия агентов в МАС определяют модели совместного поведения агентов. Главным достоинством МАС является умение организовать взаимодействие компонентов программы между собой, что является одной из самых трудных задач в сложных распределённых программных комплексах. Таким образом, главной проблемой при построении МАС является кооперация агентов в достижении общей цели и решение конфликтов между агентами.

### **Опыт практического применения МАС**

За более чем сорокалетнюю историю МАС-подход проявил себя лучше других подходов в области управления производством, в телекоммуникации, в электронной коммерции, в производственной и транспортной логистике, в области аэрокосмических приложений и в энергетике [6]. Это можно объяснить потребностью автономно в реальном времени управлять ресурсами и объектами управления, оптимизировать показатели и обучаться.

Стоит отметить, что у МАС есть серьезные конкуренты, которые превосходят МАС по отдельным областям применения, для которых, собственно, они и создавались. На практике оказалось, что многие перспективные для МАС приложения были реализованы при помощи узконаправленных технологий, таких как сервис-ориентированные технологии, грид-вычисления, облачные вычисления и другие. Хотя данные технологии появились позже МАС, они смогли вытеснить МАС-технологии из рынка промышленных приложений [6]. Следует сделать вывод, что МАС-подход привлекал внимание отдельными аспектами своей технологии, а не всей ее совокупностью как таковой, поэтому МАС-технологии не выдержали конкуренции с теми технологиями, по части которых они проигрывали.

В настоящее время активность практических разработок в области МАС не оправдывает ожидания. В качестве главных причин можно выделить следующие: отсутствие четкой терминологии МАС, отсутствие общепринятой нотации для МАС, сложность формализации ментальной модели агента и МАС, недостатки существующих стандартов, отсутствие гибких программных инструментов и методик для разработки МАС-приложений, неудачные попытки применения МАС в прикладных областях [1]. Из вышесказанного можно сделать вывод, что практика показала МАС как формирующуюся технологию, многие аспекты которой в теории казались весьма привлекательными, но в реальности все обстоит иначе. Но в ближайшем будущем эта технология может найти свое применение, благодаря появлению новых, быстроразвивающихся интернет-направлений.

### **Перспективные направления для МАС**

Главным потенциалом МАС-технологии является создание автономных самоорганизующихся сложных систем, которые производят вычисления на основе взаимодействий компонентов системы (агентов) в реальном времени [1]. Таким образом, можно значительно сузить спектр применения МАС и полностью сформировать и специфицировать технологию. Практика показала, что не следует пытаться применять МАС как базовую технологию, например, как ООП, ведь уровень текущего развития МАС-технологии пока не позволяет этого и вряд ли в скором будущем это будет возможно.

Сегодня еще продолжается поиск специализированной ниши для применения МАС, в которой она будет практически безальтернативной, и с каждым годом таких ниш становится все больше. Потенциал МАС-технологии позволяет видеть эту технологию в современных облачных сервисах, интернете вещей, цифровых экосистемах, коллективной робототехнике, человеко-машинных комплексах и в задачах современной цифровой экономики реального времени [6].

В [1] приводятся такие наиболее перспективные прикладные области для применения МАС: аэрокосмическая отрасль, В2В-сети предприятий, военные приложения, коллективная робототехника, смарт-грид, здравоохранение, интернет вещей, мобильные приложения.

Отдельно выделим наиболее перспективные и интересные, с нашей точки зрения, направления: аэрокосмическая отрасль, коллективная робототехника и смарт-грид.

Аэрокосмическая отрасль непрерывно развивается и создает ниши для приложений самых разных технологий. Для МАС созданы практически идеальные ниши: управление группой беспилотных космических и/или летательных аппаратов (БПКА и БПЛА), управление группировками малых спутников, управление авиапотоками и космическая логистика [1].

В коллективной робототехнике одной из главных проблем является проблема управления группой роботов и организации взаимодействия между ними. Группа включает в себя автономных роботов, которые взаимодействуют между собой в некой среде для решения общей задачи, при этом они должны реагировать на изменения этой среды [7]. В такой формулировке задача коллективной робототехники не отличается от задачи МАС, только вместо агентов используется физическое воплощение агента — робот.

МАС представляет большой интерес для моделирования и исследований в области смарт-грид, интегрированных энергетических систем и других приложений для энергетики, так как данный подход позволяет детально исследовать процессы взаимодействия и кооперации между элементами энергетической системы [8]. МАС-подход может повысить надежность сети и в реальном времени распределять нагрузку между

критическими элементами сети, а также вести постоянный мониторинг оборудования и в случае сбоя провести немедленную автономную перезагрузку сети.

Данные направления объединяет то, что совсем недавно такие возможности казались достижимыми в далеком будущем, но, возможно, уже совсем скоро они станут обыденностью при помощи МАС-технологии.

### **Применение МАС для управление группой автономных БПЛА**

Как было сказано выше, аэрокосмическая область является одной из самых перспективных для применения МАС-технологий. Сегодня БПЛА плотно вошли в повседневную жизнь, так как нашли обширный пласт применения в военной и гражданской сферах, например, аэросъемка, логистика, зондирование поверхности Земли, видеонаблюдение, поиск, мониторинг биологической, радиационной и экологической обстановки. Но возможности одного БПЛА ограничены, как и одного человека, появляется необходимость использования группы БПЛА для более эффективного решения задач, например, увеличение зоны покрытия при зондировании, мониторинга или поиска. При использовании групп таких аппаратов неизбежно возникает проблема управления ими. С управлением одним БПЛА справится один человек в ручном режиме, но если же БПЛА будет 100 или 500? В этом случае управление в ручном режиме может оказаться неэффективным и очень трудозатратным, поэтому, на наш взгляд, такая группа аппаратов должна обладать автономностью, чтобы адаптироваться к изменениям окружающей среды и координируясь между собой принимать решение в сложившихся обстоятельствах. В работах [Ошибка! Источник ссылки не найден.—Ошибка! Источник ссылки не найден.] обосновывается необходимость применения МАС-технологий для организации системы управления группой автономных БПЛА.

Следует сделать важное замечание по поводу аспектов применения МАС. Если задача ставится как управление скоординированным полетом группы автономных БПЛА в сложных условиях, то в данном случае, на наш взгляд, необходимо рассматривать данную группу как рой и применять алгоритмы роевого интеллекта и искусственной жизни, в частности алгоритм «Boids», предложенный Крейгом Рейнольдсом в его статье «Flocks, Herds, and Schools: A Distributed Behavioral Model» в 1987 году. Данный алгоритм имитирует поведения стаи птиц. Модифицировав и применив данный алгоритм к группе БПЛА, можно добиться скоординированного полета группы БПЛА как стаи птиц, которая может облетать препятствия и держаться заданным строем. При такой формулировке МАС, система будет состоять из множества независимых друг от друга агентов без интеллекта, которые придерживаются небольшого набора простых правил (разделение, выравнивание и сплочённость) и чьи локальные взаимодействия будут порождать так называемое эмерджентное поведения стаи, то есть такое поведение, которым не обладает ни одна особь из стаи.

Однако, если задача ставится как управление автономной группой БПЛА при выполнении некой задачи, например, разведка местности, поделённой на квадраты наблюдения, тогда, на наш взгляд, следует использовать МАС как систему распределённого искусственного интеллекта, в частности, как систему распределённого решения задач. В такой формулировке, система будет состоять преимущественно из интеллектуальных агентов, чье коллективное поведение на основе ментального состояния и знаний будет определять поведение всей системы.

Подводя итог вышесказанного, можем согласиться с автором в работе [Ошибка! Источник ссылки не найден.], в которой выделяются такие главные направления разработки МАС как распределенный искусственный интеллект и искусственная жизнь. Мы считаем, что при практическом применении МАС при управлении группой БПЛА можно совместить два этих подхода. Использовать роевой интеллект при управлении полетом группы БПЛА и распределённое решение задач при выполнении миссии группы.

Рассмотрим на примере решение задачи управления автономной группой легких БПЛА типа мультикоптер при выполнении задачи поиска цели на указанной территории, для этого применим подход распределённого решения задач.

Сначала выполним постановку задачи. Необходимо разработать многоагентную модель управления группы легких БПЛА, которая способна выполнять поиск на заданной территории при помощи аэросъемки. Входными данными являются:

- территория полета, поделенная на квадраты,  $w$  квадратов по ширине и  $h$  по высоте, таким образом количество квадратов равняется  $K=w*h$  —  $T=\{T_{1,1}, T_{1,2}, \dots, T_{w,h}\}$ ;
- множество БПЛА мощностью  $m$  —  $V=\{V_1, V_2, \dots, V_m\}$ , которые оборудованы видеокамерами, системой навигации, модулем для приема и отправки данных беспроводным способом;
- цель для поиска —  $P$ ;
- таблица коэффициентов приоритетности квадратов —  $H=\{H_{1,1}, H_{1,2}, \dots, H_{w,h}\}$ ;
- оператор группы —  $O$ , его управление ограничивается заданием базовых команд группе, например, взлет, посадка, задание цели и коэффициентов приоритетности.

Выходными данными будет служить квадрат с найденной целью —  $T_{i,j}$ .

Следовательно, если применить систему управления  $S$  к входным параметрам, то получим квадрат, в котором находится цель:

$$S(T, V, P, H, O)=T_{i,j}.$$

Теперь перейдем к описанию структуры системы. Средой для агентов выступает территория полета. Датчиками для анализа среды выступает камера БПЛА. При распределённом решении задач необходимо выделить агента-менеджера, который будет распределять задачи для других агентов, назовём его «агент-командир». Все взаимодействия с группой происходит через командира, то есть оператор передает ему базовые команды, а он в свою очередь доводит до сведения подчиненных агентов, так мы уменьшаем поток данных между оператором и агентами, а также исключаем беспорядок между агентами. Подводя итог, можно сформулировать типы агентов в нашей модели, это агент-дрон, агент-командир и агент-оператор (см. рис. 1).

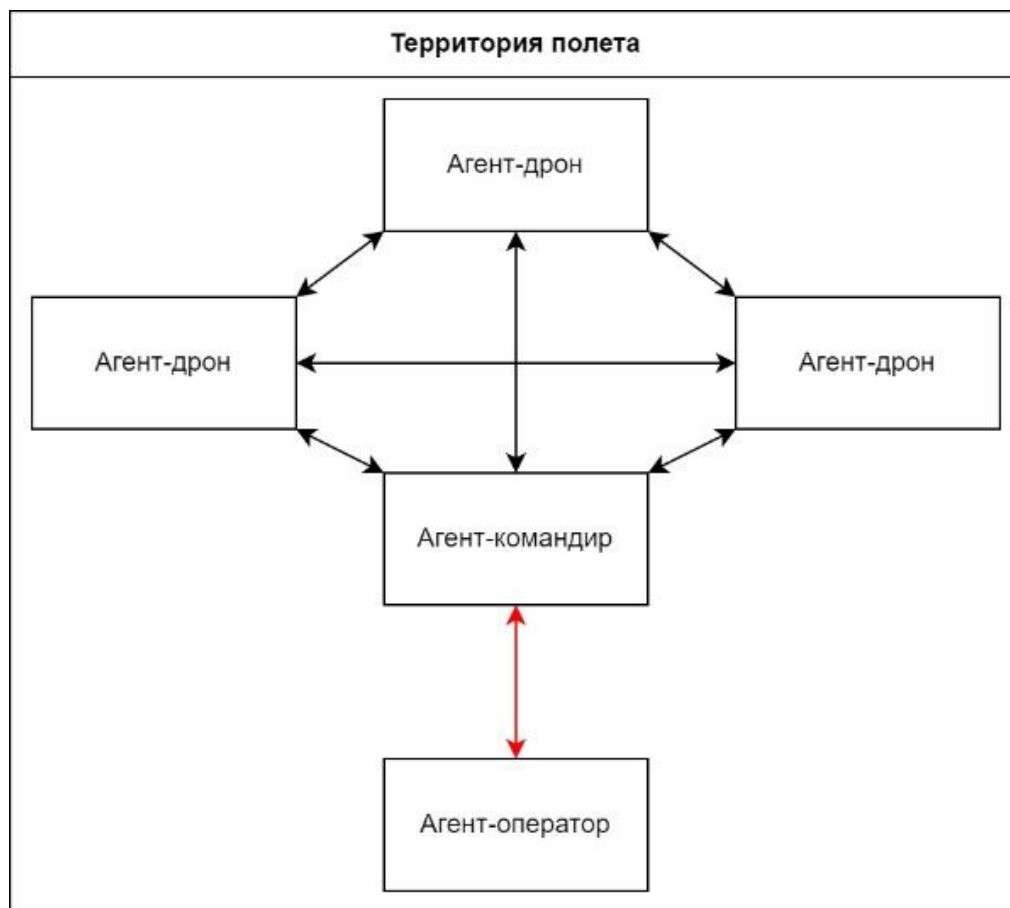


Рисунок 1 — Архитектура МАС для управления группой автономных БПЛА

Сообщения оператора, которые отмечены красной линией, сводятся к передаче базовых команд группе и контролю за их выполнением. Переговоры агентов между собой отмечены черными линиями.

Подробнее опишем роли агентов. Агент-оператор действует в интересах оператора системы, он управляется человеком напрямую. Он должен всегда знать кто является командиром группы, чтобы передать ему базовые команды. Базовые команды имеют наивысший приоритет для всех агентов. Агент-дрон следует по траектории, которая зависит от текущего его задания, выданного командиром. Он анализирует внешнюю среду при помощи камеры, одновременно с этим он обменивается с другими дронами информацией о найденной цели или об ее отсутствии. Агент-командир расширяет роль агента-дрона. Он должен всегда докладывать агенту-оператору о ходе выполнения поиска. Агент-командир выполняет декомпозицию главной задачи на подзадачи и выдает их агентам-дронам.

Для более детального пояснения приведем упрощенный пример истории переговоров агентов для группы из одного агента-командира и двух агентов-дронов во время поиска цели (см. рис. 2).

Сообщение	Отправитель	Получатель
Начинаю исследование квадрата (1, 5)	Дрон1	Все
Начинаю исследование квадрата (2, 5)	Командир	Все
Начинаю исследование квадрата (3, 5)	Дрон2	Все
Исследуются квадраты (1, 5), (2, 5), (3, 5)	Командир	Оператор
Квадрат (1, 5) чист	Дрон1	Все
Начинаю исследование квадрата (1, 6)	Дрон1	Все
Квадрат (2, 5) чист	Командир	Все
Начинаю исследование квадрата (2, 6)	Командир	Все
Цель обнаружена в квадрате (3, 5)	Дрон2	Все
Всем, прекратить поиск	Командир	Все
Цель обнаружена в квадрате (3, 5)	Командир	Оператор
Цель обнаружена в квадрате (1, 6)	Дрон1	Все
Всем, проверить квадрат (1, 6)	Командир	Все
Еще одна цель обнаружена в квадрате (1, 6)	Командир	Оператор
Проверяется квадрат (1, 6)	Командир	Оператор

Рисунок 2 — История переговоров агентов

После первого обнаружения цели командир дает задание прекратить поиск, но Дрон1 обнаруживает еще одну цель, он считает, что объектов цели сразу два. После его сообщения Командир даёт задание перепроверить квадрат (1, 6), чтобы удостовериться или опровергнуть утверждение Дрон1.

Таким образом, рассмотрена МАС-архитектура для системы управления группой автономных БПЛА, способных выполнять поиск на заданной территории при помощи аэросъемки. Переговоры агентов основаны на передаче простых сообщений, что уменьшает вычислительную сложность при их анализе и повышает эффективность МАС-реализации, но в тоже время позволяет сохранить все преимущества МАС-подхода, такие как автономность и самоорганизация в реальном времени.

### Заключение

В работе были рассмотрены особенности МАС-технологии, её суть, преимущества и недостатки. Выделены направления практического применения, в которых МАС-технология проявила себя лучше других подходов. Также выяснены причины, из-за которых МАС не всегда применяется на практике. Сформулирован главный потенциал МАС. Выявлены перспективные направления для практического применения МАС-технологии и основные направления разработок в МАС. Была исследована возможность применения МАС-технологии для управления группой автономных БПЛА.

### Литература

1. Городецкий, В. И. Современное состояние и перспективы индустриальных применений многоагентных систем / В. И. Городецкий, О. Л. Бухвалов, П. О. Скобелев // УБС. 2017. №66.
2. Михайлов, И. Ф. Виртуальные проекции человеческого мира: мультиагентные системы // Философские проблемы информационных технологий и киберпространства. 2017. №1 (13).
3. Кузнецов, А. В. Краткий обзор многоагентных моделей // УБС. 2018. №71.
4. Многоагентные системы (обзор) [Электронный ресурс] / В. И. Городецкий, М. С. Грушинский, А. В. Хабалов // Сети, когнитивная наука, управление сложностью. — Режим доступа: <https://spkurdyumov.ru/networks/mnogoagentnye-sistemy-obzor>. — Загл. с экрана.
5. Тарасов, В. Б. Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // Новости искусственного интеллекта. — 1998. — № 2. — С. 55-63.
6. Городецкий, В. И. Многоагентные технологии для индустриальных приложений: реальность и перспектива / В. И. Городецкий, П. О. Скобелев // Тр. СПИИРАН, 55 (2017), 11–45.
7. Каляев, И. А. Самоорганизующиеся распределенные системы управления группами интеллектуальных роботов, построенные на основе сетевой модели / И. А. Каляев, С. Г. Капустян, А. Р. Гайдук // УБС. 2010. №30-1.

8. Стенников, В. А. Применение мультиагентного подхода для моделирования интегрированных энергетических систем / В. А. Стенников, Е. А. Барахтенко, Г. С. Майоров // Известия вузов. Проблемы энергетики. 2020. №6.

9. Адаптивное управление автономной группой беспилотных летательных аппаратов [Электронный ресурс] / К. С. Амелин, Е. И. Антал, В. И. Васильев, Н. О. Граничина. — Режим доступа: <https://math.spbu.ru/user/gran/soi5/Amelin5.pdf>. — Загл. с экрана.

10. Довгаль, В. А. Интеграция сетей и вычислений для построения системы управления роем дронов как сетевой системы управления // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2022. №1 (296).

11. Халимов, Н. Р. Распределенная сетевая система управления группой ударных беспилотных летательных аппаратов / Н. Р. Халимов, А. В. Мефедов // Системы управления, связи и безопасности. 2019. № 3. С. 1-13. DOI: 10.24411/2410-9916-2019-10301.

**Мелещенко Н.В., Федяев О.И. Применения мультиагентных систем для управления группой беспилотных летательных аппаратов.** В статье рассмотрена технология мультиагентных систем, ее текущее состояние и перспективы. Исследована целесообразность применения мультиагентного подхода для управления группой беспилотных летательных аппаратов. В качестве практического примера рассмотрено применение мультиагентного подхода для управления группой беспилотных летательных аппаратов при поиске цели на заданной территории.

**Ключевые слова:** мультиагентная система, роевой интеллект, распределённый искусственный интеллект, группа беспилотных летательных аппаратов, система управления.

**Meleshchenko N., Fedyaev O. Applications of multi-agent systems for controlling a group of unmanned aerial vehicles.** The article discusses the technology of multi-agent systems, its current state and prospects. The expediency of using a multi-agent approach to control a group of unmanned aerial vehicles investigated. As a practical example, the application of a multi-agent approach to control a group of unmanned aerial vehicles when searching for a target in a given territory considered.

**Keywords:** multi-agent system, swarm intelligence, distributed artificial intelligence, group of unmanned aerial vehicles, control system.

## Сайт как элемент продвижения образовательных услуг

В.Р. Юрьева<sup>\*1</sup>, И.Ю. Анохина<sup>\*2</sup>

<sup>\*1</sup> бакалавр, Донецкий национальный технический университет, notorious404@mail.ru,

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
OrcID: 0000-0003-1073-6333, SPIN-код: 3555-6335  
ingatula@mail.ru

*Юрьева В.Р., Анохина И.Ю. Сайт как элемент продвижения образовательных услуг. В статье представлен краткий обзор современных методов разработки web – ресурсов. Проанализированы требования, предъявляемые к сайтам кафедр и образовательным порталам учебных заведений. Проведенные исследования позволили сформулировать требования и принять решение об оптимальной структуре учебных web – ресурсов.*

**Ключевые слова:** сайт кафедры, продвижение образовательных услуг, система создания и управления сайтом.

### Введение

Повышение конкуренции среди вузов, уменьшение числа абитуриентов, необходимость интернет-продвижения образовательных услуг, предоставляемых как в целом вузом, так и отдельными его структурными подразделениями, в частности, кафедрами, обусловили необходимость разработки концепции сайта образовательной структуры, а также выбор метода разработки сайта, от которого во многом будет зависеть его продвижение в интернете и возможности реализации поставленных задач.

На сайтах большинства российских вузов проводится анкетирование, одним из направлений анкетирования является определение, откуда получена информация о вузе, что или кто позволил определиться с выбором вуза и будущей специальностью. Как показали исследования, большинство абитуриентов получают информацию о вузе из интернета, посещая как сайт вуза, так и его кафедр. При этом многие респонденты указывали на проблемы с поиском информации на сайтах вузов, недостаточность информации и пр. [1].

Анализ результатов анкетирования с одной стороны позволяет утверждать, что интернет как канал продвижения вуза играет существенную роль в его позиционировании на рынке образовательных услуг, а с другой стороны ставит задачу разработки требований к структуре и информации, размещаемой на сайте вуза, кафедры.

### Постановка задачи

Поставлена задача провести сравнительный анализ сайтов вузов г. Донецка и вузов, входящих в Топ- 2022 России, оценить порталы кафедр различных вузов, а также проанализировать методы разработки web – сайтов с целью определения оптимальной структуры и компонентов портала кафедры как консалтинг-проекта продвижения образовательных услуг.

### Описание алгоритма

На рынке высшего образования продвижение вуза и предоставляемых им образовательных услуг стало неизменной частью политики любого учебного заведения. Тенденции в российском образовании, обусловленные дальнейшим развитием интернет-технологий, появлением новых видов образовательных услуг и в то же время рост рисков в системе образования все более повышают необходимость развития интернет-представительств не только вузов, но и других структурных единиц, таких кафедры, лаборатории и пр.

Нами проанализированы сайты ведущих вузов Донецка. Они анализировались отдельно от остальных вузов России с учетом сложившейся ситуации, которая ставит ситуацию с образованием в Донецкой Народной Республике в особые условия.

В табл. 1 приведены анализируемые данные. В первой строке таблицы указаны адреса следующих учебных заведений:

Dnmtu.ru - Донецкий национальный медицинский университет; donnu.ru/ - Донецкий национальный университет; donntu.ru – Донецкий национальный технический университет; donnuet.ru - Донецкий национальный университет экономики и торговли; donnasa.ru - Донбасская национальная академия строительства и архитектуры; donatpra.ru - Донецкая академии управления и государственной службы.

В качестве параметров анализа использованы стандартные характеристики, описывающие трафик сайта. В строке «Всего визитов» приводятся данные о числе посещений за сентябрь 2022г. по данным сайта *Spy Metrics. Анализ конкурентов* [2]. Также в таблице указано, сколько в среднем посетитель проводит времени на сайте и сколько страниц просматривает за визит. «Показатель отказов» показывает процент посетителей, которые просмотрели только одну страницу и не выполнили переход ни по одной ссылке.

«Позиция в мире» характеризует рейтинг сайта, чем меньше число, тем выше рейтинг. Отметим, что по данным большинства сайтов аналитики на первом месте в мире находится сайт Google.com. Строки «Позиция в стране» и «Позиция в категории» показывают рейтинг сайтов как в стране (для всех наших сайтов рейтинг на Украине), а также в категории – Наука и образование.

Следующие три строки описывают структуру трафика. Строка «Динамика за полгода» дает информацию о росте или убывании числа посетителей сайта: негативная отмечена знаком минус, положительная – плюс и нейтральная + -.

Как показали данные, наиболее популярным является сайт Медицинского университета, наименее - Академии управления и государственной службы, однако, именно на этом сайте посетитель проводит более пятнадцати минут, в отличие, например, от сайта Донецкого национального университета (менее четырех минут).

Таблица 1 – Вузы Донецка. Характеристики трафика

	dnmu.ru	donnu.ru	donntu.ru	donnuet.ru	donnasa.ru	donampa.ru
Всего визитов	273140	63890	39700	31140	14300	8070
Время на сайте	0:04:33	0:03:56	0:02:25	0:15:19		0:15:19
Страниц за визит	5.88	3.77	4.10	8.73		8.73
Показатель отказов	0.28	0.52	0.40	0.15		0.15
Позиция в мире	83268	585189	854720	897137	1810513	2526289
Позиция в стране	1929	15194	23060	28505	44963	58053
Позиция в категории	1310	2472	3487		6674	10294
Трафик прямой	52.18%	51.97	71.60%	58.82%	50.77%	56.18%
Трафик «Соцсети»	11.59%	2.97%	7.62%	21.63%	2.72%	2.40%
Трафик «Поиск»	30.78%	42.36%	19.51%	17.36%	45.16%	35.70%
Динамика за полгода	-	-	+	+	+ -	-
CMS сайта	Wordpress	Drupal	Drupal	Joomla	Wordpress	Joomla

В последней строке таблицы указаны системы CMS (англ. Content Management System— система создания и управления сайтом), на которых созданы сайты вузов.

Т.к. в конечном итоге ставится задача разработки электронного портала кафедры, были проанализированы различные варианты разработки сайтов, а также составляющие их наполнения.

Для сравнения рассматривались сайты нескольких вузов России, входящих в десятку лучших по данным 2022г. [3].

Сайты Московского государственного университета имени М.В. Ломоносова (Msu.ru) и Московского физико-технического института (Mipt.ru) разработаны на Битрикс; Национального исследовательского университета а «Высшая школа экономики (Hse.ru) на Wordpress; Санкт-Петербургского государственного университета (spbu.ru) на Drupal. Нельзя не отметить короткие адреса сайтов в отличие от ранее рассмотренных сайтов донецких вузов.

Разработка сайтов с использованием CMS является одним из наиболее популярных вариантов. На рис.1. показан рейтинг систем CMS за 2021г. [4]. В зависимости от вида проводимого рейтинга и анализируемых свойств систем, Wordpress и 1С-Битрикс часто меняются местами[5]. И, тем не менее, Wordpress, как правило, лидирует среди бесплатных CMS. При анализе систем, как правило рассматриваются следующие характеристики: уровень защиты от взломов, удобство использования, пригодность для SEO продвижения, возможности создание интернет - магазина, скорость работы, цена. В отличие от 1С-Битрикс Wordpress позволяет создавать сайт бесплатно, хотя разработано множество платных дополнений и функционалов[6].

К другим методам разработки веб-ресурсов можно отнести использование SaaS-платформы. SaaS (Software as a service – программное обеспечение как услуга) — облачная технология, позволяющая использовать



программное обеспечение онлайн. Некоторые из этих платформ подходят исключительно для простых сайтов, а некоторые – достаточно универсальны. Среди них такие сервисы как WIX, InSales, Shopify, uCoz.

Оформление сайтов на таких платформах происходит с помощью готовых шаблонов, которые можно изменять в дизайне, но не в функционале. Программный продукт типичен и его настройка под индивидуальные пожелания ограничена. Если проект сразу или в перспективе должен решать специфические задачи и гибко настраиваться, SaaS-платформа для его разработки не подходит.

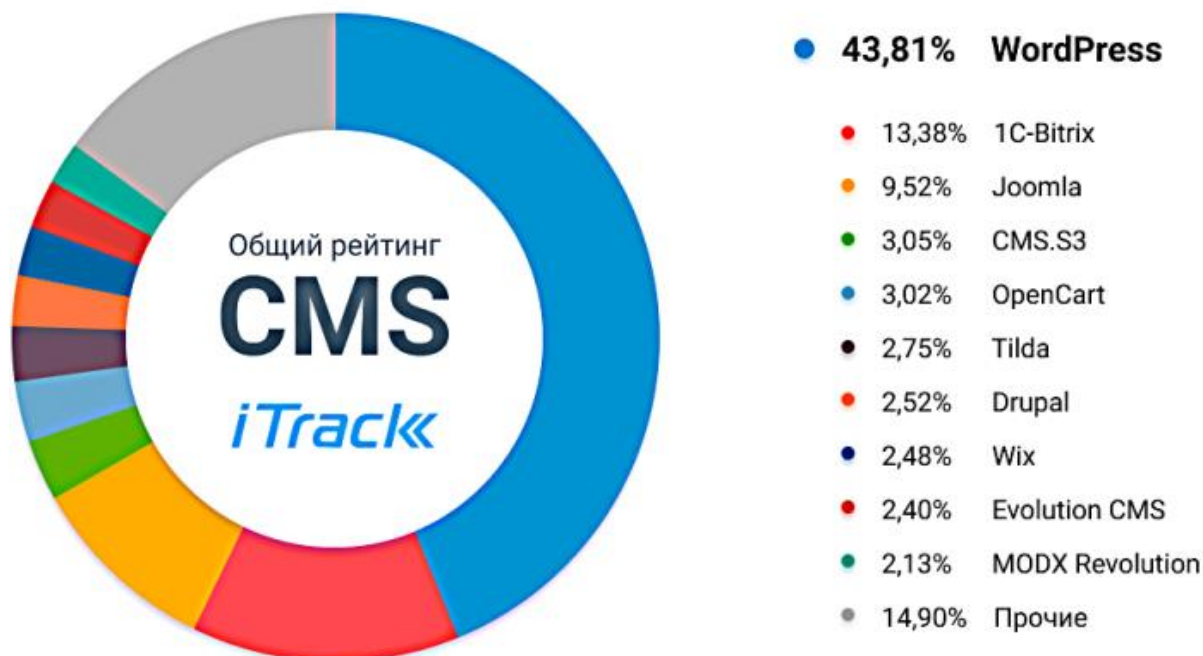


Рисунок 1 – Рейтинг CMS за 2021 год

Кроме того, существуют готовые решения для быстрой и удобной разработки сайтов с помощью программных средств, например, такие как:

- программы, имеющие в своем составе визуальные редакторы (design-based editor) – средства, автоматически формирующие необходимый HTML-код, позволяя разрабатывать Web-страницы. Пример: TinyMCE, Spaw, NicEdit, Xinha.

- программы-редакторы (code-based editors), предоставляющие редактор и вспомогательные средства для автоматизации написания кода. Программа устанавливается на ПК и помогает верстать, генерируя код любого элемента на сайте. Примером является Adobe Dreamweaver, FrontPage, Adobe Muse.

Нами проанализированы порталы ряда кафедр российских вузов с целью изучения наполнения действующих порталов кафедр различных направлений, для четкого понимания, какие блоки и разделы необходимо задействовать на собственном разрабатываемом сайте. Этот этап предварительного исследования и сбора информации определяет все последующие стадии разработки.

На рис. 2. показан фрагмент таблицы, используемой для анализа наполнения действующих порталов кафедр различных направлений и разных вузов России. Полученные результаты позволили сделать выводы о том, какие блоки и разделы необходимо разместить на разрабатываемом сайте с тем, чтобы он отвечал современным требованиям, предъявляемым к наполнению сайтов кафедр и вузов.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		1	2	3	4	5	6	7	8	9	10	11	12
2	Портал кафедры	О кафедре/Кафедра	Студенту/Учеба	Абитуриенту	Новости/форум	Контакты	Сервисы	Главная	Медиа	Авторизация	Научная работа	Вуз	Кафедра
3	<a href="https://cs.istu.ru/index.php?">https://cs.istu.ru/index.php?</a>	+	+	+	+	+	-	-	-	-	-	Ижевский государственный	Программное обеспечение
4	<a href="http://asu.ugatu.ac.ru/">http://asu.ugatu.ac.ru/</a>	+	+	-	+	-	+	-	-	+	+	Санкт-Петербургский государственный университет	Технологии программирования
5	<a href="http://is.ifmo.ru/labs/">http://is.ifmo.ru/labs/</a>	-	+	-	+	-	-	+	+	-	+		Технологии программирования
6	<a href="https://se.math.spbu.ru/back">https://se.math.spbu.ru/back</a>	+	+	+	+	+	+	+	-	+	-		Системное программирование
7	<a href="https://isse.cdo.vlsu.ru/">https://isse.cdo.vlsu.ru/</a>	+	+	+	-	+	-	+	-	-	+	Владимирский государственный Университет	Информационные системы и программная инженерия
8	<a href="http://pm.samgtu.ru/node/7">http://pm.samgtu.ru/node/7</a>	+	+	-	+	+	-	-	-	-	+	Самарский политех	Прикладная математика и информатика
9	<a href="https://fv.bmstu.net/">https://fv.bmstu.net/</a>	+	+	-	+	-	-	+	+	+	-	МГТУ им. Баумана	Физвоспитания
10	<a href="https://ru9.bmstu.ru/">https://ru9.bmstu.ru/</a>	+	-	+	+	+	-	+	-	-	+		Теоретическая информатика и компьютерные технологии
11	<a href="https://mgt.vvsu.ru/">https://mgt.vvsu.ru/</a>	+	+	+	+	+	-	-	-	-	-	Владивостокский государст	Экономики и управления
12	<a href="http://www.sppmat.ru/">http://www.sppmat.ru/</a>	+	+	+	+	+	-	+	-	+	+	Московский энергетический институт	Прикладная математика и искусственный интеллект
13	<a href="https://fmf.brgu.ru/kafedra-2">https://fmf.brgu.ru/kafedra-2</a>	+	+	+	+	+	-	+	+	-	-	Брянский государственный университет	Информатики и прикладной математики
14	<a href="http://www.amn.vsu.ru/vm">http://www.amn.vsu.ru/vm</a>	+	-	-	-	+	-	-	-	-	-	Воронежский государственный университет	Кафедра вычислительной математики и прикладных информационных технологий
15	<a href="http://programming.iis.nsk.s">http://programming.iis.nsk.s</a>	+	+	+	+	+	-	+	-	-	-	Новосибирский государственный университет	Кафедра программирования
16	<a href="https://prm.susu.ru/about/h">https://prm.susu.ru/about/h</a>	+	+	+	+	+	-	+	-	-	-	Южно-Уральский государственный университет	Прикладная математика и программирование
17	<a href="https://dep_ct.pnzgu.ru/">https://dep_ct.pnzgu.ru/</a>	+	+	+	+	+	-	+	-	-	+	Пензенский государственный	Компьютерные технологии

Рисунок 2 – Основные вкладки порталов кафедр  
Структура предлагаемого макета сайта показана на рис.3.

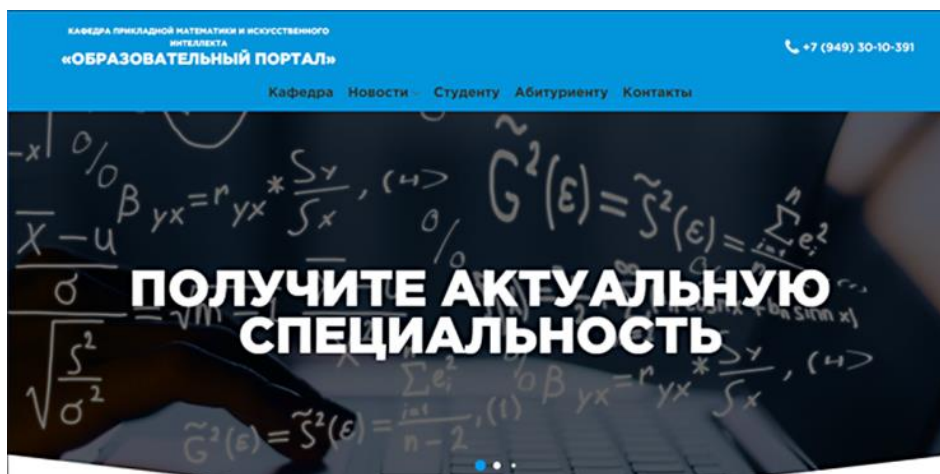


Рисунок 3 – Макет главной страницы портала кафедры

Следует отметить, что при разработке сайта оценивали, в каком направлении будет проходить развитие ресурса, каковы потребности кафедры и в чем заключается дальнейшее продвижение веб-сайта в интернете. Для разработки собственного сайта портала кафедры был использован программируемый метод с последующим интегрированием в систему управления контентом.

## Выводы

В условиях постоянного совершенствования сферы образования, изменения требований к качеству поставляемого образования актуальным становится вопрос о разработке и наполнении сайтов вузов и кафедр, несущих большую смысловую, информационную, учебную и др. нагрузку. Рассмотрены различные варианты создания сайта кафедры, проанализировано большое количество сайтов кафедр ряда российских вузов с целью формулирования требований, которым должен отвечать современный сайт. Полученная информация позволила сформулировать требования и разработать макет образовательного портала кафедры искусственного интеллекта и программирования ДОННТУ.

## Литература

1. Подкорытов В. О. Сайт кафедры вуза как элемент консалтингового проекта по продвижению образовательных услуг // Проблемы Науки. 2017. №16 (98 [Электронный ресурс] – Режим доступа:

<https://cyberleninka.ru/article/n/sayt-kafedry-vuza-kak-element-konsaltingovogo-proekta-po-prodvizheniyu-obrazovatelnyh-uslug> (дата обращения: 02.11.2022).

2. Spy Metrics. Анализ конкурентов. [Электронный ресурс] – Режим доступа: <https://spymetrics.ru/ru> - Загл. с экрана.

3. ТОП-100 ВУЗов России 2022. Поступи. ИНФО. [Электронный ресурс] – Режим доступа: <https://postupi.info/rating> - Загл. с экрана.

4. iTrack. Рейтинг CMS. [Электронный ресурс] – Режим доступа: <https://itrack.ru/rating-cms/> - Загл. с экрана.

5. IT рейтинг России. [Электронный ресурс] – Режим доступа: <https://it-rating.top/rating-cms-2021> - Загл. с экрана.

6. Русский кодекс WordPress. [Электронный ресурс] // [wordpress.org](https://wordpress.org). URL: [https://codex.wordpress.org/ru:Main\\_Page](https://codex.wordpress.org/ru:Main_Page) (дата обращения 13.10.2022).

**Юрьева В.Р., Анохина И.Ю. Сайт как элемент продвижения образовательных услуг.** В статье представлен краткий обзор современных методов разработки web – ресурсов. Проанализированы требования, предъявляемые к сайтам кафедр и образовательным порталам учебных заведений. Проведенные исследования позволили сформулировать требования и принять решение об оптимальной структуре учебных web – ресурсов.

**Ключевые слова:** сайт кафедры, продвижение образовательных услуг, система создания и управления сайтом.

**Yuryeva V.R., Anokhina I.Y. Website as an element of educational services promotion.** The article provides a brief overview of modern methods of developing web resources. The requirements for the websites of departments and educational portals of educational institutions are analyzed. The conducted research made it possible to formulate requirements and make a decision on the optimal structure of educational web resources.

**Keywords:** website of the department, promotion of educational services, system of creation and management of the website.

## **СЕКЦИЯ 4. «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, КОМПЬЮТЕРНОЙ ГРАФИКИ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ»**

УДК 004.92

### **Исследование влияния технологий сглаживания на рендеринг изображения и производительность в играх**

**О.Р. Семериков<sup>\*1</sup>, С.А. Зори<sup>\*2</sup>**

<sup>\*1</sup> магистрант кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,  
semerikov2917@gmail.com

<sup>\*2</sup> д.т.н., проф. кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,  
sa.zori1968@gmail.com, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

*Семериков О.Р., Зори С.А. Исследование влияния технологий сглаживания на рендеринг изображения и производительность в играх В статье рассматриваются технологии сглаживания, причины их появления, описывается влияние сглаживания на качество изображения и производительность. На основе анализа современных технологий сглаживания формулируется авторская гипотеза о постепенной утрате их актуальности, в связи с характеристиками современных мониторов.*

**Ключевые слова:** технологии сглаживания, ступенчатые линии, рендеринг изображения, машинное обучение, плотность пикселей.

#### **Введение**

Актуальность исследования обусловлена повсеместным использованием и развитием технологий сглаживания в играх. На протяжении долгого периода времени разработчики игр стремятся к улучшению графической составляющей своих продуктов – это относится, в том числе, и к сглаживанию. Рассматриваемая проблема заключается в том, что экран монитора представляет собой матрицу пикселей, которая корректно отображает только те линии, которые находятся в горизонтальном, либо вертикальном положениях. Когда линия находится под углом появляется так называемая ступенчатость, ухудшающая общее качество изображения. Сглаживание решает эту проблему посредством закраски смежных от ступенчатости пикселей в промежуточные цвета. Однако, данная проблема была ярко выражена в прошлом. Сейчас, в связи с современными характеристиками компьютерных мониторов (а точнее плотностью пикселей в них), проблема становится всё менее явной. Если в прошлом плотность пикселей у мониторов была недостаточной (из-за большой диагонали монитора и маленького разрешения экрана), то сейчас плотность пикселей у современных мониторов приближена практически к идеальному значению. Следовательно, возникает резонный вопрос – необходимо ли использовать сглаживание в современных компьютерных мониторах и будет ли сглаживание востребовано в обозримом будущем?

Целью исследования является рассмотрение технологий сглаживания (в том числе их влияния на качество изображения и производительность), осуществление сравнительного анализа новейших технологий. На основе анализа новейших технологий формулируется авторская гипотеза о постепенной утрате актуальности сглаживания в связи с характеристиками современных компьютерных мониторов.

Список задач для выполнения поставленных целей:

- произвести обзор технологий сглаживания;
- осуществить сравнительный анализ технологий DLSS и FSR;
- сформулировать на основе анализа новейших технологий сглаживания гипотезу о постепенной утрате их актуальности.

Новизна исследования заключается в сформулированной авторской гипотезе, рассматривающей потерю актуальности сглаживания в обозримом будущем в связи с характеристиками мониторов.

## Обзор технологий сглаживания

Технологии сглаживания были разработаны задолго до использования в играх, изначально они применялись для корректного отображения текста в программах. В связи с развитием игровой индустрии и предъявлению к ней больших требований со стороны потребителей, технологии сглаживания начали внедряться и в игры. На текущий момент времени разработано множество технологий сглаживания, отличающихся друг от друга качеством предоставляемого изображения и затратами производительности. Перейдём к рассмотрению технологий, внёсших основной вклад в развитие сглаживания.

Начнём со старейшей технологии сглаживания SSAA (суперсэмплинг). Алгоритм SSAA заключается в том, что разрешение экрана сначала увеличивается (вплоть до  $\times 4$ ), в увеличенном разрешении осуществляется отрисовка кадра, после чего происходит обратное сжатие до оригинального разрешения. Благодаря алгоритму осуществляется нахождение среднего цвета от нескольких виртуальных пикселей. Например, если присутствуют 2 пикселя чёрного и белого цветов – средним цветом станет серый (см. рис. 1) [1].

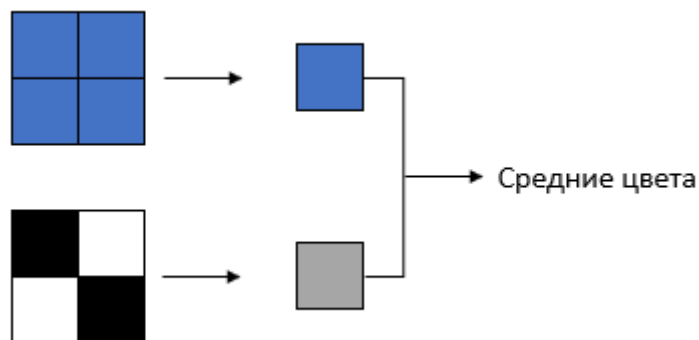


Рисунок 1 – Нахождение среднего цвета от нескольких виртуальных в SSAA  $\times 4$

В результате суперсэмплинга рендерится качественное изображение, в котором устраняется ступенчатость и, помимо этого, детализируются отдалённые объекты. Однако, SSAA существенно влияет на производительность – в связи с этим суперсэмплинг, как правило, применяется в старых (не требовательных) играх. Так, для использования SSAA в старых играх, производители видеокарт Nvidia и AMD разработали собственные вариации суперсэмплинга (DSR и VSR соответственно), добавив возможности их включения в настройках своих драйверов.

Дальнейшим развитием суперсэмплинга является MSAA (мультисэмплинг). Алгоритм мультисэмплинга по сути аналогичен SSAA, за тем лишь исключением, что в MSAA сглаживание применяется не ко всему изображению, а только к той его части, где это необходимо (ступенчатые линии, отдалённые объекты). Сравнение рендеринга изображения с использованием MSAA  $\times 8$  и без него представлено на рис. 2.



Рисунок 2 – Сравнение рендеринга изображения с использованием MSAA  $\times 8$  и без него

Как видим, изображение без сглаживания обладает чрезмерной ступенчатостью, в то время как при использовании MSAA, помимо устранения ступенчатости, улучшается и детализация отдалённых объектов. Ситуация с производительностью при использовании мультисэмплинга неоднозначная. Так, при статичном изображении MSAA гарантированно будет выдавать большее число кадров в секунду (FPS), чем SSAA. В случае же динамического изображения значение FPS будет приближено к значениям SSAA.

FXAA (быстрое приблизительное сглаживание) является самой популярной технологией на протяжении долгого периода времени. В отличие от сэмплингов, FXAA реализован с помощью шейдеров. Алгоритм FXAA заключается в нахождении резкого изменения цветов в изображении, после чего осуществляется сглаживание найденных проблемных участков. По сути, FXAA – это некий фильтр, задачей которого является размытие определённых текстур с целью устранения на них ступенчатости. Как следствие, основной недостаток FXAA



заключается в существенном размытии изображения – из-за этого, в том числе, размываются и отдалённые объекты, вследствие чего они становятся практически неразличимыми на сцене (см. рис. 3).



Рисунок 3 – Сравнение рендеринга изображения с использованием FXAA и без него

Особенность FXAA заключается в том, что он практически не влияет на производительность. Как следствие, данная технология повсеместно используется и улучшается уже на протяжении более десяти лет.

Временное сглаживание TAA, в отличие от рассмотренных ранее, устраняет не только ступенчатость линий, но и избавляет от дрожания объектов. TAA совмещает в себе технологии MSAA и SMAA, его особенность заключается в том, что при сглаживании изображения используются уже отрисованные кадры. TAA хранит значения цвета пикселей в блоке памяти, что сокращает время рендеринга изображения. Помимо этого, временное сглаживание можно использовать вместе с FXAA/SMAA, что положительно влияет на качество изображения. Недостатком TAA является его склонность к чрезмерному размытию изображения, образованию мерцающих артефактов на краях с высокой контрастностью [2].

Новейшей технологией сглаживания от Nvidia сейчас является DLSS (суперсэмплинг глубокого обучения). «DLSS, используя тензорные ядра, увеличивает отрисованное в низком разрешении изображение за счёт применения глубокого машинного обучения, что позволяет добиться качества, сравнимого с отрисовкой сцены в оригинальном разрешении».

Первая версия DLSS была провальной из-за чрезмерного размытия изображения. Ситуация изменилась с выходом второй версии, т.к. в ней применялась уже обобщённая сеть, принимающая информацию в виде векторов движения для определения вида кадра в случае его отрисовки с увеличенным изображением. Последняя версия DLSS, появившаяся вместе с видеокартами серии RTX 40, использует оптическую генерацию кадров, обеспечивающую повышение производительности (см. рис. 4) [3].

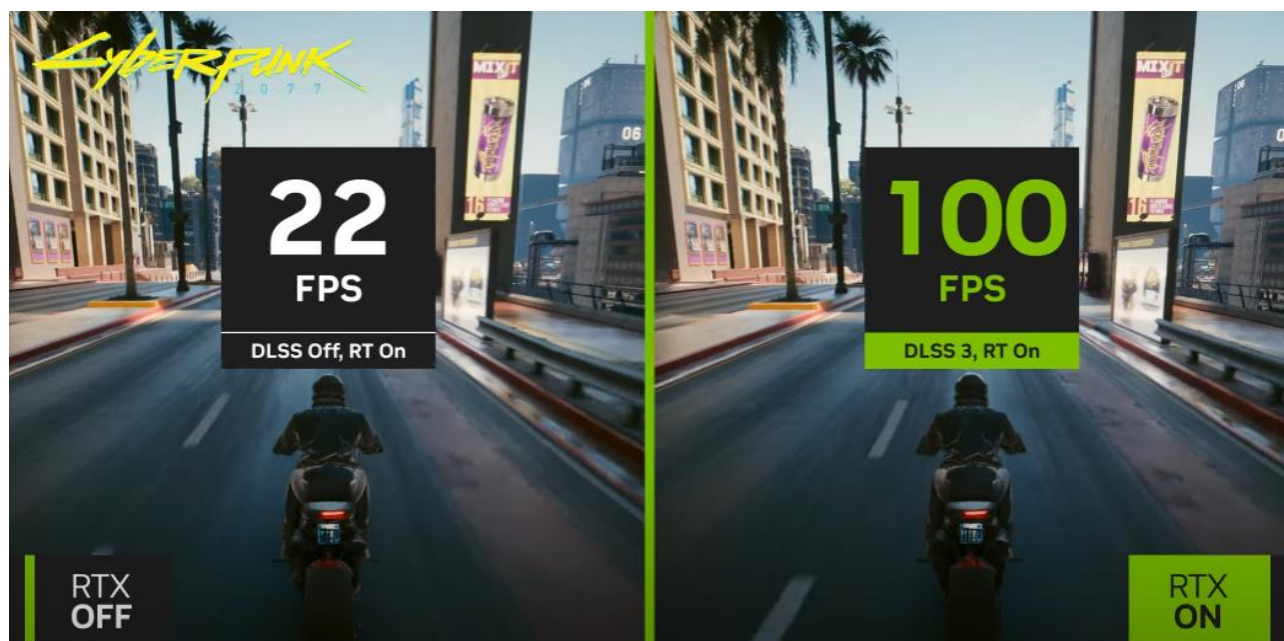


Рисунок 4 – Сравнение производительности при использовании DLSS 3 и без него

### **Сравнительный анализ DLSS и FSR**

FSR – это технология масштабирования изображения, разработанная компанией AMD с целью улучшения производительности при минимальных потерях качества изображения.

В отличие от DLSS, FSR не использует искусственный интеллект (ИИ), а полагается на пространственное масштабирование, в результате чего упрощает свой алгоритм. Пространственное масштабирование не зависит от

данных из нескольких кадров либо векторов движения, что облегчает внедрение FSR в игры, т.к. требуется меньше данных [4].

Следовательно, AMD стремится завоевать большую аудиторию посредством внедрения своей технологии во многие игры. Помимо этого, FSR поддерживает более старые видеокарты, в том числе производства Nvidia (начиная с GTX 10-й серии), в то время как возможность использования DLSS присутствует только на видеокартах Nvidia серии RTX.

На практике обе технологии разработаны с целью улучшения производительности за счёт рендеринга с более низким разрешением, но с минимальной потерей качества графики за счёт масштабирования. Однако тесты показывают, что DLSS, в силу более совершенной технологии, выдаёт более качественное изображение и лучшую производительность. FSR же составляет DLSS достойную конкуренцию, т.к. является более доступной технологией, предоставляющей близкие к DLSS результаты.

### ***Гипотеза о постепенной утрате актуальности сглаживания***

Обратимся к истории. Зачем разрабатывали технологии сглаживания? Изначально мониторы, в силу своих характеристик, не могли обеспечивать достаточную плотность пикселей, т.к. диагонали мониторов были слишком большими для имеющихся тогда разрешений экранов. Как следствие, появились технологии сглаживания, задачей которых было увеличивать разрешение, обрабатывать его в увеличенном разрешении, после чего возвращаться к исходному.

Как же обстоит ситуация сейчас? Характеристики у современных компьютерных мониторов приближены к практически идеальному значению плотности пикселей (разрешение 4K с соответствующей ему диагональю). В связи с этим, смысл использования сглаживания в подобных мониторах теряется, т.к. и без сглаживания изображение является качественным (см. рис. 5).



Рисунок 5 – Сравнение качества изображения на 4K мониторе с использованием FSR и без него

Как видим, качество изображений является по сути идентичным, что подтверждает гипотезу. Следовательно, возникает вопрос – зачем в таком случае нужно сглаживание в современных мониторах? Сейчас на примере DLSS 3 видно, что акцент сейчас делается скорее на увеличение производительности с помощью ИИ, чем на улучшение качества изображения. По мнению автора, в обозримом будущем необходимость сглаживания изображений пропадёт. Если мониторы будут выдавать идентичное изображение с использованием сглаживания и без него, то зачем тратить средства на разработку и улучшение технологий? Однако стоит отметить, что если компьютерные мониторы уже, по сути, преодолели необходимость использования сглаживания, то с телевизорами ситуация обстоит иначе. Огромные диагонали телевизоров не успевают за развитием разрешений (как правило, на 60+ диагонали используется разрешение 4K). В данном случае использование сглаживания всё ещё обоснованно, но развитие разрешений не стоит на месте и в определённый момент времени идеальная плотность пикселей будет достигнута и на телевизорах.

Из вышесказанного следует вывод: по мнению автора, в будущем может быть создан стандарт, задачей которого будет определение идеальной плотности пикселей для различных сочетаний разрешений и диагоналей в компьютерных мониторах, телевизорах, смартфонах – вследствие чего необходимость в использовании сглаживания пропадёт за ненадобностью, т.к. благодаря плотности пикселей изображение будет качественным и без него.

## Выводы

В статье были рассмотрены технологии сглаживания, причины их появления, описано влияние различных технологий на качество изображения и производительность. На основе анализа современных технологий сформулирована гипотеза об утрате актуальности сглаживания в связи с характеристиками современных мониторов.

Стоит отметить, что эволюция технологий сглаживания во многом происходила благодаря компании Nvidia (FXAA, TAA и DLSS – всё это разработки Nvidia). В процессе развития технологий сглаживания прослеживается цель разработчиков к одновременному улучшению качества изображения при сохранении оптимальной производительности. Если изначально технологии сглаживания предъявляли чрезмерные системные требования, но предоставляли качественное изображение – к текущему моменту времени всё изменилось. К примеру, при использовании технологии DLSS, помимо качества изображения, значительно увеличивается и производительность.

На основе анализа DLSS, FSR и их использовании в современных компьютерных мониторах сформулирована авторская гипотеза о постепенной утрате актуальности сглаживания. Характеристики у современных мониторов приближаются к идеальному значению плотности пикселей. Вследствие чего практически невозможно отличить разницу между использованием сглаживания и его отсутствием в подобных мониторах. Естественно, технологии сглаживания всё ещё востребованы и будут использоваться некоторое время. Например, присутствует определённая проблема с мониторами телевизоров – их диагональ слишком большая для существующих сейчас разрешений. Однако, по мнению автора, на примере современных компьютерных мониторов прослеживается постепенная утрата актуальности сглаживания, т.к. что с использованием сглаживания, что без него – качество изображения является по сути идентичным. Возможно, в будущем будет создан стандарт, задачей которого будет определить идеальную плотность пикселей для различных сочетаний разрешений и диагоналей в компьютерных мониторах, телевизорах, смартфонах – вследствие чего необходимость в технологиях сглаживания исчезнет, т.к. они не будут предоставлять каких-либо улучшений, но на разработку и развитие технологий будут тратиться умственные и финансовые средства.

В дальнейшем возможно рассмотрение перспектив создания предложенного в статье стандарта, целью которого будет определение идеальной плотности пикселей для различных сочетаний разрешений и диагоналей в компьютерных мониторах, телевизорах, смартфонах – вследствие чего необходимость в сглаживании изображения исчезнет.

## Литература

1. Сглаживание в играх [Электронный ресурс] / tproger.ru Режим доступа: <https://tproger.ru/articles/anti-aliasing-in-games/> – Загл. с экрана.
2. Как работает рендеринг в 3D-играх: сглаживание [Электронный ресурс] / habr.com Режим доступа: <https://habr.com/ru/company/pixonix/blog/555364/> - Загл. с экрана.
3. Nvidia DLSS [Электронный ресурс] / nvidia.com Режим доступа: <https://www.nvidia.com/ru-ru/geforce/technologies/dlss/> - Загл. с экрана.
4. Testing AMD FSR: How Does It Compare to Nvidia's DLSS? [Электронный ресурс] / techspot.com Режим доступа: <https://www.techspot.com/review/2277-amd-fsr-analysis-benchmark/> - Загл. с экрана.

**Семериков О.Р., Зори С.А. Исследование влияния технологий сглаживания на рендеринг изображения и производительность в играх** В статье рассматриваются технологии сглаживания, причины их появления, описывается влияние сглаживания на качество изображения и производительность. На основе анализа современных технологий сглаживания формулируется авторская гипотеза о постепенной утрате их актуальности, в связи с характеристиками современных мониторов.

**Ключевые слова:** технологии сглаживания, ступенчатые линии, рендеринг изображения, машинное обучение, плотность пикселей.

**Semerikov O.R., Zori S.A. Investigation of the impact of anti-aliasing technologies on image rendering and performance in games** The article discusses anti-aliasing technologies, the reasons for their appearance, describes the effect of anti-aliasing on image quality and performance. Based on the analysis of modern smoothing technologies, the author's hypothesis is formulated about the gradual loss of their relevance, due to the characteristics of modern monitors.

**Keywords:** smoothing technologies, stepped lines, image rendering, machine learning, pixel density.



## Прогнозирование направления будущей профессиональной деятельности учеников с помощью мультиагентной модели системы обучения

А.В. Гембицкий<sup>\*1</sup>, О.И. Федяев<sup>\*2</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
artem.gembitskiy@mail.ru

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
fedyaev@donntu.ru, OrcID: 0000-0001-6822-7306

*Гембицкий А.В., Федяев О.И. Прогнозирование направления будущей профессиональной деятельности учеников с помощью мультиагентной модели системы обучения. В статье рассмотрена разработка мультиагентной модели в инструментальной среде AnyLogic, которая моделирует сложный процесс обучения, реализуемый в распределённой интеллектуальной системе «преподаватель-ученик». Данная модель моделирует обучение школьников новым знаниям с целью определения у них менталитета технической и гуманитарной направленности.*

**Ключевые слова:** агент, мультиагентная система, инструмент AnyLogic, прогнозирование направленности менталитета учеников.

### Введение

Многоагентная технология с самого начала своего становления рассматривалась как достаточно универсальная интеллектуальная информационная технология, способная успешно конкурировать с другими технологиями в широком классе приложений, относящихся к распределённому интеллекту [1]. Однако это позиция оказалась не вполне оправданной по причине недостаточной методологической проработки данной технологии и трудности формулировки задач в области распределённого интеллекта. В результате в значительном числе прикладных задач более проработанные технологии оказались более востребованными, что подорвало доверие к МАС-технологиям в глазах индустриального сообщества и, соответственно, интерес к их использованию и финансированию.

В настоящее время существует мало многоагентных систем (МАС) [1], способных моделировать сложный процесс обучения, который представляет собой систему взаимодействия преподавателей с обучающимися, т.е. систему с распределённым интеллектом. Создание многоагентной модели, адекватно отражающей реальный процесс обучения, позволит анализировать и прогнозировать наиболее вероятные результаты обучения. В данной статье рассмотрена разработка мультиагентной модели, построенной в инструментальной системе AnyLogic [2], моделирующей обучение школьников новым знаниям с целью определения у них менталитета технической и гуманитарной направленности.

### Классификация моделей искусственных агентов мультиагентной системы

Широкое распространение различных сетей (Интернет, искусственные нейронные сети и др.) породило распределённый взгляд на мир, что способствовало появлению новой парадигмы развития программного обеспечения на основе автономных и взаимодействующих модулей – агентов [1,2]. В современной информатике с понятием «программный агент» связывают новые архитектурные принципы организации обработки информации на основе агентов. Это следует из того, что агенты рассматриваются как активные объекты, наделённые некоторой долей субъективности, т.е. способные манипулировать другими объектами, создавать и уничтожать их, а также взаимодействовать со средой и другими агентами.

Существует довольно много различных классификаций агентов на группы [1]. Наибольший интерес представляют две из них, первая из которых представлена ниже:

1. Простые агенты, которые способны выполняться автономно (то есть без вмешательства со стороны пользователя), взаимодействовать с другими агентами и пользователями, а также следить за окружением.
2. Смышлёные агенты, которые не способны выполняться автономно, но, тем не менее, могут взаимодействовать с другими агентами и пользователями, следить за окружением, а также способны использовать абстракции и предметные знания.
3. Интеллектуальные агенты, которые выполняются автономно, могут взаимодействовать с другими агентами и пользователями, следить за окружением, способны использовать абстракции и предметные знания,

имеют возможность адаптивного поведения для достижения целей, способны обучаться из окружения, являются толерантными к ошибкам и неверным входным сигналам, способны работать в режиме реального времени и взаимодействовать на естественном языке.

4. Действительно интеллектуальные агенты, которые обладают почти тем же набором функций, что и интеллектуальные, за исключением толерантности к ошибкам и неверным входным сигналам, способности к работе в режиме реального времени и взаимодействию на естественном языке.

Согласно же второй классификации программные агенты имеют следующую систему деления:

1. Интеллектуальные агенты, которые обладают определенными свойствами искусственного интеллекта.
2. Автономные агенты, которые обладают способностью изменять свой подход к решению задачи.
3. Распределенные агенты, которые выполняют действия на различных физических носителях.
4. Мобильные агенты, которые способны перемещаться на сети и выполняться на любой машине.

Каждая мультиагентная система имеет некоторую четко определенную архитектуру. Всего этих архитектур три [1]:

1. Интеллектуальная архитектура, в основе которой заложены принципы и методы работы со знаниями. Представление знаний при подобной архитектуре символическое, модель мира — исчисление, а решатель является логическим.

2. Реактивная архитектура, в основе которой заложены поведенческие модели. Представление знаний при подобной архитектуре автоматное, модель мира — граф, а решатель является автоматом.

3. Гибридная архитектура, которая совмещает в себе элементы как интеллектуальной архитектуры, так и реактивной. Представление знаний при подобной архитектуре смешанное, модель мира — гибридная, а решателем является машина вывода.

### ***Выбор инструментальной системы программирования искусственных агентов***

Инструментальная среда AgentBuilder предоставляет для разработчиков средства разработки и среду выполнения агентного приложения [1]. Среда выполнения агентного приложения состоит из агентной программы и процессора выполнения агента. Процессор использует эффективные процедуры логического вывода путём сопоставления правил поведения агента с убеждениями агента, определяемыми текущей ментальной моделью, и входящими сообщениями. На основе проводимых рассуждений процессор выполняет определенные действия, связанные с полномочиями агента. Агентная программа представляет собой определение агента в виде файла на языке RADL вместе с укомплектованной библиотекой классов проекта. Агентная программа совместно с процессором образуют выполняемого агента. При запуске среды выполнения, инициализируется процессор агента, который использует RADL-модель и онтологию агента, представленную в виде библиотеки классов проекта (Project Accessories Library). Для этого необходимы определение агента (файл RADL, который обеспечивает агента способностью рассуждения и начальной ментальной моделью) и библиотека классов проекта (вспомогательные классы проекта PACs из библиотеки классов проекта) - эти объекты используются для отображения предметной области задачи.

В среде Vee-gent разработка агентно-ориентированных приложений выполняется по методологии спецификации поведения агентов распределенной системы с использованием MAC - библиотеки, реализованной на языке Java [1,2]. На основе предлагаемых системой Vee-gent графических средств, возможна четкая структуризация поведения каждого агента в виде графа состояний и определение протоколов взаимодействий агентов. Графы состояний агентов строятся на основании жизнеспособности ролей, определенных в виде регулярных выражений на этапе агентно-ориентированного анализа. Граф состояний регистрирует все имена состояний, в которых агент может находиться. На следующем шаге разработки определяются классы для каждого состояния. Каждое состояние в графе является экземпляром класса AwtIPState из агентной библиотеки фирмы Toshiba, реализованной на языке Java. В конструкторе класса определяются пред и пост условия, т.е. условия, которые должны быть выполнены агентом в текущем состоянии для того, чтобы выполнить действия, определенные классом состояния, и определить переход в следующее состояние. Затем специфицируются действия, которые должны быть выполнены в каждом состоянии (включая собственные процессы агента и взаимодействия с другими агентами). Для начального и конечного состояний также создаются классы "INIT" и "END". Если агент взаимодействует с другими агентами, то при спецификации отдельных состояний система Vee-gent предусматривает определение протокола взаимодействия. Протокол должен отражать все линии поведения агента в данном состоянии. В каждом состоянии деятельность агента направлена на выполнение протоколов взаимодействия с целью реализации планируемой линии поведения. Деятельность каждого агента в MAC определяется, например, моделью услуг, разработанной на этапе агентно-ориентированного анализа по методологии Gaia.

JACK приложение представляет собой исходный код, реализующий характерные для агентно-ориентированного подхода понятия: агентов, способностей, события, планы, убеждения, view (запросы), а также Java класс с функцией main(), которая является точкой входа для виртуальной машины Java, и любые другие Java необходимые файлы [1]. Файлы, которые создаются для этих понятий, должны иметь такое же имя, как и у объекта, определяемого в файле. Они имеют расширение, определяющее тип JACK понятия. Компилятор агентов

JACK конвертирует исходные файлы на языке агентов JACK в код на языке Java, который затем компилируется в код виртуальной машины Java для выполнения на целевой системе.

JADE базируется на технологиях Java RMI, Java CORBA IDL, Java Serialization и Java Reflection API [1]. Разработка MAC в этой среде упрощается благодаря использованию FIPA-спецификаций и ряда инструментов поддержки фазы отладки и развертывания системы. Эта агентная платформа может устанавливаться на компьютерах с разными операционными системами, и ее можно конфигурировать через удаленный GUI-интерфейс. Процесс конфигурирования этой платформы достаточно гибкий: ее можно изменить даже во время исполнения программ, посредством перемещения агентов с одной машины на другую. Единственным требованием для работы системы является установка на машине Java Run Time 1.2. Каждый запущенный экземпляр среды JADE является контейнером, т.к. может содержать несколько агентов. Группа активных контейнеров образует платформу. Главный контейнер всегда должен быть активен, а все другие контейнеры должны быть зарегистрированы им при их создании. Поэтому, первый контейнер, запущенный на платформе является основным контейнером, а все остальные - обычными контейнерами и должны получить указания о том, где находится их основной контейнер, на котором они должны быть зарегистрированы. Если в сети запускается еще один основной контейнер, то он представляет собой другую платформу, на которой новые обычные контейнеры имеют возможность регистрироваться.

Программное обеспечение AnyLogic [2,3] представляет из себя систему для мультиметодного имитационного моделирования для таких отраслей как: транспорт, производство, логистика, добывающая промышленность, цепочки поставок, здравоохранение и других. Инструмент написан на языке Java SE. Он позволяет создавать имитационные модели с использованием различных методологий и языков моделирования, включая дискретно-событийное моделирование, агентную динамику, системную динамику, стохастическое моделирование, блок-схемы процессов, диаграммы состояний и диаграммы действий. Он позволяет представлять визуальные модели с графическими объектами для визуализации различных объектов в соответствии с бизнес-спецификациями. AnyLogic имеет встроенные ГИС-карты, которые позволяют искать и находить города, улицы, больницы, магазины и др., использующиеся для создания имитационных моделей.

При построении любой модели задаётся её структура (компоненты и связи этих компонентов) и поведение отдельных компонентов. В AnyLogic активный объект имеет структуру и поведение. Структуру активного объекта составляют его параметры, переменные, стейтчарты, а также экземпляры других активных объектов, включённых как компоненты в данный активный объект и их связи. В AnyLogic скорость выполнения модели может быть установлена максимальной (выполнение в режиме виртуального времени с максимально возможной скоростью выполнения, соответствующей программному коду) либо соответствующей (при возможности) реальному физическому времени с некоторым коэффициентом. Запуск модели на выполнение производится в AnyLogic в соответствии с определенным набором значений параметров модели, а также с некоторыми дополнительными установками (например, точность, шаг численных методов и т. п.). Совокупность всех установок для проведения компьютерного эксперимента с моделью называется в AnyLogic - экспериментом. Все эксперименты, возможные для выполнения в данном проекте, представлены как элементы группы (корня) с именем «эксперименты» в окне проекта.

Рассматриваемое «Агентное» моделирование является одним из различных подходов имитационного моделирования заложенных в системе AnyLogic. Система позволяет тестировать и исследовать сценарии «что если» при помощи имитационного 2D-и 3D-моделирования. В AnyLogic две фазы имитационного моделирования - разработка модели и её анализ - явно разделены. Разработка модели выполняется в среде редактора AnyLogic, анализ модели происходит в среде исполнения. В каждой фазе существуют свои средства управления. Переход из одной фазы в другую производится очень легко. При разработке модели можно многократно использовать переход между фазами редактирования и исполнения модели.

Агент реализован с помощью базового объекта AnyLogic - активного объекта. В модели AnyLogic можно создавать классы активных объектов и далее использовать в модели любое число экземпляров этих классов. Активный объект имеет параметры, которые можно менять извне, переменные, которые можно считать памятью агента, а также повеление. Параметры могут указывать пол агента, дату рождения и т. п. Переменными можно выразить, например, возраст агента, его координаты в пространстве, социальные свойства. Стейтчарты могут выражать поведение: состояния агента и изменение состояний под воздействием событий и условий. Кроме того, агент может иметь интерфейс для взаимодействия с окружением. Наряду со стейтчартами, для спецификации поведения агентов могут использоваться таймеры, функции - весь арсенал средств, доступный в AnyLogic, в любых их комбинациях.

Агенты обычно функционируют в некоторой среде, и взаимодействие со средой является важной функцией агента. Роль среды для агентов играет либо активный объект, в который вложены агенты, либо в качестве среды может быть использован другой активный объект. Среда может быть пассивной либо иметь свое поведение, т.к. это тоже активный объект. Динамика среды может задаваться уравнениями, стейтchartом, потоковой диаграммой, использованием таймеров и т. д. У среды может быть тактовый таймер, который циклически запускает вызов функции обращения к агентам для их продвижения или выполнения ими

собственных операций. Работу агентов в такой среде можно назвать синхронным в отличие от асинхронного их функционирования, когда каждый агент имеет свои собственные средства продвижения времени.

Определив индивидуальное поведение каждого активного объекта, можно затем наблюдать в агентной модели, как порождается коллективное поведение всей системы и как характеристики этого коллективного поведения зависят от индивидуальных параметров членов коллектива. Это именно то важное свойство, что даёт агентное моделирование. Агентный подход применяется для решения проблем во всех тех случаях, когда именно индивидуальное поведение объектов является существенным в системе, а интегральные характеристики и динамика всей системы выводятся из этих индивидуальных повелений. С помощью агентов можно моделировать рынки, конкуренцию компаний на рынке, динамику населения и многое другое [1,2,3]. По таким критериям, как арсенал инструментов, технологичность, простота и доступность, для решения задачи моделирования системы обучения выбран инструмент AnyLogic.

### **Описание агентной модели прогнозирования вида профессиональной деятельности**

В предлагаемой модели реализовано взаимодействие «популяции агентов»: школьников – MyStudent и одиночных агентов учителей – mathematicsTeacher, physicsTeacher и т. д. Для демонстрации процесса обучения были выбраны восемь школьных предметов: математика, физика, химия, информатика, русский язык, история, география, литература. Общая схема мультиагентной модели представлена на рисунке 1.

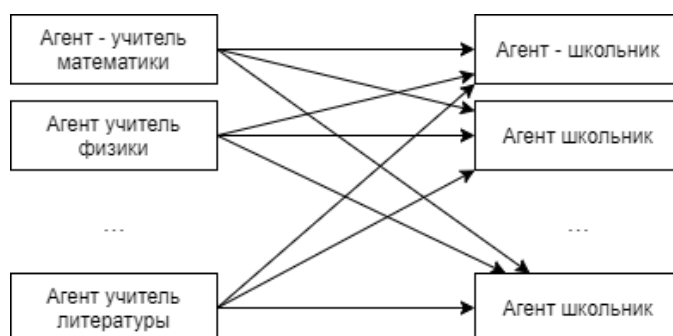


Рисунок 1 – Общая схема МАС модели

С целью демонстрации упрощённого процесса обучения школьников, как распределённой динамической системы, в агенты типа MyStudent были добавлены две группы параметров. Первая группа параметров (см. рис.2) отвечает за текущее количество знаний по каждому из предметов. Вторая группа параметров (см. рис.3) отвечает за predisposition школьника к пониманию и запоминанию знаний по каждому из предметов.

mathematicsKnowledge	-усвоенные знания по математике
physicsKnowledge	-усвоенные знания по физике
chemistryKnowledge	-усвоенные знания по химии
informaticsKnowledge	-усвоенные знания по информатике
russianLanguageKnowledge	-усвоенные знания по русскому языку
historyKnowledge	-усвоенные знания по истории
geographyKnowledge	-усвоенные знания по географии
literatureKnowledge	-усвоенные знания по литературе

Рисунок 2 – Параметры, отвечающие за усвоенные знания

mathematicsTrainAbility	-способность к усваиванию математики
physicsTrainAbility	-способность к усваиванию физики
chemistryTrainAbility	-способность к усваиванию химии
informaticsTrainAbility	-способность к усваиванию информатики
russianLanguageTrainAbility	-способность к усваиванию русского языка
historyTrainAbility	-способность к усваиванию истории
geographyTrainAbility	-способность к усваиванию географии
literatureTrainAbility	-способность к усваиванию литературы

Рисунок 3 – Параметры, отвечающие за способность к усваиванию предмета

В качестве единицы времени обучения в рамках модели были выбраны – месяцы. Каждый месяц учителя каждого предмета отправляют каждому студенту сообщение (знание), соответствующее их предмету. При получении сообщения производится пополнение текущих знаний школьника новой порцией знаний за этот месяц по формуле (аналогичные формулы используются и для других предметов):

$$\text{mathematicsKnowledge} = \text{mathematicsKnowledge} + 1 * \text{mathematicsTrainAbility} \quad (1)$$

Для демонстрации работы при каждом запуске начальные знания и предрасположенность к знаниям по предметам, генерируются случайно – от 0,5 до 4 и от 0,01 до 1, соответственно.

По истечению определённого периода обучения (в модели, в качестве примера, было использовано 99 месяцев) производится итоговый расчёт направленности знаний по следующей формуле:

$$\text{mathematicsKnowledge} + \text{physicsKnowledge} + \text{informaticsKnowledge} + \text{chemistryKnowledge} < \text{russianLanguageKnowledge} + \text{historyKnowledge} + \text{geographyKnowledge} + \text{literatureKnowledge} \quad (2)$$

При выполнении неравенства, направленность итоговых знаний школьника можно считать – гуманитарной. В противном случае, направленность считается технической.

На рисунке 4 представлена диаграмма состояний агента – учителя математики. В состоянии “Educate” агент каждый месяц отправляет всем агентам школьникам сообщение "Mathematics lesson.". По истечению 99 месяцев агент переходит в состояние “EndEducate” и в дальнейшем бездействует. Агенты учителей по другим предметам работают по схожему принципу.

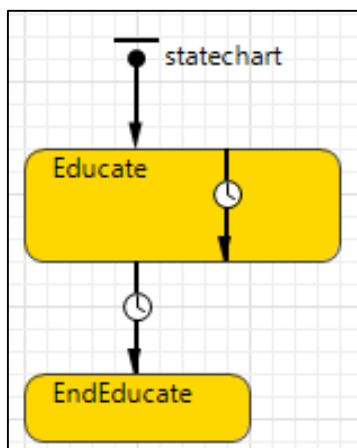


Рисунок 4 – Диаграмма состояний агента учителя математики системы AnyLogic

Диаграмма агента – школьника, представлена на рисунке 5. В состоянии “IsStudying” агент получает сообщения от различных учителей и рассчитывает новые значения показателей знаний по соответствующим предметам. Через 99 месяцев агент переходит в состояние “EntersTheUniversity”, в следствие чего изменяет свой цвет на зелёный или красный, в случае технической или гуманитарной направленности знаний, соответственно.

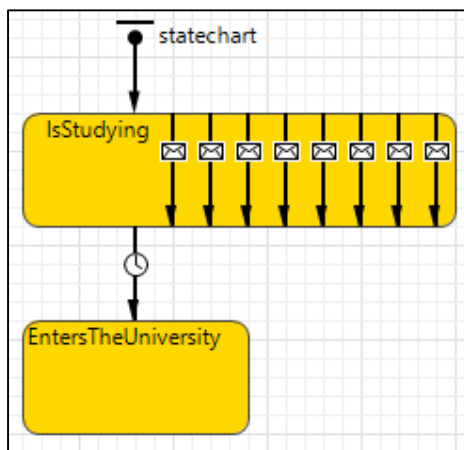


Рисунок 5 – Диаграмма состояний агента школьника системы AnyLogic

Результат моделирования процесса обучения (в рассмотренной постановке) представлен на рисунке 6. Для значений исходных данных, введенных случайным образом, путём моделирования динамики обучения получились следующие результаты: 16 учеников с технической направленностью знаний, и соответственно, 14 с гуманитарной.

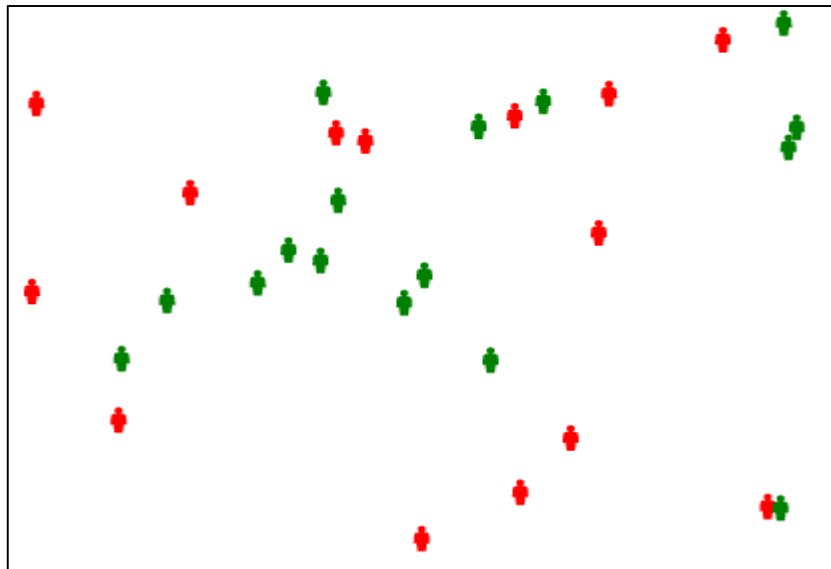


Рисунок 6 – Результат моделирования процесса обучения для определения профессиональной деятельности

### **Заключение**

В работе проведена классификация искусственных агентов, используемых в мультиагентных системах, рассмотрен подробный анализ существующих инструментальных систем программирования МАС. В инструментальной среде AnyLogic разработана упрощённая мультиагентная модель для прогнозирования направленности способностей после обучения агентов – школьников. Эта модель, конечно, отображает упрощённый метод прогнозирования вида будущей профессиональной деятельности школьников, но она, как прототип, может быть дополнена и расширена усложнёнными алгоритмами работы искусственных агентов с целью повышения точности и вариативности интересующих показателей процесса обучения и профориентации.

### **Литература**

9. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика / В. Б. Тарасов. – Москва, 2002 - 351 с.
10. AnyLogic [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/AnyLogic>
11. AnyLogic: имитационное моделирование для бизнеса [Электронный ресурс]. – Режим доступа: <https://www.anylogic.ru>

*Гембицкий А.В., Федяев О.И. Моделирование итоговой направленности знаний школьников при помощи системы AnyLogic. В статье рассмотрена разработка мультиагентной модели в инструментальной среде AnyLogic, которая моделирует сложный процесс обучения, реализуемый в распределённой интеллектуальной системе преподаватель-ученик. Данная модель моделирует обучение школьников новым знаниям с целью определения у них менталитета технической и гуманитарной направленности.*

**Ключевые слова:** агент, мультиагентная система, инструмент AnyLogic, прогнозирование направленности менталитета учеников.

**Gembitsky Artem, Fedyaev Oleg. Modeling of the final orientation of students' knowledge using the AnyLogic system.** The article considers the development of a multi-agent model in the AnyLogic tool environment, which simulates a complex learning process implemented in a distributed intelligent teacher-student system. This model simulates the teaching of new knowledge to schoolchildren in order to determine their mentality of technical and humanitarian orientation.

**Key words:** agent, multi-agent system, AnyLogic tool, forecasting the orientation of students' mentality.

## Методы генеративного моделирования в промышленном дизайне

М.П. Руденко<sup>1</sup>, Д.Е. Звягинцев<sup>2</sup>,

к.т.н, доцент, Донецкий национальный технический университет,  
m.p.rudenko@mail.ru, OrcID: 0000-0002-4892-5315, SPIN-код: 5530-6706  
магистрант, Донецкий национальный технический университет,  
zvyaginsevdmitriy00@gmail.com

*Руденко М.П., Звягинцев Д.Е. Методы генеративного моделирования в промышленном дизайне. В статье представлен краткий обзор существующих методов генеративного моделирования в промышленном дизайне. Определены преимущества и ограничения методов. Дальнейшей целью исследования является анализ средств генеративного моделирования.*

**Ключевые слова:** генеративное моделирование, алгоритмы, методы, промышленный дизайн, современные технологии, оптимизация, моделирование, искусственный интеллект.

### Введение

Современность требует новых подходов в проектировании промышленных образцов, основанных на аддитивных технологиях и искусственном интеллекте. Большинство проектов по разработке продукта являются сложными задачами оптимизации. Проектные группы подходят к ним итеративно, уточняя первоначальное наилучшее предположение с помощью инженерного анализа, интерпретаций и уточнения. Но каждая такая итерация требует времени и денег, и компании могут достичь только нескольких успешных итераций в течение срока разработки. Поскольку компании редко имеют возможность исследовать альтернативные решения, которые значительно отклоняются от их базовых предположений, слишком часто окончательный дизайн является неоптимальным.

Современные технологии предлагают альтернативу, используя цифровое моделирование и анализ, благодаря которым проекты могут быть оценены за считанные секунды или даже меньше. Алгоритмы могут автоматически корректировать геометрию детали между симуляциями без необходимости ручного уточнения. Используя методы искусственного интеллекта, новые системы генеративного моделирования могут исследовать гораздо большую вселенную возможных решений, сравнивая результаты тысяч симуляций, чтобы приблизиться к дизайну, который обеспечивает наиболее благоприятную комбинацию атрибутов [1].

### Основная часть

Генеративное моделирование – это процесс исследования дизайна. Дизайнеры или инженеры вводят цели проектирования в программное обеспечение для генеративного проектирования, а также такие параметры, как производительность или пространственные требования, материалы, методы производства и ограничения затрат. Программное обеспечение исследует все возможные перестановки решения, быстро генерируя альтернативы дизайна [2].

Генеративное моделирование позволяет разрабатывать высокопроизводительные 3D-печатные продукты и является почти необходимым инструментом для любого рабочего процесса.

Одним из ключевых преимуществ промышленной 3D-печати является то, что она дает инженерам возможность производить очень сложные и высокопроизводительные детали, которые либо невозможны, либо непомерно дороги в производстве с использованием традиционных методов.

Однако моделирование этих сложных и оптимизированных геометрий вручную в традиционном программном обеспечении САПР является почти невыполнимой задачей. Цифровой набор инструментов генеративного проектирования позволяет инженерам управлять сложностью аддитивного производства и использовать его в своих интересах.

### Преимущества и ограничения генеративного дизайна

Цифровые возможности генеративного моделирования могут разблокировать ранее недоступное пространство дизайна. Используя его инструменты, можно создавать более легкие и высокопроизводительные детали с повышенной функциональностью. Также оно имеет применение во всех областях разработки продукта

– от улучшения управления тепловыми режимами в электронных устройствах до разработки более эффективных ракетных двигательных установок.

Генеративное моделирование ускоряет все этапы разработки продукта – от концептуального дизайна до производства. Используя свои цифровые инструменты, инженеры могут быстро генерировать геометрии с высокой сложностью, от органических деталей свободного потока до повторяющихся шаблонов с миллионами элементов. Поскольку технологичность может быть принята во внимание на ранних этапах процесса проектирования, вероятность того, что позже потребуются трудоемкие пересмотры, намного ниже. За счет этого достигается более быстрый выход на рынок продукции.

При разработке новых продуктов инженеры, как правило, черпают вдохновение из своих прошлых проектов и опыта. Хотя алгоритмический подход может дать более объективные результаты, которые могут противоречить предвзятым представлениям. Сочетание этих результатов с опытом инженера дает объективное инженерное решение.

Но, как и любая другая современная технология, генеративное моделирование тоже имеет свои сложности и ограничения в проектировании продуктов. Инженерам часто приходится знать столько же о процессе, сколько и о полученном решении. Из-за сложности генеративных алгоритмов многие программные решения работают с использованием подхода «черного ящика». Инженер дает входные данные, а затем его просят оценить выходы, не имея видимости или контроля над процессом, который следовал в back-энде. Для критически важных приложений, где результаты проектирования должны давать повторяемые и воспроизводимые результаты, это значительно затрудняет производство.

Генеративное моделирование по-прежнему зависит от качества информации, которую может предоставить инженер. Оно имеет два основных входных компонента: пространство конструкции и условия загрузки. Чтобы получить оптимизированный вывод, как проблема, так и входные данные должны быть определены точно. Работа инженера заключается в определении входных параметров и цели. По этой причине генеративное моделирование является совместным процессом между инженером и программным обеспечением [3].

## **Методы генеративного моделирования**

### **1. Оптимизация топологии**

Один из методов генеративного моделирования, который хорошо зарекомендовал себя в структурной оптимизации и используется для проектирования инженерных концепций, называется оптимизацией топологии.

При оптимизации топологии конструктор сначала определяет пространство проектирования, условия загрузки, цель оптимизации и другие нетехнические требования, такие как производственные ограничения. Затем программное обеспечение удаляет материал в областях, где напряжения низкие, с помощью итеративного моделирования.

Результат имеет уменьшенный вес и высокую жесткость, которая все еще может выдерживать указанные в условии нагрузки.

Оптимизация топологии является мощным инструментом инженерного проектирования. Он может ускорить процесс проектирования продукта, снизить вес детали и производственные затраты, а также создать множество первоначальных кандидатов на проектирование для руководства с детальным проектированием.

Кроме того, можно комбинировать оптимизацию топологии с другими методами генеративного проектирования. Следуя гибриднему подходу, можно разработать, например, полые компоненты с переменной толщиной, которые имеют повышенную ударопрочность [4].

### **2. Генетический алгоритм**

Следующий метод, генетический алгоритм, является разновидностью алгоритма оптимизации, который популярен в приложениях генеративного проектирования. Генетические алгоритмы, как правило, очень полезны, когда целевая функция очень сложна, подвержена случайности или прерывиста. Генетический алгоритм начинается с исходной популяции, с которой начинается процесс отбора.

Каждый вариант дизайна является потенциальным решением общей проблемы проектирования. Каждая особь обладает уникальным набором особенностей – длинные ноги, короткие ноги, широкий верх, тонкий верх, тяжесть, легкость и т.д. Эти особенности являются генами вариантов дизайна и являются тем, что используется для развития будущего промышленного дизайна.

Некоторые из этих функций желательны, другие нет. Алгоритм использует различия между вариантами проектирования, чтобы сойтись к наилучшему возможному решению [5].

### **3. Метод симуляции**

Следующий метод – это метод симуляции. Метод симуляции подразумевает анализ потенциала структуры и процесса экспериментального формообразования. Один из очень важных методов генеративного моделирования в промышленности, где можно подвергнуть исходную модель разного типа симуляции. Например, проверить термоустойчивость модели или ее нагрузку скручивание.



Внедрение метода симуляции в проектную часть даёт неоценимый вклад в плане генерации оптимизированных форм, улучшенных прочностных характеристик структуры, возможности непрерывного экспериментального формообразования.

#### **4. Метод агентных систем**

Действия, образованные несколькими взаимодействующими, интеллектуальными агентами представляет из себя метод агентных систем. Основная цель этого метода в промышленном формообразовании заключается в генерации оптимизированных структур на основе исходных параметров путем внедрения биомиметически-самоорганизующихся процессов. Существует два вида агентных систем – стигмергические и роевые, в зависимости от органического существа, с которого была списана модель поведения алгоритма.

Данный алгоритм значительно отличается от остальных генеративных методов, так как на начальном этапе нет исходной геометрии.

Приведённый в таблице 1 анализ позволяет утверждать, что внедрение цифровых методов на этапе разработки позволяет сэкономить время проектирования продукта, уменьшить вес, оптимизировать форму и улучшить ее прочность.

Таблица 1 – Анализ методов генеративного моделирования

<b>Характеристика</b>	<b>Название метода</b>			
	<i>Оптимизация топологии</i>	<i>Генетический алгоритм</i>	<i>Метод симуляции</i>	<i>Метод агентных систем</i>
<i>Этап применения</i>	Проектирование	Разработка	Анализ	Проектирование
<i>Цель метода</i>	Ускорить процесс проектирования продукта, снизить вес детали и производственные затраты	Поиск наиболее правильных решений с точки зрения заданных параметров и ограничений.	Анализ потенциала структуры и процесс экспериментального формообразования	Генерации оптимизированных структур на основе исходных параметров путем внедрения биомиметически-самоорганизующихся процессов
<i>Оптимизация модели</i>	Есть	Есть	Есть	Есть
<i>Исходная геометрия</i>	Есть	Есть	Есть	Нет
<i>Области применения в промышленном дизайне</i>	Автомобилестроение, авиастроение, судостроение.	Проектирование и дизайн мебели, посуда	Автомобилестроение, авиастроение, судостроение, бытовая техника, посуда	Автомобилестроение, авиастроение, судостроение, проектирование мебели
<i>Результат</i>	Объект имеет уменьшенный вес и высокую жесткость, которая все еще может выдерживать указанные в условии нагрузки.	Устойчивое развитие, оптимизации геометрии объекта с точки зрения климатических и функциональных особенностей	Генерация оптимизированных форм, улучшенных прочностных характеристик структуры, возможности непрерывного экспериментального формообразования, получение уникальной геометрии, обладающей новыми художественными характеристиками.	Высокие показатели в контексте устойчивого развития и новой художественной эстетике объектов проектирования.

#### **Заключение**

В ходе исследования были проанализированы существующие методы генеративного моделирования в промышленном дизайне. Определены преимущества и ограничения методов. Основными преимуществами

являются алгоритмический подход в проектировании, благодаря которому ускоряются все этапы разработки продукта – от концептуального дизайна до производства, а также возможность генерации сложной геометрии без необходимости ручного уточнения и технологичность процесса уже на ранних этапах проектирования. Из ограничений генеративного дизайна можно выделить использование подхода «черного ящика», из-за чего инженеру необходимо уже на раннем этапе иметь четкие входные данные. Дальнейшей целью исследования является анализ средств генеративного моделирования в промышленном дизайне.

### **Литература**

1. Как генеративный дизайн может изменить будущее разработки продуктов. [Электронный ресурс] / McKinsey&Company [et al.] // Computer. – Электрон. дан. - 2022. - Vol.20, No 1. – P. 12-18. – Режим доступа: <https://ntopology.com/generative-design-guide/>. - Загл. с экрана.
2. Генеративный дизайн [Электронный ресурс] / Autodesk [et al.] // Computer. – Электрон. дан. - 2019. - Vol.12, No 1. – P. 10-11. – Режим доступа: <https://www.autodesk.com/solutions/generative-design-> Загл. с экрана.
3. Генеративное проектирование [Электронный ресурс] / nTopology [et al.] // Computer. – Электрон. дан. - 2021. - Vol.46, No 2. – P. 24-25. – Режим доступа: <https://ntopology.com/generative-design-guide/>. - Загл. с экрана.
4. Как получить максимальную отдачу от оптимизации топологии. [Электронный ресурс] / nTopology [et al.] // Computer. – Электрон. дан. - 2021. - Vol.46, No 3. – P. 1. – Режим доступа: <https://ntopology.com/blog/how-to-get-the-most-out-of-topology-optimization/>. - Загл. с экрана.
5. Генеративный дизайн в примерах. [Электронный ресурс] / matterLab [et al.] // Computer. – Электрон. дан. - 2020. - Vol.46, No 3. – P. 1-5. – Режим доступа: [https://www.generativedesign.org/02-deeper-dive/02-01\\_algorithms](https://www.generativedesign.org/02-deeper-dive/02-01_algorithms). - Загл. с экрана.

**Руденко М.П., Звягинцев Д.Е. Методы генеративного моделирования в промышленном дизайне.** В статье представлен краткий обзор существующих методов генеративного моделирования в промышленном дизайне. Определены преимущества и ограничения методов. Дальнейшей целью исследования является анализ средств генеративного моделирования.

**Ключевые слова:** генеративное моделирование, алгоритмы, методы, промышленный дизайн, современные технологии, оптимизация, моделирование, искусственный интеллект.

**Rudenko M.P., Zvyagintsev D.E. Generative modeling methods in industrial design.** The article provides a brief overview of the existing generative modeling in industrial design methods. The advantages and limitations of the methods are determined. The further purpose of the study is to analyze the means of generative modeling.

**Key words:** generative modeling, algorithms, methods, industrial design, modern technologies, optimization, modeling, artificial intelligence.

## **Функциональные возможности программы для трехмерного моделирования Blender**

**Д.М. Зеленский, Л.А. Лазебная**

бакалавр, Донецкий национальный технический университет,  
wasd3739@gmail.com,

к.т.н, доцент, Донецкий национальный технический университет,  
L\_Lazebnay@mail.ru, OrcID: 0000-0003-2442-154X, SPIN-код: 3146-9301

*Зеленский Д.М., Лазебная Л.А. Функциональные возможности программы для трехмерного моделирования Blender. В статье рассматриваются вопросы применения бесплатного 3D – приложения с открытым исходным кодом, а именно графического редактора Blender и его дополнения Blender GIS. Описаны функциональные возможности Blender GIS на примере построения карты с рельефом местности.*

***Ключевые слова:** компьютерная графика, моделирование, проектирование, трехмерное пространство, Blender 3D.*

### **Введение**

По мере формирования технологий все больше отраслей индустрии используют 3D-моделирование в своей практике. Еще пару десятилетий назад казалось, что 3D-моделирование применимо лишь для создания компьютерных игр и анимации, однако сейчас не просто назвать направление, в котором эта технология не использовалась бы. Процесс развития виртуальных моделей, позволяющий с предельной точностью показать размер, форму, внешний вид и иные характеристики объекта, используется в планировке городов, рекламе и маркетинге, архитектуре и строительстве, медицине, кино, промышленности, геологии, создании игр, техническом конструировании и т.д.

Программы 3D - моделирования позволяют рассматривать модель с разных ракурсов максимально реалистично. Благодаря этому можно устранить недостатки, обнаруженные в процессе ее создания, точнее описать будущие изделия, чтобы избежать ошибок при производстве. Результатом моделирования могут стать трехмерные макеты, напечатанные на 3D-принтере. Технология 3D-печати совершенствуется, непрерывно становясь все дешевле и все более проникая в повседневную жизнь. Всё вышеперечисленное обусловило актуальность выбранной темы.

Цель работы – обзор программы Blender, ее возможностей, а также потенциала ее дальнейшего развития.

### **Основные функциональные возможности Blender**

На сегодняшний день такая область деятельности, как 3D-моделирование, является важной и востребованной в обществе. Данная область нашла обширное применение в различных областях. 3D-моделирование играет немаловажную роль при представлении и рекламировании различных продуктов и услуг.

Визуализация объектов с помощью компьютерных программ позволяет лучше представить будущий проект в реальности. 3D - моделирование представляет собой соединение разных наборов точек с геометрическими фигурами и линиями для создания моделей. При выборе технологической составляющей 3D моделирования необходимо ориентироваться на имеющееся программное обеспечение.

В области моделирования трехмерной графики существует большое разнообразие программных продуктов [1]. Среди них существуют как платные, так и бесплатные программы. К лидерам первой категории относятся 3D max, AutoCad, Компас 3D, ко второй можно отнести Blender, Wings3D и GoogleSketchUp.

Разработчики 3D-программ постоянно внедряют новые функции и усовершенствования в предлагаемые пакеты [2]. Важными факторами при выборе программы 3D – моделирования являются требования к аппаратному обеспечению, размер программы и приемлемость цены для большинства пользователей.

В последние годы требования к качеству конечного продукта возросли по отношению к возможностям программного пакета, проектных подходов в процессе компьютерного проектирования и, особенно к конечным версиям визуализации статических и анимированных моделей [3]. Учитывая эти факторы, выбор подходящей

системы трехмерной графики является важным условием. Это обуславливает применение Blender для моделирования в качестве рекомендуемого продукта.

Свободная лицензия, открытый исходный код, удобный интерфейс многосторонние аспекты дизайна и легкая платформа по аппаратным требованиям делает Blender современным технологичным инструментом, предпочитаемым многими пользователями [4, 5]. Достижение этих преимуществ стало возможным благодаря упорному труду и длительному периоду развития программы Blender до создания профессионального 3D приложения, доступного по бесплатной лицензии и с открытым исходным кодом. Blender постоянно совершенствуется, в ней есть обширный набор различных функций, благодаря которым можно создавать качественные виртуальные объекты.

К основным функциям относятся:

- **3D-моделирование.** Изначально программа была разработана как набор инструментов для 3D-моделирования, со временем в нее были добавлены дополнительные функции. Программа не уступает другим аналогичным программам, позволяя создавать фаски, выдавливать и вырезать геометрию любой формы. Для этого применяются примитивы, полигоны, метасферы, булевы операции, NURBS-кривые или инструменты, предназначенные для скульптинга.

- **Анимация.** Blender – это не только моделирование, но и анимация, причем как простая (перемещение объекта), так и сложная: движение по заданной траектории, изменение формы и внешнего вида, циклическое движение и пр. Модное направление риггинг также реализован в Blender, позволяя создавать сложные анимации с объектами, имеющими множество движущихся частей, например, поезд или человеческий персона [6, 7].

- **Текстурирование.** Blender позволяет смоделировать практически все типы материалов, которые можно себе представить. Дополнение NodeWrangler позволяет создать материал с нуля менее чем за минуту, кроме этого, есть набор инструментов для рисования текстур.

- **Визуализация.** В программе Blender используется несколько различных движков рендеринга (процесс превращения 3D-сцены в 2D-изображение). Основными являются Cycles и Eevee, причем первый — это автономный движок рендеринга, который использует трассировку траектории для создания точного освещения окружения. Второй вариант используется для скорости в ущерб реалистичности.

- **Редактор фото/видео.** Blender способен работать в качестве фото или видеоредактора, т.к. позволяет создавать маски, корректировать цвета и многое другое, но делает это довольно уникальным способом. В то время как большинство приложений, ориентированных на редактирование фотографий, похожи на Adobe Photoshop, система, которая используется в Blender для редактирования этих изображений, выполнена в виде системы узлов. Это главное преимущество, поскольку любое внесенное изменение может быть отменено или скорректировано в любой момент.

- **Игровой движок.** BlenderGameEngine был встроенным графическим и логическим движком реального времени с такими функциями, как обнаружение столкновений, движок динамики и программируемой логики. Он также позволял создавать автономные приложения в реальном времени - от архитектурной визуализации до видеоигр. В апреле 2018 года движок был исключен из серии релизов Blender 2.8 из-за того, что обновления и доработки движка отставали от других игровых движков, таких как Godot с открытым исходным кодом и Unity. В анонсе 2.8 команда Blender специально упомянула движок Godot в качестве подходящей замены для миграции пользователей BlenderGameEngine.

### ***Пользовательский интерфейс программы Blender***

Пользовательский интерфейс (UI) Blender в прошлом был очень поляризованным. Старые версии Blender требовали от пользователя запоминания десятков горячих клавиш, прежде чем можно было выполнить даже базовые задачи, что приводило к тому, что многие пользователи находили его сложным в использовании. Когда Blender был обновлен до версии 2.50, пользовательский интерфейс получил первое серьезное обновление, которое добавило множество новых функций, больше кнопок и более удобный пользовательский интерфейс.

Версия 2.8 находилась в разработке более трех лет, в ней были переработаны многие базовые системы, которые уже начали устаревать. В результате получился стабильный, быстрый, настраиваемый и дружелюбный к пользователю интерфейс, предоставляющий все современные возможности, которые вы ожидаете от пакета для моделирования (см. рис.1).

Пользовательский интерфейс Blender, разбит на шесть областей. Четыре самые большие области в центре пользовательского интерфейса называются редакторами. Каждый редактор представляет определенный способ визуализации 3D-проекта (см. рис. 2). Существует множество типов редакторов, но эти четыре открыты в рабочем пространстве по умолчанию:

- 3D Viewport: 3D Viewport — основное рабочее окно, где выполняется 3D-моделирование;
- Outliner: панель Outliner содержит список всех объектов проекта и помогает организовать сцену;
- Свойства: панель свойств содержит настройки рендеринга и позволяет добавлять расширенные модификаторы, ограничения, частицы, физику и материалы к 3D-моделям;

– Временная шкала: временная шкала используется для организации анимации. Позволяет отслеживать параметры воспроизведения и ключевые кадры.

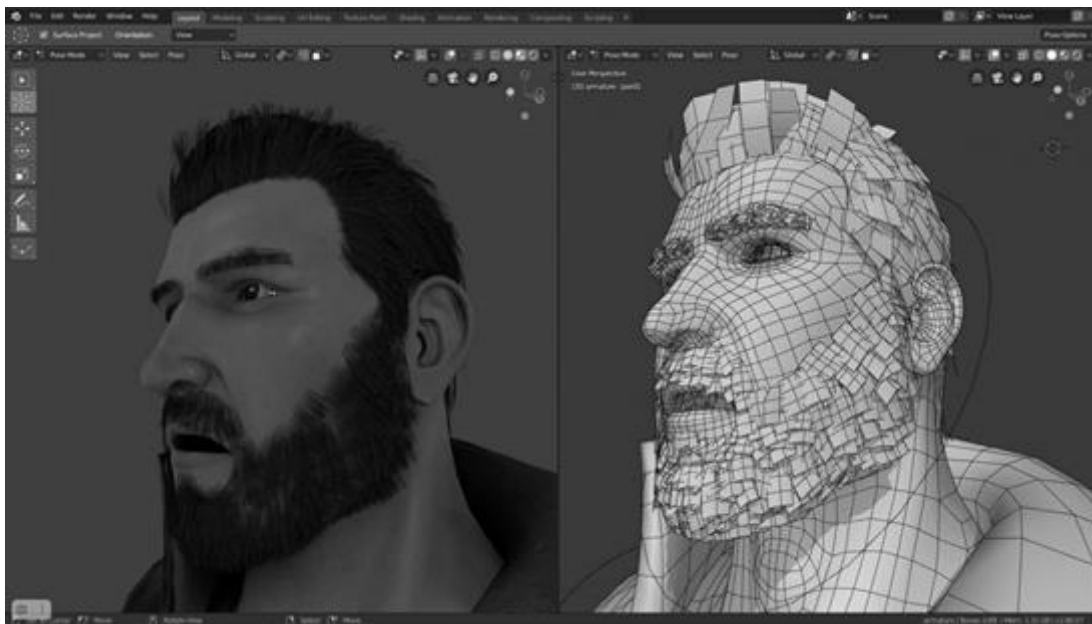


Рисунок 1– Рабочее окно программы Blender



Рисунок 2–Интерфейс Blender во время анимации

Blender 2.8 включает две новые основные части пользовательского интерфейса: верхнюю панель и панель состояния[8]:

– Верхняя панель: верхняя панель находится в самом верху пользовательского интерфейса. Логотип Blender можно увидеть в левом верхнем углу. Нажав на него, можно открыть всплывающий экран. Верхняя панель содержит типичные пункты меню, такие как «Файл», «Правка» и так далее. Самая интересная функция на верхней панели — это новые предустановки рабочего пространства, такие как Макет, Моделирование, Скульптура и другие. Эти вкладки позволят быстро перестроить пользовательский интерфейс для различных рабочих процессов.

– Строка состояния: Строка состояния находится в самом низу пользовательского интерфейса. Она содержит полезные напоминания о горячих клавишах, опции инструментов, количество полигонов и другую полезную информацию о текущем файле.

### ***Кратко о BlenderGIS***

BlenderGIS – одно из множества дополнений, созданных сообществом Blender. С помощью дополнения BlenderGIS можно легко импортировать (спутниковые) карты, карты перемещений и геометрию, например, здания. В нем можно выбрать свой регион на карте мира непосредственно в Blender, с довольно высокой точностью. Однако, в аддоне BlenderGIS, нет текстур для зданий, моделей лесных массивов и пр. Поэтому данное дополнение подойдет больше для моделирования участка местности, с последующим ручным редактированием.

Это может пригодиться, например, для презентации будущих построек на каком-либо участке местности, оптимизации маршрутов и размещения построек, т.е. в тех случаях, где нет необходимости в высокой детализации построек, но необходимы относительно точные данные рельефа местности и акватории. При этом остается возможность дальнейшей детализации модели.

### ***Пример работы с BlenderGIS***

GIS использует определенные форматы файлов в зависимости от конкретного типа данных, с которыми вы имеете дело. Например, файлы Openstreetmap (OSM) являются родными для совместного проекта Openstreetmap, в котором собраны различные геоданные, включая улицы, здания и водотоки, однако так как BlenderGIS берет данные из зарубежной службы, то и данные о регионах СНГ будут менее точными.

Рассмотрим регион туристической зоны DeerCreekDamTrailhead, USA недалеко от Wildwood (см. рис. 3).



Рисунок 3 – Реальное фото данного региона

Найдем этот регион в Blender с помощью установленного аддона (GIS→Webgeodata→Basemap), для удобства можно открыть поиск кнопкой **G** и ввести необходимое название (см. рис. 4).



Рисунок 4 – Выбор региона в BlenderGIS

После нахождения региона необходимо перенести его в рабочую область Blender. В результате получится плоское изображение карты местности (см. рис. 5).

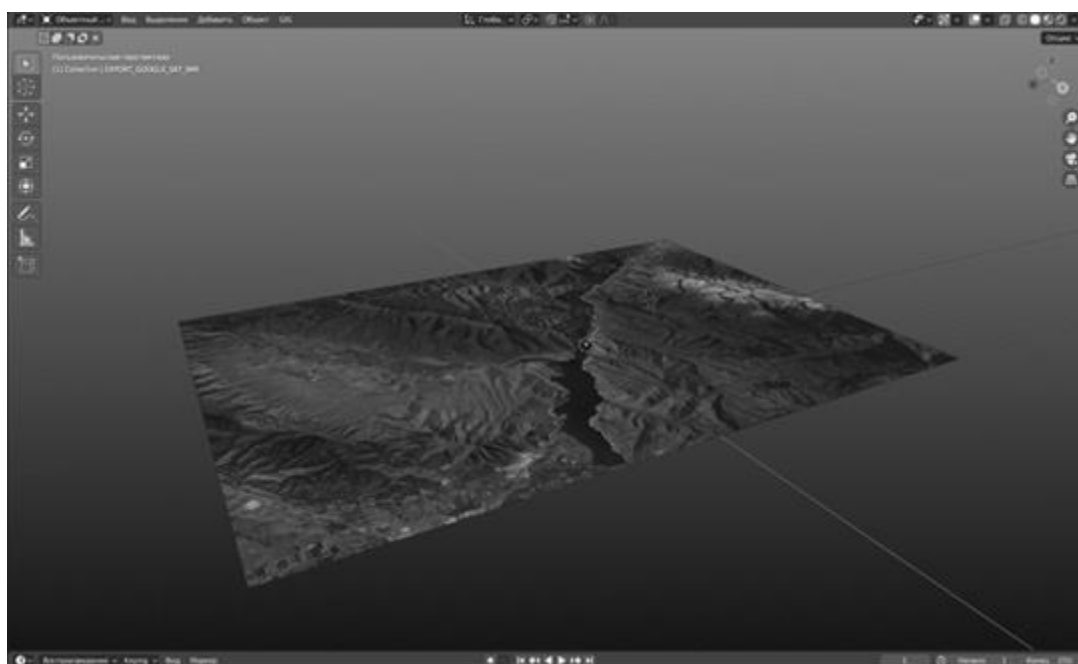


Рисунок 5 – Плоская карта местности

Теперь необходимо получить данные о карте высот, на основе которой дополнить и преобразует плоскую карту в объемную. Для этого необходимо перейти по пути GIS→Webgeodata→Getelevation (SRTM) и выбрать один из доступных источников данных. В результате получим объемную карту местности с горами и впадинами рек (см. рис. 6).

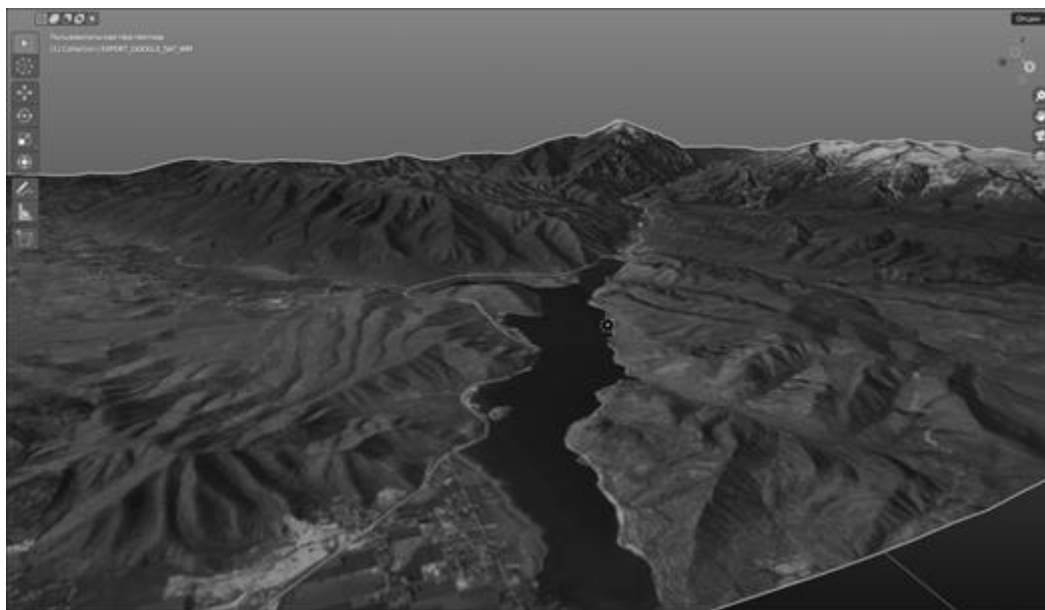


Рисунок 6 – Объемная карта местности

В дальнейшем карта может редактироваться с помощью стандартных инструментов Blender, т.е. можно при необходимости добавить сглаживание, изменить освещение, добавить различные объекты и т.д.

### **Выводы**

Система Blender — это мощный функциональный инструмент, помогающий 3D-дизайнерам в моделировании. Обеспечение полной свободы действий разработчиков Blender для потребителей является приоритетом - официально и публично регулируемым. При этом большое внимание уделяется поощрению пользователей в использовании и модификации программы. Это позволяет пользователям раскрыть свой потенциал, и параллельно самим стать соавторами в развитии Blender, что открывает перспективу будущего быстрого взаимосвязанного развития между системой с открытым исходным кодом Blender и ее пользователями [9].

Так как оборудование и программное обеспечения быстро развивается, создаются множество различных дополнений, то на наш взгляд необходимо дальнейшее исследование программы и её активное продвижение в сфере образования, так как сейчас большинство курсов по 3D-моделированию составлены на основе Maya или 3ds Max.

Аддон BlenderGIS один из популярных плагинов. Он позволяет упростить этап построения карты местности, при этом сохраняя относительную точность построения и гибкость дальнейшего взаимодействия с рельефом. Из минусов стоит отметить невозможность автоматического построения различных объектов на карте местности, а это означает, что доработка карты остается за человеком, как и исправление небольших ошибок построения, которые иногда случаются.

### **Литература**

1. Ami Chopine, 3D Art Essentials The Fundamentals of 3D Modeling, Texturing, and Animation, Elsevier, ISBN: 978-0-240-81471-1, USA, 2011, 249 – 252.
2. Regina Erak, Get started in 3D tutorials, tips and techniques to get you started in 3D art, future publishing limited, UK, 2014, 175p.
3. А.А.Портнягин, Инновационные технологии, используемые в аудитории для компьютерного моделирования, СГПИ, УДК 004.42, Россия, 2014.
4. Felician Alecu, Blender Institute – the Institute for Open 3D Projects, Open-Source Science Journal Vol. 2, No. 1, Economic Informatics Department, ASE Bucharest, Romania, 2010, 36 – 45.
5. Roger D. Wickes, Foundation Blender Compositing, ISBN-13 (pbk): 978-1-4302-1976-7, USA, 2009.
6. А.А.Прахов, Blender: 3D-моделирование и анимация. СПб : БХВ, ISBN 978-5-9775- 0393-8, Россия, 2009, 272 с: ил.
7. Lance Flavell, Beginning Blender - Open-Source 3D Modeling, Animation, and Game Design, Apress, ISBN-13 (pbk): 978-1-4302-3126-4, USA, 2010



8. А.А.Прахов, Blender 2.6, Самоучитель — СПб «БХВ-Петербург» ISBN 978-5-9775-0823-0, Россия, 2013, 384 с. ил.

9. Абрамова О.Ф. Развитие трёхмерного моделирования и анимации в программном обеспечении Blender / Агафонкин М.А., Абрамова О.Ф. // Международный студенческий научный вестник – 2018. – No 3.

*Зеленский Д.М., Лазебная Л.А. Функциональные возможности программы для трехмерного моделирования Blender. В статье рассматриваются вопросы применения бесплатного 3D – приложения с открытым исходным кодом, а именно графического редактора Blender и его дополнение Blender GIS. Описаны функциональные возможности Blender GIS на примере построения карты с рельефом местности.*

**Ключевые слова:** компьютерная графика, моделирование, проектирование, трехмерное пространство, Blender 3D.

*Zelensky D.M., Lazebnaya L.A. The functionality features of the program for three-dimensional modeling Blender. The article deals with the application of a free open-source 3D-application, namely the graphic editor Blender and its extension Blender GIS. The functional capabilities of Blender GIS are described on the example of constructing a map with terrain relief.*

**Keywords:** computer graphics, modeling, engineering, three-dimensional space, Blender 3D.

## Обзор технологий обработки больших данных

И.А. Коломойцева<sup>\*1</sup>, Д.Э. Баев<sup>\*2</sup>

<sup>\*1</sup> старший преподаватель, Донецкий национальный технический университет,  
bolatiger@mail.ru

<sup>\*2</sup> магистрант, Донецкий национальный технический университет,  
azo.cw@yandex.ru

*Коломойцева И.А., Баев Д.Э. Технологии обработки больших данных (Data Mining и Big Data). В статье описаны термины «Data Mining» и «Big Data», основные критерии определения к какой категории можно отнести имеющуюся информацию, какие языки программирования предназначены для обработки больших данных. Выполнен обзор главных принципов и основных методов работы с большими данными. Проанализированы проблемы, возникающие при анализе и обработке большого объема данных.*

**Ключевые слова:** Big Data, Data Mining, обработка больших данных, Volume, Velocity, Variety, Variability, Value, Data Lake, Hadoop.

### Введение

К Big Data относят регулярно обновляемую информацию, объем которой может превышать несколько терабайт. К такой информации относят данные, поступающие из контакт-центров; медиа социальных сетей; данные о торгах фондовых бирж [1, 2, 3] и т. п.

Термин «Big Data» подразумевает не только данные как таковые, но и принципы обработки больших данных, возможность дальнейшего их использования, порядок обнаружения конкретного информационного блока в больших массивах.

В 2001 году Meta Group сформулировала определённые критерии, которые позволяют оценить, соответствуют ли данные [2] понятию Big Data, а именно:

- volume (объем) - примерно 1 Петабайт и выше;
- vlocity (скорость) - генерация, поступление и обработка данных с высокой скоростью;
- variety (разнообразие) - разнородность данных, различные форматы и возможное отсутствие структурированности [4];
- variability (изменчивость) - разноплановая интенсивность поступления, которая влияет на выбор методик обработки;
- value (значимость) - разница в уровне сложности получаемой информации.

Так, данные, поступающие из сообщений в чат-боте интернет-магазинов, имеют один уровень сложности. А данные, которые выдают машины, отслеживающие сейсмическую активность планеты - совсем другой уровень.

Если в сжатой форме описывать процесс сбора и обработки большого массива данных, то стоит выделить основные этапы:

- 1) постановка задачи для аналитической программы;
- 2) программа осуществляет сбор данных с их параллельной подготовкой (декодирование, отсев мусора, удаление нерелевантной информации);
- 3) выбор алгоритма анализа данных;
- 4) обучение программы выбранному алгоритму с дальнейшим анализом обнаруженных закономерностей.

В большинстве случаев полученные необработанные данные хранятся в так называемом «озере данных» - Data Lake [1, 2, 3, 5, 6, 8]. Формат и уровень структуризации информации при этом может быть разнообразным:

- структурные (данные в виде строк и колонок);
- частично структурированные (логи, CSV, XML, JSON-файлы);
- неструктурированные (pdf-формат, формат документов и т. п.);
- бинарные (формат видео, аудио и изображения).

DataLake - помимо функции хранения, включает в себя и программную платформу (например, такую как Hadoop [1, 5, 7, 9] - пакет утилит и библиотек, используемый для построения систем, обрабатывающих, хранящих и анализирующих большие массивы нереляционных данных: данные датчиков, интернет-трафика, объектов

JSON, файлов журналов, изображений и сообщений в соцсетях), а также определяет источники и методы пополнения данных, кластеры узлов хранения и обработки информации, управления, инструментов обучения

### **Главные принципы работы с большими данными**

Выделяют три главных принципа, среди которых:

- 1) горизонтальная адаптивность – количество данных неограниченно, поэтому обрабатывающая их система должна иметь способность к расширению: при возрастании объёмов данных должно пропорционально увеличиваться количество оборудования для поддержания работоспособности всей системы;
- 2) стабильность в работе при отказах – горизонтальная адаптивность предполагает наличие большого числа машин в компьютерном узле [4];
- 3) концентрация данных – в масштабных системах данные распределяются по большому количеству оборудования [4].

### **Основные методы обработки больших данных**

Среди основных методов выделяют следующие девять:

1. машинное обучение;
2. нейронные сети;
3. технология Data Mining;
4. стратегия краудсорсинга;
5. метод предиктивной аналитики (методика прогнозирования);
6. принцип статистического анализа;
7. технология имитационного моделирования;
8. метод визуализации аналитических данных;
9. метод смещения и интеграции данных.

Машинное обучение содержит в своей основе способность аналитической системы самостоятельно обучаться в процессе решения различных задач [3, 6].

Нейросеть используют для распознавания визуальных образов [1, 3, 6], которая способна проделать работу за несколько десятков людей. В основном, её используют для развлечений, прогнозирования, обеспечения безопасности, медицинской диагностики и т. д.

Технология Data Mining подразумевает обнаружение определённых закономерностей в сырых данных с помощью интеллектуального анализа [2]. Data Mining используют для:

- определения нетипичных данных в общем потоке информации посредством анализа отклонений;
- поиска идентичной информации в различных источниках с помощью ассоциаций;
- определения факторов влияния на заданный параметр через регрессионный анализ;
- распределения данных по группам со схожими характеристикам, т. е. классификация данных;
- разделения записей по заранее сформированным классам, т. е. кластеризация.

Стратегия краудсорсинга применяется в тех ситуациях, когда нет экономической выгоды в разработке системы ИИ, для выполнения разовых работ привлекают большое количество людей, которые в свою очередь могут решить те задачи, с которыми компьютер не в состоянии справиться в одиночку.

Принцип предиктивной аналитики таков: сначала нужно исследовать данные за прошлый период; выявить закономерности или факторы, которые стали причиной результата; далее с помощью нейросети или математических вычислений создать модель, которая сможет производить прогнозирование [1, 3].

Суть метода статистического анализа заключается в сборе данных, их изучении на основе конкретных параметров и получении результата, выраженного, как правило, в процентах. У этого метода есть слабое звено – неточность данных в маленьких выборках [1, 6, 10].

Имитационное моделирование отличается от методики прогнозирования тем, что берутся в учёт факторы, чьё влияние на результат затруднительно отследить в реальных условиях – выстраиваются модели с учётом гипотетических, а не реальных данных, и затем эти модели исследуют в виртуальной реальности [1, 3, 6].

Для реализации метода визуализации аналитических данных, при условии работы с большими данными, используют виртуальную реальность и «большие экраны». Метод визуализации аналитических данных позволяет быстро воспринять и сравнить, например, уровни продаж в разных регионах, или оценить зависимость объёмов продаж от снижения/увеличения стоимости товара.

В подавляющем большинстве случаев Big Data получают из различных источников, соответственно, данные имеют разнородный формат [3, 4]. Загружать такие данные в одну базу бессмысленно, так как их параметры не имеют взаимного соотношения. Именно в таких случаях применяют смещение и интеграцию, то есть приводят все данные к единому виду. Для использования информации из различных источников применяют следующие методы:

- сведение данных в единый формат посредством конвертации документов, перевода текста в цифры, распознавание текста;

- информацию для одного объекта дополняют данными из разных источников;
- из лишней информации отфильтровывают и удаляют ту, которая недоступна для анализа.

### **Проблемы анализа и обработки большого объема данных**

Основная проблема обработки большого массива данных лежит на поверхности - это высокие затраты [1, 2, 6]. Здесь учитываются расходы на закупку, содержание и ремонт оборудования, а также заработанная плата специалистов, которые компетентны в работе с Big Data.

Следующая проблема связана с большим объемом информации, нуждающейся в обработке [4].

Ещё одна проблема - приватность больших данных [2]. Конфиденциальность может быть нарушена, так как все большее количество сервисов, связанное с обслуживанием клиентов, используют данные онлайн. Соответственно, это увеличивает рост киберпреступлений. Вопрос сохранности личных данных - одна из важнейших задач, которую необходимо решать при использовании методик Big Data.

### **Классификация и источники данных**

В качестве подхода к классификации данных Big Data негласно было решено использовать степень структурированности данных:

1. Структурированные данные. Как правило, хранятся в реляционных базах данных. Упорядочивают данные на уровне таблиц — например, Excel. От информации, которую можно анализировать в самом Excel, Big Data отличается большим объемом.

2. Частично структурированные. Данные не подходят для таблиц, но могут быть иерархически систематизированы. Под такую характеристику подходят текстовые документы или файлы с записями о событиях.

3. Неструктурированные. Не обладают организованной структурой: аудио- и видеоматериалы, фото и другие изображения.

В качестве главных источников данных принято выделять следующие:

- интернет вещей (IoT) и подключённые к ним устройства [1];
- соцсети, блоги [3], СМИ;
- данные компаний: транзакции, заказы товаров и услуг, поездки на такси и каршеринге, профили клиентов;
- показания приборов: метеорологические станции, измерители состава воздуха и водоёмов, данные со спутников;
- статистика городов и государств: данные о перемещениях, рождаемости и смертности;
- медицинские данные: анализы, заболевания, диагностические снимки.

### **Языки программирования для работы с Big Data**

R. Язык используется для обработки данных, сбора статистики и работы с графикой. Загружаемые модули связывают R с GUI-фреймворками и позволяют разрабатывать утилиты анализа с графическим интерфейсом [10]. Графика может быть экспортирована в популярные форматы и использована для презентаций. Статистика отображается в виде графиков и диаграмм.

Scala. Нативный язык для Apache Spark, используется для анализа данных. Проекты Apache Software Foundation, Spark и Kafka, написаны в основном на Scala.

Python. Обладает готовыми библиотеками для работы с AI, ML и другими методами статистических вычислений: TensorFlow, PyTorch, SKlearn, Matplotlib, Scipy, Pandas. Для обработки и хранения данных существуют API в большинстве фреймворков: Apache Kafka, Spark, Hadoop.

### **Примеры использования Big Data**

Большие данные используют для разработки IT-продуктов. Например, в Netflix прогнозируют потребительский спрос с помощью предиктивных моделей для новых функций онлайн-кинотеатра. Специалисты стриминговой платформы классифицируют ключевые атрибуты популярности фильмов и сериалов, анализируют коммерческий успех продуктов и фич. На этом построена ключевая особенность подобных сервисов - рекомендательные системы, предсказывающие интересы пользователей.

В геймдеве используют большие данные для вычисления предпочтений игроков и анализа поведения в видеоиграх. Подобные исследования помогают совершенствовать игровой опыт и схемы монетизации.

Для любого крупного производства Big Data позволяет анализировать доходы и обратную связь от заказчиков, детализировать сведения о цепочках производства и логистике. Подобные факторы улучшают прогноз спроса, сокращают расходы и простои.

Big Data помогает со слабо структурированными данными о запчастях и оборудовании. Записи в журналах и сведения с датчиков могут быть индикаторами скорой поломки. Если её вовремя предсказать, это повысит функциональность, срок работы и эффективность обслуживания техники.

В сфере торговли анализ больших данных даёт глубокие знания о моделях поведения клиентов. Аналитика информации из соцсетей и веб-сайтов улучшает качество сервиса, повышает лояльность и решает проблему оттока покупателей.

В медицине Big Data поможет с анализом статистики использования лекарств, эффективности предоставляемых услуг, с организацией работы с пациентами.

В банках используют распределенные вычисления для работы с транзакционной информацией, что полезно для выявления мошенничества и улучшения работы сервисов.

Госструктуры анализируют большие данные для повышения безопасности граждан и совершенствования городской инфраструктуры, улучшения работы сфер ЖКХ и общественного транспорта.

## **Выводы**

В заключение следует отметить, что развитие технологий обработки больших данных открывают широкие возможности для повышения эффективности различных сфер человеческой деятельности: медицины, транспортного обслуживания, государственного управления, финансов, производства. Именно это и определяет интенсивность развития данного направления в последние годы.

## **Литература**

1. Ын, А. Теоретический минимум по Big Data. Всё что нужно знать о больших данных / А. Ын, А. К. Су. - СПб.: Питер, 2019. - 208 с.: ил.
2. Кукьер, К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим / К. Кукьер, В. Майер-Шенбергер. - М.: ДМК Пресс, 2013. - 240 с.: ил.
3. Уоррен, Дж. Большие данные. Принципы и практика построения масштабируемых систем обработки данных в реальном времени / Дж. Уоррен, Н. Марц. - М.: Вильямс, 2018. - 368 с.: ил.
4. Сенько, А. Работа с BigData в облаках. Обработка и хранение данных с примерами из Microsoft Azure / А. Сенько. - СПб.: Питер, 2019. - 448 с.: ил.
5. Вайгенд, А. BIG DATA. Вся технология в одной книге / А. Вайгенд. - М.: Эксмо, 2021. - 384 с.: ил.
6. O'Reilly Media. Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale / 4th Edition. - V.: O'Reilly, 2015. - 754 с.: ил.
7. Зыков, Р. Роман с Data Science. Как монетизировать большие данные / Р. Зыков. - СПб.: Питер, 2022 - 320 с.: ил.
8. Благирев, А. Big data простым языком / А. Благирев. - М.: АСТ, 2019. - 256 с.: ил.
9. Грас, Д. Data Science. Наука о данных с нуля / Д. Грас. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2021. - 416 с.: ил.
10. Garrett Golemund, Hadley Wickham. R for Data Science. - Sebastopol, CA : O'Reilly, 2017. - 494 с.: ил.

**Баев Д.Э., Коломойцева И.А. Технологии обработки больших данных (Data Mining и Big Data).** В статье описаны термины «Data Mining» и «Big Data», основные критерии определения к какой категории можно отнести имеющуюся информацию, какие языки программирования предназначены для обработки больших данных. Выполнен обзор главных принципов и основных методов работы с большими данными. Проанализированы проблемы, возникающие при анализе и обработке большого объема данных.

**Ключевые слова:** Big Data, Data Mining, обработка больших данных, Volume, Velocity, Variety, Variability, Value, Data Lake, Hadoop.

**Baev D.E., Kolomoitseva I.A. Big data processing technologies (Data Mining and Big Data).** The article describes the terms "Data Mining" and "Big Data", the main criteria for determining which category the available information can be attributed to, which programming languages are designed for processing big data. The review of the main principles and basic methods of working with big data is carried out. The problems arising during the analysis and processing of a large amount of data are analyzed.

**Keywords:** Big Data, Data Mining, big data processing, Volume, Velocity, Variety, Variability, Value, Data Lake, Hadoop.

## Библиотека визуализации результатов иерархической агломеративной кластеризации

С.А. Ермоченко<sup>\*1</sup>, Н.А. Ильина<sup>\*2</sup>

<sup>\*1</sup> к.ф.-м.н, доцент, Витебский государственный университет имени П.М. Машерова,  
yermochenkosa@vsu.by

<sup>\*2</sup> магистрант, Витебский государственный университет имени П.М. Машерова,  
nadya\_ilina\_1998@mail.ru

*Ермоченко С.А., Ильина Н.А. Библиотека визуализации результатов иерархической агломеративной кластеризации. Рассматривается разработка библиотеки на языке программирования JavaScript, которая позволяет выполнять построение дендрограммы, представляющей собой графическое изображение иерархической структуры результатов кластеризации.*

**Ключевые слова:** иерархическая агломеративная кластеризация, дендрограмма, язык программирования JavaScript, JSON.

### Введение

Методы кластеризации являются важным инструментом обработки больших объёмов данных, так как позволяют существенно сократить объём анализируемых данных за счёт выявления близких по характеристикам объектов предметной области и замены множества объектов одним обобщающим объектом.

Одним из самых часто используемых методов является метод k средних, основным достоинством которого является высокая скорость работы. Но данный метод удобно применять, когда исследователю хорошо известна специфика анализируемых данных. Данный метод предполагает определение количества кластеров заранее, перед выполнением процедуры кластеризации. Если обрабатывается большой массив данных, но количество кластеров выбрано специалистом не удачно, то для некоторых особенных объектов, характеристики которых выбиваются из общей совокупности, их специфика может быть исключена из рассмотрения из-за ошибочного включения этих объектов в кластеры, в которых большинство объектов такой спецификой не обладают [1].

При обработке данных о предметной области, особенности которой исследователю не известны или известны не в полной мере, когда заранее предположить количество кластеров является затруднительно, удобнее применять метод иерархической кластеризации, применяющих подход в постепенном формировании иерархической структуры вложенных друг в друга кластеров [2].

Агломеративная кластеризация строит иерархию кластеров начиная с мелких кластеров (изначально рассматриваются кластеры, в каждый из которых включается ровно один исходный объект), далее эти кластеры объединяются в более крупные кластеры до тех пор, пока все кластеры не будут объединены в один единый кластер. Самым сложным этапом является анализ результатов кластеризации. Когда количество объектов в иерархической структуре будет достаточно большим, а сама иерархическая структура становится сложной и разветвлённой, то выбор уровня, на котором можно кластер из некоторого количества объектов рассматривать как совокупность одинаковых объектов.

Для более удобного анализа выполняется построение дендрограммы. Дендрограмма – это графическое изображение иерархической структуры кластеров, представляющей собой дерево, листовыми узлами которого являются исходные объекты, не листовыми узлами – сформированные кластеры (на дендрограмме изображаются точками). Важной особенностью дендрограммы является расстояние между кластерами различных уровней иерархии, которое должно быть пропорциональным условному расстоянию между самими кластерами, которое в свою очередь зависит от выбранной исследователем метрики степени схожести рассматриваемых объектов.

Построение дендрограмм – это стандартный функционал специализированного программного обеспечения (например, Statistica) или библиотек языков программирования, специализирующихся на обработке данных с помощью математических методов, таких как R или Python. Но такие дендрограммы, как правило, сложно стилизовать, т.е. изменять внешний вид дендрограммы и направление самого дерева иерархии. Также стандартные инструменты визуализируют сразу всё дерево, что делает дендрограмму слишком громоздкой и практически нечитаемой при большом количестве элементов. Как правило, уже при 100 исходных объектах

листовые узлы на такой дендрограмме практически невозможно рассмотреть. Что уж говорить про выборку из нескольких тысяч объектов.

Удобным решением является визуализация дендрограммы в виде интерактивного изображения, поддерживающего скрытие и отображение отдельных ветвей иерархического дерева. Такой подход позволяет облегчить работу с деревом, содержащим большое количество объектов, оптимизировать использование памяти для отображения дендрограммы с большим количеством объектов за счёт возможности постепенной подгрузки в память отдельных ветвей дендрограммы при отображении требуемой ветви.

Таким образом, целью данной работы является разработка библиотеки для визуализации иерархического дерева, содержащего результаты агломеративной иерархической кластеризации.

В качестве средства реализации библиотеки выбран язык программирования JavaScript. Это основной язык программирования для разработки клиентской части веб-приложений, которые, в свою очередь, являются наиболее востребованным типом приложений в современных информационных технологиях.

К разрабатываемой библиотеке предъявляются требования гибкой настройки визуального отображения дендрограмм, такие как цвета, толщина и стили линий, направление дендрограммы (вертикально сверху вниз, вертикально снизу вверх, горизонтально слева направо, горизонтально справа налево), количество отображаемых по умолчанию уровней иерархии, шрифтов для подписей листовых узлов и т.д.

Существующих библиотек с подобным функционалом авторами найдено не было. Поэтому работа представляется авторами актуальной в сфере обработки больших объёмов информации.

### **Результаты и их обсуждение**

Рассмотрим формальную постановку задачи кластеризации:

Пусть  $X$  – множество объектов,  $Y$  – множество номеров (названий, меток) кластеров. Задана функция расстояния между объектами  $p(x, x')$ . Имеется конечная обучающая выборка объектов  $X^m = (x_1, \dots, x_m) \subset X$ . Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких в смысле метрики  $p$ , а объекты разных кластеров существенно отличались. При этом каждому объекту  $x_i \in X^m$  приписывается номер кластера  $y_i$ .

Алгоритм кластеризации – это функция  $\alpha: X \rightarrow Y$ , которая любому объекту  $x \in X$  ставит в соответствие номер кластера  $y \in Y$ . Множество  $Y$  в некоторых случаях известно заранее (например, для алгоритма  $k$  средних), однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Кластеризация (обучение без учителя) отличается от классификации (обучения с учителем) тем, что метки исходных объектов  $y_i$  изначально не заданы, и даже может быть неизвестно само множество  $Y$ .

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин:

- не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд эвристических критериев, а также ряд алгоритмов, не имеющих четко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты.

- число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием.

- результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом.

Введём определение иерархической кластеризации – это совокупность методов и алгоритмов, которые направлены на создание иерархии (дерева) вложенных объектов. Существует два вида иерархической кластеризации:

- агломеративная – дерево создается от листьев к стволу, более мелкие кластеры объединяются в более крупные;

- дивизионная – дерево создается от ствола к листьям, более крупные кластеры делятся на более мелкие.

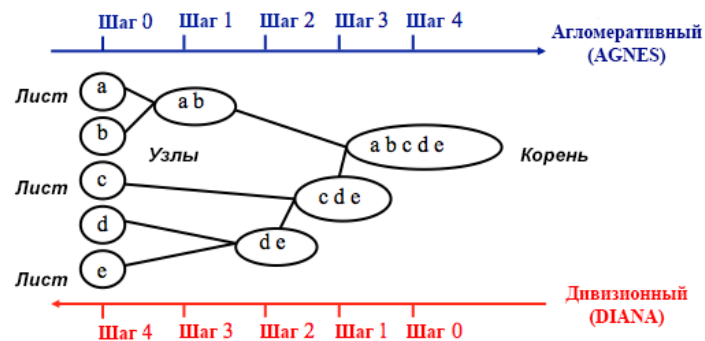


Рисунок 1 – Иерархическая кластеризация

В данной работе рассматривается агломеративная иерархическая кластеризация. Для объединения кластеров используется метод нахождения минимального расстояния между кластерами. В качестве метрики использовалась обычное Евклидово расстояние. Однако следует оговориться, что и способ вычисления расстояния между исходными объектами, и способ определения расстояния между кластерами не влияют на способ визуализации результата кластеризации. Мы приводим сведения об использованных метриках лишь для определённости, чтобы показать, на каких данных проверялась работа разработанной библиотеки.

Для демонстрации работы JavaScript-библиотеки была выполнена кластеризация записей из учебной базы данных, хранящей данные о книгах некоторой библиотеки и их использовании читателями. Книги сравнивались по различным критериям, таким как общее время использования книги, количество читателей, использовавших книгу, количество страниц в книге, автор книги (при вычислении расстояния по данному критерию расстояние между книгами одного и того же автора полагалось равным 0, а расстояния между книгами разных авторов – 1). Каждый критерий при вычислении расстояния между книгами рассматривался как координата в  $N$ -мерном пространстве, где  $N$  – количество критериев. В нашем примере  $N = 4$ . Расстояние, как было указано выше, вычисляется как Евклидова метрика, при этом значение каждого критерия приводится к безразмерному виду в диапазоне  $[0, 1]$ .

Результат кластеризации для разработанной библиотеки представляется в формате JSON (см. рис. 2). Библиотека позволяет загружать эти данные с HTTP-сервера.

```
{
  "name": "albert,pushkin,dostoevsky,sting,sira",
  "distance": 1.7976931348623157E308,
  "children": [ {
    "name": "albert",
    "distance": 0.0,
    "children": [ ]
  }, {
    "name": "pushkin,dostoevsky,sting,sira",
    "distance": 0.911895637720396,
    "children": [ {
      "name": "pushkin,dostoevsky",
      "distance": 0.3634262841795878,
      "children": [ {
        "name": "pushkin",
        "distance": 0.0,
        "children": [ ]
      }, {
        "name": "dostoevsky",
        "distance": 0.0,
        "children": [ ]
      }
    ]
  }, {
    "name": "sting,sira",
    "distance": 1.7976931348623157E308,
    "children": [ {
      "name": "sting",
      "distance": 0.0,
      "children": [ ]
    }, {
      "name": "sira",
      "distance": 0.0,
      "children": [ ]
    }
  ]
} ]
}
```

Рисунок 2 – Результат кластеризации в формате JSON

На данном рисунке для упрощения восприятия не отображаются исходные свойства объектов, на основе значений которых выполнялся расчёт расстояний.

Для визуализации графических примитивов и интерактивного взаимодействия с ними использовалась библиотека D3.js

В данном приложении используется стандартное чтение из файла с помощью языка программирования JavaScript, после загрузки данных из JSON-файла, они передаются в функцию для отображения данных



loadData(). Первый шаг создание графики на SVG:

```
var svg = d3.select("body").append("svg")
  .attr("width", width).attr("height", height).append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

Далее создаем нашу иерархию, с помощью которой будем отрисовывать дерево:

```
root = d3.hierarchy(treeData, function(d) {
  return d.children;
});
root.x0 = height / 2;
root.y0 = 1500;
```

Передаем иерархию в функцию update() для интерактивной работы с данными:

1. Задаем координаты нашим кластерам и нормализуем значения (для красивого отображения):

```
var treeData = treemap(root);
var nodes = treeData.descendants(),
    links = treeData.descendants().slice(1);
nodes.forEach(function(d) {
  d.y = d.data.distance * 100
});
```

2. Обновляем положение кластеров на странице при открытии/закрытии родительского кластера:

```
var node = svg.selectAll('g.node')
  .data(nodes, function(d) {
    return d.id || (d.id = ++i);
  });
var nodeEnter = node.enter().append('g').attr('class', 'node')
  .attr("transform", function(d) {
    return "translate(" + source.y0 + "," + source.x0 + ")";
  }).on('click', click);
```

3. Задаем внешний вид (добавляем кружок и текст):

```
nodeEnter.append('circle').attr('class', 'node').attr('r', 1e-6)
  .style("fill", function(d) {
    return d._children ? "lightsteelblue" : "#fff";
  });
nodeEnter.append('text').attr("dy", ".35em")
  .attr("x", function(d) {
    return d.children || d._children ? -13 : 13;
  })
  .attr("text-anchor", function(d) {
    return d.children || d._children ? "end" : "start";
  })
  .text(function(d) {
    return d.data.name;
  });
```

4. Отрисовка диагонали (линии), которая соединяет кластеры:

```
function diagonal(s, d) {
  return `M ${s.y} ${s.x} C ${(s.y + d.y) / 2} ${s.x},
    ${(s.y + d.y) / 2} ${d.x}, ${d.y} ${d.x}`
}
```

Дополнительные методы для кликабельности кластеров и корректного отображения кластеров и линий, здесь не приведены ввиду громоздкости.

Результат работы показан на рисунке 3

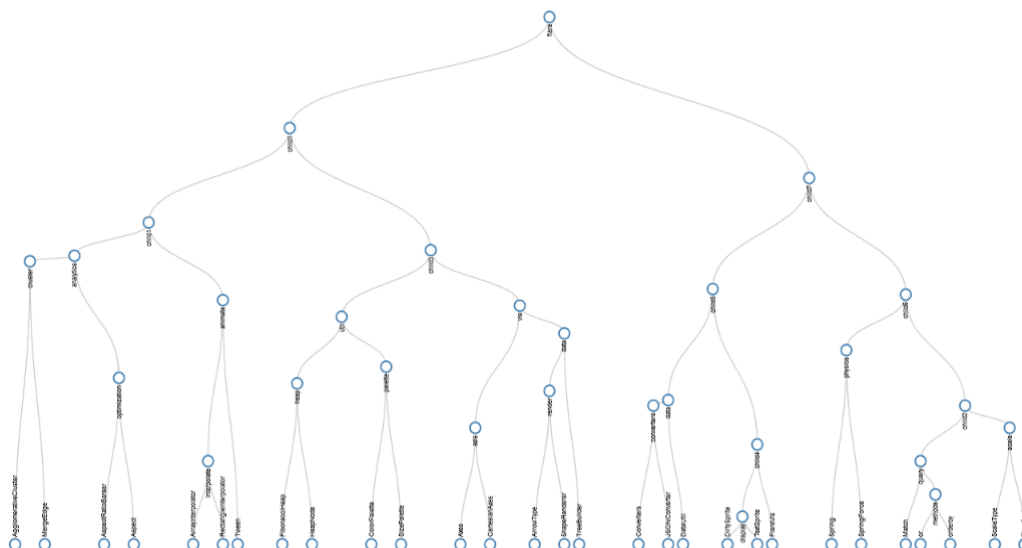


Рисунок 3 – Визуализация иерархической кластеризации

В дальнейшем планируется доработка разработанной библиотеки, добавление настроек визуализации, поддержка подгрузки данных по мере раскрытия уровней иерархии, направление дерева иерархии и т.д.

Данная библиотека используется для анализа данных об успеваемости студентов факультета математики и информационных Витебского государственного университета имени П. М. Машерова, выполняемого в рамках научно-исследовательской работы «Методы искусственного интеллекта для оптимизации образовательного процесса, №ГР 20210790» в рамках задания «Информационные технологии повышения качества образовательного процесса» государственной программы научных исследований «Цифровые и космические технологии, безопасность человека, общества и государства», подпрограммы «Цифровые технологии и космическая информатика» на 2021-2025 годы

### Литература

1. Мандель, И. Д. Кластерный анализ / И. Д. Мандель. – Москва: Финансы и статистика, 1988. – 176 с.
2. Жамбю, М. Иерархический кластер-анализ и соответствия / М. Жамбю. – Москва: Финансы и статистика, 1988. – 345 с.

**Ермоченко С.А., Ильина Н.А. Библиотека визуализации результатов иерархической агломеративной кластеризации.** Рассматривается разработка библиотеки на языке программирования JavaScript, которая позволяет выполнять построение дендрограммы, представляющей собой графическое изображение иерархической структуры результатов кластеризации.

**Ключевые слова:** иерархическая агломеративная кластеризация, дендрограмма, язык программирования JavaScript, JSON.

**Yermochenko Sergey, Ilyina Nadezhda. Library for visualization of hierarchical agglomerative clustering results.** The article considers the development of the library for JavaScript programming language, which allow building a dendrogram as graphical image of hierarchical structure of clustering results.

**Key words:** hierarchical agglomerative clustering, dendrogram, JavaScript programming language, JSON.

## **СЕКЦИЯ 5. «ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ, СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ»**

УДК 004.051

### **Перспективы применения ГИС как средств геолокации**

**В.В. Дорошенко<sup>\*1</sup>, А.В. Григорьев<sup>\*2</sup>**

<sup>\*1</sup> бакалавриат, Донецкий национальный технический университет,  
vladd267@gmail.com

<sup>\*2</sup> к.т.н., доцент, Донецкий национальный технический университет,  
grigorievalvl@gmail.com

**Дорошенко В.В., Григорьев А.В. Перспективы применения ГИС как средств геолокации.**  
*В данной статье рассмотрены проблемы и перспективы развития ГИС, как картографических систем. Было проанализировано текущее состояние данной отрасли и сделан вывод о перспективах дальнейшего роста. Для решения данных проблем были рассмотрены передовые технологии создания ГИС, содержащих данные с больших территорий. Была рассмотрена оптимальная модель организации хранения и обработки данных больших ГИС. Проведен сравнительный анализ двух популярных поисково-информационных картографических сервисов: «Яндекс Карты» и «Google Maps».*

**Ключевые слова:** ГИС, картография, маршрут, анализ, Google Maps, Яндекс Карты, Контурно-комбинированный метод, Стереотопографический метод.

#### **Введение**

Геоинформационные системы давно стали частью нашей жизни. Всем известные Google Maps вышли на свет в далеком 2005 году. В свою очередь Яндекс Карты были запущены еще раньше – в 2004 году. Сегодня трудно представить свою жизнь без данных служб. Данные сервисы хранят и анализируют огромные объемы данных: ситуацию на дороге, расписание транспорта, информацию со спутников и так далее. Они стали основными помощниками людей в вопросах геолокации, нахождения маршрутов.

Данная тема особенно актуальна в наши дни. Пользовательская база данных сервисов постоянно растет и, в общем, насчитывает уже миллиарды пользователей. Ежедневно многие люди используют подобные сервисы для упрощения жизни: формирования быстрого маршрута на работу, или домой, поиска ближайшего магазина, или нахождения ресторана с большим количеством положительных отзывов.

Целью работы является рассмотрение перспектив и проблем развития геоинформационных систем, как сервисов геолокации и справочных систем, а также анализ технологии их создания и способов хранения и обработки картографических данных.

#### **Анализ состояния**

В данный момент развитие ГИС идет очень быстро, причем скорость развития только растет. Раньше нам было необходимо часами исследовать город, а теперь для нахождения ближайшего банка требуется отсортировать значки по определенной категории и проложить маршрут. Затраты времени на изучение и анализ расположения зданий в городе уходит на второй план.

В настоящий момент времени человечество находится в технологическом укладе, в котором очень важное место занимают геоинформатика и цифровая картография. Сейчас развитие геоинформационных систем идет большими темпами. В данный момент происходит создание инфраструктуры пространственных данных, которая может объединить все существующие на данный момент ГИС технологии. Данное направление развития создает задел для следующего технологического уклада [1]. Как итог, можно считать развитие картографических систем перспективным направлением, которое будет востребовано не только сейчас, но и десятилетия вперед.

В настоящее время популярные картографические сервисы имеют огромное количество функций. Если еще 20 – 30 лет назад понятие «карта» несло за собой изображение, показывающее расположение и название

основных элементов, то сейчас карты – большие информационные системы. Сегодня карты могут не только прокладывать маршрут, но и высчитывать время поездки, мало того, они могут использовать расписание общественного транспорта для расчета оптимального времени для начала маршрута. Карты сегодня – большие справочники, в которых можно найти практически любую организацию, узнать до нее маршрут, что она собой представляет, время работы, отзывы, контактную информацию и многое другое, для чего потребовалось только написать название и выбрать нужное здание из списка.

Поисково-информационная картографическая служба – одно из популярнейших ответвлений ГИС. Аудитория сервиса Яндекс Карты уже в 2017 году достигла 24.9 миллиона пользователей [2]. А посещаемость Google Maps еще в далеком 2012 году достигла более 2 миллиардов посещений в месяц [3]. Очевидно, что на текущий момент, данные значения многократно возросли.

### **Анализ характерных представителей**

В данный момент существует огромное количество подобных сервисов. Среди известных сервисов можно назвать следующие картографические системы: Maps.Me, 2ГИС, Яндекс Карты, Google Maps, OpenStreetMap, Bing Maps.

Как уже было сказано, существует два лидера в области картографических служб: Яндекс Карты и Google Maps. Они предоставляют пользователям сопоставимый набор услуг, но все же имеют свои преимущества и недостатки.

Начиная с общих моментов, обе системы позволяют находить необходимые места на карте, строить маршрут до них. Имеется возможность строить маршрут, проходящий через определенные места. Это позволяет рассчитать оптимальный маршрут, если необходимо посетить множество мест одно за другим. Обе системы дают возможность сохранять точки на карте. Благодаря этому можно быстро строить маршрут до них, или между ними. Также данные системы имеют возможность просмотра карт в разных режимах: «Стандартный», «Спутник», «Рельеф». К тому же, данные системы имеют большой набор дополнительных функций. Службы также имеют возможность вывести фрагмент карты на печать, что может быть полезно при отсутствии гаджетов при себе. Уже было сказано про возможность фильтрации отображения значков по категориям, но также можно отобразить маршруты общественного транспорта, информацию о пробках и т.д. Оба сервиса также предоставляют доступ к интересной функции «Просмотр улиц». Данная функция позволяет пользователям загружать на карту панорамные снимки местности. С одной стороны, такая функция кажется бесполезной, но она имеет очень важную роль. Благодаря этой функции, человек может посмотреть на незнакомую страну, или город «от первого лица». Данная функция нужна не только для съемки улиц, или, может, достопримечательностей. Она полезна для съемки, допустим, музеев. Эта функция расширяет спектр применения карт до нового уровня. Теперь можно посетить другую страну, а в большинстве крупных городов можно обойти чуть ли не каждую улицу. В компании Яндекс пошли дальше и представили несколько интересных и очень полезных функций. Во-первых, карта может работать в 3D режиме под углом 45°, что позволяет рассмотреть любое здание со всех сторон и оценить его высоту относительно других. Google Maps имеют похожую функцию, но эта функция работает только в немногих городах США. В свою очередь, Яндекс сгенерировал 3D модель для всех отображаемых на карте объектов (включая достопримечательности).

На данном изображении видно, что была сгенерирована модель Эйфелевой башни в правильных пропорциях, что можно увидеть, сравнив ее высоту с другими зданиями, для которых модель была воссоздана аналогичным образом. Также Яндекс представил полезную функцию в сфере общественного транспорта. Данный сервис позволяет отслеживать местоположение всех автобусов практически во всех городах РФ и Республики Беларусь.

На представленном изображении видно, что была сгенерирована модель Эйфелевой башни в правильных пропорциях, что можно увидеть, сравнив ее высоту с другими зданиями, для которых модель была воссоздана аналогичным образом. Также Яндекс представил полезную функцию в сфере общественного транспорта. Данный сервис позволяет отслеживать местоположение всех автобусов практически во всех городах РФ и Республики Беларусь.

Пример генерации 3D модели показан на рис. 1.

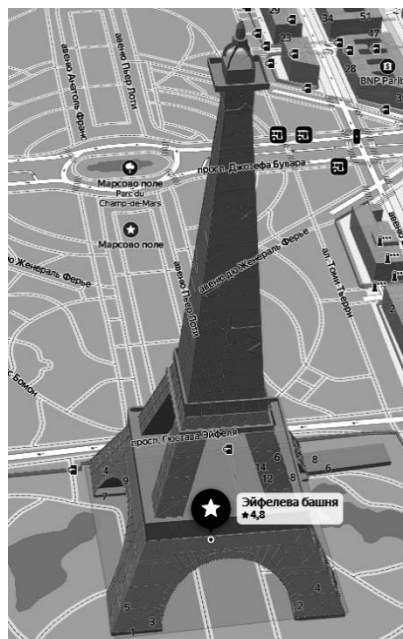


Рисунок 1 – Пример генерации 3D модели Эйфелевой башни

В свою очередь, сервис от Google имеет функцию просмотра хронологий перемещений пользователя. Информация получается достаточно полной в связи с тем, что данная компания ведет разработку системы Android, что предоставляет большой доступ для их приложений, а в связи с тем, что телефон – предмет, который человек держит при себе чаще всего – хронология не будет иметь белых пятен.

Однако данные сервисы не лишены проблем. Основная из них – неполнота данных. Google Maps долгое время не имела карт российских городов (кроме Москвы и Санкт-Петербурга), что на тот момент делало невозможным использование данной ГИС на территории России. В данный момент названная проблема уже исправлена, однако, не в полной мере. Многие функции не работают на территории России или имеют ограниченный функционал: информация о пожарах доступна только на территории США, информация о велодорожках недоступна на территории СНГ. В свою очередь данные о маршруте общественного транспорта представлены только на территории крупных городов, но информация о расписании доступна практически для каждой его остановки.

Проблема неполноты данных особенно выражена в сервисе Яндекс. Практически все функции, которые может предложить система, работают в полной мере на всей территории СНГ. Но весь функционал пропадает при необходимости поиска на территории других стран. Сервис без особых проблем генерирует 3D модель любых зданий, представленных на карте, допустим. Но на карте присутствует слишком большое количество белых пятен. На территории СНГ они представлены только в малозаселенных населенных пунктах. В остальных государствах – белые пятна преобладают над заполненными местами. Это хорошо заметно при сравнении стандартного режима работы и режима «Спутник». Сравнение данных режимов работы на примере густо застроенного города можно увидеть на рис. 2. Для сравнения был выбран один из районов Лондона.

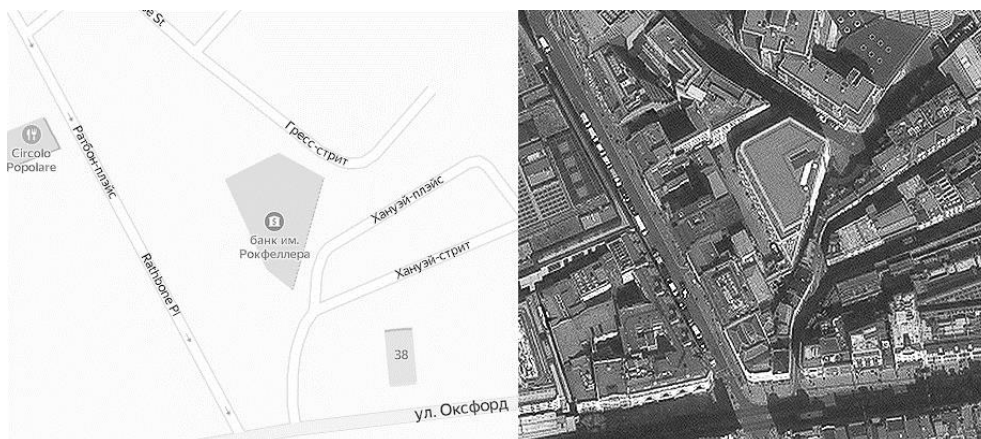


Рисунок 2 – Сравнение режимов: стандартный и «Спутник»

На данном изображении показано, что карта содержит информацию только о трёх зданиях, но в реальности их на много больше. Для примера был выбран один из крайних районов города, но данная проблема встречается

повсеместно. С другой стороны, данный сервис не пользуется популярностью у иностранцев, так как у них есть альтернатива. Данный факт практически нивелирует данную проблему, так как главные здания и достопримечательности на карте обозначены.

Как итог, можно сказать, что Google больше нацелен на Америку и Европу, хотя предоставляет информацию о странах СНГ. В свою очередь, Яндекс в большей степени работает в странах СНГ. Для того, чтобы быть универсальными, обоим сервисам не хватает картографической информации. Для исправления данной ситуации возможно использование информации от сторонних сервисов, предоставляющих подобного рода данные.

### **Обзор существующей технологии создания карт**

Для создания карты местности должно быть получено и обработано большое количество информации. На ее обработку уходит больше количество времени, из-за чего необходимо выбрать рабочий, а главное – эффективный метод работы с данными. Обычно применяется векторизация – процесс преобразования растрового изображения в векторное. Данный процесс достаточно трудоемкий и требует большого количества вычислительных мощностей. Обработка изображения – сложный с точки зрения принципа работы и организации алгоритмов процесс.

В общем случае, векторизация делится на следующий набор шагов:

1. Последовательно просматривается каждый объект на изображении;
2. Для каждого выделяемого объекта необходимо определить его тип и специфичные для его типа параметры;
3. В зависимости от типа объекта выбираются и применяются алгоритмы выделения объекта из изображения, его идентификации, а затем векторизации.

Процесс повторяется, пока не будут выделены все объекты из исходного изображения [4].

Часто для ускорения обработки используют натренированные самообучающиеся системы, которые позволяют исправить недочеты, полученные в результате обработки, например, склеивание близко расположенных зданий. Данный подход позволяет уменьшить точность работы основного алгоритма, так как большинство ошибок, обусловленных низкой точностью, будет устранено.

Если необходимо получить данные о рельефе – необходимо узнать высоты объектов. Для этого применяется два метода: контурно-комбинированный и стереотопографический.

Контурно-комбинированный метод подразумевает под собой непосредственное присутствие человека на данной местности и ручное снятие показаний, используя геодезические приборы, самых важных точек поверхности [5]. Данный способ неприменим при формировании карт большого объема, так как требует значительных затрат времени и людей для снятия показаний. Даже с учетом автоматизации, данный процесс займет слишком много времени даже в условиях одного города, не говоря о создании ГИС, объединяющую весь земной шар.

Стереотопографический метод, напротив, подразумевает под собой частичное перекрытие друг другом нескольких снимков таким образом, чтобы на каждом из них изображался один и тот же участок местности. В стереоскоп данный участок местности выглядит как трехмерное изображение. Далее, используя данную модель, с помощью приборов определяют высоты точек местности [5]. Чем больше изображений будет сделано – тем точнее будет результат. Данный метод позволяет создать достаточно точную модель здания простой формы. Преимущество данного метода над предыдущим в том, что данный метод возможно применить, используя спутники. Данный факт позволяет исключить личное присутствие человека. Так как нет необходимости в присутствии человека на каждом участке местности, данный способ подразумевает гораздо меньшие затраты времени. К тому же, использование нескольких снимков под разными углами позволяет использовать интеллектуальные системы для автоматической генерации карт.

Дополнительное преимущество использования спутников в том, что они позволяют создавать снимки в разных световых спектрах (видимых и невидимых). Информация, полученная данным способом очень полезна, так как позволяет получать больше информации о местности. Данный метод позволяет следить за ростом сельскохозяйственных культур, выявлять пожары (что умеет делать сервис Google Maps).

### **Способы хранения и обработки данных в ГИС**

Анализируя принцип работы ГИС, можно прийти к выводу, что стандартные методы хранения данных в СУБД, а конкретно, механизм индексирования, не являются эффективными. Данный механизм позволяет решить проблему доступа в общем случае. Среди многочисленных путей развития механизмов индексации ГИС можно выделить путь, представленный в [6].

Проблема может быть решена вводом хэш-функции. Благодаря этому увеличивается скорость обработки запросов. Однако, запросы могут быть разделены на 3 группы:

- пространственный;
- атрибутивный;

– смешанный.

Пространственный запрос производится при необходимости вывода данных на экран. Атрибутный запрос производится при необходимости получения атрибутной информации об объекте. В свою очередь, смешанный запрос объединяет оба этих запроса.

Данное деление затрудняет использование хэш-таблиц, так как требуется разделить информацию на две базы данных: база с пространственной информацией и база с атрибутной информацией. В данном случае при использовании хэш-функций, порядок записей в них будет отличаться. Для сохранения связи между пространственной и атрибутной информацией требуется ввести индекс, сохраняющий связь между двумя таблицами. В данном случае возникает проблема выбора одной из двух баз для сохранения индекса без потерь в скорости. Использование индекса замедляет доступ к данным в базе, которая его использует.

В связи со спецификой работы современных ГИС, пространственный запрос будет выполняться чаще всего, к тому же он один из самых трудоемких для обработки. Атрибутный запрос будет использоваться реже, из-за чего он имеет меньший приоритет. К тому же, база данных атрибутов потенциально меньше и проще организована, чем пространственная. В данном случае более целесообразно использовать индекс для работы в атрибутной таблице. Итоговый вариант организации баз данных можно увидеть на рис. 3.

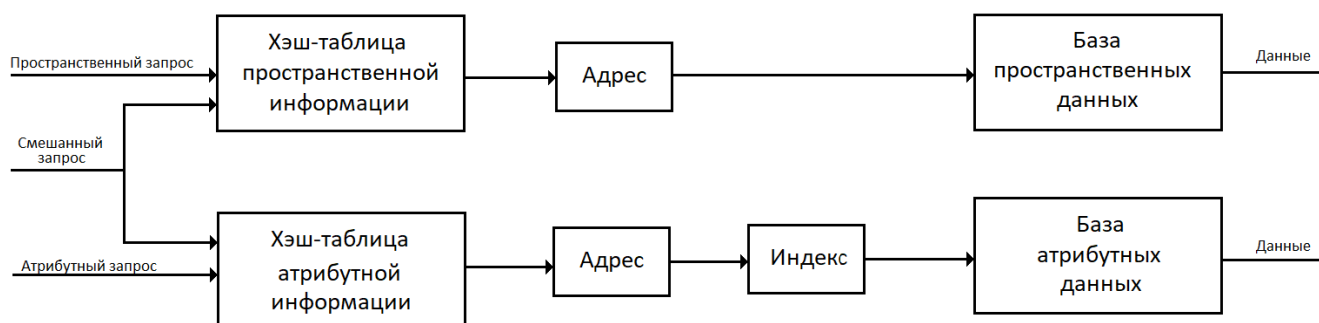


Рисунок 3 – Структура высокоскоростной ГИС ориентированной СУБД

## Вывод

В данной работе проанализированы перспективы развития ГИС, как картографических служб. В результате обнаружено то, что данным сервисам необходимо развиваться на территории большего количества стран. Это откроет данные службы для новой аудитории. Рассмотрены и проанализированы достоинства и недостатки двух самых популярных сервисов. В результате анализа выявлено, что Яндекс Карты преимущественно подходят для использования в странах СНГ, в свою очередь, Google Maps удобны для использования в любой точке мира, однако имеют небольшие пробелы в информации, особенно характерные для стран СНГ. Рассмотрены передовые способы создания карт местности. Описан способ хранения и обработки данных, который сохраняет оптимальность хранения, а также имеет высокую скорость выполнения запросов.

## Литература

1. Лисицкий, Д. В. Перспективы развития картографии: от системы «цифровая земля» к системе виртуальной геореальности / Лисицкий, Д. В. // Вестник СГУГиТ (Сибирского государственного университета геосистем и технологий). – 2013. – No 2. – С. 8-16.
2. Яндекс Карты. [Электронный ресурс]. – Режим доступа: URL: [https://ru.wikipedia.org/wiki/Яндекс\\_Карты](https://ru.wikipedia.org/wiki/Яндекс_Карты).
3. Карты Google. [Электронный ресурс]. – Режим доступа: URL: [https://ru.wikipedia.org/wiki/Карты\\_Google](https://ru.wikipedia.org/wiki/Карты_Google).
4. Андриющенко, А. А. Векторизация географических карт / Андриющенко, А. А. // Известия Южного федерального университета. Технические науки. – 1998. – No 1. – С. 199-205.
5. Как создаются карты. Блог компании VK. [Электронный ресурс]. – Режим доступа: URL: <https://habr.com/ru/company/vk/blog/406521>.
6. Андриющенко, А. А. Оптимальная по времени обработки запросов ГИС / Андриющенко, А. А. // Известия Южного федерального университета. Технические науки. – 1998. – No 1. – С. 205-208.

**Дорошенко В.В., Григорьев А.В. Перспективы применения ГИС как средств геолокации.**

*В данной статье рассмотрены проблемы и перспективы развития ГИС, как картографических систем. Было проанализировано текущее состояние данной отрасли и сделан вывод о перспективах дальнейшего роста. Для решения данных проблем были рассмотрены передовые технологии создания ГИС, содержащих данные с больших территорий. Была предложена*

оптимальная модель организации хранения и обработки данных больших ГИС. Проведен сравнительный анализ двух популярных поисково-информационных картографических сервисов: «Яндекс Карты» и «Google Maps».

**Ключевые слова:** ГИС, картография, маршрут, анализ, Google Maps, Яндекс Карты, Контурно-комбинированный метод, Стереотопографический метод.

**Doroshenko V.V., Grigoriev A.V. Prospects for the use of GIS as a means of geolocation.** This article discusses the problems and prospects for the development of GIS as cartographic systems. The current state of the industry was analyzed and a conclusion was made about the prospects for further growth. To solve these problems, advanced technologies for creating GIS containing data from large areas were considered. An optimal model for organizing storage and processing of large GIS data was proposed. A comparative analysis of two popular search-information mapping services: "Yandex Maps" and "Google Maps" was carried out.

**Keywords:** GIS, cartography, route, analysis, Google Maps, Yandex Maps, Contour-combined method, Stereotopographic method.



## Проблематика разработки современных систем управления беспилотных автомобилей

В.К. Ремизов <sup>\*1</sup>, А.В. Григорьев <sup>\*2</sup>

<sup>\*1</sup> студент, Донецкий национальный технический университет,  
vsevolod.remizov@gmail.com

<sup>\*2</sup> к.т.н, доцент, Донецкий национальный технический университет,  
grigorievalvl@gmail.com

*Ремизов В.К., Григорьев А.В. Проблематика разработки современных систем управления беспилотных автомобилей. В статье описаны ключевые моменты разработки систем управления беспилотных автомобилей. Выполнен обзор существующих систем. Проанализированы проблемы, возникающие при разработке, определены возможные пути их решения.*

**Ключевые слова:** беспилотные автомобили, разработка, управление, функциональность

### Введение

Беспилотные автомобили уже сегодня стали частью нашей жизни. Они применяются в совершенно разных областях – сервисы такси, перевозки грузов и даже наблюдение за важными объектами [1].

Разработкой беспилотных автомобилей занимаются многие крупные автопроизводители, выпуская все более совершенные прототипы. Однако в этой области все еще есть ряд проблем, которые необходимо решить для обеспечения эффективного и безопасного функционирования описанных устройств. Этим и обусловлена актуальность данной работы.

Цель предлагаемой статьи – выполнить обзор проблем разработки систем управления беспилотных автомобилей и способов их решения.

Для достижения поставленной цели необходимо решить следующие задачи:

описать основные бизнес-процессы разработки систем управления беспилотных автомобилей;

провести анализ функциональных возможностей современных систем управления;

выявить основные проблемы разработки систем управления беспилотных автомобилей и определить пути их решения.

Импульсом для написания данной статьи послужила публикация «Анализ функциональных возможностей современных систем управления беспилотных автомобилей и перспективы развития» Аксенова С.А. и Григорьева А.В. [2].

Однако со времени написания статьи многие проблемы, описанные в ней, были решены, например, системы научились распознавать сигналы светофоров и даже начали ездить на дорогах общего пользования, но появились и некоторые другие проблемы – их мы рассмотрим в данной статье. Т.о., второй задачей данной статьи является отслеживание динамики развития систем управления беспилотных автомобилей.

### Описание основных бизнес-процессов разработки систем управления беспилотных автомобилей

Беспилотный автомобиль – это транспортное средство, оборудованное системой автоматического управления, которое может передвигаться без участия человека [3]. Процесс движения беспилотного автомобиля – это непрерывное выполнение следующих подзадач в реальном времени [4]:

- локализация – определение местоположения автомобиля;
- восприятие – идентификация окружающих объектов;
- предсказание – прогнозирование движения других участников дорожного движения;
- планирование – определение собственных действий автомобиля.

Принцип работы беспилотного автомобиля представлен на рис. 1.

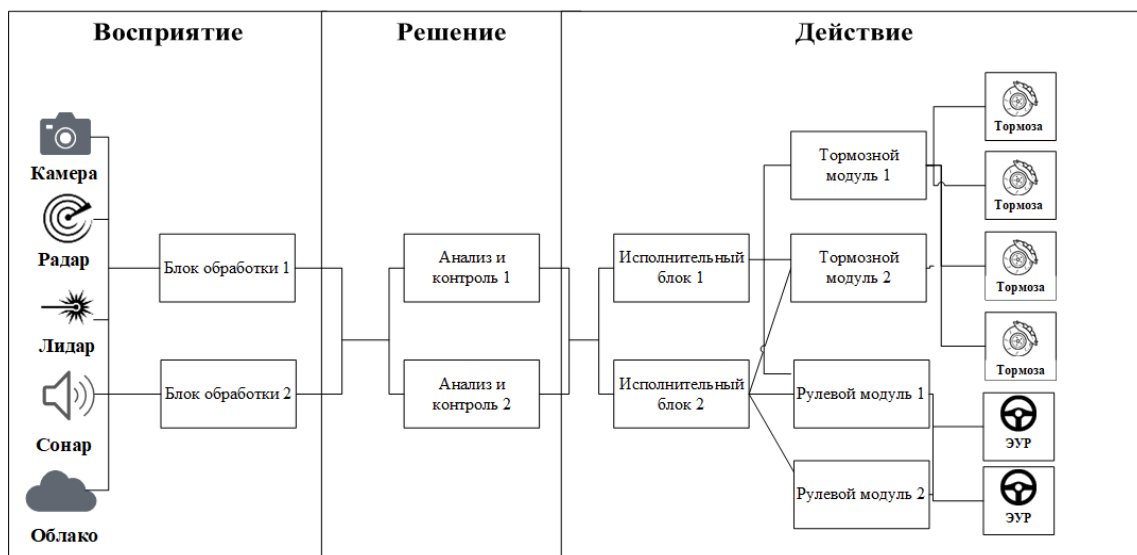


Рисунок 1 – Принцип работы беспилотного автомобиля

Основные компоненты системы представлены на рис. 2.

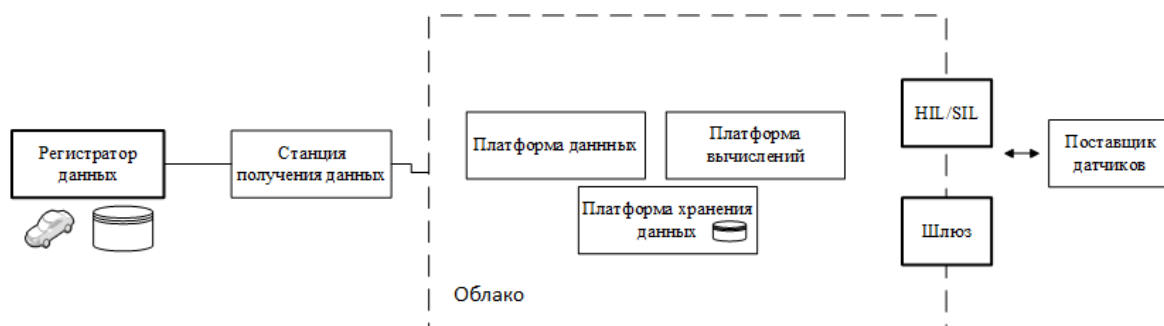


Рисунок 2 – Основные компоненты системы

Структура системы представлена на рис. 3.

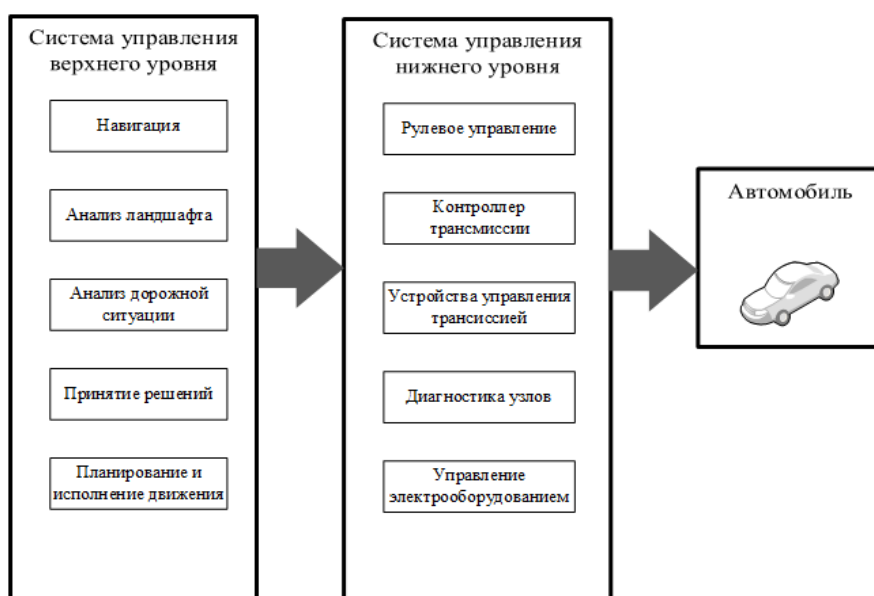


Рисунок 3 – Структура системы

Проанализировав структуру современной системы управления беспилотного автомобиля, можно таким образом определить алгоритм ее работы (см. рис. 4).



Рисунок 4 – Алгоритм работы системы

### **Анализ функциональных возможностей современных систем управления беспилотных автомобилей**

На сегодняшний день существует множество систем управления беспилотных автомобилей: «Waymo Driver», «Tesla Motors», «Ford Autonomous Vehicles», «Яндекс», «SberAutoTech» и другие. Рассмотрим функциональные возможности некоторых из перечисленных систем.

Так, самой совершенной системой управления беспилотных автомобилей является «Waymo Driver» (см. рис. 5). Беспилотные автомобили, оснащенные данной системой, могут увидеть и проанализировать дорожную обстановку на 500 метров вперед, а интерфейс системы спроектирован так, чтобы пользователь доверял автомобилю. «Waymo Driver» содержит в себе информацию, которая была собрана технологиями «Google», например – «Google карты» и «Google Street View». Однако данная система прокладывает неоптимальные маршруты, т.к. избегает потенциально опасные левые повороты.



Рисунок 5 – Интерфейс «Waymo Driver»

На отечественном рынке разработкой подобных систем занимается компания «Яндекс» (см. рис.6). Их беспилотные автомобили способны сканировать пространство вокруг себя, распознавать другие автомобили, дорожные знаки, разметку, различные препятствия, людей, животных и другие элементы окружающей среды. Однако эта система имеет следующие недостатки: ошибочное прогнозирование поведения людей на пешеходных переходах, отсутствие адаптации под погодные условия.



Рисунок 6 – Интерфейс «Яндекс»

### ***Проблематика разработки систем управления беспилотных автомобилей и возможные методы их разрешения***

После проведенного анализа функциональных возможностей современных систем управления беспилотных автомобилей было выявлено ряд проблем:

Основной проблемой является «слепота» беспилотных автомобилей в плохую погоду. Во время плохих погодных условий точность датчиков снижается, что может вызвать опасные последствия.

Успешным решением данной проблемы может стать использование сверточных нейронных сетей, объединение датчиков в единую систему, а также использование алгоритмов глубокого и взаимного обучения нейронных сетей для обнаружения проходимого пути [5].

Также решить проблему может помочь «умная» дорога, которая может взаимодействовать с автомобилями, контролировать их и давать команды, например, снизить скорость или сгруппироваться в колонну [6].

Следующей проблемой является сложность левых поворотов. Беспилотные автомобили избегают потенциально опасные левые повороты на нерегулируемых перекрестках, из-за чего маршруты прокладываются неоптимально.

Чтобы решить данную проблему, следует использовать радиолокационные радары, которые позволят автомобилю «заглянуть за угол» и облегчат левый поворот. Сигнал от радара будет отражаться от различных объектов, находящихся за углом, и частично возвращаться к источнику. По характеру вернувшегося сигнала можно будет классифицировать объект, измерить его скорость и направление движения [7].

Также серьезной проблемой является сложность взаимодействия с людьми на пешеходных переходах. Это связано с тем, что люди ждут зрительного контакта с водителем для подтверждения того, что он их видит и остановится. Понятно, что беспилотные автомобили подобный зрительный контакт осуществить не могут.

«Google» запатентовал следующую идею по решению данной проблемы – так, компания предлагает устанавливать на беспилотном автомобиле громкоговорители и дисплеи. Такой метод даст возможность подавать пешеходу понятные графические и звуковые сигналы [8].

### ***Выводы***

В статье описаны и проанализированы основные бизнес-процессы разработки систем управления беспилотных автомобилей. Выполнен анализ функциональных возможностей существующих систем. В ходе исследования выявлены и сформулированы проблемы, возникающие при разработке, а также предложены решения по устранению данных проблем. Перспективным направлением исследования является разработка нейронной сети, которая бы анализировала лица прохожих и выдавала графическое изображение глаз для установления аналога зрительного контакта.

### ***Литература***

1. Шаошань Лю, Лиюнь Ли, Цзе Тан, Шуаш Ву, Жан-Люк Годье Л91 Разработка беспилотных транспортных средств / науч. ред. В. С. Яценков; пер. с англ. П. М. Бомбаковой. – М.: ДМК Пресс, 2022. – 246 с.: ил.

2. Аксенов, С. А. Анализ функциональных возможностей современных систем управления беспилотных автомобилей и перспективы развития / С. А. Аксенов, А. В. Григорьев // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016): Сборник научных трудов I научно-практической конференции (студенческая секция), Донецк, 16–17 ноября 2016 года. – Донецк: Донецкий национальный технический университет, 2016. – С. 7-11. – EDN PQNGGG.
3. Беспилотный автомобиль [Электронный ресурс] / Википедия. Свободная энциклопедия. – Режим доступа: [https://ru.wikipedia.org/wiki/Беспилотный\\_автомобиль](https://ru.wikipedia.org/wiki/Беспилотный_автомобиль). – Загл. с экрана.
4. Как устроены беспилотные автомобили и кто их делает [Электронный ресурс] / Auto.ru Журнал. – Режим доступа: <https://mag.auto.ru/article/selfdrivingsetup/>. – Загл. с экрана.
5. Технология беспилотного вождения в плохих дорожных и погодных условиях [Электронный ресурс] / Cognitive Pilot. – Режим доступа: <https://cognitivepilot.com/autotransport1/tehnologiya-bespilotnogo-vozhdeniya-v-plohih-dorozhnyh-i-pogodnyh-usloviyah/>. – Загл. с экрана.
6. Павловский, В. Е. Задачи управления беспилотным автомобилем в проекте "АвтоНИВА" / В. Е. Павловский, В. Н. Огольцов, И. А. Спиридонова // Второй Всероссийский научно-практический семинар "Беспилотные транспортные средства с элементами искусственного интеллекта": Труды семинара, Санкт-Петербург, 09 октября 2015 года / Российская ассоциация искусственного интеллекта. – Санкт-Петербург: Политехника-сервис, 2015. – С. 107-114. – EDN UXHKMH.
7. New automotive radar spots hazards around corners [Электронный ресурс] / Tech Explore. – Режим доступа: <https://techxplore.com/news/2020-06-automotive-radar-hazards-corners.html>. – Загл. с экрана.
8. Pedestrian notifications [Электронный ресурс] / Google Patents. – Режим доступа: <https://patents.google.com/patent/US9196164B1>. – Загл. с экрана.

**Ремизов В.К., Григорьев А.В. Проблематика разработки современных систем управления беспилотных автомобилей.** В статье описаны ключевые моменты разработки систем управления беспилотных автомобилей. Выполнен обзор существующих систем. Проанализированы проблемы, возникающие при разработке, определены возможные пути их решения.

**Ключевые слова:** беспилотные автомобили, разработка, управление, функциональность

**Remizov V.K., Grigoriev A.V. Problems of development of modern control systems for unmanned vehicles.** The article describes the key points in the development of control systems for unmanned vehicles. The review of existing systems is carried out. The problems that arise during development are analyzed, and possible ways to solve them are identified.

**Keywords:** unmanned vehicles, development, control, functionality

## **Текстовый редактор с возможностью обработки документов с неформатированным и форматированным текстом**

**Е. А. Лазуренко<sup>\*1</sup>, Я. А. Рудь<sup>\*2</sup>, А. П. Ищенко<sup>\*3</sup>, А. В. Григорьев<sup>\*4</sup>**

<sup>\*1</sup> бакалавриат, Донецкий национальный технический университет,  
eugenelazurenko@gmail.com

<sup>\*2</sup> бакалавриат, Донецкий национальный технический университет,  
rezan20031@gmail.com

<sup>\*3</sup> ассистент, Донецкий национальный технический университет,  
matrix665@inbox.ru

<sup>\*4</sup> к.т.н., доцент, Донецкий национальный технический университет,  
grigorievalvl@gmail.com

*Е. А. Лазуренко, Я. А. Рудь, А. П. Ищенко, А. В. Григорьев. Текстовый редактор с возможностью обработки документов с неформатированным и форматированным текстом. В статье представлен краткий обзор текстовых редакторов, описание функций, возможностей, классификаций, сравнения популярных систем текстовых редакторов.*

**Ключевые слова:** файл, текстовый редактор, пользователь, форматирование, операционная система, шрифт, функции, классификация.

### **Введение**

Текстовый редактор один из самых распространенных видов компьютерных программ, предназначенный для редактирования и создания текстовых документов. Существует большое множество разнообразного программного обеспечения данного типа. Самыми распространенными и популярными из текстовых редакторов можно назвать текстовый процессор Microsoft Word, бесплатный онлайн офис Google Docs, свободный текстовый редактор Notepad++, кроссплатформенное приложение OfficeSuite, редактор исходного кода Visual Studio Code и огромное количества менее популярных и более узконаправленных систем.

Задача разработки текстового редактора в современном быстроразвивающемся мире компьютерных систем и технологий является актуальной, так как многие даже самые популярные текстовые редакторы не способны предоставить весь спектр необходимых функций для работы с текстовыми документами, таких как: работа с таблицами, изображениями, возможность редактировать файлы нераспространенных кодировок, предоставление шаблонов текста и другие.

Разработкой текстовых редакторов занимаются многие именитые создатели программного обеспечения.

Однако в данной области еще имеется ряд проблем, решение которых поможет как увеличить функциональность систем, так и повысить скорость взаимодействия с пользователем.

Цель предлагаемой статьи – выполнить обзор самых распространенных систем текстовых редакторов, а также указать их проблематику и пути решения проблем текстовых редакторов.

Для достижения поставленной цели, необходимо решить следующие задачи:

- рассмотреть назначение, типичные функции и классификацию текстовых редакторов;
- выполнить сравнительную характеристику самых популярных и широких в плане функционала систем для выявления их достоинств и недостатков;
- определить необходимый функционал современных текстовых редакторов и проблематику их дальнейшего развития.

### **Назначение текстового редактора**

Пользователь – человек, использующий систему для решения стоящих перед ним задач. Для текстового редактора пользователь – это человек, у которого есть необходимость работы с данными из текстовых документов. Несмотря на первичную простоту работы системы текстового редактора, существуют важное различие между обычным и форматированным текстом [1,2,3].

Обычный текст представляет из себя набор символов в определенной последовательности, каждый из которых представлен фиксированной байтовой последовательностью в соответствии с одной из кодировок. Каждый символ в этой последовательности не связан ни с каким другим и представляет из себя самостоятельную единицу, которую можно удалить без каких-либо последствий для файла.

Форматированный же текст, в свою очередь, также представляет из себя последовательность символов, однако каждый такой текст может содержать метаданные, информацию о форматировании символов, абзацев, отступов между символами, словами и абзацами, а также может содержать информацию о форматировании страницы (отступы по краям, формат страницы, наличие нумерации, максимальное количество строчек).

Текстовый редактор предназначен для работы с текстовыми документами, которые могут содержать как обычный текст, так и информацию о форматировании текста, разметки страниц, метаданные и прочую информацию, которая записана в файле и предоставлять пользователю возможность редактирования этих данных.

### **Типичные функции текстового редактора**

В каждом текстовом редакторе предусмотрен ряд типичных функций для упрощения работы с текстом. Все эти функции можно разделить на работу с текстом и работу с файлом.

Среди функций для работы с текстом можно выделить следующие [1,2,3]:

*Набор и удаление текста* – стандартная функция, лежащая в основе любых текстовых редакторов от простейших до самых сложных.

*Поиск и замена* – в большинстве текстовых редакторов присутствует данная функция, предоставляющая пользователю возможность искать части текста, для ускорения работы с большим объемом данных. Также пользователю часто предоставляют возможность замены всех совпадающих частей на ту, что необходима пользователю, для исправления ошибок текста без необходимости замены всех совпадений вручную.

*Копировать, вставить, вырезать* – функция, предназначенная для быстрого дублирования текста из разных частей текста, или из разных источников.

*Отменить действие и повторить* – функция, необходимая для исправления ошибок набранного текста. В простейших текстовых редакторах отменяется лишь последняя внесенная правка, но в текстовых редакторах высокого уровня, формируется список изменений, который пользователь при необходимости может изменять.

*Форматирование текста* – большинство текстовых редакторов предоставляют ряд простейших функций визуального выделения текста, таких как изменение шрифта, размера букв, построчный перенос слов, отступы в начале строки, выделение комментариев к тексту и так далее. В простейших текстовых редакторах все это просто набор правил, которые сам редактор и применяет к тексту, не изменяя само форматирование текста в файле.

### **Классификация текстовых редакторов**

Все текстовые редакторы можно разделить по функциональным возможностям и их реализации. Можно выделить несколько функциональных уровней работы с текстом, по которым уже можно классифицировать сами текстовые редакторы [1,2,3]:

*Простейший уровень* – представляет из себя ряд простейших функций работы с текстом. Отличается отсутствием какого-либо форматирования текста, предоставляя пользователю лишь возможность набора, удаления и дублирования текста. Подобные текстовые редакторы чаще всего уже предустановлены на большинстве операционных систем.

*Средний уровень* – характерен наличием функциональными возможностями первичного форматирования текста. Редакторы среднего уровня имеют возможность применять к тексту шаблоны форматирования, изменения шрифтов, размера текста, отступы и прочее, однако данные изменения применяются на уровне программы, и не изменяют форматирование в самом файле.

*Высокий уровень* – выделяется на фоне остальных наличием большого функционала для редактирования текста, форматирования символов, слов, отдельных частей текста, изменения разметки страницы, размеров и нумерации страниц, изменение цвета, выделения, курсива, подчеркивания текста, создание и редактирование таблиц и изображений и много другое. Такие текстовые редакторы используют в основном для оформления различных документов по ГОСТ, для устранения неоднородности текста в документе.

### **Анализ состояния развития, необходимого функционала и проблематики современных текстовых редакторов**

Основной проблемой всех текстовых редакторов является невозможность работы с разными форматами файлов. Все текстовые редакторы по функциональным возможностям остаются на среднем уровне работы с текстом, и не поддерживают работу с документами, имеющими нестандартное форматирование текста.

Для качественного взаимодействия пользователя с файлом текстовый редактор должен обладать рядом функциональных возможностей:

- система должна обеспечить удобную и точную конвертацию, позволять чтение и редактирование,

популярных текстовых форматов, таких как PDF, DOC, TXT, HTML, XML, MD, RTF, DOCX;

- возможность чтения и исправления поврежденных файлов; под поврежденными подразумеваются файлы с нарушенной последовательностью записи форматирования или кодировки файла;

- должна иметься возможность сохранения и создания текстового файла в любом популярном формате, и в формате выбранным пользователем;

- обеспечение точного преобразования веб-страниц в текстовый документ со всем их содержимым, включая изображения, таблицы, схемы;

- должны иметься хорошие алгоритмы шифрования для обрабатываемых данных и выходного файла, полностью исключающие любые попытки несанкционированного доступа;

- система должна преобразовывать текстовые файлы определенного формата в абстрактную структуру, которая позволяет написать основной функционал программного продукта без ориентирования под определенный формат текстового файла и дает возможность расширения поддерживаемых форматов текстовых файлов без изменения основного функционала;

- поддержка многоядерных и многопроцессорных систем;

- должен иметься модуль распознавания текста с картинок с дальнейшей возможностью редактирование считанного текста;

- должен иметь качественную функцию проверки ошибок правописания содержимого файла, написанного на всех популярных языках, а также иметь шаблоны исправления стандартных ошибок в режиме реального времени;

- возможность проверки соответствия и преобразования содержимого текстовых файлов к популярным стандартам и ГОСТам;

- обширный функционал работы с таблицами. Пользователь должен иметь возможность создания таблицы как по шаблонам, так и по пользовательскому форматированию, доступ к редактированию данных таблицы в интерактивном варианте;

- возможность установления авторских прав к содержимому, таких как водяных и других знаков, позволять скрывать конфиденциальную информацию в содержимом текстового файла;

- возможность работы над несколькими текстовыми файлами одновременно;

- должна иметь интерфейс с поддержкой распространенных языков мира, таких как английский, русский, немецкий, испанский и другие.

Одним из главных минусов большинства текстовых редакторов можно назвать отсутствие кроссплатформенности. Эта проблема довольно сильно распространена, так как адаптировать приложение для разных операционных систем, с разной архитектурой проектирования системы довольно трудоемкая задача.

Для сравнения лучших и самых распространенных текстовых редакторов можно привести Microsoft Word (см. рис. 1) и Google Docs (см. рис. 2). Внешний вид обоих приложений имеет общие черты: в верхней части страницы расположена шапка с разделами и функциями для работы с текстом, таблицами, изображениями и настройки формата страницы, большая часть интерфейса занята рабочей частью редактора, в которой и производится работа над текстом, таблицами и изображениями. Среди плюсов приложения Microsoft Docs можно выделить интерфейс, выполненный по принципу минимализма, качественную группировку вкладок, которая помогает пользователю быстрее находить нужную функцию, огромный функционал форматирования текста, работы с таблицами и изображениями, проверка правописания текста в режиме реального времени, и поддержку большого количества популярных расширений файлов.

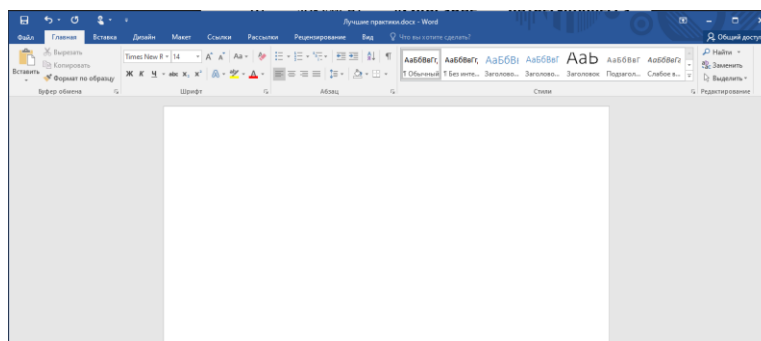


Рисунок 1 – Главная страница приложения Microsoft Word

Минусами приложения являются отсутствие возможности простого редактирования панели быстрого доступа, пользователь не может настроить внешний вид приложения в соответствии своим предпочтениям и требованиям. Также одной из главных проблем является то, что Microsoft Office работает не на всех платформах. Основной платформой работы приложения является операционная система Windows, а также имеется еще одна версия приложения для работы на Android, для запуска приложения на семействе Linux необходимо устанавливать дополнительный пакет Snap, который поддерживается лишь 75% операционных систем Linux, а



платформы IOS и Mac OS и вовсе не имеют поддержку Microsoft Word.

Приложение Google Docs также имеет недостаток в виде невозможности настраивать внешний вид приложения, и панели быстрого доступа. Однако Google Docs имеет поддержку почти на всех платформах, на которых имеется поддержка браузеров для выхода в интернет, так как Google Docs интернет средой для редактирования файлов. Благодаря отсутствию необходимости устанавливать приложение на устройство, Google Docs является одним из лучших текстовых редакторов, а возможность сохранения документов сразу на Google Disk, исключает возможность потери данных при чтении или редактировании.

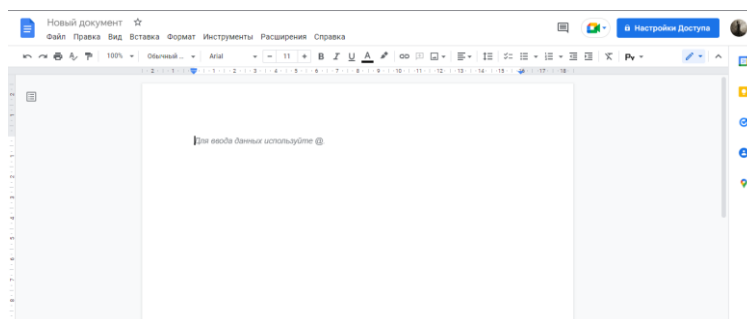


Рисунок 2 – Главная страница приложения Google Docs

Однако сама необходимость наличия у пользователя браузера и постоянного подключения к интернету, делает данное систему не лучшим выбором для редактирования данных, но в век продвинутых технологий, когда почти у любого средне продвинутого пользователя есть домашний или мобильный интернет, данная проблема не является критической для работы.

### **Вывод**

В результате проделанной работы можно подытожить, что многие текстовые редакторы обладают небольшим инструментарием для редактирования текста, не имеют поддержку файлов с нестандартным форматированием текста, что в свою очередь ограничивает возможности пользователя по редактированию и форматированию данных текстовых документов. Это делает необходимым преодоление данного недостатка.

Но при этом, несмотря на все недостатки современных текстовых редакторов, они все равно будут пользоваться спросом, так как наличие большого функционала полезно лишь пользователям узконаправленных специальностей, а для среднестатистического пользователя хватает функционала уже разработанных текстовых редакторов.

### **Литература**

1. Л. Гопалакришнан; Г. Падманабхан; Судхат Шукла. Ваш домашний компьютер: максимально используйте свой персональный компьютер. Образование Таты Макгроу-Хилл. 2003. – 190 с.
2. Петров Ю.И., Шупикова Ю.В., Викулина Д.А., Макаров С.Н. Способы и средства автоматизации текстового процессора Microsoft Word. – г. Москва, - 2012. – С. 98-107.
3. The Internet of Things: Text editor [Электронный ресурс] – Электрон. дан. - 2022. – Режим доступа: [https://translated.turbopages.org/proxy\\_u/en-ru.ru.e8b06a4d-636a149f-4b87a8fd-74722d776562/https/en.wikipedia.org/wiki/Text\\_editor](https://translated.turbopages.org/proxy_u/en-ru.ru.e8b06a4d-636a149f-4b87a8fd-74722d776562/https/en.wikipedia.org/wiki/Text_editor).

**Е.А. Лазуренко, Я.А. Рудь, А.П. Ищенко, А.В. Григорьев. Текстовый редактор с возможностью обработки документов с неформатированным и форматированным текстом. В статье представлен краткий обзор текстовых редакторов, описание функций, возможностей, классификаций, сравнения популярных систем текстовых редакторов.**

**Ключевые слова:** файл, текстовый редактор, пользователь, форматирование, операционная система, шрифт, функции, классификация.

**Lazurenko Eugene, Rud Yaroslav, Ishchenko Anton, Grigoriev Alexander. Text editor with the ability to process documents with unformatted and formatted text. The article provides a brief overview of text editors, a description of functions, capabilities, classifications, and comparisons of popular text editor systems.**

**Key words:** file, text editor, user, formatting, operating system, font, functions, classification.

## Исследование функциональных возможностей CATIA V5 в проектировании листовых аэрокосмических деталей

Семериков О.Р.<sup>\*1</sup>, Григорьев А.В.<sup>\*2</sup>

<sup>\*1</sup> магистрант, Донецкий национальный технический университет  
semerikov2917@gmail.com

<sup>\*2</sup> к.т.н., доцент, Донецкий национальный технический университет,  
grigorievalvl@mail.ru

*Семериков О.Р., Григорьев А.В. Исследование функциональных возможностей CATIA V5 в проектировании листовых аэрокосмических деталей В статье рассматривается модуль CATIA V5 – Aerospace Sheet Metal Design, освещается инструментарий модуля и применение его на практике. Описываются достоинства модуля, позволяющие организовать корректное производство листовых деталей, использующихся в аэрокосмической отрасли.*

**Ключевые слова:** аэрокосмическая отрасль, листовые детали, CATIA V5, модуль Aerospace Sheet Metal Design.

### Введение

Актуальность исследования обусловлена востребованностью аэрокосмической отрасли в современных мировых реалиях. Аэрокосмическая отрасль имеет множество применений в исследовательских, коммерческих и военных целях. Специфика отрасли побуждает наиболее развитые страны к сотрудничеству, ярким примером может служить Международная космическая станция, в проекте которой участвуют 14 ведущих стран. С применением аэрокосмической отрасли в военных целях ситуация обстоит иначе, сотрудничество стран осуществляется только в пределах соответствующих военно-политических блоков.

Как правило, разработка продуктов аэрокосмической отрасли автоматизирована и ведётся в САПР. Одной из наиболее популярных САПР сейчас является CATIA, разработанная французской компанией Dassault Systemes. CATIA предоставляет обширные функциональные возможности для проектирования и разработки множества видов деталей, в том числе деталей из листовых металлов, которые используются в аэрокосмической отрасли.

Следовательно, целью исследования является рассмотрение и применение на практике инструментария модуля Aerospace Sheet Metal Design (далее ASL).

Список задач для выполнения поставленной цели:

- осуществить обзор инструментария модуля ASL;
- спроектировать с помощью модуля ASL нервюру крыла самолёта.

Новизна исследования заключается в выявлении достоинств инструментария модуля ASL, позволяющих организовать корректное производство листовых деталей, использующихся в аэрокосмической отрасли.

### Обзор инструментария модуля ASL



Модуль ASL предназначен для проектирования и разработки деталей летательных аппаратов (ЛА) из листовых металлов. Модуль основан на технике функционального моделирования, которая использует готовый набор конструктивно-технологических элементов, свойственных деталям ЛА из листовых металлов – благодаря чему разработка и изменение подобных деталей отнимает меньше времени, чем при использовании стандартного инструментария.

Модуль ASL предоставляет большой выбор инструментов для проектирования листовых деталей. Помимо этого, параметры каждого инструмента можно гибко настраивать (например, выбирать направление выреза, его глубину и т.д.). С инструментарием модуля можно ознакомиться на рис. 1.



Рисунок 1 – Инструментарий модуля ASL


Как видно из рис. 1, некоторые инструменты содержат в себе набор вариаций, которые будут рассмотрены далее.


Посредством инструмента «Sheet Metal Parameters» () определяются общие параметры листовой детали – её толщина, методы изгиба, К-фактор и т.д. В свою очередь, инструмент «Web» () позволяет создать листовую деталь на основе определённого эскиза. Следует отметить, что листовая деталь всегда ограничена единицей (невозможно создать более одной детали) [1].



Взаимодействие с остальными инструментами осуществляется в пределах созданной листовой детали.



Например, с помощью инструмента «Surfacic Flange» () можно образовать поверхностный фланец (выступ) на детали. Для этого необходимо выбрать саму деталь и поверхность, на которой будет образован фланец.


Инструментами «Joggle» () и «Twin Joggle» () создаются изгибы на листовой детали либо на поверхностном фланце.

Инструмент «Cut Out» () используется, когда необходимо создать вырез на детали. Вырезаемая область, как правило, определяется соответствующим эскизом. Если же требуется образовать отверстие на детали





– следует использовать инструмент «Hole» ()», позволяющий определять диаметр отверстия, глубину и т.д.








«Corner» () позволяет образовать угол между двумя гранями детали, а инструмент «Chamfer» () необходим для создания фаски.




С помощью инструмента «Fold/Unfold» () изменяется отображение листовой детали (в сложенном либо разложенном виде). Помимо этого, применение инструмента «Multi Viewer» () приводит к отображению детали сразу в двух видах (в разных окнах).

Инструмент «Point or Curve Mapping» () используется для создания пользовательских точек либо кривых на детали с последующим их отображением в сложенном, либо разложенном видах (инструмент можно использовать, например, для создания логотипа).

Помимо вышеперечисленного, модуль содержит 3 набора инструментов:

1) Набор «Swept Walls» используется для создания стенок видов: «Flange» ()», «Hem» ()», «Tear Drop» ()», также присутствует возможность создания пользовательской стенки «User Flange» ()». Для создания требуется выбрать спайн (кромку, от которой будет осуществляться создание стенки), определить длину стенки и другие её параметры в соответствии с видом (угол, радиус и т.д.).

2) «Stampings» – набор, позволяющий образовывать на детали штампы следующих видов: «Circular Stamp» ()», «Surface Stamp» ()», «Flanged Hole» ()», «Bead» ()», «Flanged Cut Out» ()», «Stiffening Rib» ()», «Curve Stamp» ()», «User Stamp» ()».

3) Последний набор предоставляет возможности создания паттернов (узоров) следующих видов: «Rectangular Pattern» ()», «Circular Pattern» ()» и, как и в других наборах, присутствует возможность создания пользовательского варианта «User Pattern» ()».

### **Проектирование нервюры крыла с помощью модуля ASL**

Воспользуемся инструментарием модуля ASL для проектирования нервюры крыла. Сначала, на основе эскиза нервюры, необходимо создать листовую деталь (см. рис. 2).

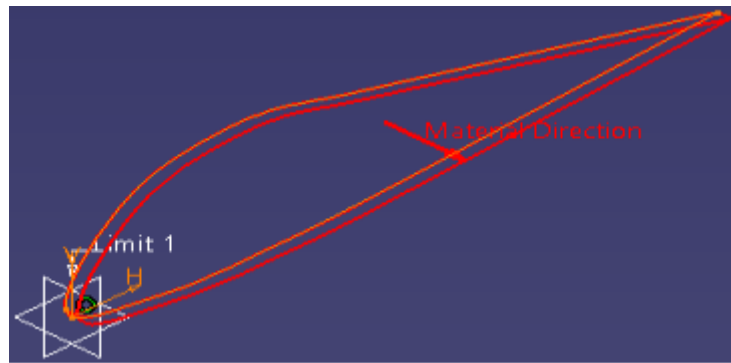


Рисунок 2 – Создание листовой детали на основе эскиза

Далее, с помощью инструмента «Cut Out», образуем вырезы для лонжеронов и стрингеров. Помимо вырезов, требуется создать отверстия облегчения нервюры. В этом случае резонным будет использование инструмента «Hole» (см. рис. 3).

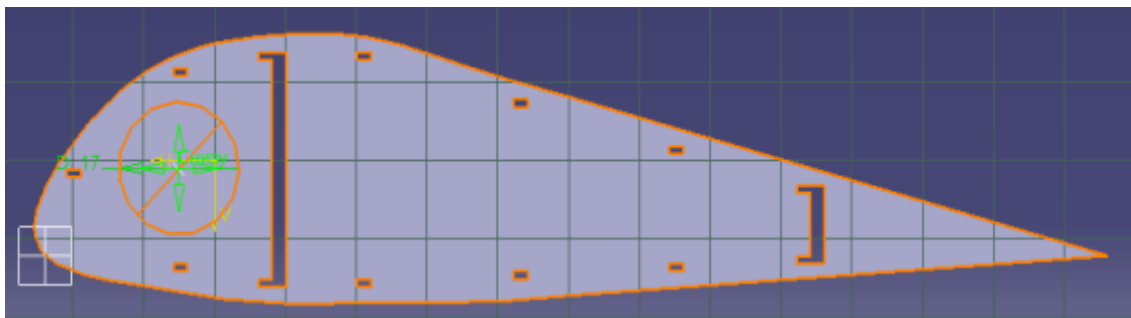


Рис. 3 – Создание отверстий облегчения

Теперь, переходим к отбортовке отверстий облегчения посредством инструмента «Chamfer». Определяем длину и угол – после чего применяем фаску (см. рис. 4).

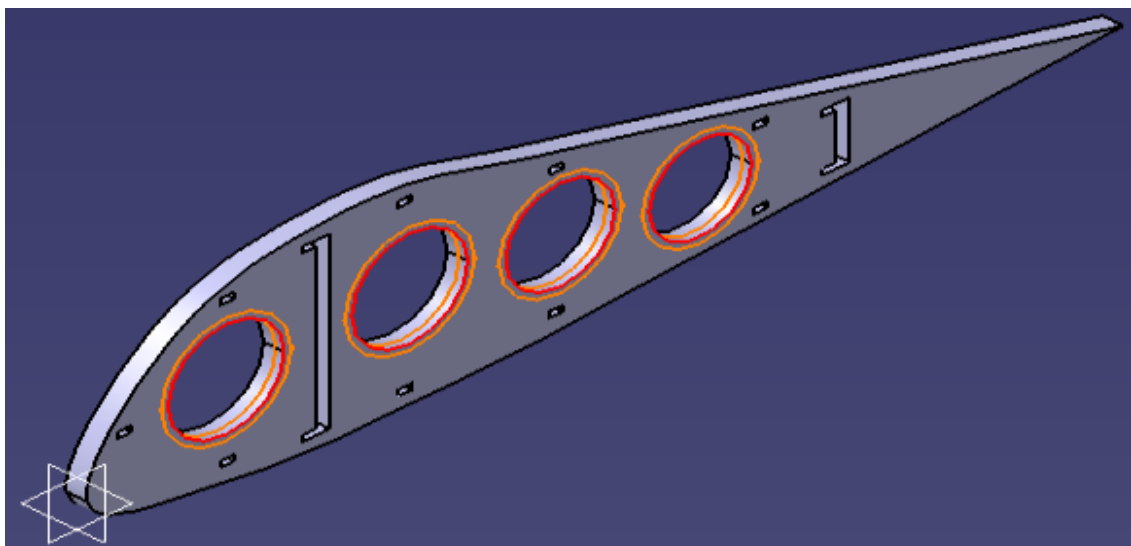


Рисунок 4 – Применение инструмента «Chamfer» к отверстиям облегчения

В конце создадим выступ с одной стороны нервюры. Спроектированная нервюра с обозначениями представлена на рис. 5.

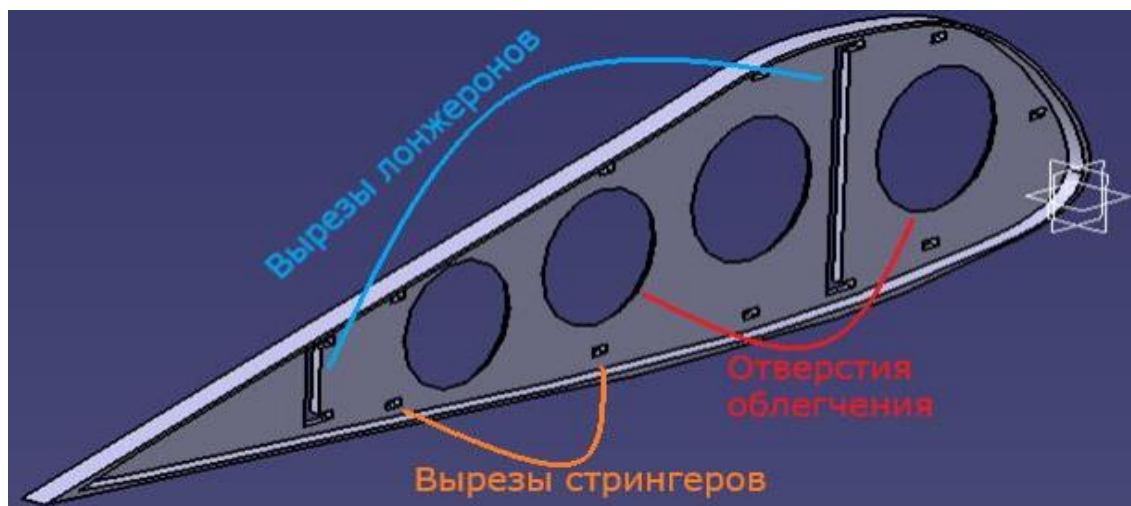


Рисунок 5 – Спроектированная нервюра крыла

## Выводы

В статье был рассмотрен инструментарий модуля ASL и, с его помощью, спроектирована нервюра крыла. Модуль содержит множество инструментов и позволяет гибко настраивать параметры при их применении (длину, глубину, диаметр и т.д.). Помимо этого, присутствуют возможности создания пользовательских видов стенок, паттернов и штампов – всё это позволяет назвать инструментарий модуля достаточным для проектирования листовых аэрокосмических деталей.

Стоит отметить, что модуль обладает ограничениями, которые, по мнению автора, являются необходимыми. Так, например, можно создавать только одну листовую деталь – это, по мнению автора, логичное ограничение, т.к. не имеет смысла создавать одновременно более 1 листовой детали в пределах одного проекта (что организовано, например в САПР NX). При производстве может возникнуть проблема, когда станок будет одновременно создавать несколько деталей, например, с разной толщиной. Подкрепляет данную гипотезу инструмент «Sheet Metal Parameters», который определяет именно общие параметры для листовой детали, а не отдельно для каждой её составляющей (стенки и т.д.). Корректное производство строится вокруг отдельного создания каждой детали и их дальнейшей сборки.

Следовательно, можно утверждать, что цели исследования были достигнуты – рассмотрен инструментарий модуля ASL, выявлены его достоинства, позволяющие организовывать корректное производство, помимо этого, с помощью инструментария модуля спроектирована нервюра крыла. Статья может в дальнейшем использоваться как руководство для начального знакомства с модулем ASL в CATIA V5.

## Литература

1. Aerospace Sheet Metal Design [Электронный ресурс] / Catiadesign.org Режим доступа: [https://catiadesign.org/\\_doc/docv5r14/Aerospace%20Sheetmetal%20Design.pdf](https://catiadesign.org/_doc/docv5r14/Aerospace%20Sheetmetal%20Design.pdf) – Загл. с экрана.

**Семериков О.Р., Григорьев А.В. Исследование функциональных возможностей CATIA V5 в проектировании листовых аэрокосмических деталей** В статье рассматривается модуль CATIA V5 – Aerospace Sheet Metal Design, освещается инструментарий модуля и применение его на практике. Описываются достоинства модуля, позволяющие организовать корректное производство листовых деталей, использующихся в аэрокосмической отрасли.

**Ключевые слова:** аэрокосмическая отрасль, листовые детали, CATIA V5, модуль Aerospace Sheet Metal Design.

**Semerikov O.R., Grigoriev A.V. Investigation of the functionality of CATIA V5 in the design of sheet aerospace parts.** The article discusses the module CATIA V5 – Aerospace Sheet Metal Design, highlights the tools of the module and its application in practice. The advantages of the module are described, which allow organizing the correct production of sheet parts used in the aerospace industry.

**Key words:** aerospace industry, sheet metal parts, CATIA V5, Aerospace Sheet Metal Design module.

## Особенности проектирование веб-приложения на основе микросервисной архитектуры

И.Р. Настеко \*1, А.В. Григорьев \*2

\*1 бакалавриат, Донецкий национальный технический университет,  
nastekoir@gmail.com

\*2 к.т.н., доцент, Донецкий национальный технический университет,  
grigorievalvl@gmail.com

**Настеко И.Р., Григорьев А.В. Особенности проектирования веб-приложения на основе микросервисной архитектуры.** В данной статье рассмотрены основные свойства и принципы микросервисов, описаны способы взаимодействия микросервисов между собой, проведён сравнительный анализ микросервисной и других типов архитектур. Были затронуты синхронные и асинхронные методы взаимодействия микросервисов. Сделаны выводы.

**Ключевые слова:** микросервис, архитектура, монолит, сервис, протокол, шина, запрос, ответ.

### Введение

Микросервисная архитектура [1-3] - принципиальная организация распределенной системы на основе микросервисов и их взаимодействия друг с другом и со средой по сети, а также принципов, направляющих проектирование архитектуры, её создание и эволюцию. Микросервисы — это небольшие, автономные, совместно работающие сервисы. Разберем это определение по частям и рассмотрим, что определяет отличительные черты микросервисов.

Микросервисы обеспечивают превосходные возможности сопровождения в крупных комплексных системах с высокой масштабируемостью за счет создания приложений, основанных на множестве независимо развертываемых сервисов с автономными жизненными циклами. Дополнительное преимущество в том, что микросервисы можно масштабировать независимо. Вместо монолитного приложения, которое нужно масштабировать как единое целое, вы масштабируете отдельные микросервисы. Тем самым можно масштабировать только функциональную область, требующую больше вычислительных или сетевых ресурсов, не затрагивая другие области приложения, которые на самом деле не нуждаются в масштабировании. Таким образом можно сократить расходы, так как требуется меньше оборудования.

Микросервисы дают нам существенно больше свободы воздействия и принятия различных решений инженеру, позволяя быстрее реагировать на неизбежные изменения, касающиеся всех нас.

Цель предлагаемой работы заключается в том, чтобы объяснить широкой аудитории особенности проектирования микросервисной архитектуры веб-приложения, её преимущества и недостатки.

Предполагается решение следующих задач:

- проанализировать основные свойства и принципы микросервиса,
- описать способы взаимодействия микросервисов между собой;
- сравнительный анализ микросервисной и других типов архитектур.

### Анализ основных свойств и принципов микросервиса

При проектировании микросервисов необходимо отталкиваться от их основных характеристик. Каждый микросервис должен быть небольшим, сфокусированным, и независимым настолько, насколько это возможно. В фокусе одного микросервиса должна быть одна задача. Это не только позволит сделать его максимально компактным, но и обеспечит самостоятельность микросервиса, исключив лишние связи. Если спроектирована система, где имеются микросервисы с плотным взаимодействием и один не может существовать без другого, это повод задуматься над тем, чтобы объединить эти приложения в одно.

Границы сервисов формируются на основе бизнес-границ, что позволяет со всей очевидностью определить местонахождение кода для заданной области выполняемых функций. Удерживая сервис в четко обозначенных границах, мы не позволяем себе мириться с его чрезмерным разрастанием со всеми вытекающими из этого трудностями.

Микросервис должен быть автономным и самостоятельным образованием, которое может быть развернуто в качестве обособленного сервиса на платформе, предоставляемой в качестве услуги. Сервисам необходимо иметь возможность изменяться независимо друг от друга и развертываться, не требуя никаких изменений от потребителей. Если объем совместно используемого будет слишком велик, потребляемые сервисы станут завязываться на внутренние представления. Это снизит автономность, поскольку при внесении изменений

потребуется дополнительного согласования с потребителями.

Микросервисная архитектура — воплощение паттернов High Cohesion: высокая сплоченность и слабая связность. Микросервис обязан быть независимым компонентом, то есть такой единицей, код которой может быть независимо заменен или обновлен. Слабая связность — это когда можно легко внести изменения в один сервис, не трогая остальные. Для достижения подобного подхода нужно гарантировать, что каждый сервис знает о других только необходимый минимум.

Определение границ микросервиса — самый важный шаг. От этого будет зависеть вся дальнейшая его жизнь и серьезно повлияет на жизнь команды, отвечающей за него. Один из наиболее известных способов разбиения на микросервисы — это определение бизнес-возможностей приложения и создание по одному микросервису на каждую из них. Бизнес-возможности представляют собой функции, которые будут доступны пользователям при работе с приложением. Одной из важнейших задач — суметь корректно выделить минимальную самостоятельную бизнес-функцию, которая в дальнейшем будет трансформирована в отдельный микросервис и обосновать решение

Чтобы минимизировать ошибки при определении границ, нужно вначале их продумать. Поэтому, как правило, оправданным является подход Monolith First, когда вначале систему развивают в традиционной парадигме, а когда появляются устойчивые области, их выделяют в микросервисы. Главное, чтобы выигрыш от разбиения превышал сложности пересмотра этих границ.

Принципы складываются из утверждений о том, как все должно делаться, и из разъяснения причин, по которым все должно делаться именно таким образом. Они помогают нам выстроить различные решения, которые приходится принимать при создании наших систем (см. рис. 1) [1-3].



Рисунок 1 – Достоинства микросервисов

Модель, построенная вокруг бизнес-концепций, интерфейсы, структурированные вокруг контекстов, связанных с бизнес-задачами, отличаются большей стабильностью, чем те, которые структурированы вокруг технических концепций. Моделированием областей, в которых работает система, мы обеспечиваем повышение возможности более свободно отображать изменения в бизнес-процессах. Для определения потенциальных границ областей следует использовать ограниченный контекст.

Внедрение культуры автоматизации. Микросервисы усложняют системы главным образом из-за необходимости работать с огромным количеством активных компонентов. Применение культуры автоматизации является одним из основных способов справиться с этой проблемой, и в этом плане имеет смысл как можно раньше приложить максимум усилий к созданию инструментария для поддержки микросервисов. Поскольку удостовериться в работоспособности сервисов по сравнению с монолитными системами намного сложнее, важную роль в этом процессе играет автоматизированное тестирование. Также может помочь использование одинакового вызова командной строки для повсеместного единообразного развертывания, который может стать основной частью внедрения непрерывной поставки с целью обеспечения быстрой оценки качества работы в производственном режиме при каждой проверке.

Скрытие подробности внутренней реализации. Чтобы максимально увеличить возможность одного сервиса совершенствоваться независимо от всех других сервисов, крайне важно скрыть подробности его



реализации. Содействие в этом могут оказать ограниченные контексты, поскольку они помогают сконцентрировать усилия на моделях, предназначенных для совместного использования и при этом обладающих скрытностью. Сервисы также должны скрывать свои базы данных во избежание попадания в одну из наиболее распространенных разновидностей связывания, которая может проявляться в традиционных сервис-ориентированных архитектурах и использовать для объединения данных от нескольких сервисов с целью создания отчетов перекачку данных или перекачку данных о событиях.

**Всесторонняя децентрализация.** Для достижения максимальной автономности, допускаемой микросервисами, нужно постоянно выискивать возможности для передачи полномочий по принятию решений и управлению тем командам, которые владеют микросервисом. Этот процесс начинается с внедрения самообслуживания везде, где только можно, позволяя людям разворачивать программные средства по мере надобности, делая как можно проще разработку и тестирование и исключая необходимость для отдельных команд выполнять эти действия.

**Независимое развертывание.** Нужно всегда стремиться к тому, чтобы микросервисы могли разворачиваться и развертывались самостоятельно. Даже когда требуются изменения, нарушающие общий режим работы, нужно добиваться совместного существования конечных точек разных версий, позволяя потребителям со временем внести изменения.

**Изолирование сбоев.** Архитектура микросервисов может быть более устойчивой, чем монолитная система, но только если мы разбираемся в ситуации и имеем план на случай сбоев в части нашей системы. Если возможность и неизбежность сбоев вызова нижестоящего сервиса в расчет не принимаются, наша система может испытать катастрофический каскадный сбой и мы останемся с системой, еще более хрупкой, чем прежняя.

**Взаимодействие микросервисов** может быть реализовано с помощью разных типов связей. Их можно разделить на две группы.

В первой группе устанавливается асинхронный или синхронный тип потока. При использовании асинхронного протокола код клиента или отправителя сообщения не ждет ответ — сообщения отправляются аналогично передаче в очередь любого брокера. Такой тип применяет протокол AMQP. Использование синхронного протокола означает, что при отправке запроса клиент ожидает ответ от службы — задачи выполняются только после того, как сервер пришлет ответ. Пример такого протокола — HTTP.

Во второй группе устанавливается сколько получателей будет у запроса. При одном получателе каждый запрос обрабатывает конкретный получатель или сервис. При нескольких получателях для каждого запроса количество получателей отличается. Взаимодействие выстраивается с использованием асинхронных протоколов.

В микросервисных приложениях часто используют связку двух стилей. Например, для взаимодействия с одним получателем — синхронный протокол, а для асинхронного — протоколы сообщений. Но при создании микросервисов большее значение имеет не тип связи, а возможность асинхронной интеграции микросервисов без снижения их независимости.

При создании приложений с микросервисной архитектурой важно правильно реализовать способ интеграции — в идеале, чтобы внутренние микросервисы взаимодействовали между собой минимально, а взаимодействие было асинхронным. Следует понимать, что использование синхронного взаимодействия между несколькими микросервисами не лучший вариант — каждый микросервис должен быть автономным и доступным клиенту, даже если другие сервисы отключены или недоступны. Если это не реализовать, архитектура будет неустойчива к сбоям — при недоступности одного сервиса будет недоступно все приложение. Кроме того, наличие зависимостей HTTP между микросервисами влияет не только на их автономность, но и на быстродействие — чем меньше синхронных зависимостей, тем выше скорость отклика в клиентском приложении (см. рис. 2) [1-3].

Преимущество асинхронного типа связи в том, что запрос клиента обрабатывается сразу, в то время, как при синхронной модели вся обработка выполняется одновременно. Поэтому синхронный способ взаимодействия лучше не использовать в рамках исходной операции «запрос-ответ». Асинхронный тип связи можно использовать, даже если исходному микросервису нужны данные из другого микросервиса. В такой ситуации отказаться от синхронных запросов можно, делая реплики данных в базу данных — это допустимо. Исходя из этого, главное что нужно понимать — не используйте синхронные зависимости между микросервисами.

В приложениях на базе микросервисов можно использовать разные стили внутреннего взаимодействия модулей. Если взаимодействие выполняется за пределами кластера микросервиса, то лучше подойдет механизм синхронных запросов и ответов. Если взаимодействие выполняется в пределах кластера, подойдет комбинированный формат взаимодействия или асинхронное взаимодействие на базе сообщений.

Кроме того, можно использовать и нестандартный комбинированный метод, но такой вариант подойдет только для внутреннего взаимодействия на кластере — сервисы с таким форматом лучше не публиковать.



При использовании взаимодействия типа «запрос-ответ», клиентский запрос поступает к службе, которая обрабатывает его и возвращает ответ. Такой тип взаимодействия — лучший вариант для запроса данных от клиентских приложений, поэтому он чаще применяется в архитектуре микросервисов. Один из распространенных стилей архитектуры при подобном типе взаимодействия — REST. Подход основан на HTTP-протоколе и позволяет работать с HTTP-командами (см. рис.3) [1-3].

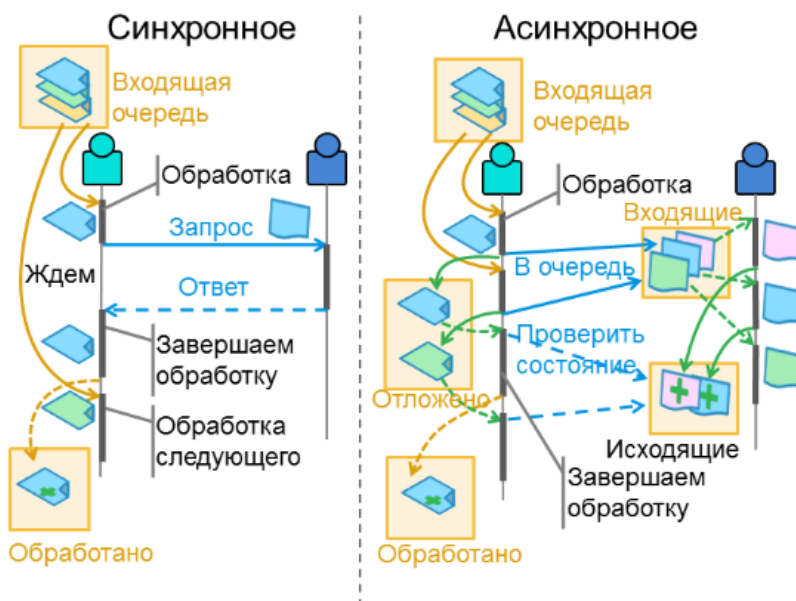


Рисунок 2 – Алгоритм синхронного и асинхронного взаимодействия

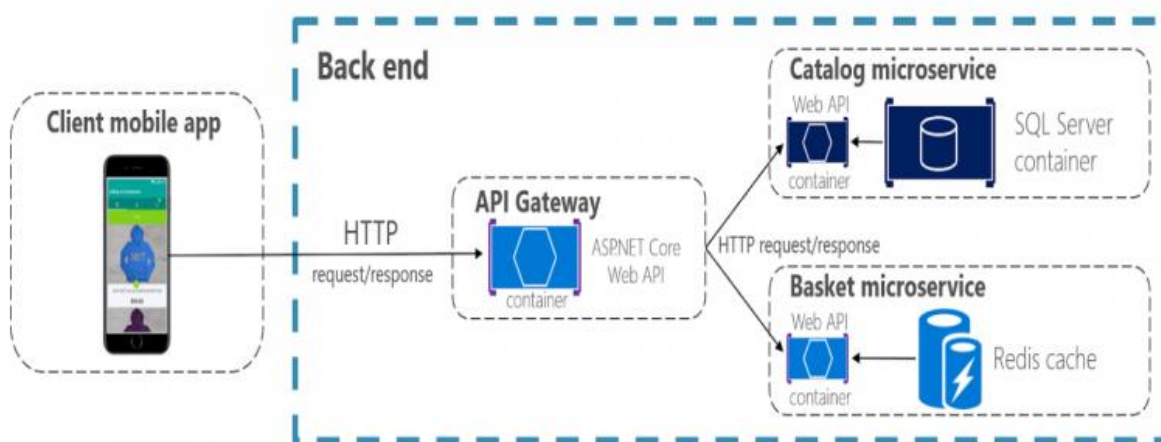


Рисунок 3 - Использование взаимодействия типа «запрос-ответ»

Еще один метод — асинхронное взаимодействие в формате «один ко многим» в режиме реального времени с платформами высшего уровня. При таком варианте взаимодействия серверный код способен автономно отправлять доступные данные клиентам, а не дожидаться, пока он их запросит (см. рис. 4) [1-3].

Один из популярных инструментов для выстраивания подобного взаимодействия для быстрой передачи данных с сервера клиентам — SignalR. При этом обработку выполняет протокол WebSockets с множеством подключенных WebSockets на стороне клиентов (один клиент — один WebSockets). Например, такой метод применяют в спортивных приложениях, когда нужно всем пользователям передать данные об изменении счета в матче.

Проведём **сравнительный анализ микросервисной и других типов архитектур**

Отметим недостатки монолитной архитектуры:

- даже небольшое изменение любого функционала приложения приводит к сборке и развертыванию новой версии всего ПО.
- масштабировать можно только все приложение, отдельный блок масштабировать невозможно.

- если откажет любой модуль приложения, то в следствии этого, может быть нарушена работа всего приложения.
- средства разработки всегда ограничены выбранным сразу стеком технологий.
- сложность в управлении большой командой квалифицированных разработчиков. Каждый разработчик должен разбираться во всем функционале приложения, а не только в своем модуле.
- любое обновление затрагивает весь функционал приложения, это приводит к рискам отказа приложения после обновлений. Поэтому производятся только редкие выпуски обновлений.
- любые изменения в базе данных могут сказываться на работе всего приложения и нуждаются в изменениях в коде.

## Push and real-time communication based on HTTP

### One-to-many communication

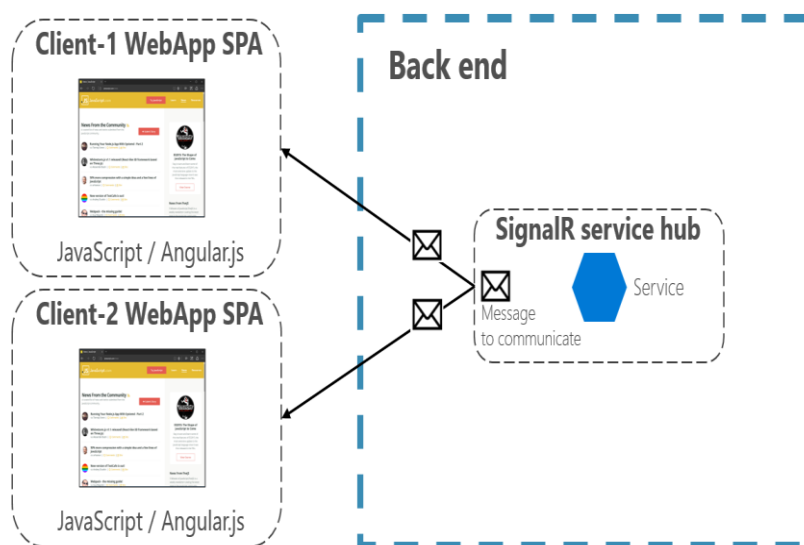


Рисунок 4 - Асинхронное взаимодействие в формате «один ко многим»

В случае, если это небольшая бесплатная программа для обучения каким-то отдельным навыкам для обычного пользователя, да еще и обновляемая достаточно редко, то монолитная архитектура вполне подойдет для такой разработки.

Если же речь идет о корпоративном приложении, да еще и часто обновляемом, то необходимо выбирать микросервисную архитектуру.

Необходимо отметить, что есть еще один тип архитектуры — это сервис-ориентированная архитектура. Иногда ее путают с микросервисной. Кажется, что отличия между микросервисной архитектурой и сервис-ориентированной не так очевидны. Но существуют различия между микросервисами и сервис-ориентированной архитектурой, это проявляется в отношении роли сервисной шины. Сервис-ориентированная архитектура представляет собой архитектуру в масштабах компании, ее цель в стандартизации взаимодействия и интеграции веб-сервисов компании. Назначение микросервисной архитектуры — в разработке конкретного приложения. Рассмотрим основные преимущества микросервисной архитектуры по сравнению с монолитной архитектурой:

- простота и независимость в развертывании приложения. В случае с микросервисами, можно развернуть только один какой-нибудь модуль приложения.
- точность и эффективность масштабирования. Для начала, необходимо определиться, какие микросервисы будут требовать частого масштабирования, а какие — нет. Модули, которые не нужно часто масштабировать можно расположить на более слабых серверах, а часто масштабируемые — масштабировать отдельно от всего остального ПО.
- повышенная отказоустойчивость приложения. Рациональное проектирование приложения и выстраивание независимых связей между модулями дает следующее преимущество: отказ одного из модулей не приводит к отказу всего программного обеспечения.
- выбор стека технологий. Разрабатывая каждый микросервис, можно подобрать наиболее соответствующий его функциям и удобству разработки стек технологий.
- гибкость в управлении командами разработчиков. Новому специалисту проще быстрее войти в работу,

так как не нужно долго изучать функционал всего приложения, достаточно освоить стек технологий для конкретного микросервиса.

- возможность использования функционала повторно (для различных целей и различными способами).
- замена или удаление ненужных заказчику сервисов решается быстро и легко.
- каждый микросервис использует собственную БД. Этот факт приводит к независимости моделей данных. К примеру, если программист изменил модель данных в одном конкретном сервисе, то это не повлияет на работу других сервисов.

Как мы видим, микросервисная архитектура имеет значительные преимущества и все больше и больше привлекает разработчиков. Однако, перед тем как выбрать архитектуру для разработки ПО, стоит посмотреть и на недостатки микросервисов, которые будут перечислены ниже:

Система работы микросервисов является распределенной. С одной стороны, это преимущество в работе ПО. С другой стороны, если микросервисов слишком много и каждый из них обращается с запросами к другим сервисам, то результирующее время отклика будет увеличиваться и появятся «точки отказа». Для решения этой проблемы существуют два пути:

- изменение детализации вызовов, которое может привести к снижению их количества;
- внедрение асинхронности, отзывы выполняются параллельно, в результате это приводит к тому, что конечное время отклика — это самое медленное время из всех, а не суммарное время всех задержек.

Постоянное усложнение процесса разработки, что приводит к повышению требований к квалификации программистов. В микросервисной архитектуре роль интеграционных процессов и процессов непрерывной доставки велика. И поэтому достаточно трудно обрабатывать множество процессов без автоматизации тестирования и развертывания сервисов. Это факторы требуют внедрения DevOps в компании и тесного сотрудничества разработчиков с системными инженерами, тестировщиками, службой тех. поддержки и т.д.

Децентрализация в микросервисной архитектуре порождает проблемы с согласованностью микросервисов. Например, в монолитном приложении за одну транзакцию возможно выполнение множества изменений, но и возможен откат назад, если произошел сбой, с сохранением согласованности данных. При применении микросервисов, возможна следующая ситуация: в случае нарушения работы одного из сервисов, перестает отвечать другой микросервис. В таком случае — это вопрос приоритетов разработчика: можно дать приоритет доступности компонентов (в случае выхода из строя одного сервиса, другие продолжают функционирование). В общем, разработчики должны находить баланс между согласованностью сервисов и их доступностью и делать это необходимо очень осторожно. Рассмотрим основные преимущества микросервисной архитектуры по сравнению с монолитной архитектурой:

- простота и независимость в развертывании приложения. В случае с микросервисами, можно развернуть только один какой-нибудь модуль приложения.

- точность и эффективность масштабирования. Для начала, необходимо определиться, какие микросервисы будут требовать частого масштабирования, а какие — нет. Модули, которые не нужно часто масштабировать можно расположить на более слабых серверах, а часто масштабируемые — масштабировать отдельно от всего остального ПО.

- повышенная отказоустойчивость приложения. Рациональное проектирование приложения и выстраивание независимых связей между модулями дает следующее преимущество: отказ одного из модулей не приводит к отказу всего программного обеспечения.

- выбор стека технологий. Разрабатывая каждый микросервис, можно подобрать наиболее соответствующий его функциям и удобству разработки стек технологий.

- гибкость в управлении командами разработчиков. Новому специалисту проще быстрее войти в работу, так как не нужно долго изучать функционал всего приложения, достаточно освоить стек технологий для конкретного микросервиса.

- возможность использования функционала повторно (для различных целей и различными способами).
- замена или удаление ненужных заказчику сервисов решается быстро и легко.
- каждый микросервис использует собственную БД. Этот факт приводит к независимости моделей данных. К примеру, если программист изменил модель данных в одном конкретном сервисе, то это не повлияет на работу других сервисов.

Как мы видим, микросервисная архитектура имеет значительные преимущества и все больше и больше привлекает разработчиков. Однако, перед тем как выбрать архитектуру для разработки ПО, стоит посмотреть и на недостатки микросервисов, которые будут перечислены ниже:

Система работы микросервисов является распределенной. С одной стороны, это преимущество в работе ПО. С другой стороны, если микросервисов слишком много и каждый из них обращается с запросами к другим сервисам, то результирующее время отклика будет увеличиваться и появятся «точки отказа». Для решения этой проблемы существуют два пути:

- изменение детализации вызовов, которое может привести к снижению их количества;
- внедрение асинхронности, отзывы выполняются параллельно, в результате это приводит к тому, что

конечное время отклика — это самое медленное время из всех, а не суммарное время всех задержек.

Постоянное усложнение процесса разработки, что приводит к повышению требований к квалификации программистов. В микросервисной архитектуре роль интеграционных процессов и процессов непрерывной доставки велика. И поэтому достаточно трудно обрабатывать множество процессов без автоматизации тестирования и развертывания сервисов. Это факторы требуют внедрения DevOps в компании и тесного сотрудничества разработчиков с системными инженерами, тестировщиками, службой тех. поддержки и т.д.

Децентрализация в микросервисной архитектуре порождает проблемы с согласованностью микросервисов. Например, в монолитном приложении за одну транзакцию возможно выполнение множества изменений, но и возможен откат назад, если произошел сбой, с сохранением согласованности данных. При применении микросервисов, возможна следующая ситуация: в случае нарушения работы одного из сервисов, перестает отвечать другой микросервис. В таком случае — это вопрос приоритетов разработчика: можно дать приоритет доступности компонентов (в случае выхода из строя одного сервиса, другие продолжают функционирование). В общем, разработчики должны находить баланс между согласованностью сервисов и их доступностью и делать это необходимо очень осторожно.

### **Заключение**

Перед тем как выбирать микросервисную архитектуру для разработки веб-приложений, разработчикам стоит оценить как ее преимущества, так и недостатки. Ведь неправильный выбор архитектуры, может отразиться в будущем на работоспособности и функциональности ПО. При неправильном применении микросервисной архитектуры у разработчиков могут возникнуть большие проблемы, которые нивелируют все преимущества микросервисов.

В статье выполнены следующие задачи: рассмотрены основные свойства и принципы микросервисов, описаны способы взаимодействия микросервисов между собой, проведен сравнительный анализ микросервисной и других типов архитектур.

### **Литература**

- 1) Создание микросервисов [Электронный ресурс] // КОД - Режим доступа [https://www.it.ua/uploads/0/800-sem\\_numen\\_sozdanie\\_mikroservisov\\_2016.pdf](https://www.it.ua/uploads/0/800-sem_numen_sozdanie_mikroservisov_2016.pdf)
- 2) Микросервисная архитектура в разработке приложений: преимущества и недостатки [Электронный ресурс] // КОД - Режим доступа <https://habr.com/en/post/682628/>
- 3) Взаимодействие в архитектуре микрослужб [Электронный ресурс] // КОД - Режим доступа <https://learn.microsoft.com/ru-ru/dotnet/architecture/microservices/architect-microservice-container-applications/communication-in-microservice-architecture>

**Настеко И.Р., Григорьев А.В. Особенности проектирования веб-приложения на основе микросервисной архитектуры.** В данной статье рассмотрены основные свойства и принципы микросервисов, описаны способы взаимодействия микросервисов между собой, проведен сравнительный анализ микросервисной и других типов архитектур. Были затронуты синхронные и асинхронные методы взаимодействия микросервисов. Сделаны выводы.

**Ключевые слова:** микросервис, архитектура, монолит, сервис, протокол, шина, запрос, ответ

**Nasteko I.R., Grigoriev A.V. Features of designing a web application based on microservice architecture.** This article discusses the main properties and principles of microservices, describes how microservices interact with each other, and conducts a comparative analysis of microservice and other types of architectures. Synchronous and asynchronous methods of interaction of microservices were touched upon. Conclusions are drawn.

**Keywords:** microservice, architecture, monolith, service, protocol, bus, request, response

## Организация поискового индекса при проектировании компьютерной библиотечной системы

Р.В. Мальчева<sup>\*1</sup>, Ю.Ю. Дьяченко<sup>\*2</sup>

<sup>\*1</sup> к.т.н, доцент, Донецкий национальный технический университет,  
raisa.malcheva@yandex.ru, OrcID: 0000-0003-4983-8878, SPIN-код: 7421-7049

<sup>\*2</sup> магистрант, Донецкий национальный технический университет,  
yurokke.yury@gmail.com

**Мальчева Р.В., Дьяченко Ю.Ю. Организация поискового индекса при проектировании компьютерной библиотечной системы.** В статье представлен краткий обзор современных проблем при организации поиска книг в информационных библиотечных системах. Выполнен анализ готовых решений для реализации системы и сформулированы требования к поисковой системе. Наиболее важным критерием использования современных способов поиска в компьютерных библиотечных системах является минимизация времени, затрачиваемого на поиск, а также уменьшение размеров баз данных, в которых записаны книги и другие электронные документы, хранящиеся в библиотечной системе. Осуществлен выбор средств организации поискового индекса. Намечены направления дальнейших исследований.

**Ключевые слова:** поиск, запрос, индекс, дерево, лемматизация, ElasticSearch, Lucene.

### Введение

Современные технологии компьютеризации и автоматизации библиотек позволяют обеспечивать учёт книжного фонда в электронном формате и организовывать доступ читателю удаленно при помощи сети Интернет [1]. Однако при большом количестве электронных документов появляется необходимость в организации удобного и быстрого поиска необходимой книги, иначе пользователю потребуется тратить часы личного времени, на то чтобы найти нужный файл, что, безусловно, является неприемлемо.

Поиск – важный процесс в любой системе. Благодаря нему пользователи и администраторы системы могут гибко получать доступ к необходимой информации. Обычно поиск начинается с формирования поискового запроса, который является краткой характеристикой искомого материала. Поисковый запрос – очень обширное понятие, это может быть, как набор ключевых слов/фраз, так и ряд условий, которым должна удовлетворять искомая информация.

В существующих информационных библиотечных системах электронные каталоги организованы таким образом, что пользователь вынужден формулировать конкретные поисковые запросы [2], в которых должно быть правильное название документа или полное совпадение года публикации или даже все фамилии авторов, если у книги их несколько.

Следует отметить, что поиск нужной информации остается проблемой для малоподготовленного пользователя [2], особенно если это касается тематического поиска. Поэтому задача эффективной организации поискового индекса является по-прежнему актуальной.

### Способы организации поиска

Поиск можно организовать с помощью обычного перебора всей хранимой информации и проверки на соответствие поисковому запросу каждой единицы информации – это называется простым поиском, что будет удобным и обладать приемлемым быстродействием, если нужно искать в сравнительно небольших объемах, к примеру, в одной книге. Но при достижении больших объемов данных, выполнять полный просмотр всей хранимой информации может быть в лучшем случае нерационально, а, зачастую, невозможно физически. Поэтому необходимо применять разные технические и другие ухищрения, чтобы оптимизировать процесс, но при этом не утратить функциональных возможностей поисковой системы. С этой задачей может справиться расширенный поиск, который подразумевает использование ключевых слов [3, 4].

Для организации поиска по ключевым словам, необходимо использовать дополнительную структуру данных – поисковый индекс. Он основан на следующей идее: если взять единицу информации, в нашем случае текст книги, и построить сбалансированное дерево (см.рис. 1), где вершинами будут выступать слова, используемые в тексте, то можно четко оценить, какой набор слов содержится в каждом тексте.

На таком принципе работают многие поисковые системы в сети «Интернет», в которых поиск происходит по всему тексту проиндексированных страниц, а не по отдельным заголовкам, тегам или полям. Такой способ называется полнотекстовым поиском.

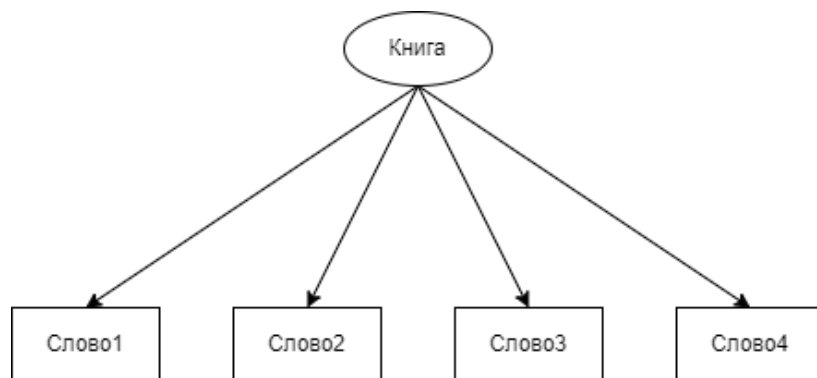


Рисунок 1 – Стандартное дерево поиска по ключевым словам

В случае организации поиска книг искать предстоит среди большого количества документов. Поэтому имеет смысл знать, что в каком документе лежит. В этом случае необходимо перевернуть дерево (см.рис. 2). Тогда будет производиться поиск по ключевому слову, к которому привязаны книги, что позволит в разы сократить время поиска по книгам. При добавлении новых данных дерево нужно будет достраивать таким же способом.

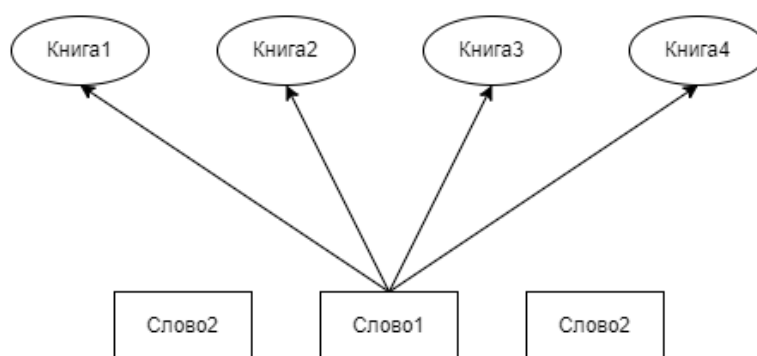


Рисунок 2 – Обратное дерево поиска по ключевым словам

В результате всегда точно известно, какое слово встречается в каких текстах. С учетом того, что в любом языке количество слов ограничено, то и дерево будет увеличиваться только за счет роста связей между узлами и добавлением одного узла на каждую новую единицу информации. Такой способ хранения ключевых слов позволит иметь базу данных в разы меньшего объема, чем при использовании стандартного способа, при котором хранятся книга-ключи, а не наоборот.

Описанный способ не является окончательным и наилучшим решением. Так как, например, любое слово имеет множество вариаций, с разными окончаниями, суффиксами, приставками и даже числами, хранить каждый вариант одного слова будет пустой тратой места и увеличением времени поиска в десятки раз. Следовательно, необходимо все слова приводить к их леммам – начальной, словарной форме слова. Этот процесс называется лемматизацией, при нем не важно, в какой конкретно форме находится слово. Использование лемм позволяет сократить количество исходных слов, которые будут храниться в качестве ключевых слов, что, следовательно, позволит уменьшить размер базы данных в десятки раз, ведь одно слово может встречаться в одной книге в своих десяти разных формах и, если хранить не в леммах, то только на одном слове будет несколько отдельных записей в базе данных, и к каждому будут записываться связанные с ним книги.

Существует ещё огромное количество оптимизаций и идей, применяемых в построении поисковых индексов в современных цифровых системах. Их описание займет большое количество времени, а реализация с нуля – еще больше. Поэтому лучшим вариантом будет ознакомиться с готовыми решениями и проверить, удовлетворяют ли они требованиям разрабатываемой системы.

Перейдем к формулированию требований к данным решениям при разработке информационной библиотечной системы.

### **Требования к поисковой системе**

Основные требования к поисковой системе заключаются в том, что она необходима для библиотечного сервиса, у которого есть свои специфические особенности при работе:

- необходим быстрый поиск;
- запись и удаление происходят редко, следовательно, высокая скорость не нужна;
- структура данных может активно изменяться и хранилище может заполняться из нескольких различных и независимых друг от друга источников.

Исходя из вышеперечисленного, можно начать поиск готовых решений, однако стоит помнить, что пользователю нужен расширенный поиск, следовательно, необходимо предоставить читателю возможность однозначно фильтровать книги, по автору, году, жанру и другим метаданным. Также необходимо обеспечить поиск по ключевым словам и целым фразам в описании книги. Индексация всей книги целиком для предоставления возможности поиска внутри ее текста может быть избыточной, ввиду специфики частоты использования данной функции и соотношения ее к затратам на хранение и обработку такого массива данных. Значит, поиск можно осуществлять в несколько этапов, и даже использовать для этого разные поисковые системы, например:

- первая – основная, для метаданных - большая часть запросов будет попадать в нее, так как обычно пользователи при поиске используют название, имя автора, год выпуска или описание, что не требует сильной нагрузки и не вызывает сложностей при поиске;
- вторая – не так часто используемая, но она сможет обрабатывать более сложные и работать с большим объемом данных.

При таком разбиении, нагрузка на одну поисковую систему не будет влиять на другую и тем более на всю цифровую библиотеку, что позволит достичь устойчивости системы при большой нагрузке.

Также необходимо упомянуть о важности обеспечения сохранности данных, в нашем случае, поискового индекса, ведь построение его с абсолютного нуля на больших объемах данных может потребовать значительного времени (недели или даже больше). Так как проектируется высоконагруженная система поиска, то индекс также необходимо масштабировать в соответствии со спецификой текущей нагрузки на каждом этапе жизни цифровой системы.

### **Выбор готового решения**

Для использования доступны такие поисковые системы: Solr, ElasticSearch, Sphinx, Splunk, Algolia и многие другие. Почти все они обещают в той или иной мере реализовать вышеописанные требования, так как они являются типовыми задачами в данной сфере. В рамках статьи выберем одну из систем и проверим, насколько она соответствует поставленным задачам. Сравнение произведем с ElasticSearch, ввиду ее популярности и небольшого опыта автора в его использовании, в то время как подобный опыт использования других систем вовсе отсутствует.

ElasticSearch (ES) — поисковая система, которая предоставляет интерфейс json rest api. Использует Lucene [5] и написана на языке программирования Java. Она хранит данные в виде документов, то есть является документо-ориентированной системой управления баз данных (СУБД). Такой подход выбран по причине того, что информация, используемая для индексации, может быть разделена на большое количество сущностей, и представление таких данных в классическом реляционном виде повлекло бы за собой создание большого количества связей и соответственно временные издержки при доступе к данным.

Огромным бонусом является то, что ElasticSearch предоставляет возможность использовать данные, как и в обычном СУБД, то есть не уступает в функционале аналогичным решениям. Это позволяет хранить часть данных только в ElasticSearch и не дублировать их. Для проектируемой библиотечной системы это полезно в двойне, ведь основная часть данных – книги - будут храниться в отдельном файловом хранилище, а метаданные можно хранить напрямую в ElasticSearch. Благодаря такому решению можно избавиться от отдельной СУБД для метаданных.

Lucene — популярная open-source библиотека полнотекстового поиска. Предоставляет основной функционал для построения поискового индекса.

ElasticSearch предоставляет возможность хранить данные распределено, то есть осуществлять горизонтальное масштабирование. Также используется и репликация данных, для отказоустойчивости. Поисковый индекс тоже может быть распределенным, что обеспечивает скорость и безопасность системы.

Для быстрых вычислений, а именно непосредственно поиска данных в соответствии поступившему запросу, данная поисковая система использует широко известный метод параллельных вычислений MapReduce. Особенностью является необходимость в рамках использования данного метода сначала предварительно обработать входные данные, сформулировать задачу и отправить ее распределенным узлам для независимого выполнения. После чего нужно собрать результат работы каждого узла и свернуть его, то есть произвести пост-обработку.

### **Выводы**

В современных информационно библиотечных системах поисковые сервисы требуют развития и адаптации, так как поиск не всегда практичен или слишком медленный, из-за чего эффективность эксплуатации хранилищ книг оставляет желать лучшего. Использование поискового индекса позволяет достичь высоких показателей быстродействия в библиотечных системах с большим количеством хранимых файлов, при этом уровень надежности и защищенности файлов будет на высоком уровне при невысоких материальных и производительных затратах.

Современные поисковые системы, такие как Elasticsearch, способны организовывать необходимую скорость поиска и возможность масштабирования книжного хранилища в электронной библиотеке, что позволяет снизить трудоемкость работы при проектировании системы и её дальнейшей поддержке. Исходя из этого направлением дальнейших исследований является реализация компьютерной библиотечной системы с реализацией поискового индекса на базе Elasticsearch.

### **Литература**

1. Дьяченко, Ю.Ю. Выбор технических средств для реализации хранилища данных при создании электронной библиотеки / Ю.Ю. Дьяченко, Р.В. Мальцева // Информационное пространство Донбасса: проблемы и перспективы : материалы V Респ. с междунар. участием науч.-практ. конф., 27 окт. 2022 г. – Донецк : ГО ВПО «ДонНУЭТ», 2022. – С. 118-120.
2. Организация поиска и навигации в электронном каталоге библиотеки [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/organizatsiya-poiska-i-navigatsii-v-elektronnom-kataloge-biblioteki>. - Загл. с экрана.
3. Стратегия поиска информации в библиотеке [Электронный ресурс]. – Режим доступа: [https://studopedia.ru/17\\_118391\\_strategiya-poiska-informatsii-v-biblioteke.html](https://studopedia.ru/17_118391_strategiya-poiska-informatsii-v-biblioteke.html). - Загл. с экрана.
4. Электронные каталоги библиотек: технология поиска и заимствования [Электронный ресурс]. – Режим доступа: [https://lib.lsept.ru/view\\_article.php?id=200901815](https://lib.lsept.ru/view_article.php?id=200901815). - Загл. с экрана.
5. С чего начинается Elasticsearch [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/489924/>. - Загл. с экрана.

**Мальцева Р. В., Дьяченко Ю. Ю. Организация поискового индекса при проектировании компьютерной библиотечной системы.** В статье представлен краткий обзор современных проблем при организации поиска книг в информационных библиотечных системах. Выполнен анализ готовых решений для реализации системы и сформированы требования к поисковой системе. Наиболее важным критерием использования современных способов поиска в компьютерных библиотечных системах является минимизация времени, затрачиваемого на поиск, а также уменьшение размеров баз данных, в которых записаны книги и другие электронные документы, хранящиеся в библиотечной системе. Осуществлен выбор средств организации поискового индекса. Намечены направления дальнейших исследований.

**Ключевые слова:** поиск, запрос, индекс, дерево, лемматизация, Elasticsearch, Lucene.

**Malcheva R. V., Dyachenko Yu. Yu. Organization of the search index in the design of a computer library system.** The article presents a brief overview of modern problems in organizing the search for books in information library systems. The analysis of ready-made solutions for the implementation of the system is carried out and the requirements for the search engine are formed. The most important criterion for using modern search methods in computer library systems is to minimize the time spent on searching, as well as to reduce the size of databases in which books and other electronic documents stored in the library system are recorded. The choice of means of organizing the search index has been made. The directions of further research are outlined.

**Keywords:** search, query, index, tree, lemmatization, Elasticsearch, Lucene.