

межвузовский сборник научных трудов

**МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

РГРТУ, 2023

Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овечкина – Рязань: РГРТУ им. В.Ф. Уткина, июнь 2023- 87 с.

Рецензент: к.т.н., доцент, зав. каф. «Информатики, информационных технологий и защиты информации» Липецкого государственного педагогического института Скуднев Д.М.

Редакционная коллегия

д-р техн. наук, проф., зав. каф. Г.В. Овечкин (отв. редактор, Рязанский государственный радиотехнический университет); д-р техн. наук, проф., засл. работник высшей школы РФ А.Н. Пылькин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. Е.Е. Ковшов (Московский государственный технологический университет «Станкин»); д-р техн. наук, проф. К.А. Майков (МГТУ им. Н.Э. Баумана); д-р техн. наук, проф. В.В. Белов (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.В. Золотарев (Институт космических исследований РАН, г. Москва); д-р техн. наук, проф. И.Ю. Каширин (Рязанский государственный радиотехнический университет); д-р техн. наук, проф. В.Н. Малыш («Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» г. Липецк); Н.Ф. Финажина (Рязанский государственный радиотехнический университет).

ISBN 978-5-7722-0387-3

Представлены статьи, посвящённые различным аспектам разработки программных средств ЭВМ, вычислительных систем и сетей, вопросам математического моделирования, методам обработки информации.

Предназначен для научно-педагогических работников вузов, инженеров, аспирантов и студентов старших курсов.

Авторская позиция и стилистические особенности публикаций полностью сохранены.

ISBN 978-5-7722-0387-3

© Коллектив авторов, 2023
© РГРТУ, 2023

ВВЕДЕНИЕ

Межвузовский сборник научных трудов содержит результаты научных исследований и разработок по следующим направлениям:

- программное обеспечение вычислительных систем, новые информационные технологии;
- прикладная математика, теория информации, искусственный интеллект;
- ЭВМ в системах обработки, управления и обучения;
- автоматизированное проектирование аппаратных средств вычислительных систем;
- информационные технологии в экономических и социальных системах.

Сборник сформирован на основе статей, в которых рассматриваются различные аспекты разработки программных средств ЭВМ, вычислительных систем и сетей, включая вопросы автоматизации проектирования, теории обработки информации и математического моделирования, и предназначен для студентов, аспирантов и преподавателей технических вузов и научных работников.

Материалы для сборника предоставлены сотрудниками и студентами:

- Федерального государственного бюджетного образовательного учреждения высшего образования «МИРЭА — Российский технологический университет», Москва;
- Федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный технический университет им. Н.Э. Баумана», г. Москва;
- Федерального государственного бюджетного образовательного учреждения высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

УДК 004.8

Белов В.В., Перехода И.Ю., Фаддеев А.О.**СПЕЦИФИКА ОБРАБОТКИ ТЕКСТОВ С ПОМОЩЬЮ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Дирекция по Арктическим программам федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный технический университет им. Н.Э. Баумана», Москва

В статье рассматриваются особенности применения нейронных сетей для обработки текстов на естественном языке, выделены существующие преимущества и недостатки использования методов машинного обучения для решения задач обработки текстов.

Нейросеть — алгоритм искусственного интеллекта, использующийся в машинном обучении [1]. Алгоритм, используя данные, должен решить определённую задачу, в процессе работы он учится находить закономерности в этих данных и автоматически их обрабатывать. Данные представляются в виде чисел, с которым алгоритм взаимодействует математически.

Одной из наиболее важных областей применения нейросетей является обработка естественного языка, известная как NLP (Natural Language Processing) [2]. Это технология, дающая компьютерам способность "понимать" естественный язык. Нейросети могут определять семантические связи между словами, учитывая контекст, и генерировать новый текст.

На основе нейросетей функционирует большое число сервисов, которые способны создавать различные текстовые материалы: электронные рассылки, отзывы о товарах, вступительные абзацы для статей и даже генерировать идеи и подходящие заголовки.

Прежде чем приводить конкретные примеры, стоит кратко объяснить, как нейросети генерируют и интерпретируют текст. В этом лучше всего справляются нейросети-трансформеры. Трансформер обрабатывает каждое слово в тексте, представляя это слово в виде вектора - группы чисел, и выполняет над ним различные математические операции. Это позволяет нейросети учитывать порядок слов и их влияние друг на друга, что в итоге определяет семантику текста.

На основе трансформеров создаются большие языковые модели. Они обучаются на основе задачи языкового моделирования: модель получает начало текста и должна его продолжить. Для обучения используются большие массивы текстов из различных источников, в том числе Интернета, которые покрывают широкий спектр тем. В результате,

такая модель может решать новые текстовые задачи без дополнительного обучения — это концепция, известная как Few-shot learning [3].

Процесс генерации текста с помощью нейронной сети состоит из нескольких шагов:

1. На первом этапе необходимо подготовить тестовые данные. Текст обычно разбивается на отдельные единицы, такие как отдельные символы, слова или фразы целиком, затем преобразуются в числовой формат, который может быть обработан нейронной сетью. Каждая единица текста обычно представляется в виде вектора (набор значений признаков, который подается на вход нейронной сети), где каждый элемент соответствует определенному символу или слову и имеет числовое представление.

2. На втором этапе нейронная сеть обучается на подготовленных данных. Обучение основано на предоставлении нейронной сети последовательности входных данных (например, предыдущих символов или слов) и ожидаемых выходных данных (следующий символ или слово). Нейронная сеть обновляет свои веса и параметры в процессе обучения, чтобы минимизировать ошибку между предсказанным и ожидаемым выходом.

3. На третьем этапе, после обучения модели, нейронную сеть можно использовать для генерации нового текста. Процесс начинается с задания начального входного состояния (например, начального символа или слова). Затем нейронная сеть генерирует предсказание для следующей единицы текста на основе текущего входа. Полученная единица текста добавляется к предыдущему контексту, и процесс повторяется для генерации следующей единицы текста. Этот процесс может быть продолжен на заданное количество шагов или до достижения определенного условия остановки (например, генерация заданного количества символов или пунктуационного знака).

Важно отметить, что качество генерируемого текста зависит от качества тестового набора данных, размера и структуры нейронной сети, а также от алгоритмов оптимизации и параметров обучения, используемых в процессе обучения модели. Помимо этого, следует отметить, что для улучшения качества генерации текста могут использоваться различные техники:

1. Управляемая генерация: вместо случайной генерации текста можно использовать управляемый подход, где начальные условия или ограничения задаются пользователем. Например, можно указать определенную тему, стиль или даже предоставить некоторые примеры текста, чтобы модель сгенерировала текст, соответствующий этим ограничениям.

2. «Температура» генерации: параметр «температуры» может контролировать степень случайности генерации. Более высокое значение

температуры делает генерацию более разнообразной, в то время как более низкое значение делает ее более детерминированной и предсказуемой.

3. Обрезка генерации: для получения более четких и ограниченных результатов можно применить обрезку генерации. Например, можно ограничить длину генерируемого текста или установить условие остановки, когда определенное слово или фраза появляется в выводе модели.

4. Смешение контекста: генерация текста может быть улучшена путем комбинирования нескольких источников контекста. Например, помимо предыдущего текста, можно включить дополнительную информацию, такую как изображения, заголовки или метаданные, чтобы модель могла генерировать текст, учитывая этот дополнительный контекст.

Таким образом, нейронные сети для генерации текста могут быть достаточно сложными и требуют значительных вычислительных ресурсов и времени для обучения. Они могут создавать связные и грамматически правильные предложения, но иногда могут проявлять тенденцию к генерации несвязного или нелогичного текста. Поэтому подбор правильных параметров и тщательная обработка данных являются важными шагами для достижения желаемых результатов при генерации текста с помощью нейронных сетей.

Нейронная сеть, способная обрабатывать запросы, сформированные на естественном языке, может существенно сократить поиск необходимой информации различного рода и для различных задач, однако использование данного инструмента имеет ряд существенных недостатков:

1. обучение и использование нейронных сетей может требовать значительных вычислительных ресурсов, включая высокопроизводительные процессоры и графические ускорители. Это может быть проблематично при работе с огромными объемами данных или на устройствах с ограниченными вычислительными возможностями;

2. для эффективной работы нейронные сети требуют большого объема размеченных обучающих данных. В случае поиска информации, где требуется широкий охват и актуальность данных, создание достаточного объема обучающих данных может быть сложной, а иногда вовсе невыполнимой задачей;

3. нейронные сети являются «черными ящиками», что означает, что их модель поведения и сформированные решения не всегда легко интерпретировать или объяснить. Это может создавать сложности в объяснении, как и почему нейронная сеть приняла определенное решение или предоставила определенный результат поиска.

Наряду со значительными преимуществами:

1. нейронные сети могут обрабатывать и анализировать огромные объемы данных, что может быть полезно при поиске информации из всего многообразия информационных источников в Интернете, однако стоит иметь в виду, что не все нейронные сети имеют доступ к актуальной информации, размещенной в Интернете;

2. нейронные сети могут обучаться на основе существующих данных и автоматически извлекать информацию из этих данных. Это позволяет им находить скрытые связи и зависимости, которые чаще всего сложно обнаружить с помощью традиционных методов поиска;

3. также нейронные сети способны обучаться на различных типах данных и задачах. Они могут быть приспособлены к поиску информации различного рода, а также могут быть модифицированы для оптимизации решения конкретно поставленных задач.

Несмотря на некоторые ограничения, нейронные сети все же имеют большой потенциал, позволяющий охватывать широкий спектр задач. Современные исследования активно разрабатывают новые методы и архитектуры нейронных сетей для более эффективного их использования.

Например, преодоление требований большого объема обучающих данных может быть достигнуто с помощью техник передачи обучения (transfer learning) [4] и генеративных моделей, которые позволяют использовать предварительно обученные модели на больших наборах данных и адаптировать их к конкретной задаче поиска.

Также стоит отметить, что нейронные сети могут быть интегрированы в более широкие системы поиска информации, комбинируя их с другими методами и техниками. Например, нейронная сеть может использоваться для ранжирования и сортировки результатов, полученных с помощью классических алгоритмов поиска или фильтрации.

В целом, нейронные сети представляют собой мощный инструмент для поиска информации, который может улучшить точность, скорость и адаптивность поисковых систем. Однако внедрение и использование нейронных сетей в реальных приложениях поиска информации требует тщательной настройки, обучения и оценки для достижения наилучших результатов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Flach Peter, Machine Learning. The Art and Science of Algorithms that Make Sense of Data, 2012.
2. Daniel Jurafsky, James H. Martin. Speech and Language Processing, 2023.
3. Archit Parnami, Minwoo Lee. Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. Department of Computer Science The University of North Carolina at Charlotte, 2022.

4. Qiang Yang, Yu Zhang, Wenyan Dai, Sinno Jialin Pan. Transfer Learning: 1st Edition, 2022.

УДК 004.9

Беляев А.В., Дмитриева Т.А.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ МЕДИЦИНСКОГО ЦЕНТРА

**Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань**

Статья посвящена разработке программного обеспечения для автоматизации деятельности медицинского центра на платформе 1С. В статье рассмотрены основные функциональные возможности разработанного программного обеспечения и показан интерфейс пользователя.

Целью данной работы является разработать программное обеспечение для автоматизации деятельности медицинского центра, которое позволит уменьшить количество действий пользователей и увеличить рентабельность за счёт внедрения сторонних сервисов и оптимизации.

Современный подход к решению этой проблемы состоит в разработке программного обеспечения, в нашем случае – расширения, для внедрения различных сервисов по интеграции и переписанные объекты типовой конфигурации медицинского центра, что позволит:

- сократить время на решение различных задач;
- повысить производительность системы;
- отслеживать динамические данные по эффективности производства;
- не затрагивать типовую конфигурацию, что облегчит дальнейшее обновление системы;

Создание программного обеспечения на базе 1С с применением типового функционала, а также внедрения различных сервисов, позволит медицинскому центру стоматологии меньше затрачивать время на выполнение бизнес-процессов. Главным достоинством разработанного расширения будет являться экономия ресурсов времени на разработку, изменение, а также степень внедрения в медицинский центр, так как она будет затрагивать минимальное количество объектов, что даст значительную экономию денежных средств, а также модернизацию программного продукта в долгосрочной перспективе.

На основании вышесказанного, разработка расширения по интеграции и оптимизации для автоматизации деятельности в медицинском центре на базе 1С является актуальной.

Разработанное при выполнении ВКР программное средство предназначено для:

- хранения данных внешних сервисов;
 - добавления, удаления заказ-нарядов;
 - добавления, удаления данных внешних сервисов;
 - изменения данных заказ-нарядов;
 - изменения данных внешних сервисов;
 - осуществления пользователями проверку соединения с центральной базой и внешними сервисами;
 - осуществления пользователями проверку данных о внешних сервисах;
 - осуществления пользователями проверку эффективности работы;
- Область применения – медицинские центры стоматологии, клиники стоматологии.

Под расширением для автоматизации деятельности медицинского центра посредством включения в это расширение внешние сервисы и оптимизацию программного кода для сотрудников (пользователей) необходимо понимать систему, которая позволяет составлять различного рода объекты конфигурации и отправлять или получать их через внешние сервисы, которые взаимодействуют с медицинским центром, хранить данные внешних сервисов, а также имеет возможность добавлять и изменять эти данные [1]. Кроме того, возможна проверка соединения с внешними сервисами и центральной базой.

После запуска программы на экране отображается форма входа в приложение (рисунок 1). Пользователь вводит логин и пароль или создаёт новую учетную запись.

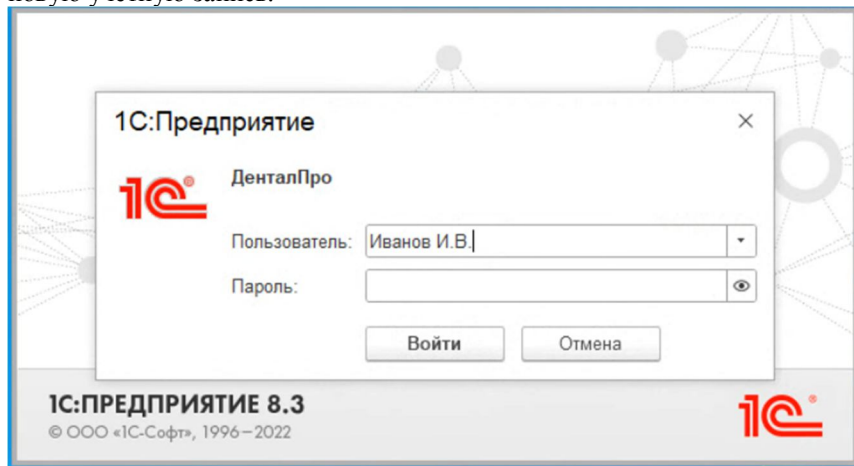


Рисунок 1 – Форма входа в приложение

После успешной авторизации пользователя открывается форма стартового экрана (рисунок 2).

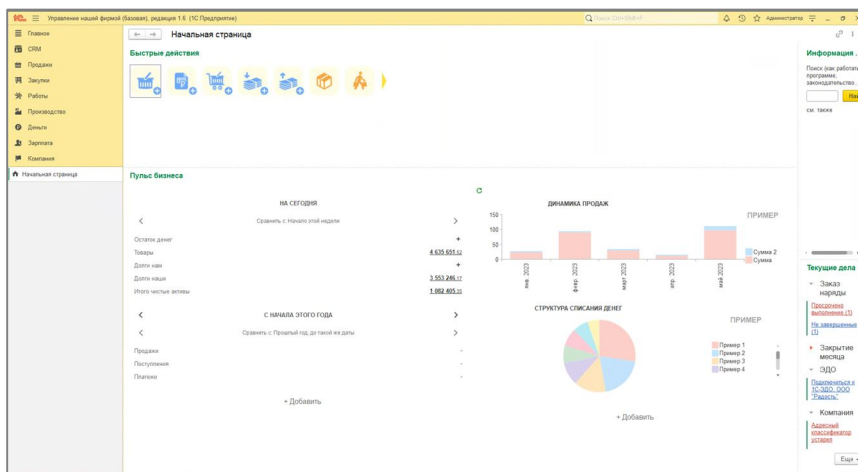


Рисунок 2 – Форма стартового экрана

На рисунке 3 показана форма рабочего стола, обработка «Рабочий стол».

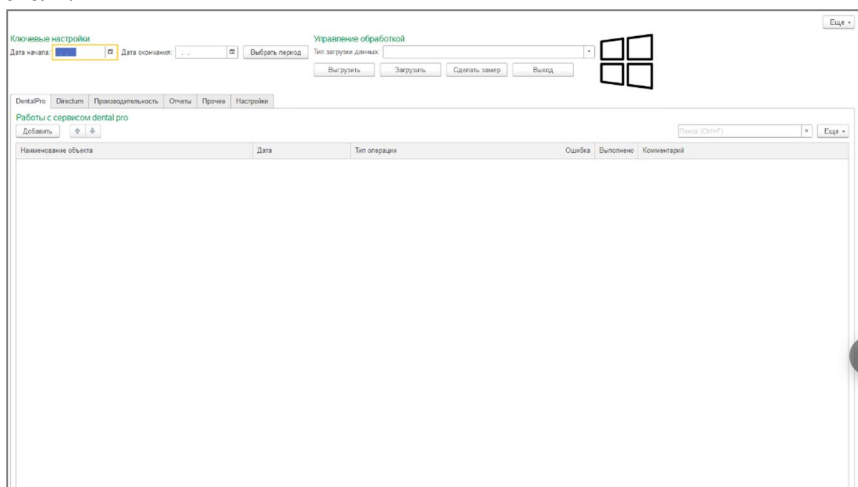


Рисунок 3 – Обработка «Рабочий стол»

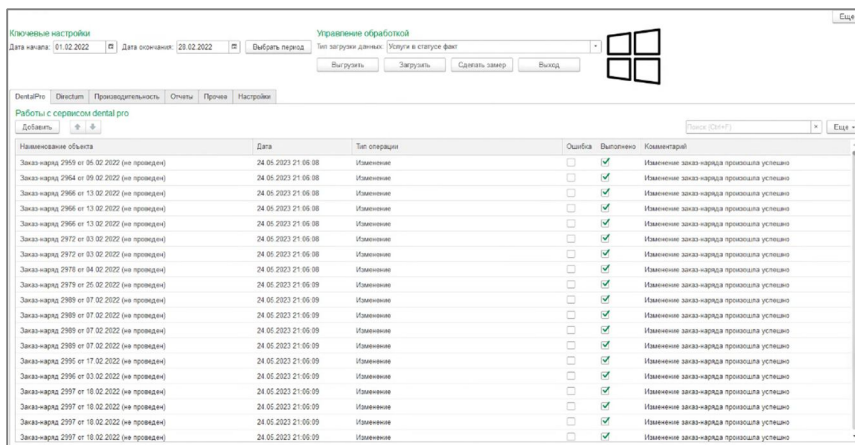


Рисунок 4 – Форма обработки «Рабочий стол» с выгруженными данными

Пользователь имеет возможность синхронизировать путем выбора разных параметров. После выполненной синхронизации, пользователю откроется сообщение с результатом обмена. В зависимости от обмена, ответ будет разным.

Данная обработка предназначена лишь для тестирования пользователем с правами «Администратор». Реальным пользователем без прав администратора данная обработка не будет доступна, так как всё будет выполняться в автоматическом режиме через регламентные задания и фоновые операции, что позволит увеличить быстродействие системы [2].

В результате реализации проекта был разработан программный продукт по автоматизации деятельности медицинского центра стоматологии посредством внедрения внешних сервисов на основе REST API и оптимизации программного кода.

Разработанное решение протестировано при помощи типичных сценариев использования приложения. В процессе тестирования ошибок в работе программного продукта не выявлено. Тестирование показало, что программа полностью соответствует заявленным требованиям к составу выполняемых функций, в связи с чем, работу по созданию программного обеспечения можно считать успешной.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ажеронков В.А., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Разработка управляемого интерфейса. – Издательство: ООО «1С-Паблишинг», 2013.

2. Филиппов Е.В. Настольная книга 1С: Эксперта по технологическим вопросам. – Москва: ООО «1С-Паблишинг», 2014.

УДК 004.4

Бубнов С.А., Трифонова Ю.С.**АНАЛИЗ СУЩЕСТВУЮЩЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В
КОМПЛЕКСНОЙ ОЦЕНКЕ СОСТОЯНИЯ ЗДОРОВЬЯ ДЕТЕЙ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В работе рассматривается существующее
программное обеспечение в комплексной оценке
состояния здоровья детей.

Развитие средств вычислительной техники повлекло за собой развитие комплекса диспансеризации и профилактической медицины и, соответственно, созданием автоматизированных информационных систем.

На территории Российской Федерации более известны следующие АИС для комплексной оценки состояния здоровья детей:

1. «ДИДЕНАС»;
2. «Автоматизированный комплекс диспансерных обследований» («АКДО»);
3. «АСПОН»;
4. «АРМИС».

«ДИДЕНАС» - одна из первых АИС мониторинга детского здоровья. Использовалась в начале 1980-х годов и включала в себя три подсистемы: контроля массовых профилактических осмотров; формирования групп риска и выявления детей с пограничными состояниями и ранними проявлениями заболеваний; контроля диспансеризации пациентов с хроническими заболеваниями и состояниями, требующими длительного наблюдения. В настоящее время не используется [1].

«АКДО» представляет собой комплекс автоматизированных информационных систем. Включает в себя два варианта исполнения диспансеризации: обследование детей и подростков «АКДО-ДП» и обследование взрослых «АКДО-В». Одним из преимуществ данного комплекса является высокая точность результатов обследования. Для работы с «АКДО» необходимо высшее медицинское образование.

«АСПОН», так же как «АКДО» - это целый комплекс АИС, выпускается в трех исполнениях: «АСПОН-Дт» (охватывает детей от 3-х до 14-ти лет), «АСПОН-Дт-П» (11,5 - 18,5 лет) и «АСПОН-РВ» (1 нед. до 3 лет). Комплекс позволяет проводить профилактический медицинский осмотр по многим профилям (ортопедии, стоматологии, неврологии, кардиологии, хирургии и др.).

«АРМИС» – это моноблочный, малогабаритный, универсальный аппаратно-программный комплекс. К нему через цифровой интерфейс подключаются датчики для регистрации физиологических параметров

человека. Данный комплекс удобно использовать в муниципальных бюджетных образовательных учреждениях, так как он способен заменить несколько приборов и аппаратов. Уникальной особенностью аппарата является то, что обслуживается он одним оператором со средним медицинским образованием.

Анализ функциональных возможностей рассматриваемых автоматизированных информационных систем позволил выявить их общие черты:

1. регистрация детей и корректировка регистрационных данных;
2. ввод и корректировка данных анамнеза;
3. ввод и корректировка данных антропометрии;
4. ввод и корректировка данных обследования;
5. получение заключения и его печать;
6. получение расшифровки и её печать.

Главное отличие – это количество педиатрических профилей, по которым проводится комплексная оценка состояния здоровья детей.

Результат работы систем – это формирование заключения – карты профилактического медицинского осмотра несовершеннолетнего.

В настоящее время большое внимание уделяется комплексной оценке состояния здоровья детей для своевременного выявления патологий, так как формирование организма ребенка имеет такие критические периоды, когда он подвержен наиболее высокому риску развития пограничных и патологических состояний [2].

Таким образом, развитие и создание новых автоматизированных систем позволит более детально производить комплексную оценку состояния здоровья детей, что в свою очередь позволит на ранних стадиях обнаружить заболевания и принять плановые или экстренные меры.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Игишева Л.Н. Информационные системы в комплексной оценке и прогнозе состояния здоровья детей школьного возраста. // Вестник новых медицинских технологий. – 2004. – №1 – 2. – С. 5.
2. Основы здоровья детей и подростков. Комплексная оценка здоровья детей и подростков. Часть I: руководство для врачей / Т.В. Бородулина, Н.Е. Санникова, Л.В. Левчук и др. – Екатеринбург: УГМУ, 2018. – 91 с.

УДК 517.925

Вапилина Е.Г., Симонова А.И., Билык Н.В., Дейнего И.С.**ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОБЫЧНЫХ
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Данная статья представляет обзор современных способов решения обычных дифференциальных уравнений: метод Эйлера, метод Хойна, метод середины и метод Рунге-Кутты 4-го порядка.

Обычные дифференциальные уравнения (далее ОДУ) применяются для математического описания природных явлений. Они широко применяются для описания биологических популяций, физических законов, а также в моделях экономической динамики. В ОДУ отражается не только зависимость переменных от времени, но и их взаимосвязь.

Решение ОДУ является важным инструментом для моделирования и анализа различных процессов и явлений. Однако, аналитическое решение возможно только для небольшого числа простых уравнений. В большинстве случаев, решение ОДУ может быть получено только численными методами, которые позволяют получать приближенные решения с заданной точностью. Однако, эти методы могут быть неустойчивыми, неэффективными или давать неточные результаты, особенно при решении сложных уравнений. Соответственно для каждого случая необходимо рассмотреть и сравнить способы решения ОДУ.

При рассмотрении существующих методов решения для сравнения эффективности и точности были выбраны для реализации следующие методы: метод Эйлера, метод Хойна, метод середины и метод Рунге-Кутты 4-го порядка.

Метод Эйлера является одним из наиболее простых и распространенных методов для численного решения обыкновенных дифференциальных уравнений (ОДУ). Он основан на аппроксимации значения функции в следующей точке с помощью значения функции в текущей точке и ее производной.

Рассмотрим ОДУ первого порядка:

$y' = f(x, y)$, где $y(x)$ - искомая функция, $f(x, y)$ - заданная функция. Для решения этого уравнения методом Эйлера необходимо задать начальное условие: $y(x_0) = y_0$, где x_0 и y_0 - известные значения.

Идея метода Эйлера заключается в том, что мы можем приближенно вычислить значение функции в следующей точке $x_1 = x_0 + h$, где h - шаг метода, используя значение функции в текущей точке x_0 и ее производную в этой точке:

$$y_1 = y_0 + h f(x_0, y_0)$$

Таким образом, мы получаем приближенное значение функции в следующей точке[1]. Затем мы можем продолжать этот процесс, используя полученное значение y_1 как новое начальное условие и повторяя вычисления для следующей точки $x_2 = x_1 + h$:

$$y_2 = y_1 + h f(x_1, y_1)$$

И так далее, пока не достигнем конечной точки x_N .

Метод Эйлера имеет некоторые ограничения и недостатки, такие как неустойчивость при больших значениях шага h и возможность накопления ошибок при длительном интегрировании. Однако он все еще широко используется благодаря своей простоте и универсальности.

Метод Хойна является улучшением метода Эйлера и позволяет уменьшить ошибку приближенного решения ОДУ. Он основан на аппроксимации значения функции в следующей точке с помощью значения функции в текущей точке и ее производной, вычисленной в средней точке между текущей и следующей точками.

Для решения ОДУ первого порядка методом Хойна необходимо задать начальное условие:

$$y(x_0) = y_0$$

где x_0 и y_0 - известные значения. Затем мы можем приближенно вычислить значение функции в следующей точке $x_1 = x_0 + h$, где h - шаг метода, используя следующую формулу:

$$y_1 = y_0 + h f(x_0 + h/2, y_0 + (h/2) f(x_0, y_0))$$

Эта формула использует среднее значение производной на интервале $[x_0, x_1]$, что позволяет уменьшить ошибку приближенного решения.

Затем мы можем продолжать этот процесс, используя полученное значение y_1 как новое начальное условие и повторяя вычисления для следующей точки $x_2 = x_1 + h$:

$$y_2 = y_1 + h f(x_1 + h/2, y_1 + (h/2) f(x_1, y_1))$$

И так далее, пока не достигнем конечной точки x_N .

Метод Хойна является более точным, чем метод Эйлера, но также имеет свои ограничения и недостатки. Он может быть неустойчив при больших значениях шага h и требует вычисления производной в середине интервала, что может быть затруднительно для некоторых функций. Однако он все еще широко используется благодаря своей универсальности и возможности уменьшения ошибки приближенного решения.

Метод середины (или метод Рунге-Кутты второго порядка) является усовершенствованным вариантом метода Эйлера. Он также используется для численного решения обыкновенных дифференциальных уравнений первого порядка.

Идея метода середины заключается в том, что мы используем среднее значение производной функции на интервале $[x_0, x_1]$ для

приближенного вычисления значения функции в точке x_1 . Для этого мы сначала вычисляем значение функции в точке $x_{1/2} = (x_0 + x_1)/2$, используя метод Эйлера:

$$y_{1/2} = y_0 + (x_{1/2} - x_0) f(x_0, y_0)$$

Затем мы используем это значение для приближенного вычисления значения функции в точке x_1 :

$$y_1 = y_0 + (x_1 - x_0) f(x_{1/2}, y_{1/2})$$

Таким образом, мы получаем более точное приближенное значение функции в следующей точке, чем при использовании метода Эйлера.

Метод середины также имеет свои ограничения и недостатки, но он является более точным и устойчивым методом, чем метод Эйлера. Он также может быть расширен до методов Рунге-Кутты более высоких порядков для еще более точного численного решения дифференциальных уравнений.

Метод Рунге-Кутты 4-го порядка является одним из наиболее распространенных методов численного решения обыкновенных дифференциальных уравнений[2]. Он использует четыре значения производной функции на интервале $[x_0, x_1]$, чтобы приближенно вычислить значение функции в точке x_1 .

Сначала мы вычисляем значение производной функции в точке x_0 :

$$k_1 = f(x_0, y_0)$$

Затем мы используем это значение для вычисления трех промежуточных значений:

$$k_2 = f(x_0 + h/2, y_0 + h/2 k_1)$$

$$k_3 = f(x_0 + h/2, y_0 + h/2 k_2)$$

$$k_4 = f(x_0 + h, y_0 + h k_3)$$

Здесь $h = (x_1 - x_0) / n$, где n - число шагов разбиения интервала $[x_0, x_1]$.

Наконец, мы используем эти промежуточные значения для приближенного вычисления значения функции в точке x_1 :

$$y_1 = y_0 + h/6 (k_1 + 2k_2 + 2k_3 + k_4)$$

Этот метод имеет более высокую точность, чем методы Эйлера и середины, и может быть использован для решения более сложных дифференциальных уравнений. Однако он также требует больше вычислительных ресурсов и может быть менее эффективным для больших систем уравнений или при использовании нелинейных функций.

Сравнение методов решения ОДУ.

Общая формула сложности для всех этих методов: $O(n)$, где n - количество шагов.

Общая формула погрешности для всех этих методов: $O(h^p)$, где h - размер шага, а p - порядок метода (2 для метода Эйлера, 3 для метода Хойна и середины и 5 для метода Рунге-Кутты 4 порядка).

При исследовании были рассмотрены многие распространенные и современные методы для решения обычных дифференциальных уравнений, каждый из которых обладает своими достоинствами и недостатками в работе. Выбирая тот или иной алгоритм, необходимо четко понимать цели расчета, имеющиеся в запасе ресурсы, необходимую точность расчета и другие факторы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пименов В. Г., Ложников А. Б. ЧИСЛЕННЫЕ МЕТОДЫ / Пименов В. Г., Ложников А. Б. – М.: Изд-во Уральского ун-та, 2014.
2. Крайнов А.Ю., Моисеева К.М. Численные методы решения краевых задач для обыкновенных дифференциальных уравнений / Крайнов А.Ю., Моисеева К.М. – Томск, 2016.

УДК 517.925

Воробьев Я.В., Кузнецов М.Г., Журавлев П.В.

ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОДУ ДЛЯ МОДЕЛИРОВАНИЯ ОБЪЕКТОВ В ВИДЕОИГРАХ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В данной статье осуществляется обзор способов решения обычных дифференциальных уравнений, которые используются в компьютерной графике: RK4, прямой метод Эйлера и метод средней точки.

Обычные дифференциальные уравнения (далее по тексту ОДУ) позволяют описывать динамику систем, которые изменяются во времени, и находят широкое применение в физике, химии, биологии, экономике. В том числе, ОДУ имеют важное значение при моделировании систем объектов в такой распространенной области, как создание видеоигр. Движение волн и листья, перемещение игрового персонажа, полет пули, скольжение автомобиля на повороте в гонках – все это программируется на основе расчетов производимых с помощью обычных дифференциальных уравнений.

Многие системы могут быть описаны только с помощью сложных для аналитического решения дифференциальных уравнений. Поэтому исследование применения ОДУ для моделирования и реализации объекта в визуальном пространстве остается актуальной задачей в настоящее время.

При выборе для реализации наиболее перспективных вычислительных алгоритмов среди существующих и применяемых для моделирования объектов в компьютерных играх были рассмотрены следующие: метод RK4 (RK4, Runge-Kutta 4) и некоторые другие.

Алгоритм RK4-метода описывается следующим образом: на каждом шаге сначала вычисляем значения функции и ее производных в

текущей точке. Затем используем эти данные, чтобы вычислить значения функции и ее производных на полпути к следующей точке. После применяем эту информацию, чтобы вычислить другие значения функции и ее производных на полпути к следующей точке, и т.д. В конечном итоге, получаем значение функции в следующей точке, используя эти приближения[2].

Метод Рунге-Кутты является универсальным методом для решения дифференциальных уравнений и систем. Он также является более точным, чем метод Эйлера или метод средней точки.

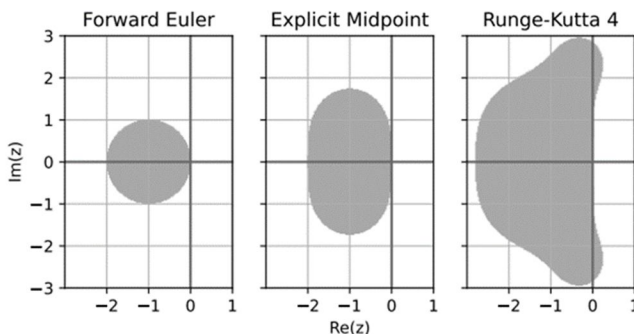


Рисунок 1 – Области устойчивости для явных методов (слева-направо: прямой метод Эйлера, метод явной средней точки и метод Рунге-Кутты четвертого порядка).

На рисунке 1 можно наблюдать, что область стабильности метода RK4 более высокого порядка, чем у двух левых методов. Фактически, если мы рассмотрим вычислительную сложность каждого временного шага для этих алгоритмов, метод RK4 обычно превосходит задачи, в которых временной шаг определяется стабильностью. Знание теоретической точности решения важно по ряду причин, и одна из них заключается в том, что оно предоставляет метод проверки реализации методов. Если можно продемонстрировать, что ошибка ведет себя так, как подсказывает теория, — это хороший признак того, что алгоритм реализован правильно[1].

ForwardEuler			ExplicitMidpoint			RungeKutta4		
Time step (dt)	Error (e)	e/dt	Time step (dt)	Error (e)	e/dt	Time step (dt)	Error (e)	e/dt
0.0033333	0.0001838	0.0551	0.0033333	0.0000002	0.0001	0.0033333	0.0000000	0.0000
0.0016667	0.0000920	0.0552	0.0016667	0.0000001	0.0000	0.0016667	0.0000000	0.0000
0.0008333	0.0000460	0.0552	0.0008333	0.0000000	0.0000	0.0008333	0.0000000	0.0000
0.0004167	0.0000230	0.0552	0.0004167	0.0000000	0.0000	0.0004167	0.0000000	0.0000
0.0002083	0.0000115	0.0553	0.0002083	0.0000000	0.0000	0.0002083	0.0000000	0.0000
0.0001042	0.0000058	0.0553	0.0001042	0.0000000	0.0000	0.0001042	0.0000000	0.0000
0.0000521	0.0000029	0.0553	0.0000521	0.0000000	0.0000	0.0000521	0.0000000	0.0000
0.0000260	0.0000014	0.0553	0.0000260	0.0000000	0.0000	0.0000260	0.0000000	0.0000
0.0000130	0.0000007	0.0553	0.0000130	0.0000000	0.0000	0.0000130	0.0000000	0.0000
0.0000065	0.0000004	0.0553	0.0000065	0.0000000	0.0000	0.0000065	0.0000000	0.0000

Рисунок 2 – Оценка погрешности решения ОДУ, полученная программным расчетом (слева-направо: прямой метод Эйлера, метод средней точки и RK4).

Метод Рунге-Кутты четвертого порядка имеет следующую вычислительную сложность: для каждого шага метода RK4 необходимы 4 вычисления правых частей дифференциальных уравнений, что соответствует 4 тестам на условие и 12 арифметическим операциям. Количество шагов, которые требуются для достижения достаточной точности, зависит от конкретной задачи и размера шага h .

Временная сложность метода RK4 составляет примерно $O(h^4)$, что означает, что уменьшение шага на 10 раз приведет к уменьшению ошибки на 10^4 раз.

В целом, метод RK4 дает достаточно точные результаты при разумном количестве шагов и при этом имеет небольшую вычислительную сложность.

Подводя итог, RK4 демонстрирует более эффективные показатели чем другие рассмотренные способы и может быть рекомендован для решения обычных дифференциальных уравнений в сфере моделирования объектов в видеоиграх.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Гольденвейзер А.Л. Решение дифференциальных уравнений в примерах и задачах / Гольденвейзер А.Л. – М.: Высшая школа, 1983
- 2 Гольдберг В.В. Численное моделирование процессов в механике сплошной среды / Гольдберг В.В., Крылов М.В., Кузнецов А.А. – Новосибирск: Наука, 1995

УДК 004.9+338.46

Дмитриева Т.А., Лапкин М.И., Лягушева М.А.**РАЗРАБОТКА НА ПЛАТФОРМЕ 1С МОБИЛЬНОГО ПРИЛОЖЕНИЯ
СИСТЕМЫ ОБСЛУЖИВАНИЯ ДЛЯ ЗАКАЗА ТОВАРОВ И УСЛУГ****Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань**

Статья посвящена разработке мобильного приложения на платформе 1С, предназначенного для системы обслуживания заказов товаров и услуг. В статье рассмотрены основные функциональные возможности приложения для упрощения процесса заказа товаров и услуг, что значительно повышает удобство обслуживания клиентов.

Актуальность данной научной статьи заключается в том, что современный рынок требует от предприятий быстрого и эффективного обслуживания клиентов. Однако отсутствие должного контроля приема и обработки заказов может привести к задержкам в доставке товаров и услуг, ухудшению качества обслуживания клиентов и, как следствие, потере клиентской базы. Это особенно актуально для стремительно развивающихся предприятий, которые не всегда успевают адаптироваться к новым требованиям рынка. Разработка мобильного приложения, способного эффективно обрабатывать заказы и упрощать процесс обслуживания клиентов, может значительно улучшить работу предприятий и повысить их конкурентоспособность на рынке. Таким образом, данная статья является актуальной и важной для бизнеса в целом.

Одним из наиболее перспективных инструментов для разработки мобильных приложений является платформа 1С[1]. Эта платформа обладает рядом технических преимуществ, которые делают ее идеальным выбором для создания приложений, способных эффективно обрабатывать заказы и упрощать процесс обслуживания клиентов. В частности, платформа 1С обеспечивает высокую производительность приложений, быстрый доступ к базе данных, возможность интеграции с другими системами и сервисами, а также широкие возможности для настройки интерфейса. Благодаря этим преимуществам, приложения на платформе 1С могут быстро и без проблем обрабатывать большие объемы данных и обеспечивать высокое качество обслуживания клиентов.

Приложение, реализация которого представлена в данной публикации, подходит для ресторанного бизнеса, так как масштабируемость исходного приложения позволяют легко изменить оболочку для потребностей производства. Приложение реализовано для 1С платформы на базе операционной системы Android.

Системные требования.

1. Операционная система: Android 4.1 — 11.X.
 2. Процессор: Intel x86, x86_64, ARM с архитектурой ARMv7a, ARM64.

3. Операционная память: минимальный размер 256 мегабайт.

4. Прочие требования: сенсорный экран.

Логическая модель базы данных является важным инструментом для организации и хранения информации в системе. В данной модели используются таблицы *Меню*, *Категории*, *Корзина*, *Товары*, *Заказы* и *Статусы заказов*. Категории служат для упорядочивания списков меню, которые в свою очередь представляют собой набор товаров, доступных для заказа. Каждый товар описывается объектом, содержащим информацию об описании и цене. Корзина представляет собой временное хранилище товаров, выбранных пользователем для заказа. Заказы и статусы заказов отражают информацию о процессе обработки заказов и их текущем состоянии. Такая логическая модель позволяет эффективно управлять информацией о товарах и заказах в системе, обеспечивая быстрый доступ к необходимой информации и удобную работу пользователям. Логическое представление базы данных изображено на рисунке 1.

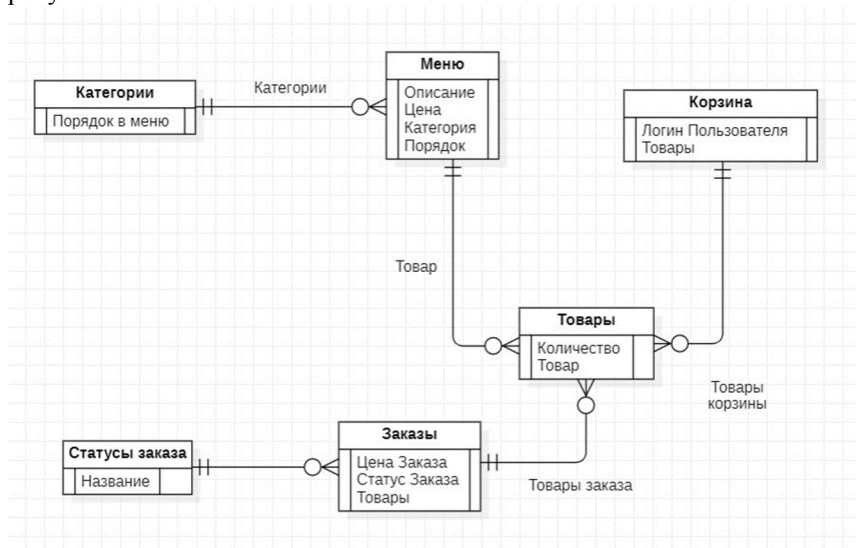


Рисунок 1 – Логическое представление базы данных

Ролевая модель является неотъемлемым элементом системы обслуживания для заказа товаров и услуг, поскольку она позволяет эффективно управлять информацией и процессами в системе. Эта модель определяет три основные роли – *Пользователь*, *Администратор* и

Исполнитель, каждая из которых играет свою ключевую роль в процессе заказа и обслуживания клиентов. Администратор контролирует и управляет информацией в системе, обеспечивая эффективное управление товарами и заказами. Исполнитель отвечает за выполнение заказов и обеспечение высокого качества обслуживания клиентов. Пользователь же использует функциональность системы для выбора и заказа товаров или услуг. Таким образом, ролевая модель является необходимым инструментом для эффективного управления процессами в системе обслуживания и обеспечения удовлетворения потребностей клиентов.

Роль администратора в системе обслуживания для заказа товаров и услуг является критически важной для эффективной работы системы. Он отвечает за управление информацией о сотрудниках, товарах и заказах. Действия администратора, такие как добавление нового сотрудника, удаление сотрудника, редактирование товара, добавление нового товара, удаление товара и распределение товаров, позволяют эффективно управлять информацией в системе. Например, добавление нового сотрудника позволяет увеличить производительность системы, а удаление сотрудника необходимо для контроля доступа к системе. Редактирование товара и добавление нового товара помогают обновлять информацию о товарах и предоставлять пользователям актуальную информацию. Удаление товара позволяет избежать ошибок при заказе, а распределение товаров помогает управлять отображением меню у пользователя. В целом, действия администратора имеют большое значение для эффективного функционирования системы обслуживания для заказа товаров и услуг.

Роль исполнителя в системе обслуживания для заказа товаров и услуг играет ключевую роль в обеспечении высокого уровня качества обслуживания клиентов. Исполнитель выполняет ряд важных действий, таких как изменение статуса заказа, подтверждение заказа, подтверждение возможности доставки и уведомления о новом заказе. Кроме того, исполнитель может просмотреть все отмененные заявки и использовать геолокацию для определения адреса доставки. Все эти действия позволяют исполнителю эффективно управлять заказами и обеспечивать своевременную доставку товаров или услуг, что является важным фактором для удовлетворения потребностей клиентов и повышения уровня их лояльности к компании.

Роль пользователя в системе обслуживания для заказа товаров и услуг имеет ключевое значение. Пользователь может добавлять товары или услуги в корзину, изменять ее содержимое и получать уведомления о статусе заказа. Кроме того, пользователь может использовать геолокацию для определения адреса доставки. Эти действия позволяют пользователю эффективно управлять своими заказами, что является важным фактором для удовлетворения потребностей клиентов и повышения уровня их лояльности к компании.

При разработке появилась необходимость в соотношении объектов, предусмотренных в процессе анализа, на существующие объекты 1С конфигурации [2].

1. Общие формы – для отображения состояния системы.
2. Справочники – в них представлена основная часть реализации реализации логической модели базы данных необходимые для работы системы.
3. Перечисления – для отображения заранее заданных элементов системы (Статусы заказа).
4. Параметры сеанса – для отображения информации о пользователе действующим в системе.

На форме в конфигураторе (рисунок 2) расположены основные реквизиты: таблица с наименованием и описанием товара, ценой и функцией «добавить в корзину».

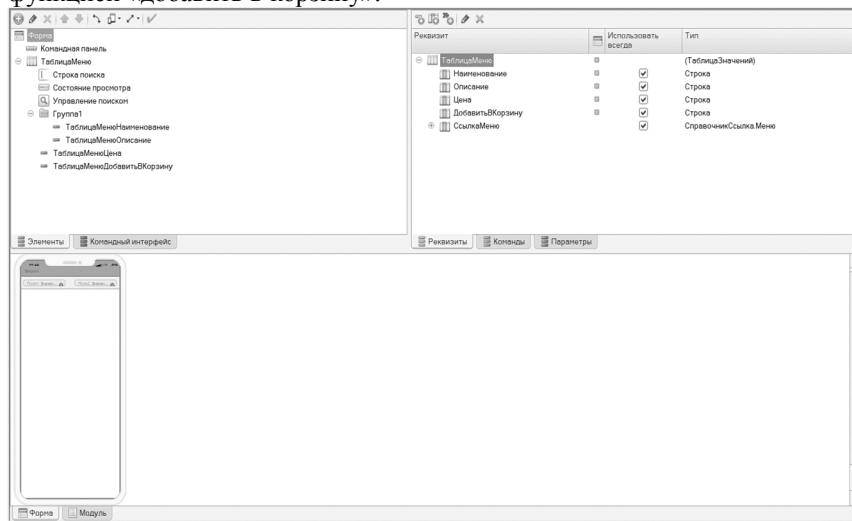


Рисунок 2 – Основная форма в конфигураторе

В ходе исследования было разработано мобильное приложение на платформе 1С, предназначенное для системы обслуживания заказов товаров и услуг. Приложение обладает широким спектром функциональных возможностей и эффективно интегрируется с базой данных, что значительно упрощает процесс заказа товаров и услуг. Благодаря этому, удобство и скорость обслуживания клиентов повышаются, что является важным фактором в современном бизнесе. В целом, разработанное приложение является перспективным решением для оптимизации различных бизнес-процессов и повышения качества обслуживания клиентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Хрусталева Е.Ю. Знакомство с разработкой мобильных приложений на платформе «1С:Предприятие 8». – М.: ООО «1С-Паблишинг», 2022.

2. Султанова А.И., Шаронова А.А. 1С:Счетчик ворон. Мастер-класс по мобильной разработке в среде «1С:Предприятие» . – М.: ООО «1С-Паблишинг», 2022.

УДК 004.9

Дмитриева Т.А., Лутиков В.А., Малинин А.Д., Фирсова Е.Д.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ 1С ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ КРЕАТИВНОГО ПРОСТРАНСТВА

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В статье рассматривается разработанное мобильное приложение на платформе 1С для автоматизации работы общественного пространства.

В современном мире, где время ценится особенно высоко, эффективность является ключевым фактором успеха. Нехватка времени становится все более распространенной проблемой, и каждый человек стремится оптимизировать свои затраты времени: время на дорогу или время организацию мероприятий [1].

Одним из препятствий на пути к эффективности являются архаичные способы получения информации, которые требуют присутствия человека и, следовательно, снижают его продуктивность. Такие методы могут стать невыгодными и неэффективными, поскольку человек стремится максимально сократить свои затраты времени и трудозатраты.

В настоящее время активно развиваются общественные креативные пространства, которыми не так просто управлять без электронного документооборота. Создание таких пространств имеет особое значение и актуальность для всех заинтересованных сторон: представителей креативных индустрий, крупного бизнеса, городских властей и жителей. Государство активно развивает общественные пространства, так как они способствуют быстрому росту экономики города за счет коммерциализации креативных разработок на производственных мощностях крупного бизнеса. Кластеры креативных пространств особенно важны для депрессивных индустриальных городов, так как они способны обеспечить качественные социальные и экономические результаты.

Для резидентов кластеров выгоды заключаются в инфраструктуре: они могут сконцентрироваться на своей деятельности, не отвлекаясь на

вопросы бэк-офиса, так как эти услуги обычно предоставляются в рамках создания креативной экосистемы. Для собственников и операторов кластеров это становится доходным бизнесом. Креативные кластеры могут находиться в бывших промышленных зданиях или целых кварталах, которые ранее считались маргинальными или неблагополучными. Творческие люди, художники и предприниматели приносят с собой оригинальные и недорогие решения для переустройства этих зданий и создания новых общественных пространств.

Такое развитие имеет важное значение в создании среды нового качества. Районы начинают возрождаться и становятся привлекательными для работы и жизни. Это дает импульс для развития, начинает развиваться малый бизнес, туристы и горожане стремятся в такие точки за культурным опытом и досугом.

Приложение предназначено для автоматизации работы креативного кластера.

При разработке приложения выявлены следующие роли [2].

Администратор отвечает за работу пространства и имеет следующие возможности:

- добавление сотрудников, оборудования, контрагентов;
- создание мероприятий;
- формирование отчетов;
- осуществление вызовов внутри приложения.

Пользователь арендует пространство и имеет следующие возможности:

- создание мероприятий;
- просмотр запланированных мероприятий;
- просмотр пространств, сотрудников, других арендаторов;
- поиск оборудования по штрих-коду;
- использование карты для просмотра местоположения пространства.

Программист имеет полный доступ к приложению, объединяя в себе возможности предыдущих ролей, работает с ним в режиме конфигуратора.

Современные приложения для мобильных устройств становятся все более продвинутыми, и многие из них предоставляют уникальный функционал, который требует хранения большого объема информации на сервере [3].

Когда пользователь взаимодействует с приложением, он взаимодействует не только с данными, которые хранятся на его устройстве, но и с данными, которые находятся на сервере. Для того чтобы эти данные были доступны пользователю, приложение использует парсер, который обеспечивает обмен данными между сервером и клиентом.

Парсер получает информацию с сервера и обрабатывает ее в соответствии с требованиями приложения. Например, если администратор открывает страницу в приложении, парсер загружает информацию о административную информацию, которую не должен увидеть пользователь.

Важно понимать, что хранение информации на сервере может повысить скорость работы приложения и обеспечить более широкие возможности для пользователей, но также может повысить уровень риска в случае, если сервер станет объектом атаки злоумышленников. Поэтому важно выбирать надежные провайдеры услуг хранения данных и обеспечивать достаточный уровень защиты информации.

Для достижения наибольшей эффективности использования приложения, в него добавлены печатные формы, которые позволяют вести электронный документооборот. Например, документы по конкретному мероприятию для налоговой или документы по мероприятиям за месяц для ведения аналитики.

Для обеспечения коммуникации между пользователями и администратором создан telegram-бот, который подключен к базе [4].

Разработанное приложение для автоматизации работы креативного кластера представляет собой эффективный инструмент для управления пространством и мероприятиями. Оно позволяет упростить и ускорить процессы, связанные с арендой пространства, созданием и организацией мероприятий, а также ведением электронного документооборота. При этом важно обеспечить надежность и безопасность хранения данных на сервере, выбрав надежного провайдера услуг и обеспечив при этом достаточный уровень защиты информации. Разработанное приложение позволяет удовлетворить потребности всех ролей, включая администратора, пользователя и программиста, и может быть полезным инструментом для управления креативными пространствами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кови С., Джонсон С. Фокус: Достижение приоритетных целей. – М.: Альпина Диджитал, 2011.
2. Хрусталева Е.Ю. Знакомство с разработкой мобильных приложений на платформе «1С: Предприятие 8». – М.: ООО «1С-Пабблишинг», 2022.
3. Султанова А.И., Шаронова А.А. 1С:Счетчик ворон. Мастер-класс по мобильной разработке в среде «1С: Предприятие» . – М.: ООО «1С-Пабблишинг», 2022.
4. Благин К. Телеграм бот 1С. [Электронный ресурс]. 2021 –URL: <https://blagin.ru/telegram-bot-1s/?ysclid=li78kyssfp771556384> (дата обращения: 28.05.2023).

УДК 004.428.4

Журавлев П.В.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ОНЛАЙН-ПЛАТФОРМЫ СОЦИАЛЬНОЙ СЕТИ МЕСТО ПРИ ПОМОЩИ REACT КОМПОНЕНТОВ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Данная статья представляет обзор React компонентов, которые широко используются при разработке программного обеспечения.

На данный момент редко кто из разработчиков использует чистый JavaScript (далее по тексту JS) для создания сайтов [1]. Его заменили библиотеки и фреймворки. Примером может послужить React – одна из наиболее используемых в этой сфере библиотек [2]. Она с успехом доказала свою эффективность в разработке пользовательских интерфейсов, что гарантировало ей широкое применение. Кроме того, React включает в себя Document Object Model (далее по тексту DOM) – имитационный метод представления структурного документа с помощью объектов. Специалист способен создавать динамические, интерактивные веб-приложения: любой элемент интерфейса можно обновить, не затрагивая оставшуюся страницу.

Библиотека React.js содержит наборы разнообразных готовых компонентов ответственных за решение многих сложных задач: общение в социальных сетях, взаимодействие человека с программой, управление состоянием приложения и другие. Тем самым она упрощает проектирование.

Стоит отметить следующие особенности библиотеки React:

1. Высокий уровень производительности за счёт использования «виртуальной» DOM;
2. Широкий спектр компонентов;
3. Применение условных выражений (информация на странице браузера отображается согласно условиям в JSX);

Что такое «виртуальная» DOM? В некотором смысле, это концепция, в которой «виртуальное» представление пользовательского интерфейса содержится в памяти и сравнивается с «реальной» DOM при помощи библиотеки ReactDOM. Указанный процесс имеет название – «согласование».

Описанный подход позволяет применить декларативный API-интерфейс React: пользователь сообщает React, в каком состоянии хочет видеть пользовательский интерфейс, и тот проверяет, соответствует ли DOM этому состоянию. Таким образом, достигается обработка событий, абстрагирование манипуляций с атрибутами, ручное обновление DOM,

которые в противном случае необходимо было бы использовать для создания вашего приложения.

Принимая во внимание, что «виртуальная» DOM — это скорее шаблон, чем конкретная технология, люди иногда утверждают, что это противоположные вещи. В React термин «виртуальная» DOM обычно ассоциируется с компонентами библиотеки. Однако, React также применяет внутренние объекты, называемые «волоконнами», для хранения вспомогательных данных о дереве элементов (рисунок 1).

Рассмотрим этот момент подробнее. Когда программист вставляет новые объекты, создается виртуальная модель, представленная в виде дерева, и узлом является каждый элемент в нем. Если состояние любого из них подвергается изменению, генерируется новое дерево, а затем происходит сравнение с предыдущим. По прохождению данного этапа, «виртуальная» DOM рассчитывает наиболее благоприятный из возможных способов внесения изменений в «реальную». Данный алгоритм обеспечивает самое малое количество операций с настоящей моделью. Отсюда скорость обновления оригинала будет увеличена.

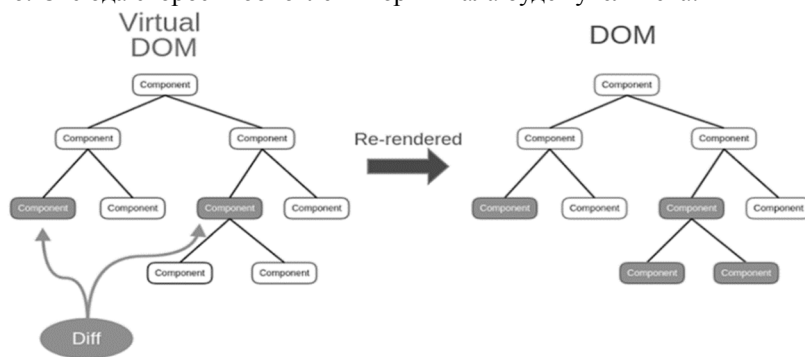


Рисунок 3 – Virtual Dom в сравнении.

Выбирая React для разработки, не стоит забывать, что открыт доступ к исходным кодам готовых объектов. Этот факт очень важен для людей со «стартапами», которые нуждаются в объективном результате, как можно быстрее при минимальных затратах денег, сил и времени.

React обладает гибкостью, потому что разработчик может выбирать, какие библиотеки использовать (например, для контроля над переходами между экранами и хранения данных), менять инструменты разработки, делать приложение полностью на React или частично (рисунок 2).

```

return (
  <article className="card">
    <img onClick={handleClick} className="card_picture" src={card.link} alt={card.name}></img>
    <button onClick={handleDeleteClick} className={cardDeleteButtonClassName}></button>
    <div className="card_container">
      <h2 className="card_description">{card.name}</h2>
      <div className="card_like-container">
        <button type="button" onClick={handleLikeClick} className={cardLikeButtonClassName}></button>
        <span className="card_like-count">{card.likes.length}</span>
      </div>
    </div>
  </article>
);
}

```

Рисунок 4 – Пример переиспользуемого компонента «Карточка» в социальной сети Mesto.

Именно при помощи данной библиотеки была разработана социальная сеть Mesto. Результат доступен по ссылке:

//URL: <https://social-network.mesto.nomoredomains.work/>

Подводя итог вышеизложенному, стоит отметить, что React зарекомендовывает себя по многим критериям: сокращение времени, затрачиваемого на работу, помощь в написании логичного структурированного и интуитивно понятного кода, переиспользование больших блоков. Кроме того, можно значительно снизить стоимость разработки, поддержки, обновления и отладки приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Введение в React:[Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/ruvds/blog/519902>;
2. Рейтинг frontend фреймворков:[Электронный ресурс] – Режим доступа: <https://stackdiary.com/front-end-frameworks>.

УДК 004.4

Бубнов С.А., Журавлева В.С.

АНАЛИЗ ТЕХНОЛОГИЙ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ГОЛОСОВ ПТИЦ В ЭПОХУ ГЛУБОКОГО ОБУЧЕНИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В работе рассматриваются существующие технологии автоматического распознавания вокализации птиц на основе глубокого обучения.

Способы быстрой и эффективной идентификации птиц по голосу очень актуальны для многих областей орнитологии и экологии. Мониторинговые исследования используются при изучении пищевых систем, состояния лесов и видового или популяционного состава птиц. Методы глубокого обучения (DL) не требуют отлова птиц и не отнимают

много времени по сравнению с используемыми ранее технологиями неавтоматического распознавания.

Технологию автоматического распознавания голосов птиц можно разделить на следующие три основных этапа:

1. предварительная обработка;
2. выделение признаков;
3. распознавание.

Этап предварительной обработки включает в себя предварительный акцент, шумоподавление и сегментацию и позволяет получить относительно чистые вокализации птиц из записей. Целью предварительного акцента является калибровка относительных амплитуд звуковых сигналов в разных частотных диапазонах. Ключ к шумоподавлению заключается в различении шума и голосов птиц. Следовательно, мы можем создавать различные модели шума и вокализаций птиц, а затем различать их. Модели глубокого обучения могут автоматически изучать свойства шума из обучающего набора. Широко используются такие из них, как глубокие автокодеры (АЕ) и глубокие нейронные сети (DNN). Для получения высокой эффективности шумоподавления методы на основе этих моделей требуют огромного и разнообразного набора данных о шумах. Перед распознаванием голосов необходимо сегментировать записи, что означает определение начала и конца каждой вокализации с помощью сегментации. Точность сегментации влияет на производительность последующего распознавания. Метод энергетического порога является одним из простейших и наиболее широко используемых методов сегментации [3].

Этап извлечения признаков – это процесс извлечения отличительных характеристик различных видов или отдельных особей, которые могут быть либо созданы вручную, либо автоматически. Методы выделения признаков сигналов вокализации птиц можно разделить на две категории:

Методы ручного извлечения вычисляют созданные вручную характеристики из звуковых сигналов.

Методы автоматического извлечения, которые изучают особенности непосредственно из звуковых сигналов с использованием моделей, таких как сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и автокодеры (АЕ) [2].

Например, глубокие нейронные сети со сверточной долговременной памятью (CLDNNs) извлекают частотно-временные признаки, используя сочетание временных и частотных сверточных слоев, и объединяют сети долговременной кратковременной памяти (LSTM) для извлечения последовательных признаков между соседними записями. Автокодеры (АЕ) - это нейронные сети, которые воспроизводят входные

данные на выходе. АЕ разделен на две части: кодер и декодер. Кодировщик преобразует входной звук в представление в скрытом пространстве. Декодер реконструирует входные данные из представления в скрытом пространстве для получения выходного звука. АЕ максимально приближает выходные данные к входным данным, указывая на то, что представление в скрытом пространстве содержит большую часть входной информации. Представление в скрытом пространстве можно рассматривать как изученный признак. Следовательно, мы можем извлекать особенности голосов птиц непосредственно из необработанного аудио, используя нейросетевой кодировщик временных рядов, такой как LSTM.

Этап распознавания включает классификацию или группировку особенностей голосов птиц по видам или особям. Существует два типа решений: одно основано на обучении с частичным контролем, в котором система сначала обучается с использованием образцов вокализации птиц для известных классов. Затем программа оценивает разницу между характеристиками новых образцов и характеристиками известных классов и считает их новыми классами, если эта разница превышает заданный порог. Второе решение использует неконтролируемые подходы кластеризации для разделения набора данных, содержащего неизвестные классы, на несколько разных. На практике, после определения классов с помощью вышеупомянутых двух подходов, количество классов, которые может идентифицировать модель, может быть дополнительно расширено путем проверки экспертом.

Модели, используемые для автоматического распознавания вокализации птиц с известным количеством классов:

Глубокая нейронная сеть (DNN). Несмотря на отсутствие впечатляющей производительности, это первый шаг к распознаванию вокализации птиц на основе DL.

Сверточная нейронная сеть (CNN). В настоящее время является предпочтительной моделью DL для классификации из-за ее локального подключения, распределения веса и операций объединения, которые могут значительно снизить сложность сети и параметры обучения.

Сверточные рекуррентные нейронные сети (CRNN). CRNN предлагается для захвата последовательной корреляции между кратковременными записями путем добавления RNN к CNN. Модель CRNN в настоящее время часто используется с хорошими результатами. Однако она может потребовать больше вычислительных ресурсов для обучения [1].

Таким образом, можно сказать, что хотя автоматическое распознавание вокализации птиц на основе DL в последнее время достигло успехов в отношении конкретных видов, для достижения надежного и общего распознавания птиц в дикой природе требуются

более надежные подходы к шумоподавлению, большие общедоступные наборы данных и более мощные возможности обобщения для извлечения и распознавания признаков. Расширение и применение комплексного подхода к мониторингу разнообразия других животных может способствовать развитию интеллекта в области сохранения биоразнообразия.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ганчев Т.Д., O. Jahn, M.I. Marques, Дж.М. de Figueiredo. Автоматическое акустическое обнаружение *vanellus chilensis lampronotus*. // *Expert Syst. Appl.* – 2015. – №42 – С.15-16.
2. A.M. Searfoss, J.C. Pino, N. Creanza. Open-source software for semi-automated segmentation and analysis of birdsong and other natural sounds. // *Methods Ecol. Evol.* – 2020. – №11 – С.524-531.
3. J. Xie, J. Yang, Z. Xing, Z. Zhang, X. Chen. Bird species recognition method based on multi-feature fusion. // *J. Appl. Acoust.* – 2020. – №39 – С.199-206.

УДК 004.4

Кабочкин А.Н., Жулева С.Ю.

РЕАЛИЗАЦИЯ ПАРСЕРА НА ОСНОВЕ БИБЛИОТЕКИ HTMLAGILITYPACK: ПРИМЕНЕНИЕ СИНТАКСИЧЕСКОГО АНАЛИЗА HTML-ДОКУМЕНТОВ В КОНТЕКСТЕ ИЗВЛЕЧЕНИЯ ДАННЫХ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Работа описывает функциональные возможности библиотеки, включая навигацию по DOM- структуре, поиск элементов и извлечение содержимого.

С внедрением интернета во все сферы нашей жизни количество доступной веб-информации растёт в геометрической прогрессии. Это открывает огромные возможности для сбора и анализа данных с веб-страниц. Однако, чтобы извлечь ценную информацию из HTML-документов, требуется специализированный инструментарий. В этой статье мы рассмотрим реализацию парсера с использованием библиотеки *HtmlAgilityPack* и роль синтаксического анализа HTML-документов в контексте извлечения данных [1].

Парсинг HTML-документов представляет собой сложную задачу из-за разнообразия структур и форматов, с которыми они могут быть написаны. Процесс извлечения данных из HTML-страницы требует обработки DOM-структуры (Document Object Model), построенной на основе HTML-кода. Однако, напрямую работать с DOM-структурой может быть трудоемким и сложным. В этом контексте библиотека

HtmlAgilityPack представляет собой мощный инструмент для парсинга и анализа HTML-документов [4].

HtmlAgilityPack является .NET библиотекой, специально разработанной для работы с HTML-документами. Она предоставляет удобные и мощные методы для синтаксического анализа HTML-кода и извлечения данных из него. Благодаря своей гибкости и простоте использования, HtmlAgilityPack стал популярным выбором для разработчиков, занимающихся веб-скрапингом, автоматическим сбором данных и анализом информации в Интернете [1].

Одним из ключевых преимуществ HtmlAgilityPack является его способность работать с HTML-кодом, который не является полностью совместимым с XML-стандартом. Библиотека может обрабатывать различные HTML-структуры и автоматически выполнять исправления ошибок, что делает ее надежным и гибким инструментом для парсинга HTML-документов [4].

Синтаксический анализ HTML-документов играет важную роль в процессе извлечения данных из веб-страниц. HtmlAgilityPack предоставляет широкий набор методов для навигации по DOM-структуре, поиска и выбора элементов, а также получения содержимого и атрибутов. Это позволяет разработчикам легко определить требуемые HTML-элементы и извлечь нужные данные для дальнейшей обработки или анализа [3].

HtmlAgilityPack предлагает широкий набор функций и методов для работы с HTML-документами, основными достоинствами которой являются [1].

- Загрузка и парсинг HTML: HtmlAgilityPack позволяющий загружать HTML-страницу из различных источников, включая файлы, строки или веб-запросы. Библиотека автоматически выполняет синтаксический анализ HTML и создает DOM-структуру для дальнейшей обработки [3].

- Навигация по DOM-структуре: HtmlAgilityPack осуществляет перемещение по DOM-структуре HTML-документа, переходя от одного элемента к другому. Методы, `SelectSingleNode` и `SelectNodes`, позволяют выбирать элементы на основе селекторов или XPath-выражений [2].

- Извлечение данных: HtmlAgilityPack предоставляет простые методы для получения содержимого элементов, таких как текст, HTML-код, атрибуты и другие свойства. Использование этих методов необходимо для извлечения данных с веб-страницы [1].

- Манипуляции с DOM: библиотека позволяет вносить изменения в DOM-структуру HTML-документа, добавлять, удалять или изменять элементы и их атрибуты при модифицировании HTML-кода перед извлечением данных.

HtmlAgilityPack широко применяется в области веб-скрапинга [2], автоматического сбора данных и анализа информации в Интернете. Его гибкость и мощные возможности позволяют разработчикам эффективно извлекать ценные данные с веб-страниц, такие как новости, цены товаров, рейтинги, контактные данные и многое другое.

Одно из распространённых применений HtmlAgilityPack - это создание веб-скрейперов, которые автоматически собирают данные с различных веб-ресурсов. Благодаря своей гибкости, библиотека позволяет обрабатывать и анализировать сложные HTML-структуры и извлекать только необходимые данные.

```
var doc = new HtmlDocument();
doc.LoadHtml(html);

// Извлечение данных
var products = doc.DocumentNode.SelectNodes("//*[@class='product']");
if (products != null)
{
    foreach (var product in products)
    {
        var name = product.SelectSingleNode("./h2").InnerText;
        var price = product.SelectSingleNode("./span[@class='price']").InnerText;

        Console.WriteLine($"Название: {name}");
        Console.WriteLine($"Цена: {price}");
        Console.WriteLine();
    }
}
```

Рисунок 1 - Пример реализации парсера

В данном примере (рисунок 1) мы загружаем HTML-страницу в объект `HtmlDocument`, а затем используем методы `SelectNodes` и `SelectSingleNode` для выбора соответствующих элементов и извлечения необходимых данных. Затем мы выводим названия и цены каждого товара на консоль.

Вывод. Реализация парсера на основе библиотеки `HtmlAgilityPack` предоставляет разработчикам мощный инструмент для синтаксического анализа HTML-документов и извлечения данных. `HtmlAgilityPack` обладает широким набором функций, позволяющих эффективно работать с DOM-структурой и получать необходимую информацию с веб-страниц.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. А.А. Золотов "Парсинг веб-страниц. Универсальные методы и инструменты", - 2019. -С. - 90.
2. Рогов А.И. "Программирование парсеров и обработка веб-страниц" // ДМК Пресс, 2015. - 320 с.
3. Кзлов А.М. "Разработка веб-парсеров на C#"//Питер, 2016. - 384 с.
4. "НАР - Documentations" // Официальная документация разработчиков, - 2019. -Раздел "Parser" .

УДК 519.876.5

Камордин А.А., Крошилин А.В.**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА В ОБУЧЕНИИ НЕЧЕТКИХ
КОГНИТИВНЫХ КАРТ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Использование нечеткой когнитивной карты подразумевает установку большого количества связей между факторами. Данная задача решается в момент обучения когнитивной нечеткой карты.

Когнитивные карты позволяют построить модель социальной экономической системы, что позволяет качественно оценить её, отследить наиболее слабые места и определить силу взаимодействия между её компонентами [1].

Когнитивные нечеткие карты строят на основе мнений экспертов. При этом, субъективные взгляды экспертов, различия в мнениях и человеческий фактор могут повлиять на качество модели. Поэтому задача по коррективке, называемая обучением когнитивной карты, является актуальной[2].

Использование генетического алгоритма является одним из подходов, к наиболее лучшей и более качественной расстановке весов между факторами в когнитивной нечеткой модели.

Генетический алгоритм – это эволюционный алгоритм, применяемый при решении задач оптимизации и моделирования путем подбора целевых параметров способом, который является аналогичным естественному отбору в природе и состоит из следующих этапов:

1. Выделение начальной популяции;
2. Скрещивание и мутация;
3. Селекция;
4. Формирование нового поколения.

Этапы повторяются циклично, до достижения необходимых результатов.

При использовании генетического алгоритма, задача обучения сводится к определению приращений значений факторов, возникающих в карте при внесении в неё изначального вектора приращений значений факторов [3, 4].

При обучении когнитивной карты, в качестве хромосомы используется набор весов. Алгоритм, касательно когнитивных карт, будет выглядеть следующим образом:

1. Для ненулевых значений весов, связывающих факторы, будут определены новые ненулевые значения (исходные значения весов задаются экспертом). Таким образом формируется изначальная популяция;

2. Определяется функция приспособленности для каждой хромосомы. Она является обратной к функции минимизации ошибки, определяемой по формуле (1):

$$E(w) = \frac{1}{2} \sum_{i=1}^k \sum_{l=1}^n (o(i) - y(i))^2 \quad (1)$$

3. Определяется пул родителей по методу “рулетки”. Он состоит в том, что выбирается случайная особь, но шанс её выбора прямо зависит от величины приспособленности.

4. Составляется пул родителей из “элитных особей” (это те особи, которые показали наилучшее значение функции приспособленности).

5. Происходит скрещивание хромосом с заранее заданной вероятностью. В результате будут получены новые хромосомы, а также некоторое их количество не измененных.

6. Из полученных хромосом формируется новая популяция, чей размер строго совпадает с размером предыдущей популяции.

7. С заранее заданной вероятностью происходит мутация, при которой будут получены новые случайные значения. Часть хромосом также могут остаться без изменений.

8. Определяются параметры для поколения: элитные особи, значения приспособленности особей, среднее значение приспособленности популяции.

9. Если значение приспособленности будет достигнуто целевого показателя (установлен заранее), то обучение можно считать завершённым. Если показатель не будет достигнут, то данная популяция вернется к этапу 7, и будет подвергнута повторной мутации.

10. Результирующая хромосома раскладывается на веса, для последующей установки в когнитивную нечеткую карту.

В результате обучения когнитивной нечеткой карты, будут изменены её коэффициенты, установленные экспертной группой. Это позволит сделать модель более объективной, избавить от недостатков, которые привносит человек, а именно объективному мнению человека о конкретной экономической системе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Papageorgiou E. Review study on Fuzzy Cognitive Maps and their applications during the last decade. / E. Papageorgiou // Paper presented at the IEEE International Conference on Fuzzy Systems; – Taipei. Taiwan: the IEEE International Conference on Fuzzy Systems, 2011.

2. Аверкин А.Н., Паринков А.А., ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ОБУЧЕНИЯ НЕЧЕТКИХ КОГНИТИВНЫХ КАРТ // Научные труды Вольного экономического общества России, 2010.

3. Силов В.Б. Принятие стратегических решений в нечеткой обстановке.-М.:ИНПРО-РЕС, 1995.

4. Кулинич А.А. Разработка принципов и методов построения программных систем поддержки принятия решений в слабо структурированных ситуациях на основе моделирования знаний эксперта. Автореферат на соискание ученой степени кандидата технических наук.- М:РАН, 2003.

УДК 007:681.512.2

Каширин И.Ю.

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ML-МОДЕЛЕЙ С ПОМОЩЬЮ АВТОМАТИЗИРОВАННОГО ИСПЫТАТЕЛЬНОГО СТЕНДА

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Рассмотрена технология проектирования моделей машинного обучения на основе разработки инструментального испытательного стенда, содержащего основные программы предварительной обработки данных и последующего обучения моделей.

Модели машинного обучения (ML-модели) применяются в качестве главного метода проектирования аналитического программного инструментария в сфере интеллектуальной обработки данных (Data Mining) [1]. Упрощенная схема жизненного цикла таких моделей представлена на рисунке 1.

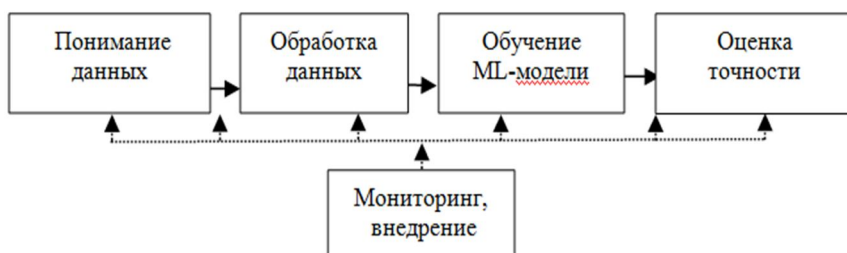


Рисунок 1 – Схема жизненного цикла моделей машинного обучения

Целью работ было выполнение научного исследования в предметной области анализа параметров работы пользователей с предметными Web-сайтами в целях определения вероятных предпосылок их желания использовать эти Web-сайты в дальнейшем. В качестве метода реализации была поставлена задача создания автоматизированного испытательного стенда в виде библиотеки программ, адекватных и достаточных для интеллектуального исследования больших данных указанной предметной области на основе методов Data Mining в форме алгоритмов

статистического анализа и нейронных сетей.

Для реализации поставленной задачи была произведена аналитика входного набора данных, экспертная оценка и обработка признаков входного набора. В частности, выполнены следующие этапы предварительной обработки входного набора данных и последующего проектирования моделей:

- кодирование данных,
- исключение некорректных значений,
- определение и устранение информационных выбросов,
- поиск неявных дубликатов,
- обработка пропусков (Nan-данных),
- нормализация,
- разделение входного набора данных на подвыборки,
- автоматический тюнинг гиперпараметров моделей,
- контроль качества обучения.

В соответствии с концепцией использования семантики признаков входных наборов данных [2,3] можно выделить основную таксономию причинно-следственных связей между классами признаков. В соответствии с этой концепцией необходимо сформулировать вопросы к экспертам о возможных причинно-следственных связях в предметной области.

Если рассматривать в качестве основной цели разрабатываемых ML-моделей увеличение посещаемости клиентами Web-ресурсов, главными влияющими факторами следует признать:

- содержательную часть Web-сайта,
- особенности клиента (пользователя),
- статистические данные о предшествующих действиях пользователя в рамках каждого из Web-сайтов.

С учетом этих факторов можно сформулировать основные задачи Data Mining в предметной области анализа посещаемости Web-ресурсов:

- как можно больше узнать информации о каждом из клиентов,
- как можно больше узнать информации о каждом из сайтов,
- как можно полнее и выбрать сбалансированную информацию о предыдущем поведении каждого из клиентов.

Для продвижения предложенных перспективных проектов в качестве рабочего примера предлагается следующая тематическая таксономия, которая может выглядеть следующим образом (рисунок 2).

Результаты работы были положены в основу итогового программного комплекса, который позиционирован как автоматизированный испытательный стенд для интеллектуального анализа больших данных.

В проекте были апробированы методы:

- опорных векторов,
- градиентного спуска,
- случайного леса,
- логистической регрессии,
- К-ближайших соседей,
- нейронных сетей,
- градиентный бустинг деревьев,
- XBoost и AdaBoost методы,
- смешанный ансамбль алгоритмов.

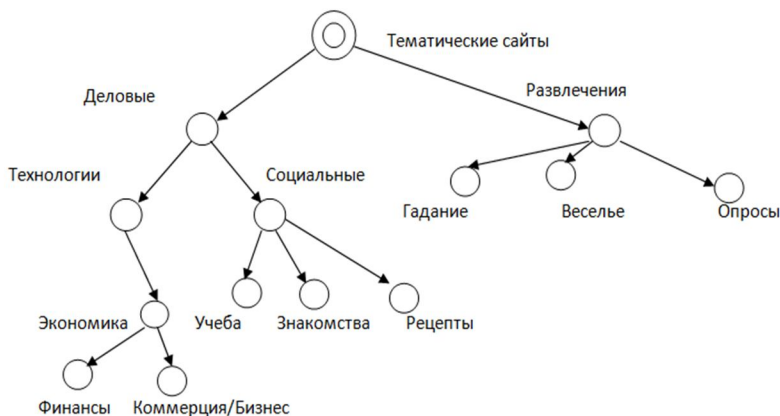


Рисунок 2 – Родовидовая тематическая таксономия Web-сайтов

Разработки представляют собой отдельные IT-программы на языке Python v.3.00, выполненные в облачной инструментальной среде Google Colab v. 2023.3.10 [4].

Как показали экспериментальные результаты, наибольшей эффективностью располагают XBoost и AdaBoost методы, а также другие ансамбли ML-моделей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каширин И.Ю. Применение теории иерархических чисел в проектировании ICF-таксономии для оптимизации нейронных сетей. Вестник Рязанского государственного радиотехнического университета. Рязань, 2022. № 80 – с.36-43.
2. Сатыбалдина Д.Ж., Овечкин Г.В., Калымова К.А. Система распознавания статических жестов рук с использованием камеры глубины. Вестник рязанского государственного радиотехнического университета. Рязань, 2020. № 72 с. 93-105.

3. P. Chapman, J. Clinton, R. Kerber and others. CRISP-DM 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, 2000.

4. Каширин И.Ю. Data Mining с использованием иерархических чисел в ретроспективной диагностике. Вестник Рязанского государственного радиотехнического университета. Рязань, 2022. № 79. – С.118-126.

УДК 007:681.512.2

Каширин И.Ю.

КАЛИБРОВКА ML-МОДЕЛЕЙ ДЛЯ ИДЕНТИФИКАЦИИ ФЕЙК-НОВОСТЕЙ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Рассмотрена технология калибровки моделей
машинного обучения для программного
инструментария, идентифицирующего достоверность
материалов новостных электронных лент.
Калибровка предполагает определение вероятности
принадлежности материала к фейк-новостям.

Модели машинного обучения с эффективностью используются не только в задачах прогнозирования, но и в задачах идентификации объектов предметной области или сложных ситуативных структур [1,2].

В частности, такие модели используют для исследования естественно-языковых текстов, например, с целью определения эмоциональной окраски текста или отнесения электронных новостей к классам различной степени достоверности. Существует множество зарубежных разработок в области автоматического отнесения новости к фейку или истине [3]. Такие идентифицирующие системы приобретают все большую популярность, особенно в молодежной среде.

В то же время зарубежные автоматизированные системы построены на аксиоматике множества неоспоримых исторических фактов, и сделанные ими заключения об отнесении новости к фейку основаны на этих первоначальных фактах. Такими фактами, например, могут быть «Победителями во второй мировой войне были Англия и Америка» или «Вероятнее всего, газопровод «Северный поток» был уничтожен Россией». Если исходить из подобных «фактов» как из априори истинных аксиом, любые электронные новостные материалы будут оцениваться соответствующим образом.

Изложенные соображения свидетельствуют о необходимости создания отечественных программных продуктов, трактующих исторические факты, исходя из документально подтвержденных исторических документов и информационных материалов.

Предлагаемая в настоящей статье технология идентификации достоверности новостей основана на изучении особенностей существующих в информационной среде реалий и использует механизм калибровки моделей машинного обучения для вычисления вероятности отнесения новостного материала к недостоверным сведениям.

Калибровка предполагает уточнение бинарного классификационного признака («ложь», «истина») более точным числом, представляющим собой вероятность признания факта лживым или достоверным. Существующая библиотека scikit-learn 1.2.2 дает возможность использовать множество готовых методов калибровки, например OneVsOneClassifier:

```
from sklearn import datasets
from sklearn.multiclass import OneVsOneClassifier
from sklearn.svm import LinearSVC
X, y = datasets.load_iris(return_X_y=True)
OneVsOneClassifier(LinearSVC(random_state=0)).
    fit(X, y).predict(X)
```

В предлагаемом здесь подходе используются лингвистические правила, основанные на наборах именных групп как фрагментов синтограмм, применяемых на этапе парсинга для получения дополнительных коэффициентов достоверности (таблица 1).

Более полный фрагмент правил приведен в [4]. Для применения правил используется база знаний, содержащая продукции следующего вида:

```
<Синтограмма> <Новостной факт X1><Новостной факт X2><...>
<Новостной факт XN> <Электронное издание><Автор> =>
KBChange(<X1,K>,<X2,K>,<...>,<XN,K>,
<Электронное издание,K>,<Автор,K>),
```

где <Синтограмма> – применение схемы синтаграммы при парсинге текста новости, <Новостной факт XI> – текст изложенного в новости факта, <Электронное издание> и <Автор> – наименование электронного новостного ресурса и имя автора соответственно. KBChange – метод актуализации базы знаний исправлением рейтинга новости, издания и автора коэффициентом K из таблицы 1.

Для калибровки модели обучения может быть применена дополнительная конструкция:

```
KBChange(LinearSVC(random_state=0)).fit(X, y).predict(X).
```

В настоящее время программный инструментарий, проектируемый по предложенной технологии, находится в разработке.

Таблица 1 – Фрагмент набора именных групп индикаторов лжи

№	Признаки достоверной/недостоверной новости	Формула синтограммы	Базовый коэффициент достоверности
1	Автор избегает конкретных цифр	«было убито много» X, «большое количество жертв» X	-0,2
2	Лицо, заинтересованное в фейке, но опровергающее его, может говорить правду.	«мы были неправы», «приносим извинения за X»	+0,1
3	Лицо, узнавшее о важной информации, делает вид, что не знает этой информации.	«нам неизвестен факт» X, X «не могло происходить»	-0,7
4	Косвенная ложь	«представитель X сообщил, что», «X говорит, что»	-0,5

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каширин И.Ю. Применение теории иерархических чисел в проектировании ICF-таксономии для оптимизации нейронных сетей. Вестник Рязанского государственного радиотехнического университета. Рязань, 2022. № 80 – с.36-43.

2. Сатыбалдина Д.Ж., Овечкин Г.В., Калымова К.А. Система распознавания статических жестов рук с использованием камеры глубины. Вестник рязанского государственного радиотехнического университета. Рязань, 2020. № 72 с. 93-105.

3. Maria D. Molina, S. Shyam Sundar, Thai Le, and Dong-won Lee. “Fake news” is not simply false information: A concept explication and taxonomy of online content. American Behavioral Scientist, 2021, 65(2):180–212.

4. Каширин И.Ю. идентификация достоверности новостей с помощью моделей машинного обучения Вестник Рязанского государственного радиотехнического университета. Рязань, 2023. № 83. – С.36-47.

УДК 517.518.85

Клюкин Д.Д.

ИССЛЕДОВАНИЕ ИНТЕРПОЛИРОВАНИЯ ФУНКЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Задачи анализа часто требуют прибегать к аппроксимации, интерполирование же один из вариантов аппроксимации в котором функция жёстко привязана к точкам. В этой статье разбираются формулы интерполирования Лагранжа и Ньютона.

Численные методы вычислительной математики часто прибегают к замене функции, используемой при построении задачи, схожими и более простым для расчётов.

Интерполяция функций – один из наиболее простых и активно используемых методов сближения функций.

В настоящее время это понятие имеет весьма широкий смысл, однако суть остаётся неизменной: основываясь на промежутке $[a, b]$, где задана функция $y = f(x)$ или множество её значений $f(x_0), f(x_1), \dots, f(x_n)$, найти функцию $P(x) \approx f(x)$, такую что бы значения $P(x_i) = f(x_i)$.

Точки $x_0, x_1, \dots, x_n$ используемые при интерполировании, называются узлами интерполирования, а $P(x)$ – интерполирующей функцией. Важно помнить, что $P(x)$ может относиться ко многим классам, в зависимости от функции $f(x)$. Например, если $f(x)$ – циклическая, то $P(x)$ будет относиться к классу тригонометрических функций, а если $f(x)$ становится бесконечной в одно или нескольких заданных точка, $P(x)$ желательно представить в качестве рациональной функции.

Интерполяционная формула Лагранжа. Предположим, что на известном нам отрезке $[a, b]$ задано множество точек $x_k, k = 0, 1, \dots, n$ (узлы интерполирования), в которых известно значение, принимаемое интерполируемой функцией $f(x)$. Вопрос интерполяции функции с многочленом состоит найти n порядкового

$$L_n(x) = a_0 + a_1x + \dots + a_nx^n \quad (1.1)$$

многочлена, значение которого в точках узла равно значению функции $f(x)$. Для определения коэффициентов $a_0, a_1, \dots, a_n$ напрашивается составить систему линейных уравнений:

$$a_0 + a_1x_1 + \dots + a_nx_n = f(x_i), i = 0, 1, 2, \dots, n \quad (1.2)$$

Очевидно, что детерминант будет принимать отличные от нуля значения лишь тогда, когда точки $x_i, i = 0, 1, 2, \dots, n$ различны.

Получим что многочлен $L_n(x)$, удовлетворяющий условию должен иметь следующий вид:

$$L_n(x_i) = f(x_i), i = 0, 1, 2, \dots, n \quad (1.3)$$

Многочлен такого вида, называется интерполяционным многочленами. С помощью интерполяционной формулы Лагранжа, можно записать многочлен $L_n(x)$ в виде линейной комбинации точек узлов интерполяции функции интерполируемой функции и значений её интерполяции:

$$L_n(x) = \sum_{k=0}^n C_k(x) f(x_k) \quad (1.4)$$

Теперь найдём явное представление коэффициентов $C_k(x)$. Если вспомнить равенство (1.3) и условия интерполирования, то обнаружится, что это отношение выполняется при следующем условии:

$$C_k(x_i) = \begin{cases} 0, & \text{ага } i = k \\ 1, & \text{ага } i = k, i = 0, 1, \dots, n \end{cases}$$

Из этого следует, что на заданном отрезке $[a, b]$ каждая функция $C_k(x)$, $k = 0, 1, \dots, n$ имеет по крайней мере n ноль. $L_n(x)$ – n -порядковый многочлен, значит вполне закономерно искать коэффициенты $C_k(x)$ в виде n -порядкового многочлена

$$C_k(x) = \lambda_k (x - x_0)(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)$$

Из условия

$$C_k(x_k) = 1$$

Получаем

$$\lambda_k^{-1} = (x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)$$

Теперь у нас есть всё для выражения коэффициентов интерполяции многочлена (1.4), получим следующую формулу:

$$C_k(x) = \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)}$$

Подводя итог, мы можем получить интерполяционный многочлен Лагранжа в следующем виде:

$$L_n(x) = \sum_{k=0}^n \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)} f(x_k) \quad (1.5)$$

Интерполяционная формула Ньютона. Формула ньютона, в чём-то похож на формулу Лагранжа и так же использует интерполяционный многочлен. Однако в этом случае он вычисляется через один из узлов интерполяции и разностное деление функции $f(x)$. И так попробуем составить уравнение используя такие данные данные:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x - x_0)^2}{2f} f''(x_0) + \dots$$

Получившееся выражение состоит из аналогов формулы Тейлора. Приведём данные о разностных делениях, предположим что значения $f(x)$ известны в точках узлов интерполирования $x_k \in [a, b]$, $k = 0, 1, \dots, n$. Получим соотношение:

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_i - x_j}, i, j = 0, 1, \dots, n, i \neq j$$

Оно называется разностным делением первого порядка. Теперь мы можем сконструировать разностные деления первого порядка для соседних точек и на основе их получить разностные деления второго порядка.

$$f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0}$$

$$f(x_1, x_2, x_3) = \frac{f(x_2, x_3) - f(x_1, x_2)}{x_3 - x_1}$$

...

$$f(x_{n-2}, x_{n-1}, x_n) = \frac{f(x_{n-1}, x_n) - f(x_{n-2}, x_{n-1})}{x_n - x_{n-2}}$$

Аналогичным образом можно составить и разностные деления более высоких порядков. Если известны разностные деления k -того порядка, разностные деления $k+1$ порядка можно получить следующим образом:

$$f(x_j, x_{j+1}, \dots, x_{j+k+1}) = \frac{f(x_{j+1}, \dots, x_{j+k+1}) - f(x_j, \dots, x_{j+k})}{x_{j+k+1} - x_j}$$

Теперь выразим разностные деления k -того порядка через значения в точках узлов интерполирования функции $f(x)$. Получим следующее выражение:

$$f(x_j, x_{j+1}, \dots, x_{j+k}) = \sum_{i=j}^{j+k} \frac{f(x_i)}{\prod_{\substack{l=j \\ l \neq i}}^{j+k} (x_i - x_l)}$$

Из этого выражения можем получить многочлен:

$$P_n(x) = f(x_0) + (x - x_0)f(x_0, x_1) + (x - x_0)(x - x_1)f(x_0, x_1, x_2) + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})f(x_0, x_1, x_{n-1}) \quad (1.7)$$

Этот многочлен и называется интерполяционным многочленом Ньютона.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Исроилов М.И. «Численные методы». Т:Учитель, 2000.С 258-277.
2. Самарский А.А. «Введение в численные методы». М: Наука, 1987.
3. Имомова Ш.М., Фаттоева Г.Ш. Методика преподавания темы квадратных уравнений // Ученый XXI века. № 1 (48), 2019. С. 29-30.

УДК 004.056.52

Краев Д.В.

РАЗРАБОТКА ПОЛИТИКИ ВЗАИМОДЕЙСТВИЯ МОДУЛЕЙ ЗАГРУЗКИ ДОКУМЕНТОВ ПРИЛОЖЕНИЯ ЗАПУСКАЕМОГО В КИБЕРИММУННОЙ СРЕДЕ

Российский Технологический Университет МИРЭА, Москва

В данной статье представлен пример реализации политики взаимодействия модулей загрузки документов, запускаемого в кибериммунной среде. Поскольку актуальность подобного направления в сфере информационной безопасности растет, уровень умений специалистов ИБ по описанию подобных компонентов системы также обязан повышаться.

Актуальность данной работы заключается в том, что в современных компаниях остро стоит вопрос безопасности документов, в особенности от их утечки в открытый доступ. Для этого компании используют разные программные решения, и одним из них может быть специальная политика безопасности.

Кибериммунитет, как направление информационной безопасности, набирает обширные обороты. В России разработкой подобной системы занимается Лаборатория Касперского, успешно развивая данное направление с 2002 года. В других странах к реализации такой информационной безопасности уже успел присоединиться, например, google. Направление позволит защищать не только привычные нам телефоны/ПК от атак, но и так называемые IOT (internet of things) устройства.

Основными подходами к реализации кибериммунитета являются следующие особенности систем:

Запрет по умолчанию.

Secure-by-design.

Изоляция компонентов системы друг от друга.

Запрет по умолчанию означает, что любое действие в системе, если оно не разрешено, будет запрещено. Secure-by-design – это подход к реализации

системы, когда она изначально проектируется с учетом реализации информационной безопасности. Изоляция – особенность системы, при которой свойства безопасности отделены от компонентов системы.

Таким образом, для разработки политики взаимодействия модулей приложения запускаемого в кибериммунной среде нам необходимо понимать, что данная политика должна быть описана следующим образом:

1) Все компоненты системы, участвующей в работе приложения должны быть описаны в политике безопасности, иначе приложение не

сможет работать нормально (не описанный компонент не получит доступа к системе, из-за запрета по умолчанию).

2) Политика безопасности должна храниться отдельно от компонентов системы, чтобы удовлетворить требованию изоляции в кибериммунной среде.

3) Спроектировать политику безопасности также необходимо так, чтобы она корректно описывала взаимодействие модулей и не хранила в самой себе ошибки. Это не столько относится к *secure-by-design*, сколько к самой сути политики безопасности – четкое и правильное описание правил взаимодействия компонентов в системе.

Для примера возьмем политику безопасности для приложения, скачивающего файлы из хранилища по запросу авторизованных пользователей. Техническое задание для такой политики безопасности в рамках статьи будет короткое – в компании есть роль «менеджер», сотрудники с такой ролью могут скачивать файлы из хранилища только согласно своему уровню доступа (1- имеют доступ, 2 – не имеют), но видеть их могут все.

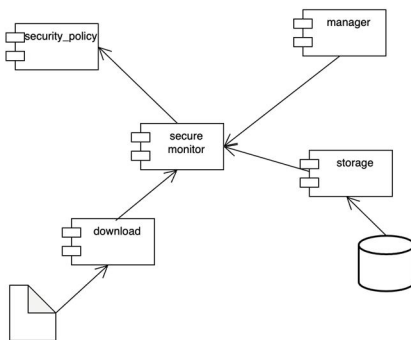


Рисунок 1 – Диаграмма компонентов.

Ниже описана функция-пример реализации подобной политики безопасности. Функция описана на языке python, используется компонентом *secure monitor*.

На вход она получает «событие» (массив), в котором содержится информация, на основании которой функция определит значение параметра *authorized*, как раз и отвечающего за разрешение/запрет запрашиваемого действия. Решение проводится на основании данных из массива событий (*details*) и того, удовлетворяют они условиям политики безопасности, либо нет:

Source – источник события, компонент системы, запросивший действие.

Destination – модуль, к которому обратился источник.

Operation – функция в модуле destination.

Access_level – уровень доступа сотрудника, запросившего действие.

По умолчанию параметр authorized принимает значение false, что означает, что любое взаимодействие, не описанное в политике безопасности приведет к отказу в выполнении запрашиваемого действия. В случае, если в поступившем в функцию событии содержатся необходимые параметры и они имеют в себе ожидаемые значения, параметр примет значение true и действие будет разрешено.

Соответственно, на выход функция вернет значение authorized для дальнейшей работы системы.

```
def security_policy(details):
    authorized = False # запрет по умолчанию
    source = details['source']
    destination = details['destination']
    operation = details['operation']
    access_level = details['access_level']
    if source == 'manager' and destination == 'download' \
        and operation == 'download_file' \
        and access_level == '1':
        authorized = True
    if source == 'download' and destination == 'manager' \
        and operation == 'download_done':
        authorized = True
    if source == 'manager' and destination == 'storage' \
        and operation == 'view_files':
        authorized = True
    if source == 'storage' and destination == 'manager' \
        and operation == 'send_files_list':
        authorized = True
    return authorized
```

На примере данной функции мы рассмотрели пример реализации политики безопасности модулей загрузки документов. В функции отражены все компоненты системы (менеджер, хранилище, компонент который скачивает файлы), уровень доступа менеджера, а также выполняемые операции.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кибериммунитет [Электронный ресурс]. – URL: <https://os.kaspersky.ru/technologies/cyber-immunity/> (дата обращения: 16.05.2023)
2. Видеоуроки Kaspersky [Электронный ресурс] – URL: <https://os.kaspersky.ru/resources/video/> (дата обращения: 16.05.2023)

3. Google Open Source Blog, KataOS [Электронный ресурс] – URL: <https://opensource.googleblog.com/2022/10/announcing-kataos-andsparrow.html> (дата обращения: 16.05.2023)

4. Secure update проект [Электронный ресурс] – URL: <https://github.com/sergey-sobolev/secureupdate/blob/main/docs/report/report.md> (дата обращения: 16.05.2023)

УДК 519.65

Копылов И.Ю., Лапкин М.И., Лягушева М.А., Овечкин Г.В., Хлебников Н.М.

ЧИСЛЕННОЕ ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

В статье проведено исследование и анализ различных алгоритмов интерполяции для выявления их преимуществ и недостатков по определённым метрикам. В результате исследования было выявлено, что каждый метод имеет свои преимущества и недостатки, и выбор конкретного алгоритма зависит от конкретной задачи.

Изучение и оценка алгоритмов интерполяции являются важными задачами в области математики, компьютерных наук и инженерии. Эти алгоритмы используются для построения гладких кривых и поверхностей на основе набора дискретных данных, что является необходимым для решения многих задач, таких как моделирование, графика, обработка изображений и видео [1].

Оценка и сравнительный анализ различных алгоритмов интерполяции также поможет улучшить экономическую эффективность разработки и использования соответствующих программных продуктов, что будет полезно для широкого круга потребителей и предприятий. Поэтому, исследование данной темы имеет огромную практическую значимость и может привести к новым открытиям и развитию области компьютерной графики и математики в целом [4].

Было проведено исследование следующих алгоритмов интерполяции: метод Лагранжа, метод Стирлинга, метод Бесселя, метод Ньютона. При работе над статьей использовались методы вычислительной математики и математической статистики, а также теория планирования экспериментов.

Интерполяция Лагранжа является одним из наиболее распространенных методов интерполяции, который используется для аппроксимации функций. Этот метод был разработан математиком Жозефом Лагранжем в XVIII веке и с тех пор нашел широкое применение в различных областях науки и техники. Основным преимуществом интерполяции Лагранжа является то, что он является простым и понятным

методом, который может быть легко реализован на практике. Данный метод основывается на использовании полинома, который проходит через заданные точки и позволяет аппроксимировать функцию с высокой точностью [2]. Тем не менее, интерполяция Лагранжа может быть менее точной, чем другие методы интерполяции, такие как интерполяция Бесселя.

Интерполяционный метод Стирлинга является одним из наиболее эффективных методов интерполяции. Этот метод был разработан математиком Джеймсом Стирлингом в XVIII. Основным преимуществом интерполяционного метода Стирлинга является то, что он обеспечивает высокую точность аппроксимации функций. Данный метод основывается также на использовании полинома, который проходит через заданные точки и позволяет аппроксимировать функцию с высокой точностью даже при использовании большого числа узлов интерполяции. Кроме того, интерполяционный метод Стирлинга может быть менее эффективным, чем метод интерполяции Бесселя.

Интерполяционный метод Ньютона - это еще один из наиболее распространенных методов интерполяции, разработанный английским математиком Исааком Ньютоном в XVII веке. Основным преимуществом интерполяционного метода Ньютона является его простота и удобство использования. Данный метод основывается на использовании разделенных разностей, которые позволяют вычислить коэффициенты полинома интерполяции. Это делает данный метод особенно полезным для решения задач, где требуется быстрое и эффективное решение. Кроме того, интерполяционный метод Ньютона может быть менее эффективным, чем другие методы интерполяции, такие как интерполяция Лагранжа.

Однако, методы интерполяции Лагранжа, Стирлинга и Ньютона также имеют свои недостатки. Один из них заключается в том, что использование данных методов может приводить к появлению феномена Рунге [3], который проявляется в виде осцилляций и неустойчивости при использовании большого числа узлов интерполяции.

С другой стороны, преимуществом данных интерполяций является то, что они могут быть использованы для аппроксимации функций любой формы и сложности. Это делает данные методы особенно полезным для решения задач в области математики, физики, химии и других наук.

Интерполяция Бесселя также является одним из методов интерполяции для аппроксимации функций. Этот метод был разработан математиком Фридрихом Бесселем в XIX веке. Основным преимуществом интерполяции Бесселя является то, что она позволяет получить более точные результаты, чем другие методы интерполяции. Это связано с тем, что в данном методе используется ряд Фурье, который позволяет представить функцию в виде бесконечной суммы синусов и косинусов. Такой подход позволяет учесть особенности функции и получить более

точный результат. Другим преимуществом интерполяции Бесселя является то, что она может быть использована для аппроксимации функций, которые имеют сложную форму или не могут быть представлены в явном виде. Недосток данного вида интерполяции заключается в том, что данный метод требует большого количества вычислительных ресурсов, что может быть проблематично при работе с большими объемами данных. Кроме того, интерполяция Бесселя может быть достаточно сложной для понимания и реализации, что также является ее недостатком.

Сравнительные характеристики, выявленные при анализе алгоритмов численного интерполирования функций, оценка сложности и эффективности алгоритмов и их программной реализации по заданным метрикам представлены в таблице 1.

Таблица 1 – Сравнительные характеристики исследуемых методов интерполяции

Характеристика	Метод			
	Бесселя	Стирлинга	Ньютона	Лагранжа
Временная сложность	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
Вспомогательное пространство (память)	$O(n^2)$	$O(1)$	$O(n^2)$	$O(1)$
Время выполнения при 6 значениях (в наносекундах)	6170.299	7435.299	7499.0	5143.547
Точность (в процентах)	99.9%	99.8%	99.8%	99.8%
Преимущества	Лучшее значение при интерполяции вблизи центра	Наиболее быстрое уменьшение	Таблица различий упрощает поиск высоких порядков	Работает при неравномерно расположенных аргументах

На основании полученных результатов можно сделать вывод, что метод Лагранжа [5] обладает наилучшей точностью при малом количестве узлов интерполяции, но имеет высокую временную сложность и требует большого вспомогательного пространства. Метод Ньютона, в свою

очередь, имеет более низкую точность при малом количестве узлов, но работает быстрее и требует меньше памяти. Методы Стирлинга и Бесселя также имеют свои преимущества и недостатки в зависимости от конкретного случая.

В данной работе были проанализированы различные методы численного интерполирования функций, такие как метод Ньютона, метод Лагранжа, метод Стирлинга и метод Бесселя. Данные способы интерполирования являются эффективными методами для аппроксимации функций, которые используются в различных областях науки и техники. Были рассмотрены их особенности, преимущества и недостатки. В результате анализа было установлено, что каждый из методов имеет свои преимущества и недостатки, и выбор метода зависит от многих факторов. Кроме того, было отмечено, что для достижения наилучшей точности необходимо использовать достаточно большое количество узлов интерполяции. Таким образом, выбор метода численного интерполирования функций должен быть основан на анализе конкретной задачи и требуемой точности, а также необходимо учитывать особенности и ограничения каждого метода, чтобы получить наиболее точный результат.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Вычислительная математика для программистов: в 3 ч. Ч. 1: учеб. пособие / А.А. Бубнов, С.А. Бубнов, Е.Н. Проказникова. — М.: КУРС, 2018. — 144 с.
2. Знаменский, С. В. Численная оценка точности интерполяции несложных элементарных функций / С. В. Знаменский // Программные системы: теория и приложения. — 2018. — Т. 9, № 4(39). — С. 69-92.
3. Ситник, С. М. Приложения экспоненциальной аппроксимации по целочисленным сдвигам функций Гаусса / С. М. Ситник, А. С. Тимашов // Вестник Воронежского государственного университета инженерных технологий. — 2013. — № 2(56). — С. 90-94.
4. Подсадная, Е. А. Методы интерполяции высотных данных / Е. А. Подсадная, Я. В. Павлюк // Почвы и земельные ресурсы: современное состояние, проблемы рационального использования, геоинформационное картографирование: Материалы международной научно-практической конференции, посвященной 85-летию кафедры почвоведения БГУ и 80-летию со дня рождения В.С. Аношко, Минск, 20–23 сентября 2018 года. — Минск: Белорусский государственный университет, 2018. — С. 167-171.
5. Гончаров, А. С. Применение интерполяционных полиномов Лагранжа в кораблестроительных расчетах / А. С. Гончаров, С. В. Одинец, А. Д. Бурменский // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований: Материалы V Всероссийской национальной научной конференции молодых учёных,

Комсомольск-на-Амуре, 11–15 апреля 2022 года. Том Часть 2. – Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2022. – С. 193-195.

УДК 004.42

Моисеев А.В.

АНАЛИЗ СПОСОБОВ ИНТЕГРАЦИИ ПЛАТФОРМЫ 1С:ПРЕДПРИЯТИЕ И МЕССЕНДЖЕРА TELEGRAM

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Предмет исследования – способы интеграции платформы 1С:Предприятие и мессенджера Telegram.

Цель работы – сократить временные затраты на поиск информации студентами и повысить эффективность работы над курсовыми и выпускными квалификационными работами по схожей тематике.

В современном мире использование информационных технологий является необходимым условием эффективной и успешной работы предприятий в различных отраслях. В связи с этим автоматизация бизнес-процессов и интеграция различных программных систем становятся все более актуальными задачами. Одной из таких задач является интеграция системы управления 1С с мессенджером Telegram.

Представленный анализ позволит выбрать оптимальный метод интеграции, что повысит эффективность его работы и конкурентоспособность на рынке.

Существует два наиболее популярных способа интеграции 1С и Telegram. Первый из них – это встроенная в платформу 1С:Предприятие система взаимодействия, которая даёт возможность пользователям отправлять друг другу текстовые сообщения, файлы, устраивать видео и аудио конференции и демонстрировать экран компьютера. Вторым способом является непосредственное программирование backend части Telegram-бота на платформе 1С.

Оба способа интеграции 1С:Предприятие и Telegram имеют свои преимущества и недостатки.

Первый способ – использование встроенной системы взаимодействия в платформе 1С:Предприятие – предоставляет пользователям удобный интерфейс для обмена информацией [2]. Также, этот метод не требует дополнительного программирования и может быть легко настроен администратором, однако, этот способ имеет свои ограничения – например, он не позволяет программировать ботов с определенными функциями, которые могут быть полезны студентам в рамках курсовых и выпускных квалификационных работ. Однако, такой способ не подходит для проектов, которые используют другие

мессенджеры, такие как WhatsApp или Viber, и хотят интегрировать их с 1С:Предприятие [1].

Рассмотрим первый метод более подробно. Для интеграции 1С:Предприятие и Telegram с помощью встроенной системы взаимодействия в платформе 1С:Предприятие можно использовать следующий алгоритм.

1. Создать бота Telegram и получить API-ключ.
2. Запустить 1С:Предприятие и открыть нужную базу данных.
3. Создать новый сервис обмена с типом "Telegram" в меню "Настройка" → "Системные настройки" → "Сервисы обмена данными".
4. Указать необходимые параметры, такие как API-ключ, ID чата и другие.
5. Добавить обработчик на событие, которое необходимо отслеживать (например, новый заказ в интернет-магазине).
6. В обработчике вызвать метод отправки сообщения через Telegram API, передав текст сообщения и ID чата.
7. Сохранить настройки сервиса обмена данными и закрыть окно настроек.

После данных действий при возникновении события, на которое настроен обработчик, будет автоматически отправлено сообщение в указанный чат Telegram.

Второй способ – программирование backend части Telegram бота на платформе 1С – позволяет создавать ботов с любыми функциями и интегрировать их со сторонними сервисами [3]. Это даёт больше возможностей для студентов в рамках курсовых работ, так как они могут создавать ботов с уникальными функциями, которые помогут им с их исследованиями.

Однако такой способ требует программирования и не может быть легко настроен администратором без знания соответствующих технологий. Также этот метод может потребовать дополнительных затрат на серверное оборудование для запуска бота. Рассмотрим алгоритм создания backend-части Telegram бота на платформе 1С.

1. Определить функциональность бота и его возможности.
2. Создать бота Telegram и получить API-ключ.
3. Подготовить сервер для развертывания backend-части бота (возможно, готовый сервер уже имеется, если нет, то можно использовать любой облачный сервис).
4. Написать код backend-части бота на языке 1С, используя Telegram API для взаимодействия с ботом (подразумевается, что данные хранятся в БД 1С).
5. Скомпилировать код и загрузить на сервер.
6. Запустить backend-часть бота на сервере, чтобы он мог начать обрабатывать запросы от Telegram API.

Теперь можно использовать Telegram-бот с backend-частью на платформе 1С, который будет отвечать на сообщения пользователей, обрабатывать команды, отправлять уведомления и работать с базой данных.

Схемы алгоритмов рассмотренных способов интеграции приведены на рисунке 1.

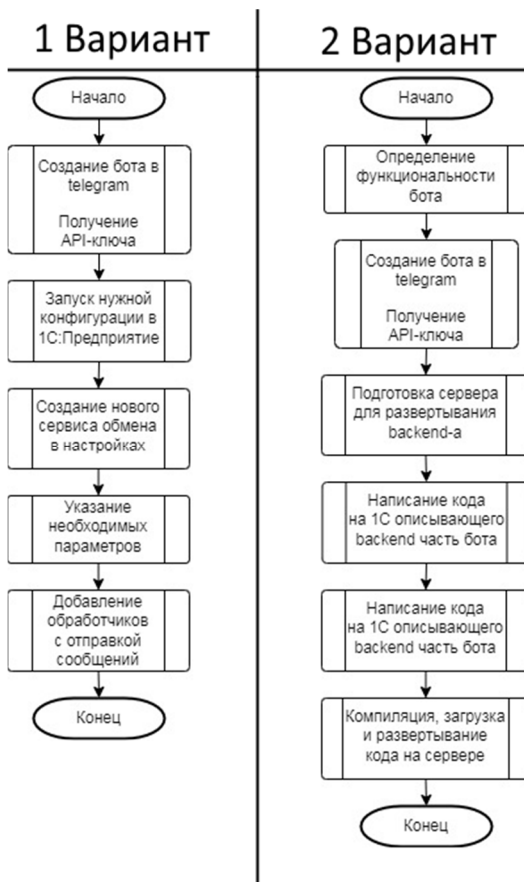


Рисунок 1 – Схемы алгоритмов интеграции 1С:Предприятие и Telegram

Таким образом, выбор способа интеграции 1С:Предприятие и Telegram зависит от конкретных потребностей студента. Если нужна простая система отправки сообщений и файлов, то первый способ может быть предпочтительнее. Если же требуются более сложные функции, связанные с работой ботов, то второй способ подойдет лучше.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. VachKirp Опыт интеграции мессенджера Telegram с 1C [Электронный ресурс] URL: <https://infostart.ru/1c/articles/819087/> (дата обращения: 21.05.2023).
2. «IRONSKILLS - Курсы по 1С» Интеграция 1С с Telegram за 12 минут [Электронный ресурс] URL: <https://youtu.be/UdtT3Oj-De4> (дата обращения: 13.05.2023).
3. «Желтый клуб — 1С программирование» Пишем TELEGRAM бота на 1С. С нуля и до ... [Электронный ресурс] URL: <https://www.youtube.com/live/peZsik57m4k?feature=share> (дата обращения: 13.05.2023).

УДК 517.925.

Поликашкина А.Р., Бабаян А.П.

ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОБЫЧНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ДЛЯ РАСЧЕТА И ПРОГНОЗИРОВАНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ ОБЪЕКТА В ПРОСТРАНСТВЕ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

Данная статья представляет обзор современных способов решения обычных дифференциальных уравнений, которые широко используются в механике: RKF45, метод Эйлера-Хойна и адаптивный метод TR_BDF2.

Обычные дифференциальные уравнения (далее по тексту ОДУ) являются одними из основных инструментов математического моделирования в различных областях науки и техники. Они позволяют описывать динамику систем, которые изменяются во времени [2]. Важное прикладное значение ОДУ имеют в физике. Например, они оптимизируют многие процессы, такие как исследование траектории перемещения физических тел в пространстве. Всякое движение происходит вследствие совместного действия ряда сил. Изучение механики перемещения требует знания характера этих сил и законов, определяющих их изменения. Если это известно, то исследование сводится к составлению дифференциальных уравнений движения и к последующему расчету этих уравнений и их интегралов.

В связи с этим, исследование применения ОДУ для расчета и прогнозирования траектории движения объекта в пространстве остается актуальной задачей в настоящее время.

При рассмотрении существующих методов описания траектории движущегося тела среди вычислительных алгоритмов были выбраны для реализации как наиболее перспективные: метод RKF45 (Runge–Kutta–

Fehlberg method, and is a method of order $O(h^4)$ with an error estimator of order $O(h^5)$) и некоторые другие [1].

Основная идея данного метода RKF45 состоит в использовании двух алгоритмов численной аппроксимации (четвертого и пятого порядков) на каждом этапе интегрирования. Оценка погрешности между результатом каждого из двух методов позволяет принять решение о том, когда следует изменять шаг времени. Если решение системы локально простое, то шаг можно увеличить. Также его можно уменьшить, чтобы обеспечить точность, когда система находится вблизи точки максимальной изменчивости.

Метод RKF45 может быть использован для решения широкого класса обычных дифференциальных уравнений, в том числе уравнений с переменными коэффициентами и нелинейных систем. Благодаря его скорости и адаптивности, в настоящее время он является одним из наиболее популярных способов решения ОДУ.

Метод RKF45 имеет порядок точности 5. Это означает, что ошибка метода пропорциональна пятой степени шага интегрирования. В сравнении, метод Эйлера-Хойна имеет порядок точности только 2, что приводит к гораздо большей погрешности приближения решения. Кроме того, метод Эйлера-Хойна не является адаптивным методом и требует фиксированного шага интегрирования, в результате чего, чтобы достичь минимальной погрешности вычисления, может потребоваться гораздо большее количество шагов, чем в методе RKF45.

В результате сопоставления с адаптивным методом TR_BDF2 – погрешность решения методом RKF45 во времени ниже (рисунок 1). Стоит отметить, RKF45 является более эффективным методом решения ОДУ, чем адаптивный TR_BDF2. Однако, адаптивный TR_BDF2 может быть полезным для быстрой оценки или для простых задач.

EulerHeun			RKF45			TR_BDF2_Adaptive		
Time step (dt)	Error (e)	e/dt	Time step (dt)	Error (e)	e/dt	Time step (dt)	Error (e)	e/dt
0.0033333	0.0051709	1.5513	0.0033333	0.0000000	0.0000	0.0033333	0.0000443	0.0133
0.0016667	0.0051709	3.1026	0.0016667	0.0000000	0.0000	0.0016667	0.0000443	0.0266
0.0008333	0.0051709	6.2051	0.0008333	0.0000000	0.0000	0.0008333	0.0000443	0.0532
0.0004167	0.0051709	12.4102	0.0004167	0.0000000	0.0000	0.0004167	0.0000443	0.1064
0.0002083	0.0051709	24.8204	0.0002083	0.0000000	0.0001	0.0002083	0.0000443	0.2128
0.0001042	0.0051709	49.6408	0.0001042	0.0000000	0.0001	0.0001042	0.0000443	0.4255
0.0000521	0.0051709	99.2816	0.0000521	0.0000000	0.0002	0.0000521	0.0000443	0.8510
0.0000260	0.0051709	198.5633	0.0000260	0.0000000	0.0004	0.0000260	0.0000443	1.7020
0.0000130	0.0051709	397.1265	0.0000130	0.0000000	0.0009	0.0000130	0.0000443	3.4040
0.0000065	0.0051709	794.2530	0.0000065	0.0000000	0.0018	0.0000065	0.0000443	6.8081

Рисунок 5 – Оценка погрешности решения ОДУ, полученная программным расчетом (слева-направо: метод Эйлера-Хойна, RKF45 и адаптивный TR_BDF2).

Метод RKF45 имеет вычислительную сложность, которая зависит от числа шагов, необходимых для достижения требуемой точности результатов. В среднем, он выполняет от 3 до 6 вычислительных операций на этапе. Это означает, что его вычислительная сложность линейна по числу шагов. Однако, метод RKF45 является адаптивным методом, что означает, что количество шагов, необходимых для достижения определенной точности, может быть неоднородным и зависит от конкретной задачи. Учитывая эту особенность, был сделан вывод, что вычислительная сложность метода RKF45 обычно лучше, чем у других методов того же порядка точности и составляет примерно $O(h^4)$, где h - размер шага.

При исследовании были рассмотрены многие распространенные и современные методы для решения обычных дифференциальных уравнений, каждый из которых обладает своими достоинствами и недостатками в работе. Выбирая тот или иной алгоритм, необходимо четко понимать цели расчета, имеющиеся в запасе ресурсы, необходимую точность расчета и другие факторы.

RKF45 показывает наибольшую эффективность в отличие от других рассмотренных способов и может быть рекомендован для решения обычных дифференциальных уравнений для расчета и прогнозирования траектории движения объекта в пространстве.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кокс Д. Численные методы для физиков / Кокс Д, Мэтьюс Р. – М.: Изд-во Моск. ун-та, 1973.
2. Колесников А.Н. Численные методы решения дифференциальных уравнений / Колесников А.Н. – М.: ФИЗМАТЛИТ, 2006.

УДК 004.056.53

Руди Д.И.

СРАВНЕНИЕ АУТЕНТИФИКАЦИЙ НА ОСНОВЕ СЕССИИ И JWT ТОКЕНЕ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Предмет исследования – аутентификация с использованием сессии и с использованием JWT токена.

Цель работы – описать преимущества и недостатки методов аутентификации.

Аутентификация - это процесс проверки и подтверждения подлинности личности пользователя, который пытается получить доступ к системе или ресурсу. В цифровой среде аутентификация может быть осуществлена различными методами, включая ввод логина и пароля, токены доступа, использование биометрических данных (например,

отпечатков пальцев или сканирование лица), или использование электронных ключей [1].

Аутентификация выполняется для того, чтобы защитить данные и ресурсы от несанкционированного доступа. Она также помогает определить, кто именно имеет доступ к определенным данным и ресурсам, и обеспечивает юридическую важность для документооборота.

Для сетевой информационной системы аутентификацию можно реализовать несколькими способами, которые обладают своими особенностями и недостатками, рассмотрим некоторые из них [1]:

На основе сессий;

С использованием токенов;

Аутентификация на основе сессий/cookie реализуется по принципу отслеживания состояний. Аутентификационная запись или сессия должны храниться и на сервере, и на клиенте. Для отслеживания активных сессий сервер использует базу данных или память, а на клиенте создается файл cookie с идентификатором сессии. С каждым запросом клиент отправляет идентификатор сессии. Таким образом, получив cookie, сервер сравнивает содержащийся в нем идентификатор сеанса с информацией, хранящейся в его памяти. Это позволяет серверу проверить вашу личность и предоставить ответ на основе соответствующего состояния. Этот метод является самым распространённым и широко используемым методом аутентификации, который применяется уже давно.

Преимущества сессионной аутентификации:

Легко использовать

Если вы работаете с веб-приложением, файлы cookie поддерживаются браузером на стороне клиента. Нет необходимости использовать какой-либо JavaScript для создания интерактивности.

Гибкость

Сервер может устанавливать продолжительность сессии и удалять ее, когда пользователь закрывает браузер, или когда происходит неактивность в течение некоторого времени. Это позволяет снизить нагрузку на сервер и сделать систему более гибкой в управлении.

Аналитика

Сервер может отслеживать активность пользователей в рамках каждой сессии и использовать эту информацию для лучшего понимания поведения пользователей и улучшения функциональности сайта или приложения.

Недостатки:

Ограниченная масштабируемость

Поскольку файлы cookie хранятся в памяти вашего сервера, масштабирование становится затруднительным, особенно там, где в системе слишком много одновременных пользователей. Однако с

аутентификацией на основе токенов все обстоит наоборот. Читай дальше, чтобы узнать больше.

Уязвимость к атакам

Если злоумышленник получит доступ к идентификатору сессии, он может изменить ее и получить доступ к аккаунту пользователя. Существуют различные методы защиты от этого, но риск всё равно остается.

Потребление ресурсов

Сервер должен постоянно поддерживать записи сессий, что может потреблять много ресурсов. Это может привести к проблемам с производительностью при использовании больших объемов данных.

Аутентификация на основе JWT (JSON Web Token) является методом аутентификации, который работает на основе токена, созданного в формате JSON. Этот токен содержит информацию об авторизации и пользователях, что позволяет использовать его для проверки подлинности пользователей.

JWT состоит из трех частей: заголовок, полезная нагрузка и подпись. Заголовок указывает алгоритм шифрования, используемый для подписи. Полезная нагрузка содержит информацию об авторизации, например, идентификатор пользователя, его роль и другие данные. Подпись является результатом выполнения алгоритма шифрования над заголовком и полезной нагрузкой и позволяет проверить подлинность токена [1].

Преимущества использования JWT:

Безопасность

JWT токен использует современную технологию шифрования для подписи и проверки. При этом все данные в токене зашифрованы и недоступны для третьих лиц. Кроме того, он не хранится на сервере, что минимизирует риски утечки пользовательских данных.

Масштабируемость

JWT позволяет распределенным приложениям, работающим на разных серверах, обмениваться данными аутентификации без необходимости передачи логинов и паролей. Токены удобны для использования в микросервисных архитектурах или API, где необходима безопасность и быстрый обмен данными между устройствами.

Упрощение разработки

Использование единого токена для управления аутентификацией пользователей позволяет упростить процесс разработки приложений и сайтов. Не нужно обеспечивать хранение логинов и паролей в базе данных, что снижает риски утечки пользовательских данных.

Недостатки аутентификация на основе JWT токенов:

Невозможность отзыва токенов

Один из главных недостатков JWT состоит в том, что они невосстанавливаемые. После того, как токен был создан, сервер не может его отозвать, что означает, что, если полученный токен будет скомпрометирован, злоумышленник сможет получить доступ к ресурсам и данные будут утрачены.

Сложность в управлении ключами

Еще одной проблемой ключа является сложность при их управлении на масштабах. Существуют сложные системы управления ключами, но они требуют дополнительных ресурсов.

Серверная нагрузка

Создание JWT токена, расшифровка его и проверка его подписи обеспечивают дополнительную серверную нагрузку. Если число пользователей высоко, это может привести к проблемам с производительностью системы.

Выше были приведены лишь некоторые из способов реализации аутентификации. Существуют также беспарольная аутентификация, когда определение пользователя происходит через одноразовые ссылки или коды, метод единой точки входа, позволяющий использовать общий сервис для управления входом пользователя в несколько разделенных сервисов системы, вход через сторонние общеиспользуемые сервисы, такие как Google или VK, двухфакторная аутентификация, подразумевающая 2 этапа проверки личности пользователя. У каждого способа есть свои преимущества и недостатки, которые необходимо учитывать при выборе способа подтверждения пользователя в разрабатываемой информационной системе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Молдовян А.А., Молдовян Д.Н., Левина А.Б. Протоколы аутентификации с нулевым разглашением секрета.– СПб: Университет ИТМО, 2016. – 55 с.

УДК 519.688

Саморукова О.Д., Крошилин А.В.

ПОСТАНОВКА И МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ИСПОЛНЕНИЯ БЮДЖЕТА ПРЕДПРИЯТИЙ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В статье сформулирована основная задача прогнозирования исполнения бюджета, а также рассмотрены основные возможные методы ее решения и их проблемные стороны.

Одной из важнейших задач экономического прогнозирования является задача оценки предполагаемых выплат и поступлений предприятий, а также оценка вероятности появления дефицита денежных

средств и способов его покрытия. Эту задачу можно назвать основной задачей бюджетирования, решение которой невозможно без прогностического управления расходами и доходами и оценки вероятных сценариев [4, 6, 7].

Бюджетирование – это система обязательных процедур и правил на всех этапах, начиная с планирования и заканчивая анализом исполнения бюджета, и, конечно, включая промежуточные этапы – учет, контроль и принятие решений [2].

Исполнение бюджета – это осуществление в течение бюджетного периода деятельности по управлению предприятием, направленной на достижение производственных и финансовых результатов в соответствии с показателями утвержденного бюджета.

Существуют различные формы бюджета: краткосрочный (месяц, квартал), оперативный (год) или прогнозный (больше года). В дальнейшем речь будет идти об исполнении годового (оперативного) бюджета предприятия [6].

На первоначальном этапе строится плановой годовой бюджет и утверждаются ключевые (целевые) показатели предприятия. С целью оперативного управления бюджетом предприятия используется уточненный годовой прогноз, который формируется путем последовательного замещения плановых показателей на фактические и актуализации первоначально сформированных данных на оставшиеся периоды. В сформированном прогнозном бюджете целевые показатели могут отличаться от первоначально запланированных. В таком случае компания может попытаться достигнуть утвержденных целевых показателей путем перераспределения бюджета будущих периодов, либо принять решение о их корректировке.

Поскольку управленческая отчетность и бюджетная деятельность предприятий строго не регламентируется, то конкретные форматы управления финансами сильно разнообразны. Они разрабатываются и утверждаются индивидуально для каждого предприятия, отражая его специфику. Общефирменные и коммерческие расходы распределяются на всех предприятий различным способом, так же различается группировка статей расходов и доходов. Таким образом, изначально ставится задача разработки некоего универсального алгоритма-оболочки, позволяющего анализировать одинаковые по своей логике, но различные по способу отражения данные управленческого учета [7].

С целью принятия управленческих решений в каждом временном промежутке необходимо построение, как минимум, трех вариантов прогноза: пессимистичный, оптимистичный и реалистичный. На практике же количество вариантов может быть гораздо больше. Например, зачастую возникает ситуации возможности корректировки ключевых показателей. Таким образом, варианты прогноза должны строиться как с

учетом проведения корректировок одного или нескольких показателей (совместно или попеременно), так и в условиях их сохранения на первоначальном уровне. Таким образом, необходимо построение такой модели, скорость решения которой будет достаточно высокой.

В настоящее время в прикладной экономике параллельно развивается два процесса: первый предполагает дальнейшее развитие классических инструментов, таких как эконометрическое моделирование, второй направлен на поиск и внедрение не классических для экономики средств и методов, таких как построение нейронных сетей [1]. Сформулированная задача прогнозирования исполнения бюджета также может быть решена с помощью указанных методов.

Традиционная укрупненная схема построения эконометрической модели прогнозирования включает следующие этапы:

1. Определение цели построения модели;
2. Определения множества переменных, включаемых в модель;
3. Сбор статистических данных, используемых для построения модели. Данные желательно собрать за наиболее длинный период времени;
4. Построение и оценивание модели;
5. Проверка адекватности полученной модели, по возможности ее упрощение [3].

Эконометрическое моделирование, как правило, с высокой точностью справляется с решением вопроса прогнозирования глобального изменения экономических показателей, исходя из полученных по статистическим данным трендов. При этом возникают проблемы с высокоточным краткосрочным прогнозированием, что обуславливается невозможностью модели улавливать быстрые изменения прогнозируемых показателей. Данный вопрос может быть решен построением нескольких моделей, использующих наиболее подходящие методы для конкретного горизонта прогнозирования, однако тогда не будет решена задача получения универсального инструмента прогнозирования исполнения бюджета предприятия.

Отличительная черта нейронных сетей заключается в том, что они позволяют решать задачи, с которыми не могут справиться традиционные методы. Они способны находить решения, опираясь на неполную, зашумлённую, искажённую информацию. Кроме того, нейросетевые модели обладают более высоким быстродействием процесса, также появляется возможность изменения топологии сети, исходя из требований решения задачи [5].

Однако существенной сложностью прогнозирования экономических показателей с использованием нейронных сетей является отсутствие на настоящий момент каких-либо четких алгоритмов создания

и улучшения моделей. В результате исследований, проводимых ранее, удалось лишь выявить ряд характерных особенностей:

1. Использование 1 или 2 скрытых слоев;
2. Количество нейронов не превышает число переменных входного слоя;
3. Обучение модели требует прохождения в среднем 3000 эпох;
4. Использование в качестве функции активации гиперболического тангенса.

При создании нейронной сети необходимо учитывать два важных правила, касающихся выбора количества слоев и числа эпох обучения. Небольшое количество скрытых слоев или прохождение малого числа эпох может привести к недообучению нейронной сети. Это означает, что она будет генерировать высокую ошибку на тестовой выборке с крайне низкими шансами получения точных прогнозов необходимых показателей. Наличие избыточного числа скрытых слоев, чрезмерно большого количества нейронов или отсутствия ограничения на число эпох может привести к переобучению сети, что означает, что на выходе максимально точное описание влияния переменных на тестовых данных на фоне неверных прогнозов.

С целью корректного выбора функции активации в процессе создания нейросети очень важно учитывать характер связей, возникающих между переменными. При моделировании экономических показателей необходимо учесть использование систем как с положительными, так и с отрицательными эффектами. В таких случаях, целесообразно применять в качестве функции активации функцию гиперболического тангенса. В отличие, например, от сигмоиды, которая принимает значения от 0 до 1, она центрирована относительно 0 [1].

В текущее время развитие получает совместное использование эконометрических методов и нейронных сетей. Такая процедура позволяет сохранить и эффективно совместить объяснительные возможности эконометрических моделей с вычислительными возможностями нейросетей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Балацкий Е.В., Юревич М.А. Использование нейронных сетей для прогнозирования инфляции: новые возможности // Вестник УрФУ. Серия экономика и управление. 2018. Т. 17, № 5. С. 823–838. DOI: 10.15826/vestnik.2018.17.5.037.
2. Бюджетирование. Контроль и анализ бюджета: [Электронный ресурс]. URL: <https://alfaseminar.ru/budgeting/3>
3. Некоторые подходы к прогнозированию экономических показателей.– М.: ИЭПП, 2005. С. 195 Агентство СІР РГБ

4. Прогнозирование бюджета: постановка непрерывного и сквозного управления: [Электронный ресурс]. URL: <https://tr52.ru/upravlenie-finansami/rashody/prognozirovanie-byudzheta-postanovka-neprepryvnogo-i-skvoznogo-upravleniya/>

5. Соловьева Ю.С., Грекова Т.И. Моделирование экономических процессов с применением нейросетевых технологий: Вестник ТГУ. 2009 №1 (6)

6. Саморукова О.Д., Крошилин А.В. Обзор математических методов прогнозирования бюджета предприятия // Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. Г.В. Овнчкина - Рязань: РГРТУ им. В.Ф. Уткина, январь 2023-124 с. (84-89)

7. Крошилина С.В., Саморукова О.Д. Роль информационных технологий в системе контроля и прогнозирования исполнения бюджета предприятий // Новые информационные технологии в научных исследованиях: материалы XXVII Всероссийской научно-технической конференции молодых ученых и специалистов; Рязань: ИП Коняхин А.В. (Book Jet), 2022 – 294 с.(72-74)

УДК 004.891.2

Серов И.Н.

ОБЗОР АВТОМАТИЗИРОВАННЫХ СИСТЕМ ДИАГНОСТИКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ АВТОМОБИЛЕЙ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В статье рассматриваются существующие
автоматизированные системы диагностики
технического состояния автомобилей.

Автоматизированные системы диагностики

Технический осмотр автомобиля – это обязательная процедура для всех транспортных средств. Она позволяет убедиться, что автомобиль работает исправно и не несет опасности ни водителю, ни пассажирам, ни окружающим участникам дорожного движения. С помощью технического диагностирования удастся узнать состояние узлов автомобиля, и на основе этих данных произвести категоризацию. Техническое состояние автомобиля делится на 6 видов:

1. исправное;
2. неисправное;
3. работоспособное состояние;
4. неработоспособное;
5. критическое состояние;
6. предельное состояние.

По итогу технического осмотра выносится вердикт, в каком состоянии находится автомобиль и можно ли его выпускать для дальнейшей эксплуатации по дорогам общего пользования.

За последние годы вышло несколько программ, предназначенных для упрощения внесения сведений о проводимых проверках технического состояния автомобилей и их последующей категоризацией. Постоянные обновления для соответствия современным требованиям сделали их функциональными. Разница при этом между ними ощутима, поэтому необходимо проанализировать и сравнить существующие программные обеспечения. Вот следующие из них:

«Диагностика ТС».

Программа проста в использовании и отвечает всем требованиям пунктов проведения техосмотра для осуществления необходимых манипуляций и передачи сведений в ЕАИСТО. Пользователи среди преимуществ отмечают невысокую стоимость и гибкую подстройку под нужды отдельной организации.

«Гелика».

Используется как для автоматизации перевода информации от измерительных приборов в компьютер, так и для создания сопровождающих документов. Это важно при выписке счетов и актов.

«Гелика» имеет такие преимущества:

- постоянное обновление с учетом пожеланий пользователей;
- просмотр статистики по любому из полей;
- формирование типа шин для второй оси;
- функцию подключения внешних отчетов;
- своевременное внесение изменений в соответствии с действующим законодательством.

«Пункт технического осмотра 3.0».

Она немного схожа по функционалу с Web-dk, однако имеет особые преимущества:

- большое количество выдаваемых отчетов (более 30-ти);
- ведение предварительной записи на техосмотр;
- гибкий интерфейс – можно настроить под себя, добавив или удалив ненужные графы;
- подключение контрольно-кассовой машины для печати чеков;
- выгрузка и интеграция с 1С;
- разделение по ролям: руководитель, бухгалтер, эксперт, офис-менеджер.

«АИСТО».

Программа направлена на прямое получение информации с измерительных приборов. И лишает пользователя выписки бухгалтерских документов.

Среди плюсов:

- передача сведений в ЕАИСТО одним нажатием кнопки;
- возможность отсрочить получение номера диагностической карты (при отсутствии интернета).

«ТО+».

Популярная программа со следующими особенностями:

- модуль для расчета страховки;
- подключение неограниченного количества пунктов ТО;
- оформление документов для оплаты – счета, акты, при этом все формы можно редактировать и подстраивать под себя;
- работа с мобильного телефона;
- голосовой ввод данных ускоряет процесс внесения информации в диагностическую карту;
- понятный функционал;
- доступная цена.

«ПИК».

Программа для технического осмотра ПИК имеет особенность: автоматическое получение данных сразу со многих диагностических станций.

Программа способна:

- создавать реестры прохождения ТО выбранным ТС;
- получать и расшифровывать информацию с диагностических стендов, аппаратов измерения уровня дыма и газов;
- импортировать регистрационные сведения из базы ГИБДД;
- ограничивать права пользователей, в том числе и на удаление результатов проведения техосмотра;
- распечатывать фотографии ТС в диагностическую карту;
- определять стоимость прохождения техосмотра (как первичного, так и повторного).

Из минусов – отсутствие выписки бухгалтерских документов.

Проанализировав вышеперечисленные программы, стало понятно, что они имеют свои характерные особенности, а также ряд плюсов и минусов. Стоит подметить, что данные автоматизированные системы очень хорошо справляются со своей работой, т.е. обеспечивают корректную работу станций технического обслуживания и осмотра, а также облегчают работу операторам и мастерам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Как выбрать программу для технического осмотра ТС <https://avtobddinfo.ru/faq/programma-dlya-tehnicheskogo-osmotra-ts>
2. Технический осмотр автомобилей: зачем он нужен и когда проводится <https://prav.io/browse/blogs/avtomobilnoe-pravo/tehnicieskiy-osmotr-avtomobiley-zachem-on-nuzhen-i-kogda-provoditsya>

УДК 004.021

Сидоров К. А., Крошилина С.В.

МОДИФИКАЦИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА RSYNC ДЛЯ МИНИМИЗАЦИИ ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ ПРИ СИНХРОНИЗАЦИИ ХРАНИЛИЩ ДАННЫХ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В статье рассматривается возможность улучшения
алгоритма Rsync для уменьшения объема
передаваемой информации при синхронизации
хранилищ данных.

Rsync – это алгоритм, задачей которого является минимизация передаваемой информации при синхронизации данных. Алгоритм был впервые описан в диссертации «Efficient Algorithms for Sorting and Synchronization» [3]. В этой работе представлена математическая модель алгоритма, которая показывает зависимость объема передаваемой информации по сети от нескольких переменных:

$$t(L) = (s_f + s_s) \frac{n}{L} + QL + s_t \left(\frac{n}{L} - Q \right) + O(1), \quad (1)$$

где t – количество байтов, которые необходимо передать по сети для выполнения синхронизации; L – длина блока разделения данных в байтах; s_f – размер быстрой хеш-суммы; s_s – размер стойкой хеш-суммы; n – размер измененного файла; Q – количество блоков, в которых были сделаны изменения (причем каждая последовательность байт изменения меньше, чем размер блока L и отделена от следующей последовательности блоком данных без изменений); s_t – размер ссылки на блок данных; $O(1)$ – дополнительная сервисная информация, необходимая, например, для указания позиции, с которой была удалена последовательность байт, и т.д.

Также автором работы [3] была предложена метрика «speedup», которая является показателем минимизации количества передаваемой информации при синхронизации. Метрика вычисляется по следующей формуле:

$$speedup = \frac{n}{t}, \quad (2)$$

где n – размер файла с изменениями в байтах; t – количество переданных байтов при синхронизации.

В статье [3] представлена только формула (2), полноценного исследования этой математической модели не проводится, в заключении автор лишь приводит выражение для нахождения оптимального размера блока L , принимая $s_f = 4$ байта («быстрый» хеш Adler32), $s_s = 16$ байт («стойкий» хеш MD4/MD5), $s_t = 4$ байта (размер ссылки на блок данных):

$$L = \sqrt{\frac{24n}{Q}}, \quad (3)$$

где n – размер файла с изменениями в байтах; Q – количество блоков, в которых были сделаны изменения (причем каждая последовательность байт изменения меньше, чем размер блока L и отделена от следующей последовательности блоком данных без изменений).

Выражение (3) не учитывает возможность использования других хеш-функций и длины ссылок на блоки данных, поэтому является недостоверным для реализаций, в которых не используются хеш-функции Adler32 и MD4/MD5, а также размер ссылки на блок данных, не равный 4 байт.

Также формула (3) на практике неприменима, так как для определения количества блоков Q , в которые были внесены изменения, необходим какой-то первоначальный блок разделения, от выбора которого зависит рассчитываемое значение L . Например, предположим, что имеется файл размером 1Мб, который был суммарно изменен на 16 Кб. В таблице 1 представлен расчет оптимального блока разделения L для данного примера.

Таблица 1 – Пример расчета оптимального блока L по выражению (3)

Размер файла и количество изменений	Возможные варианты Q	Возможные варианты L в байтах
Размер файла – 1048576 байт; количество изменений – 16384 байт.	$Q_1 = \frac{16384}{2048} = 8$	$L = \sqrt{\frac{24(1048576+16384)}{Q_1}} = 1787$
	$Q_2 = \frac{16384}{8192} = 2$	$L = \sqrt{\frac{24(1048576+16384)}{Q_2}} = 3575$
	$Q_3 = \frac{16384}{16384} = 1$	$L = \sqrt{\frac{24(1048576+16384)}{Q_3}} = 5055$

Вычисления в таблице 1 показывают, что использование формулы (3) не дает однозначного ответа, каким должен быть оптимальный размер блока L .

Исследование математической модели алгоритма Rsync представлено в магистерской диссертации «Tuning of rsync algorithm for optimum cloud storage performance» [2]. В этом исследовании модель (1) дорабатывается для случая, когда каждое изменение в файле может быть больше размера блока L :

$$t(L) = 4\frac{n}{L} + L \sum_{i=1}^Q \left\lceil \frac{n_i}{L} \right\rceil + 4\left(\frac{n}{L} - \sum_{i=1}^Q \left\lceil \frac{n_i}{L} \right\rceil\right) + O(1), \quad (4)$$

где t – количество байтов, которые необходимо передать по сети для выполнения синхронизации; L – длина блока разделения данных в байтах; n – размер измененного файла; Q – количество измененных частей

в файле; n_i – количество измененных байтов в i -ой части изменения; $O(1)$ – дополнительная сервисная информация.

Автор исследования [2] принял сумму $s_f + s_s = 4$ байт. В стандартной реализации алгоритма Rsync только один «быстрый» хеш равен 4 байта, а суммарный размер хеша для каждого блока равен $4 + 16 = 20$ байт (при использовании в качестве «стойкого» хеша MD4/MD5). В завершении исследования [2] приводится итоговое выражение нахождения оптимального размера блока L , однако данное выражение является недостоверным:

$$L = \sqrt{\frac{8n}{\sum_{i=1}^Q B_i}}, \quad (5)$$

где n – размер измененного файла; Q – количество измененных частей в файле; $B_i = \left\lceil \frac{n_i}{L} \right\rceil$ – количество блоков, которая занимает i -я последовательность (замена используется для возможности поиска минимума функции (4)).

Учитывая все вышесказанное, было предложено доработать математические модели (1) и (4) с учетом обнаруженных недостатков:

$$t(L) = (s_f + s_s) \frac{n}{L} + L \sum_{i=1}^Q B_i + s_t \left(\frac{n - C}{L} \right) + O(1), \quad (6)$$

где t – количество байтов, которые необходимо передать по сети для выполнения синхронизации; L – длина блока разделения данных в байтах; s_f – размер быстрой хеш-суммы; s_s – размер стойкой хеш-суммы; n – размер измененного файла; Q – количество измененных частей в файле; $B_i = \left\lceil \frac{n_i}{L} \right\rceil$ – количество блоков, которая занимает i -я последовательность (замена используется для возможности поиска минимума функции); s_t – размер ссылки на блок данных; $C = \sum_{i=1}^Q n_i$ – количество измененных байтов в новой версии файла; $O(1)$ – дополнительная сервисная информация.

Для нахождения минимума функции необходимо приравнять выражение (6) к нулю и затем продифференцировать по L [4], в итоге получаем следующую формулу для нахождения оптимального размера блока L :

$$L = \sqrt{\frac{n(s_f + s_s + s_t) - s_t C}{\sum_{i=1}^Q B_i}}, \quad (7)$$

где n – размер файла с изменениями в байтах; s_f – размер быстрой хеш-суммы; s_s – размер стойкой хеш-суммы; s_t – размер ссылки на блок данных; B_i – количество блоков, которая занимает i -я последовательность;

s_t – размер ссылки на блок данных; C – количество измененных байтов в новой версии файла.

Таким образом, было получено выражение (7) для расчета подходящего блока разделения файла. Для подтверждения достоверности предложенной математической модели (6) и полученного выражения (7) было проведено исследование со следующими этапами [5].

1. Создание компьютерной программы и ее проверка.
2. Выполнение предварительных прогонов, проверка соответствия программной модели.
3. Планирование экспериментов.
4. Выполнение рабочих прогонов.
5. Анализ выходных данных исследования.

На первом этапе была создана компьютерная программа на языке программирования Python. В качестве реального алгоритма Rsync в программе использована реализация «rutsync» из репозитория «GitHub» [1].

Для проверки соответствия программной модели была выполнена серия экспериментов с использованием алгоритма Rsync, после чего при помощи критерия согласия Пирсона [5] произведена оценка значимости различий между фактическими исходами работы реального алгоритма и теоретическими результатами, полученными с помощью математической модели (6).

Основной гипотезой H_0 является предположение о том, что модель (6) соответствует реальному алгоритму Rsync. Противоположной гипотезой H_1 выдвигается предположение, что модель (6) не соответствует алгоритму.

В качестве тестовых синхронизируемых данных использован текстовый файл размером 546 Кб – это средний размер файла в степенном распределении размеров файлов в файловой системе [2]. Файл модифицируется случайным образом в трех местах: добавлением случайной строки размером от 1 байта до 136 Кб; добавлением случайной строки размером от 1 байта до 13 Кб; удалением случайной строки размером от 1 байта до 27 Кб. Серия экспериментов была выполнена для блоков разделения L : 512 байт, 1024 байт, 2048 байт. Результаты представлены на рисунке 1.

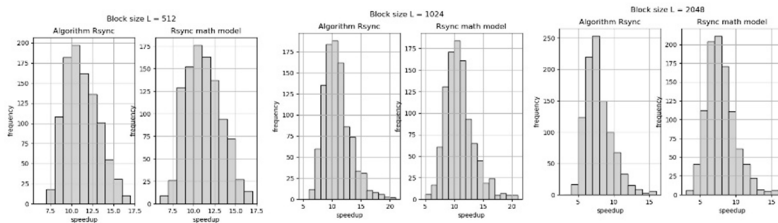


Рисунок 1 – Сравнение гистограмм метрики «speedup» реального алгоритма Rsync и его математической модели (6)

В конце каждого эксперимента был найден критерий согласия χ^2 (хи-квадрат), после чего найденное значение было сопоставлено с критической точкой распределения χ^2 [5], результаты представлены в таблице 2.

Таблица 2 – Сравнение $\chi^2_{\text{факт}}$ и $\chi^2_{\text{крит}}$ критериев согласия

Размер блока L	Число степеней свободы	$\chi^2_{\text{факт}}$	$\chi^2_{\text{крит}} \alpha = 0.05$	Вывод
512 байт	11	19.03	19.68	$\chi^2_{\text{факт}} < \chi^2_{\text{крит}}$ можно принять H_0
1024 байт	16	24.70	26.30	$\chi^2_{\text{факт}} < \chi^2_{\text{крит}}$ можно принять H_0
2048 байт	16	26.12	26.30	$\chi^2_{\text{факт}} < \chi^2_{\text{крит}}$ можно принять H_0

Таким образом, на основе данных таблицы 2 можно сделать вывод, что, действительно, математическая модель соответствует реальному поведению алгоритма, то есть модель адекватна исследуемой системе [5].

Далее было проведено основное исследование со следующими экспериментами.

1. Сравнение объема переданной информации для синхронизации при использовании оптимального блока L для соответствующего размера файла из исследования [3] и оптимального блока L, полученного с помощью выражения (7). В качестве тестовых синхронизируемых данных использованы абстракции файлов размерами: 512 Кб, 1 Мб, 10 Мб, 512 Мб, 10 Гб. В файлы вносятся изменения случайным образом размером от 1 байта до одной десятой размера файла.

2. Сравнение объема переданной информации для синхронизации при использовании постоянного размера ссылки на блок данных s_t (4 байта) и вычисляемого размера ссылки s_t в зависимости от текущего размера файла. В качестве тестовых синхронизируемых данных использованы абстракции файлов размерами: 127 Кб, 10 Мб. В файлы

вносятся изменения случайным образом размером от 1 байта до одной второй размера файла.

Первый эксперимент позволил подтвердить предположение о том, что для минимизации передаваемой информации при синхронизации размер блока разделения файла можно найти с помощью выражения (7). Второй эксперимент помог доказать возможность уменьшения объема переданной информации при использовании вычисляемого размера ссылки на блок данных s_t в зависимости от размера файла. Результаты выполнения рабочих прогонов экспериментов №1 и №2 представлены на рисунках 2 и 3.

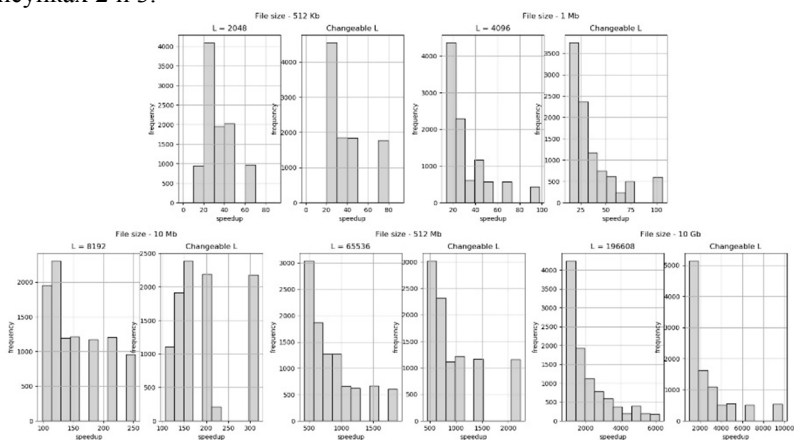


Рисунок 2 – Результаты эксперимента №1

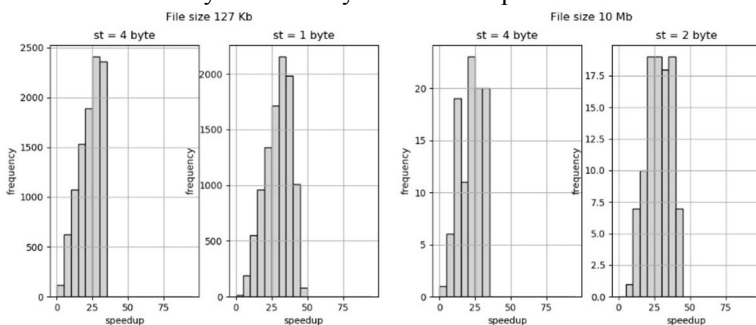


Рисунок 3 – Результаты эксперимента №2

На полученных гистограммах для эксперимента №1 видно, что при использовании вычисляемого размера блока L (правая часть рисунков – «Changeable L ») вместо рекомендованного размера L для данного размера файла [3] удастся достигать больших значений «speedup». Сравнение максимальных значений метрики «speedup» представлены в таблице 3.

Таблица 3 – Сравнение максимальных значений метрики «speedup» в эксперименте №1

Размер файла	Максимальное значение «speedup» при использовании рекомендованного значения L из исследования [3]	Максимальное значение «speedup» при использовании вычисляемого размера L с помощью выражения (7)
512 Кб	65	80
1 Мб	97	105
10 Мб	253	318
512 Мб	1870	2230
10 Гб	6200	9700

Таким образом, применение выражения (7) для вычисления блока L позволяет уменьшить количество передаваемой информации при синхронизации для некоторых вариантов изменений в исходном файле. При использовании выражения (7) также не ухудшается среднее значение «speedup» при сравнении с гистограммой, на которой L константа.

Эксперимент №2 позволил подтвердить предположение о возможности уменьшения количества передаваемой информации при использовании подходящего размера ссылки s_i на блок данных. В этом эксперименте в одном случае использовался максимальный размер ссылки, равный 4 байта, а в другом – минимальный размер для данного файла в соответствии с таблицей 4. Сравнение максимальных значений метрики «speedup» для эксперимента №2 представлены в таблице 5.

Таблица 4 – Максимальный размер синхронизируемых файлов в зависимости от размера ссылки на блок данных

Размер ссылки на блок данных	Максимальный размер синхронизируемых файлов
1 байт	$127 * 1024 = 127 \text{ Кб}$
2 байта	$32\,768 * 1024 = 32 \text{ Мб}$
3 байта	$8\,388\,608 * 1024 = 8 \text{ Гб}$
4 байта	$2\,147\,483\,648 * 1024 = 2048 \text{ Гб}$

Таблица 5 – Сравнение максимальных значений метрики «speedup» в эксперименте №2

Размер файла	Максимальное значение «speedup» при использовании размера ссылки 4 байта	Максимальное значение «speedup» при использовании подходящего размера ссылки на блок данных (1 или 2 байта)
127 Кб	35	50
10 Мб	35	43

По данным таблицы 5 видно, что применение подходящего размера ссылки на блок данных позволяет уменьшить объем передаваемой информации при синхронизации для файлов размером до 8 Гб (в соответствии с таблицей 4).

В результате проведенного исследования была доработана и проверена на адекватность исследуемой системе математическая модель алгоритма Rsync (6). Было получено выражение расчета размера блока L для минимизации передаваемой информации при синхронизации (7), актуальность которого была подтверждена после выполнения

экспериментов. Дополнительного уменьшения количества отправляемых данных можно также достигнуть при использовании ссылок на блоки данных переменного размера, в соответствии с таблицей 4.

Таким образом, результаты исследования можно применить на практике при разработке нового программного обеспечения.

В дальнейшем могут быть проведены следующие исследования алгоритма Rsync, его математической модели и выражения (7).

1. Исследование расчета оптимального размера блока L в зависимости от конкретных форматов синхронизируемых файлов (.doc, .ppt, .pdf и т.д.) и варианта изменений (добавление и удаление информации) в этих файлах.

2. Исследование связи оптимального размера блока L и вариантов применяемых хеш-функций («слабой» и «стойкой»).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. A pure Python module which implements the rsync algorithm [Электронный ресурс]. – URL: <https://github.com/isislovecruft/pyrsync> (дата обращения: 22.03.2023).

2. Constantinou M. Tuning of rsync Algorithm for Optimum Cloud Storage Performance. Bath, U. K. : Department of Computer Science, University of Bath, 2013. – 51 с.

3. Tridgell A. Efficient Algorithms for Sorting and Synchronization. PhD thesis. Australian National University, 1999. – 105 с.

4. Зельдович Я.Б., Яглом И.М. Высшая математика для физиков и техников. – М.: Наука, 1982. – 512 с.

5. Компьютерное моделирование: Учебник/ В.М. Градов, Г.В. Овечкин, П.В. Овечкин, И.В. Рудаков – М.КУРС:ИНФРА-М, 2019. – 264 с.

УДК 004.4

Слобожанин Ф.В., Жулева С.Ю.

РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВИЗУАЛИЗАЦИИ ДАННЫХ ПРЕДПРИЯТИЯ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина», Рязань

Работа описывает программную реализацию ПО визуализации данных и используемые при реализации программные средства разработки.

В настоящее время многие предприятия все чаще начинают использовать информационные технологии на своем производстве, улучшая или облегчая таким образом работу своих сотрудников.

Одним из способов упростить работу сотрудника и уменьшить возможные риски возникновения ошибки, является визуализация данных

предприятия. Визуализированные данные намного легче воспринимать, они наиболее наглядно демонстрируют аномалии, возникающие в процессе работы предприятия. [1].

Для решения выбран язык программирования Python, функциональные возможности которого позволяют реализовать любые потребности пользователя. Для визуального представления решаемых задач, например, отслеживания состояния объекта или мониторинга возникающих аномалий используется фреймворк Dash, а также библиотеки Plotly, numpy, pandas и др.

Dash - это бесплатный веб-фреймворк для создания интерактивных приложений на Python. Он позволяет быстро создавать интерфейсы для визуализированных данных и взаимодействия с ними. Dash использует Flask для создания веб-сервера и React для построения пользовательского интерфейса. Главная особенность данного фреймворка заключается в возможности совмещения кода Python и JavaScript.

Plotly - библиотека для создания графиков, диаграмм, презентаций и уникальных дашбордов. Она позволяет загружать данные и отображать их в Excel, MySQL, Redshift и т.д.. Отличительной чертой данной библиотеки от сравнительно похожих Matplotlib и прочих, является ее удобная интеграция с Dash, что делает использование обоих инструментов вместе наиболее удобным.

Используя данные средства разработки реализовано веб-приложение по визуализации данных атрибутов объекта. Например, ампераж и вольтаж генератора №1 (Рисунок 1).

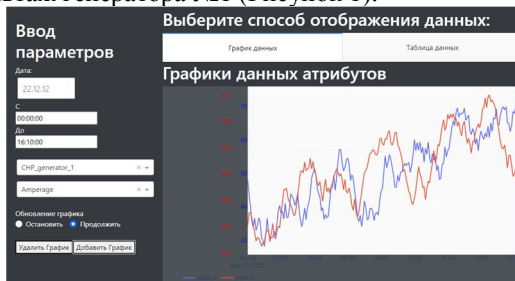


Рисунок 1 - пример веб-приложения

Для данного приложения были созданы модули MainLayout.py Runner.py. Разделение на модули необходимо для повышения читаемости кода и его структурированности. Это позволяет лучше организовать функциональность и обеспечить простую отладку.

В модуле MainLayout были описаны основные элементы формы, например, ввод параметров, создание графика. Пример кода, описывающего правила создания графика и таблиц:

```
graph_layout = html.Div([
```

```

        html.H1(children="Графики      данных      атрибутов",
style={'color': 'white'}),
        dcc.Graph(id="Chart",  figure=fig),      # График на
макете
        dcc.Interval(id="animateInterval",      interval=1500,
n_intervals=0)
    ])

    table_layout = html.Div([
        html.H1(children="Таблица      Атрибутов      Объекта",
style={'color': 'white'}),

dash_table.DataTable(data=object_data_array.to_dict('records'),
id='Table', page_size=12,
                        style_header={
                            'backgroundColor': '#18191b',
                            'color': 'white'
                        },
                        style_data={
                            'backgroundColor': '#4e5259',
                            'color': 'white'
                        })
    ])
В Runner же запускается локальный сервер, который использует
описанную выше компоновку:
dash.register_page(__name__, path='/')
app.title = 'Визуализация данных'
app.layout = cont
if __name__ == "__main__":
    # open_browser()
    app.run_server(debug=True)

```

Вывод. Фреймворк Dash при совместном использовании с Plotly позволяют быстро и просто создать макет для веб-приложения. Dash позволяет создавать не только элементы формы, но и сделать их интерактивными с помощью методов обратной связи, что дает разработчику возможность только ресурсами Dash создавать веб-приложение.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Слобожанин Ф.В. “Важность правильной визуализации данных в информационной системе предприятия и ее влияние на восприятие информации” // НИТ-2022, 2022. - 79с
2. Лещенко И.А. Разработка информационной системы визуализации данных для мониторинга и управления процессами на предприятии // Научный вестник НГТУ. 2017. №2 (67). С. 70-74.

3. Нгуен Х.Т. Подход к разработке и реализации информационной системы визуализации данных предприятия // Системы и средства информатики. 2018. Т.28. №2. С. 78-92.

4. Колибасов Д.А. Опыт применения программного обеспечения визуализации данных для управления бизнес-процессами на предприятии // Информационные технологии в бизнесе и образовании. 2019. №3 (37). С. 42-48.

УДК 004.422

Старостин М. А.

UML-МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПРИЛОЖЕНИЯ ДЛЯ ПРОКАТА ЭЛЕКТРОСАМОКАТОВ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина», Рязань

В статье рассматриваются особенности алгоритмической и логической реализации процесса аренды электросамоката на примере диаграммы деятельности.

Вот уже несколько лет подряд с наступлением весны во многих городах России можно наблюдать скопления электросамокатов на тротуарах, которые можно взять на прокат. Трудно поспорить с тем фактом, что за последние несколько лет кикшеринг (стяжение от кикскутер-шеринг, от англ. kick scooter sharing – краткосрочная аренда электросамокатов) стал невероятно популярным, и с каждым днём эта популярность только растёт. Эта популярность обусловлена прежде всего простотой эксплуатации электросамоката – для этого не нужно иметь водительские права или какие-то специальные навыки – достаточно одного смартфона со скачанным приложением. С ростом популярности кикшеринга растёт и количество организаций, предоставляющих данную услугу, а значит и количество мобильных приложений, реализующих необходимую функциональность для того, чтобы аренда электросамоката для новых пользователей была максимально простой и удобной в использовании.

Для более эффективной разработки какого-либо приложения необходимо сначала не только выделить его основные функции, но и детализировать особенности алгоритмической и логической реализации выполняемых приложением операций. Одним из наиболее часто используемых графических средств представления алгоритмов при проектировании информационных систем и мобильных приложений являются диаграммы. Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности.

Диаграмма деятельности – это своеобразная блок-схема, которая описывает последовательность выполнения операций во времени. Каждое

состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии [1].

В диаграмме деятельности обычно используются узлы трёх типов: узлы действий, узлы деятельности и узлы управления (разновидность узлов деятельности).

Графически состояния действий и деятельности обозначаются одинаково – в виде прямоугольника, боковые стороны которого заменены выпуклыми дугами, – но только в случае, если деятельность имеет известный алгоритм реализации (рисунок 1). Деятельности же с неизвестными алгоритмами реализации в местах их использования представляются ссылками на деятельности. На диаграммах они изображаются в виде закругленных по бокам прямоугольников со специальной пиктограммой в правом нижнем углу, напоминающей трезубец [2].

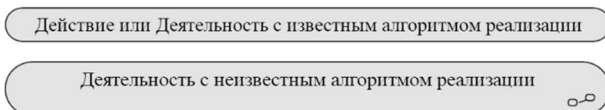


Рисунок 1 – Узлы действий и деятельности

Узлы управления в диаграммах деятельности с их изображениями и обозначениями представлены на рисунке 2 [3].

Название	Изображение	Что обозначает
Начальное состояние (initial node)	● →	Начало деятельности
Заключительное состояние (final node)	→ ●	Завершение деятельности
Разветвление управления (decision node)	→ (Y-образное разделение)	Начало альтернативных ветвей деятельности
Объединение управления (merge node)	(Y-образное слияние) →	Конец альтернативных ветвей деятельности
Развилка управления (fork node)	→ (горизонтальная линия с точкой)	Начало параллельных ветвей деятельности
Слияние управления (join node)	(горизонтальная линия с точкой) →	Конец параллельных ветвей деятельности
Комбинированное соединение-разветвление управления	→ (Y-образное разделение)	Последовательность из узла соединения и узла разветвления.
Комбинированное слияние-развилка управления	(горизонтальная линия с точкой) →	Последовательность из узла слияния и узла развилки.

Рисунок 1 – Узлы управления

Для мобильного приложения для кикшеринга была построена диаграмма деятельности, моделирующая алгоритм аренды электросамоката (рисунки 3, 4, 5). Диаграмма была создана с помощью программного инструмента моделирования StarUML. Стоит отметить, что на данной диаграмме также используются так называемые плавательные дорожки (swimlanes), основное назначение которых – показывать характер распределения функциональных обязанностей субъектов

(пользователя, системы проката электросамокатов и платёжной системы) на протяжении всего процесса.

Первая часть алгоритма реализации деятельности «Аренда электросамоката» представлена на рисунке 3.

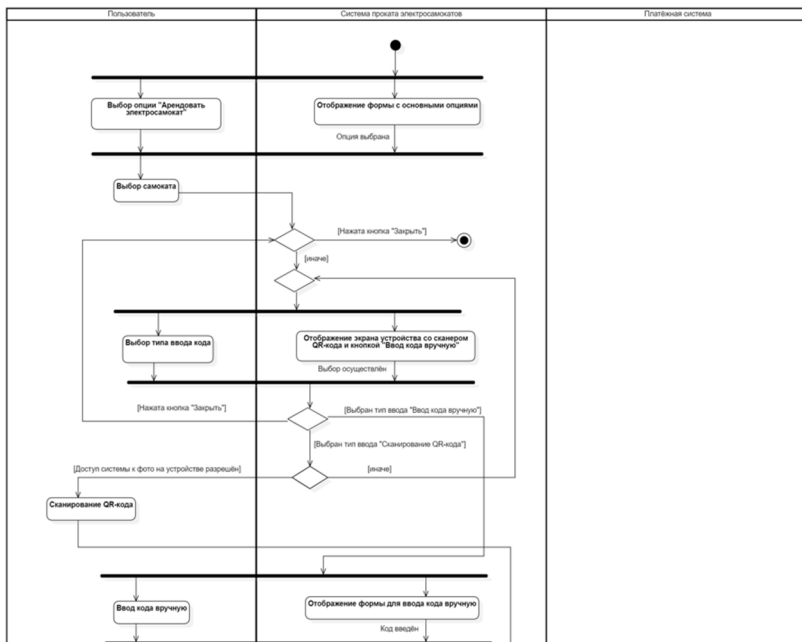


Рисунок 2 – Первая часть алгоритма реализации деятельности «Аренда электросамоката»

Для аренды электросамоката пользователь открывает мобильное приложение (предполагается, что пользователь уже зарегистрировался в нём), нажимает на пункт меню «Арендовать электросамокат» в приложении и выбирает любой понравившийся ему самокат на улице. Если после этого нажата кнопка «Закрыть», происходит завершение деятельности, иначе пользователь переходит к выбору ввода кода электросамоката. Он может сканировать QR-код, предварительно разрешив приложению доступ к фото на устройстве, или ввести код вручную.

Вторая часть алгоритма реализации деятельности «Аренда электросамоката» представлена на рисунке 4.

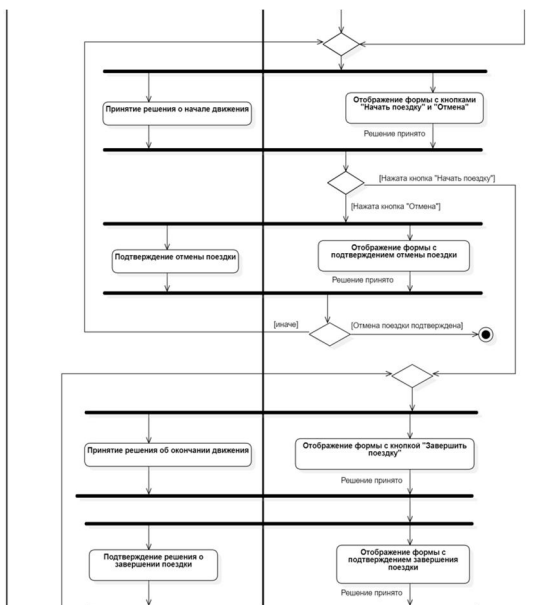


Рисунок 4 – Вторая часть алгоритма реализации деятельности «Аренда электросамоката»

Для начала движения пользователь нажимает на кнопку «Начать поездку». В случае, если пользователь решил отказаться от поездки, он нажимает на кнопку «Отмена» и подтверждает отмену поездки, после чего деятельность «Аренда электросамоката» завершается. Для прекращения движения пользователь нажимает кнопку «Завершить поездку». Чтобы избежать ситуации, когда пользователь случайно нажал на кнопку «Завершить поездку», в приложении также предусмотрена форма подтверждения завершения поездки.

Последняя часть алгоритма реализации деятельности «Аренда электросамоката» представлена на рисунке 5.

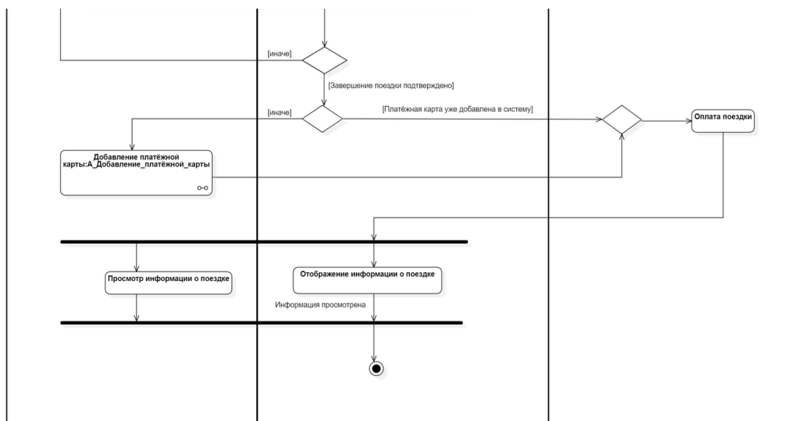


Рисунок 5 – Последняя часть алгоритма реализации деятельности «Аренда электросамоката»

Если платёжная карта уже добавлена в приложение, после прекращения движения происходит автоматическое списание денег с карты за поездку. В противоположном случае пользователю необходимо добавить платёжную карту в приложение. Деятельность «Добавление платёжной карты» представлена ссылкой на деятельность с помощью инструмента «SubactivityState», так как алгоритм реализации этой деятельности неочевиден. После оплаты пользователь имеет возможность просмотреть информацию о поездке – время в пути, расстояние, среднюю скорость и стоимость поездки – после чего деятельность «Аренда электросамоката» завершается.

Таким образом, в данной статье была детально рассмотрена и описана алгоритмическая реализация процесса «Аренда электросамоката» в мобильном приложении для проката электросамокатов, что позволит облегчить работу разработчиков.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Технология разработки ПО: Диаграмма деятельности [Электронный ресурс] / – URL: https://technologiarp.blogspot.com/p/blog-page_59.html
2. Белов, В. В. UML-моделирование функциональности и поведения программно-информационных систем: методические указания к лабораторным работам и практическим занятиям / В. В. Белов, В. И. Чистякова. – Рязань, Рязан. гос. радиотехн. ун-т. им. В. Ф. Уткина, 2022. – 48 с.
3. Белов, В. В. Проектирование информационных систем: учебник для студ. учреждений высш. проф. образования / В. В. Белов, В. И. Чистякова; под ред. В. В. Белова – М. : Издательский центр «Академия», 2013. – 352 с.

СОДЕРЖАНИЕ

Белов В.В., Перехода И.Ю., Фаддеев А.О.	4
СПЕЦИФИКА ОБРАБОТКИ ТЕКСТОВ С ПОМОЩЬЮ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ	
Беляев А.В., Дмитриева Т.А.	8
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ МЕДИЦИНСКОГО ЦЕНТРА	
Бубнов С.А., Трифонова Ю.С.	12
АНАЛИЗ СУЩЕСТВУЮЩЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В КОМПЛЕКСНОЙ ОЦЕНКЕ СОСТОЯНИЯ ЗДОРОВЬЯ ДЕТЕЙ	
Вапилина Е.Г., Симонова А.И., Билык Н.В., Дейнего И.С.	14
ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОБЫЧНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ	
Воробьев Я.В., Кузнецов М.Г., Журавлев П.В.	18
ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОДУ ДЛЯ МОДЕЛИРОВАНИЯ ОБЪЕКТОВ В ВИДЕОИГРАХ	
Дмитриева Т.А., Лапкин М.И., Лягушева М.А.	21
РАЗРАБОТКА НА ПЛАТФОРМЕ 1С МОБИЛЬНОГО ПРИЛОЖЕНИЯ СИСТЕМЫ ОБСЛУЖИВАНИЯ ДЛЯ ЗАКАЗА ТОВАРОВ И УСЛУГ	
Дмитриева Т.А., Лутиков В.А., Калинин А.Д., Фирсова Е.Д.	25
РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ 1С ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ КРЕАТИВНОГО ПРОСТРАНСТВА	
Журавлев П.В.	28
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ОНЛАЙН- ПЛАТФОРМЫ СОЦИАЛЬНОЙ СЕТИ МЕСТО ПРИ ПОМОЩИ REACT КОМПОНЕНТОВ.	
Бубнов С.А., Журавлева В.С.	30
АНАЛИЗ ТЕХНОЛОГИЙ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ГОЛОСОВ ПТИЦ В ЭПОХУ ГЛУБОКОГО ОБУЧЕНИЯ	
Кабочкин А.Н., Жулева С.Ю.	33
РЕАЛИЗАЦИЯ ПАРСЕРА НА ОСНОВЕ БИБЛИОТЕКИ HTMLAGILITYPASC: ПРИМЕНЕНИЕ СИНТАКСИЧЕСКОГО	

**АНАЛИЗА HTML-ДОКУМЕНТОВ В КОНТЕКСТЕ ИЗВЛЕЧЕНИЯ
ДАННЫХ****Камордин А.А., Крошилин А.В..... 36****ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА В ОБУЧЕНИИ
НЕЧЕТКИХ КОГНИТИВНЫХ КАРТ****Каширин И.Ю..... 38****РАЗРАБОТКА И ИССЛЕДОВАНИЕ ML-МОДЕЛЕЙ
С ПОМОЩЬЮ АВТОМАТИЗИРОВАННОГО ИСПЫТАТЕЛЬНОГО
СТЕНДА****Каширин И.Ю..... 41****КАЛИБРОВКА ML-МОДЕЛЕЙ ДЛЯ ИДЕНТИФИКАЦИИ
ФЕЙК-НОВОСТЕЙ****Клюкин Д.Д..... 44****ИССЛЕДОВАНИЕ ИНТЕРПОЛИРОВАНИЯ ФУНКЦИИ****Краев Д.В..... 47****РАЗРАБОТКА ПОЛИТИКИ ВЗАИМОДЕЙСТВИЯ МОДУЛЕЙ
ЗАГРУЗКИ ДОКУМЕНТОВ ПРИЛОЖЕНИЯ ЗАПУСКАЕМОГО В
КИБЕРИММУННОЙ СРЕДЕ****Копылов И.Ю., Лапкин М.И., Лягушева М.А., Овечкин Г.В.,
Хлебников Н.М..... 50****ЧИСЛЕННОЕ ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ****Моисеев А.В. 54****АНАЛИЗ СПОСОБОВ ИНТЕГРАЦИИ ПЛАТФОРМЫ
1С:ПРЕДПРИЯТИЕ И МЕССЕНДЖЕРА TELEGRAM****Поликашкіна А.Р., Бабаян А.П..... 57****ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ОБЫЧНЫХ
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ДЛЯ РАСЧЕТА И
ПРОГНОЗИРОВАНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ ОБЪЕКТА В
ПРОСТРАНСТВЕ****Руди Д.И. 59****СРАВНЕНИЕ АУТЕНТИФИКАЦИЙ НА ОСНОВЕ СЕССИИ И JWT
ТОКЕНЕ**

Саморукова О.Д., Крошилин А.В.	62
ПОСТАНОВКА И МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ИСПОЛНЕНИЯ БЮДЖЕТА ПРЕДПРИЯТИЙ	
Серов И.Н.	66
ОБЗОР АВТОМАТИЗИРОВАННЫХ СИСТЕМ ДИАГНОСТИКИ ТЕХНИЧЕСКОГО СОСТОЯНИЯ АВТОМОБИЛЕЙ	
Сидоров К. А., Крошилина С.В.	69
МОДИФИКАЦИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА RSYNC ДЛЯ МИНИМИЗАЦИИ ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ ПРИ СИНХРОНИЗАЦИИ ХРАНИЛИЩ ДАННЫХ	
Слобожанин Ф.В., Жулева С.Ю.	76
РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВИЗУАЛИЗАЦИИ ДАННЫХ ПРЕДПРИЯТИЯ	
Старостин М. А.	79
UML-МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПРИЛОЖЕНИЯ ДЛЯ ПРОКАТА ЭЛЕКТРОСАМОКАТОВ	

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Межвузовский сборник научных трудов

Подписано в печать 30.06.2023. Формат 60х84 1/16.

Бумага офсетная. Печать цифровая.

Гарнитура «Times New Roman».

Усл. печ. л. 5,5.

Тираж 100 экз. Заказ № 6359

Рязанский государственный радиотехнический
университет имени В.Ф. Уткина
390005, г. Рязань, ул. Гагарина, д. 59/1

Отпечатано в типографии Book Jet
390005, г. Рязань, ул. Пушкина, д. 18

Сайт: [http:// book jet.ru](http://book.jet.ru)

Почта: [info@ book jet.ru](mailto:info@book.jet.ru)

Тел.: +7(4912) 466-151