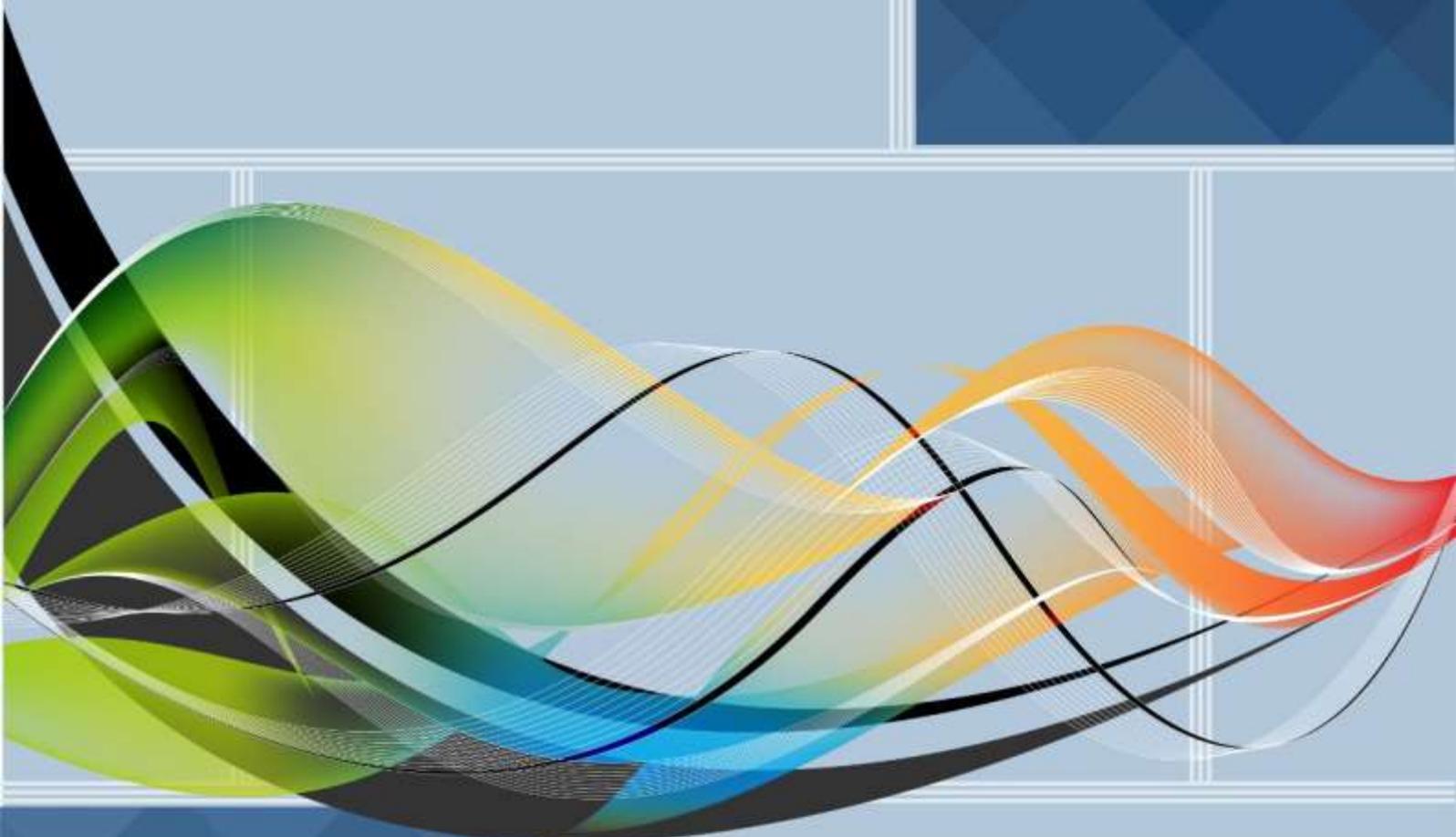


16+



XXIV Всероссийская
студенческая научно-практическая
конференция Нижневартовского
государственного университета



Часть 3

Информационные технологии

Нижневартовск, 5-6 апреля 2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Нижевартовский государственный университет»

**XXIV Всероссийская студенческая
научно-практическая конференция
Нижевартовского
государственного университета**

Часть 3

Информационные технологии

*г. Нижевартовск,
5-6 апреля 2022 г*

Нижевартовск
НВГУ
2022

Печатается по решению Ученого совета
ФГБОУ ВО «Нижевартовский государственный университет»
(протокол № 1 от 18.01.2022 г.)
Приказ № 043-О от 05.03.2022

В 85 **XXIV Всероссийская студенческая научно-практическая конференция
Нижевартовского государственного университета (г. Нижевартовск, 5-6
апреля 2022 г) / Под общей ред. Д.А. Погоньшева. Ч. 3. Информационные
технологии. Нижевартовск: изд-во НВГУ, 2022. 268 с.**

ISBN 978-5-00047-646-8

ББК 72я43



Тип лицензии CC, поддерживаемый журналом: Attribution 4.0 International (CC BY 4.0).

ISBN 978-5-00047-646-8



9 785000 476468 >

© НВГУ, 2022

УДК 004.9

Аверьянова И.Г., Лосева А.В., Башарина С.О.
Воронежский государственный педагогический университет,
г. Воронеж, Россия

ПЛАТФОРМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ: ПОПУЛЯРНОСТЬ И ФУНКЦИОНАЛ

С древних времен люди были заинтересованы вопросом коммуникации на расстоянии: почта, радио, телефон, интернет. В условиях постиндустриального общества технический прогресс затронул каждую отрасль жизни людей, в том числе и образование. В настоящее время изменчивость информационной среды находится на высоком уровне: с каждым годом внедряются все больше и больше технологий в повседневность.

В эпоху развития «всемирной паутины» обратимся к теме онлайн-образования. В условиях пандемии данная тема является особенно актуальной. Начиная с 2020 года, все российские школы в разной степени практиковали дистанционное обучение в рамках борьбы с распространением COVID-19.

Дистанционное обучение — взаимодействие учителя и учащихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты и реализуемое специфичными средствами Интернет-технологий или другими средствами, предусматривающими интерактивность [1]. В самом определении дистанционного обучения сделан акцент на оснащение техническими средствами. Для эффективного достижения образовательных целей посредством информационных технологий важно организовать двухсторонний доступ к таковым: со стороны ученика и со стороны учителя.

Анализируя рынок образовательных услуг, можно отметить активную тенденцию к увеличению оборота. По результатам различных исследований, рост рынка образовательных услуг за последние 5 лет составил от 20 до 30%. Самыми популярными причинами получения онлайн-образования являются стремление повысить квалификацию и развить кругозор в наиболее интересной для человека теме.

Каждый вид дистанционного обучения базируется на интернет-платформах. На сегодняшний день специалистами разработано большое количество разнообразных сервисов, однако широкое распространение получают лишь некоторые из них. Платформы различаются по возможностям и набору функций, интерфейсу и доступности.

В рамках проведенного исследования по использованию образовательных платформ в общеобразовательных школах Воронежской области, наиболее распространены «Дневник.ру» и «Учи.ру». Также некоторые школы работают на базе платформ «РЭШ» и «ЯКласс» (рис 1.).

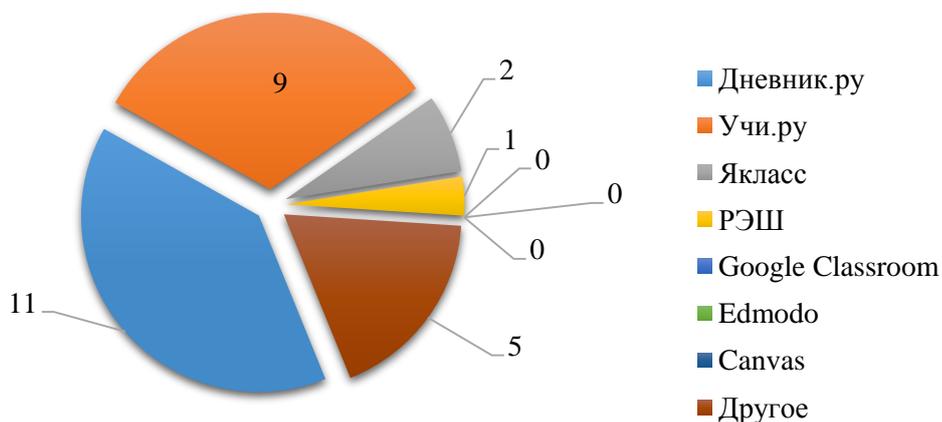


Рис. 1. Анализ используемых интернет платформ

Отличительной особенностью проведенного опроса стало то, что использование малоизвестных, но более эффективных платформ в школах ограничено. Существуют практичные платформы на базе всемирно известных сервисов: “Google Classroom”, “Edmodo” и “Canvas”. Данные платформы многофункциональны и удобны в использовании, так как основаны на привычных комбинациях и содержат доступный интерфейс. Рассмотрим наименее популярные платформы, выделим их преимущества и недостатки для того, чтобы расширить область применения. Начнем с образовательной платформы от Google – “Google Classroom” (<https://clck.ru/VBjAm>). Данная платформа создана с использованием популярных приемов программирования и привычных интерфейсов, разработанных корпорацией Google (рис. 2.). На данном программном обеспечении есть возможность организации обучения не только школьников, но и студентов, а также использования платформы для проведения курсов повышения квалификации специалистов различных сфер деятельности.

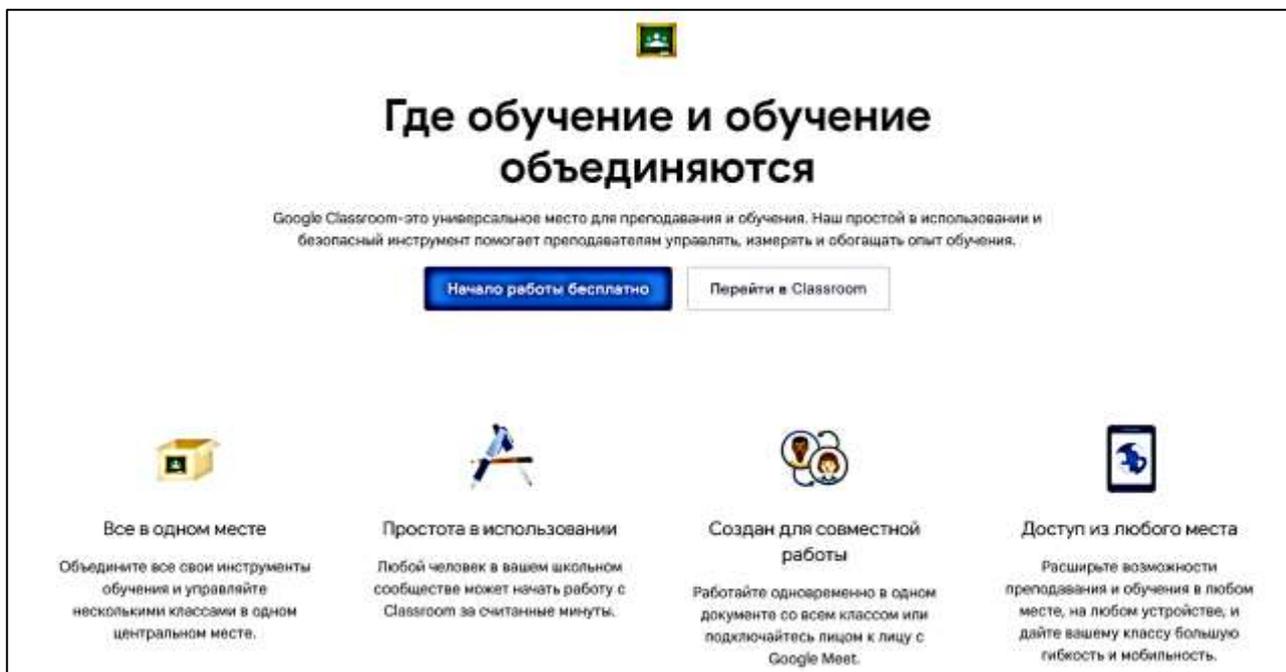


Рис. 2. Начальная страница сайта “Google Classroom”

Главная особенность платформы – возможность неформального общения между преподавателями и студентами в режиме реального времени. «Google Classroom» объединяет возможности хранения информации такие как «Google.Диск», возможности мессенджеров «Google.Почта», «Google.Hangouts» и возможности обработки информации «Google.Таблицы», «Google.Документы». Привычные алгоритмы взаимодействия значительно облегчают использование платформы и этап приспособления к ней. Образовательные ресурсы — учебники, видеофайлы, презентации, таблицы хранятся в облачном хранилище и имеют общий доступ для всех пользователей образовательного сервиса. Более того, интеграция с «Google.Видеовстречи» позволяет проводить онлайн-конференции, так как возможности видеочатов расширены. Среди незначительных недостатков образовательной платформы отметим следующие:

1. Отсутствует возможность проведения онлайн-конференций для большой аудитории студентов;

2. Отсутствует фиксация успеваемости в бесплатной версии программы;

3. Установлен максимальный порог пользователей в группе.

Отдельно отметим, что на данной платформе существует два типа аккаунтов:

1. Учебного заведения — создается администрацией образовательного учреждения, включает расширенный объем функций;

2. Студента (личный аккаунт) — может быть создан самостоятельно, может не быть привязан к учебному учреждению и использоваться для достижения целей самообразования.

Также стоит рассмотреть сервис обмена сообщениями «Edmodo» (<https://clck.ru/Ju2mg>). Данную платформу можно характеризовать как специализированную социальную сеть, созданную для осуществления образовательного процесса (рис. 3).

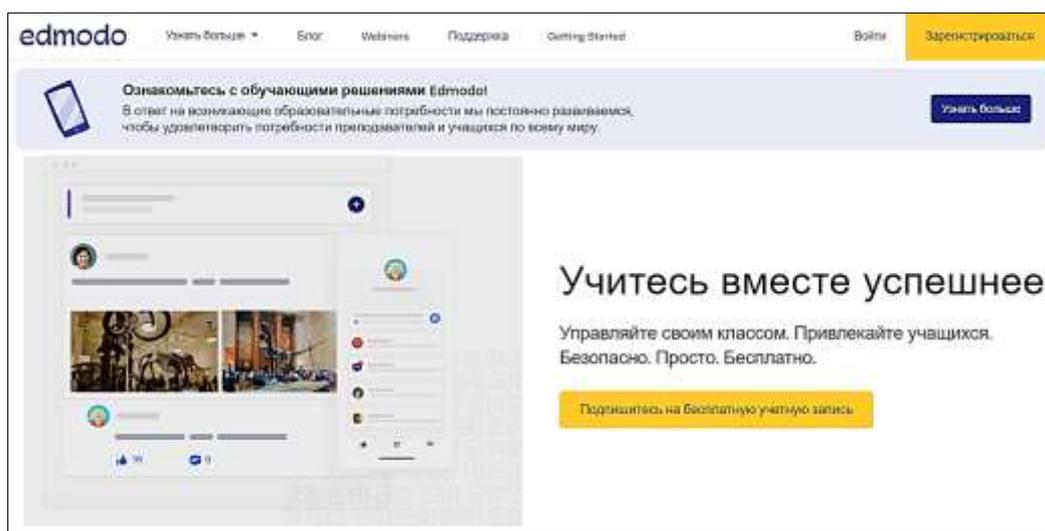


Рис. 3. Начальная страница сайта Edmodo

Данная платформа осуществляет обмен информацией между преподавателями, студентами и их родителями. Среди основных функций выделяются следующие:

1. Обмен информацией;

2. Генерация и распространение заданий (теоретических и практических);
3. Сбор и обработка ответов.

“Edmodo” позволяет формировать викторины и прикреплять файлы для отправки учащимся. Хранение образовательных материалов осуществляется в формате библиотеки из гиперссылок на оригинальные источники. В интерфейсе платформы отдельно выведен блок информации для публичных объявлений, доступных для всех пользователей. Более того, в рамках платформы сформирован сервис создания опросов, что позволяет учитывать мнения студентов и преподавателей по различным образовательным и организационным вопросам.

Общий вид платформы также прост и понятен для начинающих пользователей. Визуально интерфейс можно сравнить с популярной российской социальной сетью «ВКонтакте», которой пользуются более 95% школьников и студентов на территории Российской Федерации. Данная визуальная схожесть помогает оперативно адаптироваться и сразу начать обучения, минуя этап изучения самой платформы (рис. 4).

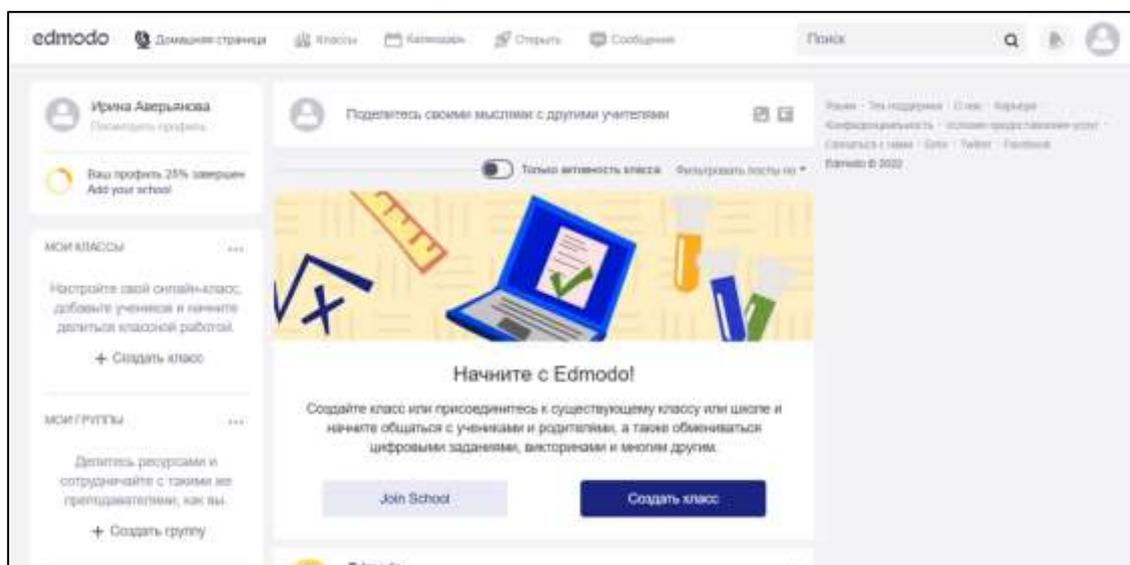


Рис. 4. Рабочая страница сайта «Edmodo».

Профили каждого пользователя имеют высокую степень защиты персональных данных с привязкой к электронной почте и телефону, что минимизирует вероятность несанкционированного использования учебных материалов и личных разработок учеников и преподавателей. Отдельно ведется планер-календарь, где в режиме реального времени формируется таблица проводимых конференций и вебинаров, которые могут посещать пользователи со схожими интересами.

Отдельно стоит выделить факт, что данная платформа адаптирована под все виды устройств и использование ее как планера ближайших событий. Выделим в использовании данной платформы следующие недостатки:

1. Ограниченный функционал — формат социальной сети предопределяет отсутствие некоторых технических возможностей;

2. Низкая популярность — платформа не занимает лидирующих позиций в рейтинге образовательных сервисов, используемых в России и, более того, не ведет активную деятельность по привлечению целевой аудитории;

3. Основная функция — мессенджер.

Далее рассмотрим еще одну малоизвестную образовательную платформу – «Canvas» (<https://clck.ru/ebpr6r>). Это бесплатная образовательная электронная среда для организации массовых курсов в режиме онлайн. Данный формат обучения возможно интегрировать в образовательную программу школ и высших учебных заведений, а также использовать как отдельную платформу для проведения вебинаров. Данная платформа отличается свободным доступом и многообразием инструментов для создания уроков, курсов и вебинаров. Бесплатный доступ реализован для студентов и преподавателей.

Программный комплекс «Canvas» изначально был создан как инструмент дистанционного обучения. Он применяется некоторыми колледжами, школами, институтами и другими учебными заведениями. Более того, во многих организациях через данную платформу проводятся курсы повышения квалификации персонала. Важной особенностью является и то, что помимо веб-версии, платформа доступна в виде приложений для мобильных устройств. В функционал данной платформы включены все функции, необходимые для реализации процесса обучения: управление классами, генерация тестовых заданий, промежуточных аттестаций, мониторинг успеваемости, существует возможность проектной работы учеников и обширная система хранения литературы.

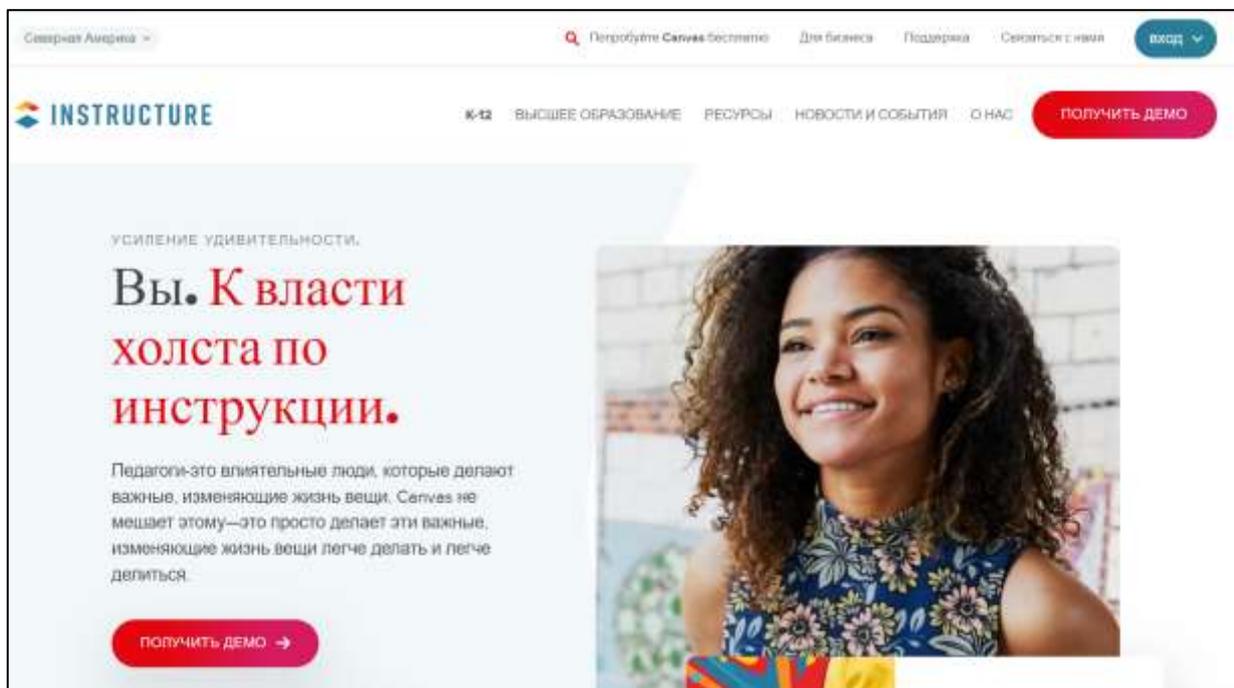


Рис. 5. Стартовая страница платформы «Canvas»

Кратко приведём обзор основных возможностей программного обеспечения «Canvas»:

1. Разработка курсов с иллюстративным содержанием;
2. Интеграция с разнообразными, ранее созданными курсами;

3. Возможность организации дебатов онлайн с привлечением экспертов и зрителей;
4. Мониторинг процесса обучения преподавателями и учащимися, возможности стимулирования и изменения процесса обучения;
5. Проведение онлайн-вебинаров, конференций, объединяющих большие группы слушателей;
6. Доступ родителей, кураторов, руководителей к данным о прогрессе обучающегося для своевременного реагирования.

Стоит отметить, что данная платформа поддерживает все известные источники информации: видеолекции, тесты, викторины, конференции, графическое представление материала. В интерфейсе программного обеспечения предусмотрен календарь, планер, почта, техническая поддержка, а также поисковая строка. Также на платформе можно не только проводить и создавать курсы, но и проходить уже созданные, для этого предусмотрена «Библиотека курсов». Особенно удобно, что все курсы, представленные на платформе, бесплатные, не ограничены во времени и подходят для самостоятельного обучения.

Явных недостатков платформы не выявлено, но отметим, что данное программное обеспечение ориентировано на самостоятельный сегмент и практическую образовательную деятельность. Применение данного сервиса предполагает наличие творческих способностей у преподавателей.

Анализ малоизвестных платформ показал, что отсутствие активного продвижения и распространения сохраняют способность платформ выполнять образовательные функции. Выбор платформы зависит от учебного заведения и целей обучения, но малоизвестный не означает неприемлемый. Данные платформы могут занять заслуженное место среди популярных образовательных платформ, значительно улучшить процесс дистанционного образования и найти пользователей, которым существующий функционал оптимально подходит для достижения образовательных целей.

Литература

1. Лебедева М.Б. Дистанционные образовательные технологии: проектирование и реализация учебных курсов. СПб: БХВ-Петербург, 2010. 330 с.

© Аверьянова И.Г., Лосева А.В., Башарина С.О., 2022

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕРВИСОВ ПО СОЗДАНИЮ ИНТЕРАКТИВНОГО ВЕБ-КОНТЕНТА

Интерактивный веб-сайт – это динамический, изменяющийся каждый день по содержанию, ресурс. Слово «интерактивность» происходит от однозвучного английского термина “interaction”, что в переводе означает «связь». В сети интернет человек считается соучастником интенсивного взаимодействия с иными субъектами. Целями данного взаимодействия являются приобретение данных, либо контакт. Уровнем интерактивности считается степень стремительности и удобство пользователя. Компоненты интерактивности – это функции либо программные модули веб-сайта, вызванные прийти на помощь людям по взаимодействию с ресурсом либо иными пользователями [1].

Создание интерактивного веб-сайта предполагает под собой разработку интерактивных элементов, при поддержке которых станет происходить взаимодействие с посетителями. Такими элементами могут быть: интерактивные иллюстрации и инфографика, различного рода конфигураторы, калькуляторы, тесты, викторины, опросы и т.д. Данный контент можно создавать с помощью специальных сервисов и затем размещать на сайте.

Преимуществами интерактивного веб-контента являются:

- возможность выделиться на фоне конкурентов, которые имеют статические сайты.
- повышение вовлеченности целевой аудитории и конверсии.
- долгое время не теряет актуальности.

Приведем статистику по интерактивному веб-контенту:

1. Интерактивный контент может генерировать в 2 раза больше конверсий и 4-5 раз больше просмотров страниц, чем пассивный контент.
2. 87% маркетологов, использующих интерактивный контент, согласны с тем, что он привлекает внимание зрителя более эффективно, чем статический контент.
3. 93% маркетологов заявили, что интерактивный контент намного более эффективен при обучении пользователя.

Во многих сферах человеческой деятельности часто возникает задача оптимального выбора, т.е. выбора объекта с максимальной количественной оценкой его качества из некоторого имеющегося набора исследуемых объектов [2, с. 16]. При этом оценка качества может быть как интегральной, т.е. по всем рассматриваемым критериям, так и частной (по одному или группе критериев).

Для осуществления такого выбора необходимо проведение анализа, который для изучения сложных, многоаспектных и противоречивых объектов требует серьезных аналитических усилий. В общем случае анализ должен быть комплексным, т.к. нет смысла анализировать отдельно взятые объекты, процессы и явления без конкретного социально-

экономического, культурного или какого-либо другого контекста, без всестороннего анализа его предпосылок и последствий [2, с. 24].

Проведем сравнительный анализ для интегральной оценки качества ресурсов по созданию интерактивного веб-контента. Критерии для проведения сравнительного анализа:

- функционал — какой тип интерактивного веб-контента можно создать с помощью данного ресурса;

- легкость и понятность в использовании – насколько просто неподготовленному пользователю разобраться в программе и насколько понятно, как в ней работать;

- быстрота создания контента – насколько быстро и правильно можно создать интерактивный веб-контент неподготовленному пользователю;

- интерфейс — насколько интерфейс ресурса является приятным и понятным пользователю;

- стоимость использования ресурса в месяц.

Рассмотрим десять наиболее популярных сервисов по созданию интерактивного веб-контента: Ceros; SnapApp; MapMe; Apester; Ion Interactive; ContentTools; Playbuzz; Calculoid; LeadQuizzes; Qzzr (Таблица).

Сервис по созданию интерактивного веб-контента Ceros дает возможность создавать как уникальный интерактивный контент, так и добавлять интерактивные элементы в уже существующий. С помощью данного сервиса можно интегрировать динамичные и интерактивные элементы в инфографику, электронные книги, журналы, баннерные материалы и т.д. [3, с. 87]. Кроме визуальных эффектов, разнообразных анимаций и шаблонов ресурс предоставляет также аналитику: показатели работы готового материала, время загрузки, ожидания и как каждый пользователь взаимодействует с контентом.

Платформа SnapApp также предназначена для создания интерактивного веб-контента. На ней можно создавать различные интерактивные видеоролики, инфографику, калькуляторы, тесты и викторины. Платформа очень проста в использовании и ее сильной стороной является то, что на ней есть возможность интеграции с такими системами, как Pardot, HubSpot и Salesforce для максимального использования веб-контента и привлечения еще большего количества потенциальных клиентов [4]. Платформа также предоставляет обширную аналитику для интерактивного веб-контента.

Сервис MapMe позволяет создавать интерактивные 3D-карты с изображениями и видео, которые помогут увеличить время пребывания клиентов на сайте и предоставить им больше информации о товарах [5, с. 4].

С помощью инструмента Apester есть возможность создавать такой интерактивный веб-контент, как тесты, опросы, викторины и интерактивные видео [6]. Этот инструмент очень прост в использовании и в создании контента. Его можно добавить на веб-сайт, чтобы улучшить взаимодействие и сделать больше конверсий.

В сервисе Ion Interactive можно использовать один из тысячи полностью настраиваемых шаблонов для создания любого из видов интерактивного веб-контента: от интерактивной инфографики и документов до опросов, викторин и калькуляторов [7, с. 18].

Таблица

Сравнительный анализ сервисов по созданию интерактивного веб-контента

	Функционал	Легкость и понятность в использовании	Быстрота создания контента	Интерфейс	Стоимость
Ceros	Инфографика Электронные книги Журналы Баннерные материалы	8	8	10	~ 130 000 р./мес.
SnapApp	Видеоролики Инфографика Калькуляторы Тесты Викторины	6	7	5	~ 200 000 р./мес.
MapMe	Карты	8	8	8	~ 4000 р./мес., бесплатная демоверсия
Apester	Тесты Опросы Викторины Видео	10	10	9	~ 8000 р./мес., бесплатная демоверсия
Ion Interactive	Инфографика Документы Опросы Викторины Калькуляторы	8	9	9	~ 10000 р./мес., бесплатная демоверсия
Content Tools	Викторины Конкурсы Карты Калькуляторы Инфографика	7	7	9	~ 12000 р./мес., бесплатная демоверсия
Playbuzz	Викторины Видео	10	8	8	~ 50000 р./мес.
Calculoid	Калькуляторы	6	6	8	~ 3000 р./мес., бесплатная демоверсия
Lead Quizzes	Викторины Опросы	7	8	9	~ 6000 р./мес.,
Qzzr	Викторины	8	7	10	~ 3000 р./мес. бесплатная демоверсия

ContentTools — это программное обеспечение для создания интерактивного веб-контента с целью привлечения большего количества потенциальных клиентов. С помощью ContentTools можно создать викторины, конкурсы, интерактивные карты, калькуляторы, инфографику и многое другое [8].

Еще одним инструментом для создания интерактивного веб-контента является инструмент Playbuzz. Это сервис интерактивного веб-контента, который работает для издателей, рекламодателей и создателей контента [9]. С помощью него можно добавить такие интерактивные элементы, как видео и викторины, для того, чтобы сделать веб-контент более интересным и читабельным.

Онлайн-калькуляторы это один из очень простых и эффективных способов привлечения посетителей на сайт и увеличения трафика [10, с. 43]. Сервис Calculoid позволяет создавать интерактивные калькуляторы, которые являются актуальными для пользователей сайта. Одним из видов интерактивного веб-контента являются викторины. Существует достаточно большое количество сервисов для создания викторин. Одним из таких сервисов является сервис LeadQuizzes. Это программа для создания викторин и опросов предназначена для маркетологов и предпринимателей. Программа предоставляет возможность привлечь больше потенциальных клиентов с помощью разработки такого простого, но чрезвычайно эффективного вида интерактивного веб-контента [11, с. 9].

Еще одним инструментом для создания онлайн-викторин является инструмент Qzzr. Этот инструмент стремится помочь компаниям, которым необходимо привлечь потенциальных клиентов. Инструмент имеет такое достоинство, как перенаправление на целевое предложение после прохождения теста посетителем сайта [12, с. 16].

Результаты сравнительного анализа вышеописанных сервисов по созданию интерактивного веб-контента можно увидеть в таблице. В таких колонках, как «Легкость и понятность в использовании», «Быстрота создания контента», «Интерфейс» находятся оценки по шкале от 0 до 10, где 0 – наихудшая характеристика у данных критериев, а 10 – наилучшая.

Таким образом, можно сделать вывод, что наилучшими сервисами по созданию интерактивного веб-контента из представленных выше являются такие сервисы, как Apester и Ion Interactive. Оба этих сервиса достаточно просты и понятны в использовании, с помощью них можно быстро создать интерактивный веб-контент различного вида. Цена у данных сервисов весьма приемлема и соответствует их функционалу.

Литература

1. Прокин А.А., Богатырская В.А., Сергушина Е.С., Лукин М.А. Создание и продвижение интерактивного веб-сайта для коммерческой организации // E-Scio. 2018. № 5(20). С. 39-44.
2. Гудков П.А. Методы сравнительного анализа. Пенза: Изд-во Пенз. гос. ун-та, 2015. 81 с.
3. Симакова С.И., Топчий И.В. Информативно-выразительный потенциал визуального компонента в современных медиатекстах (на примере использования инфографики) // Известия высших учебных заведений. Уральский регион. 2017. № 3. С. 84-89.
4. Boyer J.M., Wiecha C., Akolkar R. Interactive Web Documents // Computer Science – Research and Development. 2012. Т. 27. № 2. С. 127-145.
5. Mwalongo F. Interactive web-based visualization. 2018.
6. De Mattos Fortes R.P., Antonelli H.L., Watanabe W.M. Dynamic Web Content // Web Accessibility. London: Springer, 2019. P. 373-395.
7. Wright T. Learning JavaScript: a hands-on guide to the fundamentals of modern JavaScript. Addison-Wesley, 2012.



8. Knight I. Introduction to Creating Web Content // Connecting Arduino to the Web. Apress, Berkeley, CA. 2018. P. 93-127.
9. Raney A.A., Arpan L., Pashupati K., Brill D. At the Movies, on the Web: An Investigation of the Effects of Entertaining and Interactive Web Content on Site and Brand Evaluations // Journal of Interactive Marketing. 2003. № 17(4). С. 38-53.
10. Haque M.A., Haque S. Web Content Mining – A Tradition and Prospect // Emerging Trends in Management & Information Technology. Edition: I. Publisher: National Press Associates. 2020.
11. Pandey S. Programming Based Interactive Content // Verizon Patent and Licensing. U.S. Patent No. 8,869,196. 21 Oct. 2014.
12. Wuttke H.D., Hutschenreuter R., Sukiennik D., Henke K. Interactive Content Objects for Learning Digital Systems Design // Systems Thinking and Moral Imagination. 2019.

© Андреева А.Ю., 2022

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ:
СЕГОДНЯ И ЗАВТРА, МЕЧТЫ И РЕАЛЬНОСТЬ.
АНАЛИЗ ФИЛОСОВСКИХ ПРОБЛЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

Когда говорят об искусственном интеллекте, то сравнивают его с нашим природным, а что же это такое? Искусственный интеллект (ИИ) - способность компьютера или робота, контролируемого компьютером выполнять задачи, которые присущи только человеку. Интеллектуальные системы, которые всецело стараются подражать человеку, например, двигаясь и разговаривая, рассуждая, делая какие-то нетривиальные умозаключения, обучаясь из предыдущего опыта. Однако, не смотря на все технологические достижения, не появились роботы или другие компьютерные системы, которых было бы трудно отличить от людей. Но все же представим себя в будущем и попробуем предсказать поведение столь прогрессивных технологий.

А здесь уже непросто, начинает возникать целый ряд философских проблем. Можно ли в некоторый момент сказать, что машина близка к человеку? Что вообще в человеке ценим и осознаем, как собственно человека? Может просто разговаривать, помогать нам или выполнять за нас какую-то работу сможет ИИ, основываясь на накопленном опыте человечества? Нужно ли дать ей в этом контексте какие-то особые права?

Остановимся на вопросах идентичности и самосознания живых людей, на том, что именно делает их людьми, так как в случае изобретения довольно мощного ИИ, этот вопрос будет интересовать в первую очередь. Также в этой статье приводится подробное сравнение возможностей ИИ и реального человека. Нужно четко понимать, в чём разница между умной машиной и людьми. Как следствие, изучаются вопросы предоставления прав человека для умных машин и возможные сценарии развития изобретения сверхмощного ИИ. И насколько возможно воссоздать естественный интеллект искусственно.

Для начала нужно определить, а как работает сознание самого человека. Естественный интеллект есть совокупность способностей человека выделять нечто существенное в наличных данных опыта и знания. Кроме того, естественный интеллект есть способность к целеполаганию и отбору знаний. Этот интеллект связан со способностью к оценке знаний, а затем со стремлением к воссозданию «картинного» мышления. Естественный интеллект способен выстраивать социальное действие в условиях быстро изменяющегося социума [4, с. 15].

Естественный интеллект изучается различными науками от физиологии и биохимии до психологии. Люди давно пытаются разгадать загадку строения нашего самого важного органа - мозга.

Информация о человеке на молекулярном уровне хранится в генах и выражается в функциональной анатомии организма и архитектуре его нервной системы. Формирование

интеллекта происходит в процессе всестороннего взаимодействия с окружающим миром. По мере взросления в мозге формируются множественные связи между нейронами, именно они, а не сами нейроны являются структурно-функциональной основой человеческого интеллекта.

А как работают системы на ИИ? Искусственный интеллект представлен в существующей литературе в качестве понятия для обозначения целого комплекса средств, воспроизводящих определённые функции мышления. Заметим, что само происхождение языковой метафоры искусственный интеллект авторы традиционно связывают с употреблением самого слова «интеллект», причём, для обозначения способности человека к постижению чего-либо [4, с. 15].

ИИ подразделяется на сильный (выполняется множество задач, старается во всем походить на человека) и слабый или узкий (выполняет одну задачу, например, играет в шахматы или распознает человеческие лица в толпе). Многие виды любого ИИ базируются на таком понятии, как искусственная нейронная сеть. Для понимания работы ИИ приведем пример работы такой сети. В ней так же, как в естественной сети анализируются входные сигналы, используется сеть нейронов, их объединение. Каждый нейрон отвечает за ответ на какой-то вопрос. Например, в задаче о распознавании цифр, каждый слой нейронов отвечает за характерную черту цифры (палочки, изгибы). Настраивается набор весов, по которым происходит дальнейшее определение. Что-то похожее происходит в человеческом мозге, правда мозг человека состоит из около 100 миллиардов нейронов. Компьютерам такое пока невозможно осуществить. Дело в том, что важны не сами нейроны, а их связи с другими нейронами, которые еще больше.

За последние десятилетия в области ИИ и развитии техники произошел грандиозный прорыв благодаря внедрению и применению искусственных нейронных сетей. Их можно обучить распознавать и предсказывать различную информацию. Они очень похожи на нейронные сети реальных живых существ. Нейроны объединяются в последовательно расположенные слои. Отдельно выделены два крайних слоя - входной и выходной. Через входной слой нейросеть получает информацию, через выходной передаёт результат её обработки. Все промежуточные слои называются скрытыми. Каждый скрытый слой соединён с двумя соседними (предыдущим и следующим) сложной системой связей. В простейшем случае в каждый его нейрон попадают сигналы от каждого нейрона предыдущего слоя, обрабатываются, а затем из него уходят в каждый нейрон следующего слоя. Однако сегодня нейросети содержат более сотни слоев и миллионы узлов, за что получили название «сети глубокого обучения». Такие системы превосходят профессиональных игроков в Го, правильно распознают и классифицируют огромное количество объектов, преобразовывают словесные команды в действия и управляют машинами (знаменитая Алиса).

Хотя для обучения этих сетей требуются миллионы тренировочных испытаний с очень большим количеством образцов, их потрясающие характеристики часто воспринимаются как свидетельство того, что они функционируют по тем же принципам, что и природный мозг. Таким образом, приходим к мысли, а возможно ли живое вне природного происхождения, то есть можно ли будет считать искусственный разум равным человеческому?

Важной характеристикой естественного интеллекта можно назвать самосознания. Самосознание – это способность человека одновременно отображать явления и события внешнего мира и знать о самом сознании. Самосознание интегрирует личность, оно объединяет воедино мысли и чувства человека, его поступки, оценки своего места в обществе и в мире (<https://clck.ru/ep2qd>).

Наверное, это один из самых сложных вопросов для ИИ. Мы можем осознать себя, но вот сможет ли это каким-то образом сделать компьютер? Искусственные нейронные сети сегодня демонстрируют большой потенциал, но все же основываются на изменении параметров и математических счетных операциях. Будучи натренированными на большом количестве входных данных они без проблем определяют человека на картинке или любой другой объект, но все же сознание вещь куда более сложная, чем просто математические расчеты. Человек во многом не понимает механизмы работы своего мозга и сознания.

В научной фантастике часто описывается сценарий оцифровки головного мозга человека. И как же можно будет отличить такую хорошо обученную машину-робота от живого настоящего человека? Будут ли какие-то правила для определения? Даже для примерного моделирования мозга потребуются огромные вычислительные ресурсы, но можно ли будет считать такой оцифрованный мозг «живым»?

Это и есть одна из главных философских проблем: существование сознания вне человека. Сознанием мы считаем возможность человека осознавать, осмыслять свое существование. Но вот будет ли машина осознавать себя? Скорее всего, нет, и на то есть ряд причин. Рассмотрим их далее, а также попробуем дать критерии и сравнить искусственный и естественный интеллект. Увидев разницу, станем лучше понимать суть вопроса.

По сравнению с людьми компьютеры могут обрабатывать больше информации с намного большей скоростью. Например, если человеческий разум может решить математическую задачу за 5 минут, ИИ может решить множество таких задач за минуту. Скорость выполнения большинства механических задач у компьютера будет намного больше, чем у человека, но все же есть ли задачи, справиться с которыми человек сможет быстрее? Определенно, такие задачи есть и их не мало. Среди примеров: осмысление человеком окружающей реальности, создание целостной картины мира, понимание своей принадлежности к природе. Как объяснить такое компьютеру? Пока не понятно способен ли вообще на это ИИ.

ИИ очень объективен в принятии бизнес-решений, автоматизации производства, хозяйственной жизни людей, поскольку он формально анализирует многочисленные собранные данные, которые бывает не под силу осмыслить человеку. Даже при всей точности на решения людей влияют чисто субъективные вещи, которые основаны на личном опыте. Иногда человек даже об этом и не задумывается. Это дает нам как преимущества, так и недостатки. Но все же в большинстве случаев ИИ дает более быстрые, взвешенные решения проблем материальной части человеческого бытия.

Стоит отметить также, что ИИ дает точные результаты, поскольку технологии работают на основе набора запрограммированных правил или любых других технологий,

базирующихся на двоичной логике. Что касается человеческого интеллекта, обычно есть место для «человеческого фактора», постоянно или время от времени возникающих во всех сферах жизни людей. Некоторые детали могут быть упущены в тот или иной момент, что-то недоделано. Однако здесь нужно отметить такой важный момент как сбой в программе. Это довольно частый сценарий для фильмов, где умная машина сходит с ума и уничтожает человечество. От ошибок никто и ничто не застраховано, даже в самом точном и хорошо отлаженном механизме.

Человеческий мозг потребляет намного меньше энергии, чем современные компьютеры (примерно в 10 раз), хотя компьютер запитать гораздо проще, чем человека. Ему не нужно строить клетки и поддерживать их жизнь. Клетка гораздо более сложная система по сравнению с любой деталью компьютера, пусть и самого современного и высокопроизводительного. Все процессы в клетке динамичны, используется очень много различных веществ, которые надо постоянно пополнять и воссоздавать. Иными словами для работы человека нужно очень много всего и постоянно, потребности компьютера не столь многочисленны.

Человеческий интеллект весьма гибок к изменениям в окружающей среде. Это позволяет людям учиться и осваивать различные навыки. ИИ требуется гораздо больше времени, чтобы адаптироваться к новым изменениям. Создать алгоритмы адаптации будет довольно сложно даже на уровне животных. Человеческий интеллект поддерживает многозадачность, о чем свидетельствуют разнообразные и одновременные роли, но ИИ может выполнять намного больше задач одновременно. В этом плане у компьютера может выполняться задач тем больше, чем количество и производительность ядер у процессора. Уникальная способность человека, над которой ИИ предстоит большая работа. Основным механизмом, лежащим в основе конструирующих возможностей человека, является сознание. По мнению Е.Н. Князевой «сознание - эмерджентная, сложноорганизованная и автономная сеть элементов» [1, с. 55].

Как было выяснено ранее, способность к самосознанию пока не доступна для компьютера. Это одно из главных отличий человека от умной машины. Люди естественным образом осознают себя и стремятся понять себя по мере взросления. Один из главных вопросов сейчас, это: «Способна ли вообще машина на это?» Пока ответ на этот вопрос – «нет».

Живя в социуме, люди постоянно взаимодействуют с себе подобными, испытывают различные эмоции. Человек может обрабатывать абстрактную информацию, обладает самосознанием и совершенно по-разному воспринимает действия людей. ИИ еще не научился в должной мере улавливать соответствующие социальные и эмоциональные сигналы, несмотря на наличие ряда исследований направленных на воспроизведение сложных эмоций человека, рассматривающих их как часть интеллекта.

Они известны под общим названием эмоциональный интеллект. Здесь мы упираемся в понятие интеллект. А это тоже важная философская проблема. Вообще кого считать умным и по каким критериям? В разное время люди понимали под разумом, мудростью и ценностями

разное. Очевидно, что разум должен способствовать процветанию, пользе, реализации человека. Кто-то выделял нравственные характеристики человека, кому важна была точность, быстрота и слаженность действий, а есть и те, кому важна только выгода. А зачем нам вообще нужен разум? Разве его предназначение и смысл жизни не об одном и том же? Тут можно обратиться к античной философии. Древнегреческие философы уже тогда пытались дать ответы на совершенно разные вопросы, и во многом они достигли значительных результатов, оставаясь при этом актуальными и по сей день. Для древнегреческого философа Сократа смыслом жизни является мудрость, познание и самопознание (<https://clck.ru/UL4wo>). По мнению Аристотеля, счастье есть смысл и назначение жизни, единственная цель человеческого существования (<https://clck.ru/ep3Ti>). В чем будет заключаться это счастье и что даст самопознание? Сильный ИИ в случае его изобретения тоже будет решать эту проблему. И немаловажный вопрос, в какую сторону он ее решит. От этого будет зависеть судьба человечества в случае изобретения таких технологий.

Предназначение человеческого интеллекта - открытие нового, адаптация к окружающей среде, поскольку он может создавать, кооперировать, внедрять что-то новое для улучшения качества жизни. Что касается ИИ, то его предназначение больше связано с рутинной, долгой и сложной для человека работой, поскольку он эффективно выполняет расчетные задачи в соответствии с тем, как он запрограммирован. Способность выполнять сложные задачи, абстрактное мышление, воля, самокритика, эмоции – всё это пока еще только мечта для ИИ.

Если смотреть на далекие перспективы развития ИИ, то стоит поднять вопрос, который с некоторых пор занимает умы теоретиков от юриспруденции, имеющий самое непосредственное отношение к теме нашего разговора, но в некотором смысле более практический. Суть его заключается в следующем: не предстоит ли нам в не столь отдаленном будущем задуматься над тем, обладают ли компьютеры законными правами и несут ли они ответственность за свои действия. В самом деле, если со временем компьютеры смогут достичь уровня человека (а то и превзойти его) в самых разных областях деятельности, то подобные вопросы неминуемо должны приобрести определенную значимость, в том числе и для философии [3, с. 69].

С одной стороны, ответ очевиден, как могут быть даны права человека существу/созданию, которое не является человеком. Но с другой стороны границы человек/машина в будущем начнут понемногу стираться. Приведу пример. Человекоподобный робот, созданный российской компанией Promobot, начал работу в пермском многофункциональном центре «Мои документы». Это первый в России случай, когда робот-двойник человека заменяет специалиста по оказанию госуслуг, выглядит достаточно убедительно, не знающий человек не сразу заметит в нём подвох. А что будет дальше, если и сейчас такое создание человека весьма неплохо моделирует работу человека?

Их «сознание», а точнее работа является полностью оцифрованным мозгом человека, то есть такой робот будет очень точно повторять все модели поведения человека в теории, конечно. Пока не понятно, как это будет работать, но с точки зрения философии нужно понимать соотношение идеального и материального в ИИ. Будет ли первоначальную роль

играть идеальное начало? Если главенствующую роль будет у идеи, то нам нужно будет действительно задуматься о том, насколько сильный ИИ будет осуществим и убедиться в его безопасности для нас самих [2].

Это компьютерная программа, которая учится и в результате получает социальный опыт, так же как и человек. Пусть она это делает каким-то своим особенным образом, но это будет именно социальный опыт. Очень точно подражает человеку, делает все так же как человек. Тогда будет уже трудно отличить такую машину.

Основой всякого ИИ являются достаточно сложные, но все же просто компьютерные программы, которые работают с данными, накапливают их, а потом воспроизводят, совершенно не осознавая того, что происходит. Возникает другая философская проблема, что же считать человеком? Человеком мы считаем биолого-социальное существо, обладающие отличающимися его от животного чертами, такими как абстрактное мышление, связанная и раздельная речь, самосознание, способность к созданию и открытию нового. Осознание и воля человека, его жизнь в обществе, настоящие эмоции никогда не сравнятся с неживым компьютером. Мы не сможем считать машину равной человеку. Компьютеры не в состоянии убедительно демонстрировать душевные переживания и, уж конечно же, ничего похожего не чувствуют и чувствовать никогда не будут. Соответственно, компьютеры не могут ни обладать правами, ни нести ответственность [3, с. 70].

Они не сделаны из железа, как же можно считать данное изделие человеком? Пожалуй, самый отчаянный аргумент, но все же, он здесь имеет существенное значение. Тут стоит обсудить проблему соотношения идеального и материального бытия человека. Одни полагают материальное продуктом духовного, а другие наоборот. Так вот если материальное есть основа бытия, то нельзя сказать, что машину будут каким-либо образом равны человеку, ведь компьютерные алгоритмы на железе никак не будут равны живой клетке.

Сегодня компьютер, управляемый программами, написанными человеком мы не можем ни в чем обвинить по понятным причинам, ведь это просто инструкция, которой следует машина, не осознавая свои действий. Таким образом, сегодня искусственный интеллект пока не идет в сравнении с естественным интеллектом благодаря таким возможностям человека как осознание, воля, эмоции, способность к творчеству. Именно эти качества будут отличать человека от самой умной машины. Сейчас компьютеры могут обрабатывать данные быстрее, чем мы. Но они не способны мыслить абстрактно, продумывать стратегию, а также использовать мысли и воспоминания, чтобы принимать обоснованные решения или выдвигать творческие идеи. Благодаря этому типу интеллекта мы превосходим машины. В то же время работу этих когнитивных механизмов сложнее всего понять, а значит, и сложнее всего воспроизвести (<https://clck.ru/ep3kc>).

Несмотря на технические и научные достижения мы не можем пока сказать, что ИИ приблизилось к возможностям человека и способен его не заменить. Также нельзя дать машине права человека или какие-то иные особенные права. Применение ИИ сейчас находит в разных сферах жизни общества, но все же пока больше в качестве помощника в делах. Хотя сейчас технические возможности ИИ весьма ограничены, нам стоит более внимательно

отнестись к его осмыслению. И это касается не только науки, которая по мере развития технологий должна следить за их безопасностью для человека, но и глубоких философских вопросов, которыми мы до сих пор задаёмся. Одним из таких, например, является присуждение роботам или другим интеллектуальным системам каких-то особых прав (человеческих как вариант). Это вопрос на самом деле содержит в себе целый комплекс философских проблем, которые человечеству надо будет решать и возможно уже скоро. Будущее предсказать невозможно. Нельзя однозначно сказать к чему приведет изобретение настолько сверхмощного ИИ, которое может сравниться с человеком. До сих пор не до конца знаем, как работает наш мозг, наверное, это и будет ключом к разгадке вопроса о том, возможно ли сознание вне разума.

В заключении приходим к выводу, что пока ИИ заменить человека не может, нужно изучить работу собственного мозга, решить возникающие философские и этические проблемы, возникающие во время проектирования ИИ. В философском контексте также нельзя считать машину живой, ведь она не способна на многие вещи, которые присущи человеку.

Литература

1. Князева Е.Н. Сознание как синергетический инструмент // Вестник международной академии наук (Русская секция). 2008. № 2. С. 55–59.
2. Лабушева Т.М. Сознание вне мозга: миф или реальность? // Исторические, философские, политические и юридические науки, культурология и искусствоведение. Вопросы теории и практики. 2016. № 1(63). С. 95-97.
3. Пенроуз Р. Тени разума: в поисках науки о сознании. Москва–Ижевск: Институт компьютерных исследований, 2005. 688 с.
4. Пушкарёв А.В. Философские основания искусственного интеллекта: Дисс. ... канд. филос. наук. Уфа, 2017. 145 с.

© Антонов Д.Г., 2022

УДК 004.9

Байкова К.Д., Медведева Т.А., канд. техн. наук
Донской государственный технический университет
г. Ростов-на-Дону, Россия

АНАЛИЗ ЭФФЕКТИВНЫХ СПОСОБОВ СОЗДАНИЯ TELEGRAM-БОТОВ ДЛЯ ВНЕДРЕНИЯ В БИЗНЕС-ПРОЦЕССЫ

Telegram – кроссплатформенный мессенджер для мгновенного обмена данными, аудио и видеозвонками, разработанный в 2013 году Павлом Дуровым, создавшим ранее социальную сеть ВКонтакте. Популярность мессенджера обусловлена высокой степенью анонимности, которой удалось добиться благодаря внедрению шифрования данных. Специально для Telegram был создан протокол MTProto, использующий несколько протоколов шифрования. Для шифрования используются алгоритмы RSA и DH при аутентификации и авторизации пользователя и алгоритм AES при отправке сообщений протокола в сеть с ключом, известным только клиенту и серверу. После перехода на новую версию протокола MTProto 2.0, стал использоваться криптографический хеш-алгоритм SHA-256. В 2013 году в Telegram появилась функция «секретных» чатов, сообщения отправленные в этом режиме не расшифровываются сервером и переписка сохраняется только на тех устройствах, на которых создан чат (<https://clck.ru/ebx3a>).

Популярность Telegram значительно выросла в начале 2021 года, после того как другой популярный мессенджер WhatsApp внес изменения в политику конфиденциальности данных. По данным аналитической компании «SensorTower» за январь 2021 года, представленным на рисунке 1, Telegram стал самым популярным приложением в мире, самым скачиваемым в магазине приложений для Android и на четвертом месте в рейтинге для iOS (<https://clck.ru/W2oDu>).



Рис. 1. Лучшие приложения в мире за январь 2021 года по загрузкам

В январе 2021 года аудитория Telegram за все время существования мессенджера достигла 500 млн активных пользователей в месяц, что представлено на рисунке 2 (<https://clck.ru/ebx5s>).

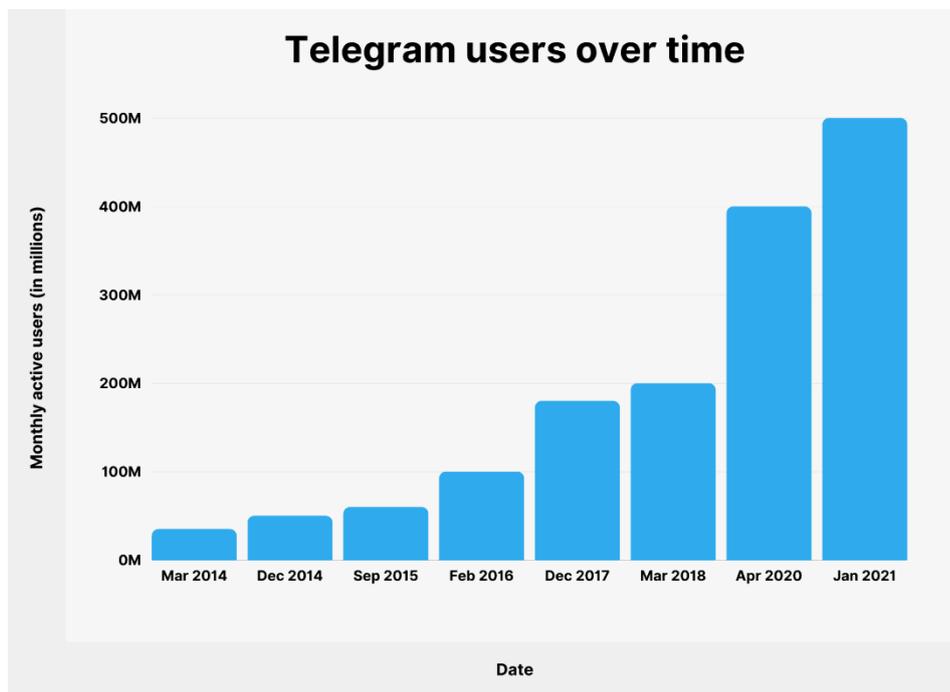


Рис. 2. Количество пользователей Telegram с 2014 по 2021 годы

В 2022 году популярность мессенджера Telegram продолжает расти, в частности в качестве платформы для внедрения чат-ботов. Бот (чат-бот) – это программа, функционирующая по заданному алгоритму и автоматически выполняющая какие-либо действия по заданному сценарию. Такие программы функционируют внутри мессенджеров и могут как отвечать на вопросы, так и самостоятельно задавать их. Для реализации ботов Telegram имеет несколько преимуществ перед другими поддерживающими их созданием мессенджерами:

1. Безопасность.
2. Развитая структура ботов и каналов.
3. Кроссплатформенность.
4. Простота внедрения и использования.
5. Понятный для пользователя интерфейс.

Внедрение ботов в бизнес позволит сократить затраты, увеличить конверсию и улучшить качество обслуживания клиентов [3, с. 33]. Автоматизированные помощники могут эффективно и круглосуточно решать рутинные задачи, помогая экономить время, ресурсы и, как следствие – деньги [1, с. 7]. При этом, необходимо учитывать, что программное обеспечение бота может значительно отличаться в зависимости от его назначения.

Ботов можно классифицировать по нескольким параметрам: по алгоритму, по платформе внедрения и по функционалу. По алгоритму боты делятся на ограниченные,

работающие по заранее написанному сценарию, и саморазвивающиеся, функционирующие на основе нейронной сети, что позволяет сделать беседу более реалистичной. Боты могут быть внедрены в веб-сайты (в качестве онлайн-консультантов), в мессенджеры, в социальную сеть ВКонтакте, в мобильные приложения и в поисковые системы. По функционалу ботов можно разделить на коммуникационные, применяемые только для общения, и функциональные, с помощью которых можно совершить какое-либо целевое действие. Существует так же несколько наиболее востребованных и популярных ботов для бизнеса, которые классифицируют по сценарию использования. Наибольшую популярность имеют:

1. Бот-консультант, осуществляющий общение с клиентом, сокращающий нагрузку операторов-консультантов и затраты на техническую поддержку [4, с. 18].

2. Бот-информатор, распространяющий контент, рекламные и информационные статьи, структурирующий информацию и помогающий быстро найти материалы в соответствии с запросом.

3. Бот-поисковик, выполняющий поиск по заданным параметрам, например @AviaSalesBot, подбирающий авиабилеты для конкретных направлений и дней.

4. Бот-сборщик информации, собирающий данные о потенциальном покупателе, например, номер телефона, электронную почту, ФИО и т.д.

5. Бот для продажи и оформления заказов, с помощью такой программы можно оформить заказ на товар в чате, что помогает снизить нагрузку операторов-консультантов.

Все telegram-боты функционируют по заданному алгоритму, который заключается в том, что команды пользователя передаются в запущенное на сервере программное обеспечение. После обработки полученного запроса на сервере, в соответствии с описанным алгоритмом пользователь получает ответ. Этот цикл повторяется каждый раз, когда бот получает новый запрос. Алгоритм, описывающий цикл взаимодействия бота с пользователем показан на рисунке 3 [2, с. 17].

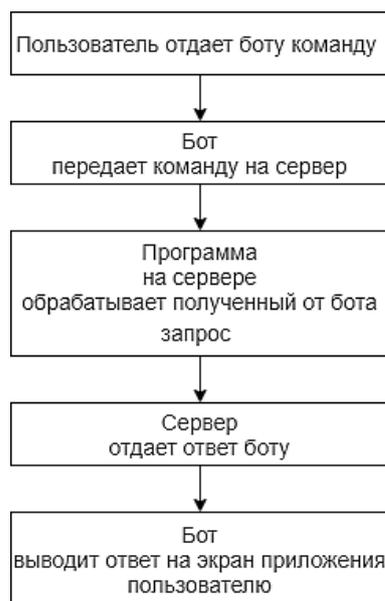


Рис. 3. Алгоритм взаимодействия бота и пользователя

Боты и чат-боты завоевали мировую популярность благодаря возможности создания гибкого функционала, применяющегося во многих сферах. Все существующие технологии для разработки ботов можно разделить на два типа по способу реализации:

1. С помощью готовых инструментов для разработки ботов.
2. Программирование чат-ботов, используя один из языков: Java Script, Python или C#.

Рассмотрим возможности сервисов для создания ботов на примере программы “Manychat”. Это конструктор, не требующий знаний в области программирования, поэтому такой способ можно назвать наиболее простым и наименее затратным по времени. Созданные с помощью “Manychat” Telegram-боты могут: отправлять сообщения и графические материалы подписчикам, поддерживать собственный набор команд, управляться с помощью многоуровневых меню (<https://manychat.manybot.io/>). К минусам такого сервиса можно отнести ограниченный доступный функционал, так как некоторые задачи требуют отдельной реализации и не предусмотрены программой.

Для программной реализации ботов популярностью пользуется язык программирования Java Script и Node.js. Node.js – серверная платформа, которая работает на движке V8. Она является перспективным и мощным инструментом для работы с кодом, позволяет превратить язык Java Script из узкоспециализированного в язык общего назначения, не требуя при этом больших усилий разработчика. Платформу разработал Райан Даль в 2009 году. Node.js стала результатом его двухлетних экспериментов над разработкой серверных веб-компонентов. В основе Node.js лежит событийно-ориентированное и реактивное программирование с неблокирующим вводом/выводом. К преимуществам Node.js можно отнести:

1. Возможность взаимодействия JavaScript с устройствами ввода-вывода через свой API, который написан на C++.
2. Возможность подключать внешние библиотеки, которые написаны на разных языках, и обеспечивать вызов к ним из JavaScript кода. Использование таких библиотек значительно упрощает и ускоряет работу с кодом.
3. Возможность использовать Node.js не только как веб-сервер, но и как разработчика десктопных оконных приложений.

Для создания ботов популярными являются библиотеки “Telegraph” и “nedb”. “Telegraph” помогает удобнее работать с кодом для реализации функционала и кнопок бота, а “nedb” отвечает за автоматическую генерацию базы данных для хранения необходимой информации.

Создание бота в Telegram подразумевает взаимодействие с @BotFather. Это технология, позволяющая создать и зарегистрировать своего уникального бота в мессенджере Telegram. Взаимодействие осуществляется с помощью команд. Так, с помощью команды /newbot запускается процесс создания нового бота, для этого необходимо придумать имя, которое будет отображаться в чатах и контактах. Затем @BotFather просит придумать уникальный username, он должен быть написан буквами латинского алфавита, оканчиваться на «bot» и может включать цифры и символ подчеркивания. После регистрации уникального имени

бота пользователь получает сообщение с токеном, который необходим для дальнейшей работы с API. С помощью @BotFather также можно настроить аватар или логотип и описание бота.

Такая информационная технология как бот, разработанный на базе приложения-мессенджера, является универсальным и удобным методом для развития аккаунта. Боты выполняют широкий спектр задач: улучшение сервиса, реклама компании или продукта, распространение развлекательного контента и другие. Кроме этого, они могут быть универсальными помощниками на этапе развития и продвижения аккаунта, например, напоминать о публикации контента. Основным преимуществом ботов, функционирующих в Telegram, является их кроссплатформенность, так как мессенджер работает практически на всех видах устройств. Бот, который управляется программным обеспечением, может быть настроен на решение практически любой задачи, что делает технологию популярной и способствует её дальнейшему развитию.

Литература

1. Акулич М. Чат-боты и маркетинг. Екатеринбург: Ridero, 2022. 172 с.
2. Байкова К.Д, Медведева Т.А. Анализ и разработка функционального Telegram-бота // Молодой исследователь Дона. 2021. № 6(33). С. 15-20.
3. Гогохия И. Добавь клиента в друзья. Продвижение в Telegram, WhatsApp, Skype и других мессенджерах. М.: Эксмо, 2018. 230 с.
4. Джанарсанам С. Разработка чат-ботов и разговорных интерфейсов. М.: ДМК Пресс, 2019. 340 с.

© Байкова К.Д., Медведева Т.А., 2022

УЯЗВИМОСТИ БИОМЕТРИЧЕСКИХ СИСТЕМ

Биометрическая система уязвима для различных типов атак, которые могут поставить под угрозу безопасность, обеспечиваемую системой, что приведет к ее сбою. Все атаки можно разделить на два основных типа:

– атаки с нулевым усилием. Биометрические характеристики злоумышленника могут быть достаточно похожи на законно зарегистрированного пользователя, что приводит к ложному совпадению и нарушению безопасности системы. Это событие связано с вероятностью случайного совпадения степени сходства между шаблонами, полученными от разных лиц;

– атаки злоумышленников. Это относится к возможности того, что злоумышленник сможет выдать себя за зарегистрированного пользователя, используя физический или цифровой артефакт законно зарегистрированного пользователя. Человек может также намеренно манипулировать своими биометрическими характеристиками, чтобы избежать обнаружения автоматизированной биометрической системой.

Атаки с нулевым усилием. Какова вероятность того, что биометрические данные, полученные от двух разных людей, будут достаточно схожими? Этот вопрос приводит к проблеме индивидуальности в биометрии. Индивидуальность определенного биометрического признака является функцией межклассового сходства и внутриклассовой изменчивости, связанной с этим признаком. Чтобы решить эту проблему, можно смоделировать источник, генерирующий биометрический сигнал, или смоделировать параметры, составляющие шаблон, т. е. набор функций. Проблема индивидуальности, в контексте, скажем, отпечатков пальцев, может быть сформулирована многими различными способами в зависимости от того, какой из следующих аспектов проблемы рассматривается:

1) определите вероятность того, что у любых двух (или более) индивидуумов могут быть достаточно похожие отпечатки пальцев в данной целевой популяции;

2) учитывая образец отпечатка пальца, определите вероятность нахождения достаточно похожего отпечатка пальца в целевой популяции;

3) учитывая два отпечатка с двух разных пальцев, определите вероятность того, что они достаточно похожи. Научная основа для сравнения отпечатков пальцев может установить верхнюю границу производительности систем отпечатков пальцев.

Учитывая схему представления (например, распределение деталей) и меру сходства (например, сопоставление строк), существует два подхода для определения индивидуальности отпечатков пальцев. При эмпирическом подходе собираются репрезентативные образцы отпечатков пальцев, и с использованием типичного устройства для сопоставления отпечатков, в котором точность сопоставления образцов указывает на уникальность отпечатка пальца по отношению к устройству для сопоставления. Однако

существуют известные проблемы (и затраты), связанные со сбором репрезентативных образцов. Кроме того, даже если для эмпирической оценки индивидуальности отпечатков пальцев используется большая база данных отпечатков пальцев, такая как база данных ФБР (FBI), которая содержит около 600 миллионов отпечатков пальцев (десять отпечатков примерно 60 миллионов человек) (<https://clck.ru/ep6G3>), потребуется примерно 169 лет, чтобы сопоставить все отпечатки пальцев в базе данных друг с другом с помощью процессора со скоростью один миллион совпадений в секунду. В рамках теоретического подхода к оценке индивидуальности моделируются все реалистичные явления, влияющие на межклассовые и внутриклассовые вариации отпечатков пальцев. Учитывая метрику подобия, можно было бы теоретически оценить вероятность ложного соответствия. Теоретические подходы часто ограничены степенью соответствия предполагаемой модели действительности.

Общее количество степеней свободы пространства шаблонов (например, пространство конфигурации деталей) напрямую не связано с различимостью различных шаблонов (например, деталей с разных пальцев). Эффективная оценка информации может быть достигнута только за счет учета внутриклассовых различий. Существует несколько источников изменчивости множественных отпечатков пальца: неравномерный контакт с датчиком, невозпроизводимый контакт, непоследовательный контакт и характеристики изображения. Эта изменчивость в множественных отпечатках пальца проявляется как:

- 1) обнаружение ложных мелочей или отсутствие подлинных мелочей;
- 2) смещение или дезориентация (также называемая деформацией) подлинных мелочей;
- 3) трансформация типа мелочей (соединительная двусмысленность).

Однако разработка совпадений с учетом этих внутриклассовых различий может привести к значительному увеличению вероятности ложных соответствий между точками мелочей.

Атаки злоумышленника. Биометрические данные зачастую не являются «секретом». Физические характеристики, такие как снимки лица и отпечатки пальцев, могут быть тайно получены от человека (например, скрытое получение изображений лица или снятие скрытых отпечатков с объекта) для создания цифровых или физических артефактов, которые затем могут быть использованы для подделки личности законно зарегистрированного лица. Помимо этого, существуют и другие атаки, которые могут быть запущены против приложения, ресурсы которого защищены с помощью биометрии:

- 1) обход. Злоумышленник может обманным путем получить доступ к системе, обойдя биометрическое сопоставление, и просмотреть конфиденциальные данные, например, такие как медицинские записи, относящиеся к законно зарегистрированному пользователю. Помимо нарушения конфиденциальности зарегистрированного пользователя, злоумышленник может изменять конфиденциальные данные, включая биометрическую информацию;

- 2) отказ. Законный пользователь может получить доступ к возможностям, предлагаемым приложением, а затем заявить, что злоумышленник обошел систему. Банковский сотрудник, например, может изменить финансовые записи клиента, а затем

отрицать ответственность, утверждая, что злоумышленник, должно быть, подделал его биометрические данные и получил доступ к записям;

3) сговор. Лицо с привилегиями суперпользователя (например, администратор) может намеренно изменять параметры биометрической системы, чтобы разрешить вторжение злоумышленника, с которым он находится в сговоре;

4) принуждение. Самозванец может заставить законного пользователя (например, под воздействием физического насилия) предоставить ему доступ к системе;

5) отказ в обслуживании (DoS-атака). Злоумышленник может перегружать системные ресурсы до такой степени, что законным пользователям, желающим получить доступ, будет отказано в обслуживании. Например, сервер, обрабатывающий запросы на доступ, может быть «засыпан» большим количеством поддельных запросов, тем самым его вычислительные ресурсы будут перегружены и это будет препятствовать обработке действительных запросов.

К. Ратха и др. [1] определили несколько различных уровней атак, которые могут быть реализованы против биометрической системы. Эти атаки предназначены либо для обхода безопасности, которая обеспечивается системой, либо для предотвращения нормального функционирования системы:

1) поддельный биометрический признак, например, искусственный палец, может быть представлен на датчике;

2) незаконно перехваченные данные могут быть повторно отправлены в систему;

3) средство извлечения функций может быть заменено программой «троянского коня», которая создает заранее определенные наборы функций;

4) законные наборы функций могут быть заменены синтетическими наборами функций;

5) сопоставитель может быть заменен программой «троянского коня», которая всегда выдает высокие баллы, тем самым нарушая безопасность системы;

6) шаблоны, хранящиеся в базе данных, могут быть изменены или удалены. В качестве альтернативы в базу данных могут быть введены новые шаблоны;

7) данные в канале связи между различными модулями системы могут быть изменены;

8) окончательное решение, выводимое биометрической системой, может быть отменено.

Процесс реализации данных атак представлен на рисунке.

Безопасность шаблонов является важным фактором при разработке биометрической системы. Можно выделить несколько факторов, которые могут повлиять на целостность шаблона:

1) случайное повреждение шаблона из-за сбоя системы (сбой оборудования);

2) преднамеренное изменение зарегистрированного шаблона злоумышленником;

3) замена действительного шаблона поддельным шаблоном с целью ограничения функциональности системы.

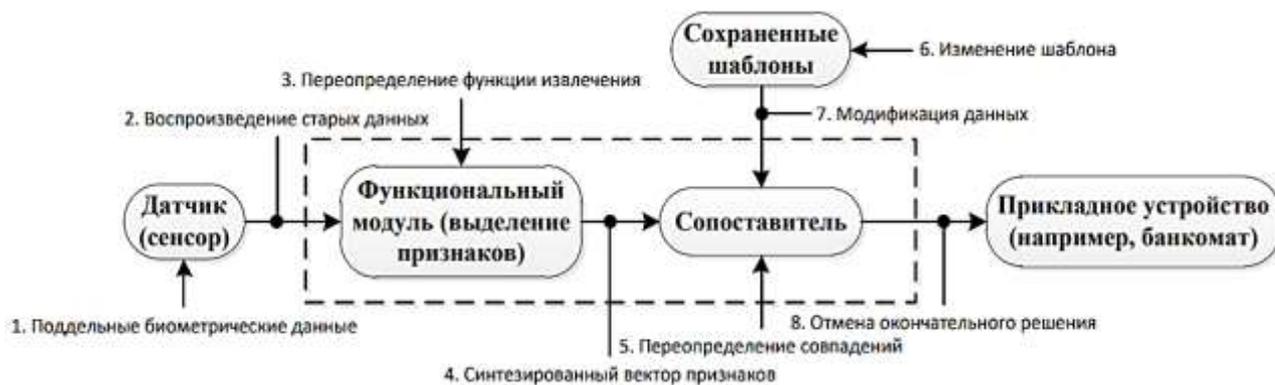


Рис. Процесс реализации атак на биометрическую систему

Шаблон представляет собой набор основных функций, которые суммируют биометрические данные (сигнал) человека. Из-за своей компактности обычно предполагается, что шаблон не может быть использован для получения полной информации об исходном биометрическом сигнале. Кроме того, поскольку шаблоны обычно хранятся в зашифрованном виде, существенно сложно расшифровать и определить содержимое сохраненного шаблона (без знания правильных ключей дешифрования). Таким образом, традиционно алгоритмы генерации шаблонов рассматривались как односторонние алгоритмы. Однако в недавней литературе были представлены методы, которые противоречат этим предположениям. Изображение лица может быть восстановлено из шаблона лица с помощью «Атаки восхождения на холм» (<https://clck.ru/argLR>). Итеративная схема используется для восстановления изображения лица с использованием системы проверки лица, которая выдает результаты совпадения. Учеными был разработан генератор синтетических шаблонов (STG), который также использует «Атаку восхождения на холм» для определения содержимого целевого шаблона деталей (<https://clck.ru/argLR>).

В литературе было предложено несколько методов защиты биометрических шаблонов от раскрытия важной информации. Чтобы предотвратить успешную конвергенцию «Атаки восхождения на холм», Колин Сутар предложил использовать грубо квантованные результаты совпадений с помощью сопоставителя (<https://clck.ru/ep6P3>). Однако, все еще возможно оценить неизвестное зарегистрированное изображение, хотя количество итераций, необходимых для сходимости, сейчас значительно выше. «Атаку восхождения на холм» можно предотвратить, если биометрическая система прервет процесс сопоставления при обнаружении нескольких (скажем, 3) неудачных попыток. Было предложено использовать невидимую хрупкую технику нанесения водяных знаков для обнаружения областей на изображении отпечатка пальца, которые были подделаны злоумышленником [2]. В предлагаемой схеме используется процедура хаотического смешивания для преобразования визуально воспринимаемого водяного знака в текстурированное изображение случайного вида, чтобы сделать его устойчивым к атакам. Это «смешанное» изображение затем встраивается в изображение отпечатка пальца. Авторы показывают, что наличие водяного знака не влияет на процесс извлечения объектов. Кроме того, исходный «несмешанный» водяной знак может быть восстановлен с помощью обратного отображения. Использование

водяного знака также обеспечивает возможность защиты авторских прав путем определения происхождения необработанного изображения отпечатка пальца.

Джейн и Улудаг предлагают использовать принципы стеганографии для сокрытия биометрических данных (например, собственных коэффициентов изображения лица) в изображениях хоста (например, отпечатки пальцев) [3]. Это особенно полезно в распределенных системах, где необработанные биометрические данные, возможно, придется передавать по незащищенному каналу связи. Встраивание биометрических данных в безобидное изображение хоста предотвращает доступ подслушивающего к конфиденциальной информации шаблона. Кроме того, встроенные данные существенно не изменяются, когда изображение хоста подвергается серьезному вмешательству, такому как обрезка. Авторы также обсуждают новое приложение, в котором черты лица пользователя (т. е. собственные коэффициенты) встроены в изображение отпечатка пальца хоста (пользователя). В этом случае изображение отпечатка пальца с водяным знаком человека может храниться на смарт-карте, выданной этому человеку. На сайте контроля доступа отпечаток пальца человека, владеющего картой, сначала сравнивается с отпечатком пальца, присутствующим на смарт-карте. Собственные коэффициенты, скрытые в изображении отпечатка пальца, затем могут быть использованы для восстановления лица пользователя, тем самым служа вторым источником аутентификации. Перед шифрованием необработанных биометрических данных или шаблона следует применить методы нанесения водяных знаков и стеганографии. Стоит обратить внимание на то, что шифрование основано на использовании трудно вычисляемых секретных ключей для защиты биометрической информации. С другой стороны, принципы водяных знаков и стеганографии защищают данные, даже если этот секретный ключ скомпрометирован и биометрические данные расшифрованы. Поэтому поставщикам биометрических данных необходимо использовать оба этих метода в сочетании, чтобы повысить конфиденциальность хранящейся информации. Поскольку биометрические данные человека не могут быть легко заменены (в отличие от паролей и PIN-кодов), скомпрометированный шаблон будет означать потерю личности пользователя.

Ратха и др. [4] предлагают использовать функции искажения для генерации биометрических данных, которые при необходимости могут быть отменены. Они используют функцию необратимого преобразования, которая искажает входной биометрический сигнал (например, изображение лица) перед извлечением объекта или, альтернативно, изменяет сам извлеченный набор объектов (например, мелкие детали). Когда сохраненный шаблон скомпрометирован, текущая функция преобразования заменяется новой функцией, тем самым «отменяя» текущий (скомпрометированный) шаблон и создавая новый. Это также позволяет использовать один и тот же биометрический признак в нескольких различных приложениях, просто используя функцию преобразования для конкретного приложения. Однако неясно, как можно было бы обеспечить, чтобы биометрическая различимость не ухудшалась в преобразованной области. Линнарц и Туилс [5] предложили использовать функции защиты для обеспечения безопасности биометрических шаблонов пользователя от

неправильного использования администратором биометрической системы. Авторы достигают этого, используя функции дельта-сжатия и выявления эпсилон для предварительной обработки биометрических данных, полученных от физического лица. Эти функции затрудняют администратору вычислительную обработку исходных данных пользователя. Хотя для повышения безопасности пользовательского шаблона было предложено несколько методов, также необходимо будет разработать правительственные нормативные акты, чтобы решить проблему конфиденциальности шаблона. Например, вопросы, связанные с обменом биометрическими шаблонами между учреждениями (например, поставщиками медицинских услуг и правоохранительными органами) и выводом личной информации о зарегистрированном пользователе (например, склонен ли этот человек к диабету или нет) с биометрическими данными, необходимо бороться путем создания соответствующей правовой базы.

Таким образом, биометрия представляет собой важные технические, политические и системные проблемы, которые необходимо решить, поскольку у этой технологии нет замены для решения многих важнейших проблем информационной безопасности. Учитывая недавние правительственные мандаты на национальное и международное использование биометрии для выполнения важнейших социальных функций, необходимо развивать базовые биометрические возможности и интегрировать их в практические приложения. Поскольку биометрические данные не могут быть легко переданы, потеряны или подделаны, результирующая безопасность более надежна, чем современные системы паролей, и не обременяет конечного пользователя запоминанием длинных криптографически надежных паролей. Доступ системного администратора на основе биометрических данных к конфиденциальной информации пользователя обеспечивает эффективную подотчетность.

Литература

1. Болл Р.М., Коннел Дж.Х., Панкантн Ш., Ратха Н.К., Сеньор Э.У. Руководство по биометрии. М.: Техносфера, 2007. 367 с.
2. Нуруллоев Ф.Н. Защита отпечатков пальцев в системах биометрической аутентификации с использованием методов криптографии и водяных знаков // *Universum: Технические науки*. 2020. № 10-1 (79). С. 40-42.
3. Джейн А.К., Улудаг У. Соккрытие биометрических данных, перевод IEEE, Т. 25, № 11. – С. 1493-1498. – ноябрь 2003;
4. Ратха Н., Коннелл Дж., Болл Р. Повышение безопасности и конфиденциальности в системах аутентификации на основе биометрии // *IBM Syst.* 2001. Т. 40. № 3. С. 614-634.
5. Линнарц Д., Туилс П. Новые функции защиты для повышения конфиденциальности и предотвращения неправильного использования биометрических шаблонов. Гилфорд, Великобритания. 2003.

© Бакулин М.А., 2022

УДК 004

Бедняк С.Г., Захарова О.И., Федулова А.А., Кузнецова А.А.
Поволжский государственный университет
телекоммуникаций и информатики
г. Самара, Россия

ИСПОЛЬЗОВАНИЕ ЧАТ-БОТОВ

В наше время проблема виртуального общения актуальна как никогда. Быстрый доступ к информации и обмен ею в большом количестве, возможность дистанционного обучения, необходимость поддержки клиентов и партнеров по бизнесу - все это требует определенного подхода к реализации и способствует появлению на рынке виртуальных помощников, позволяющих управлять всем этим потоком данных с помощью простых команд, понятных пользователю, не требующих особых навыков. Для решения этих вопросов стали все чаще использовать чат-ботов в мессенджерах, которые просты в использовании и позволяют создать необходимый индивидуальный функционал.

Понятие «чат-бот» исходит от двух английских слов: to chat — разговор в сети Интернет и bot (robot) — сокращенно робот, что явно дает понять человеку — это роботы, предназначенные для автоматической коммуникаций с пользователями в сети Интернет и выполняющие действия согласно сценарию разработчика или заказчика. На сегодняшний день технологии чат-ботов находятся в области пика завышенных ожиданий и зачастую применяются не в силу необходимости, а потому что это тренд, и они не требуют каких-либо существенных затрат [1, с. 159]. Несомненно, чат-боты можно внедрить в основные мессенджеры. Особой популярностью пользуются следующие платформы: Telegram, Facebook Messenger, Slack, в России ещё и социальная сеть Вконтакте.

В настоящее время все большее количество организаций и компаний, в том числе некоммерческих и бюджетных, стремятся внедрить технологию чат-ботов в свои бизнес-процессы, что может оптимизировать процессы сбора информации и позволит сократить штат работников [2, с. 85].

Чат-бот — это удобный и многофункциональный помощник, который не только экономит большую рабочую времени и поможет быстрее реагировать на запросы клиентов, но и послужит обычному пользователю мессенджеров развлечением, обучением и маленьким помощником. С развитием информационных технологий прогрессирует рост интернет - мошенничества, при котором происходит кража конфиденциальной и личной информации пользователей за счет ненадежных паролей. Киберпреступник может взломать аккаунт с помощью автоматического подбора паролей, также используется сайт, похожий на социальную сеть внешне, но с измененным адресом, либо пользователя заманивают на сайт, который предлагает ввести личные данные. Для того чтобы защитить личную информацию нужен надежный пароль, но многие пользователи используют короткий пароль, составленный из букв одного регистра и без цифр, поэтому его легко взломать. А в его создании надежного пароля может помочь специальный чат-бот с такой функцией, встроенный прямо в их привычную социальную сеть или месенджер (рис. 1). Если не

использовать современные технологии для создания паролей, то пользователь может стать жертвой кибератак и подвергнет себя риску кражи идентификационных данных.

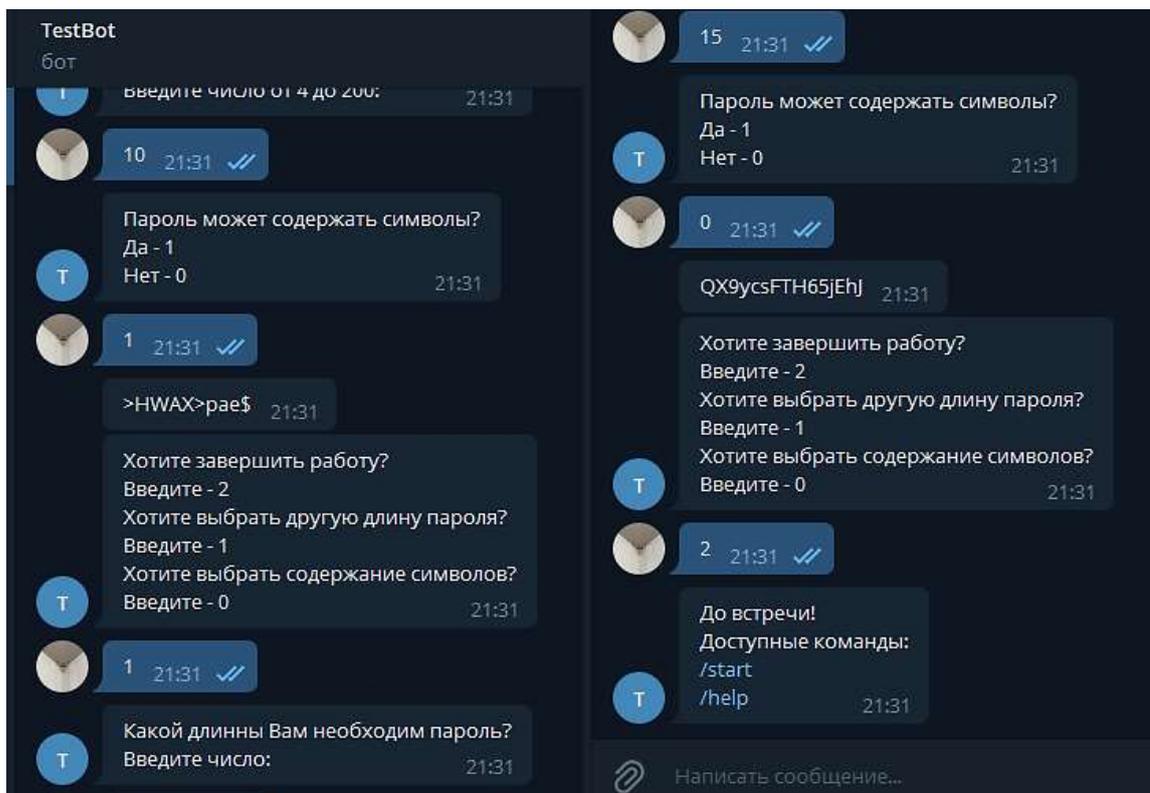


Рис. 1 Пример работы с чат-ботом в социальной сети “Telegram”

Кроме того, чат-боты выполняют множество полезных функций по исполнению рутинных операций, поиску информации, объединению данных, работе с клиентурой. Чат-бот как виртуальный собеседник имеет базу знаний, которая представляет собой наборы возможных вопросов пользователя и соответствующих им ответов. Наиболее распространенными вариантами для получения нужного ответа являются ключевые слова, совпадение фразы, совпадение контекста [3, с. 195].

Помимо всего, эти, буквально, карманные помощники позволяют оптимизировать в работе такие процессы как: бронирование переговоров, информирование сотрудников о датах отпуска, расписание корпоративного транспорта, сроки зарплаты и многое другое. Преимущества от разработки чат-бота для бизнеса распространяются как на внутренние бизнес-процессы компании, так и на индустрии в целом. Чат-ботов можно разделить на несколько видов:

Поддержка. Техническая поддержка осуществляется круглосуточно, отвечает оперативно и без задержки, если нет ответа на нераспространенные вопросы клиентов, то они перенаправляют человека к менеджеру, также при возникновении проблем помогают в вопросах использования продукта или услуги.

Ассистирование. Чат-бот связывается с другими платформами и помогает решать мелкие, но нужные пользователю задачи, кроме этого устанавливает напоминания и собирает статистику, ищет информацию, планирует рабочий день, напоминает о встречах.

Продажа товаров. Реализует продажи в мессенджере или социальной сети, консультирует покупателей и подбирает нужные товары, уведомляет о статусе заказа, принимает оплату и сообщает об акциях и скидках.

Организация досуга. Бот может присылать смешные картинки, видео, музыку, цитаты по запросу, переводить тексты. Кроме этого можно создать небольшую онлайн-игру, в которой победители получают фирменные стикеры или другие мини-подарки.

Еще одно преимущество - пользователям не нужно скачивать приложение, чтобы использовать чат-бот, ведь он находится в мессенджере, которым они и так пользуются, и не занимает память на телефоне. Для создания чат-бота необходимо не только иметь специальный аккаунт в мессенджере, но и написать код программы на одном из языков программирования, где указать сформированный для бота токен, который является «ключом доступа», и использовать библиотеки для работы с API. API — это набор функций. Это может быть одна функция, а может быть много. С помощью него системы взаимодействуют между собой. Можно сказать, что API отвечает на вопрос “Как ко мне, к моей системе можно обратиться?”, и включает в себя три параметра:

- саму операцию, которую нужно выполнить,
- данные, которые поступают на вход,
- данные, которые оказываются на выходе (контент данных или сообщение об ошибке).

Вызвать API можно как напрямую, так и косвенно. При использовании способа напрямую пользователь работает с функционалом API.

1. Система вызывает функции внутри себя

Самый простой способ, при котором разные части программы связываются между собой на программном уровне, а именно на уровне API.

2. Система вызывает метод другой системы

Одна система через API хочет получить метод другой системы, кроме этого она может получить информацию или отправить данные в ту систему.

3. Человек вызывает метод

Данный способ используется тестировщиками, чтобы ускорить работу и устранить ошибки. Может применяться без графического интерфейса, если система предоставляет API.

При использовании косвенного способа вызов происходит при участии другой функции или другого API, то есть происходит взаимодействие с API браузера, но применяется с использованием графического интерфейса.

1. Пользователь работает с GUI. Этот способ предполагает связь пользователя с вызовом API, для этого используется пользовательский интерфейс.

По некоторым зарубежным опросам (рис. 2):

- 40% миллениалов говорят, что пользуются чат-ботами ежедневно;
- 35% потребителей хотят, чтобы большее количество компаний использовали чат-боты;

- 55% респондентов говорят, что предпочли бы получить моментальный ответ на простой вопрос от чат-бота;

- 21% респондентов считают чат-бот самым простым способом взаимодействия с брендом.

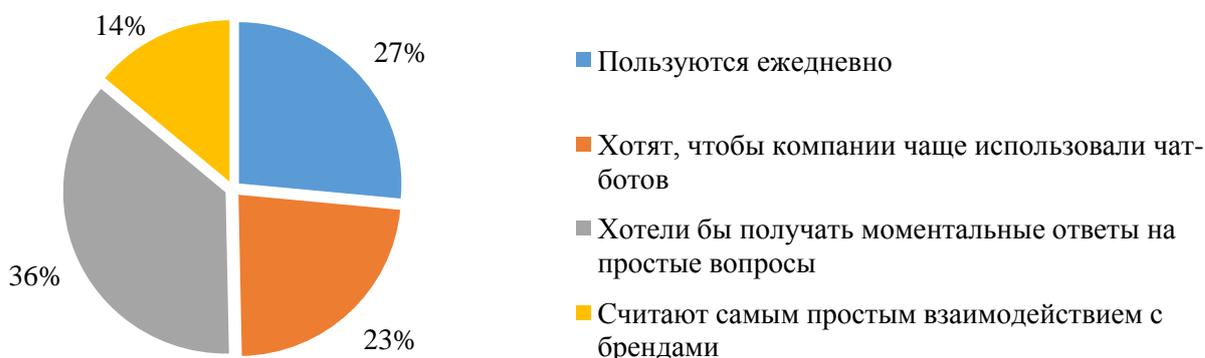


Рис. 2. Мнение о чат-ботах за рубежом

Подводя итог вышеизложенного, можно сделать вывод о полезности использования ботов не только по работе, но и в повседневной жизни. Чат-боты – это полезные современные средства, позволяющие с помощью языка программирования, автоматизировать очень многие рутинные процессы, а также простые в использовании помощники по поиску и управлению информацией.

Литература

1. Екатериничев А.Л., Антоненко Н.А., Бабаев А.Б., Наташкина Е.А. Области эффективного применения чат-ботов // Информационные технологии в экономике и управлении: Сб. материалов IV Всероссийской научно-практической конференции (с международным участием) (г. Махачкала, 11–12 ноября 2020 года). Махачкала, 2020. С. 158-162.

2. Сметкина О.М., Травин Д.Н. Использование чат-ботов в качестве средства оптимизации бизнес-процессов // Цифровая конвергенция в экономике и управлении: Сб. научных трудов. СПб., 2020. С. 83-91.

3. Тарасова Н.С., Сергеева Н.Ю. Использование чат-ботов в повседневной жизни // Вестник современных исследований. 2017. № 12-1(15). С. 195-197.

© Бедняк С.Г., Захарова О.И., Федулова А.А., Кузнецова А.А., 2022

РАЗРАБОТКА КОМПЬЮТЕРНОЙ ЛОГИЧЕСКОЙ ИГРЫ «ДЕНЬ СТУДЕНТА»

В настоящее время компьютерные игры являются неотъемлемой частью жизни молодежи. Игры набирают все большую популярность, в том числе и логические. Логические игры используются для развития и развлечения. Существуют целые компании, основной целью которых является разработка логических игр. Целью работы — разработка логической игры «День студента».

В соответствии с этой целью были поставлены следующие задачи:

- 1) изучить научную литературу и интернет-ресурсы по данной теме;
- 2) рассмотреть классификацию компьютерных игр;
- 3) проанализировать возможности современных программных продуктов для разработки компьютерных игр;
- 4) разработать компьютерную логическую игру.

Компьютерные игры — это программа, служащая для организации игрового процесса для персональных компьютеров, планшетов или других мобильных телефонов. Самая первая компьютерная игра появилась в начале 1940 года, когда Эдвард Кондон выпустил для игры первый игровой автомат Nimatron. С тех времен компьютерные игры пережили эволюцию, превратившись из простеньких игр, в «роскошные» создания с великолепной графикой. В настоящее время существует множество жанров и разновидностей компьютерных игр: логические, симуляторы, ролевые, стратегии и многие другие. Единственное что их объединяет то, что в них играют посредством компьютера. С появлением компьютера, появились компьютерных игр, ставшие неотъемлемой частью человека. Игры с одной стороны способны развивать и обучить человека, с другой стороны замедление развития и отрицательно влияет на физическое развитие. Изучив материалы, можно сформулировать следующие достоинства и недостатки компьютерных игр.

Достоинства:

- Развивают творческое мышление и способности.
- Развивают логику, мышление и память.
- Способствуют овладению персональным компьютером.
- Воспитывают внимательность, сосредоточенность, целеустремленность.
- Помогают овладеть в быстром темпе чтением и письмом.
- Тренируют внимание и память.
- Развивают быстроту действий и реакций.

Недостатки:

- Оказывают отрицательное влияние на физическое развитие.
- Повышают состояние нервозности.

- Повышают страх стремления в не достижения победы.
- Содержание некоторых игр снижает интеллектуальную активность.
- При длительном использовании ухудшают зрение.

Несмотря на недостатки, компьютерные игры крайне востребованы и стали обыденной частью жизни многих людей. Важно лишь правильно выбирать для себя подходящие игры и верно дозировать количество времени, проведенного у монитора.

Головоломки (логические игры) — жанр компьютерных игр, в котором необходимо решать логические задачи, посредством задействования логики, стратегии и интуиции. К головоломкам, или логическим (иногда их называют еще классическими) играм, относятся игры со словами и настольные игры: шахматы, шашки, нарды, крестики-нолики, разнообразные пасьянсы, викторины [1]. Эти игры заставляют игрока преодолевать препятствия, используя ум, а не силу. Они развивают мышление, быстроту реакции, умение ассоциативно думать, улучшают память, координацию движения, также повышают внимательность, усидчивость.

При разработке любой компьютерной игры, в том числе и логической, надо выбрать программу. В современном мире существует множество программных продуктов для создания приложений. Каждая программа по-своему уникальна. Для разработки выбрана среда Microsoft Visual Studio — это программа, входящая в состав Microsoft Office, и представляющая собой интегрированную среду разработки. Она предлагает пользователю разрабатывать приложения на разных платформах с графическим интерфейсом на разных языках (рис. 1). Visual Studio – является самой востребованной средой для разработки приложений.

Visual Studio может создавать собственный код или управляемый код. Она также включает поддерживающий компонент завершения кода (IntelliSense) и рефакторинг кода, для улучшения дизайна и реализации программного обеспечения. Позволяет редактору кода, отладчику поддерживать практически все языки программирования. Встроенные языки C#, VB.NET, F# и C++/CLI, также поддержка Python, Ruby и других (<https://clck.ru/Nj4w9>). Для создания приложения используется язык программирования C#.

При запуске программы появляется стартовое окно, при нажатии на кнопку, щелчком мыши, «Играть», запускается сама игра. Суть игры в том, что участник отвечает на вопросы различной стоимости. Правильные ответы на вопросы оцениваются от 10 до 40 баллов. В случае правильного ответа баллы начисляются на счет игрока. В случае неверного ответа баллы снимаются со счета игрока. Игра продолжается до тех пор, пока не будут разыграны все вопросы. Выбор вопроса осуществляется щелчком мыши в игровом поле на определенном числовом значении. После появляется форма с вопросом. Выбор ответа и возвращение к полю осуществляется щелчком по кнопке с выбранным ответом.

Программа запускается с помощью файла Игра_День студента.exe. После запуска появится стартовое окно (рис. 1).

На данной форме по нажатию на кнопку «Играть» появляется игровое поле (рис. 2).

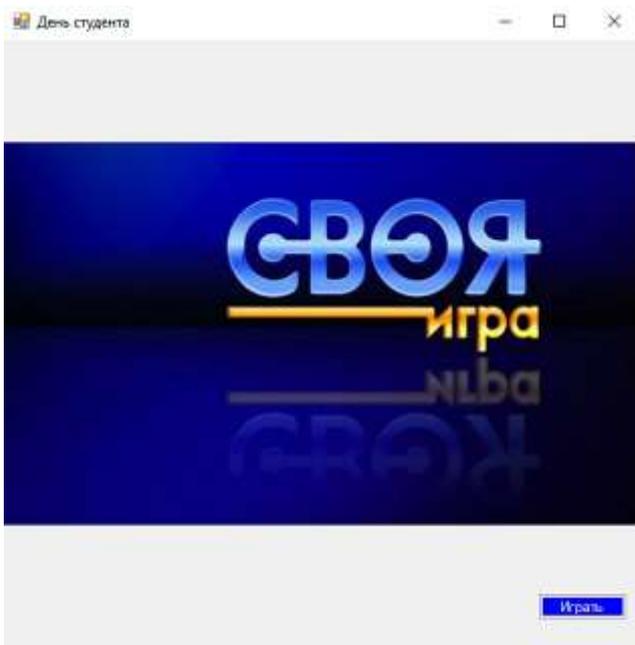


Рис. 1. Стартовое окно

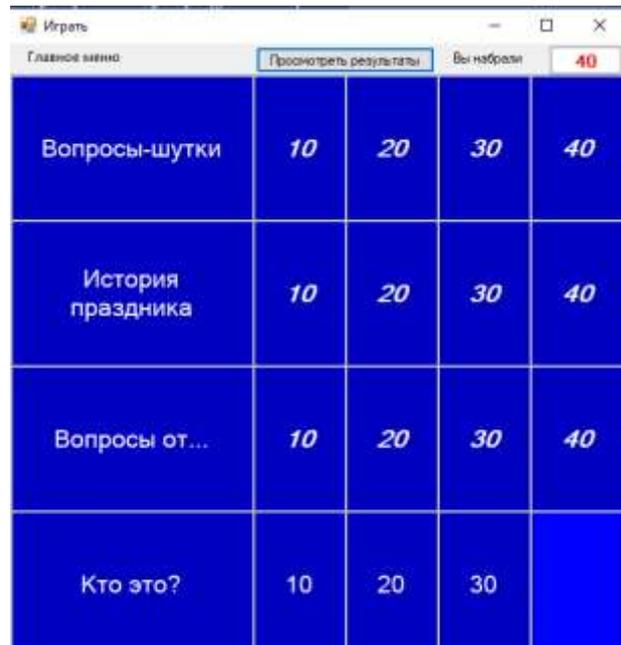


Рис. 2. Игровое поле

На данной форме можно посмотреть результаты игрока, нажав на кнопку «Просмотреть результаты». Выбрав один из вариантов вопросов, откроется форма с вопросом (рис. 3).

В случае правильного ответа стоимость вопроса прибавляется к счету игрока. В случае неверного ответа очки снимаются со счета игрока. При выборе «Добавить вопрос» в меню откроется форма с добавлением вопроса (рис. 4).

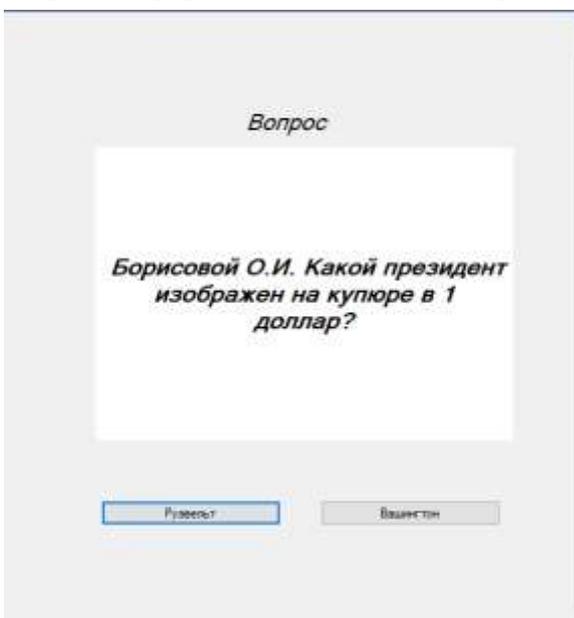


Рис. 3. Форма с вопросом

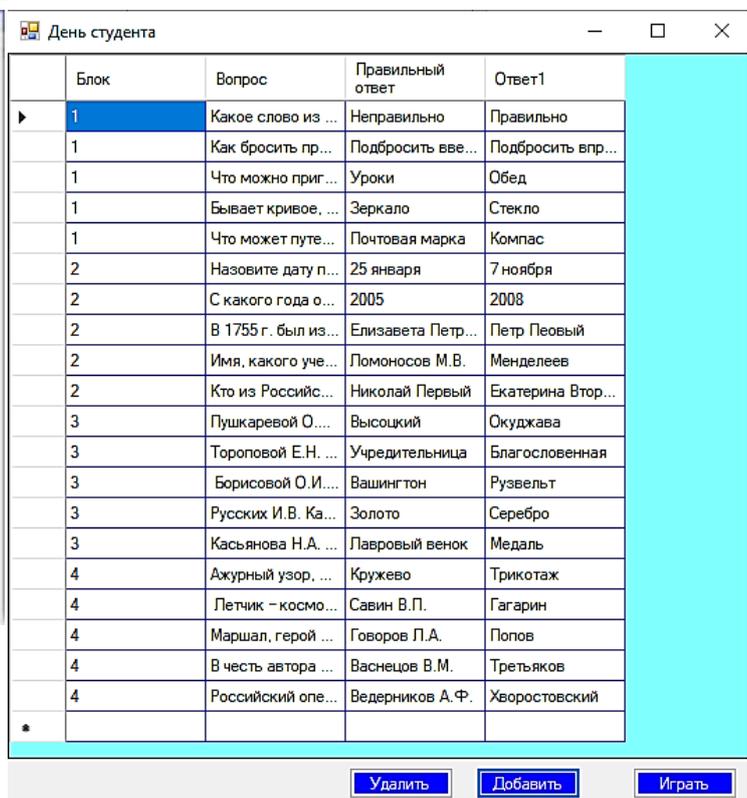


Рис. 4. Форма для добавления вопросов

Также в «Главном меню» можно посмотреть «О программе» или «Автор». На форме «Добавить вопрос» можно удалять и сохранять вопросы. При нажатии на кнопку «Добавить» откроется новая форма для заполнения всех полей таблицы (рис. 5).

Рис. 5. Добавление вопроса

На данном этапе программа была неоднократно протестирована, ошибок не выявлено. Данная игра может быть использована для проведения контроля знаний студентов по теме и/или курсу, а также на этапе мотивации при изучении новой темы или для внеклассных занятий. Таким образом, были решены следующие задачи:

- 1) проведен анализ научной и научно-методической литературы и Интернет-ресурсов по данной теме;
- 2) рассмотрена классификация компьютерных игр;
- 3) выделены особенности логических компьютерных игр;
- 4) разработана компьютерная логическая игра «День студента».

Найденные и проанализированные примеры позволили сделать следующие выводы:

1. Созданная компьютерная логическая игра «День студента» на Visual Studio, будет полезна обучающимся.
2. Разработанная программа позволит игрокам проверить свои знания в совершенно разных областях, развивает логику и смекалку.

Таким образом, задачи решены в полном объеме, цель достигнута – разработан образовательный ресурс, который можно использовать в качестве учебно-методического сопровождения занятия, урока.

Литература

1. Зальцман Марк. Компьютерные игры: как это делается. М.: Логрус, 2000. 530 с.

© Борисова Е.В., 2022

Брагин К.А., Кушев В.О., канд. пед. наук
Пермский государственный
национальный исследовательский университет
г. Пермь, Россия

МЕТОДИКА ВНЕШНЕГО ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Во время разработки программного обеспечения (ПО), перед компанией возникает потребность в его объективном тестировании, т.к. не всегда разработчикам удается предугадать все «узкие» места разрабатываемого ПО. Немногие компании имеют собственный штат тестировщиков, для решения этой проблемы. Одним из вариантов решения задач независимого от разработчика тестирования ПО является передача этого бизнес-процесса внешней организации.

Аутсорсинг¹ получил в последние годы широкое распространение как высокоэффективная форма организации бизнеса, позволяющая предприятию снизить издержки и сосредоточиться на наиболее рентабельных для него видах деятельности [1]. В качестве примера можно указать повсеместный отказ от собственного штата сотрудников и привлечение внешних специалистов.

Первым этапом в организации тестирования ПО, компания-заказчик и компания-исполнитель является разработка плана взаимодействия между компаниями, методики тестирования, и др. документов. Определяются сроки проведения тестирования, необходимый штат сотрудников, квалификация сотрудников, а также необходимые инструменты тестирования. На этом этапе наблюдаются трудности в обеспечении согласованности. Процесс от заявки на тестирование программного продукта до плана тестирования, занимает достаточно долгое время. Для выбора подходящего плана тестирования сотрудникам компании, которая будет проводить тестирование, необходимо вручную проанализировать ПО и предоставить готовый вариант тестирования. Этот процесс, возможно автоматизировать, что позволит сократить время составления плана тестирования.

План тестирования - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения (<https://clck.ru/ejHmc>). В плане тестирования должно быть зафиксированы следующие ключевые моменты:

1. Объект тестирования:
 - Описание объекта тестирования: системы, приложения, оборудования.
2. Функциональность приложения:
 - Список функций и описание тестируемой системы, её компонентов в отдельности.
3. Стратегия тестирования:

¹ Аутсорсинг - передача бизнес-процессов на исполнение сторонним заказчиком [1].

- Виды тестирования и их применение по отношению к объекту тестирования.
- 4. Последовательность проведения работ:
 - Подготовка, тестирование, анализ результатов в разрезе запланированных фаз разработки.
- 5. Критерии начала тестирования:
 - готовность тестовой платформы (тестового стенда);
 - законченность разработки требуемого функционала;
 - наличие всей необходимой документации.
- 6. Критерии окончания тестирования:
 - результаты тестирования удовлетворяют критериям качества продукта:
 1. требования к количеству открытых багов выполнены
 2. выдержка определенного периода без изменения исходного кода приложения;
 3. выдержка определенного периода без открытия новых багов.
 - 7. Окружение тестируемой системы (описание программно-аппаратных средств)
 - 8. Необходимое для тестирования оборудование и программные средства (тестовый стенд и его конфигурация, программы для автоматизированного тестирования и т.д.)
- 9. Риски и пути их разрешения

Без плана тестирования производить оценку качества программы крайне неэффективно и дорого. В зависимости от объема производимых тестов возрастают или уменьшаются трудозатраты тестировщиков, которые проводят эти тесты. Важность правильно составленного плана тестирования очень высока. Немаловажно в процессе подбора оптимальной плана тестирования не выходить за рамки требований к этапу проверки качества программ. Для этого производится специальный анализ требований, на самом раннем этапе всего жизненного цикла тестирования [2, с. 30].

При формировании плана тестирования программного обеспечения, тестировщики определяют необходимые методы тестирования, которые будут использоваться для оценки качества конкретных программ, а также определяют временные сроки тестирования. Универсального плана тестирования для разнотипных программ не существует. Если провести анализ выбора тестировщиками типов и методов тестирования, возможно, сформировать их классификацию, которая поможет оптимизировать выбор типов тестов и методов тестирования для конкретного ПО. Оптимальный план тестирования позволит произвести эффективное тестирование за необходимое и достаточное время [3, с. 87]. Для формирования такого плана следует учитывать принимаемую во внимание методологию разработки ПО.

При использовании линейной методологии разработки ПО высока вероятность потери времени, из-за задержек и препятствий при переходах между этапами разработки, в следствии чего возрастает риск сдвига сроков тестирования.

В гибкой методологии разработки ПО предусматривается, что с течением времени происходит постепенное изменение функционала в программную систему и, соответственно, коррекция её исходного кода, что в свою очередь подразумевает возможность возникновения

новых, еще незарегистрированных дефектов [4, с. 391]. Таким образом, существует необходимость в периодической проверке программного продукта, называемое регрессионным тестированием [5, с. 80].

В процессе разработки программных систем, в зависимости от дальнейшего использования ПО, применяются разные планы тестирования. При наличии нескольких планов тестирования, которые после всех пройденных тестов показали некоторую зависимость между выбранным планом и результатом тестирования, возможно произвести расчет закономерности выбора тех или иных способов тестирования и сформировать классификацию типов тестирования в зависимости от:

- параметров, характеризующих тестируемое ПО,
- вида программного обеспечения,
- операционной системы устройства,
- аппаратного обеспечения и т. д.

Процесс заявки на проведения тестирования начинается с обращения компании заказчика к компании исполнителю. Менеджер компании исполнителя передает сведения о заявке команде аналитиков. Далее происходит анализ обращения, прогноз сроков тестирования, составление плана тестирования. Команда аналитиков передает возможные варианты проведения тестирования менеджеру, тот в свою очередь связывается с заказчиком, для предоставления планов тестирования. Если заказчика устраивает один из предложенных планов тестирования, он может утвердить его. Все перечисленные ранее операции изображены на схеме (рис. 1).

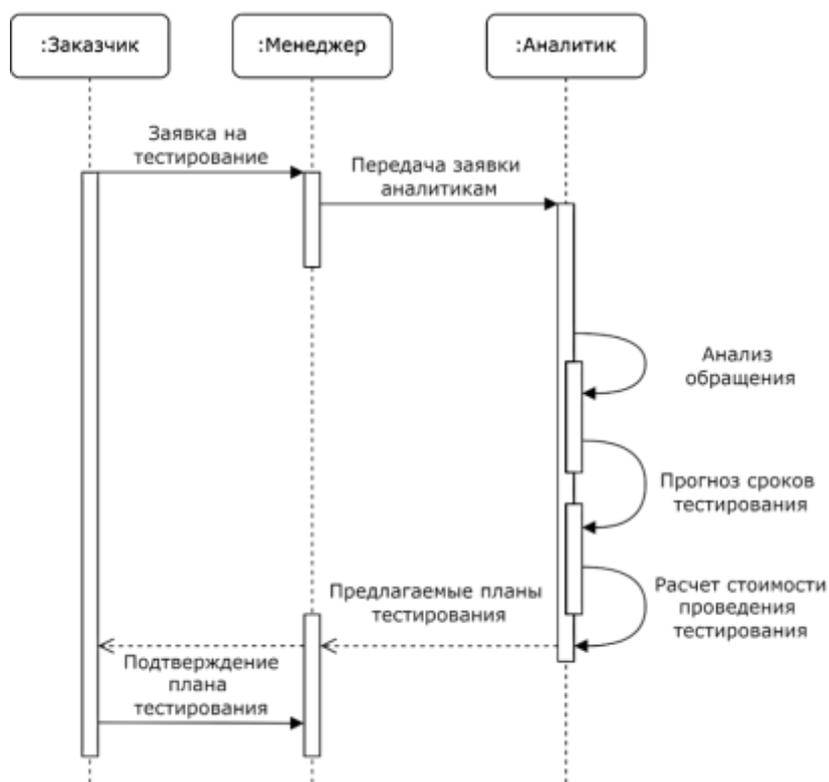


Рис. 1. Схема стандартной заявки на тестирования ПО

Наличие посредника в передаче заявки и вариантов тестирования, сильно замедляет процесс обработки обращения. Вторым замедляющим звеном является команда аналитиков. Проведение анализа, прогноз сроков тестирования и расчет стоимости тестирования, производятся в большинстве случаев без использования средств автоматизации. Автоматизируя эти этапы, возможно, значительно сократит время обработки обращения на проведение тестирования ПО. Возможным вариантом автоматизации процесса подготовки заявки, является использование информационной системы (ИС), которая может произвести все функции обработки заявки без присутствия человека. ИС получает на вход некоторый перечень параметров, относящиеся к требованиям к проведению тестирования ПО, по которым происходит автоматизированный подбор плана тестирования.

Выбор плана тестирования ИС должен происходить по ранее сформированным классификациям. Такой метод позволит определить план тестирования, опираясь на результаты предыдущих тестирований ПО. После выбора тех или иных параметров, в формировании обращения на проведение тестирования, появляется закономерность в выборе плана тестирования. Одним из основных условий правильного функционирования ИС, является наличие обратной связи, которая позволяет корректировать план тестирования или обращение на проведение тестирования, что дает возможность выбора оптимального для компании-заказчика плана тестирования. Пример автоматизированной заявки изображен на схеме (рис. 2).



Рис. 2. Схема автоматизированной заявки на тестирования ПО

Методика внешнего тестирования программного обеспечения описывает процесс передачи тестирования ПО внешней организации, а также возможный вариант автоматизации процесса планирования проведения тестирования. Автоматизация позволяет уменьшить трудозатраты на планирование тестирования вручную, приступить к тестированию сразу после обработки заявки ИС.

Литература

1. Котляров И.Д. Аутсорсинг как форма межфирменной кооперации: теоретический анализ // Вестник Института экономики Российской академии наук. 2015. № 5. С. 19-31
2. Куликов С.С. Тестирование программного обеспечения. Базовый курс. Минск: Четыре четверти, 2017. 312 с.
3. Брагин К.А., Кушев В.О. Разработка оптимальной методики тестирования программного обеспечения // XXIII Всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета: сб. статей. (г. Нижневартовск, 06–07 апреля 2021 года). Нижневартовск, 2021. С. 86-90.
4. Кон М. Scrum: гибкая разработка ПО. М., 2011. 576 с.
5. Канер С., Фолк Д., Кек Нгуен Е. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. К.: ДиаСофт, 2001. 544 с.

© Брагин К.А., Кушев В.О., 2022

Витченко В.А.

Новосибирский национальный исследовательский
государственный университет
Институт автоматике и электрометрии СО РАН
г. Новосибирск, Россия

ИССЛЕДОВАНИЕ СПОСОБА РАСШИРЕНИЯ WEB-IDE ПРОЦЕСС-ОРИЕНТИРОВАННОГО ЯЗЫКА REFLEX

Разработка промышленных процесс-ориентированных алгоритмов управления кибер-физическими системами сопряжена с рядом сложностей и рисками, связанными с высокой стоимостью ошибок. Для оптимизации процесса разработчики вынуждены использовать нестандартные средства разработки [1, с. 189].

Решения предложенные в ИАиЭ СО РАН включают процесс-ориентированную парадигму программирования, язык Reflex [2, с. 194; 3, с. 23]. Разрабатывается интерактивная среда разработки RIDE 2.0 на базе платформы Eclipse Theia (<https://clck.ru/ey5nU>). RIDE 2.0 позволяет разрабатывать промышленные процесс-ориентированные алгоритмы управления на языке Reflex.

Для эффективной разработки алгоритмов управления помимо непосредственно редактора кода необходимы специализированные трансляторы в другие языки программирования, а также вспомогательные средства разработки, например отладчики, редакторы диаграмм, верификаторы, инструменты для реверсивного инжиниринга. Чтобы единообразно добавлять дополнительную функциональность в среду, необходим механизм интеграции расширений. Также необходимо обеспечить изолированность кода расширений друг от друга и от ядра среды разработки. В связи с этим, целью работы стало исследование способов расширения WEB-IDE процесс-ориентированного языка Reflex. Были поставлены следующие задачи:

1. Проанализировать специфику RIDE 2.0;
2. Проанализировать существующие механизмы расширения функциональности RIDE 2.0;
3. Сформулировать требования к механизмам расширения;
4. Реализовать программные средства, упрощающие разработку расширений;
5. Описать процедуру разработки и интеграции новых расширений.

Анализ Reflex IDE 2.0

В процессе анализа было выявлено, что RIDE 2.0 это web-приложение, созданное на базе платформы Eclipse Theia. Eclipse Theia является платформой, предоставляющей возможность создавать среды разработки для различных языков программирования, а также имеющей модульную структуру (<https://clck.ru/ey4XT>). Eclipse Theia и RIDE 2.0 написаны на языке TypeScript и состоит из Node.js модулей (<https://clck.ru/ey5Ac>). Каждый модуль представляет собой клиент-серверное приложение (<https://clck.ru/ey5CQ>).

Клиентский код запускается в браузере каждого пользователя отдельно, когда пользователь переходит по определенному адресу в сети. Серверный код работает на

удаленном сервере и обрабатывает запросы от всех клиентов. Добавление расширения, предоставляющего новую функциональность, требует двух этапов – разработки и интеграции. В процессе разработки создается исполняемый код расширения. После того как расширение разработано, его необходимо интегрировать в существующую сборку среды разработки, которую непосредственно используют пользователи. RIDE 2.0 уже реализует два механизма расширения, один из которых наследуется от родительской платформы Eclipse Theia. Другой механизм, созданный разработчиками RIDE 2.0 и позволяющий расширять функциональность среды посредством добавления предметно-ориентированных модулей.

Анализ существующих механизмов расширения

В процессе анализа был проведен анализ существующих механизмов расширения. Подход, унаследованный от родительской платформы RIDE 2.0, предполагает тесную интеграцию с внутренним устройством Theia. Это позволяет обеспечить максимальную гибкость в организации пользовательского интерфейса расширения. В данном случае разработчику в процессе реализации расширения необходимо изучить детали внутреннего устройства платформы Eclipse Theia, так как разработка расширения требует реализации абстракций, описанных в кодовой базе платформы. Это возводит определенный порог входа для разработчиков расширений. Интеграция нового расширения при таком подходе производится путем изменения конфигурационных файлов сборки среды разработки (<https://clck.ru/ey5DX>).

Подход, предложенный разработчиками RIDE 2.0, с другой стороны, полностью абстрагирует разработчиков расширений от внутреннего устройства Theia. В результате гибкость сведена к минимуму. Расширение в данном случае является Java приложением, представляющее собой HTTP сервер с одним доступным запросом. Возможности изменять графический интерфейс расширения не предоставляется. Также отсутствует возможность программного взаимодействия расширений друг с другом. Процесс интеграции помимо изменения конфигурационных файлов также требует добавления дополнительных исполняемых файлов в репозиторий проекта (<https://clck.ru/ey5nU>).

Принимая во внимание существующие решения, были сформулированы следующие требования к механизму расширения:

1. Механизм расширения должен предоставлять возможность создания расширений с собственным графическим интерфейсом и произвольным программным интерфейсом.
2. Механизм расширения должен абстрагировать разработчиков от внутреннего устройства Theia.
3. Механизм расширения должен предоставлять процесс интеграции новых расширений путем только модификации конфигурационных файлов.

Предложенное решение

Решение предполагает использование гибкого подхода, унаследованного от родительской платформы Theia. Возможность гибкой настройки является следствием из необходимости тесно интегрироваться с внутренним устройством Theia. Поэтому при любом абстрагировании от внутреннего устройства гибкость настройки будет снижена. Поэтому

более гибкий подход был урезан до уровня, при котором решение все еще удовлетворяет необходимым требованиям. Для того, чтобы избавиться от излишней зависимости кода расширений от реализации платформы, было решено реализовать прослойку, которая реализует все необходимые платформу-зависимые абстракции и предоставляет упрощенный интерфейс взаимодействия клиентского и серверного модулей расширения.

Вынесение прослойки в отдельный жестко закодированный модуль сделало бы интеграцию с платформой Theia «черным ящиком» для разработчиков расширений, поэтому было решено оформить модуль в виде генератора репозитория расширения. Генератор был реализован с использованием библиотеки Yeoman (<https://clck.ru/ey5Fb>). Генератор позволяет создать базовый репозиторий с уже настроенной интеграцией с платформой Theia. В то же время, так как весь код интеграции находится в репозитории расширения, разработчик имеет возможность при необходимости изучить пример реализации интеграции, поэкспериментировать с кодом интеграции.

В результате, клиентская часть расширения ограничена единственным React виджетом, который, однако, может быть расширен стандартными средствами React, что позволяет реализовать графическое приложение с произвольным графическим интерфейсом. Серверная часть расширения ограничивается одним TypeScript интерфейсом, который может быть расширен произвольным количеством методов с различными сигнатурами.

Архитектура получаемого расширения

Расширение является Node.js модулем и поделено на три подмодуля — модуль коммуникации, серверный модуль, клиентский модуль. Модуль коммуникации определяет протокол клиент-серверного взаимодействия остальных модулей. Здесь определен TypeScript интерфейс, упомянутый выше. Остальные модули зависят от модуля коммуникации.

Серверный модуль ответственен за реализацию бизнес логики расширения, исполняемой на удаленном сервере. Помимо этого, он отвечает за интеграцию с другими расширениями. Серверный модуль реализует интерфейс, определенный в модуле коммуникации. Генератор реализует интеграцию серверного модуля с серверной инфраструктурой платформы Theia и другими расширениями. В частности, с “AST-Service” модулем, который предоставляет доступ к абстрактному синтаксическому дереву Reflex программы, над которой пользователь работает в данный момент.

Клиентский модуль отвечает за логику взаимодействия с пользователем. Для клиентского модуля генератор также создает код интеграции с клиентской частью платформы Theia и создает входную точку для реализации пользовательского интерфейса — React виджет. Клиентский модуль взаимодействует с серверным модулем посредством использования объекта, который реализует интерфейс из модуля коммуникации.

Таким образом, клиентский и серверный модули содержат компоненты зависимые от Theia, но их реализация при разработке расширения будет полностью сгенерирована разработанным генератором. В то же время, код интеграции доступен для модификации, что позволяет при необходимости выйти за рамки текущей реализации интеграции.

В стандартном случае разработчику расширения необходимо определить интерфейс коммуникации, разработать бизнес логику приложения и логику взаимодействия с оператором в соответствующих модулях. При этом разработчик будет абстрагирован от платформы Theia. Необходимые требования к навыкам разработчика в данном случае сводятся к базовому умению разрабатывать программы, используя TypeScript и React.

Описание процесса разработки и интеграции

Был предложен следующий процесс разработки и интеграции:

1. Разработчик расширения клонирует репозиторий RIDE 2.0 для локальной разработки. В репозитории представлена минимальная кодовая база, необходимая для корректной работы базовых функций.

2. Разработчику необходимо локально установить библиотеку Yeoman и генератор репозитория, используя пакетный менеджер Node.js — NPM (<https://clck.ru/ey5HV>).

3. Далее разработчику необходимо запустить генератор в директории репозитория RIDE 2.0 и указать имя желаемого расширения. Так как имена пакетов NPM должны быть уникальными следует убедиться, что пакета с таким именем еще не существует в реестре, а также использовать постфикс “reflex-ide-extension” для единообразия (<https://clck.ru/ey5LL>).

4. Затем разработчику написать код, реализующий всю необходимую логику расширения.

5. Как только расширение написано и протестировано, необходимо опубликовать расширение в регистре NPM пакетов. На этом разработка первой версии расширения будет завершена.

6. Далее администратору сервера RIDE 2.0 необходимо интегрировать новое расширение в сборку, посредством добавления в конфигурационный файл строки вида: <имя расширения: версия расширения,>.

7. Затем развернуть среду разработки на сервере с измененным конфигурационным файлом. В процессе развертки приложение само обратится в регистр NPM и установит зависимость на новое расширение.

Администратору сервера необходимо обратить внимание на то, что при включении расширений в сборку, на сервере будет исполняться код соответствующих расширений, в связи с этим необходимо быть полностью уверенным в том, что расширение не содержит вредоносного кода. Поэтому необходимо закреплять конкретные версии расширений и производить проверку кода расширений. Либо использовать расширения только от проверенных разработчиков.

Было проведено исследование способа расширения RIDE 2.0, был предложен механизм расширения, предполагающий использование созданного генератора репозитория на базе библиотеки Yeoman, была описана процедура разработки и интеграции расширений. Механизм расширения удовлетворяет сформулированным требованиям. Генератор репозитория публично доступен и может быть установлен с помощью NPM. Разработанное решение унифицирует процесс разработки и интеграции расширений для RIDE 2.0 и

уменьшает порог входа при разработке расширений с собственным графическим интерфейсом.

На данный момент проводится исследование на тему расширения возможностей механизма, позволяющих встраивать модули, написанные на других языках программирования и также имеющих произвольный графический интерфейс пользователя. Планируется исследовать расширения, которые будут написаны с использованием механизма, чтобы выявить наиболее часто встречающиеся паттерны и вынести их в отдельный шаблон генератора. Планируется разработать формальную процедуру проверки расширений на наличие вредоносного кода. Планируется адаптировать генератор к использованию его не только для RIDE 2.0, но и для других сред разработки, созданных на основе платформы Eclipse Theia.

Литература

1. Lyakh T.V., Zyubin V.E., Garanina N.O. Automatic verification of control software in cyber-physical systems with plant simulators // Optoelectronics, Instrumentation and Data Processing. 2019. Т. 55. № 2. С. 189-197.
2. Зюбин В.Е. Процесс-ориентированное программирование. Новосибирск: НГУ, 2011.
3. Zyubin V. E., Liakh T. V., Rozov A. S. Reflex language: a practical notation for cyber-physical systems // Системная информатика. 2018. № 12. С. 85-104.

© Витченко В.А., 2022

УДК 004

Гафетдинова Г.Х., Рахматуллина А.М., Федорова О.В., канд. пед. наук
Университет управления «ТИСБИ»
г. Казань, Россия

РАЗРАБОТКА И ВНЕДРЕНИЕ В ЭКСПЛУАТАЦИЮ ВЕБ-САЙТА ПО ПОДБОРУ ЭКСКУРСИЙ ПО ГОРОДУ КАЗАНЬ

В настоящее время сфера туризма – это одна из стремительно развивающихся сфер социальной жизни, тесно взаимодействующая со многими другими отраслями. Республика Татарстан является лидирующим регионом Российской Федерации. Если обратиться к итогам развития отрасли туризма в Татарстане за 2021 год, которые были опубликованы на официальном сайте комитета по туризму в РТ (<https://clck.ru/ep9vm>), то можно проследить за темпом повышения интереса, как к самому туризму, так и к городу Казань. По динамике туристского потока в РТ наглядно видно, что самым посещаемым туристами объектом является Казанский Кремль (рис. 1).

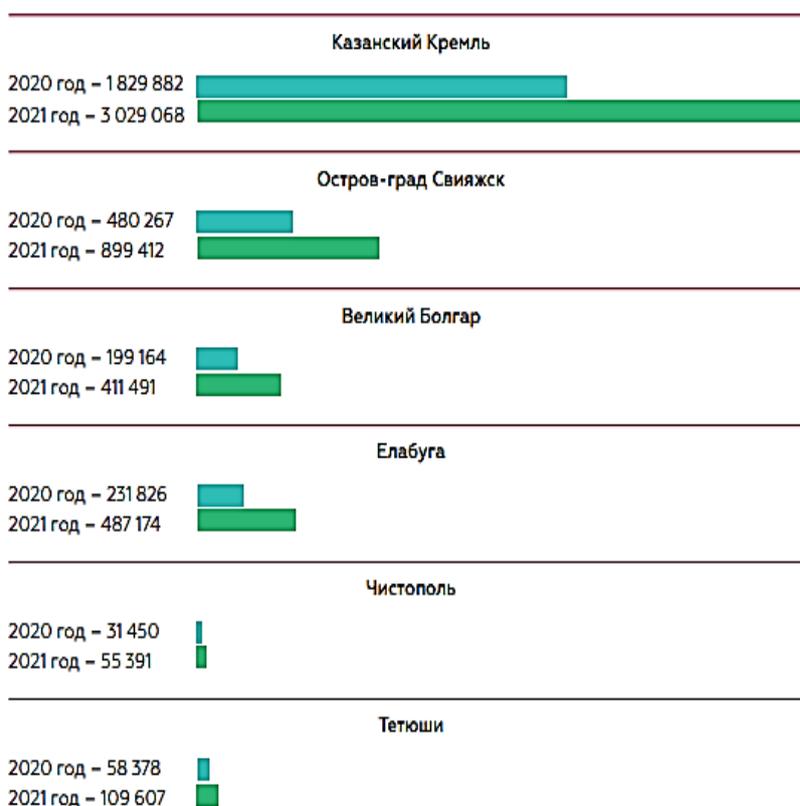


Рис. 1. Динамика туристского потока в основные туристские центры РТ

Повышение качества обслуживания невозможно без современных систем автоматизации деятельности. Это способно значительно упростить работу, повысив производительность труда. Поэтому было принято решение создать веб-приложение по подбору экскурсий по городу Казань. С помощью данной системы появится возможность ознакомления с экскурсионным фондом, изменения экскурсионного тура под желания туриста, бронирования не только групповых, но и индивидуальных экскурсий, обработки заявок. До начала разработки веб-приложения необходимо сделать анализ предметной

области и конкурентов, сравнить функции, которые выполняют найденные сервисы.

Основными конкурентами являются:

www.tatturist.ru – экскурсионное бюро «Казань»;

kazantravel.ru – туристско-информационный центр г. Казань;

tur-kazan.ru – экскурсионный сервис «Казань»;

При проведении анализа, были выявлены основные функциональные возможности данных сервисов:

- Расписание групповых экскурсий
- Наличие индивидуальных экскурсий
- Возможность редактирования индивидуальных экскурсий
- Уведомление о дате и месте встречи
- Авторизация пользователя
- Возможность оставлять отзывы и комментарии
- Наличие чат-бота
- Привязка к социальным сетям
- Обратная связь
- Наличие корзины (или «избранного»)

На основе данного списка были выявлены недостатки конкурентов. На веб-сайте экскурсионного бюро «Казань» отсутствуют следующие возможности:

- Авторизация пользователя
- Наличие индивидуальных экскурсий
- Возможность редактирования индивидуальных экскурсий
- Наличие чат-бота
- Привязка к социальным сетям
- Наличие корзины (или «избранного»)

На сервисе туристско-информационный центр г. Казань kazantravel.ru отсутствует следующий функционал:

- Авторизация пользователя
- Возможность редактирования индивидуальных экскурсий
- Наличие чат-бота
- Привязка к социальным сетям
- Наличие корзины (или «избранного»)
- На сайте экскурсионного сервиса «Казань» tur-kazan.ru нет следующих возможностей:
- Возможность редактирования индивидуальных экскурсий

Целью проекта является разработка веб-приложения со следующими функциональными возможностями: выбор и бронирование интересующих экскурсий, добавление выбранного в избранное/корзину, подбор индивидуального обзора, возможность регистрации и авторизации, наличие чат-бота и обратной связи.

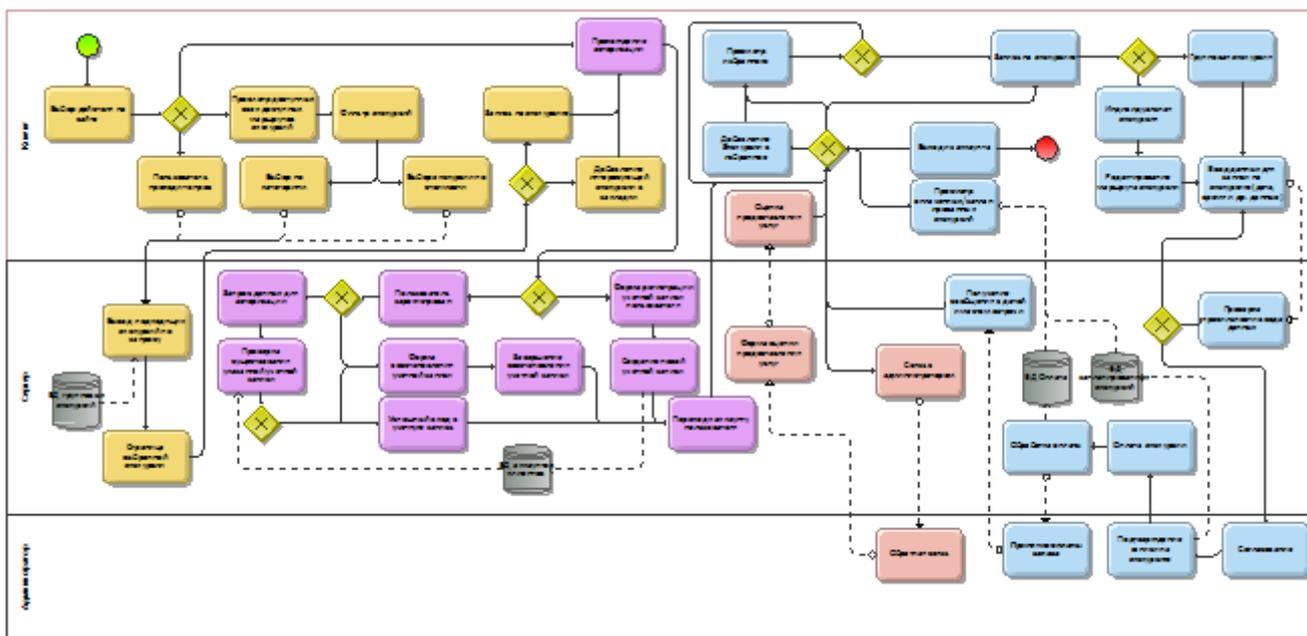


Рис. 2. BPMN-модель

«Казан Трип» - это веб-приложение с возможностью заблаговременного планирования досуга, которое помогает выбрать экскурсию по вкусу и по материальным возможностям.

Преимуществом является индивидуальный подход по запросам гостя, широкий ассортимент при выборе культурных и исторических достопримечательностей города. Благодаря этому сервису выявлено удобство в выборе и бронировании экскурсий, а также гарантия возврата оплаты обеспечены.

Нотация моделирования бизнес-процессов изображена на рисунке 2. Для наглядности она выделена разными цветами, выполняемые задачи поделены на 4 части. Первая часть – выбор экскурсий по категориям, вторая – авторизация пользователя, третья – бронирование, оплата, четвертая – обратная связь. Клиент выбирает экскурсии и бронирует. Подтверждение бронирования и работу с базой данных выполняет сервер. Функции согласования и обратной связи – это роль администратора.

При разработке веб-сайта были поставлены следующие задачи, которые представлены в следующем формате:

1. Аналитическая фаза
 - 1.1. Анализ предметной области и формирование требований
 - 1.2. Формирование требований к системе
 - 1.3. Изучение целевой аудитории
 - 1.4. Анализ конкурентов
 - 1.5. Формулировка цели и предварительный план сайта
 - 1.6. Техническое задание
2. Разработка дизайна
 - 2.1. Концепция
 - 2.2. Разработка структуры сайта
 - 2.3. Создание внешнего вида страниц сайта

- 2.4. Создание каталога сайта
- 3. Верстка [2].
 - 3.1. Первоначальная вёрстка
 - 3.2. Вёрстка страниц сайта
- 4. Кодинг
 - 4.1. Итоговое согласование данных
 - 4.2. Создание и заполнение базы данных [1]
 - 4.3. Подключение базы данных
 - 4.4. Написание кода
 - 4.5. Выбор хостинга
 - 4.6. Регистрация доменного имени
 - 4.7. Перенос на постоянный хостинг
- 5. Тестирование и отладка
 - 5.1. Тестирование
 - 5.2. Испытание работы сайта на различных примерах, анализ результатов
 - 5.3. Исправление ошибок
 - 5.4. Итоговое тестирование
- 6. Сдача проекта
 - 6.1. Ввод в Production

Для выполнения поставленных задач были определены следующие специалисты: руководитель проекта, аналитик, frontend-разработчик, backend-разработчик, веб-дизайнер, системный архитектор, тестировщик, администратор баз данных. Для расчета фонда оплаты труда был проанализирован сервис интернет-рекрутмента сайта hh.ru, с помощью которого была вычислена медиана зарплат специалистов в Республике Татарстан (рис. 3).

	Руководитель проекта	Аналитик	Frontend-разработчик	Backend-разработчик	Веб-дизайнер	Системный архитектор	Тестировщик	Администратор БД
1	73620	63695	69840	61840	40950	61650	35630	39900
2	78650	67560	72360	70500	42960	66500	39770	43280
3	85250	70658	796540	76820	50950	73780	43360	51130
4	105630	86000	87455	95900	55950	78520	60500	59800
5	115200	93500	94500	100000	65500	89240	76600	60630
6	119000	96350	95980	101500	69280	90800	105690	60660
7	120850	105700	100600	105500	79280	95360	106000	62660
8	140600	120500	11000	115000	90500	110600	119500	70500
Медиана	110415	89750	90977,5	97950	60725	83880	68550	60215
В час	460,06	373,96	379,07	408,13	253,02	349,50	285,63	250,90

Рис. 3. Медиана зарплат

Экономический расчет, который включает в себя итоговую стоимость с учетом прибыли в 10%, страховые взносы (30%), фонд оплаты труда, аренда офиса (включая оборудование, оплата связи, интернет) составил 768 748, 89 руб. (рис. 4).

	Руководитель проекта	Аналитик	Frontend-разработчик	Backend-разработчик	Веб-дизайнер	Системный архитектор	Тестировщик	Администратор БД	Итого
Итого человеко-часов	202	256	256	160	128	128	64	24	1218
Оплата труда за 1 час	460,06 Р	373,96 Р	379,07 Р	408,13 Р	253,02 Р	349,50 Р	285,63 Р	250,90 Р	
Стоимость ФОТ	92 932,63 Р	95 733,33 Р	97 042,67 Р	65 300,00 Р	32 386,67 Р	44 736,00 Р	18 280,00 Р	6 021,50 Р	452 432,79 Р
Страховые взносы (30%)	27 879,79 Р	28 720,00 Р	29 112,80 Р	19 590,00 Р	9 716,00 Р	13 420,80 Р	5 484,00 Р	1 806,45 Р	135 729,84 Р
Аренда офиса на одного человека (3 месяца)	13 837,50 Р	13 837,50 Р	13 837,50 Р	13 837,50 Р	13 837,50 Р	13 837,50 Р	13 837,50 Р	13 837,50 Р	110 700,00 Р
Себестоимость	134 649,91 Р	138 290,83 Р	139 992,97 Р	98 727,50 Р	55 940,17 Р	71 994,30 Р	37 601,50 Р	21 665,45 Р	698 862,63 Р
Итоговая стоимость с учетом прибыли в 10 %	148 114,90 Р	152 119,92 Р	153 992,26 Р	108 600,25 Р	61 534,18 Р	79 193,73 Р	41 361,65 Р	23 832,00 Р	768 748,89 Р

Рис. 4. Экономический расчет

При создании веб-приложения были использованы такие технологии, как:

- редактор кода Visual Studio Code;
- HTML – язык гипертекстовой разметки веб-страниц;
- язык таблиц стилей для визуального оформления веб-сайтов CSS;
- язык программирования PHP;
- веб-сервис для совместной работы и управления репозиторием кода GitHub;
- динамический язык программирования JavaScript;
- фреймворки Bootstrap и React [3];
- в качестве системы управления базами данных MySQL;
- веб-приложение с открытым кодом, представляющее собой веб-интерфейс для администрирования СУБД MySQL – MyPhpAdmin
- для установки серверной среды на локальном компьютере Mamp.

Часть программной реализация компонента Navbar, который используется на страницах веб-приложения на основе главной панели, представлена на рисунке 5.

```

17 <div class="navbar navbar-expand-lg">
18 <div class="container-fluid">
19 <div class="navbar-brand" href="#">Катастрофа</div>
20 <div class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarToggleGetSection"
21 aria-controls="navbarToggleGetSection" aria-expanded="false" aria-label="Toggle navigation">
22 <span class="navbar-toggler-icon"></span>
23 </div>
24 <div class="collapse navbar-collapse" id="navbarToggleGetSection">
25 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
26 <li class="nav-item">
27 <a class="nav-link active" aria-current="page" href="#">Главная</a>
28 </li>
29 <li class="nav-item">
30 <a class="nav-link" href="#">0 max</a>
31 </li>
32 <li class="nav-item dropdown">
33 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown"
34 aria-expanded="false">
35 <span>
36 </span>
37 </a>
38 <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
39 <li class="dropdown-item" href="#">Получение информации</li>
40 <li class="dropdown-item" href="#">Исторические экскурсии</li>
41 <li class="dropdown-item" href="#">Найти экскурсии</li>
42 </ul>
43 </li>
44 <li class="nav-item">
45 <a class="nav-link" href="#">Друзья</a>
46 </li>
47 <li class="nav-item">
48 <a class="nav-link" href="#">Контакты</a>

```

Рис. 5. Программная реализация компонента Navbar

В веб-приложении (рис. 6, 7) размещена форма регистрации и авторизации, главное меню с возможностью раскрытия, чат-бот, а также некоторые экскурсии.

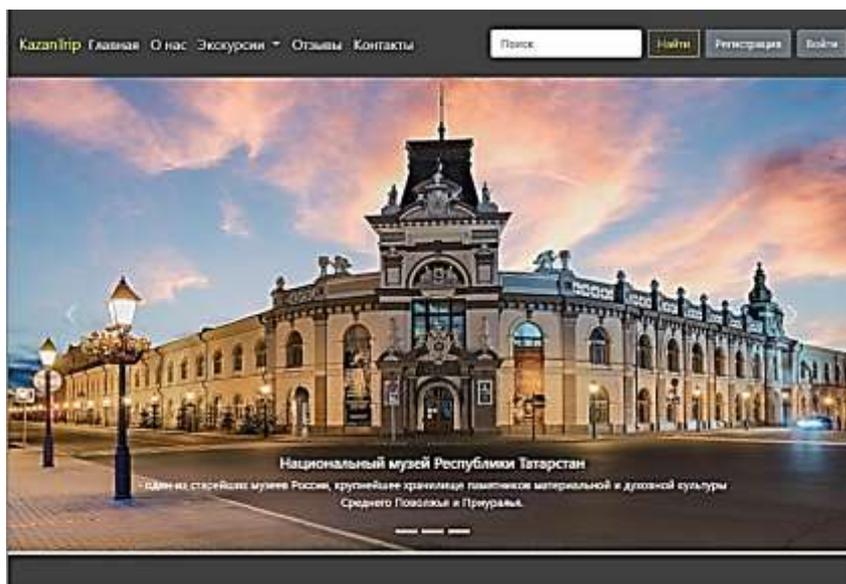


Рис. 6. Реализация главного экрана

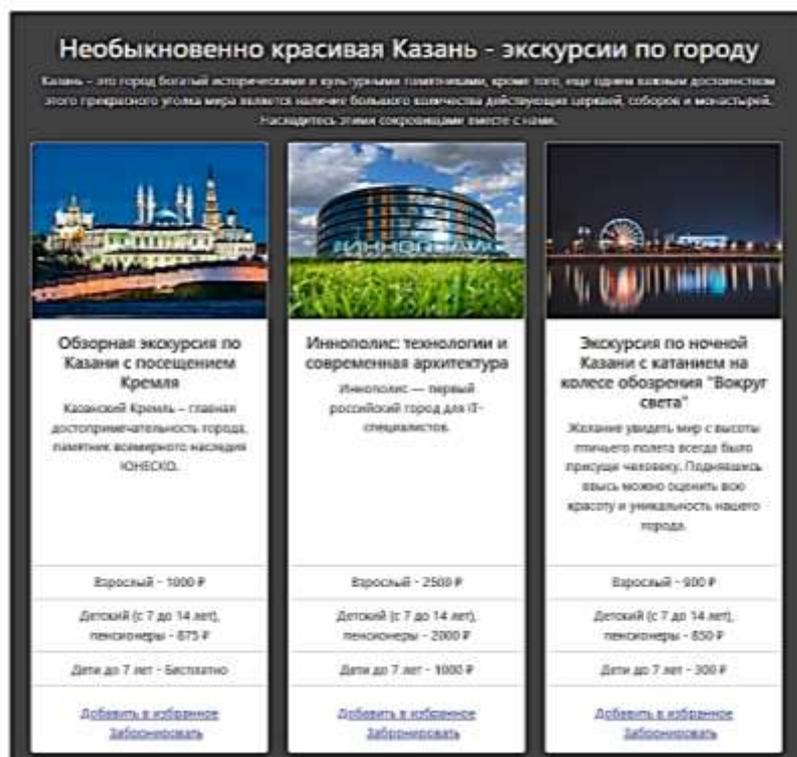


Рис. 7. Реализация главного экрана

С веб-приложения Kazan Trip можно обратиться к чат-боту Telegram. На рисунке 8 реализована онтология бота. Желтые формы – главное меню, нажимая на него, можно перейти к следующей форме. Из каждой формы есть возможность возвращения на предыдущую форму.

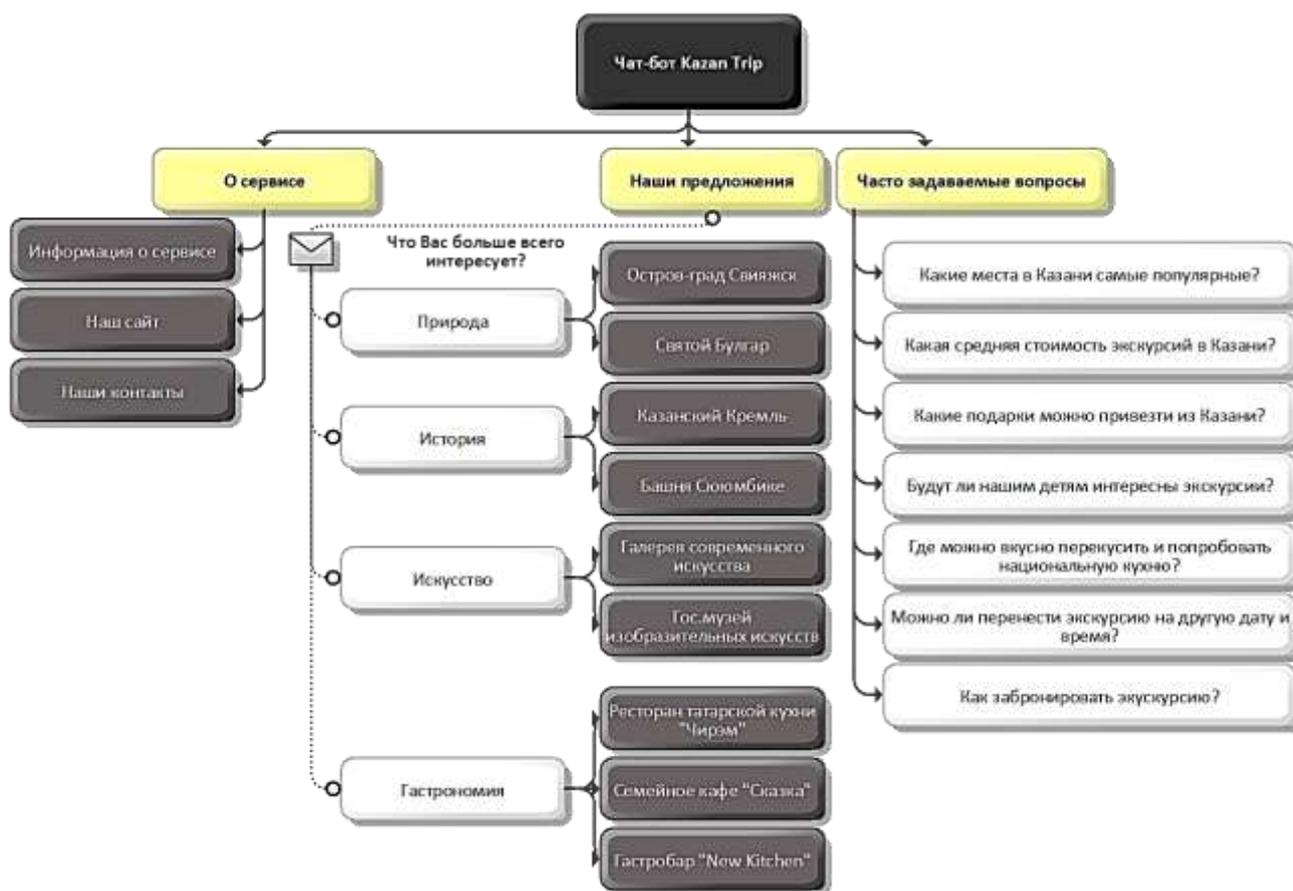


Рис. 8. Онтология чат-бота Telegram

Основываясь на официальный сайт kzn.ru, было собрано несколько наиболее часто задаваемых вопросов. Бот позволяет перейти пользователю к привлекающей внимание категории и подробно рассмотреть предлагаемые варианты с возможностью бронирования через сайт (рис. 9).

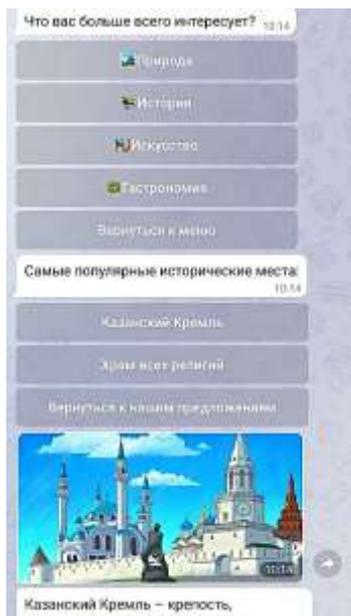


Рис. 9. Скриншот чат-бота в Telegram

В современном мире все быстро меняется, развивается, что во многих случаях возможно не уследить за нововведениями. Поэтому практически все веб-приложения и сервисы нуждаются в обновлении и оптимизации. Анализ предметной области, построение нотации моделирования бизнес процессов, экономический расчет – это важные этапы разработки любого веб-приложения, грамотный подход к этим этапам позволит определить возможные риски при разработке.

Литература

1. Волк В.К. Базы данных. Проектирование, программирование, управление и администрирование. СПб.: Лань, 2022. 244 с.
2. Кириченко А.В., Хрусталева А.А. Html5+css3. Основы современного web - дизайна. СПб: Наука и Техника, 2019. 352 с.
3. Сильвио М. Bootstrap в примерах. М.: ДМК Пресс, 2017. 314 с.

© Гафетдинова Г.Х., Рахматуллина А.М., Федорова О.В., 2022

УДК 004

Ермишина Е.В., Шарыпова Т.Н.
Ростовский государственный экономический университет (РИНХ)
г. Ростов-на-Дону, Россия

ПОЛЬЗА И ВРЕД ЗАЩИТЫ ИНФОРМАЦИИ В СОЦИАЛЬНЫХ СЕТЯХ И МЕССЕНДЖЕРАХ

С развитием всемирной паутины начали создаваться социальные сети и мессенджеры для удобства общения. Основной целью их создания является объединение и общение людей с разных уголков мира, а также получение различной информации от первоисточников.

Многие прекрасно знают, что почти все средства интернет-общения можно легко взломать и воспользоваться информацией и данными. Также известно, что за некоторыми социальными сетями и мессенджерами могут вести контроль органы власти страны, в которой создали данное средство. Самое главное в современном мире для граждан является защита информации в интернете. Никто не хочет, чтобы какие-либо их данные или секреты всплыли на обозрение всему миру. Самым безопасным средством общения сейчас считается Telegram. Telegram – это бесплатная программа для обмена сообщениями в реальном времени. Данным мессенджером удобно пользоваться. В нем есть: возможность переписки с друзьями или коллегами, возможность проводить конференции, обмен фото, видео и любыми файлами, обмен местоположениями, доступ к сообщениям с любого устройства, возможность создания ботов и каналов. Также данная программа использует низкое потребление трафика, но при этом имеет высокую скорость. В нем есть возможность удалять как сами сообщения, так и сам чат с собеседником, при этом удаление произойдет с обеих сторон. Главным минусом данного мессенджера является то, что авторизоваться в нем можно только с помощью номера мобильного телефона.

В Telegram имеется возможность создавать различные группы и каналы. Группа является своего рода чатом для общения между её участниками. Существуют закрытые (доступные только по ссылке-токену) и открытые группы (вступить могут все), количество участников может достигать 200 тысяч человек [1]. Канал представляет собой инструмент коммуникации между автором и его подписчиками. Данный формат позволяет автору (или авторам) делиться информацией с неограниченным кругом лиц, сохраняя при этом анонимность. Также в Telegram существует вкладка «Избранное», куда можно загружать очень большой объем разных файлов, туда же можно отправлять сообщения из различных источников, и они сохраняются. Здесь имеются боты, которые очень облегчают жизнь людей. В них можно следить за гороскопом, искать работу или подработку, узнавать новости в любых направлениях, читать книги, учиться иностранным языкам, узнавать об автодорожных ситуациях или о местонахождении сотрудников ДПС, следить за курсом валюты, находить информацию о нужных сервисах и услугах, а также слушать музыку.

В Telegram во время переписки с собеседником существует возможность редактирования отправленного сообщения, такая функция имеется не во всех мессенджерах и социальных сетях. Кроме того, здесь можно найти пользователя не только по номеру

телефона, но и по юзернейму², который пользователям нужно придумать при регистрации. Данный «ник»³ является уникальным в мессенджере. Многие пользуются юзернеймом для общения, когда не хотят раскрывать своей настоящей личности. Именно этим и пользуются террористы, кардеры⁴, закладчики наркотиков, владельцы нелегального бизнеса и другие вредители мира и спокойствия. Они могут создавать анонимные каналы и распространять неподтвержденную, недостоверную, ложную или даже вредоносную информацию [2].

Распространение фальшивой информации является меньшим из зол. В данном мессенджере наркодилеры могут общаться с закладчиками, а те, в свою очередь, способны связываться с людьми, употребляющими наркотики, и сообщать им о месте нахождения запрещенных веществ. Участники террористических организаций имеют возможность связываться друг с другом и обсуждать информацию о том, где и когда будет происходить следующий террористический акт, какие методы захвата и давления можно использовать, а также какие требования нужно будет предъявить правительству. Здесь люди могут распространять интимные фотографии других людей с помощью закрытых Telegram-каналов.

Многие пользователи Telegram даже не задумываются о том, что некоторые каналы, на которые они подписаны, могут нести вред для здоровья и психики человека. Здесь могут настраивать людей против друг друга, вербовать в террористические организации, привлекать к применению наркотических веществ, а также настраивать пользователей на употребление алкогольной или табачной продукции. В некоторых странах мира запрещено использование мессенджера Telegram, так как это может нанести вред государственному строю.

Еще одним популярным мессенджером является WhatsApp. WhatsApp – это бесплатный сервис обмена мгновенными сообщениями и голосовой связи по IP. Данным приложением очень удобно пользоваться. Здесь есть возможность отправлять текстовые и голосовые сообщения, фотографии, видеозаписи, документы и аудиозаписи любого формата создавать общие беседы (до 256 человек), а также совершать аудио- и видеозвонки как в чате с одним человеком, так и в какой-либо беседе. Данный мессенджер не расходует много интернет-трафика, в нем постоянно работает шифрование сообщений, существует синхронизация с контактами в телефоне, благодаря чему не нужно вбивать номера вручную. Главным минусом WhatsApp является то, что зарегистрироваться в нем можно только с номером мобильного телефона.

WhatsApp имеет Web-версию – WhatsApp Web. WhatsApp Web – это дополнение к аккаунту WhatsApp на смартфоне, работающее на ноутбуке или компьютере. Вся информация с телефона становится доступной на экране монитора, можно отправлять сообщения и делиться файлами. Данная версия не работает, если смартфон не подключен к сети Интернет. При использовании Web-версии на экране мобильного телефона всплывает незакрывающееся окно с информацией о том, что используется WhatsApp Web. Отключиться от Web-версии можно как с компьютера, так и с телефона. Зачастую при использовании этой версии

² Англ. Username – имя пользователя.

³ Англ. Nickname – кличка, прозвище, псевдоним. Сленг – «ник».

⁴ Англ. Carding – мошенничество с платежными картами. Кардеры – мошенники.

пользователи забывают выходить из аккаунта, и тогда посторонние люди могут воспользоваться этим и распространить какую-либо информацию от имени данного пользователя. Этот мессенджер очень популярен в Российской Федерации. Граждане пользуются им для общения с коллегами, друзьями, родственниками. Во многих организациях используют мессенджер для обмена различной информации. Для этого создаются общие беседы среди одного подразделения или между несколькими сразу, среди начальников каких-либо отделов, среди руководящего звена. По этим причинам данный мессенджер любят люди.

В использовании WhatsApp довольно-таки сильно отличается от Telegram. В WhatsApp сообщение, полученное от неизвестного лица не будет шифроваться юзернеймом, а будет просто указан номер мобильного телефона, с использованием которого зарегистрировался пользователь, по этой причине здесь нельзя оставаться анонимным пользователем. Также здесь нельзя публиковать какую-либо информацию для большого количества лиц. С одной стороны, это плохо, а с другой очень даже хорошо. Во время использования данного мессенджера пользователь не будет получать посторонней информации. Также в WhatsApp есть возможность пересылать сообщения, но не более чем в 5 групповых чатах.

Многие пользователи WhatsApp давно знают, что этот мессенджер не самый безопасный, так как за ним могут наблюдать различные спецслужбы и киберпреступники. По этой причине многие не пользуются этим мессенджером для распространения информации о запрещенных действиях. Но это не останавливает преступников. Здесь так же могут общаться террористы, закладчики наркотиков, владельцы нелегального бизнеса, только они будут более осторожны в своих переписках и скорее всего будут использовать SIM-карты, не зарегистрированные на свое имя.

Самым большим минусом в WhatsApp является то, что посторонние люди могут добавлять обычных пользователей в общие беседы, без их согласия на это действие. Это происходит, когда номер телефона добавляют в контакты. Таким образом, неизвестные могут добавлять хоть все номера телефонов в мире. Конечно можно сразу же выйти из этой беседы, но иногда человек, может просто даже не заметить, что его добавили куда-либо, например, просто очень сильно занят и нет свободного времени, чтобы посмотреть в телефон.

Так в чем же кроется реальная угроза использования WhatsApp? Даже если пользователь использует эту версию только на личном компьютере или ноутбуке, то киберпреступники могут взломать этот ПК и просматривать личные сообщения. Также они могут шантажировать людей какой-либо полученной преступным путем информацией.

При использовании WhatsApp нужно учитывать, что любое оскорбление здесь можно распечатать и заверить у нотариуса, а с этим можно уже будет пойти в суд и подать заявление на оскорбившего человека. Пользователей WhatsApp могут даже посадить, если какие-либо отправленные ими текстовые или голосовые сообщения, фотографии или видеозаписи признают несущими экстремистский или террористический характер. Другой проблемой WhatsApp является обмен данными с Facebook. Передается номер телефона, а также данные о времени последнего использования сервиса. Из-за передачи данных Facebook сообщения в

WhatsApp могут оказаться под угрозой. Однако крупнейшая социальная сеть заверила, что личные данные пользователей не будут общедоступными на Facebook.

В WhatsApp 2017 и 2018 годах в Индии распространялись сфабрикованные подробности похищения детей и личные данные о «преступниках». Это привело к линчеванию обвиняемых, которые в итоге могли быть ни в чем не виноваты. Из-за этого WhatsApp оказался замешан в массовом насилии. В 2018 году в Бразилии во время выборов в данном мессенджере распространялась о кандидатах дезинформация, созданная бразильскими бизнесменами (<https://clck.ru/epB6j>).

Самой популярной и крупнейшей социальной сетью в Российской Федерации является ВКонтакте. Данный ресурс изначально позиционировал себя в качестве социальной сети студентов и выпускников российских вузов. Позже стал считаться современным, быстрым и эстетичным способом общения в сети. Для пользователей ВКонтакте доступен набор возможностей: создавать профиль с данными и информацией о себе, производить и распространять различный контент, быть в полном взаимодействии с другими пользователями приватно (общаться в личных сообщениях) и публично (публиковать различную информацию с помощью записей на «стене» личной страницы и сообществ), отслеживать новую информацию об активности друзей и групп через новостную ленту. Здесь есть возможность создавать общие беседы до 500 человек. Этим очень активно пользуются студенты и школьники, создавая чаты класса или группы. Это делается для того, чтобы можно было обмениваться домашним заданием, обсуждать выполнение поставленных задач и просто общаться. Многие компании друзей создают общие беседы для обмена данными со всеми своими товарищами, ведь легче отправить какую-либо информацию всем сразу, а не по отдельности каждому человеку.

ВКонтакте есть возможность комментировать различные публикации, тем самым показывая свое мнение о данном посте. Здесь можно смотреть видеозаписи и слушать аудиозаписи, выгруженные другими пользователями. Помимо этого, можно самому выкладывать на сайт свои аудиозаписи, часто так делают начинающие артисты в мире музыкального шоу-бизнеса. Также здесь можно играть в игры или использовать различные приложения, облегчающие жизнь человека. У социальной сети ВКонтакте есть свое мобильное приложение с аналогичным названием.

В мире социальную сеть ВКонтакте считают потворствующей пиратству, так как на этом сайте нелегально распространяется музыка (<https://clck.ru/epV8y>). Главной проблемой ВКонтакте является распространение порнографических материалов, в том числе детской. Из-за этого отдельные компании блокируют доступ с рабочих мест к данной социальной сети для своих сотрудников, а ограничения на доступ вводились многими учебными заведениями России. В ВКонтакте очень активно распространяются интимные фотографии граждан любого возраста. Обычно это происходит из-за того, что хакеры взламывают аккаунты пользователей и используют полученную информацию против собственника. Также киберпреступники могут совместно и незаметно пользоваться взломанным аккаунтом или даже удалить его.

С данной социальной сетью было очень много скандалов связанных с призывом людей к экстремистским действиям, вербованием в террористические организации или религиозные секты. Пользователи иногда даже не замечают, что на них могут оказывать давление с помощью разнообразных постов, например, с распространением ложной и недостоверной информацией. ВКонтакте может содержать в себе закрытые сообщества, которые используют для обсуждения политических действий. Главной проблемой данной социальной сети является то, что не всегда можно будет отыскать распространителей недостоверной информации и запрещенных файлов.

В современном мире главной проблемой является то, что люди пытаются защитить свои личные данные, полученную информацию, личные фотографии и видеозаписи, private переписки. Поэтому они создают защищенные мессенджер и социальные сети или же просто укрепляют защиту от киберпреступников. В погоне за безопасностью они забывают, что этой защитой могут пользоваться настоящие преступники. Именно из-за этого потом страдают дети и подростки, которым опытные и взрослые люди качественно промывают мозг, из-за этого начинают шантажировать полученной информацией пользователей, из-за этого многие люди подвергаются кибербуллингу⁵. Граждане различных стран иногда даже не задумываются по каким причинам блокируют различные мессенджеры и социальные сети в их регионах. А ведь правительство государства специально старается ограничить народ от плохих сайтов.

Литература

1. Иванова Е.Д., Шарыпова Т.Н. Безопасность переписки в социальных сетях и мессенджерах // Инновации. Наука. Образование. 2022. № 50. С. 1663-1666.
2. Пиджикян Д.С., Шарыпова Т.Н. Система защиты персональных данных в Российской Федерации // Наука и технологии: актуальные вопросы, достижения и инновации: Сб. научных трудов по материалам XXVI Международной научно-практической конференции (г. Анапа, 25 сентября 2021 г.). Анапа, 2021. С. 14-18.

© Ермишина Е.В., Шарыпова Т.Н., 2022

⁵ Англ. Cyberbullying – интернет запугивание.

УДК 004.942

Захаров Д.М., Слива М.В., канд. пед. наук
Нижневартковский государственный университет
г. Нижневартовск, Россия

ИСПОЛЬЗОВАНИЕ ДВИЖКА ROASSAL ДЛЯ ВИЗУАЛИЗАЦИИ СТАТИСТИЧЕСКИХ ДАННЫХ

В стремительно меняющемся мире существенно продолжает трансформироваться экономика, производство, торговля, финансы, сфера услуг. При этом большинство изменений в значительной степени основаны на технологиях обработки и анализа данных. Сегодня анализ данных позволяет не только оптимизировать операционную деятельность и анализировать риски, но и создавать новые ценностные предложения, открывать новые рынки [4].

Анализ данных включает в себя множество различных аспектов и подходов, и используется в различных сферах деятельности и науки. В современном мире анализ данных играет важную роль в области принятия решений – он позволяет выбирать самые эффективные решения с научной точки зрения [2].

Движок Roassal позволяет создавать визуализации данных максимально эффективно и кратко благодаря гибкости и динамичности среды Pharo, что является важным преимуществом в условиях сжатых сроков [1]. А богатый инструментарий Pharo позволяет оптимизировать процесс сбора и обработки данных с помощью использования широкого набора встроенных библиотек, пакетов и классов.

Zinc HTTP Components — это фреймворк с открытым исходным кодом, реализованный в среде Pharo Smalltalk и предназначенный для работы с протоколом HTTP. Он предлагает функционал как стороны клиента, так и стороны сервера. Одна из ключевых целей данного фреймворка состоит в обеспечении понятности и удобочитаемости создаваемых запросов. Авторы Zinc создавали свой продукт таким образом, чтобы любой человек, знакомый с базовыми знаниями языка Smalltalk и принципами HTTP, был способен понимать и учиться, просто глядя на реализацию запросов. Начиная с версии Pharo Smalltalk 1.3, Zinc является встроенным в среду фреймворком. Также он был портирован и на другие реализации Smalltalk, такие как Squeak и Gemstone.

Ключевым объектом для создания HTTP-запроса является класс «ZnClient». Чтобы сформировать запрос, необходимо создать экземпляр этого класса, и использовать его набор методов, чтобы провести требуемые настройки (<https://clck.ru/epUC5>). Далее представлен простой пример запроса, результатом выполнения которого будет экземпляр ZnResponse, чьи данные содержат строку с небольшим html-документом.

```
ZnClient new
  beOneShot;
  url: 'http://zn.stfx.eu/zn/small.html';
  get;
  response.
```

Сообщение «beOneShot» предписывает клиенту закрыть соединение после выполнения первого и единственного запроса.

Roassal – это свободный движок для визуализации данных с открытым исходным кодом, реализованный для диалектов языка Smalltalk – Pharo и VisualWorks. Roassal способствует быстрому созданию визуализаций путём выполнения последовательных коротких шагов длительностью от нескольких секунд до нескольких минут [3]. Также данный движок позволяет импортировать данные из широко распространённых текстовых форматов: XML, CSV, JSON.

Roassal поддерживает экспортирование визуализаций в различные форматы, включая PNG, SVG и HTML5. Экспорт в HTML позволяет открывать визуализацию в браузере с сохранением интерактивности. Этот процесс создаёт два файла: «index.html» и «roassal.js». Последний включает в себя реализацию функционала движка на языке JavaScript [1].

Далее представлен пример сценария на языке Pharo, загружающий CSV-данные из сайта «covidtracking.com» при помощи фреймворка Zinc, и использующий эту информацию для визуализации показателей госпитализации граждан США с диагнозом covid-19 в период с 17 марта 2020 года по 7 марта 2021 года.

Создание визуализации происходит поэтапно. Прежде всего необходимо загрузить данные с сервера.

```
| response covidCsvData table dateColumn hospitalizedColumn redundantColumns grapher data  
sortedValues |  
response := ZnClient new  
    beOneShot;  
    url: 'https://api.covidtracking.com/v1/us/daily.csv';  
    get;  
    response.  
covidCsvData := response entity string.
```

Далее полученные данные нужно обработать и отфильтровать пустые значения.

```
table := RTTabTable new  
    input: covidCsvData;  
    removeFirstRow;  
    yourself.  
  
dateColumn := table convertFromNameToIndexIfNecessary: #date.  
hospitalizedColumn := table convertFromNameToIndexIfNecessary: #hospitalizedCurrently.  
  
table  
    rejectRow: [ :r | (r at: hospitalizedColumn) isEmpty ];  
    convertColumnAsInteger: hospitalizedColumn;  
    convertColumn: dateColumn to: [ :v | Date readFrom: (v readStream) pattern:  
'yyuymmdd' ].
```

```
redundantColumns := (1 to: table numberOfColumns) asOrderedCollection
  removeAll: { dateColumn. hospitalizedColumn };
  yourself.

redundantColumns reverseDo: [ :c |
  table removeColumn: c
].

sortedValues := table values sorted: [ :row1 :row2 || date1 date2 |
  date1 := (row1 at: 1).
  date2 := (row2 at: 1).
  date1 <= date2
].
```

После подготовки данных создаётся визуализация.

```
grapher := RTGrapher new.
data := RTData new.

data dotShape
  size: 1;
  color: Color red.

data
  points: sortedValues;
  y: [ :row | row at: 2 ];
  connectColor: Color red.

grapher add: data.
grapher axisX
  title: 'Дата';
  noTick;
  noLabel.
grapher axisY title: 'Кол-во госпитализаций'.
grapher
  extent: 800@600;
  open.
```

Выполнение этого сценария открывает окно с графиком, изображенным на рисунке.

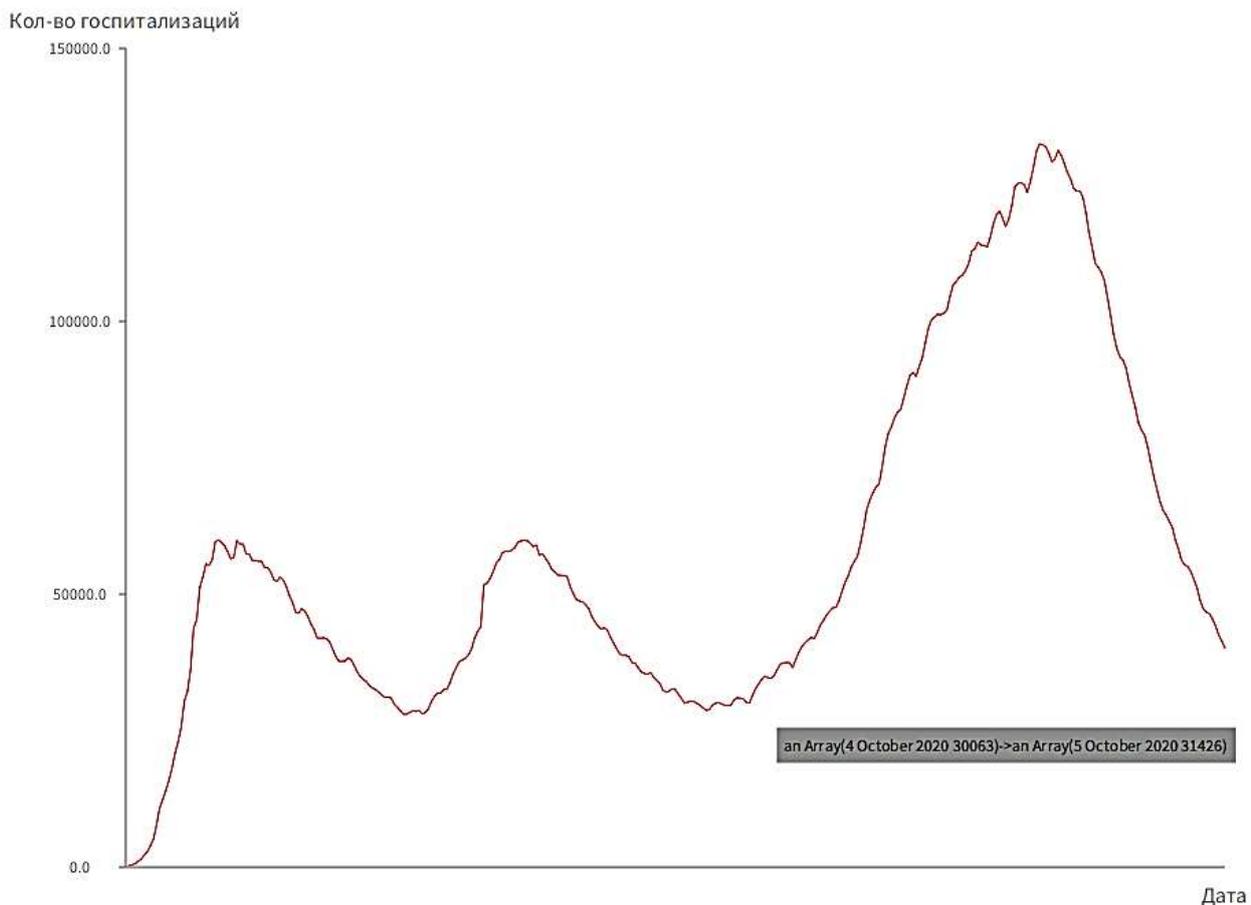


Рис. Визуализация статистических данных

При помощи указателя мыши можно просматривать конкретные значения на графике, как показано на данном рисунке. Таким образом, Roassal хорошо подходит для создания визуализаций в том числе и статистических данных.

Литература

1. Bergel A. Agile Visualization. Lulu.com, 2016.
2. Selene Xia B., Gong P. Review of business intelligence through data analysis // Benchmarking: An International Journal. 2014. Vol. 21. No. 2. Pp. 300-311. <https://doi.org/10.1108/BIJ-08-2012-0050>
3. Захаров Д.М. Движок визуализации данных roassal // XXIII Всероссийская студенческая научно-практическая конференция Низневартовского государственного университета (г. Низневартовск, 06–07 апреля 2021 года). Низневартовск, 2021. С. 118-121.
4. Соловьев В.И. Анализ данных в экономике: теория вероятностей, прикладная статистика, обработка и визуализация данных в Microsoft Excel: для направления бакалавриата «Экономика и управление». М.: КноРус, 2019. 498 с.

© Захаров Д.М., Слива М.В., 2022

ПЕРСОНАЛИЗАЦИЯ ОБУЧЕНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ЧАТ-БОТОВ

В последние годы интенсивно развиваются человеко-машинные диалоговые системы, использующие достижения искусственного интеллекта (ИИ), и, в частности, чат-боты – интеллектуальные компьютерные программы, способные взаимодействовать с людьми с помощью естественно-языковых интерфейсов. Так, к концу 2016 г. в различных сферах, таких, как маркетинг, образование, здравоохранение, и других, использовалось более 34 000 чат-ботов [5, с. 3]. Широко известны популярные голосовые помощники Apple Siri, IBM Watson Assistant, Google Assistant, Microsoft Cortana Home Assistant, Amazon Alexa, понимающие голосовые команды и отвечающие на естественном языке, способные выполнять достаточно нетривиальные задания по мониторингу домашних автоматизированных устройств, календарей, электронной почты и т.д. Крупнейшая маркетинговая компания MarketsandMarkets™, отслеживающая глобальные быстрорастущие рынки, прогнозирует, что объем мирового рынка разговорного искусственного интеллекта (Conversational AI) вырастет с 6,8 млрд долларов США в 2021 году до 18,4 млрд долларов США к 2026 году (<https://clck.ru/dXckF>).

Хотя российский рынок диалоговых систем искусственного интеллекта пока относительно невелик: по данным Just AI, одной из крупнейших российских компаний на рынке разговорного ИИ, в 2020 г. его объем составлял около 5,5 млрд руб. с учетом госзаказов, в ближайшие годы ожидается его значительный рост с выходом на уровень 41 млрд руб. к 2025 г. (<https://clck.ru/dXXk6>).

По-видимому, наиболее полный и обстоятельный обзор современного состояния работ по применению чат-ботов в образовании был дан в недавнем обзоре [8, стр. 11], где учитывались публикации из БД Web of Science за период с 1999 по 2020 гг. (всего более 500 работ). Согласно проведенному анализу, исследования по этой тематике все еще находятся на ранней стадии, хотя большинство авторов отмечают положительный эффект влияния чат-ботов на образовательный процесс, простоту и удобство их использования, отсутствие ограничений по времени и месту использования и другие достоинства, делая вывод о большом потенциале технологии чат-ботов в образовании.

Большинство российских вузов регулярно использует чат-ботов на своих сайтах и в социальных сетях для помощи абитуриентам и студентам в быстром поиске нужной информации, для ориентации в многообразии информационных ресурсов вуза, а также для доведения до студентов организационной и методической информации. Уровень технологий разработки чат-ботов в российских вузах характеризует, в частности, победа команды МФТИ в соревновании чат-ботов Conversational Intelligence Challenge в рамках крупнейшей

конференции по машинному обучению NIPS – Conference on Neural Information Processing Systems, США, декабрь 2021 г. (<https://clck.ru/dXXvE>).

Существуют различные классификации чат-ботов в зависимости от критериев классификации. Так, по цели использования чат-боты могут быть классифицированы на информационные, разговорные, и чат-боты, основанные на задачах [5, с. 3].

Когда пользователи общаются с чат-ботом, чтобы получить конкретную информацию, хранящуюся в фиксированном источнике, используются информационные чат-боты; разговорные чат-боты ведут диалог с пользователем на естественном языке. Наконец, чат-боты на основе задач выполняют различные фиксированные функции, такие, как помощь в бронировании отелей, приобретении билетов и т.д.

В зависимости от применяемых алгоритмов и методов чат-боты подразделяются на чат-боты на основе правил, и чат-боты, использующие методы машинного обучения. Чат-боты на основе правил сопоставляют введенные пользователем данные с шаблоном правила и выбирают predetermined ответ из набора ответов с использованием алгоритмов сопоставления шаблонов. Этот подход ограничен заданным шаблоном, но имеет достоинство быстрого отклика, поскольку более глубокое синтаксическое или семантическое исследование вводимого текста не выполняется, в отличие от подхода на основе машинного обучения. Преимуществом последнего является большая гибкость и «антропоморфность» ответа; он способен извлекать контент из пользовательского ввода и учитывать весь контекст диалога, а не только текущий ход, и не требует predetermined ответа для каждого пользовательского запроса. Диалоговые системы на основе этого подхода используют последние достижения Data Science и методов обработки естественного языка, таких как NLU (Natural Language Understanding) и NLP (Natural Language Processing), нейросети для распознавания речи и обработки текстов, а также другие инструменты ИИ [5, с. 18].

Быстрому развитию технологии чат-ботов и, в частности, их растущему использованию в образовательном процессе способствовали их несомненные достоинства, среди которых круглосуточная доступность для обучаемых, относительно низкая стоимость разработки и использования, стабильность работы и быстрое время отклика, наконец, возможность масштабирования без дополнительных затрат при росте числа пользователей [1, с. 32].

Анализ результатов работы чат-ботов в образовании показал, что они использовались в различных аспектах, включая преподавание и обучение, администрирование и управление процессом обучения, сбор и обработка данных, в том числе, о проблемах, возникающих у обучаемых и преподавателей, консультации и репетиторство, и т.д. [9, с. 4].

Чат-боты в образовании получают все большее распространение, поскольку могут мгновенно предоставлять студентам всю необходимую информацию по курсу, такие, как его содержание, критерии оценивания, сроки выполнения заданий, рекомендации и учебные материалы. Но, помимо предоставления этих стандартных функций, развитие технологий дает возможность использования систему чат-ботов как эффективный инструмент повышения уровня индивидуализации процесса обучения, его персонализации. Эти системы могут не только повысить эффективность и комфортность обучения, мотивированность

студентов и их поддержку, но и существенно снизить административную нагрузку преподавателей. Многочисленные иные способы взаимодействия обучаемых друг с другом, с преподавателями и администрацией (например, общение по электронной почте, через LMS, через сайты учебных заведений и социальные сети), к сожалению, не дают достаточных инструментов для создания среды индивидуализированного обучения, способной учитывать особенности обучаемых. По сравнению с ними чат-боты потенциально более дружелюбны и привлекательны для пользователей, поскольку предлагают им удобную и эффективную помощь при общении, дают более релевантные ответы, непосредственно реагируя на их проблемы [5, с. 1].

Чат-боты способны персонализировать процесс обучения и сделать учебные материалы доступными для студентов в любом месте и в любое время. Технология чат-ботов может предоставить учащимся, таким образом, более персонализированную и комфортную среду обучения.

Под персонализированным обучением понимается процесс обучения, в котором его темп, формы и сам подход к обучению оптимизированы к потребностям каждого обучающегося. Цели обучения, его содержание и последовательность могут варьироваться в зависимости от пожеланий обучаемых, естественно, в определенных рамках ограничений, в том числе административных, вытекающих из содержания и программы курса. Такое обучение допускает и учебные мероприятия, инициируемые самими обучающимися, обусловленные их интересами и потребностями. Предпосылкой для персонализированного обучения является создание индивидуальной учебной программы и учебных целей, основанных на потребностях каждого отдельного студента.

Персонализация как в традиционном, так и в электронном обучении, предполагает дифференцированность обучения, расширение автономности обучающихся и адаптивность обучения [4, с.185].

Дифференцированное обучение подразумевает таксономию обучающихся на кластеры с учетом их уровня, степени мотивированности, интересов и т.д. Для каждой из таких подгрупп затем подбираются адекватные формы и методы учебной работы. Этот путь обычно характеризуется меньшими затратами по сравнению с остальными формами.

Для расширения автономности обучающегося необходима разработка его индивидуальной учебной траектории. Концепция адаптивного обучения предполагает, помимо индивидуальной учебной траектории, возможность выбора обучающимися подходящих инструментов и сред обучения, объема получаемых знаний и т.д., что требует привлечения административного ресурса.

В современных педагогических исследованиях адаптивное обучение рассматривается и как единство технологий и процесса обучения, достигаемое на основе динамической корректировки различных аспектов представления учебного контента с учетом предпочтений обучаемых за счет постоянной обратной связи с ними и контроля характеристик процесса обучения [3, с. 35].

Пользователи в «веб-аудиториях» часто ощущают дискомфорт в отсутствие поддержки и обратной связи. Этот разрыв чат-боты помогают преодолевать, функционируя в качестве учебных ассистентов. Основной функцией обучающих чат-ботов является сопровождение учащегося в течение всего процесса обучения, варьируя сложность объяснений и заданий. Например, предоставление студенту апостериорных данных о процессе обучения может дать ему представление о ресурсах повышения его успеваемости, открыть новые возможности для формирования обратной связи и саморефлексии, а значит, и для его развития. Такая обратная связь создает условия для самостоятельного обучения и эффективного метакогнитивного контроля над процессом обучения [10, с. 2].

Развитие образовательных технологий в первую очередь ориентируется на студентов. Новые технологии, несмотря на их инновационную мощь, пока не меняют существенно роль преподавателей. Даже в среде цифрового обучения типичные учебные действия – предоставление студентам обратной связи, их мотивация или адаптация содержания курса к конкретным студенческим группам по-прежнему возложены исключительно на преподавателей [10, с. 1]. Вместе с тем, технология чат-ботов имеет потенциал для предоставления быстрых и персонализированных услуг не только студентам, но и преподавателям, а также административному персоналу.

Однако использование чат-ботов в процессе обучения может создать ряд проблем. Чтобы их избежать, функции любого чат-бота должны быть подробно описаны; пользователи до взаимодействия с чат-ботом должны четко представлять свои возможные действия, варианты реакции чат-бота, возможные последствия и риски взаимодействия с чат-ботом. В свою очередь, понимание ожиданий пользователя имеет решающее значение для предотвращения злоупотребления доверием пользователя [9, с. 7].

Наблюдаемый прогресс в персонализации образования связан в настоящее время в большей степени с развитием технологий, теоретическому анализу их использования и соответствующей корректировке педагогических аспектов обучения пока уделяется значительно меньшее внимание. Однако, именно разработка современной научно-педагогической теории персонализации образования должна способствовать лучшему пониманию его перспектив [6, с. 15].

Заметим также, что российское образование выполняет социальный и государственный заказ на подготовку профессиональных кадров определенной квалификации, что накладывает определенные ограничения на степень индивидуализации обучения. Кроме того, в реалиях российского образования процесс его персонификации связан с рисками различного характера, как для обучаемых, так и для преподавателей и административного персонала, которые необходимо прогнозировать и нивелировать, создавая соответствующие механизмы их регуляции [2, с. 3].

Прогресс в использовании чат-ботов в большей степени связан с появлением удобных инструментов (платформ) для создания чат-ботов без необходимости работы с кодом напрямую: Chatfuel, ManyChat, Dialogflow и т. д. [1, с. 29]. Существуют также языки сценариев, такие как язык разметки ИИ (Artificial Intelligence Markup Language – AIML)

ChatScript и RiveScript, которые применяются для создания чат-ботов. Эти сценарии содержат диалоговый контент и поток, которые затем должны быть отправлены программе интерпретатора или механизму правил, чтобы запустить чат-бота. Интерпретатор решает, как продолжать диалог, сопоставляя пользовательские высказывания с шаблонами в сценариях. Из отечественных таких разработок одной из наиболее известных и широко используемых как в России, так и за рубежом является платформа Just AI Conversational Platform компании Just AI, среднегодовой оборот которой составляет более 500 млн руб. (<https://just-ai.com/ru>).

В последнее время в связи с развитием систем ИИ появилась возможность разработки человеко-машинных диалоговых систем, и, в частности, чат-ботов, учитывающих особенности психоэмоционального состояния обучаемого. Включение эмоционального фона в общение обучаемого и чат-бота, с одной стороны, несомненно, на порядок повышает трудоемкость разработки, но, с другой стороны, существенно повышает привлекательность использования чат-ботов для обучаемых. Задача моделирования эмоциональной составляющей диалога – одна из тех сложных задач, которые называются «плохо формализуемыми». Возможность восприятия чат-ботом эмоционального состояния обучаемого по особенностям его запроса – один из важнейших аспектов гуманизации чат-бота, и уже существуют подходы к созданию «эмоциональных» чат-ботов [7, с. 8].

В МИЭМ НИУ ВШЭ под руководством кандидата технических наук Королева Д.А. разрабатывается система чат-ботов для использования в учебном процессе. В рамках проектной деятельности, являющейся частью учебного плана на образовательных программах «Информатика и вычислительная техника», «Информационные технологии и системы связи» и «Прикладная математика», используются инструменты проектной деятельности. Проекты в МИЭМ НИУ ВШЭ ведутся по упрощенной и приспособленной под реалии университетского образования методологии разработки Agile. Для совместной разработки проекта используются методы Scrum, например, разбиение разработки на спринты длительностью в месяц (отличие от оригинального Scrum обусловлено различиями в длительности учебного процесса и постоянной full-time работы). На каждый спринт ставятся свои независимые задачи, которые должны быть завершены именно в этот спринт; по окончании каждого спринта в идеальной модели проектного обучения команда должна представить отчуждаемую и работоспособную часть разработки.

Для удобства реализации проектов и их типизации в МИЭМ НИУ ВШЭ применяются специализированные инструменты проектной деятельности. Каждый из них имеет свою область применения и конкретные задачи:

- GitLab является веб-инструментом, представляющим систему управления репозиториями кода для Git с системой отслеживания ошибок, CI/CD-пайплайном и другими функциями. В рамках проектного обучения используется командами разработчиков ПО для ведения совместной разработки.

- Taiga\Trello\Wekan – проектные задачи обязаны вестись в некоем унифицированном формате для получения отчетности выполнения, отслеживания результативности каждого из участников и прогресса в спринте в целом. Для этих целей руководитель каждого из

проектов выбирает наиболее удобную систему из перечисленных. Все они в своей основе имеют технику Канбана. Канбан – одна из наиболее наглядных систем разработки, показывающей, что необходимо создать, когда и сколько. В методике Канбан условная доска делится на необходимое количество колонок, обозначающих жизненные этапы задачи. Создаваемые задачи перемещаются в колонках по мере выполнения этапов.

- Wiki MIEM – сервис, использующийся для создания и хранения проектной документации. Здесь проектные команды создают документации администратора, разработчика, пользователя и другие. Инструмент поддерживает различные варианты создания документов (Markdown, HTML)

Для создание рабочей атмосферы и унификации общения по проектам в МИЭМ НИУ ВШЭ используется корпоративный мессенджер Zulip.

Примерно 80% опрошенных студентов с различных курсов обучения в той или иной степени испытывали проблемы при использовании проектных инструментов. Для решения этих проблем и облегчения «входа» студентов в проекты создан онлайн-курс «Онлайн обучение в МИЭМ». Курс состоит из письменных материалов-инструкций, объясняющих основные принципы проектной модели. Особенностью курса является добавление в него интерактивной составляющей, позволяющей персонализировать обучение. По каждому из перечисленных выше инструментов создан урок в чат-боте, целью которого является знакомство с интерфейсом инструмента и основными сценариями его использования.

В ходе урока чат-бот дает задания, которые необходимо выполнять в изучаемом инструменте; например, если изучается Trello, то первым заданием будет «Создай карточку в Trello». В процессе выполнения задания осуществляется его мониторинг – действия обучающегося находятся под постоянным контролем со стороны чат-бота. В момент получения сигнала о совершенном действии чат-бот проверит правильность выполнения задания. При успешном выполнении предлагается следующий, более сложный шаг. При возникновении проблем чат-бот подсказывает, как можно выполнить вызвавшее трудности задание и предлагает попробовать вновь. По истечении трех попыток, сопровождающихся подсказками, чат-бот предлагает выполнить задание меньшей сложности, если такое существует. По завершении урока выставляется оценка в LMS.

В основе разрабатываемого подхода находятся механизмы обучения непосредственно на изучаемой платформе: студент учится там же, где в будущем будет работать в рамках проектной деятельности. Это позволяет отработать реальные кейсы в рабочей среде под постоянным контролем чат-бота.

Применение API-технологий позволяет отслеживать все действия, совершенные учеником на платформе. Благодаря этому существует возможность детектирования ошибок и предложения подсказок по их исправлению.

Данная система обучения была впервые опробована в 2020-2021 учебном году на студентах первого курса образовательной программы «Информатика и вычислительная техника» в рамках курса «Проектный семинар». Прохождение уроков системы обучения было обязательным условием успешного завершения курса, поэтому более 150 учащихся

приняли участие в тестировании созданной системы. Система показала достаточно хорошие результаты в части улучшения осведомленности о проектных инструментах. Благодаря этому в 2021-2022 учебном году доработанную систему начали использовать и на образовательной программе «Прикладная математика».

На данном этапе ведется разработка урока по работе в LMS-системе SmartLMS, которая используется в НИУ ВШЭ.

После первого тестирования системы и сбора фидбека были выявлены блоки, требующие доработки. Среди них можно выделить некоторое неудобство представления результатов прохождения курса для преподавателя. В связи с этим планируется доработка блока, осуществляющего сбор и визуализацию данных по освоению уроков студентами.

Кроме того, была осознана необходимость включения третьей стороны в процесс взаимодействия «преподаватель-студент», в связи с чем в настоящее время разрабатывается концепция «Родительского контроля» процесса обучения.

Таким образом, разрабатываемая нами система чат-ботов показала свою полезность в создании более индивидуализированной среды для студентов при освоении ряда учебных курсов и имеет перспективы для развития в качестве одного из инструментов создания персонализированной среды обучения в НИУ ВШЭ.

Литература

1. Джанарсанам С. Разработка чат-ботов и разговорных интерфейсов. М.: ДМК-Пресс, 2019. 340 с.
2. Конищева Е.В. Регулирование института высшего образования в современных условиях: риски персонализации // Общество: социология, психология, педагогика. 2020. №10(78). С. 39-43. <https://doi.org/10.24158/spp.2020.10.5>
3. Кравченко Д.А., Блескина И.А., Каляева Е.Н., Землякова Е.А., Аббакумов Д.Ф. Персонализация в образовании: от программируемого к адаптивному обучению // Современная зарубежная психология. 2020. Т. 9. № 3. С. 34-46. <https://doi.org/10.17759/jmfp.2020090303>
4. Прохорова М.П. Шкунова А.А. Гуреева Е.П. Средства персонализации образовательного процесса в рамках электронных курсов // Проблемы современного педагогического образования. 2021. № 71(3). С. 183-187.
5. Adamopoulou E., Moussiades L. Chatbots: History, technology, and applications // Machine Learning with applications. 2020. Vol. 2. 100006. <https://doi.org/10.1016/j.mlwa.2020.100006>
6. Bartolomé A., Castañeda L., Adell J. Personalisation in educational technology: the absence of underlying pedagogies // International Journal of Education Technology in Higher Education. 2018. Vol. 15. Article number: 14. <https://doi.org/10.1186/s41239-018-0095-0>
7. Huo P., Yang Y., Zhou J., Chen C., He L. TERG: Topic-Aware Emotional Response Generation for Chatbot // 2020 International Joint Conference on Neural Networks (IJCNN). 2020. P. 1-8. <https://doi.org/10.1109/IJCNN48605.2020.9206719>



8. Hwang G.-J., Chang C.-Yi. A review of opportunities and challenges of chatbots in education // Interactive Learning Environments. 2021. <https://doi.org/10.1080/10494820.2021.1952615>

9. Okonkwo C.W., Abejide A.-I. Chatbots applications in education: A systematic review // Computers and Education: Artificial Intelligence. 2021. Vol. 2. 100033. <https://doi.org/10.1016/j.caeai.2021.100033>

10. Wollny S., Schneider J., Di Mitri D., Weidlich J., Rittberger M., Drachsler H. Are We There Yet? – A Systematic Literature Review on Chatbots in Education Front // Frontiers in Artificial Intelligence. 2021. 4:654924. <https://doi.org/10.3389/frai.2021.654924>

© Заякина А.М., 2022

УДК 62-503.56

Иванова В.Р., Шарифуллин Б.Р.

Казанский государственный энергетический университет
г. Казань, Россия

ИНСТРУМЕНТЫ ИНТЕРНЕТА ВЕЩЕЙ ДЛЯ АВТОМАТИЗАЦИИ ВНУТРЕННИХ ПРОЦЕССОВ ОРГАНИЗАЦИЙ

Основным способом фактического контроля за сохранностью имущественных ценностей и средств является организация процесса сличения фактических данных с данными бухгалтерского учёта, называемая инвентаризацией. Указанный процесс проводится разными путями, к ним относятся пересчёт, измерения, взвешивание и т.п. Количество инвентаризационных мероприятий в году устанавливается руководителем организаций. Важность этой процедуры заключается в выявлении несоответствий и расхождений. Известно, (согласно Приказу Минфина РФ от 13.06.1995 № 49 (ред. от 08.11.2010) «Об утверждении Методических указаний по инвентаризации имущества и финансовых обязательств») что инвентаризации подлежит все имущество организации независимо от его местонахождения и все виды финансовых обязательств. Наименования инвентаризируемых ценностей и объектов, их количество указывают в описях по номенклатуре и в единицах измерения, принятых в учете [2].

Проведение инвентаризации – это длительный процесс, который сопровождается финансовыми затратами в виде дополнительной оплаты сверхурочных сотрудникам, проводящим проверку. Такими сотрудниками являются делегаты, входящие в состав ревизионной комиссии, на которую возлагаются эти обязанности соответствующим приказом руководителя, а также материально ответственные лица. В результате проведения мероприятий проверки создаются инвентаризационные описи, которые заполняются ручным способом либо при помощи вычислительных средств.

Безусловно такой трудоемкий процесс при наличии большого количества единиц, подлежащих учету, имеет свои недостатки. К ним можно отнести во-первых – большие временные затраты на выполнение мониторинга, а во-вторых – значительный риск появления ошибок, спровоцированных человеческим фактором из-за большого количества однотипных данных. Для решения указанных проблем применяют новые технологии идентификации, что существенно снижает нагрузку и облегчает процесс

В настоящее время существуют и разработаны системы, учитывающие размерные сетки, вес и т.п. Они реализуются по принципу штрихкодирования с использованием терминалов сбора данных. Учет товара происходит таким образом, что аудиторы с устройствами производят сканирование продукции путем поднесения терминала в единицу товара. В этом случае нужно сказать о необходимости работы двух бригад, задачей которых будет получение двух комплектов документов, где будет отражаться все остатки и результаты сравнений инвентаризаций. Подобный подход имеет преимущества, согласно которому вероятность ошибки подсчета будет снижена до минимума. Недостатком такой реализации техники инвентаризации выступают дополнительные временные издержки [3].

При использовании вычислительных машин инвентаризацию проводить проще, а обеспечить исключение ошибок до 0,1 %. Широко применяется пересчет в прикассовой зоне то есть кассир проводит все товары через кассу путем сканирования штрихкода товара с помощью оптической системы идентификации и обеспечивает тем самым ввод в учетную систему, где формируется база единиц товара и производится сравнительный учет [5].

Целью данной работы стало исследование возможности применения в процессе инвентаризации усовершенствованных идентификаторов и вопроса разработки алгоритма и выбора способов хранения данных. Для исследования были выбраны учебные помещения ФГБОУ ВО «Казанского государственного энергетического университета». Инвентаризация здесь проводится непосредственно путем сверки фактического наличия имущества, после данные вносятся в инвентаризационные описи. Там указывают наименование инвентаризируемых ценностей и объектов, их количество в единицах измерения, принятых в учете. По нашему мнению есть необходимость усовершенствования процесса инвентаризации с целью экономии времени и исключения ошибок. Таким образом, предполагается, что суть процесса будет заключаться в его автоматизации.

На первом этапе предполагается решение вопроса маркировки товарно-материальных ценностей и основных средств. В качестве маркировки предложено сделать виниловые стикеры с нанесенными на них идентификаторами. Использование стикеров сократит время нанесения инвентарного номера на объекты основных средств и объекты материальных ценностей. А также сделает инвентаризационный процесс более эстетичным в силу отсутствия лакокрасочных изделий. Первый этап становится самым затратным по времени, однако в последующем облегчит работу. На втором этапе производится непосредственно инвентаризация, в течении которой происходит считывание идентификатора с объектов основных средств. А третий этап посвящен выгрузке и анализу проведенной работы и автоматическому формированию всех документов.

В качестве идентификатора для выбранных помещений предлагается использовать новый тип, который носит название *QR*-код. В настоящее время он используется все чаще и становятся все более популярными, поэтому многие смартфоны теперь выпускаются со встроенным сканером *QR*-кода. *QR*-код представляет собой двухмерный штрихкод, который позволяет кодировать до 7000 знаков. Данная технология обладает рядом преимуществ перед альтернативными способами хранения и отображения информации. Первым из них является удобство сканирования под любым углом. Архитектура идентификатора несет закодированную информацию для декодирования которой используются следующие поля:

- поисковые узоры – три квадрата по углам;
- выравнивающие узоры – модули;
- полосы синхронизации – линии чередующихся между собой черных и белых точек;
- код маски и уровня коррекции;
- код версии;
- отступ вокруг кода.

Технологии *QR*-кода распространяются бесплатно, существует множество бесплатных вариантов программного обеспечения, позволяющего генерировать и считывать *QR*-коды. Все ПО можно условно разделить на два вида: стационарные и мобильные. Для считывания с помощью мобильных устройств необходимо приложение с фотокамерой. К операционным системам, поддерживающим установку приложений для работы с *QR*-кодировкой, можно отнести: *Android*, *iOS (Apple)*, *Symbian (Nokia)*, *Windows Phone*, *Windows Mobile*, *Blackberry OS (Blackberry)*. Так *QR*-код постепенно проник во все отрасли промышленности.

Его создатели из *Denso Wave* сделали технологию создания и чтения *QR*-кодов общедоступной отказавшись от своих патентных прав. Это означает, что даже сейчас любой может создать и использовать свой собственный *QR*-код. Для генерирования изображений с кодом могут быть использованы специализированные интернет-сайты. К числу таких сайтов можно отнести, например, сайт qr-code-generator.com. После выполнения трех операций, сгенерирует и предложит сохранить изображение с кодом. Стоит отметить, что размеры изображения позволяют напечатать код в достаточно большом формате и использовать его, например, на вывесках и плакатах [1].

В работе была проведена генерация примерного содержания идентификатора для нужд инвентаризации выбранного объекта. Таким образом определено, что *QR*-код будет содержать следующую информацию:

- инвентаризационный номер объекта;
- дату установки;
- название;
- производитель;
- ФИО материально-ответственного лица.

Согласно перечню был сгенерирован *QR*-код на сайте <https://qr-online.ru/> (рис. 1).



Рис. 1. Пример *QR*-кода для инвентаризации учебной аудитории Д-728 в ФГБОУ ВО КГЭУ

После присвоения каждому объекту своего *QR*-кода, приступили к описанию алгоритма системы инвентаризации.

Согласно ему инвентаризация будет проходить раз в полгода согласно этапам описанным ранее. После сканирования данные о каждом объекте будут передаваться в облачное хранилище, представляющее собой сервер, к которому можно будет обращаться по запросу (согласно топологии клиент-сервер). База данных будет выстроена с помощью офисной программы Microsoft Excel. Последняя имеет простой интерфейс и удобство пользования [7]. Содержание отображаемой информации на сервере будет выглядеть следующим образом: в колонке слева, будет отображаться информация об имеющемся оборудовании в учебных аудиториях (рис. 2), а в колонке справа будет появляться информация после сканирования. После сканирования предмет будет отображаться в списке. Соответственно, автоматически в случае отсутствия в наличии предмета строка в правой колонке подсветится желтым цветом для легкости обнаружения несовпадений (рис. 3).

Бухгалтерский Учет			Данные СКАНИРОВАНИЯ		
Штрихкод	Наименование	Кол-во	Штрихкод	Наименование	Кол-во
55002964	Стул мягкий без ручек офисный изванов	1			
207RAGK0Y023	Монитор ЖК LG FLATRON IPS236V-PNX	1			
55002965	ЗЕРКАЛО овальное в фигурной металлической рамке	1			
3982611180	XEROX Принтер Phaser 3250N	1			
26245	Монитор ЖК PHILIPS Z-TECH IP26245	1			
16127201100289	Системный БЛОК компьютера ПЗЕМ Z-TECH	1			
55002967	Блок бесперебойного питания APC BACK-UPS RS 500	1			
55002968	Доска офисная для маркеров	1			
55002969	Стул мягкий без ручек офисный	1			
55002970	Стул мягкий вращающийся офисный с подлокотниками Петров А.В.	1			
55002971	Стул мягкий вращающийся офисный с подлокотниками Симонов И.Г.	1			
55002972	Телефон офисный проводной PANASONIC KX-TS2350RU	1			
55002973	Телефон офисный проводной SIEMENS 5015 0683	1			

Рис. 2. Содержание отображаемой информации на сервере

Бухгалтерский Учет			Данные СКАНИРОВАНИЯ		
Штрихкод	Наименование	Кол-во	Штрихкод	Наименование	Кол-во
55002964	Стул мягкий без ручек офисный изванов	1	55002964	Стул мягкий без ручек офисный изванов	1
207RAGK0Y023	Монитор ЖК LG FLATRON IPS236V-PNX	1	207RAGK0Y023	Монитор ЖК LG FLATRON IPS236V-PNX	1
55002965	ЗЕРКАЛО овальное в фигурной металлической рамке	1	55002965	ЗЕРКАЛО овальное в фигурной металлической рамке	1
3982611180	XEROX Принтер Phaser 3250N	1	3982611180	XEROX Принтер Phaser 3250N	1
26245	Монитор ЖК PHILIPS Z-TECH IP26245	1	26245	Монитор ЖК PHILIPS Z-TECH IP26245	1
16127201100289	Системный БЛОК компьютера ПЗЕМ Z-TECH	1	16127201100289	Системный БЛОК компьютера ПЗЕМ Z-TECH	1
55002967	Блок бесперебойного питания APC BACK-UPS RS 500	1	55002967	Блок бесперебойного питания APC BACK-UPS RS 500	1
55002968	Доска офисная для маркеров	1	55002968	Доска офисная для маркеров	1
55002969	Стул мягкий без ручек офисный	1	55002969	Стул мягкий без ручек офисный	1
55002970	Стул мягкий вращающийся офисный с подлокотниками Петров А.В.	1	55002970	Стул мягкий вращающийся офисный с подлокотниками	1
55002971	Стул мягкий вращающийся офисный с подлокотниками Симонов И.Г.	1	55002971	Стул мягкий вращающийся офисный с подлокотниками	2
55002972	Телефон офисный проводной PANASONIC KX-TS2350RU	1			
55002973	Телефон офисный проводной SIEMENS 5015 0683	1			

Рис. 3. Содержание после проведения инвентаризационных мероприятий

В результате исследования пришли к выводу о том, что нанесенный стикер с QR-кодом ведущим на веб-страницу позволит обновлять информацию о предмете. Таким образом, даже в небольшом отделе, перемещение служебного оборудования между сотрудниками не будет вызывать проблемы при очередной инвентаризации. Замена инвентарного номера на QR-этикетку позволит: сократить время на поиск оборудования в несколько раз; сократить трудовые и временные затраты на проведение инвентаризационного учёта; обеспечить оперативный доступ к исчерпывающей информации об объекте, подлежащем

инвентаризационному учёту; снизит риска появления ошибок, спровоцированных человеческим фактором при работе с однотипными данными (<https://clck.ru/et2wp>).

Для упрощения и усовершенствования системы следующим этапом разработки станет реализация мобильного приложения не только для сканирования и обновления информации о предмете, но и для возможности доставки уведомлений о необходимости следующих манипуляций. А также применение радиочастотных считывателей и радиометок [6]. Подобная система будет функционировать самостоятельно согласно туманным вычислениям без центра обработки данных, при необходимости передачи информации в облачную систему будет использован стандартный протокол. Организация такой сети позволит с легкостью по запросу стать частью вычислительной сети физических предметов – Интернета вещей (IoT) [4].

Исследования выполнены при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках госзадания на выполнение НИР по теме «Распределенные автоматизированные системы мониторинга и диагностики технического состояния воздушных линий электропередачи и подстанций на основе технологии широкополосной передачи данных через линии электропередач и промышленного интернета вещей» (номер темы 075-03-2021-175).

Литература

1. Николаенко Г.А., Евсикова Е.В. Перспективы использования QR-кодировки в академической сфере // Социология науки и технологий. 2015. Т. 6. № 2. С. 109-118.
2. Иванова В.Р. Об актуальности современных информационных и телекоммуникационных технологий в образовании // Стратегические ориентиры развития высшей школы: Сб. научных трудов участников II Национальной научно-практической конференции (г. Москва, 25 ноября 2020 года). М., 2020. С. 95-102.
3. Иванова В.Р. О принципах построения цифровых промышленных сетей // Novaum. 2020. № 25. С. 7-9.
4. Ли П. Архитектура интернета вещей. М.: ДМК-Пресс, 2019. 454 с.
5. Чудинов И.Л., Осипова В.В. Информационные системы и технологии. Томск: Изд-во Томского политехнического университета, 2013. 145 с.
6. Иванова В.Р., Гусамов Д.И. Аналитический обзор IoT технологии // Приборостроение и автоматизированный электропривод в топливно-энергетическом комплексе и жилищно-коммунальном хозяйстве: Мат-лы VII Национальной научно-практической конференции. Казань, 2021.
7. Конноли Т., Бегг К. Базы данных. Проектирование реализация и сопровождение. М.: Вильямс, 2017. 1440 с.

© Иванова В.Р., Шарифуллин Б.Р., 2022

УДК 004.03

Исаева Е.М.¹, Моисеева Т.В.², Поляева Н.Ю.^{1,2}, Точильникова К.А.¹
Поволжский государственный университет телекоммуникаций и информатики¹
Институт проблем управления сложными системами РАН
Самарского федерального исследовательского центра РАН²
г. Самара, Россия

СТРУКТУРА СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО РАЗРЕШЕНИЮ ПРОБЛЕМНЫХ СИТУАЦИЙ

Проблемные ситуации, возникающие в жизни людей, могут создавать им значительные неудобства, препятствовать эффективной работе, снижать качество жизни. Для эффективного разрешения проблемных ситуаций применяется теория интересубъективного управления, разработанная В.А. Виттихом [1-8]. В ее основе лежит организация взаимодействия людей для поиска выхода из проблемной ситуации. Для того чтобы привлечь дополнительные ресурсы, человек вынужден (но по собственной воле и желанию) искать других людей, оказавшихся в той же проблемной ситуации, обсуждать с ними проблему и совместно искать выход.

Под проблемной ситуацией в интересубъективном управлении понимают возникающее в результате активного взаимодействия человека и внешних условий противоречие между определенной социальной потребностью и наличием средств ее удовлетворения, когда неудовлетворительное состояние дел уже осознано, но не ясно, что нужно делать для его изменения, что вызывает интеллектуальное затруднение актора и заставляет его начать действовать, т.е. для того, чтобы ситуация оказалась проблемной, необходимо выполнение ряда условий:

- человек осознает наличие некоторого затруднения;
- он не может найти выход без посторонней помощи;
- он не желает смириться с ситуацией и существовать дальше, не пытаясь найти выход из нее;
- человек становится актором, который начинает действовать.

Понять проблему сразу часто достаточно трудно, собственных ресурсов актору может не хватить, иначе он нашел бы выход самостоятельно, поэтому может потребоваться поиск единомышленников и сбор дополнительной информации для проведения анализа проблемы.

Как только акторы начинают осознавать проблемную ситуацию, они приступают к обсуждению, чтобы согласованно понять ситуацию, достигнуть взаимопонимания, а затем найти совместными усилиями решение по её урегулированию. Обсуждение побуждает акторов к совершению действий. Если обеспечить возможность каждому актору ознакомиться с точками зрения его коллег, поделиться уже имеющейся информацией, то это может привести к изменению его собственных взглядов на проблемную ситуацию, дополнить их, а в дальнейшем даст возможность разрешить проблемную ситуацию. Для этого могут применяться системы поддержки принятия решений (СППР) [9-12], целью которых является повышение эффективности принятия решений.

Укрупненная структура СППР, реализуемая в виде сайта, представлена на рисунке. Главным вопросом, который встал при разработке СППР, стал вопрос «что система должна делать?», то есть какие функции выполнять. СППР при интересубъективном управлении разрешением проблемных ситуаций должна уметь:

- представлять подобные прецеденты;
- предлагать средства поддержки акторов;
- получать и хранить информацию об акторах, присоединившихся к обсуждению конкретной проблемной ситуации;
- классифицировать проблемные ситуации (ПрС).

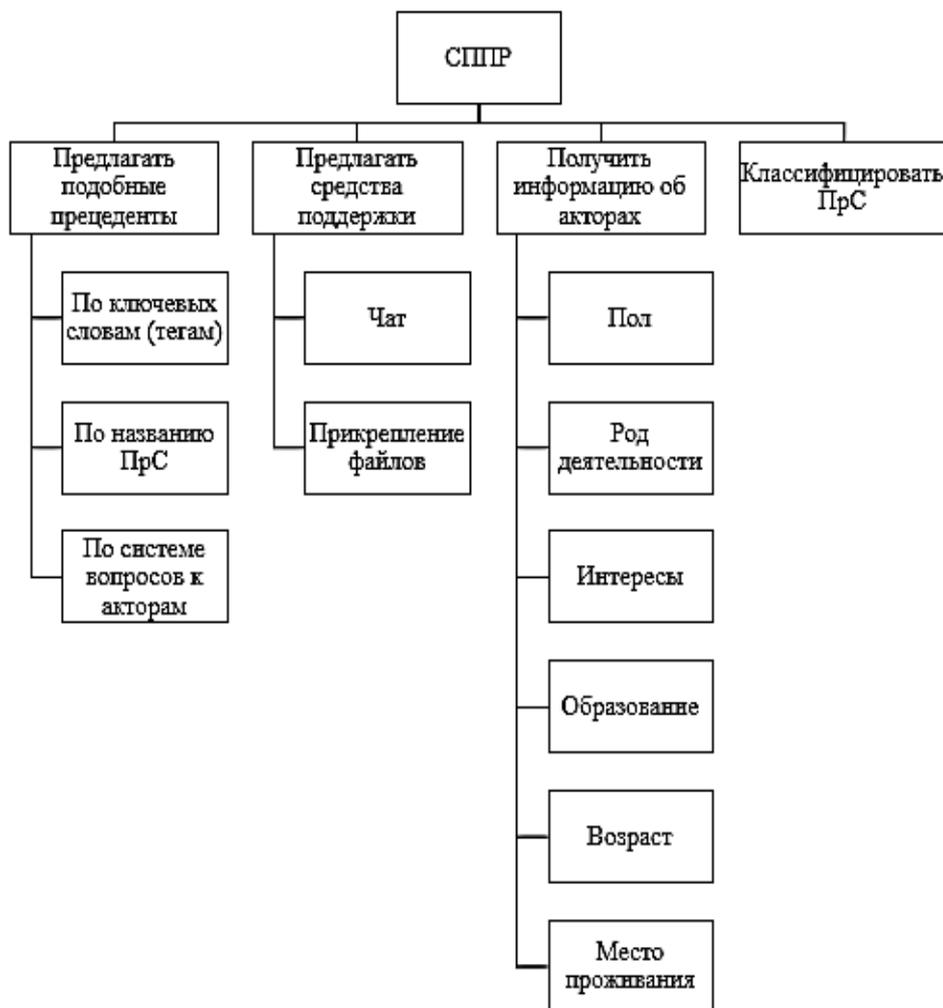


Рис. Укрупненная структура СППР

Система разрабатывается таким образом, что на этапе формирования проблемы в системе поддержки принятия решений акторам предлагается выбрать тэг проблемной ситуации из предложенных вариантов. Ситуация может иметь несколько тэгов и находиться в нескольких классификациях на сайте.

Для первичной апробации СППР предлагается использовать несколько начальных вариантов классификации, например, «Коммунальные услуги», «Проектная деятельность», «Управление персоналом» и т.д. Развивая направление «Коммунальные услуги», можно выделить наиболее популярные тэги: #проблемыжкх; #холодныебатареи; #нетгорячейводы.

Дальнейшая модификация направления будет расширяться уже непосредственно акторами, участвующими в данном обсуждении. Акторами выбираются предложенные теги и формируется ветка обсуждения, а также дополнительные, уже заранее предложенные разработчиками варианты решения ситуационной проблемы. Исключается вариант игнорирования акторами блока с выбором класса, к которому относится конкретная ситуация, так как они сами заинтересованы в разрешении данной проблемной ситуации, а заполнение информации в данном блоке поможет ее разрешению.

Имеется вариант указания акторами самостоятельных тэгов в варианте ответа «Другое». Все предложенные акторами дополнительные тэги рассматриваются модератором, который впоследствии может добавить элементы классификации на сайт. Для этого модератор рассматривает проблемную ситуацию, одобряет тэги, предложенные пользователями, создает новую ветвь классификации или дополняет уже имеющуюся. Дополнительно модератором идет отслеживание написания акторами ключевых слов с допуском орфографических ошибок, синонимов и анафразы.

В дальнейшем акторам предлагается прецедент, хранящийся в библиотеке прецедентов, имеющий совпадения с текущей ПрС по максимальному количеству параметров. Это поможет в принятии решения по ее урегулированию.

Литература

1. Vittikh V.A. Introduction to the Theory of Intersubjective Management // Group Decision and Negotiation. 2015. Vol. 24. No. 1. Pp. 67–95. <https://doi.org/10.1007/s10726-014-9380-z>
2. Vittikh V.A. Evergetics: science of intersubjective management processes in everyday life // International Journal Management Concepts and Philosophy. 2016. Vol. 9. No. 2. Pp. 63-72.
3. Vittikh V.A. Heterogeneous Actor and Everyday Life as Key Concepts of Evergetics // Group Decision and Negotiation. 2015. Vol. 24. Issue 6. Pp. 949-956. <https://doi.org/10.1007/s10726-014-9423-5>
4. Виттих В.А. Введение в теорию интерсубъективного управления. Самара: Самарский научный центр РАН, 2013. 64 с.
5. Виттих В.А. Онтологический анализ и синтез при управлении сложными открытыми системами // Проблемы управления и моделирования в сложных системах: Труды V Международной конференции. Самара, 2003. С.50-60.
6. Виттих В.А. Принцип дополнительности в науке о процессах управления в обществе // Вестник СамГТУ. Серия «Технические науки». 2016. № 4(52). С. 15-21.
7. Виттих В.А., Игнатъев М.В., Смирнов С.В. Онтологии в интерсубъективных теориях // Мехатроника, автоматизация, управление. 2012. № 5. С. 69-70.
8. Виттих В.А., Моисеева Т.В. Интерсубъективное управление: от теории к практике // Проблемы управления и моделирования в сложных системах: труды XVIII Международной конференции (г. Самара, 20-25 сентября 2016 года). Самара, 2016. С. 53-62.

9. Загорулько Г.Б. Модель комплексной поддержки разработки интеллектуальных СППР // Онтология проектирования. 2019. Т. 9. № 4(34). С. 462-479. <http://doi.org/10.18287/2223-9537-2019-9-4-462-479>

10. Загорулько Г.Б., Загорулько Ю.А. Онтологический подход к разработке систем поддержки принятия решений на нефтегазодобывающем предприятии // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2012. Т. 10. № 1. С. 121-128.

11. Загорулько Г.Б., Загорулько Ю.А. Проект открытого расширяемого инструментария для построения интеллектуальных СППР // Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011): Мат-лы Международной научно-технической конференции (г. Минск, 10-12 февраля 2011 года). Минск, 2011. С. 233-236.

12. Загорулько Г.Б., Загорулько Ю.А. Современные средства формализации семантики областей знаний на основе онтологий // Информационные и математические технологии в науке и управлении. 2018. № 3(11). С. 27-36. <http://doi.org/10.25729/2413-0133-2018-3-03>

© Исаева Е.М., Моисеева Т.В., Поляева Н.Ю., Точильникова К.А., 2022

Карпова Т.С., Малышева С.Ю., Мун Д.Ч.
Петербургский государственный университет
путей сообщения Императора Александра I
г. Санкт-Петербург, Россия

АДАПТИВНАЯ ЧЕЛОВЕКО-МАШИННАЯ СИСТЕМА ФОРМИРОВАНИЯ ТЕХНИЧЕСКИХ ЗАДАНИЙ НА ДОРАБОТКУ И ИЗМЕНЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Процесс передачи и сохранения накопленных знаний связаны с задачами искусственного интеллекта. Экспертные системы, являющиеся одним из видов интеллектуальных систем, эффективно применяются в различных областях деятельности, например, медицина, логистика, производство. Цель экспертных систем – получение результата, не уступающего по качеству и эффективности решениям, полученным экспертами. Под экспертом понимается высококвалифицированный специалист, обладающий опытом и большими познаниями, использующий различные методы в своей профессиональной деятельности, в том числе для формирования заключения, постановки диагноза для дальнейшего принятия необходимых действий.

Область знаний, связанная с анализом требований на изменение существующего программного обеспечения как раз относится к такой области, где в большей степени требуются знания экспертов. Новичку в этой области может не хватать знаний, в таком случае для накопления опыта нужны годы работы с опытным коллегой, которого может не оказаться рядом, или у коллеги ограничено время на пояснение своих действий, причин сделанного выбора и, как следствие, помощником для нового сотрудника может быть система, которая поможет выделить требования из поступившей от заказчика заявки на расширение функционала используемого в его организации программного обеспечения или исправление выявленной ошибки. Также данная система может быть актуальна для предприятий, которые желают сохранить опыт текущих сотрудников на будущее, зафиксировав их в базе знаний. Экспертные системы, как любые системы, основанные на знаниях, позволяют сохранять и пополнять базу знаний, являющуюся основой данной системы, и делать ее доступной для новых пользователей. Однако слабой стороной экспертных систем является механизм первоначального наполнения базы знаний правилами. Чаще всего это происходит вручную, специалистами, которые берут интервью у экспертов, следят и пошагово фиксируют их работу, полученные протоколы пытаются перевести в формализованные правила, которые в дальнейшем могут быть использованы при решении новых задач. Задачи, характерные для экспертных систем, являются слабо формализованными, не имеющими однозначного решения, работающие со слабо структурированными данными. Данные задачи обладают следующими отличительными чертами:

1. Не могут быть сформулированы в числовой форме.

2. Цели в данных задачах не могут быть выражены в терминах точно определённой целевой функции.

3. Не существует однозначного алгоритмического решения задачи.

Одной из особенностей слабо структурированных задач является большая размерность пространства решения, т.е. в таком случае перебор при поиске решения обширен. Другими особенностями таких задач являются периодически изменяющиеся данные и ошибочность / неоднозначность / неполнота исходных данных, как раз присутствующие в работе аналитика, особенно при ведении масштабных и продолжительных по времени проектов.

В рассматриваемой предметной области – сбору и анализу бизнес-требований в сфере разработки программного обеспечения присутствуют все перечисленные выше особенности:

1. Проекты могут пересекаться в части требований, зависеть друг от друга, являться продолжением друг друга, а также требовать изучения выполненных ранее проектов, в том числе иными сотрудниками. В таком случае количество проектов и выделенных в рамках них требований достаточно большое для анализа, что затрудняет выполнение работы в сжатые сроки.

2. Исходные требования по проекту в процессе работы могут претерпевать изменения по сравнению с исходными, определёнными на начальном этапе работы.

3. Весьма часто бывает, что поступившие от заказчика требования неоднозначные, т.е. ожидаемый результат неясен, также требования могут быть неполными. Полный их перечень составляется после работы со всеми заинтересованными лицами, которых может быть много. По незнанию/недостаточному уровню знаний данными лицами могут быть указаны ошибочные данные по процессу/интеграции и обработке данных.

Группы задач, которые успешно могут быть решены с использованием экспертных систем: интерпретация; диагностика; проектирование; мониторинг; планирование; прогнозирование.

Экспертные системы для своего построения могут использовать специальные языки, например, Prolog. Данный язык позволяет сформулировать продукционные правила и заносить их в базу знаний. Механизм вывода зашит в транслятор данного языка. В предлагаемом решении исходная база знаний строится с использованием методов аффинитивного анализа путем анализа текстов заявок на естественном языке [1, с. 182]. Первоначальная база знаний может быть так же автоматически расширяема, если применить метод обратной связи и включать в исходный список вновь поступающие заявки. Алгоритм APRIORI позволяет сформировать базу знаний ключевых выражений автоматически. Использование же полученной базы знаний уже на первом шаге обработки новой заявки обеспечивает расширение множества ключевых фраз до максимально возможного, в соответствии с текущим состоянием базы знаний. Предлагаемая система реализуется в основном на отечественной платформе 1С: Предприятие, где спроектирован ряд объектов для хранения заявок и правил. На рис. 1 представлены объект с типом «Документ» и объекты с типом «Справочник», используемые при реализации описанной системы.

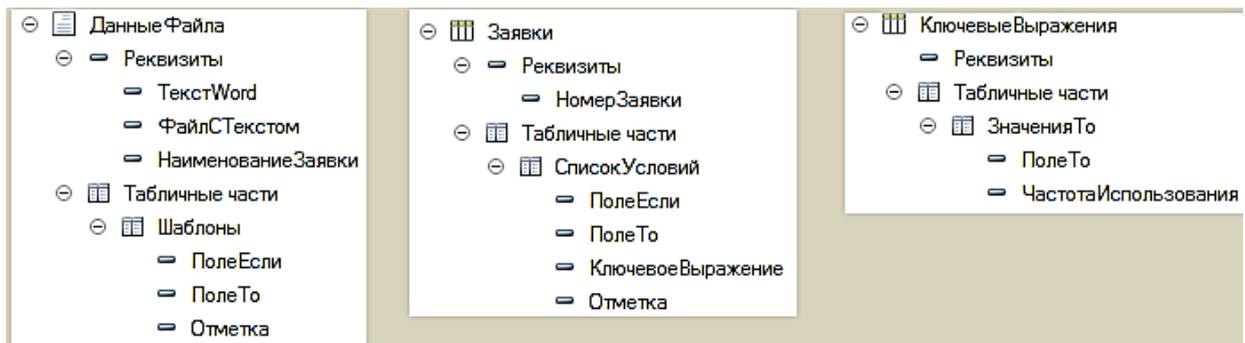


Рис. 1. Структура объектов

В справочнике, названном «Ключевые выражения», хранятся выделенные при обработке заявок фразы и перечень действий, связанный с каждой из фраз. Для одного ключевого выражения может быть несколько действий. Например, для ключевого выражения «Создать отчёт» перечнем возможных действий, входящих в техническое задание, может быть следующее: «Создать отчёт», «Предоставить доступ», «Добавить столбцы», «Указывать в отчёте следующие данные ...».

В документе «Данные файла» хранятся данные по заявке, а именно: наименование заявки; текст заявки из выбранного пользователем файла; выделенные ключевые выражения, занесённые ранее в базу; перечень соответствующих им действий; отметки у подходящих правил. Процесс работы создаваемой системы, изображён на рисунке 2.

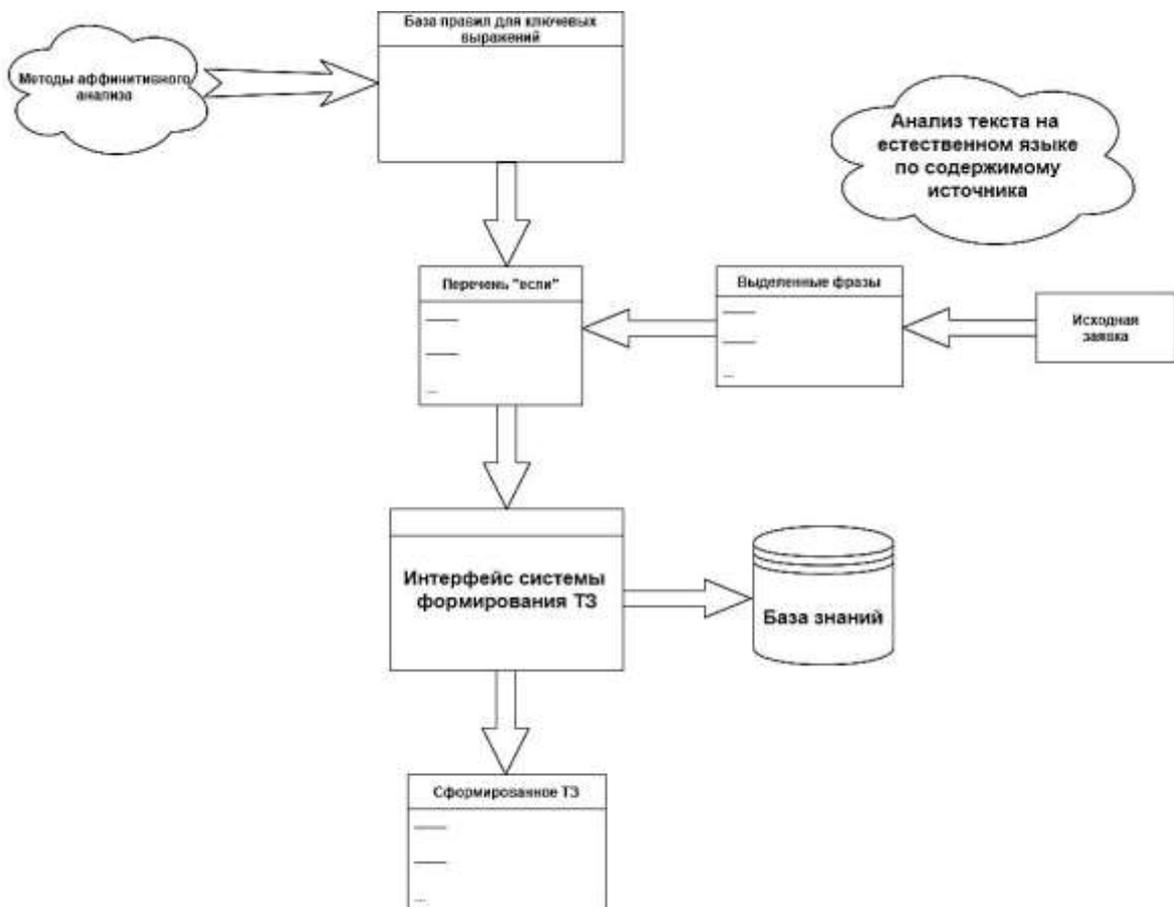


Рис. 2. Архитектура системы

На вход поступает текстовый файл с описанием требований заказчика – исходная заявка. Сначала выполняется поиск наличия в данном тексте уже имеющихся в системе ключевых выражений. Банк ключевых выражений заполняется предварительно экспертами, хранится в специализированном справочнике платформы 1С: Предприятие.

В зависимости от реализации предусмотрено 2 механизма:

-обработка и расширение исходного перечня ключевых фраз с использованием встроенных средств платформы 1С: Предприятие;

-справочник выгружается во внешний файл, и обработка происходит внешними модулями, которые могут быть реализованы на любом языке программирования.

Полученный расширенный список ключевых выражений составляет базовую часть «ЕСЛИ» продукционных правил экспертной системы формирования технического задания. После того, как сформирован расширенный набор требований, для каждого из них подбирается техническая формулировка действия, которое необходимо выполнить для достижения результата. Фактически это части «ТО» продукционных правил.

В случае, если нескольким требованиям соответствует одно и тоже действие, то в техническое задание оно заносится один раз. Аналогично для каждого требования. В экспертных системах участие человека предусмотрено только на этапе получения знаний от эксперта для формирования правил и ввода исходных данных по задаче. В разрабатываемой системе предусмотрено участие пользователя для выбора наиболее подходящего перечня действий из предложенного системой, поэтому данная система является автоматизированной. Также в системе присутствует возможность добавления при обработке заявки связи «ЕСЛИ ..., ТО ...» вручную. Формирование перечня действий при наличии подходящих записей, как и в экспертных системах, является автоматическим. После окончания работ по анализу заявки в системе формируется новая запись в объекте с типом «Документ» по обработанной заявке. Техническое задание выгружается в файл формата doc/docx для его дальнейшего заполнения и обработки, дополнения и редактирования сотрудником. Адаптационный механизм предполагает ранжирование правил по частоте их использования. Пример расчёта значения частоты представлен на рис. 3. Ранги правил хранятся в базе знаний и при выводе их в интерфейсе пользователя наверху списка находятся правила, имеющие максимальную частоту использования.

При учете выбранных результирующих правил в дальнейшем может происходить экспертиза и обучения новых сотрудников путем сравнения выбранных ими результирующих действий с теми, что, допустим, были ранее зафиксированы опытными специалистами. Исходные данные, вводимые в систему задаются на естественном языке. Интерактивный механизм выбора аналитиком окончательного решения имеет простой понятный интерфейс и может быть использован практически в любой сфере деятельности, где есть потребность в обучении новых сотрудников и передаче им хранимых в системе знаний путем, например, дополнительного тренинга.

← → ☆ **Данные файла 000000005 от 01.03.2022**

Провести и закрыть | Записать | Провести

Номер: 000000005 | Дата: 01.03.2022 23:36:35

Файл с текстом: C:\Users\... | Получить текст из MS Word

Текст из файла MS Word:
Доработать отчёт в Excel «...», добавив столбец с ФИО сотрудника, заключившего договор. Новый столбец расположить после столбца А.

Наименование заявки: Проверка | Вывод в файл MS Word

Добавить | ↑ ↓ | Анализ текста | [Icons]

N	Поле "Если"	Поле "То"	Отметка
1	Доработать отчёт	Добавить столбец	<input type="checkbox"/>
2	<u>Добавив столбец</u>	<u>Расположить после ...</u>	<input checked="" type="checkbox"/>

← → ☆ **Ключевые выражения**

Создать | [Icon]

Наименование	Код
Внести изменения в форму документа	000000020
выбрать в списке	000000017
<u>Добавив столбец</u>	000000005

☆ **Добавив столбец (Ключевые в...)**

Записать и закрыть | Записать

Код: 000000005

Если: Добавив столбец

Добавить | ↑ ↓ | Поиск (Ctrl+F) | Ещё ▾

N	Поле то	Частота
1	<u>Расположить после ...</u>	1
2	В столбец «...» выводить значение р...	

+1

Рис. 3. Структура объектов

Так как в основе продукционной модели представления знаний находится продукция, т.е. правило вида «IF <условие>, THEN <действие>», то отчасти описанный выше подход в разрабатываемой системе схож с продукционной моделью представления знаний в экспертных системах. Продукционная модель представления знаний широко распространена в автоматизированных системах управления технологическими процессами.

Литература

1. Карпова Т.С., Малышева С.Ю. Разработка архитектуры системы автоматического формирования требований на разработку программного обеспечения // Наука настоящего и будущего: Материалы VIII научно-практической конференции. Т. I. СПб., 2020. С. 182-185.

© Карпова Т.С., Малышева С.Ю., Мун Д.Ч., 2022

УДК 004.5

Коноваленко О.В., Сиволобов С.В.
Волгоградский государственный университет
г. Волгоград, Россия

УСТРОЙСТВО ДЛЯ ГОЛОСОВОГО УПРАВЛЕНИЯ БЫТОВОЙ ТЕХНИКОЙ

Ввиду появления большого количества новой техники важным критерием является эффективное взаимодействие ЭВМ с человеком. Методы, использующие голосовое управление и графическое отображение информации, являются наиболее продуктивными при реализации человеко-машинного интерфейса. Именно поэтому одним из самых распространённых видов пользовательского интерфейса является голосовой [9]. Пользователи могут подавать машине различные голосовые команды и получать в итоге нужную информацию за достаточно короткий период, например, узнать текущее время, поставить будильник, совершить телефонный звонок, найти ответ на интересующий вопрос в Интернете [14]. А для некоторых людей, с ограниченными возможностями, распознавание речи открывает много возможностей (например, упрощает ввод текста).

Умный дом является системой, достаточно эффективной и комфортной для пользователей [10]. Её контроллер позволяет управлять освещением, включать и выключать подключённые к розеткам электрические приборы, выполнять функции охраны дома (сигнализация, видеонаблюдение, имитация наличия хозяина, открытие шлагбаума и дверей) [7]. Однако он не может осуществлять управление функциями бытовых приборов: включать и выключать технику, переключать каналы на телевизоре, регулировать громкость, что упрощало бы повседневную жизнь человека [11].

В последнее время активно разрабатываются и широко применяются устройства, в основе которых находятся голосовые команды [2]. В настоящее время реализуется большое количество технологий распознавания речи, публикуется множество научных исследований [1; 4] в этой области. Большинство их реализаций сосредоточено на создании более совершенной системы распознавания речи с высокой точностью.

Управление некоторыми интеллектуальными устройствами (например, Smart TV) осуществляется через Интернет или беспроводные каналы связи, например, через Wi-Fi- или Bluetooth-пульт дистанционного управления или смартфон (как правило, через мобильное приложение). Несмотря на это, наиболее распространённым устройством для дистанционного управления является ИК-пульт, передающий команды с помощью инфракрасного излучения. Для расширения возможностей контроллера умного дома, в частности, для управления функциями бытовых устройств, можно реализовать в нём механизм формирования команд в инфракрасном оптическом диапазоне, а также добавить в него возможность обработки голосовых команд пользователя [3].

В данной работе предлагается автономное устройство (для работы которого не требуется постоянно включенный персональный компьютер), которое будет принимать и распознавать голосовые команды и в соответствии с ними управлять бытовыми

устройствами, а также как это делается с помощью прилагаемых к ним пультов дистанционного управления.

Для реализации устройства был выбран миникомпьютер Raspberry Pi 3 Model B+, построенный на четырёхядерном процессоре ARM Cortex-A53 [8]. Его выбор обусловлен тем, что для реализации данного устройства может не хватить вычислительной мощности 8-битных микроконтроллеров серии ATmega, используемых в платах Arduino. Также его преимуществом является наличие встроенного Wi-Fi модуля и, что значительно более важно, программы для него разрабатываются, в первую очередь, на языке программирования Python, для которого существует большое количество библиотек для выполнения различных задач, в том числе в области распознавания речи.

Для реализации распознавания речи было принято решение использовать сервис Google Speech Recognition [13]. Данный сервис выбран, в первую очередь, в связи с тем, что он не требует покупки лицензионного ключа или пополнения баланса для работы с ним и при этом обеспечивает приемлемое качество распознавания, а также не требует авторизации. Использовать данный сервис, так же как и другие готовые решения для распознавания речи, на языке Python удобнее всего с помощью библиотеки SpeechRecognition. Одним из преимуществ использования данной библиотеки является уже реализованная в ней возможность фоновое распознавание речи, которая позволяет с помощью функции обратного вызова обрабатывать голосовые команды сразу после их получения.

На рисунке 1 показана диаграмма развёртывания реализуемой системы. Для подключения микрофона к Raspberry Pi применена звуковая карта, подключаемая по интерфейсу USB.

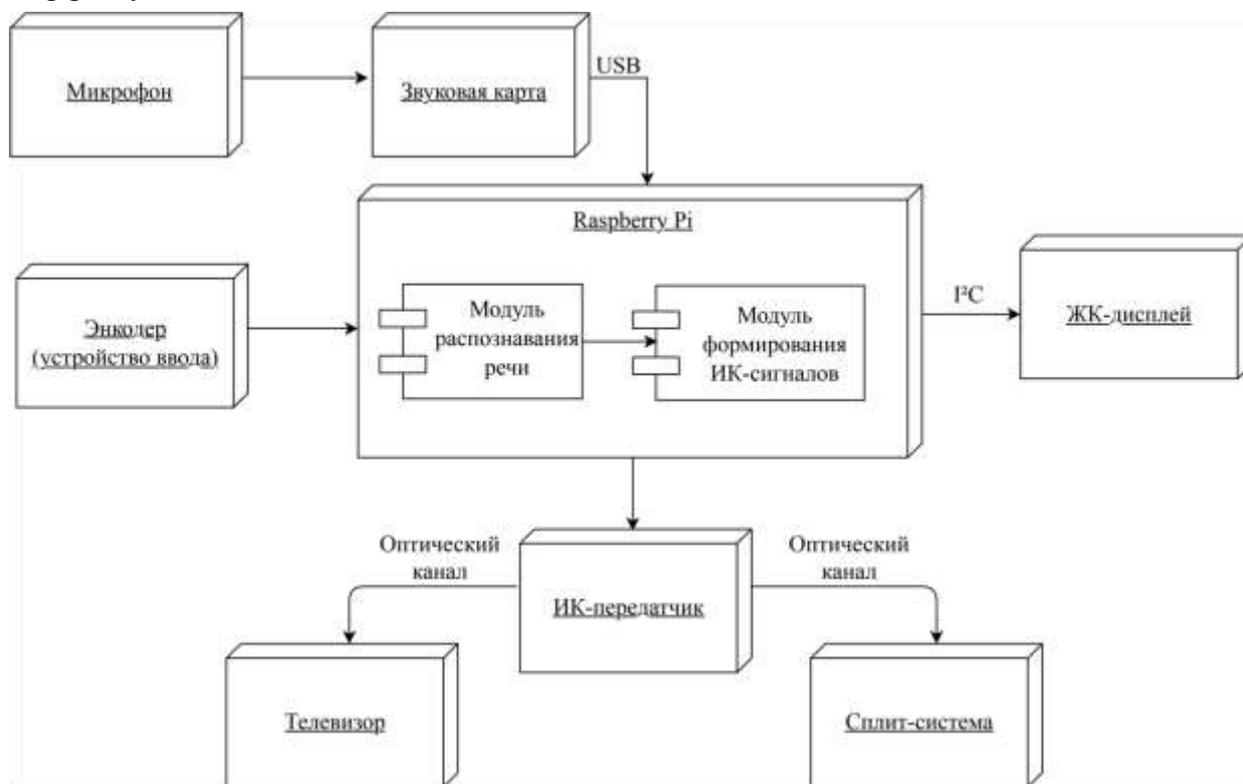


Рис. 1. Диаграмма развёртывания системы голосового управления бытовой техникой

Настройка устройства пользователем осуществляется с помощью энкодера, при этом сообщения пользователю отображаются на ЖК-дисплее.

Передача управляющих сигналов бытовой технике осуществляется точно так же, как это реализуется с помощью штатного пульта дистанционного управления. По аналогии с ними, для этого используется инфракрасный светодиод. Основным отличием от обыкновенного пульта дистанционного управления является мощность излучения. Для того, чтобы одно управляющее устройство, размещённое в комнате, могло управлять сразу несколькими бытовыми устройствами (например, телевизором, сплит-системой) был применён мощный инфракрасный светодиод, подключенный через усилительный каскад, выполненный на одном транзисторе. На рисунке 2 показана схема готового устройства [6]. Микрокомпьютер Raspberry Pi 3 Model B+ имеет 3,5 мм разъём, который позволяет подключить наушники, а также может использоваться для вывода композитного видео и аудио (например, для подключения к телевизору). Однако данный разъём не предназначен для подключения микрофона. Для подключения микрофона использована малогабаритная звуковая карта, подключаемая по интерфейсу USB.

В семействе операционных систем Linux для реализации приёма и передачи команд в инфракрасном диапазоне чаще всего применяется программный пакет LIRC (Linux Infrared Remote Control). Данный пакет позволяет создать полноценный пульт дистанционного управления с нуля, а также копию имеющегося пульта. С помощью имеющейся в пакете утилиты система была обучена командам для нескольких пультов дистанционного управления. Для этого к плате Raspberry Pi был подключен инфракрасный приёмник VS1838B. Для того, чтобы реализованное устройство могло передавать команды аналогично имеющемуся пульту дистанционного управления, необходимо запустить утилиту `irrecord` и следовать выводимым ею на экран инструкциям. В процессе записи необходимо нажимать на пульте дистанционного управления различные клавиши и указывать какие имена будут использоваться для отправки соответствующих им команд. Чтобы все команды были записаны максимально точно, необходимо, чтобы излучатель пульта дистанционного управления находился на как можно меньшем расстоянии от инфракрасного фотоприёмника.

Для анализа распознанного текста с целью извлечения команд применены регулярные выражения. При этом команды, направляемые системе, должны иметь строго определённый вид, например: «Уменьши громкость» или «Убавь громкость». Для того, чтобы система могла понимать произносимые пользователем команды независимо от того, как именно они были сказаны, можно применить средства для семантического анализа текста [5]. Для анализа текста часто применяется рекуррентные нейронные сети, которые могут анализировать контекст, в котором были сказаны слова. Также для этого существуют готовые решения [12]. Поскольку применённый сервис для распознавания речи можно использовать бесплатно, для анализа текста тоже стоит применять решения, не требующие оплаты. Одним из таких является платформа Rasa, которая также может работать без подключения к Интернету. Rasa – это платформа машинного обучения с открытым исходным кодом для автоматизации обработки текста на естественном языке. Rasa позволяет преобразовывать необработанные

текстовые сообщения в структурированные данные. Как и многие средства для машинного обучения платформа Rasa предназначена для использования в проектах на языке Python.

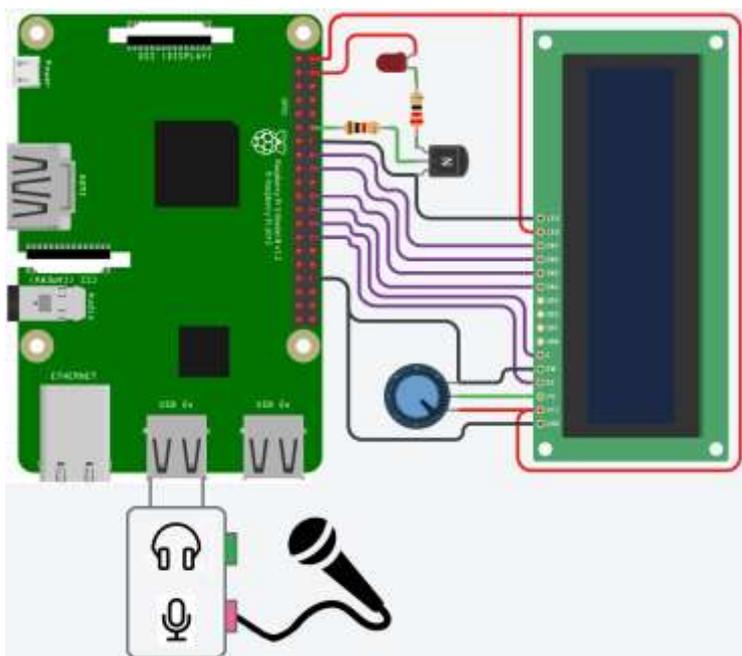


Рис. 2. Схема готового устройства

Качество распознавания речи, в значительной степени, определяется качеством захвата аудиосигнала, и в первую очередь, это определяется качеством и функциональностью микрофона. Для успешного распознавания речи при значительном удалении устройства от говорящего могут применяться микрофоны с высокой чувствительностью, автоматической регулировкой усиления, а также различные системы шумоподавления. При использовании разработанной системы с микрофоном без предварительного усиления точность распознавания команд в условиях шума составила 81%, а в условиях тишины – 92%.

Литература

1. Беленко М.В. Сравнительный анализ систем распознавания речи с открытым кодом // Международный научно-исследовательский журнал. 2017. № 4-4(58). С. 13-18. <https://doi.org/10.23670/IRJ.2017.58.141>
2. Каменская А.С. Адаптация Google Cloud Speech-to-text API для автоматической транскрипции веб-конференций в реальном времени // Автоматика и программная инженерия. 2019. № 2(28). С. 19–23.
3. Кашкаров А.П. Электронные схемы для «Умного дома». М.: НТ Пресс, 2007. 256 с.
4. Кипяткова И.С, Карпов А.А., Ронжин А.Л. Автоматическая обработка разговорной русской речи. СПб.: ГУАП, 2013. 314 с.
5. Макаревич Т.И. Интеллектуальный анализ текстовой информации в специализированных областях в системе электронного правительства // Цифровая трансформация. 2019. № 2(7). С.46-52.
6. Петин В.А. Микрокомпьютеры Raspberry Pi. Практическое руководство. СПб.: БХВ-Петербург, 2015. 240 с.

7. Петин В.А. Создание умного дома на базе Arduino. М.: ДМК Пресс, 2018. 180 с.
8. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. СПб.: БХВ-Петербург, 2016. 320 с.
9. Портянкин Иван. Swing. Эффективные пользовательские интерфейсы. М.: ЛОРИ, 2011. 608 с.
10. Стариков А.Н, Рощина С.И, Власов А.В. Проектирование и эксплуатация конструкций и инженерных систем современных энергоэффективных зданий в развитии концепции Умный дом. Владимир: Изд-во ВлГУ, 2014. 123 с.
11. Тесля Е.В. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире. СПб.: Питер, 2008. 360 с.
12. Aggarwal С.С., Zhai С. Mining Text Data. Springer, 2012. 527 p.
13. Anggraini N., Kuniawan A., Wardhani L. Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API // TELKOMNIKA. 2018. Vol. 16. No. 6. P. 2733–2739.
14. Oparin I., Talanov A. Stem-Based Approach to Pronunciation Vocabulary Construction and Language Modeling for use in the interactive information systems // Speech and Computer. 2005. No. 6. P. 519–522.

© Коноваленко О.В., Сиволобов С.В., 2022

УДК 621.391.812.7

Коновской А.С., Нестеров С.В.

Филиал «Взлёт» Московского авиационного института
г. Ахтубинск, Россия

МОДЕЛИРОВАНИЕ ВЫСОКОГОРНОЙ ОПТИКО-ЭЛЕКТРОННОЙ ЛИНИИ СВЯЗИ

В процессе обучения студентов радиотехнического направления подготовки, одной из дисциплин, на которых базируется понимание функционирования оптических радиосистем, является «оптико-электронные системы передачи информации». Крайне важно раскрыть перед обучающимися её основные положения в полной мере. В противном случае на выходе их стен ВУЗа специалист не обладает компетенциями, необходимых для работы с оптическими радиосистемами [1, с. 68]. Использование при обучении только натуральных установок не позволяет охватить всё многообразие оптических систем и условий их функционирования. Освоение дисциплины «оптико-электронные системы передачи информации» возможно с использованием компьютерных моделей оптических линий связи в различных условиях с визуальным представлением данных.

В настоящей работе описывается компьютерная высокогорная оптико-электронная линия связи. Данная модель разработана в среде MATLAB с задействованием интерактивной средой программирования для разработки и размещения приложения и программирования его поведения App Designer, а также с задействованием средств графического пользовательского интерфейса MATLAB GUI (<https://clck.ru/epWSE>; <https://clck.ru/epWTB>).

Модель высокогорной оптико-электронной линии связи представляет собой два файла: файл программный код (рис. 1) и файл – визуализация ввода и вывода данных (рис. 2). При работе с моделью устанавливаются предварительные данные, такие как: «Высота передатчика(м)», «Высота приёмника (м)», «Длина линии связи (км)», «Мощность излучения (Вт)» при помощи соответствующих ползунков, также эти данные можно ввести вручную, но только в заданном диапазоне ползунков. Эти входные параметры учитываются для расчёта количества частиц, которые и осуществляют рассеяние лучистой энергии. Рассеяние лучистой энергии частицами подчиняется закону Рэлея, так как передача осуществляется в сухой и чистой газовой среде.

Также пользователю представляется возможность задания таких входных данных, как диаметры линз приёмника и передатчика, а также диаметр излучательной системы, что, в свою очередь, позволяет провести исследование конструктивных особенностей приёмо-передающих частей промоделированной оптической линии связи. Данные параметры не обладают предельным значением вводимых данных. Результаты моделирования представлены на рис. 3 [2, с. 882].

Затем, происходит выбор вида лазера, с которым будет производиться работа. На выбор пользователю даются два лазера: лазер на стекле с неодимом и лазер на CO_2 . Что, в свою очередь, позволяет провести исследование применения различных лазеров для осуществления передачи данных с помощью оптической линии связи. Результаты

моделирования представлены на рис. 4, 5 [3, с. 106-120, с. 146-190; 4, с. 20-23, с. 33; 5, с. 49-99].

При нажатии кнопки «Расчёт» начинается моделирование работы оптической линии связи и расчёт мощности сигнала в точке приёма в соответствии с теоретическими положениями оптической радиотехники.

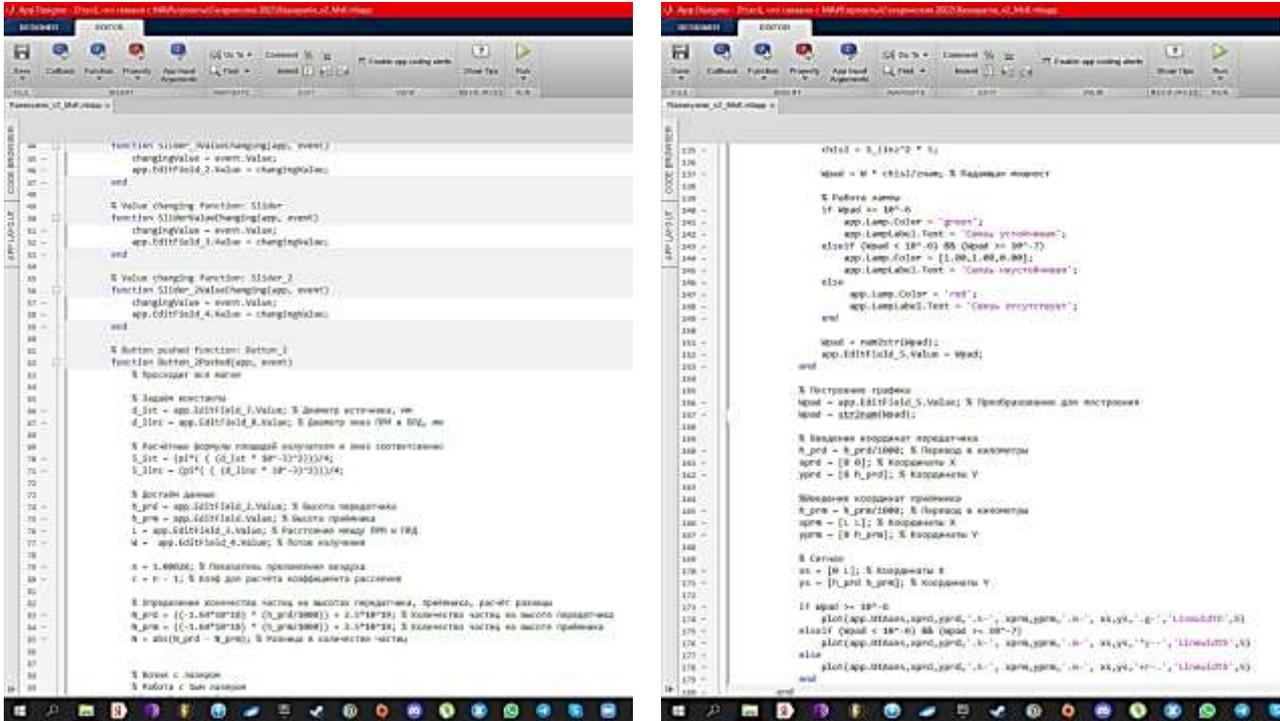


Рис. 1. Части программного кода модели высокоскоростной оптико-электронной линии связи для исследования рассеяния оптического излучения

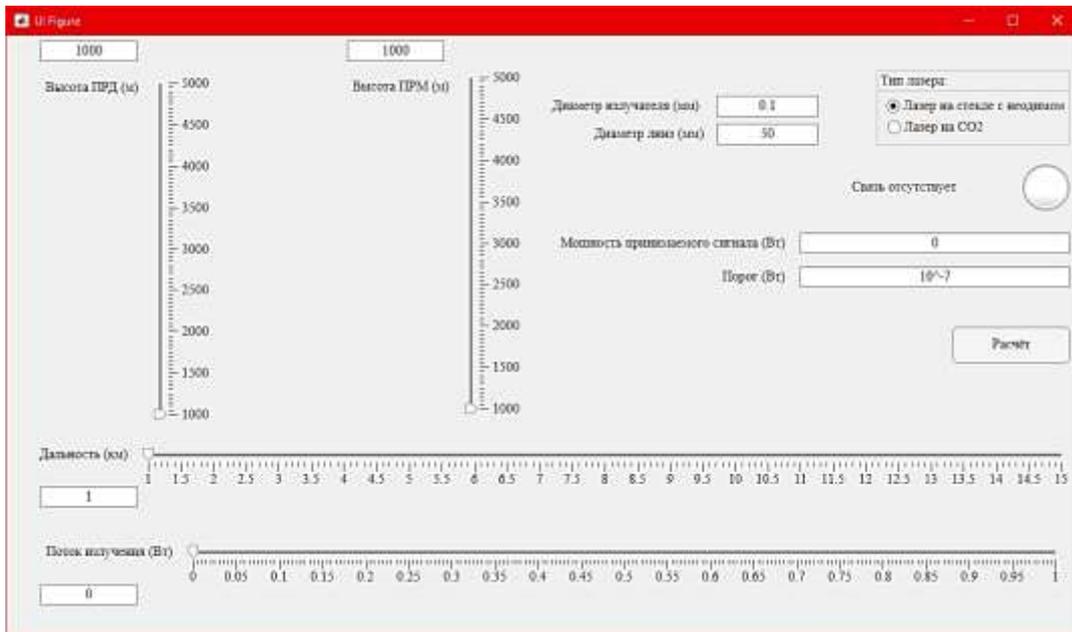
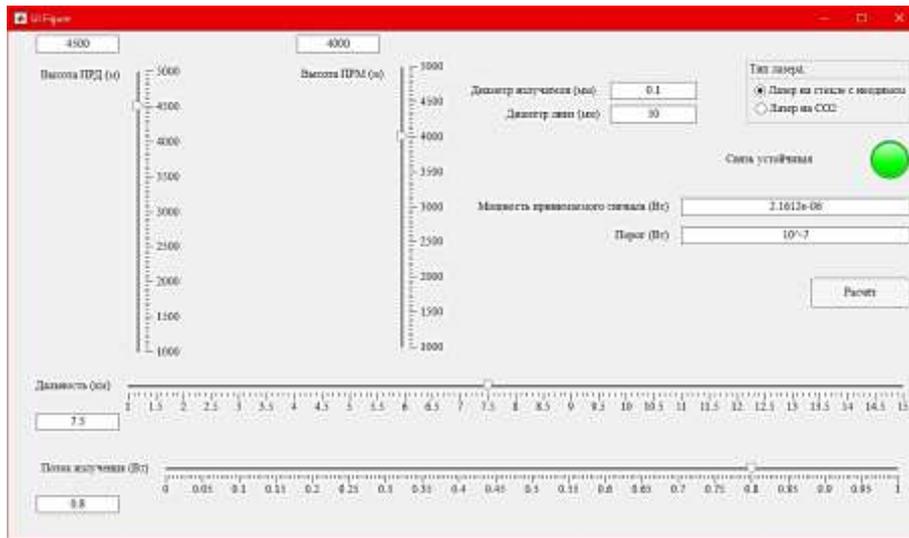
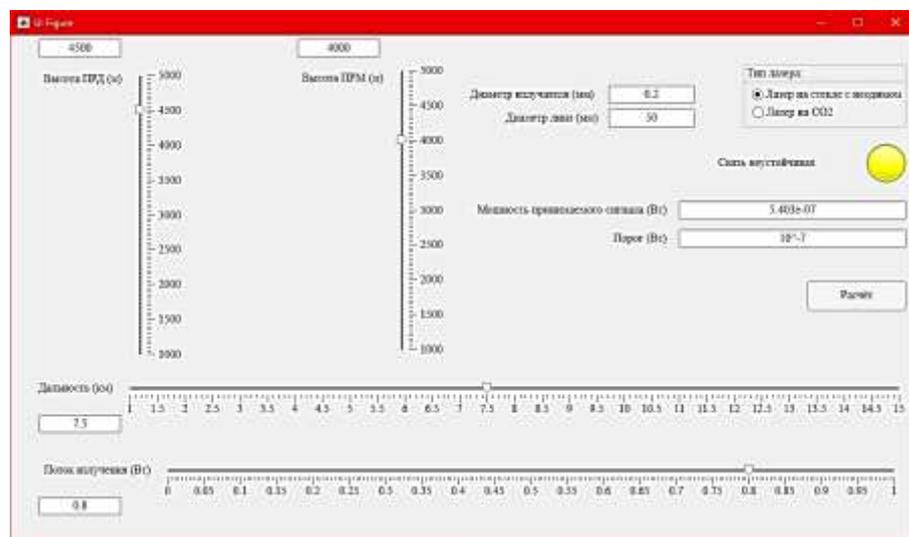


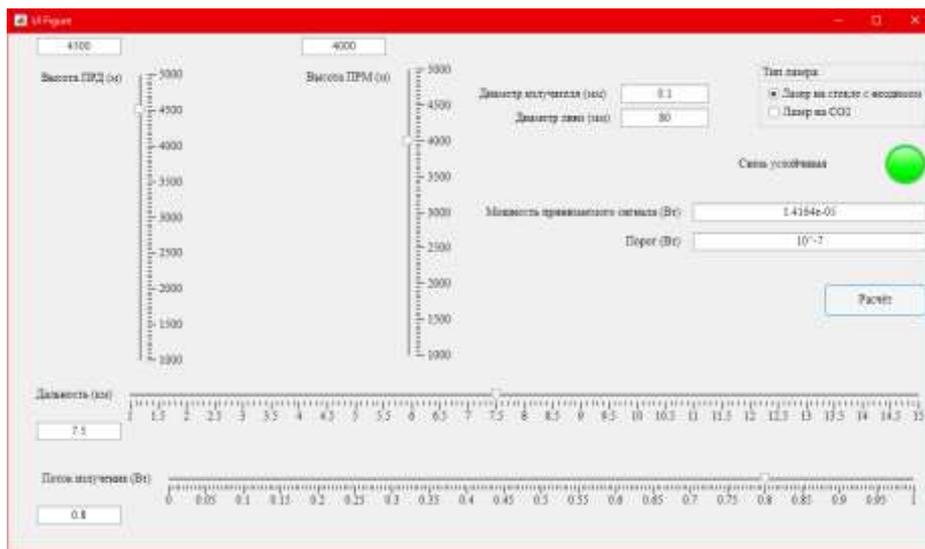
Рис. 2. Пользовательское окно компьютерной модели высокоскоростной оптико-электронной линии связи



а)



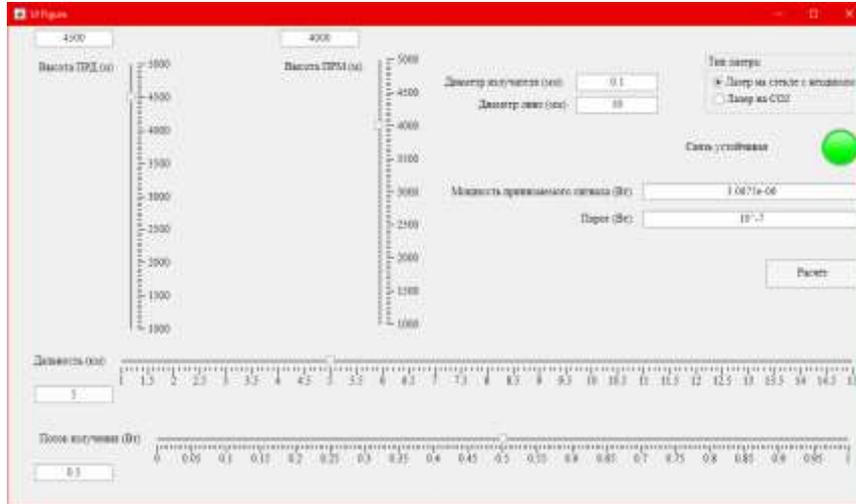
б)



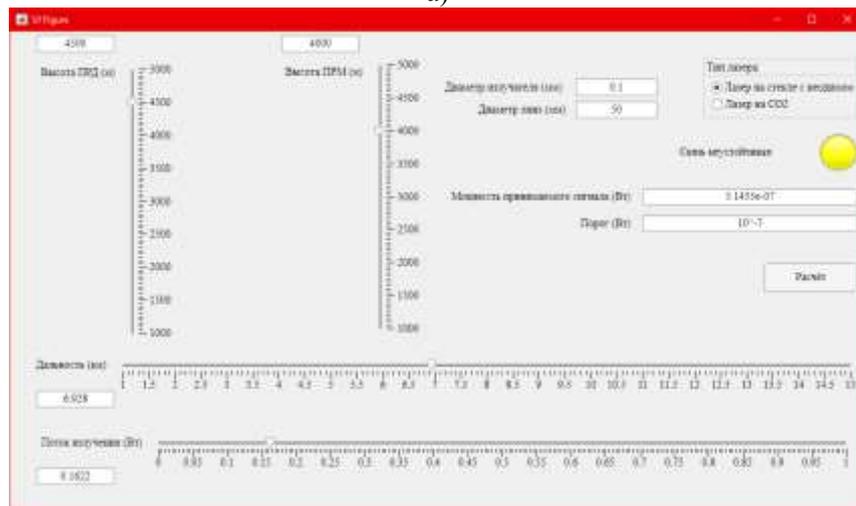
в)

Рис. 3. Окно модели высокогорной оптико-электронной линии связи при изначальных условиях (а), при большом диаметре излучателя (б), при большом диаметре линз (в)

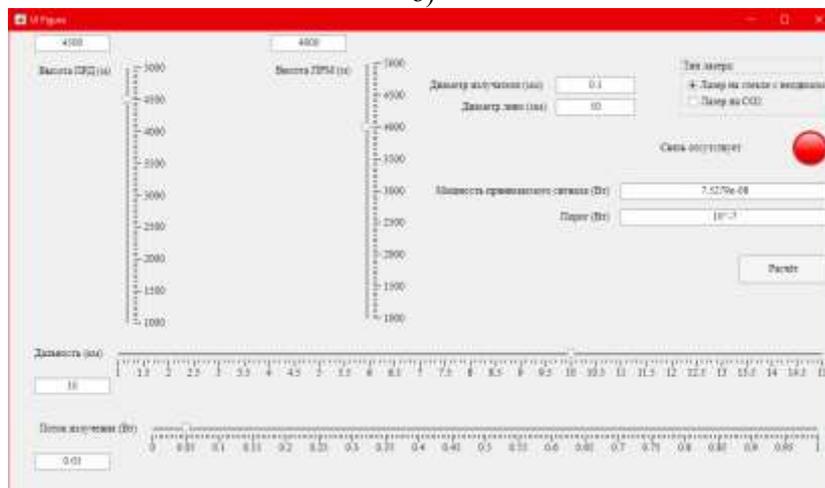
В пользовательском окне выводится мощность принимаемого сигнала, сравниваемое с пороговым значением, указанным в специальном окне. Затем, происходит индикация о состоянии связи: если лампа горит «зелёным», то связь является устойчивой; если лампа горит «жёлтым», то связь является неустойчивой; если же лампа горит «красным», то связь между приёмником и передатчиком отсутствует (рис.5).



а)

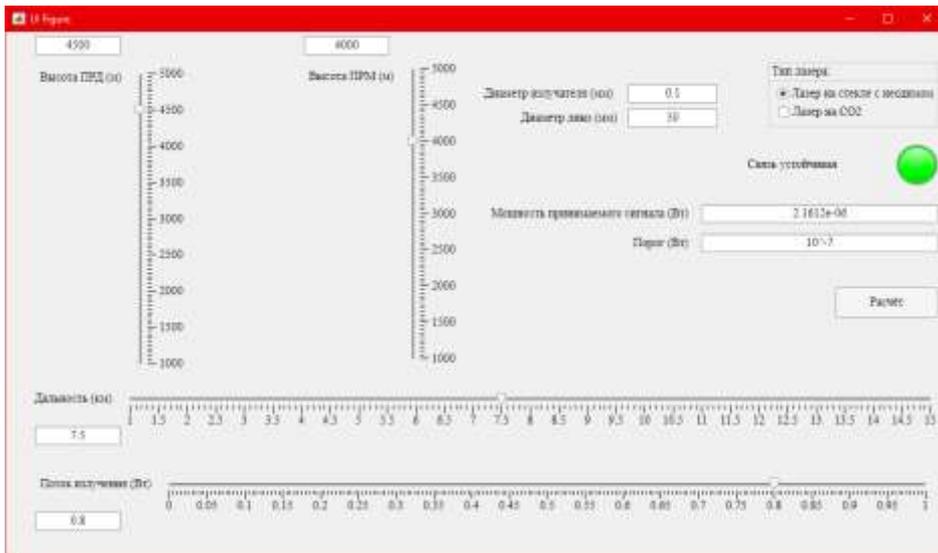


б)

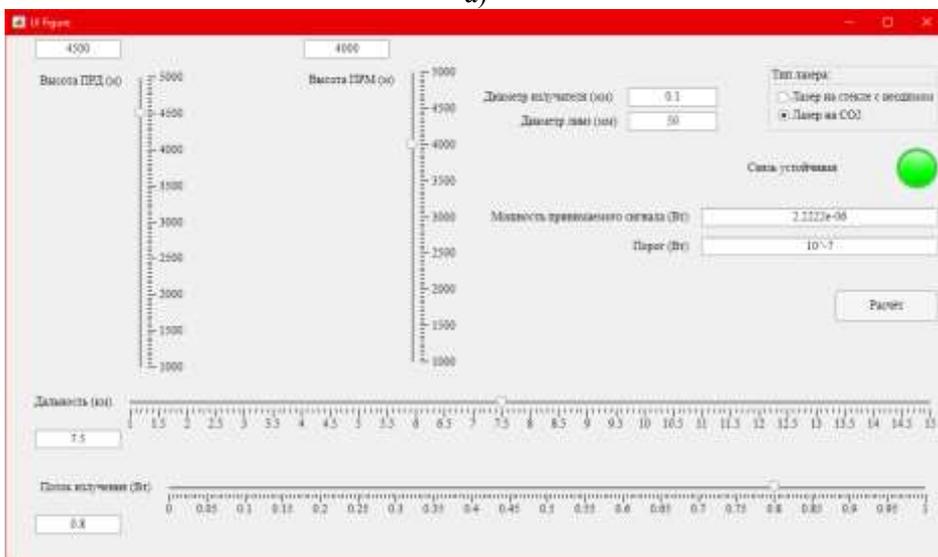


в)

Рис. 4. Пользовательское окно компьютерной модели высокогорной оптико-электронной линии связи при устойчивой связи (а), при неустойчивой связи (б), при отсутствии связи (в)



а)



б)

Рис. 5. Пользовательское окно компьютерной модели высокогорной оптико-электронной линии связи при использовании лазера на стекле с неодимом (а), при использовании лазера на CO_2 (б)

Студенты имеют возможность понаблюдать, как изменяется мощность принимаемого сигнала от задающих данных, обладающие большим диапазоном выбора. Результаты экспериментов, выполненных на модели, соответствуют положениям оптико-электронных систем и оптической физики. Помимо использования в изучении оптико-электронных систем передачи информации в горных условиях, данная модель может применяться в исследовательской части дипломных работ.

Литература

1. Кутарова Е.И., Рыжкова М.Н. Модель процесса обучения студентов радиотехнического направления подготовки дисциплинам естественнонаучного цикла // Радиотехнические и телекоммуникационные системы. 2017. № 2(26). С. 68-75.



2. Денисов А.В. Моделирование оптико-электронных систем космического назначения // Известия высших учебных заведений. Приборостроение. 2015. Т. 58. № 11. С. 882-888. <https://doi.org/10.17586/0021-3454-2015-58-11-882-889>

3. Криксунов Л.З. Справочник по основам инфракрасной техники. М.: Сов. радио, 1978. 400 с., ил.

4. Карпов Ю.Г. Имитационное моделирование систем. Введение и моделирование с AnyLogic 5. СПб.: БХВ-Петербург. 2005. 400 с. ил.

5. Максимей И.В. Имитационное моделирование на ЭВМ. М.: Радио и связь, 1988. 232 с.: ил.

© Коновской А.С., Нестеров С.В., 2022

УДК 681.3:796

Кривогузова М.В., Ерошенко И.А., Низовцева Ю.Я.,
Волгоградский государственный технический университет
г. Волгоград, Россия

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СФЕРЕ СПОРТА

Большая часть 2020 года прошла без массовых мероприятий. Из-за коронавируса всем пришлось менять планы. Эта проблема также коснулась и спортивного сектора. Матчи отменялись или переносились, личное присутствие заменила удаленная работа, цифровизация превратилась из инновации в необходимость, а креативные усилия пришлось удвоить — чтобы хоть как-то выделиться в переполненной медиа среде.

В 2021 году мы начинаем адаптироваться к присутствию вируса, в отличие от 2020-го, когда весной мировая пандемия только начиналась и никто не знал к чему она приведет и какие средства борьбы необходимы. Теперь уже для всех очевидно, что и в 2022 году COVID и его различные мутации, а также возможные новые болезни, скорее всего, будут оказывать влияние на нашу жизнь. Это приводит к выводу о том, что необходимо научиться жить и проводить мероприятия в данных условиях. Поэтому есть тенденции в индустрии спорта, которые появились во многом из-за мировой пандемии, однако несмотря на этот фактор существуют также другие изменения, связанные с появлением новых технологий в спорте.

В 2021 году проходили такие спортивные события, как Чемпионат Европы по футболу, летние Олимпийские игры в Токио и др. И каждое крупное международное соревнование является не только способом проверить свое мастерство и силы, но и возможность увидеть, как технологии меняют сферу спорта. В настоящее время происходит стремительное развитие научно-технического прогресса, формирование информационного общества.

Поскольку многие соревнования были перенесены, вообще отменены или имели какие-либо ограничения по проведению, практика показала, что по некоторым видам спорта можно проводить соревнования онлайн. К таким видам спорта отнеслись художественная гимнастика, фигурное катание и даже легкая атлетика. Соревновались спортсмены на спортивных площадках в разных странах, даже на разных континентах. Хотя, конечно, необходимо признать, что соревнования были небольшими и во многом носили тренировочный характер. В целом опыт был положительным и показал возможности дальнейшей реализации в онлайн формате.

В особенности такой формат очень неплохо показал себя в шахматах. За этот год много турниров прошло онлайн, а в сентябре 2021 года в шахматах прошла уже Вторая Всемирная онлайн-олимпиада, в которой приняли участие сборные 153 стран. Если говорить о профессиональном спорте, то можно предположить, что тенденция сохранится и будет развиваться дальше.

Цифровизация - это тема, которая затрагивает все отрасли, и спортивная индустрия тоже не является исключением. Направление на развитие спортивной индустрии закреплено на государственном уровне, существует национальный проект «Спорт — норма жизни», которая стремится достичь того, чтобы россияне к 2024 году регулярно занимались спортом

и их количество было около 55%. Кроме того, об особом внимании к данной теме со стороны государства говорит и недавнее создание департамента цифровой трансформации Министерства спорта.

Компьютер и программное обеспечение, являются ключом к успеху для современного спортивного менеджера. Это та часть оборудования, которая позволяет спортивному администратору максимизировать отдачу от ограниченных ресурсов, будь то люди, инвентарь или финансы. В свою очередь, интернет, а также другие средства передачи информации, возможно, самый главный инструмент для спортивного менеджера, чтобы увеличить охват различных спортивных программ для как можно большего числа потенциальных участников. Кроме того, использовавшиеся первый раз в 2014 году системы видеоповтора смогли сделать спорт более справедливым и объективным [4, с.338].

В настоящее время существует мощная и полезная база данных для спортивных менеджеров, которая называется реляционная база данных. Наглядным примером может служить сотрудник, которому поручено наблюдать за рейтингом спортивных соревнований, сведения о которых были занесены в реляционную базу данных. Аналогичное программное обеспечение для управления событиями может помочь спортивному менеджеру с множеством других задач, связанных с соревнованиями, например, планирования мероприятия, установки оборудования и многого другого [5, с. 89].

Из вышесказанного можно легко увидеть ценность использования ИТ-инструментов для организации соревнований. Эти инструменты еще более важны для повседневной работы спортивной организации, о чем свидетельствуют виды информации о спортивных программах, которые могут содержаться в этих базах данных. Прежде всего, это конкретная информация спортсмена, такая как списки команд, которые включают биографическую информацию, включая имя, пол, возраст, контактную информацию и даже размеры одежды для униформы команды. Эта же база данных может также содержать подробную информацию о состоянии здоровья, истории выступлений или других характеристиках спортсменов и их игр.

Более того, информационные технологии применяются для жеребьевки и организации турнирных таблиц, что объясняется трудоемкостью процесса, при проведении данной сортировки вручную. Стоит отметить, что при применении данных технологий для жеребьевки гарантируется и устранение субъективных факторов при формировании групп, что, несомненно, является плюсом при организации соревнований [3, с. 226-227].

Базы данных также необходимы для других видов административной информации и анализа. Примерами могут служить бухгалтерские и деловые записи, файлы сотрудников, инвентарные запасы оборудования или записи о техническом обслуживании объекта. Чтобы быть эффективными, базы данных должны регулярно обновляться для регистрации изменений.

Если говорить о государственных спортивных школах, то в помощь их потребностям (вести учёт клиентов, учебных групп, коммуницировать с тренерским составом и родителями учеников, принимать оплаты) приходит такой инструмент как CRM-система. Во время

пандемии коронавируса весной 2020 года единственной возможностью заниматься был онлайн: благодаря «цифре» тренеры могли проводить занятия, следить за учениками по Zoom и фиксировать их результаты в CRM системе.

Информационные технологии в физической культуре активно используются в образовательных целях. Существует множество обучающих технологий, с помощью которых, обычный, даже не подготовленный человек может изучать различные спортивные дисциплины, боевые искусства, может получить достаточно полную информацию, необходимую ему для обучения. Компьютерные программы помогают преподавателю планировать физическую подготовленность, общую двигательную активность и контролировать психофизическое состояние студентов. Компьютерные технологии очень востребованы в физическом развитии, так как они базируются на быстром действии и универсальности. На методах кодировки и передачи информации основана система, которая позволяет в кратчайшие промежутки времени совершать множество разноплановых задач [6, с.552-554].

Большое распространение получили программы, позволяющие визуализировать на экране монитора изучаемый процесс. С помощью таких программных средств пользователь получает возможность изучить объект в деталях, принимая в расчет временное и пространственное движение процесса. Благодаря этим технологиям, можно добиться повышения эффективности тренировок и обучения за достаточно короткий срок [1, с. 92-93].

Цифровые технологии позволяют государству собирать данные со всех спортивных объектов, школ, мероприятий и соревнований, контролировать единые календарные планы, информацию об участниках соревнований, присвоение званий и категории. Регулируя эту информацию оперативно, повышается управляемость.

Одним из важных моментов является маркетинговое и коммерческое применение интернета. В России практически нет профессиональных спортивных команд, у которых не было бы веб-сайта, и большинство из них связаны друг с другом через различные сети. Кроме того, некоторые из них теперь требуют от своих игроков иметь адреса электронной почты для взаимодействия как с администрацией команды, так и с ее болельщиками.

Цифровые технологии активно внедряются как в профессиональный, так и в любительский спорт, что сильно мотивирует граждан подключаться к здоровому образу жизни. Умные вещи и мобильные приложения становятся неотъемлемой частью быта и помогают организовать режим дня, правильное питание, эффективные индивидуальные тренировки и многое другое. На базе фитнес-платформ клиентам предлагается персонализированный сервис, в рамках которого они получают информацию о новинках, основанных на анализе их предыдущих покупок и клиентском поведении.

Интернет, несомненно, повлиял на продажу и распространение спортивных товаров, которые являются необходимыми для спортивных команд. Ведь стоимость и качество спортивного инвентаря может быть проблемой для профессии. Однако, появление интернет-магазинов, а также различных торговых площадок помогают уменьшить затраты на спортивное оборудование.

Среди крупных брендов, которые уже достаточно продвинулись в цифровой трансформации, такие компании как Adidas, Nike. Adidas еще в 2014 году решил повысить эффективность направления электронной розничной торговли. В результате упрощения оформления заказа на сайте, а также предоставления клиенту гарантий о быстрой и бесплатной доставке конверсия заказов выросла, а также увеличилась дополнительная выручка.

В то же время, есть восторженные разговоры о молодых стартапах, которые якобы переворачивают спортивный рынок с ног на голову с помощью своих инновационных цифровых продуктов. Компании-производители спортивной одежды стали покупать технологические стартапы в сфере digital-спорта, продавать цифровые решения и устройства, а также предлагать информационные продукты для тех, кто является потребителем их традиционной продукции. В будущем, пишут визионеры, спорт может измениться более радикально. Новые ракетки, кроссовки, перчатки, футбольные мячи для достижения высоких показателей и многие другие приспособления и инновации, представляют собой изобретение новых образов, символов, норм и правил поведения, новых форм деятельности, направленных на изменение условий жизни людей, формирование нового типа мышления или восприятия мира [2, с. 69].

С помощью системы датчиков и микропроцессора спортивная электроника может контролировать многие показатели физического состояния человека: силу воздействия на тело, электрические импульсы от сердца и нервной системы, кровяное давление, ритм ходьбы или бега, нагрузку на суставы. Данные могут быть обработаны «на месте» или переданы на другое устройство [7, с. 113-115].

Еще один очевидный сценарий — продление игровой практики за счет щадящих покрытий кортов, беговых дорожек и стадионов, что снизит нагрузки на суставы и существенно отодвинет возраст «спортивной старости». Не исключено, что к 2050 году атлеты начнут использовать бионические дополнения.

На футбольном поле в обиход войдут бутсы и гетры, которые снизят утомляемость мышц. Помимо уменьшения риска травм это улучшит скорость, реакцию, выносливость, и спорт станет еще динамичнее.

Наибольший интерес вызывают технологии смешанной реальности, способные дать информацию о ситуации на поле и об игроках соперника. И если на Кубке мира 2018 года вратарь сборной Англии Джордан Пикфорд [делал](#) шпаргалки на бутылке с водой, чтобы вспомнить, кто из колумбийцев в какой угол бьет пенальти, в будущем эти данные будут перед глазами, скорее всего в виде контактных линз, что размоет связь между реальным и виртуальным миром.

И последнее, но немаловажное — аналитика. Она уже используется во всех видах спорта для выявления потенциала юниоров. Со временем искусственный интеллект научится делать прогноз до наступления половой зрелости у детей. Самые перспективные смогут претендовать на элитные программы подготовки.

Таким образом, очевидно, что информационные технологии имеют достаточно широкое применение как в спорте, так и в нашей повседневной жизни. Сегодня в индустрии спорта активно внедряются многие инновации цифровой сферы. Поэтому федерациям, профессиональным клубам и лигам приходится осваивать инструменты цифрового маркетинга и SMM-технологии. Это очень важно для увеличения доходов спортивных организаций. Цифровые платформы и интернет-ресурсы уже составляют конкуренцию в показе спортивных событий традиционным телеканалам. Можно сказать, спортивные события стремительно уходят в цифровую среду. Тенденция будет только усиливаться, причем достаточно быстро.

Литература

1. Анжаурова Е.Н., Егорычева Е.В., Шлемова М.В., Чернышева И.В. Использование информационных технологий в спорте и физической культуре // Международный журнал экспериментального образования. 2014. № 7-2. С. 92-93.
2. Васюкова В.А., Воробьева И.В., Коваленко Н.П., Куштова М.Х. Инновационная деятельность в сфере физической культуры и спорта как неотъемлемая часть элемента национальной экономики // Азимут научных исследований: экономика и управление. 2016. Т. 5. №3(16). С. 68-73.
3. Глушань В.М. Кажаров А.А., Пономарев В.К. Методы оптимального автоматизированного формирования турнирных таблиц // Программные продукты и системы, 2015. № 3(111). С. 226-232.
4. Еременко В.Н. Синько О.В. Федорова Н.П. Особенность развития спорта под влиянием научно-технического прогресса и его принятие человеком // Азимут научных исследований: педагогика и психология. 2019. Т. 8. № 3(28). С. 336-338. <https://doi.org/10.26140/anip-2019-0803-0087>
5. Ильдус Я.Л., Фанис А.М., Резеда Р.Х. Основные направления использования информационных технологий в практике спорта// Ученые записки университета им. П.Ф. Лесгафта. 2012. № 9(91). С. 88-93.
6. Калинин В.С., Машичев А.С. Компьютерные технологии в физкультуре и спорте // Молодой ученый 2020. № 49(339). С. 552-554.
7. Черепанов Е.В. Актуальные инновации в спорте // Международный студенческий научный вестник. 2015. № 3-1. С. 113-115.

© Кривогузова М.В., Ерошенко И.А., Низовцева Ю.Я., 2022

ЦИФРОВИЗАЦИЯ БОРЬБЫ С ЭКОНОМИЧЕСКИМИ ПРЕСТУПЛЕНИЯМИ

Цифровой след пользователя является одним из самых опасных явлений в информационном пространстве. Он представляет собой информацию, которая остается в Сети после различных действий пользователя. Следы — это большие данные, которые являются целью обработки многих организаций, начиная монопольными гигантами вроде Amazon и Google и заканчивая рекламными компаниями. Цифровой след бывает активным и пассивным. Активный след — это сознательные действия пользователя в Сети (например, комментарии, фотографии, переписка). Пассивный же след — это непредумышленно оставленные данные - геолокация мобильного устройства, история посещений сайтов, IP-адрес устройства, с которого пользователь выходил в сеть.

Динамически развивающаяся современная преступность на сегодняшний день ставит перед правоохранительными органами новые задачи и заставляет постоянно задумываться о разработке и использовании новых современных технологий. На начальном этапе расследования преступления важную роль отводят осмотру места происшествия, позволяющий получить всю необходимую криминалистически значимую информацию, а также изъять все вещественные доказательства, которые в последующем будут способствовать правильной квалификации преступного деяния и установлению причастных к нему лиц. Точная фиксация обстановки и объектов является важной задачей, определяющей результативность и объективность расследования. В настоящее время на смену традиционным средствам фиксации приходит современная цифровая техника.

На сегодняшний день совершается множество преступлений против собственности, к которым следует отнести кражи, мошенничества, присвоения, фальшивомонетничество, преступления в банковской сфере, совершаемые с использованием компьютерных средств и систем, оставляя цифровые следы. Так как цифровые следы быстро трансформируются и легко уничтожаются, неосторожные и медлительные действия при их выявлении в ходе производства следственных действий могут привести к непоправимым последствиям. Преступления, совершенные с помощью использования информационно-телекоммуникационных технологий, продолжают расти. Согласно статистическим данным за семь месяцев 2021 г. произошло почти 320 тысяч киберпреступлений. Это на 16% больше, чем за тот же период в прошлом году (<https://clck.ru/etB9E>).

Следовательно, выросла потребность правоохранительных органов в совершенствовании методических основ выявления, расследования и профилактики данной группы преступлений, которая обусловлена постоянно изменяющимися возможностями правоохранительных органов по сбору, использованию, хранению криминалистически значимой информации. Таким образом, цифровые методы и средства открыли новые возможности в данном направлении.

Однако необходимо сделать вывод о том, что трудности в получении криминалистически значимой информации опосредуются, в том числе, сложностью выявления пользователей информационно-телекоммуникационного оборудования, учитывая активное распространение средств сокрытия идентификационной информации пользователей сети «Интернет» и преобладание иностранных аппаратно-программных средств, выходящих за пределы правового поля законодательства Российской Федерации. Соответственно, исключается возможность установления лиц, совершивших преступления, и, следовательно, затрудняется своевременное направление дел в суд.

Немаловажную роль имеет специфичность механизмов слепообразования киберпреступлений. Это обусловлено особенностями совершения таких преступлений, в рамках которых зачастую осуществляется замена адреса злоумышленника в глобальной сети «Интернет», использование технологии «подменных» IP-адресов и абонентских номеров. На сегодняшний день, к сожалению, несмотря на очевидную необходимость их использования для раскрытия и расследования киберпреступлений, наблюдается расхождение потребностей и возможностей правоохранительных органов в этом отношении. Необходимо выделить самые распространенные способы кибермошенничества. К ним относятся:

- внедрение в системы банковского обслуживания вредоносных программ, в том числе в мобильные приложения банков;

- подмена настоящих POS-терминалов на поддельные;

- хакерские взломы мобильных устройств и банковских систем в информационно-телекоммуникационной сети «Интернет»;

- использование злоумышленниками методов социальной инженерии (например, фишинг, претекстинг и др.), вследствие которых жертвы по своей воле предоставляют им реквизиты банковских карт и иную личную информацию, позволяющую совершить хищение путем обмана. Так, например, 26.02.2020 в ходе рассмотрения Карабулакским районным судом Республики Ингушетия уголовного дела было установлено, что подсудимый ФИО7 совершил мошенничество с причинением значительного ущерба гражданину.

Осознавая противоправный характер и общественную опасность своих действий, реализуя преступный умысел путем обмана, ФИО7, открыв сайт в сети «Интернет», на котором размещена информация об абонентских номерах различных операторов сотовой связи, нашел абонентский №, находившийся в пользовании и зарегистрированный на ФИО3. После этого он попросил свою знакомую позвонить с мобильного номера оператора сотовой связи ПАО «Вымпелком», представившись сотрудником службы безопасности и сообщить абоненту о несанкционированном списывании денежных средств с его банковской карты. Будучи не осведомленной о преступных намерениях знакомая подсудимого исполнила его просьбу, после чего по его указанию прервала связь с абонентом. Затем подсудимый позвонил со своего мобильного номера и, представившись сотрудником службы безопасности, введя потерпевшего в заблуждение, сообщил ему о необходимости следования его инструкциям в целях избежания незаконного списания денежных средств с его банковской карты. Далее, подсудимый в продолжение своего преступного умысла, начал отправлять Мухиеву А.Ю. на

номер «900» посредством услуги «Мобильный банк» смс-сообщения с кодами в виде цифр «9845.50» и «4567.50», пояснив, что при условии, если в течение часа он не наберет данные коды с указанием номера, произойдет списание всех денежных средств. Согласившись, потерпевший произвел две операции, отправив на номер «900» смс – сообщения с указанием вышеуказанных кодов и абонентского номера, после чего произошло списание со счета его банковской карты двух денежных сумм в размере 9 845,50 и 4 567,50 руб. Затем, в продолжение своего преступного умысла, выяснив наличие у потерпевшего банковской карты, подсудимый попросил его открыть на своем сотовом телефоне мобильное приложение банка, после чего, введя потерпевшего в заблуждение, указал ему ввести в приложение по отдельности два кода в виде цифр «4200», «5500», произвел операции по переводу со своей банковской карты денежных средств двумя суммами в размере 4 200 и 5 500 руб. на вышеуказанный абонентский номер, оплатив при этом услугу его сотовой связи. После чего, подсудимый скрылся, получив реальную возможность распорядиться похищенными денежными средствами по своему усмотрению, в результате чего причинен значительный материальный ущерб на сумму в размере 24 113 рублей.

Суд признал подсудимого виновным в совершении преступления и назначил ему наказание в виде лишения свободы сроком два года без ограничения свободы с отбыванием наказания в исправительной колонии строгого режима [2];

- использование программ, позволяющие определить персональные данные методом подбора;

- незаконное использование электронной цифровой подписи. Так, например, 25.02.2020 в ходе рассмотрения Бийским городским судом Алтайского края было установлено, что Корневская совершила мошенничество в особо крупном размере, при следующих обстоятельствах. Вопреки своим служебным обязанностям, у бухгалтера Корневской, в период времени с 25.09.2017 по 27.11.2018, из корыстных побуждений, возник преступный умысел, направленный на хищение денежных средств в особо крупном размере, принадлежащих ФКУ КП-2 УФСИН России по Алтайскому краю (далее – Учреждение), выделенных на оплату труда и иные выплаты его работникам, на протяжении длительного периода времени.

Реализуя свой преступный умысел в действие, Корневская, находясь на рабочем месте в помещении Учреждения, руководствуясь корыстной целью, зная, что ей к выплате за отработанное время в качестве оплаты труда бухгалтера Учреждения полагаются денежные средства в сумме 246 540,69 рублей, используя установленный в помещении бухгалтерии Учреждения компьютер, подключенный для работы посредством электронного документооборота с УФК по Алтайскому краю, не посвящая должностных лиц Учреждения, обладавших правом электронной цифровой подписи, выданных для работы в системе СУФД, и передавшим ей соответствующие электронные носители с ЭЦП, сформировала электронные заявки на кассовый расход ФКУ КП-2 УФСИН России с основанием платежей в качестве оплаты труда и иных выплат, в которые вносила заведомо ложные для нее сведения о якобы полагающихся ей к выплате денежных

средствах, завывсив их на сумму, равную 1 702 434,52 руб. Продолжая реализовывать свой преступный умысел Корневская подписала подготовленные ею заявки на кассовый расход ЭЦП и посредством системы СУФД отправила в УФК по Алтайскому краю, тем самым обманывая сотрудников указанного органа, которые, проверив их на подлинность ЭЦП и на правильность заполнения, не подозревая, что в указанных документах содержатся заведомо ложные для Корневской сведения, сформировали платежные поручения на зачисление денежных средств со счета открытого УФК по Алтайскому краю для ФКУ КП-2 УФСИН России в Отделении Сибирского главного управления Центрального банка РФ на счета сотрудников Учреждения согласно реестрам. Работники банка на основании указанных поступивших платежных поручений перечислили денежные средства в сумме 1 702 434,52 рубля с названного расчетного счета на счет Сбербанка.

Корневская, продолжая реализовывать свой преступный умысел, составила электронные реестры на зачисление денежных средств сотрудникам ФКУ КП-2 УФСИН России в качестве оплаты труда и иных выплат, внося в них заведомо ложные сведения о якобы полагающихся ей к выплате денежных средствах в сумме 1 948 975,21 рублей, заввышенных на 1 702 434,52 рубля, после чего отправляла их посредством системы «Сбербанк Бизнес Онлайн» в Сбербанк, тем самым обманывала сотрудников указанной организации.

Своими умышленными преступными действиями Корневская похитила денежные средства в сумме 1 702 434,52 рубля, принадлежащие ФКУ КП-2 УФСИН России по Алтайскому краю, чем причинила материальный ущерб в особо крупном размере.

Суд пришел к выводу признать Корневскую виновной в совершении преступления, предусмотренного ч. 4 ст. 159 УК РФ, и назначить ей наказание в виде 3 лет лишения свободы [1].

Таким образом, механизм слепообразования компьютерного мошенничества достаточно специфичен, в связи с особенностью его способа, который предполагает замену адреса пользователя в информационно-телекоммуникационной сети «Интернет», использование технологии «подменных» мобильных номеров посредством SIP-телефонии, что способствует превалированию цифровых следов.

Как правило, в этих случаях необходимо производство следственных действий, направленных на выявление следов в оперативной памяти компьютера, на носителях информации различных типов, в линиях связи. Особенность таких следов заключается в том, что они имеют высокую скорость трансформации, легко уничтожаются и модифицируются, могут быть представлены практически бесконечным количеством копий, легко распространяются в компьютерных сетях и доступны в любой точке, где имеется подключение к сети «Интернет». При этом их локализация осуществляется не в конкретном месте (например, местонахождение преступника), а по пути прохождения всего информационного сигнала, что позволяет правоохранительным органам осуществлять их выявление и фиксацию.

Характерными особенностями цифровых следов является:

1. вероятность многопользовательского доступа;
2. возможность создания копий на цифровые носители;
3. дистанционный доступ злоумышленников к информации, способствующий модификации или ее уничтожению.

Очевидно, что перечисленные черты препятствуют эффективной и грамотной работе лица, осуществляющего производство по уголовному делу, который не обладает соответствующими знаниями. Также необходимо обратить внимание на проблему выбора специалиста, имеющего необходимую компетенцию.

Немаловажно, что существующие в настоящее время методики работы с цифровыми следами недостаточно разработаны, что в свою очередь ограничивает возможности специалистов в данном направлении. Представляет также интерес комплекс факторов, отрицательно сказывающихся на деятельности уполномоченных государственных органов по выявлению, раскрытию и расследованию преступлений, совершенных с использованием IT-технологий, а именно с использованием криптовалют в криминальных взаиморасчетах при совершении незаконных сделок. Данная противоправная деятельность осуществляется активно посредством конвертации денежных средств в виртуальную валюту, оборот которой не подконтролен для правоохранительных органов, что препятствует реализации механизмов противодействия легализации доходов, полученных противоправным путем, применительно к подобным цифровым финансовым активам. В инфокоммуникационной сети «Интернет» представлено множество площадок по обмену криптовалют на их рублевый эквивалент, находящихся вне юрисдикции Российской Федерации. Использование криптовалют во взаиморасчетах, в совокупности с осуществлением маскировки реальных IP-адресов отрицательно сказывается на деятельности государственных органов по выявлению, раскрытию и расследованию киберпреступлений, что в свою очередь способствует ограничению их возможностей касемо использования информации о финансовых операциях в процессе доказывания (например, наличные и безналичные расчеты, кассовые операции, перевод или размен денежных средств, обмен одной валюты на другую) лиц, в отношении которых имеется достаточно веских оснований считать их причастным к совершению киберпреступлений. Подводя итоги, следует сделать следующие выводы:

1. Необходимо регламентировать алгоритм деятельности специалиста по выявлению, фиксации и изъятию информационных следов.
2. Разработать стандарт, предусматривающий квалификационные требования к специалистам в данном направлении.
3. Создать программы переподготовки или повышения квалификации специалистов в исследуемой области.
4. Сформулировать перечень необходимых сертифицированных программных средств для обеспечения эффективного производства следственных действий.

Хотелось бы отметить, что, несмотря на высокие достижения научно-технического прогресса, а также внедрение новых возможностей цифровизации, существуют пробелы, связанные с недостатком формирования у обучающихся необходимых навыков и умений

аналитического подхода к оценке ситуации, выявлению и фиксации информационных следов, что, в свою очередь, способствует снижению результативности производства следственных действий, и, соответственно, влечет утрату криминалистически значимой информации.

Данная проблема требует радикальных шагов со стороны соответствующих ведомственных организаций. Ведь тщательная подготовка и неформальный аналитический подход к проведению следственных действий, бесспорно, способствует высокой эффективности производства по расследованию киберпреступлений. Достаточно высокую эффективность в раскрытии информационно-телекоммуникационных преступлений в сети «Интернет» имеют экспертные исследования, проведение которых возможно только при условии использования компетентностного практикоориентированного подхода в обучении.

Сегодня сложность расследования и раскрытия киберпреступлений заключается в том, что существующие базы данных и информационные системы хранят только персональные данные пользователей, а не данные об используемых ими устройствах. Не стоит забывать о таком важнейшем направлении в формировании баз данных, как автоматизированные поисковые системы. Они аккумулируют большие объёмы информации, что в перспективе позволяет их идентифицировать на основе электронно-цифрового следа многих гаджетов. На сегодняшний день распространение цифровых технологий развивается огромными темпами, соответственно увеличение их оборотов также усиливает риски возникновения новых способов преступлений.

В связи с появлением новых видов шифрования информации, имеющей решающее значение для установления истины по уголовным делам о преступлениях рассматриваемого вида, все более широкое применение технологии «блокчейн» требует интенсификации научных исследований и разработок для пресечения таких специфических действий криминала. В связи со стремительными темпами развития экономической парадигмы современного информационного общества данные разработки становятся актуальными.

В связи с этим очевидной становится роль криминалистики в противодействии с современной передовой экономической преступности XXI века. Говоря не только о реализации устоявшихся традиционных направлениях криминалистической тактики, её техники и методике в новых реалиях информационного общества, криминалистика способна объединить возможности всех наук уголовно-правового блока и эффективно бороться с современной экономической преступностью.

Литература

1. Приговор Бийского городского суда Алтайского края от 25.02.2020 г. по делу № 1-35/2020. <https://sudact.ru>
2. Приговор Карабулакского районного суда Республики Ингушетия от 26.02.2020 г. по делу № 1-11/2020. <https://sudact.ru>

© Кучинская А.В., 2022

УДК 004.02

Лац Е.В., Кетько Н.В., д-р экон. наук
Волгоградский государственный технический университет
г. Волгоград, Россия

ПОВЫШЕНИЕ ЭКОНОМИЧЕСКОЙ ГРАМОТНОСТИ ПОТЕНЦИАЛЬНЫХ ПРЕДПРИНИМАТЕЛЕЙ, С ИСПОЛЬЗОВАНИЕМ ОБУЧАЮЩЕЙ ЭКОНОМИЧЕСКОЙ ИГРЫ-ЭМУЛЯТОРА

Малый бизнес во всем мире играет важную роль проводника достижений НТП в массовое производство, что и определяет актуальность его развития в России. В настоящее время наблюдается массовое банкротство предприятий сферы малого бизнеса, основной причиной сложившейся ситуации наряду с экономическими, политическими, демографическими является общая экономическая безграмотность будущих предпринимателей, которые не знают или очень слабо представляют, с чем им придется столкнуться, какие задачи им придется решать. Из-за нехватки знаний они могут совершить немало ошибок, которые способны привести к банкротству.

Например, несостоятельности компании может поспособствовать отсутствие сбыта продукции из-за некачественной рекламы или ошибочных маркетинговых исследований. Также со временем фирмы могут терять прибыль из-за устаревания оборудования и невозможности состязаться с конкурентами на равных. Большинство организаций, которые успешно выходят на рынок, начинают масштабировать свой бизнес путем привлечения заемных средств, но необдуманность заимствования и отсутствие тщательного планирования могут привести к банкротству. Также случается, что к банкротству приводит слишком узкая специализация фирмы. Например, если у предприятия очень мало клиентов, или оно реализует проекты для государственных закупок, то высока вероятность, что со временем предприятие перестанет отвечать тендерным требованиям, либо потеряет небольшую клиентскую базу. В этом случае придется искать новых клиентов, а также другие средства к существованию (<https://clck.ru/dWVbq>).

В России остро стоит проблема экономической грамотности, как обычных граждан, так и предпринимателей. Низкий уровень знаний о финансовых продуктах и отсутствие сформированных навыков финансового поведения (например, отсутствие «подушки безопасности») приводит к экономической нестабильности предприятий. Нехватка знаний не дает возможности бизнесменам подходить к развитию своего предприятия с точки зрения долгосрочного планирования (так, по данным исследования аналитического центра НАФИ 2017 года, 77% респондентов планируют расходы максимум на год и всего 4% — более чем на год (<https://clck.ru/dWVes>)).

Субъекты малого и среднего предпринимательства (МСП) играют огромную роль в обеспечении экономической и политической стабильности, они выступают ключевым фактором эффективного развития рыночной экономики. Благодаря своему огромному созидательному потенциалу МСП придает экономике необходимую гибкость, активизирует ресурсный потенциал, выступает в качестве инструмента социально-экономического

развития, стимулирует потребительский рынок и предпринимательскую инициативу. МСП в Российской Федерации являются не только необходимым этапом для развития экономической системы нашей страны, но и ключевым элементом для развития общества в целом, так как они в значительной степени являются основным элементом для формирования среднего класса, который обеспечивает социально-экономическую стабильность общества, а также его политическое равновесие и демократическое развитие [4, с. 68].

Поскольку малое и среднее предпринимательство является значимой составляющей экономики любой страны, повышение уровня экономической грамотности будущих предпринимателей представляет собой одну из важных задач его развития. Так как повышение экономической грамотности в рамках посещения различных образовательных программ занимает много времени и отвлекает от производственного процесса, более эффективным способом в настоящее время является создание условий ведения бизнеса в игровой форме. В связи с этим автором была разработана «игра-эмулятор», суть которой состоит в том, что пользователь, вводя различные ключевые для работы предприятия данные, моделировал свою будущую работу, пробуя управлять различными направлениями еще до создания реального бизнеса [3, с. 567-604].

Программа призвана помочь будущим предпринимателям распланировать свою деятельность в процессе создания и развития бизнеса. Благодаря ей у пользователя будет возможность узнать о важных моментах и оценить величину затрат заранее, а не в момент их возникновения. Например, в начале игры пользователю понадобится ввести данные об оборудовании, и будущий предприниматель в этот момент сможет оценить, какое оборудование ему нужно, сколько оно стоит и какой у него срок службы. Программа поможет оценить, хватит ли стартового капитала, познакомит с основными финансовыми показателями, важными для оценки эффективности предприятия и даже поможет выявить реальное состояние своего смоделированного предприятия (рис. 1).

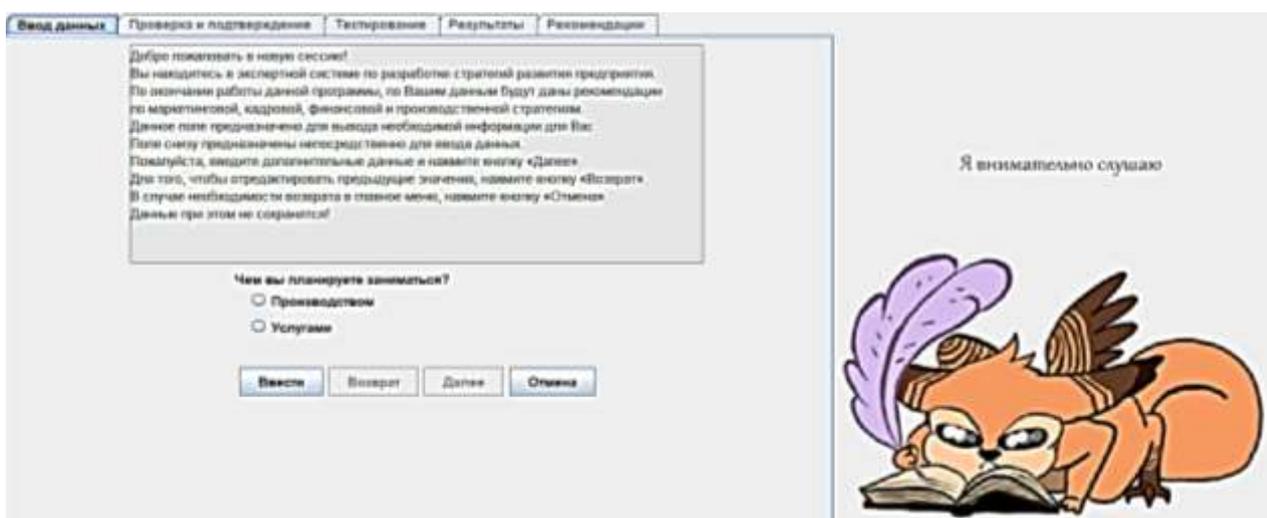


Рис.1. Выбор вида деятельности

Поскольку от выбора вида деятельности зависят особенности функционирования предприятия, на первом этапе пользователю предлагается выбрать, чем он будет заниматься: оказывать услуги или производить продукцию. Его выбором будет определен набор задач и вопросов, которые необходимо будет решить в процессе игры. Для того, чтобы адекватно рассчитать себестоимость продукции или услуги, любому предпринимателю необходимо определить сумму всех расходов своего предприятия. Одним из ключевых пунктов расходов является зарплата персонала (рис. 2).

Введите зарплату основного персонала:

Ввести Возврат Далее Отмена

Рис. 2. Ввод зарплаты персонала

Деятельность предприятия возможна без сотрудников, но в настоящее время персонал является главным ресурсом любой организации, поскольку успех любого бизнеса напрямую зависит от работы его сотрудников. Квалифицированные работники в хорошей рабочей среде отличаются высокой производительностью [5, с. 135]. В связи с этим заработная плата является основным источником стимулирования и дохода занятых на предприятии. Она зависит от квалификации работника, сложности, количества, качества и условий осуществляемого труда. Для успешного функционирования предприятия важно грамотно определять размер заработной платы, поскольку адекватный её уровень позволит привлечь и удержать хороших специалистов, что в свою очередь будет влиять на успех деятельности компании [2, с. 131].

Другой важной составляющей, влияющей на функционирование предприятия, является оборудование. В сфере производства оно играет ключевую роль, тогда как сфера услуг зависит от оборудования гораздо меньше. Для успешной конкуренции необходимо качественное оборудование, поскольку это будет непосредственно влиять на качество выпускаемой продукции. Хорошие качественные товары обладают высокой конкурентоспособностью. Оборудование, которое позволит производить такие товары является дорогостоящим. Помимо кадрового состава предпринимателю до момента начала хозяйственной деятельности необходимо также определить, какое оборудование нужно будет закупить: дорогое или более дешевое, в соответствии с особенностями, выбранного вида деятельности (рис. 3). Это позволит не только оценить и спланировать действия по обеспечению предприятия необходимым оборудованием, но и определить необходимый размер стартового капитала.

Введите данные об оборудовании:

Наименование	Цена	Срок службы
PC	50000	2-3 года
Точка доступа Wi-Fi	65000	5-7 лет
Отбойный молоток	15000	1-2 года
Весы лабораторные	50000	3-5 лет
Котлы паровые	300000	7-10 лет

Добавить строку Удалить строку

Ввести Возврат Далее Отмена

Рис. 3. Ввод данных об оборудовании

Важнейшей расходной частью производства и его организации на предприятии являются сырье и материалы. В сфере производства материалы нужны будут в любом случае: сырье, смазочные материалы, топливо, комплектующие и многое другое. А со сферой услуг всё не так однозначно, не во всякой деятельности, связанной с оказанием услуг, материалы составляют значительную долю расходов [6, с. 118]. Например, агентство по интернет-продвижению не нуждается в материалах, а вот служба по ремонту и обслуживанию принтеров должна будет закупать как минимум картриджи для различных моделей принтеров.

Материалы являются важной составляющей хозяйственной деятельности предприятия, так как от их качества зависит и качество производимой продукции или оказываемых услуг. Следовательно, на начальных этапах предпринимателю следует оценить, какие именно материалы и в каких количествах нужны ему для начала деятельности (рис. 4). В процессе прохождения этапов игры она будет запрашивать у пользователя сумму материалов (независимо от того, выбрал он сферу производства или услуг), тем самым обращая его внимание на этот важный пункт и побуждая учесть его при планировании будущей деятельности (рис. 5).

Требуется ли оказание услуг при использовании материалов?

Материалы нужны

Материалы не нужны

Ввести Возврат Далее Отмена

Рис. 4. Вопрос про материалы

Введите сумму этих материалов:

Ввести Возврат Далее Отмена

Рис. 5. Ввод суммы материалов

Если в процессе прохождения этапов игры пользователь понимает, что ему не хватает собственного капитала, то он может смоделировать ситуацию с привлечением дополнительных финансовых ресурсов из кредитных источников (рис. 6).

Рис. 6. Вопрос про кредит

В первую очередь программа предлагает рассмотреть возможность развития предприятия за счет кредитных средств. Пользователь может подумать над оформлением кредита для начала деятельности, а программа позволит определить, сможет ли он его выплатить, т.е. определит его целесообразность (рис. 7).

Рис. 7. Поля для ввода суммы, ставки и срока кредита

Собственных средств организации и ее владельцев не всегда хватает на создание, расширение или сохранение бизнеса. Кредит позволит предпринимателю быстрее открыть свой бизнес, так как при выборе формирования первоначального капитала в виде накопления средств, необходимо делать поправку на инфляцию, которая обесценивает сбережения (<https://clck.ru/dWVim>) (так, по оценке Росстата инфляция в России в 2021 году составила 8,39% (<https://clck.ru/dWVh4>)).

Данный программный модуль позволяет пользователю выбрать наиболее эффективный путь привлечения дополнительных финансовых потоков для своего будущего бизнеса. Для оценки уровня теоретической экономической базы в программе предусмотрено тестирование по следующим направлениям: процесс организации производства, маркетинг, управление кадрами, финансовое состояние (рис. 8-10).

В рамках тестирования программа задает пользователю ряд вопросов, отвечать на которые пользователь должен и в процессе ведения деятельности. В игре предусмотрена проверка на противоречие ответов пользователя, за счет взаимосвязанности задаваемых вопросов (так, например, если пользователь сказал, что у него высокая конкуренция, а потом при перечислении конкурентов указывает всего троих, то он противоречит ранее введенному ответу). Также в процессе тестирования программа на основе ответов пользователя и введенных им ранее данных формирует стратегии: финансовую, производственную, кадровую и маркетинговую [1]. В случае противоречивых данных программа выводит сообщение, что та или иная стратегия не была сформирована. Если же противоречий не возникло, то программа выводит информацию о том, какую стратегию она считает оптимальной для предполагаемого предприятия и выдает некоторые советы о том, что предпринимателю следует сделать (например, если в процессе работы программа определит, что для предприятия характерна открытая кадровая политика, то в качестве одного из советов будет выведено, что пользователю «необходимо стабилизировать возможности карьерного роста для персонала»).

Ввод данных | Проверка и подтверждение | **Тестирование** | Результаты | Рекомендации

Каким образом будут организованы технологические процессы?

- Технологические процессы организуются по новым продуктам и современным технологиям
- Максимально автоматизировать процесс выпуска на имеющемся оборудовании без значительных капитальных вложений
- Будет произведена модификация оборудования для повышения гибкости производства

89 %

Ответить
Далее
Отмена

Рис. 8. Вопрос про процесс организации производства

Ввод данных | Проверка и подтверждение | **Тестирование** | Результаты | Рекомендации

Перечислите основных конкурентов: (если конкурентов нет, введите в строку цифру 0)

Конкуренты
0

Добавить строку | Удалить строку

4 %

Ответить
Далее
Отмена

Рис. 9. Вопрос про конкурентов

Какое количество сотрудников Вы планируете принимать в кадровую службу?

- Бухгалтер будет совмещать, этого достаточно
- Один сотрудник отдела кадров
- Планирую отдел, но не более 3 человек
- Планирую полноценную кадровую службу, так как кадры решают всё

39 %

Ответить
Далее
Отмена

Рис. 10. Вопрос про управление кадрами

В настоящей статье автором приведены результаты собственной разработки – обучающей «игры-эмулятора» повышения экономической грамотности потенциальных предпринимателей. Данная обучающая игра является актуальной в настоящее время, так как множество банкротств в сфере малого бизнеса связано с переоценкой предпринимателей собственных возможностей и недооценкой отраслевых особенностей сферы, где они вели свой бизнес, конкурентной среды и многих других факторов. Данная игра позволит потенциальным предпринимателям объективно оценить размер стартового капитала, а также текущие расходы, определить востребованность продукции или услуги на рынке. Пройдя обучение, пользователь сможет оценить возможности успешного развития предполагаемой деятельности, что в свою очередь позволит сократить банкротства предприятий сферы малого бизнеса и будет способствовать ее эффективному развитию.

Литература

- Ансофф И. Стратегическое управление. М.: Экономика, 1989.
- Атамазова А.А. Заработная плата: ее роль и функции // Молодой ученый. 2016. № 16(120). С. 131-133.
- Шилдт Г. Java 8: руководство для начинающих. М.: И.Д. Вильямс, 2015. 720 с.
- Маковецкий М.Ю. Малое и среднее предпринимательство как ключевой элемент экономического развития Российской Федерации // Вестник Московского университета имени С. Ю. Витте. Серия 1: Экономика и управление. 2020. №4 (35). С. 66-74. <https://doi.org/10.21777/2587-554X-2020-4-66-74>
- Романцова О.Н. Персонал как главный ресурс любой организации // Молодой ученый. 2020. № 18 (308). С. 135-138.
- Турлович Я.В. Сущность и виды материальных ресурсов, их значение для предприятия // Молодой ученый. 2020. № 24 (314). С. 117-119.

© Лац Е.В., Кетько Н.В., 2022

АНАЛИЗ ПРИЧИН ОТКАЗА ОТ ИСПОЛЬЗОВАНИЯ VIRTUAL DOM НА ПРИМЕРЕ SVELTE

В современном интернет пространстве среди разнообразия сайтов и веб-приложений важную роль в успехе компании или бизнеса играет пользователь. С развитием подходов к созданию интерфейсов он становится все более требовательным к дизайну, взаимодействию с элементами, времени, которое тратит на получение необходимой информации. Например, одним из таких подходов является применение анимаций.

Использование анимации в интерфейсах стало определённым подходом к повышению качества взаимодействия с пользователем. Такой подход используется для большого числа элементов страницы, например, анимирование элементов при наведении или клике, плавное появление меню. В то же время анимации, а также другие CSS свойства, снижают производительность веб-ресурса, так как могут заставлять браузер повторно проходить этапы рендеринга, что занимает время и снижает эффективность работы сайта или приложения для пользователя.

Рендеринг страницы в браузере содержит следующие этапы (<https://clck.ru/epUrT>):

1. Построение Document Object Model (DOM).
2. Построение CSS Object Model.
3. Компоновка этих двух деревьев в единый render tree.
4. Этап Layout, в течение которого для элементов рассчитывается положение на странице.
5. Этап Painting, в течение которого происходит отрисовка на экране.

Следовательно, некоторые CSS свойства могут требовать больших затрат, чем другие, затрагивая несколько или все этапы рендеринга. Например, свойство transform, которое часто используется для создания анимаций, не требует пересчета положения элемента и повторной отрисовки (<https://clck.ru/epUt9>).

С ростом требований к производительности и интерактивности появлялись JavaScript фреймворки. Каждый из них предлагает собственные подходы к созданию эффективных приложений, в том числе к работе с DOM API. Например, фреймворки Vue и React используют механизм Virtual DOM (VDOM).

Модель виртуального DOM представляет собой легковесную копию реального DOM. Данная концепция была создана для того, чтобы иметь возможность часто обновлять DOM без серьезных потерь производительности. Далее будут рассмотрены детали работы Virtual DOM на примере React.

Для обновления модели DOM React использует алгоритм сверки (Reconciliation) (<https://clck.ru/epUus>). При изменении компонентов в React приложении будет создана новая копия Virtual DOM, содержащая уже новые измененные данные. Для вычисления тех узлов,

которые должны быть обновлены в реальном DOM, используется алгоритм diffing (<https://clck.ru/epUus>). Он сравнивает предыдущую и новую версию Virtual DOM, выделяет точно их различия. После сравнения алгоритм будет искать наилучший подход для внесения изменений в реальный DOM, что дает возможность снизить затраты на обновление реальной модели DOM.

В 2019 году был представлен новый фреймворк Svelte, одна из концепций которого - это отказ от виртуального DOM. По утверждению разработчиков модель программирования, аналогичная той, которую хотели достичь разработчики React с Virtual DOM, может быть достигнута без его использования (<https://clck.ru/SEsZm>).

Svelte представляет собой технологию для создания веб-приложений, основанных на создании компонентов пользовательского интерфейса и реализации взаимодействий между ними. Svelte дает возможность писать код с использованием привычных приемов для веб-разработчика, то есть с помощью HTML, CSS и JavaScript.

На официальном сайте Svelte указаны основные особенности данного инструмента, а именно (<https://clck.ru/SEsZm>):

1. Меньшее количество кода,
2. Нет Virtual DOM,
3. Реактивность.

Svelte требует меньшего количества кода по сравнению с другими фреймворками, например, с Vue, Angular. При малом объеме кода разработчикам нужно меньше времени, чтобы разобраться в чужом коде, это означает, что большая часть времени будет посвящена разработке компонентов, приложения, оптимизации. Соответственно, меньшее количество кода поможет легче находить и исправлять возможные ошибки.

С применением реактивного подхода изменяемые данные будут моментально внесены в DOM. Реактивные переменные, присваивания отмечаются в коде особым знаком. В случае, если одно значение изменилось, тогда все другие значения, связанные с первым, будут также вычислены повторно.

Использование Virtual DOM имеет свою цену, которая выражается как во времени, затрачиваемом на повторную отрисовку, так и в размере готового приложения. Принцип работы алгоритма, лежащего в основе концепции использования VDOM требует некоторых временных затрат в начале каждого цикла повторного рендеринга для сравнения состояний виртуального и реального DOM. По этой причине Рич Харрис, создатель Svelte, называет Virtual DOM «чистым накладным», то есть расходуя ресурсы вне зависимости от того будет ли нужен повторный рендеринг или нет (<https://clck.ru/SEsZm>). В то же время чтобы React просто запустился в приложении, клиенту в любом случае отправляется дополнительный пакет кода, в котором содержатся в том числе и инструкции по постройке VDOM (по некоторым показателям до 150 КБ). Идеология создателя Svelte и здесь идет вразрез с подходом React и заключается в том, что «самый оптимизированный код - код, которого не существует», то есть стремится к приложению, которое не содержит в себе любой части кода без необходимости.

Чтобы реализовать эту идеологию, в основу Svelte был вложен производительный механизм по-настоящему реактивного рендеринга без использования VDOM. Вместо этого, Svelte действует в качестве компилятора и генерирует «идеальный» JavaScript код, который изменяет DOM напрямую, без использования алгоритма диффинга. Для этого компилятор получает на вход файлы “.svelte”, в несколько шагов считывает из них JavaScript, стили и разметку, разбирает и строит из них AST (Абстрактное дерево синтаксиса):

1. JavaScript код преобразуется в AST при помощи пакета `acorn`. Этот шаг нужен для того, чтобы извлечь из кода, содержащегося в теге “script”, операторы с реактивностью и операторы с входными данными, именуемые “export”. По полученному AST программе легко найти все операторы нужного типа.

2. CSS стили считываются и выражения исследуются на наличие селекторов, которые затем трансформируются в AST с дополнениями в виде уникальных селекторов (`.example` into `example.xyz123``)

3. Для обработки HTML используется свой собственный парсер, который трансформирует код в древовидную структуру наподобие той, что получается при трансформации JavaScript кода в AST. Каждый тег представляется объектом, у которого есть такие атрибуты как название, свойства и дочерние элементы.

После всех преобразований на выходе получается JavaScript файл, содержащий в себе логику для монтирования компонентов, обработки событий и обновления состояния DOM (<https://clck.ru/epV2M>). Таким образом Svelte на стадии компиляции известны все точки в приложении, где происходит изменение состояния и на основе этих данных принимает решение обновлять настоящий DOM.

Для оценки эффективности веб-приложений различные программы веб-аудита дают возможность измерить метрики производительности, в том числе те, которые связаны с рендерингом. Существует большое число различных метрик, но есть основные полезные метрики, которые наиболее точно отражают производительность. Далее будут рассмотрены такие метрики, как Time to Interactive (TTI) и Byte Weight.

Time to Interactive показывает сколько времени требуется странице, чтобы стать полностью интерактивной. Это значит, на странице уже начали отображаться данные из DOM, было зарегистрировано большинство обработчиков событий у видимых элементов, и страница отвечает на ввод пользователя за 50 миллисекунд или быстрее.

Byte Weight – общий размер всех ресурсов, запрошенных страницей без сжатия.

После рассмотрения технологии virtual DOM, которую используют React и Vue, а также проанализировав причины отказа от данной концепции и новый подход, используемый в Svelte вместо VDOM, было решено провести экспериментальное исследование, которое будет заключаться в измерении метрик производительности и загрузки приложений.

В качестве базы было взята работа Штефана Краузе - JS Framework Benchmark (<https://clck.ru/UVZvW>). Она заключается в проведении набора тестов для некоторого числа фреймворков, в результате создается таблица, которая содержит итоговые значения для различных тестов. Интерфейс всех приложений единый (<https://clck.ru/UVZvW>). На главной

странице расположены кнопки для управления списком. Например, рендеринг тысячи элементов или их удаление.

Метрики, которые измеряются в результате проведения тестов, направлены на получение показателей производительности, запуска приложения, а также использования памяти (<https://clck.ru/epVD6>). Рассматриваемые метрики производительности собираются после первоначальной загрузки приложения, то есть на них не влияют такие факторы, как, например, размер пакета.

Было решено сравнивать метрики приложений на React, Vue, так как данные фреймворки используют концепцию virtual DOM, и, соответственно, на Svelte. В тестируемых приложениях были использованы отличные данные для создания случайного списка, а также добавлены CSS свойства, снижающие эффективность рендеринга, а именно тени (box-shadow) и градиент (linear-gradient).

Каждое действие пользователя в приложении соответствует измерению одной метрики производительности. Из девяти доступных метрик для эксперимента были выбраны:

1. Рендеринг одной тысячи элементов,
2. Рендеринг десяти тысяч элементов,
3. Удаление одного элемента,
4. Удаление одной тысячи элементов,
5. Замена одной тысячи элементов,
6. Обновление каждого десятого элемента,
7. Поменять местами две строки.

В качестве метрик, отражающих эффективность первоначальной загрузки приложения, были выбраны метрики TTI и Byte Weight, о которых было сказано выше. Эти метрики были измерены с помощью инструмента веб-аудита Lighthouse (табл. 1, 2).

Таблица 1

Результаты выполнения набора тестов

	React	Vue	Svelte
Создать 1000 элементов	166.2	142.0	123.3
Создать 10000 элементов	1,741.1	1,274.2	1,264.3
Удалить 1 элемент	21.1	20.1	18.5
Удалить 1000 элементов	541.8	455.3	669.7
Заменить 1000 элементов	141.6	115.9	117.0
Обновить каждый десятый элемент	1,939.5	2,122.2	1,795.4
Поменять местами две строки	889.8	79.4	76.1

Таблица 2

Результаты измерения метрик с помощью Lighthouse

	React	Vue	Svelte
TTI	1.3	1.1	0.9
Byte Weight	45 kB	24 kB	5 kB

По результатам вычисления значений набора метрик производительности Svelte имеет лучшие показатели практически по всем тестам, кроме тех, которые направлены на замену тысячи элементов и удаления так же тысячи элементов. По результатам измерения метрик загрузки, Time to Interactive и Byte Weight, Svelte, в том числе показал наилучшие результаты по сравнению с React и Vue. Таким образом, можно сделать вывод, что приложения на Svelte показывают хорошую производительность, с учетом остальных преимуществ данного инструмента, таких как реактивность, меньший объем кода, удобная стилизация компонентов, Svelte является эффективным и перспективным фреймворком для создания пользовательских интерфейсов.

© Летон Г., Глазько П.Е., 2022

УДК 004

Ликсина Е.В., канд. пед. наук, **Вьюнов Д.А.**
Пензенский государственный технологический университет
г. Пенза, Россия

РАЗРАБОТКА МОБИЛЬНОГО НАВИГАТОРА ПО ЗДАНИЮ

Проблема ориентации в учебной организации является актуальной в последние время. Каждый год в ведущие вузы страны поступают более тысячи студентов РФ (ВО и СПО), а также сотни иностранцев. Они прибывают в Российскую Федерацию на обучения, из следующих стран, таких как Таджикистан, Турцию, Украину, Египет и др. Первостепенная задача процесса адаптации студентов – это умение ориентироваться в образовательном учреждении, быстро найти аудиторию, необходимый отдел, центр, подразделение [1, с. 87].

Для изучения подразделений университета можно воспользоваться данными с официального сайта вуза, они представляют из себя сумбурный набор информации, который не запоминается. В связи с этим, выбор технологических решений по ориентации в здании принимает первостепенное значение [3, с. 77].

Для решения проблемы необходимо было разработать приложение для ориентации в помещении организации. При разработке приложения необходимо было создать макет внутреннего обустройства учебной организации. При создании 3D-компонентов для последующего конструирования виртуального пространства необходимо выбрать программное обеспечение для 3D-моделирования. На рынке 3D-редакторов имелось большое количество программ, начиная от платных, до бесплатных и условно бесплатных. Согласно статистика сайта 3dtoday.ru был проведен анализ самых популярных программ для 3D-моделирования. По результатам анализа самой популярной программой была признана Blender, которая обладает большими функциональными возможностями, а также возможностями настройки отдельных инструментов. Несомненными преимуществами является и то, что она бесплатная и для ее освоения не требуется никаких специализированных умений (<https://clck.ru/ep9jA>).

После определения программы для создания 3D объектов необходимо было приступить к непосредственному моделированию первого этажа здания. За основу был взят план эвакуации, как корректный и подлинный образец. После анализа плана эвакуации 1 этажа, была смоделирована первая модель виртуального пространства (рис. 1).

Аналогично было проведено моделирования второго, третьего и четвертого этажей главного корпуса университета. Моделирование также основывалось на подробном изучении планов эвакуации вышеперечисленных этажей. Каждый этаж создавался отдельно друг от друга, это было сделано для удобства и упрощения работы. По завершении этапа моделирования отдельных этажей, все модели были соединены в одну единую модель [2, с. 51].

Полученный FBX-файл был импортирован в среду разработки Unity, и после настройки положения и отображения камеры, был адаптирован для последующей работы.

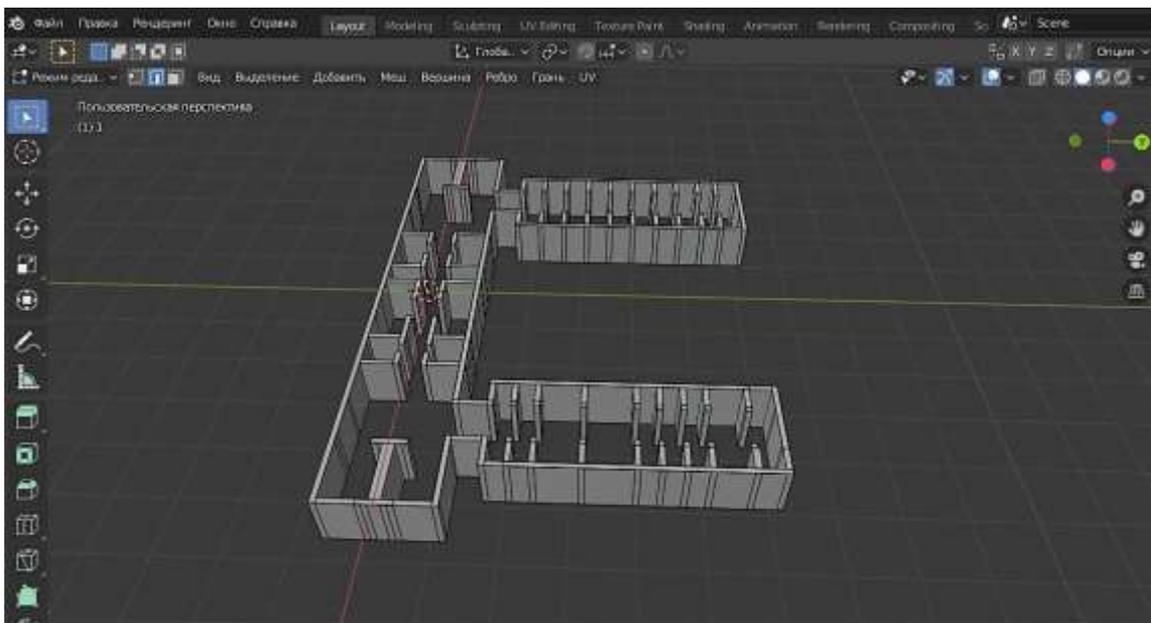


Рис. 1. Модель этажа

Следующим шагом стало написание кода приложения, в который входило разработка алгоритмов действия приложения при выполнении пользователем определенных действий (нажатие кнопок, взаимодействие с «ползунком», настройка поведения камеры). Общий объем системного кода приложения составлял 615 строк.

После разработки программной части приложения был создан и спроектирован дизайн приложения.

Для проведения этапа первичной апробации в качестве целевой аудитории были отобраны следующие категории студентов:

- 10 студентов обучающиеся по программе ВО (граждане РФ);
- 5 студентов обучающиеся по программе СПО (граждане РФ);
- 10 студентов обучающиеся по программе ВО (иностранцы).

По результатам тестирования приложения, которое длилось около 40-50 минут, была получена необходимая обратная связь. У 60% первых пользователей не возникло вопросов по использованию приложения. Однако были выявлены и некоторые проблемы в использовании приложения. Так 20% пользователей не смогли до конца разобраться в функциональных возможностях приложения, а еще 20% – в особенностях навигации по приложению.

По результатам первого тестирования на основании обратной связи, полученной от пользователей было принято решение о доработке отдельных элементов дизайна и модулей приложения.

- В ходе исправления были осуществлены следующие действия:
- упрощена навигация по приложению;
 - облегчена цветовая гамма интерфейса;
 - создан новый интерфейс приложения;
 - прописаны подсказки для пользователей.

Итоговый вариант приложения имеет следующий вид (рис. 2).

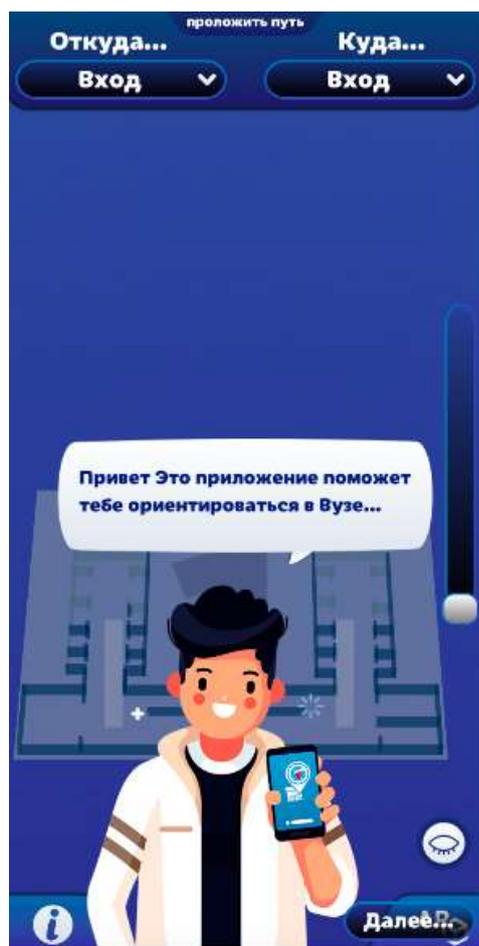


Рис. 2. Мобильное приложение

Повторное тестирование обновленной мобильной версии приложения показало улучшение его эргономики. Тестирующими было отмечено, что изменение цветовой гаммы улучшило взаимодействие пользователей с элементами интерфейса, добавленные подсказки облегчили понимание навигационных возможностей приложения, а новый интерфейс упростил работу с приложением. Все эти моменты повлияли на улучшение работы приложения, были оставлены в его окончательном варианте,

На завершающем этапе работы над мобильным приложением необходимо, было все вышеупомянутые наработки соединить в единую систему. Для этого надо было сделать следующие действия:

- проверить приложение на наличие багов;
- проверить стабильности работы;
- оптимизировать итоговый объем приложения.

После вышеперечисленных доработок приложение было подготовлено к экспорту из среды разработки. Для этого в среде разработки был выбран формат. арк, который позволяет при установке приложения на мобильное устройство использовать самый простой алгоритм. Общий объем готового приложения составил 26 Мбайт. Файл с расширением. арк можно перенести на смартфон и далее установить как обычный установочный пакет.



Литература

1. Ликсина Е.В., Вьюнов Д.А. Технологии реализации проекта «Разработка AR навигатора» // Сборники конференций НИЦ Социосфера. 2020. №29. С. 87-89.
2. Ликсина Е.В., Вьюнов Д.А. Условия внедрения технологии дополненной реальности в профессиональное образование // Непрерывное образование в вузе: вызовы и тренды, меняющие пространство технологий управления, обучения воспитания и развития: Сб. трудов Международной научно-практической конференции. 2019. С. 50-52
3. Ликсина Е.В., Вьюнов Д.А. Применение технологии дополненной реальности в адаптации иностранных студентов в образовательных организациях высшего образования // Социосфера. 2019. №41. С. 76-78.

© Ликсина Е.В., Вьюнов Д.А., 2022

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ПРОГНОЗИРОВАНИЯ ИЗМЕНЕНИЯ ЦЕН АКЦИЙ НА МОСКОВСКОЙ БИРЖЕ РАЗЛИЧНЫМИ МОДЕЛЯМИ МАШИННОГО ОБУЧЕНИЯ

В связи с ростом числа частных инвесторов на российском рынке, возросла актуальность применения компьютерных технологий для прогнозирования изменения роста цен акций. Прежде чем конкретизировать задачу, рассмотрим способы участия в работе рынка более подробно. Существуют довольно успешные попытки анализировать актуальную информацию из новостей методами машинного обучения, к примеру, Yumo Xu and Shay B. Cohen смогли добиться точности выше 58% в своей работе, используя информацию из Twitter для прогнозирования цен [6]. Метод использования фундаментального анализа, бесспорно, представляет большой интерес, однако стоит отметить, что помимо своей сложности, он не является универсальным, поскольку далеко не по каждому активу легко получить актуальную информацию. В то же время для сторонников краткосрочной торговли, эти знания не являются необходимыми, поскольку для прогнозирования цен трейдеры, как правило, пользуются техническим анализом, который основан на поиске определенных последовательностей цен и объемов торгов.

Стоит отметить, что технический анализ очень богат на подходы к прогнозированию движения цен. Так, например, многие трейдеры пользуются классическим анализом, включающим построение уровней поддержки и сопротивления, линий тренда и поиск различных паттернов. Другие используют компьютерный анализ, основным методом которого является расчет индикаторов, значения которых зависят от цен и объемов торгов в определенные даты. Кроме этого, существует метод свечного анализа, придуманный в Японии, когда прогноз делается, исходя из особого представления графика движения цены. Несмотря на такое разнообразие инструментов, все они могут быть получены, если известны цены на актив и объемы торгов. Также, пожалуй, их общим недостатком является некоторая неоднозначность сигналов – нередки случаи, когда два разных человека видят на одном графике прямо противоположные сигналы. Чтобы избежать подобных неточностей и ошибок в этой работе рассматривается задача создания модели, которая была бы способна достаточно хорошо прогнозировать направление движения цены на коротком временном промежутке.

Существует немало подходов к решению этой проблемы. Например, авторы в [5] решали задачу классификации недельного направления движения выбранного портфеля. В своей работе они использовали фьючерсы на американские казначейские облигации сроком на 5 и 10 лет с целью хеджирования и минимизации риска. В работе [1] представлен метод, заключающийся в непосредственном предсказании значения цены, в [2] проводился анализ возможности прогноза сразу по нескольким акциям и выбор тех из них, которые показали наилучший результат.

В итоге задача была сформулирована следующим образом: зная цены и объемы торгов выбранным активом за i -ый день, а также скорректированные цены закрытия за несколько предыдущих дней, классифицировать, будет ли в $i + 1$ -ый день движение цены вверх или вниз, или его не будет вовсе. Таким образом, решалась задача тернарной классификации. В качестве актива были выбраны обыкновенные акции ПАО «Сбербанк».

Все инструменты технического анализа опираются на 2 важнейших параметра – это цена и объем. В течение торгового периода принято выделять 5 цен, а именно: максимальная и минимальная цены, цены открытия, закрытия, а также скорректированная цена закрытия, которая учитывает корпоративные действия, такие как выплата дивидендов или дробление акций. Все параметры были получены нами при помощи Yahoo Finance.

К каждой записи в наборе данных в качестве параметров были добавлены цены закрытия (ЦЗ) за n предыдущих дней. Довольно сложно сказать, сколько именно дней нужно взять, чтобы прогнозы получались наилучшими, поэтому было принято решение сделать это настраиваемым гиперпараметром модели. Целевая переменная выделялась по следующему правилу: если разность между ЦЗ и ценой открытия в i -ый день положительна/отрицательна и ее модуль составляет $a\%$ или более от цены открытия, а она, в свою очередь, больше/меньше минимума/максимума цены i -ого дня не более чем на $a\%$, то запись $(i - 1)$ -го дня принадлежит классу $+/-$. В противном случае запись принадлежит классу 0. Подобное деление на классы дает возможность использования модели на реальных данных: получая прогноз модели в i -ый день, мы полагаем, что на следующий день цена изменится по меньшей мере на $a\%$, а значит можно открыть сделку в соответствующем направлении. Однако, анализировать только разность цен открытия и закрытия нельзя, поскольку в таком случае мы лишаемся возможности оценить доходность модели – всегда можно сказать, что если прогноз сделан верно, то мы гарантированно заработаем $a\%$ от суммы сделки, открывая позицию утром и закрывая, как только цена изменится на нужный процент. Но если прогноз будет ошибочен, то наши потери ничем не ограничены: так, например, ожидая роста на 1%, мы можем держать позицию до конца дня, при этом цена может упасть на 10%. Получится, что каждая ошибка модели будет стоить нам слишком дорого. В предложенном нами варианте, вывод об ошибке модели можно сделать сразу, как только цена достигнет определенного порога, поэтому в таком случае в каждой сделке наши потери будут составлять не более $a\%$ (на самом деле, из-за существования комиссионных и проскальзывания выигрыш всегда будет меньше, чем $a\%$, а проигрыш – больше, однако учет этого – дело контроля над капиталом, а не торговой стратегии).

Для проведения экспериментов были загружены данные с 2012 по 2021 года, данные с 2012 по 2019 год были разделены на тренировочный и валидационный наборы в соотношении 70:30, последние 2 года были выбраны в качестве тестового периода.

Для оценки качества работы использовался следующий подход: модель, принимая на вход ряд параметров, в качестве ответа выдает один из трех классов: «+», «-», «0». Истинное значение тоже можно отнести к одному из трех этих классов. Если получен класс «+», значит прогнозируется рост цены, значит нужно покупать, если «-» - продавать, а если «0» - не стоит

входить в рынок. Это значит, что если модель спрогнозировала «0», то независимо от реального положения дел открытия сделок не происходит, а значит потери денег также нет. А поскольку при прогнозировании «0» мы точно не получим ни прибыли, ни убытка, в конечной оценке объекты этого класса не участвуют. Если было верно предсказано движение в определенном направлении (две ситуации из шести оставшихся), значит, может быть получена прибыль. Количество таких случаев будем прибавляться к общей оценке. Количество оставшихся случаев будет вычтено из общей суммы. Таким образом, будет получено число сделок, прибыль от которых составит $a\%$. Стоит отметить, что a тоже является гиперпараметром нашей модели, и, очевидно, что 5 сделок с прибылью в 0,5% и 5 сделок с прибылью в 2% - это совсем не одно и то же. Поэтому для получения конечной легко интерпретируемой оценки домножаем полученную сумму на число a .

Прогнозирование направление движения цены – задача нетривиальная. Изначально невозможно сказать, в какой зависимости находятся цены предыдущих дней и цена следующего дня, поэтому заранее трудно определить, какая модель лучше подойдет для решения поставленной задачи. Изучение работ той же тематики тоже не помогло определить четко выбор модели: например, в работе [4] предлагается использовать сложные составные модели, а в работе [3] авторы используют относительно простые нейронные сети. В настоящей работе проводятся испытания для следующих моделей и методов: метод k ближайших соседей, логистическая регрессия, случайный лес, градиентный бустинг, метод опорных векторов.

Метод k ближайших соседей был выбран первым, т.к. он является простым и легко интерпретируемым. Результаты экспериментов для метода ближайших соседей показаны на рисунке 1. При увеличении числа соседей наша оценка увеличивается, что свидетельствует об улучшении качества модели. Увеличение значения параметра приводит к увеличению значения оценки, однако сделать ее положительной не удаётся. Возможно, такая проблема возникает из-за того, что в тренировочных данных слишком много объектов классифицированы как «0». Чтобы исправить это, уменьшили пороговое значение отклонения, однако результат остался неудовлетворительным (рис. 1).

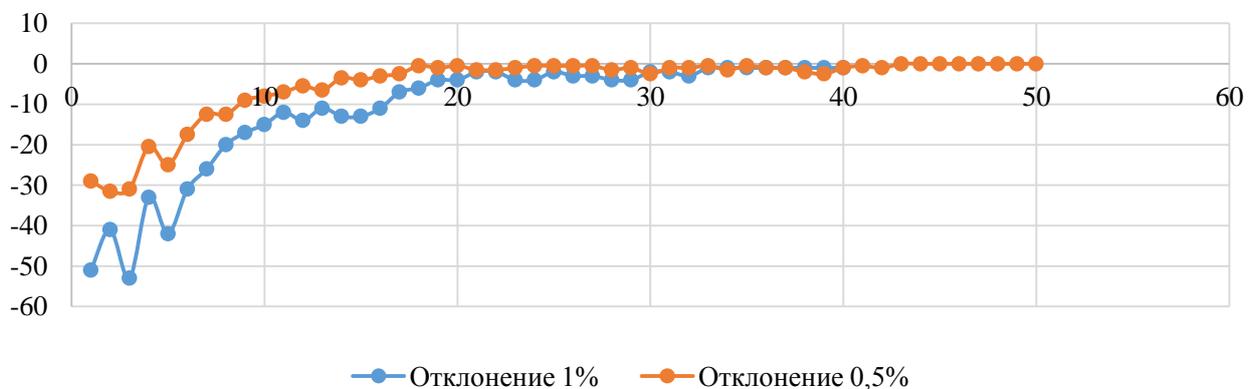


Рис. 1. Зависимость доходности от параметра k в методе ближайших соседей при различных значениях отклонений на валидационном наборе данных

Логистическая регрессия. Линейные классификаторы являются одними из самых простых, однако некоторые исследователи не пренебрегают их использованием в своих работах [5, с. 4]. Но линейная модель не сработала – все данные валидационного набора были классифицированы, как «0», независимо от заданного отклонения, набора дат, метрики расчета штрафа и параметра C .

Случайный лес – одна из наиболее распространенных моделей машинного обучения. В отличие от куда более примитивных деревьев решений, случайные леса, могут быть достаточно хорошо настроены на тренировочных наборах данных и при этом иметь неплохую обобщающую способность. Большим плюсом этой модели является то, что один из ее параметров, а именно количество деревьев, участвующих в голосовании, связан с качеством монотонной зависимостью – чем больше, тем лучше. Это связано с тем, что увеличение числа деревьев в модели позволяет уменьшить роль каждого конкретного дерева, а значит препятствует переобучению. Еще 2 важных параметра – максимальная глубина и максимальное количество признаков, участвующих в вычислении штрафа, также влияют на способность модели обучаться. В качестве максимальной высоты дерева брали простые числа от 2 до 53, а второй параметр вычисляли, как округленный вниз квадратный корень из количества признаков объекта. На рисунке 2 показана зависимость доходности от максимальной высоты дерева.

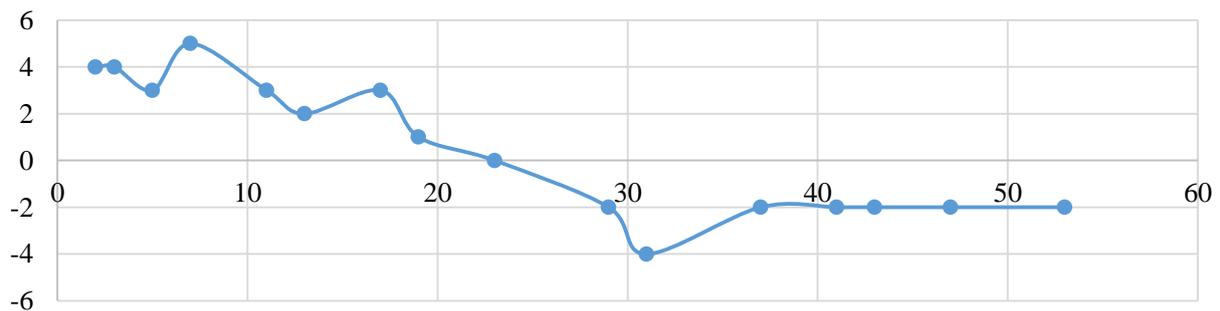


Рис. 2. Зависимость доходности от максимальной высоты дерева при обучении случайного леса

Анализ метода на тестовом наборе показал, что при большинстве значений параметра высоты, наша оценка составила порядка –100%. Это в некотором смысле является подтверждением теоретических сведений о моделях на основе деревьев решений. Дело в том, что с марта 2020 года по октябрь 2021 обыкновенные акции ПАО «Сбербанк» находились в растущем тренде и, что самое важное, большую часть времени торговались на исторических максимумах. Таким образом, деревьям из случайного леса было необходимо экстраполировать результаты, полученные в процессе обучения, на новые значения цен. Однако идея дерева решений такова, что оно не может корректно распространять известные результаты на новые данные, возможно, именно поэтому получили такую большую разницу между прибыльностью на валидационном и тестовом наборах.

Градиентный бустинг деревьев регрессии. Базовой единицей модели для этого метода также является дерево решений, однако принцип работы у этого алгоритма несколько иной. Теперь каждое следующее дерево пытается улучшить результаты, полученные предыдущим,

поэтому их количество в ансамбле нельзя увеличивать сколь угодно сильно. В целом, при помощи наиболее важных параметров настраивается то, насколько сложной может получиться новая модель, поэтому важно грамотно подобрать параметры так, чтобы с одной стороны учесть как можно больше деталей, а с другой стороны получить хорошую обобщающую способность. Результаты экспериментов для метода градиентного бустинга представлены на рисунке 3.

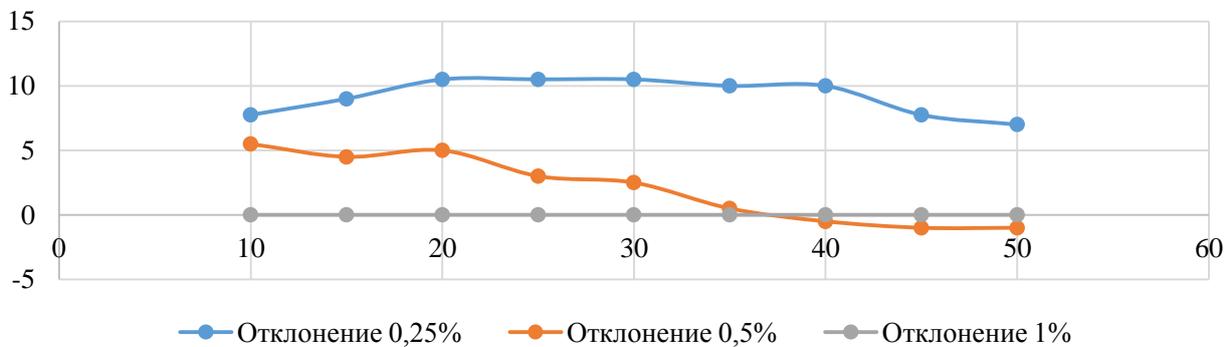


Рис. 3. Зависимость доходности от числа деревьев в методе градиентного бустинга на валидационном наборе данных

В итоге получили модель, которая работает хоть и лучше, чем случайный лес, все же дает слишком нестабильные результаты, поэтому говорить о применимости ее на реальных данных в том виде, в котором она есть сейчас, нельзя.

Метод опорных векторов — последняя модель, которая была рассмотрена. Качество работы модели регулировалось параметрами C и γ . Наиболее интересные результаты были получены при анализе цен последних двух недель с пороговым отклонением 1,5%. Результаты экспериментов для метода опорных векторов показаны на рисунке 4.

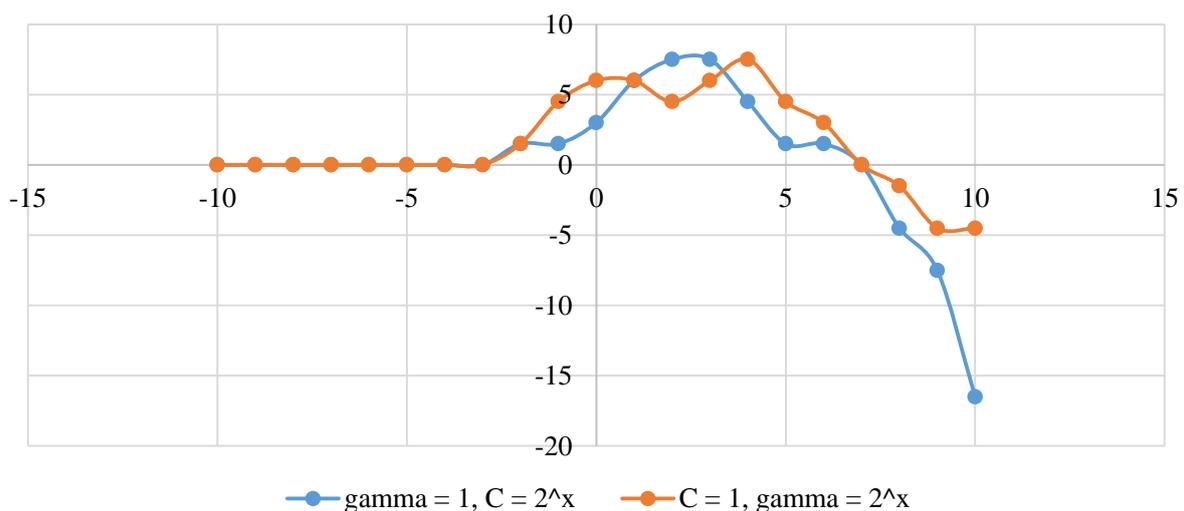


Рис. 4. Зависимость доходности от параметров C и γ в методе опорных векторов на валидационном наборе данных

Таким образом, в ходе работы были проанализированы 5 различных методов. В таблице представлены результаты исследований по каждой модели на наилучших наборах параметров.

Таблица

Анализ значений ошибок для различных методов прогнозирования

Модель	Набор дат	Порог отклонения	Параметры	Оценка
Метод k ближайших соседей	1 неделя	0,50%	k = 18	0,50%
Логистическая регрессия	1 неделя	0,25%	penalty = "l2", C = 1	0%
Случайный лес	1 неделя	1%	max_depth = 7, max_features = 4	-12%
Градиентный бустинг	1 месяц	0,25%	n_estimators = 20, learning_rate = 0,01, max_features = 3	7,75%
Метод опорных векторов	1 месяц	1,50%	gamma = 1, C = 8	7,50%

Лучший результат среди всех моделей показал градиентный бустинг деревьев регрессии. Метод опорных векторов дал почти тот же результат, но на других значениях гиперпараметров. Подтверждение оценки, полученной на валидационном наборе тестовым показывает, что модель способна делать прогнозы на новых данных.

Таким образом, в ходе экспериментов были получены две модели с высокой точностью работы, что подтверждается экспериментами на валидационной и тестовой выборках. Однако делать выводы о применимости этих алгоритмов к реальным данным преждевременно. Во-первых, полученные результаты будут проверены на новых данных – очевидно, что хорошая модель должна уметь корректно обрабатывать и другие финансовые инструменты, поэтому в следующих экспериментах будут проводиться многочисленные проверки на новых акциях Московской биржи. Во-вторых, действуя описанным способом, мы создали нечто похожее на торговый индикатор, поскольку мы не используем ничего, кроме цен и объемов. Он отличается от привычных аналитикам тем, что его сигнал невозможно трактовать двусмысленно и, пожалуй, явную формулу его расчета выписать не получится. А значит в следующие эксперименты также необходимо включить анализ существующих рыночных индикаторов. Еще одним способом повышения качества работы моделей может оказаться рассмотрение других таймфреймов, или целых их совокупностей.

Литература

1. Agarwal A.K., Kumari S. Gold Price Prediction using Machine Learning // International Journal of Trend in Scientific Research and Development (IJTSRD). Vol. 4. Issue 5. 2020. Pp. 1448-1456.
2. Ghosh P., Neufeld A., Sahoo J. K. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests // Finance Research Letters. 2021. Pp. 102280
3. Madahar A., Ma Y., Patel K. Application of a Shallow Neural Network to Short-Term Stock Trading // arXiv preprint arXiv:1703.10458. 2017.



4. Sawhney R. et al. Deep attentive learning for stock movement prediction from social media text and company correlations // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. С. 8415-8426.

5. Sharang A., Rao C. Using machine learning for medium frequency derivative portfolio trading // arXiv preprint arXiv:1512.06228. 2015.

6. Xu Y., Cohen S.B. Stock movement prediction from tweets and historical prices // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers). 2018. С. 1970-1979.

© Маляров Д.А., Волкова С.С., 2022

БЕЗОПАСНОСТЬ «УМНОГО ГОРОДА» В ПРОЦЕССЕ ЦИФРОВИЗАЦИИ ГОРОДСКОЙ СРЕДЫ

Бесспорно, утверждение, что мы движемся к эре умных городов. Страны по всему миру внедряют умные города, например, в Великобритании, Китае, России и Сингапуре. Умные города – это область, в которой пересекаются информационно–коммуникационные технологии (ИКТ), науки о городах, экологические науки и социальные науки. Эта междисциплинарная область, хотя и сложна в своем составе, но в то же время представляет интерес, так как она изменит жизнь миллионов жителей городов. В тоже время в умных городах появятся угрозы в различных формах, будь то данные, доступ, контроль, услуги, подключение и т.д. Предотвращение угроз всегда является требованием и проблемой для умных городов. Существует также недостаток внимания к кибербезопасности умных городов.

Между людьми и городами есть некоторые сходства. Люди живут и работают в городах. Жизнеспособность города связана с его деловой активностью и деятельностью горожан. Если посмотреть, из чего состоит человек, то это мозг, органы чувств и тело. Мозг человека – это интеллектуальный центр, где происходят рассуждения и решения, где обрабатывается и хранится информация. Наши органы чувств (глаза, уши, нос, кожа и т.д.) позволяют нам собирать информацию и отправлять ее в мозг для обработки. Наконец, человеческое тело обеспечивает все остальные функции, необходимые для поддержания жизнедеятельности. Аналогичным образом, мы можем рассматривать умный город, имеющий подобную человеческому телу форму строения.

Как показано на рисунке 1, три строительных блока умного города выглядят следующим образом:

1. Мозг — центр управления и принятия решений умного города. Это центральный командный пункт, где анализируются наблюдения, отслеживаются и выявляются аномалии и запускаются соответствующие действия.

2. Органы чувств — входы в системы умного города включают: зрение, звук, показания датчиков и т.д. Зрение принимает форму камер, расположенных в определенных местах. Микрофоны могут использоваться для записи разговоров на местах преступлений и в других случаях. Так же, датчики расхода воды могут использоваться для обнаружения наводнений, а умные счетчики – для учета использования электроэнергии и воды.

3. Тело — тело системы умного города относится как к инфраструктуре, так и к ИКТ-платформам. Под инфраструктурой понимаются здания, дороги, водостоки и т.д. ИКТ представляет собой три основные составляющие: связь, данные и искусственный интеллект (ИИ).

С точки зрения ИКТ, умный город имеет четыре основных уровня: датчики или конечные точки, края (шлюзы), платформа (данные, ИИ, управление связью) и их применение. На более высоком уровне абстракции умный город является аналогом живого существа (ведь без Интернета вещей (IoT) не обойтись) [2, с. 207]. Компания Alibaba рассматривает умный город как «мозг города» [4, с. 8], где находится основная часть ИИ, для управления его деятельностью и контролировать все объекты (внутренние и внешние).

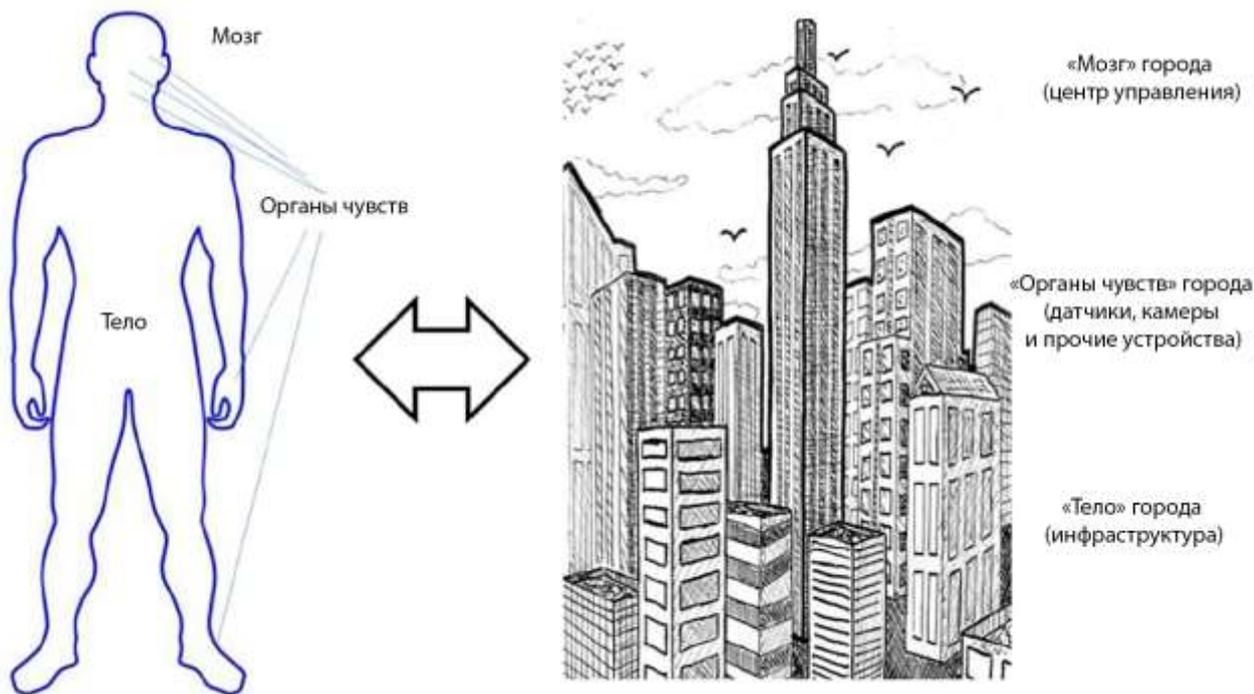


Рис. 1. Аналогия человека и умного города

Ожидается, что для умного города IoT станет неотъемлемой частью всей городской платформы [2, с. 209]. «Входами» системы умного города будут датчики, будь то видекамеры, микрофоны, датчики температуры, водомеры и т.д. Датчики – это устройства, которые взаимодействуют с физическим миром, собирают данные и передают их обратно в точки сбора по различным проводным (оптоволокно, провода и т.д.) или беспроводным (Wi-Fi, Bluetooth, узкополосный IoT и т.д.) каналам связи. Точки сбора обычно называются шлюзами. Датчики обычно используют как телеметрический транспорт с очередью сообщений или как протокол обмена сообщениями с ограниченным прикладным протоколом для отправки данных по беспроводным каналам связи на шлюз. Сообщения обычно представляют собой пакеты небольшого размера и имеют малый вес для экономии энергии, потребляемой датчиками при передаче этих данных. Отдельные секции умного города, расположенные на облачной платформе, могут взаимодействовать со шлюзом через стандартный REST API.

Как показано на рисунке 2, системная архитектура умного города включает четыре горизонтальных и два вертикальных слоя, а их приложение располагаются сверху. Основными четырьмя горизонтальными слоями являются слои устройств и датчиков, за

которыми следуют слои подключения, управления IoT и данных ИИ. Первый уровень касается датчиков и других устройств (камер и т.д.), которые обеспечивают входные данные для системы умного города. Данные датчиков затем подключаются к точкам сбора (шлюзам) через слой подключения. Уровень управления соединениями IoT, расположенный на границе, обеспечивает сбор различных данных датчиков, отправленных по различным каналам и различным протоколам, обеспечивая совместимость разнородных устройств и соединений. Наконец, уровень данных ИИ обеспечивает обработку данных, аналитику и интеллект.



Рис. 2. Предполагаемая архитектура системы умного города, в которой безопасность и управление городом охватывают все уровни

На верхнем уровне различные части умного города будут по-разному использовать данные и идеи, а также инициировать соответствующие действия (например, вызов полиции на место преступления для обеспечения безопасности умного города). Технологии искусственного интеллекта могут быть использованы для анализа данных и прогнозирования правильных результатов (например, распознавание лиц для идентификации преступника, находящегося под следствием). Таким образом, уровни данных и обработки присутствуют в каждом разделе умного города. В идеале можно создать один общий уровень платформы данных и ИИ для обслуживания всех возможных разделов умного города, но в реальности этого трудно достичь, учитывая тот факт, что существуют различные административные домены управления (когда транспортный уровень может не разрешить, например, уровню здравоохранения доступ и обмен данными), а централизованный уровень платформы может привести к узкому месту во время сбоев.

Вертикальные уровни архитектуры касаются безопасности, эксплуатации и управления контролем. Поскольку некоторые фрагменты умного города разрабатываются для частных лиц, например, для частных домовладений или частных бизнес-зданий, они считаются самостоятельными системами, практически не требующими взаимодействия с другими разделами умного города. Однако некоторые из них (например, умный транспорт, здравоохранение и окружающая среда) обслуживают население и создаются правительством, следовательно, нуждаются в управлении операциями и системой, взаимосвязи и

совместимости между собой, комплексной защитой. Например, интеллектуальный операционный центр (ИОС) [4, с. 10] или центр командования и управления – это структура, которая позволяет правительству централизованно управлять различными подсистемами умного города (транспорт, окружающая среда, общественная безопасность и т.д.). Такая конструкция также позволит соединять и взаимодействовать различным правительственным частям умного города. Другой пример, в случае преступления, произошедшего на дорожной развязке, полиции потребуется доступ к таким «приложениям», как «умный транспорт», «умный перекресток» и «умные фонарные столбы», чтобы собрать все данные, связанные с преступлением. Она также должна иметь возможность управлять камерами, расположенными вблизи или на месте преступления. Таким образом, для государственных отделов умного города необходимо, чтобы эти отделы взаимодействовали друг с другом и обеспечивали обмен соответствующими данными [1, с. 3].

Система умного города и все ее части должны быть защищены, а средства безопасности должны быть встроены в каждый уровень платформы умного города, из конца в конец, чтобы обеспечить целостность, предотвратить вторжение и злонамеренные атаки. Это объясняет, почему безопасность охватывает все горизонтальные слои. Устройства должны быть защищены, чтобы обеспечить их защиту от взлома. Данные датчиков, передаваемые по защищенному проводному или беспроводному каналу связи, должны быть зашифрованы. Доступ к устройствам должен предоставляться только авторизованному персоналу, как локально, так и удаленно. Данные должны надежно храниться в распределенном хранилище. Доступ к данным должен быть предоставлен только в авторизованном виде. Данные, связанные с каждым аспектом умного города, могут рассматриваться как отдельная сущность, надежно хранящаяся в частном или публичном облаке. Они могут быть общими или храниться в секрете. Многие страны разработали политику, связанную с управлением данными, и в зависимости от места хранения данных власти имеют юрисдикцию над данными и их неправильным использованием.

Безопасность умных городов относится к более широкой проблеме, и ее не следует рассматривать изолированно. Под изолированностью мы подразумеваем разработку решений безопасности для конкретных сегментов умного города, таких как «умный транспорт», «умное здравоохранение», «умная окружающая среда», «умная жизнь» и т. д. [3, с. 85] Следует рассмотреть и понять, каковы основополагающие факторы, которые будут способствовать безопасной работе умного города.

В сущности, необходимо обеспечить безопасность «мозга», «тела» и «органов чувств» умного города в дополнение к обеспечению безопасности каждой части умного города. Это требует комплексного и полного решения по безопасности для защиты умного города. Как показано на рис. 3, решение по безопасности не только защитит город от злоумышленников за его пределами, но и защитит его внутреннее пространство – инфраструктуру, связь и сервисы.



Рис. 3. Защитный экран безопасности умного города эквивалентен наличию многоуровневой защиты, начиная с защиты мозга, органов чувств и тела

При разработке решений по обеспечению безопасности для «умных» городов необходимо изучить: что нужно защищать, каковы конечные цели злоумышленников и типы атак. Чтобы понять, что нужно защищать в умном городе, необходимо рассмотреть его активы. Активы умного города — это его ресурсы, люди и услуги. Под ресурсами подразумеваются финансовые активы – деньги, товары (золото и т.д.), другие активы (акции и т.д.), инфраструктуру – телекоммуникации, интернет, центры обработки данных, офисы, дома и т. д. и предметы первой необходимости – воду, электричество, продукты питания, медицинские препараты и т. д. (рис. 4).



Рис. 4. Три актива умного города. а) Люди, б) Ресурсы, с) Услуги.
Все три актива нуждаются в безопасности и защите

Люди формируют кадровый резерв и рабочую силу города, они поддерживают бизнес-операции и двигают экономику города. Наконец, под городскими службами имеется в виду здравоохранение, водо- и электроснабжение, санитарные и канализационные службы, полицию, скорую помощь и т.д. Все это — важные активы для умного города. У злоумышленников может возникнуть соблазн украсть такие активы. Однако не все из них соблазняются на кражу. Другие конечные цели включают злоумышленников, которые больше заинтересованы в создании хаоса путем нарушения или прекращения работы служб. Кроме того, большинство атак являются злонамеренными, и они, как правило, делятся на следующие две категории: 1) атаки, нарушающие жизнь жителей города и 2) атаки,

нарушающие бизнес–операции в городе. Обе эти конечные цели травмируют жизнь и экономическое благосостояние города.

Кроме того, тип атак может определяться местом. Локальные атаки – физическое присутствие на месте или поблизости от места атаки. Именно здесь командные центры с видеонаблюдением помогают захватить нападающих и идентифицировать их. Удаленные атаки – находятся вдали от места проведения атаки, но используют подключение к Интернету для получения контроля над компьютерными системами, чтобы нарушить работу городских служб. Такие злоумышленники обычно проживают за пределами города, и их необходимо выслеживать по сетевым адресам интернет–протокола (IP). Не теряя внимания, следует определить, какие ресурсы и услуги являются более критичными по сравнению с другими. Критически важные ресурсы и услуги — это те, которые необходимы для повседневной нормальной работы города, которые поддерживают жизнь граждан и ведение бизнеса. Исходя из этого, были определены следующие шесть наиболее критических ресурсов и услуг, которые необходимо защитить, и они представлены в таблицах 1 и 2.

Таблица 1

Список критически важных ресурсов, обеспечивающих существование умного города

Номер ресурса	Ресурс
1	Вода
2	Энергия (электричество)
3	Связь
4	Данные
5	ИОС («мозг» умного города)
6	Финансовые активы

Таблица 2

Список критически важных служб, обеспечивающих существование умного города

Номер службы	Служба
1	Обеспечение правопорядка
2	Здравоохранение (медицинские услуги)
3	Пожарно-спасательные службы
4	Транспортные услуги
5	Услуги ЖКХ

Развитие умных городов началось по всему миру и поставило перед промышленностью и правительством серьезные задачи. Умный город неполноценен без комплексной системы безопасности и механизмов защиты его граждан, предприятий, операций и услуг. В данном исследовании подчеркивается важность безопасности для умного города, приводится список возможных атак и определяются области для обеспечения безопасности и защиты, такие как вода, энергия, данные, связь (интернет), городской «мозг» и критически важные городские службы (полиция, пожарные, банки, здравоохранение и транспорт).

Безопасность умных городов — это новая важная тема, которая требует более глубоких и дальнейших исследований, и проблему следует рассматривать как безопасность системы систем, начиная с устройств и заканчивая провайдерами и пользователями.

Литература

1. Пахомов Е.В. Технологическая основа умного города // Инженерный вестник Дона. 2017. № 3(46). ivdon.ru/ru/magazine/archive/n3y2017/4366
2. Строев П.В., Решетников С.Б. «Умный город» как новый этап городского развития // Экономика промышленности. 2017. Т. 10. № 3. С. 207-214. <https://doi.org/10.17073/2072-1633-2017-3-207-214>
3. Цибарева М.Е., Васяйчева В.А. Оценка эффективности внедрения элементов «умного города» в процессе цифровизации городской среды // Вестник Самарского университета. Экономика и управление. 2020. Т. 11. № 2. С. 83-91. <https://doi.org/10.18287/2542-0461-2020-11-2-83-91>
4. Jansäter G., Olsson J. Cybersecurity in smart cities: Not a primary concern. Master thesis. 2018. <https://clck.ru/eq6k7>

© Массеров Д.Д., 2022

АВТОМАТИЗАЦИЯ ПРОВЕРКИ ДОКУМЕНТОВ ФИЛИАЛА ОАО «РОССИЙСКИЕ ЖЕЛЕЗНЫЕ ДОРОГИ»

Отдел приема документов и сопровождения учета хозяйственных операций является подразделением Самарского территориального общего центра обслуживания – структурного подразделения Куйбышевского регионального общего центра обслуживания – структурного подразделения Центра корпоративного учета и отчетности «Желдоручет» – филиала ОАО «Российские железные дороги». Сотрудники данного отдела взаимодействуют с заказчиками, входящими в сеть ОАО РЖД. Их работа заключается в контроле правильности и полноты заполнения первичной учетной документации (ПУД) по совершенным заказчиками операциям в сфере хозяйственной жизни в соответствии с альбомом форм ПУД.

Альбом форм ПУД – это сборник эталонных унифицированных форм документов, применяемых для документирования деятельности организации (<https://cde.osu.ru/>). В качестве хозяйственной жизни в данном отделе рассматриваются все операции, которые способны повлиять на финансовое положение организации и на движение денежных средств. Примерами могут быть: оплата труда, покупка необходимой техники, оплата задолженностей, наличие штрафов. Документы в отдел передаются заказчиками в отсканированном или в оригинальном виде. В случае несоответствия документа с образцом заполнения ПУД, происходит возврат документа обратно заказчику с целью исправления недочетов. На настоящий момент сверка и проверка правильности заполнения пришедшей документации с образцами происходит вручную сотрудниками, что значительно снижает продуктивность работы данного отдела. Поэтому возникла потребность в создании автоматизированной проверки текстовых документов. Для решения поставленной задачи были проанализированы формы и шаблоны документов. Выяснилось, что в организации 100 видов документации. Данные виды делятся на 3 группы:

1. учет основных средств (38 видов документов);
2. учет нематериальных активов (4 вида документа);
3. учет труда и его оплаты (58 видов документов).

Все виды документов можно разделить на несколько подгрупп по однородности их структуры и содержания.

Всего удалось выделить 11 подгрупп, в которых чаще всего встречаются следующие поля:

1. Наименование формы документа;
2. ОКПО (Общероссийский классификатор предприятий и организаций);
3. БЕ (Балансовая единица);
4. номер документа;

5. дата составления документа;
6. ОКУД (Общероссийский классификатор управленческой документации) и др.

Каждая подгруппа также имеет свои дополнительные схожие элементы.

В связи с анализом было принято решение разделить задачу проверки наполнения электронных документов на несколько этапов: распознать текст документа, реализовать классификатор документов на основе данных, содержащихся в самих документах, и соотнести их к определенной группе документов с целью дальнейшего анализа их содержимого. На рисунке 1 представлен один из примеров документа с указанными часто встречающимися полями.

			Специализированная форма № ФНА-2 Утверждена распоряжением ОАО «РЖД» от 15.12.2008 № 2688р		
УТВЕРЖДАЮ Руководитель организации-сдатчика			УТВЕРЖДАЮ Руководитель организации-получателя		
должность	подпись	расшифровка подписи	должность	подпись	расшифровка подписи
« _____ »	_____	_____	« _____ »	_____	_____
Организация-получатель _____ (наименование) просп. Карла Маркса, 134, Самара (этаж 1, 2) (адрес, телефон, факс) ИНН _____ (банковские реквизиты) КЦМ _____			по ОКПО	755444	
Структурное подразделение _____ (наименование) Куйбышевский учебный центр профессиональных квалификаций Красноармейская ул., 139, Самара (адрес, телефон, факс) ИНН _____ (банковские реквизиты) КУЦК _____			БЕ	5855	
Организация-сдатчик _____ (наименование) Красноармейская ул., 139, Самара (адрес, телефон, факс) ИНН _____ (банковские реквизиты) КУЦК _____			по ОКПО	11111	
Структурное подразделение _____ (наименование)			БЕ	5955	

Рис. 1. Пример документа с шаблонами и полями для заполнения

Для извлечения данных из документов, с целью дальнейшей проверки считанного текста, используется библиотека tesseract. Данная библиотека используется для распознавания символов (OCR) с открытым исходным кодом, является самой популярной и качественной OCR-библиотекой.

OCR использует нейронные сети для поиска и распознавания текста на изображениях и базируется на модели LSTM. LSTM (Long short-term memory) — особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. Они прекрасно решают целый ряд разнообразных задач и в настоящее время широко используются. LSTM разработаны специально, чтобы избежать проблемы долговременной зависимости. Запоминание информации на долгие периоды времени — это их обычное поведение, а не что-то, чему они с трудом пытаются обучиться. Tesseract ищет шаблоны в пикселях, буквах, словах и предложениях, использует двухэтапный подход, называемый адаптивным распознаванием. Требуется один проход по данным для распознавания символов, затем второй проход, чтобы заполнить любые буквы, в которых он не был уверен, буквами, которые, скорее всего, соответствуют данному слову или контексту предложения.

Плюсами данной библиотеки можно отметить обученные языковые модели (свыше 192 языка), разные виды распознавания (изображение как слово, блок текста, вертикальный текст), легкая настройка (<http://bytepace.com/>).

Задача классификации заключается в определении принадлежности входных данных о документах для соотнесения их в определенную группу. Для каждого класса отбираются тестовые примеры, которые используются системой классификации в режиме обучения. После завершения обучения система с помощью специальных алгоритмов должна распределять входные потоки информации по классам. В данной работе было принято решение рассмотреть применение методов машинного обучения для задачи классификации документов. Методы машинного обучения обладают высокой скоростью и точностью классификации и относятся к группе методов обучения с учителем. Такой тип обучения подразумевает, что данные, подаваемые на входы системы, уже помечены, а важная часть признаков уже разделена на отдельные категории или классы. Поэтому сеть уже знает, какая часть входов важна, а какую часть можно самостоятельно проверить.

Процесс обучения модели — это подача данных для нейросети, которая в результате должна вывести определённые шаблоны для данных. В процессе обучения модели с учителем на вход подаются признаки и метки, а при прогнозировании на вход классификатора подаются только признаки. В задачах классификации каждому обучающему примеру сопоставлена метка класса. Если в задаче классификации число классов больше двух, то классификация называется многоклассовой. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества. Для классификации документов был выбран метод деревьев решений (Decision Trees, DT)

Метод деревьев решений для задачи классификации состоит в том, чтобы осуществлять процесс деления исходных данных на группы, пока не будут получены однородные множества. Совокупность правил, которые дают такое разбиение, позволяют затем определять наиболее вероятный номер класса для новых данных. Каждый узел дерева включает в себя условие ветвления по определенному признаку. У каждого узла количество ветвлений равняется количеству значений выбранного признака. Во время классификации происходит переход от одного узла к другому в соответствии с параметрами объекта. Процесс классификации будет идти до тех пор, пока не будет достигнут один из конечных узлов дерева. Значение этого узла позволит определить класс, которому принадлежит классифицируемый объект. Метод деревьев решений используется для классификации данных в самых разных областях и считается одним из самых популярных алгоритмов. Модель дерева решений на основе некоторых признаков классов документации представлена на рисунке 2.

В листьях разрешающего дерева размещаются значения целевой функции, в прочих узлах — условия перехода, определяющие направление движения вдоль ребер дерева. Для классификации каждого примера алгоритму необходимо пройти все дерево от корня до одного из листьев.

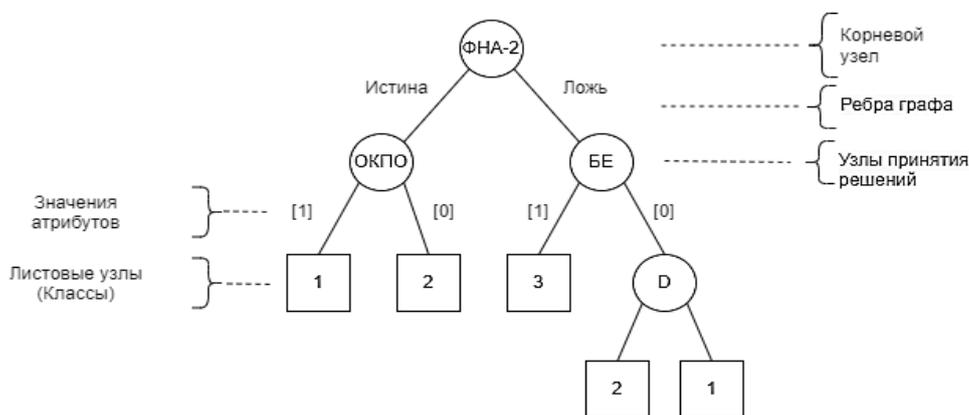


Рис. 2. Модель простого дерева решений

Пусть задано некоторое обучающее множество документов D , содержащее объекты в виде документов. Каждый документ характеризуется n атрибутами (признаками) $(D1 \dots Dn)$, причем один из них указывает на принадлежность объекта к определенному классу – A . Документы классифицируются по виду документов. Признак, который для каждого документа задает принадлежность к определенной группе документов, является предсказываемым атрибутом. Этот признак также входит в обучающую выборку. Через $A_k, (k = 1, \dots, m)$ обозначены классы видов документов, где $m = 11$.

Далее для построения дерева на каждом внутреннем узле необходимо найти такое условие (проверку), которое бы разбивало множество, ассоциированное с этим узлом на подмножества. Для того чтобы выбрать нужный атрибут, то он должен разделить множество таким образом, чтобы получившиеся подмножества состояли из S объектов, которые принадлежат одному классу или максимально приближены к данному классу. Выявленный вышеописанным подходом атрибут является более информативным среди всех остальных атрибутов.

В качестве информативного критерия, когда имеется n равновероятных значений признака документа, вероятность p каждого из них равна $1/n$ и информация, связанная со значением признака, то передаваемая информация вычисляется по формуле:

$$I(P) = - \sum_{i=1}^n p_i * \log p_i \quad (1)$$

Значение $I(P)$ дает оценку среднего количества информации, необходимого для определения класса примера из множества S .

Если множество S объектов разбито на попарно непересекающиеся классы A_1, A_2, \dots, A_m то информация, необходимая для того, чтобы установить класс примера, равна $Info(S) = I(P)$, где P – дискретное распределение вероятностей появления соответствующего примера при условии его принадлежности классу A_1, A_2, \dots, A_m . Каждая из оценок вероятностей p_i того, что случайно выбранный пример из множества S будет принадлежать к классу A_i , вычисляется как

$$p_i = \frac{|A_i|}{|S|}, \quad (2)$$

где $|A_i|, |S|$ – мощности как отдельных классов, так и всей обучающей выборки соответственно под мощностью класса понимается количество элементов класса. Разбив множество примеров на основе значений некоторого признака класса X (один из атрибутов) на подмножества S_1, S_2, \dots, S_e , мы можем вычислить $Info(S)$ как взвешенное среднее информации, необходимой для установления принадлежности примера определенному классу в каждом подмножестве:

$$Info(X, T) = \sum_{i=1}^e \frac{|S_i|}{|T|} * Info(S_i) \quad (5)$$

Величина

$$Gain(X, S) = Info(S) - Info(X, S) \quad (6)$$

показывает количество информации, которое мы получаем благодаря признаку X . Данная величина используется как критерий оценки информативности признака при построении решающих деревьев. Если в процессе работы алгоритма получен узел, в который не попал ни один пример, то он помечается как лист (терминальный узел), и в качестве решения для листа выбирается наиболее часто встречающийся класс у непосредственного предка данного листа. После этапа классификации документа выполняется его проверка по соответствующим особенностям класса.

В самом тексте документа содержатся выделенные для каждой группы ключевые слова, которые необходимо проверить. После данных слов в документе в обязательном порядке должны быть заполнены некоторые поля. При нахождении в тексте ключевого слова происходит проверка строки, стоящий за ключевым словом, и выполняется сопоставление данной строки со «строками-шаблонами». Шаблон состоит из требуемых слов для заполнения полей в документации.

В ходе исследования задачи автоматизации анализа документов отдела приема документов филиала ОАО «Российские железные дороги» был выполнен анализ ручной проверки и проверки при помощи автоматизированной системы (табл.).

Таблица

Анализ времени при ручной и автоматической проверке документа

Наименование документа	Время ручной проверки документа, с	Время автоматической проверки документа, с
ФНА-2	300	2,637

Таким образом, можно сделать вывод, что время автоматической проверки документа меньше, чем время ручной проверки.

© Муртазина Р.Р., 2022

КЛАССИФИКАЦИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Информационные технологии тесно связаны с жизнью современного общества. В понятие информационные технологии входят такие процессы как: поиск, обработка, сбор, хранение, передача и предоставление информации. Информационные технологии позволяют совершать глобальные операции над массивами данных. Это обеспечивает работоспособность сложных информационных систем. Информационная система включает в себя аппаратное и программное обеспечения. Под аппаратной частью подразумевается вычислительная система на основе электроники. Под программным обеспечением подразумевается блок команд, способных решать поставленную задачу. С увеличением числа производимой продукции все более актуальным становится вопрос о правильности функционирования приборов. От функциональности и удобства оборудования зависит его дальнейшее использование. Для достоверной оценки работоспособности изделий необходимо проводить тестирование программного обеспечения. Оно позволит сформировать конечное представление о данных, измеряемых прибором. Контроль качества и работоспособности электронных приборов - один из ключевых и наиболее важных этапов производства. Современное оборудование имеет сложную структуру и требует профессионального подхода при проверке. В настоящее время существует множество видов тестирования программного обеспечения. Каждый вид зависит от целей и задач тестирования. Данная обзорная статья предполагает систематизацию информации о видах тестирования программного обеспечения.



Рис. Классификация тестирования программного обеспечения

Классификация тестирования программного обеспечения достаточно сложна, ее различают по: степени автоматизации; этапам выполнения; масштабу проверяемых элементов; характеристикам качества; уровню доступа к данным; роли человека в выполнении тестирования. *Степень автоматизации тестирования* зависит от многих факторов. Ключевыми являются: квалификация специалистов; финансовые возможности;

Исходя из возможностей, реализуемы следующие виды тестирования: ручное (при данном тестировании не используются никакие программы. Все выполняется вручную по заранее продуманному алгоритму с предположительным результатом); автоматизированное (это тестирование предполагает использование специально написанных программ, которые способны выполнять проверку часто повторяющихся блоков программного обеспечения); полуавтоматизированное (совмещает в себе ручной и автоматизированный подход). При комбинации данных видов тестирования можно получить наилучший результат. По масштабу или уровню проверяемых элементов программное обеспечение разделяют на: модульное или компонентное (на модульном уровне тестирования проверяется корректная работа одного отдельного модуля внутри себя. тесты для такого тестирования обычно пишутся еще до написания основного кода программы [1, с. 107]). Для данного тестирования характерно разделение программы на отдельные функциональные модули или компоненты. Основной задачей этого метода тестирования является корректность работы отдельных частей кода: функции, метода, процедуры, модуля или объекта. Это самое первое и основное тестирование, которое чаще всего производится разработчиком. Модульное тестирование позволяет исправить ошибки на ранних этапах разработки.

Интеграционное. В процессе интеграционного тестирования происходит проверка: как взаимодействуют между собой различные компоненты системы после проведенного модульного тестирования, используя следующие подходы [2, с. 106].

Большого взрыва

Для использования данного подхода изначально предполагается, что созданные модули объединены между собой. Этот подход удобен только при тестировании систем, включающих в себя небольшое количество модулей.

Снизу вверх

Данный подход предполагает, что тестирование системы ведется от модулей нижнего уровня, до модулей верхнего уровня.

Сверху вниз

Этот подход является абсолютной противоположностью подходу «снизу вверх». Тестирование системы ведется от модулей верхнего уровня, до модулей нижнего уровня.

Смешанной интеграции

В данном подходе объединена логика сразу двух подходов «снизу вверх» и «сверху вниз». Модули высшего уровня тестируются отдельно, а модули нижнего уровня тестируются по принципу «снизу вверх».

Системное

Завершающий вид тестирования данного блока – системное. Оно проводится только после успешного завершения модульного и интеграционного тестирования. Цель: проверка соответствия системы в целом изначально заявленным характеристикам.

По характеристикам качества тестирование программного обеспечения разделяют на:

Функциональное

Осуществляется над компонентами, где компонент - это множество модулей, классов,

объединенных по общему признаку. В качестве примера работы сразу нескольких модулей можно рассмотреть API какого-либо Интернет-ресурса. Посылая определенные запросы к API сервиса, мы ожидаем получить определенный результат. Это может быть ответ из базы данных в формате JSON или что-то подобное. На основании ТЗ или задачи из предметной области можно протестировать работу того или иного компонента [3].

Функциональное тестирование включает в себя:

Тестирование совместимости

Заключение о работе готовой программы на корректное функционирование с другими программами.

Нефункциональное

Тестирование свойств, не относящихся к функциональности системы.

Тестирование производительности

Оценка быстродействия программного обеспечения при различных условиях. Данное тестирование разделяется на:

Стрессовое

Оценка длительности работы программного обеспечения при высоком уровне нагрузки.

Устойчивое

Оценка работы программного обеспечения при среднем уровне нагрузки.

Тестирование пользовательского интерфейса

Проверка удобства работы с интерфейсом. Данное тестирование является наиважнейшим для вывода продукции на продажу. Ясность в использовании и прибора его управлении играет важную роль при эксплуатации.

Тестирование конфиденциальности

При использовании конфиденциальной информации необходимо обеспечить ее безопасность. Для этого программа должна проверяться на различного типа уязвимости. Один из видов тестирования входных уязвимостей называется Fuzzing (фаззингом). Fuzzing – это практика предоставления случайного ввода в программное обеспечение, чтобы увидеть, как оно обрабатывает неожиданные данные [4].

По уровню доступа к данным тестирование программного обеспечения разделяют на три метода:

Метод черного ящика

Данный метод не дает тестирующему доступ к коду программного обеспечения. Тестируется реакция системы на различные внешние воздействия.

Преимуществами данного метода являются возможность исследовать сложные динамические системы без знания их структуры и устройства и возможность тестировать большое количество устройств без отдельной настройки каждого из них. Кроме того, для тестирования методом «черного ящика» необходимый уровень подготовки тестирующего минимален; тестирование не требует большого количества специальных знаний [5, с. 19].

Метод белого ящика

Данный метод предполагает полный доступ к внутренней структуре. Опираясь на

данные знания, проверяется соответствие ожидаемых выходных характеристик при известных входных.

Метод серого ящика

В данном методе комбинируются подходы к тестированию как методом черного ящика, так и белого. По роли человека в выполнении тестирования программного обеспечения разделяют на:

Независимое тестирование

Данное тестирование выполняется людьми не причастными к разработке и тестированию на предыдущих этапах программного обеспечения. Это условие необходимо для максимально независимой и объективной оценки качества функционирования конечного продукта.

Аттестационное тестирование

Тестирование, которое производится непосредственно заказчиком при приеме готового продукта. Необходимо для подтверждения работы основных функций.

Пользовательское тестирование

Существует две разновидности:

Альфа-тестирование. Данная разновидность тестируется непосредственно разработчиками. Они рассматривают наиболее вероятные пути использования системы пользователями. Это тестирование для поиска всех возможных ошибок и проблем во всем программном обеспечении. Этот вид тестирования выполняется на последнем этапе разработки приложения, перед запуском продукта или перед его отправкой клиенту, чтобы убедиться, что пользователь / клиент получает безошибочное программное приложение. Альфа-тестирование запускается до бета-тестирования, это означает, что после выполнения альфа-тестирования необходимо выполнить бета-тестирование. Альфа-тестирование не проводится в реальной среде. Тесты такого рода выполняются путем создания виртуальной среды, напоминающей реальную [6, с. 79].

Бета-тестирование. Проводится непосредственно после альфа-тестирования. Подразумевает тестирование системы ограниченным кругом пользователей. Это необходимо для выявления всех неточностей до перехода программного обеспечения в общественное пользование. Тестирование программного обеспечения - многоуровневый процесс, требующий индивидуального подхода. Разнообразие методов позволяет выбрать оптимальное решение. Тестирование системы дает возможность устранить недочеты на ранних стадиях, что ускорит разработку. Методы тестирования, рассмотренные в данной статье, формируют представление о некоторых типах тестирования.

Литература

1. Моисеев Д.А. Методология и процесс ручного тестирования // Надежность и качество сложных систем. 2017. № 3(19). С. 107-112. <https://doi.org/10.21685/2307-4205-2017-3-16>
2. Даутов Э.Ф., Хасьянов А.Ф., Шакирова А.И. Сокращение времени тестирования



программного обеспечения. // Современные наукоемкие технологии. 2019. № 7. С. 104-109.

3. Филиппов С.А., Таренко Л.Б. Виды и способы тестирования современного программного обеспечения // Общество, государство, личность: молодежное предпринимательство в поведенческой экономике в условиях цифровизации: Мат-лы XXI Международной научно-практической конференции студентов, магистрантов, аспирантов и молодых ученых (г. Казань, 29 апреля 2021 года). Казань, 2021. С. 232-237.

4. Дьяченко Н.В. Тестирование прикладного программного обеспечения // E-Scio. 2020. № 6(45). С. 605-609.

5. Чащина К.А. Тестирование устройств методом «черного ящика» // Прорывные научные исследования: проблемы, закономерности, перспективы: сб. статей XII Международной научно-практической конференции (г. Пенза, 27 января 2019 года). Пенза, 2019. С. 19-21.

6. Кириллов Д.С., Левинсон А.Д., Ямалеев Р.Р. Тестирование программного обеспечения // European Scientific Conference: сб. статей Международной научно-практической конференции (г. Пенза, 08 января 2020 года). Пенза, 2020. С. 77-79.

© Новикова Э.А., 2022

РАЗРАБОТКА ИНТЕРАКТИВНОГО WEB-ПРИЛОЖЕНИЯ «КОНСТРУКТОР САЙТА ПРЕПОДАВАТЕЛЯ»

Сайт – неотъемлемый атрибут современного преподавателя. С помощью web-сайта можно организовать самопрезентацию, взаимодействие с целевой аудиторией, хранение файлов и многое другое. Главное преимущество персонального файла заключается в том, что его содержимое доступно в любом месте, где есть подключение к сети Интернет. Преподавателю достаточно лишь знать доменное имя вместо того, чтобы пользоваться материальными носителями. Существует несколько способов создания сайта. Во-первых, web-страницу можно создать с помощью универсальных конструкторов сайтов. Во-вторых, верстку сайта можно заказать профессионалу. А в-третьих, сайт можно написать самому. У каждого из перечисленных способов есть недостатки. В случае с конструкторами сайтов необходимо изучать интерфейс, выбирать наиболее подходящий шаблон, проектировать макет. Стоит отметить, что домен у такого сайта будет длинным и сложным для восприятия. Создание web-страницы «на заказ» потребует внесения денежных средств. А чтобы написать сайт самому, необходимо знать несколько языков программирования и тонкости размещения странички в сети Интернет [2, с. 56].

Из-за сложности процесса создания персонального сайта немногие учителя могут похвастаться его наличием. Для того, чтобы решить данную проблему, проводится данное исследование. Цель исследования: создать узко специализированный конструктор сайтов для преподавателей, который при минимальных затратах ресурсов будет обеспечивать преподавателей качественным сайтом. Проект «Разработка интерактивного web-приложения «Конструктор сайта преподавателя» уже прошел несколько этапов разработки: постановка проблемы исследования, многоступенчатый анализ аналогов web-приложения и существующих в сети Интернет сайтов преподавателей, разработка информационной модели. В настоящее время осуществляется этап создания листингов и баз данных, обеспечивающих работу web-приложения.

«Конструктор сайта преподавателя» представляет собой динамический web-сайт. Технология динамического web-конструирования подразумевает сочетание PHP (язык препроцессинга, который позволяет в реальном времени изменять содержимое web-страницы в зависимости от действий пользователя) и MySQL (реляционная база данных). При работе код, написанный на языке PHP, обращается к элементам базы данных, встраивая в web-страницу, удаляя или изменяя их [3, с. 33].

Рассмотрим алгоритм работы разрабатываемого web-приложения. У пользователя (преподавателя) возникает необходимость в сайте. Он создает учетную запись в web-приложении, в результате чего ему присваивается уникальный идентификационный номер (id). Данный номер будет использоваться в базе данных.

Далее конструктор предлагает заполнить форму, цель которой – поместить в базу данных информацию, необходимую для размещения на заранее созданном макете сайта преподавателя. Форма состоит из множества вопросов которые затрагивают такие сферы жизни, как образование, карьера, достижения и другие. При этом вопросы сформулированы так, чтобы у пользователя создавалось впечатление, будто он проходит анкетирование, а не занимается версткой сайта. Ответы заносятся в поля базы данных, соответствующие идентификационному номеру [1, с. 22]. Во избежание потери информации из-за сбоев подключения сети Интернет или по причине отказа аппаратуры предусматривается функция автоматического сохранения заполнения полей формы.

Основная информация

Введите ваши Фамилию, Имя и Отчество полностью

Где вы преподаете?

учреждение среднего общего образования

учреждение среднего профессионального образования

учреждение высшего образования

Какую должность вы занимаете?

Какие дисциплины вы преподаете?

+ поле

Рис. 1. Фрагмент формы «Основная информация»

Сведения об образовании

Название учреждения

Годы обучения

Специальность (по диплому)

+ образование

Ваш портрет

Выберите файл Файл не выбран

Рис. 2. Фрагменты формы «Сведения об образовании» и «Ваш портрет»

Персональный сайт преподавателя может создаваться для разных целей. Поэтому при создании web-приложения будет предусмотрена возможность изменения пользователем структуры сайта.

Выберите разделы для сайта

Автобиография ?

Портфолио ?

Методическая копилка ?

Обратная связь ?

Фотогалерея ?

Нормативные документы ?

Рис. 3. Изменение структуры сайта (макет)

Прежде, чем отвечать на вопросы формы, создавая тем самым контент, пользователю будет предложена страница, фрагмент макета которой представлен на (рис. 1). В правой части находятся все возможные разделы будущего сайта. Пользователь может «ухватить» необходимый элемент и поместить его на шаблон в левой части (рис. 2). Таким образом реализуется возможность гибкой настройки структуры сайта, что невозможно при использовании большинства универсальных конструкторов.

После того, как структурные элементы сайта выбраны, и форма заполнена, вся необходимая информация оказывается в базе данных. На основе этого начинается конструирование персонального сайта преподавателя. При создании макетов сайта преследовалась цель сделать их минималистичными, удобными, интуитивно-понятными и привлекательными. Большинство существующих сайтов преподавателей выглядят отталкивающими и нагроможденными, в них сложно ориентироваться. Поэтому задача конструктора – исправить этот недостаток. Главная страница – место, где размещена основная информация: место работы, должность, оконченное учебное заведение и фото (рис. 4).



Рис. 4. Главная страница сайта преподавателя (макет)

В верхнем правом углу находится кнопка «Меню». При нажатии на нее открывается страница, на которой размещены элементы, соответствующие структуре сайта, заданной пользователем на этапе формирования (рис. 3). Каждый элемент представляет собой кнопку, при нажатии на которую осуществляется переход на соответствующую страницу (рис. 5).

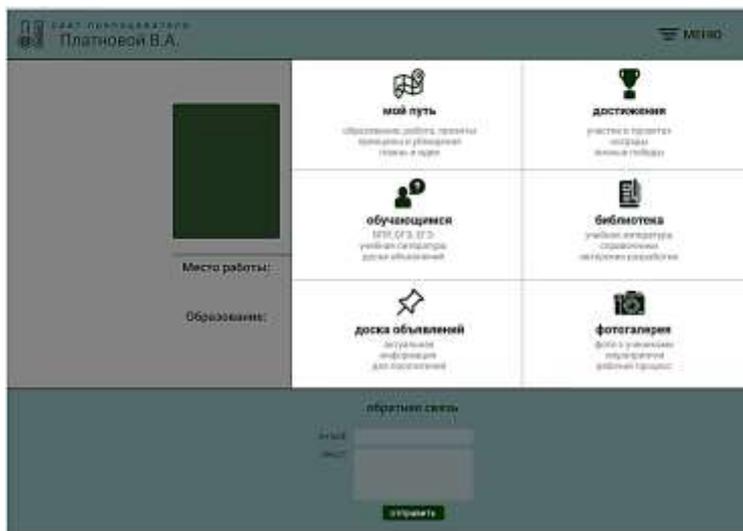


Рис. 5. Меню (макет)

Рассмотрим другие страницы web-сайта преподавателя. Страница «Мой путь» предназначена для того, чтобы пользователь мог в хронологической последовательности рассказать о своем профессиональном пути. Предусмотрена возможность указать дату события и прикрепить соответствующее изображение. При нажатии на определенное событие открывается всплывающее окно с более подробной информацией и возможности увеличения изображения (рис. 6, 7).

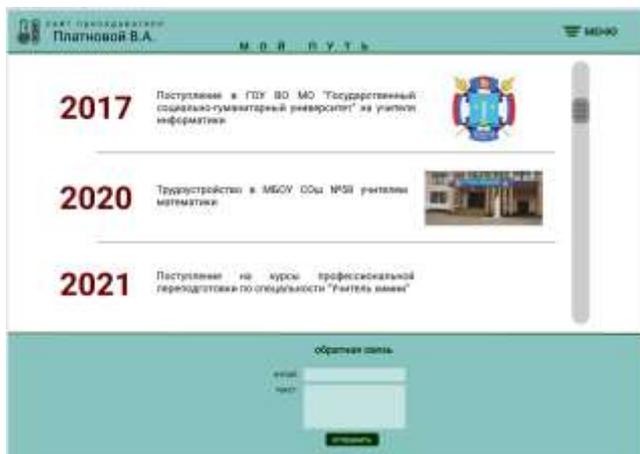


Рис. 6. Web-страница «Мой путь»



Рис. 7. Web-страница «Всплывающее окно»

Страница «Достижения» конструируется по аналогии. Многие преподаватели используют сайт в качестве удобного и мобильного хранилища файлов. Поэтому к задачам конструктора также относится организация быстрого доступа к тексту и медиа. Предполагается, что файлы будут разбиты по разделам. Каждый блок занимает условную строку. Внутри строки файлы представлены не списком (характерно для большинства сайтов), а в виде галереи, которую можно листать с помощью кнопок «вправо», «влево». Файл, расположенный в центре строки, находится в «фокусе». Все разделы, предполагающие размещение файлов, организованы аналогичным образом (рис. 8).



Рис. 8. Web-страница «Библиотека»

На данный момент составлены макеты web-страниц, ведется создание листингов и базы данных, а также налаживание взаимодействия между этими неотъемлемыми частями процесса конструирования динамических сайтов. Тестирование и отладка – следующий этап работы, который будет проводиться с привлечением преподавателей школы и ВУЗа.

В результате среди конструкторов сайтов появится узкоспециализированный, созданный специально для преподавателей сервис, который будет способствовать популяризации персонального сайта и информатизации образования.

Литература

1. Еманаева А.А., Садовина Н.В. Использование СУБД MYSQL в сайтостроении // Информация как двигатель научного прогресса: сб. статей международной научно-практической конференции (г. Екатеринбург, 15 апреля 2017 года). Екатеринбург, 2017. С. 22-24.
2. Михайлов А.А. Инновационные способы создания веб-сайтов // Colloquium-Journal. 2018. № 8-2(19). С. 56-57.
3. Робин Н. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. СПб.: Питер, 2016. 816 с.

© Платонова В.А., 2022

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ РАБОТЫ ТЕХНИЧЕСКОЙ СЛУЖБЫ

Техническая поддержка – это служба, в которую пользователи продукта или услуги могут обратиться за оказанием технической поддержки по решению возникшей проблемы, а также за получением дополнительной информации по интересующему вопросу. Во времена, когда жизнь без компьютера не возможна, служба технической поддержки крайне необходима. Техническая поддержка нужна не только чтоб узнать ответы на часто задаваемые вопросы, но и, если необходима помощь с компьютерной техникой (сломался компьютер, принтер и т.д.) или не работает программное обеспечение (ПО).

К основным задачам технической поддержки по работе с клиентами относятся: знание стандартных решений и ответов на наиболее часто задаваемые вопросы пользователей, ответ на обращения пользователей, анализ проблем и разработка рекомендации по их устранению, оказание помощи в решении технических проблем при использовании продукта, оказание помощи по восстановлению работоспособности программного продукта после сбоев, а также ведение журнала с описанием выполненных действий и принятых решений. Наиболее часто за оказанием технической поддержки можно обратиться через чат, электронную почту или форму «обратной связи». В случае, обращения пользователей, специалисты из технической поддержки обязаны знать нюансы проблемы, найти и решить данную проблему. Обработка обращений пользователей происходит с помощью программного обеспечения.

Актуальность технической поддержки заключается в том, что программа автоматизирует работу технической службы, тем самым значительно уменьшая трудоемкость принятия заявок, время на отслеживание проблем с техникой и программным обеспечением. Со стороны пользователя информационная система исключает поиск специалиста по ремонту и обслуживанию технических проблем и программного обеспечения. Со стороны технической поддержки информационная система увеличивает прием заявок, что дает возможность узнавать о проблемах в любое время, к тому же давать обратную связь.

Актуальным является доступность и быстрота, благодаря которым пользователь сможет быстро сформировать заявку, а специалист ее увидеть и приступить к работе. Так же пользователь сможет на доступном языке найти ответы на часто задаваемые вопросы, тем самым самостоятельно решить проблему без вызова специалиста. Целями создания автоматизированной информационной системы (АИС) поддержки работы технической службы являются:

1. Ускорение получения заявок специалистом за счет создания системы по приему заявок.

2. Обработка заявок специалистом за счет создания системы хранения и обработки заявок.

3. Отслеживание пользователем исполнения заявок за счет получения письма о выполнении работы.

АИС поддержки работы технической службы предназначена для решения перечисленных ниже задач: формирование заявки; получение ответа от службы поддержки о принятии заявки; получение служебной записки на основании заявки; отслеживание исполнения заявки пользователем; информирование пользователей; хранение заявок.

К виду автоматизируемой деятельности относятся: процессы принятия обращения пользователей в тех поддержку и отслеживание исполнения заявки пользователем; процессы помощи пользователю получить ответы на стандартные вопросы, в случае проблемы с техническим средством или ПО; процессы хранения заявок пользователей.

В основу создания АИС положены следующие принципы: многопользовательский режим работы; информирование пользователей о наступивших и ожидаемых событиях, требующих их внимания и контроля; поиск и получение информации.

Для создания поддержки технической службы поддержки будем использовать бесплатный конструктор сайтов – Wix (<https://clck.ru/CCRiL>).

Порядок подключения к системе состоит из следующих шагов:

1. Запустить веб-обозреватель Google Chrome или другой любым удобным способом: либо с помощью ярлыка на рабочем столе, либо с помощью ярлыка в меню «Пуск».
2. В адресной строке указать адрес информационной системы <https://helpdeskspu.wixsite.com/iitimoi>.
3. После правильного указания адреса на экране откроется главная страница сайта (рис. 1).

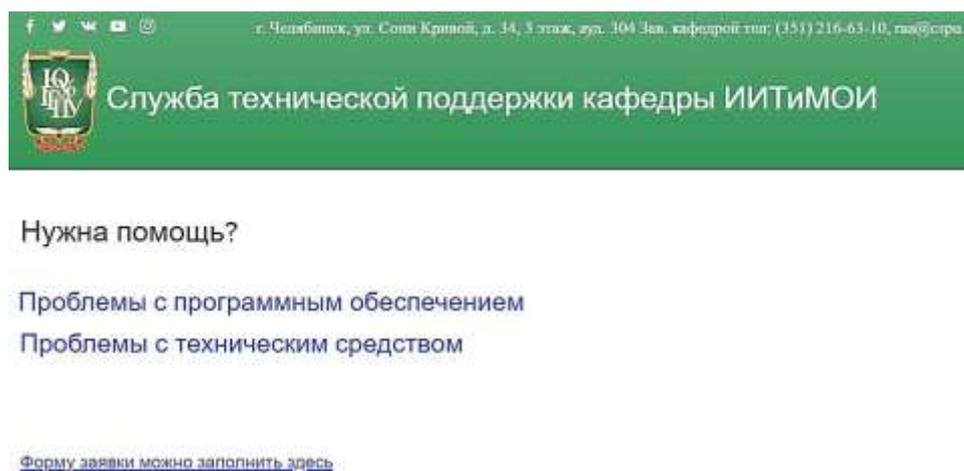


Рис. 1. Главная страница

Пользователь имеет доступные пункты меню: Главная страница; Страница «Проблемы с программным обеспечением»; Страница «Проблемы с техническим средством»; Страницы с ответами на вопросы; Страница отправки формы заявки.

Доступные операции: находясь на любой странице сайта, пользователь может переходить по ссылкам в шапке сайта на страницы социальных сетей ЮУрГГПУ; находясь на главной странице сайта, пользователь может перейти на страницы «Проблемы с программным обеспечением» и «Проблемы с техническим средством»; находясь на любой странице сайта, кроме страницы заполнения формы заявки, пользователь может перейти на страницу формы заявки; находясь на страницах с вопросами проблем с ПО или ТС, пользователь может переходить на страницы ответов на вопросы; находясь на странице формы заявки, пользователь может заполнить форму заявки и отправить ее. На страницах «Проблемы с программным обеспечением» (рис. 2) и «Проблемы с техническим средством» (рис. 3) указаны часто задаваемые вопросы, нажав на которые откроются страницы с ответами. Так же перейдя по ссылке можно попасть на страницу для заполнения формы заявки.

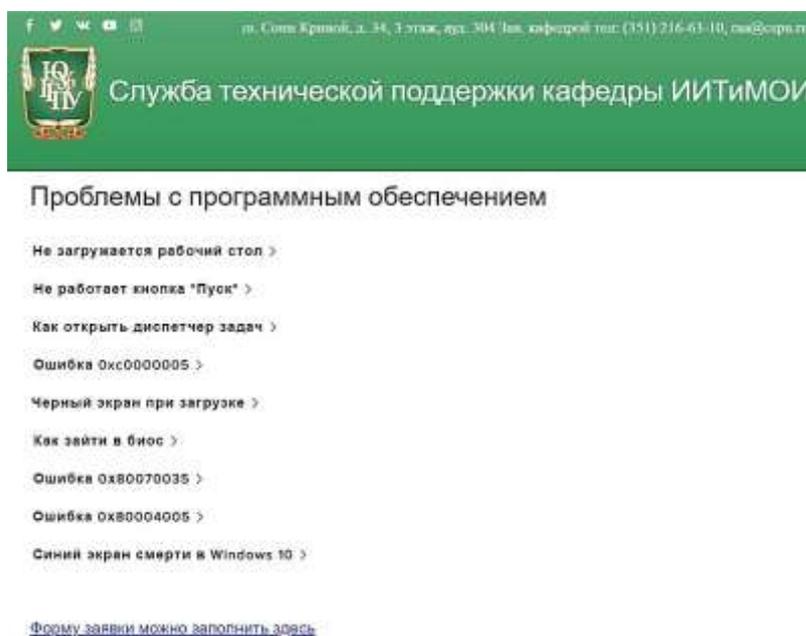


Рис. 2. Страница «Проблемы с программным обеспечением»

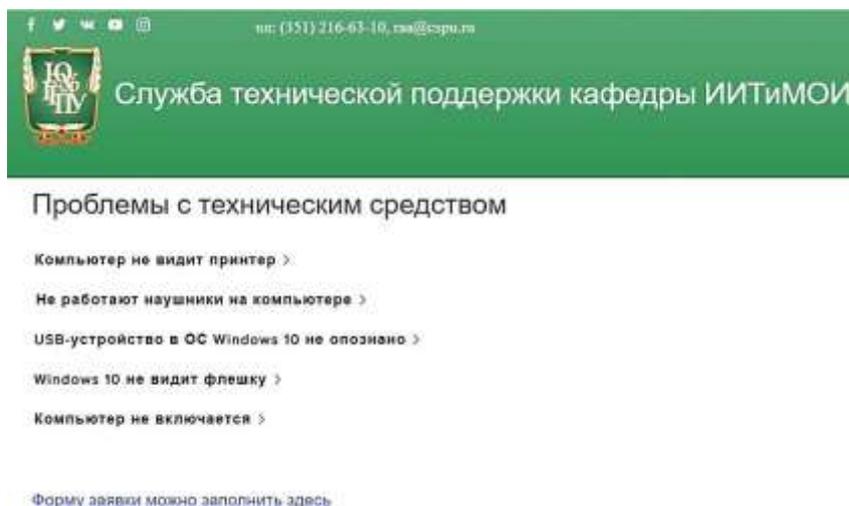


Рис. 3. Страница «Проблемы с техническим средством»

Для того чтобы отправить форму заявки, необходимо заполнить заявку. Все поля формы должны быть правильно заполнены. Когда форма заявки заполнена нужно нажать кнопку «Отправить», тогда заявка дойдет до специалиста (рис. 4).

fb vk odn @

бизнес, ул. Соци.Кривой, д. 34, 3 этаж, ауд. 304 Зам. кафедрой тел: (351) 216-63-10, zam@iitimo.ru

Служба технической поддержки кафедры ИИТиМОИ

Главная

Оставьте заявку

Введите ваше имя

Эл. почты

Выберите кабинет

Выберите № компьютера

Выберите вариант

Программное обеспечение

Техническое средство

Введите текст

Я ознакомился с Политикой конфиденциальности персональных данных и даю согласие на обработку персональных данных

Отправить запрос

Рис. 4. Страница формы заявки

В форме заявки необходимо указывать свой e-mail для того, чтобы в дальнейшем от технической службы получить уведомление о выполнении заявки (рис. 5).

Новое сообщение с сайта HelpDeskIITiMOI

- HelpDeskIITiMOI Сегодня, 15:02
Кому: вам

Ваша заявка исполнена!
Спасибо за заявку

Рис. 5. Уведомление от технической службы

При отсутствии интернета на компьютере или другом техническом устройстве (телефон, планшет, ноутбук) работа с программным средством не возможна.

При других аварийных ситуациях, связанных с работой сайта, обратиться к Системному Администратору.

Администратор имеет доступные пункты меню такие же, как и пользователь, помимо этого у него есть панель управления сайтом (входящие, контакты, ответы форм, задачи).

Администратору доступны те же операции что и пользователю, помимо этого: находясь на панели управления сайтом, администратор может просматривать таблицу форм заявок; находясь на панели управления сайтом, администратор может просматривать контакты и входящие сообщения, так же может отвечать на сообщения; находясь на панели управления сайтом, администратор может просматривать задачи, сформированные на основании заявок, и помечать задачи как выполненные; на почту администратор получает служебную записку, которую может просматривать и с этой же почты написать письмо отправителю заявки.

На панели управления сайтом на вкладке «Управление клиентами» – «Ответы форм» находится вся таблица присылаемых заявок (рис. 6). В данной таблице хранятся все отправленные заявки от пользователей со всеми данными, указанными в заявке. Данную таблицу можно экспортировать и импортировать, нажав в правом верхнем углу «Импорт/экспорт».

	Дата отправки	Введите ваше имя	Эл. почта	Выберите кабинет	Выберите № компьютер...	Выберите вариант Э
1	05/07/2021 22:20	Анастасия	guzko.1411@mail.ru	303	№5	Техническое средство
2	05/07/2021 09:25	анастасия	guzko.1411@mail.ru	313	№6	Техническое средство
3	05/06/2021 23:15	Анастасия	guzko.1411@mail.ru	313	№8	Техническое средство
4	05/06/2021 22:27	Летая	60080405434@mail.ru	318	№10	Техническое средство
5	05/04/2021 22:31	Анастасия	guzko.1411@mail.ru	314	№8	Техническое средство
6	05/04/2021 22:30	Алина	guzko.1411@mail.ru	318	№11	Техническое средство
7	05/04/2021 22:07	анастасия	guzko.1411@mail.ru	315	№7	Программное обеспечение
+						

Рис. 6. Таблица форм заявок

Когда пользователь отправил заявку, формируется контакт на основе его имени и e-mail (рис. 7). В самом контакте указана информация о пользователе такая как: имя, e-mail, все поля формы, которые он заполнил. Контакт можно удалить, отправить сообщение, добавить задачу, а также просмотреть уже созданные задачи и просмотреть журнал действий.

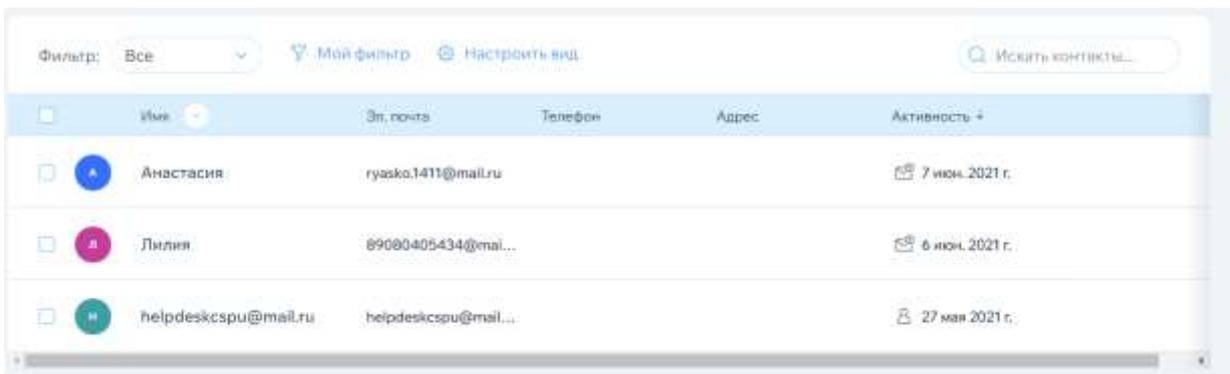


Рис. 7. Сформированные контакты пользователей

Для информирования пользователя с ним необходимо вести переписку, в этом помогут «Входящие» (рис. 8). Во входящих создается переписка с контактом, где так же отображена вся форма заявки. На почту, которую пользователь указал при заполнении заявки, можно отправлять сообщения, помимо отправлять вложения.

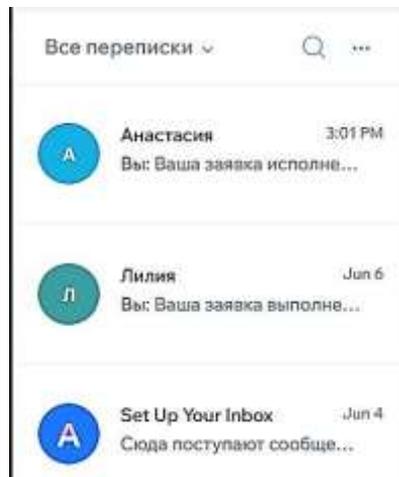


Рис. 8. Входящие сообщения

После того как пользователь отправляет форму заявки, формируется задача на основе заявки (рис. 9). На выполнение заявки дается 7 дней и для каждой задачи подписано от кого она.

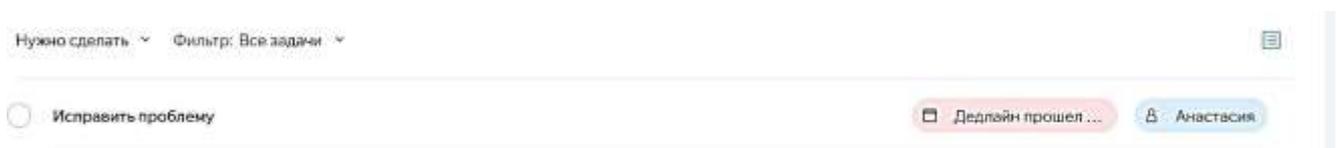


Рис. 9. Задача на основании заявки

Каждый раз как пользователь отправляет заявку, на почту приходит сообщение в виде служебной записки, на основании заявки (рис. 10).

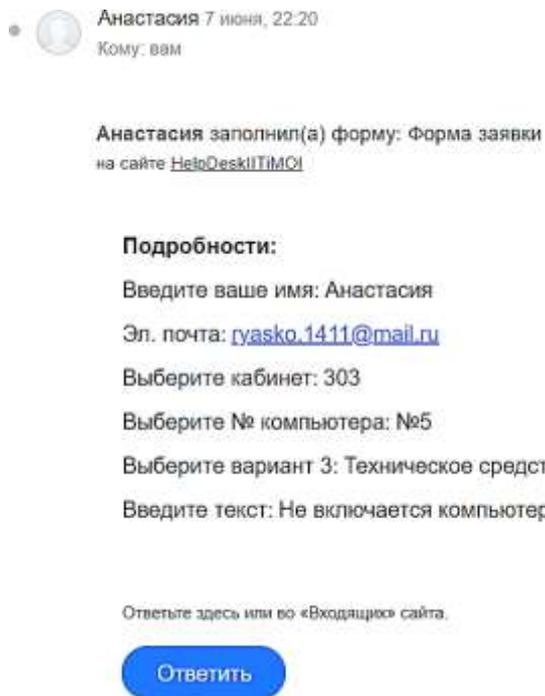


Рис. 10. Служебная записка

Создана информационная система поддержки работы технической службы. Главная задача системы – помогать пользователям оставлять заявку на сайте, а специалистам получать заявку, обрабатывать ее и хранить в базе данных. ИС облегчает работу специалистов путем быстрого сбора заявок и хранения необходимых сведений. Данная система позволила осуществить все поставленные цели.

© Попова А.С., 2022

ДИЗАЙН ЛИТЕРАЛОВ КОЛЛЕКЦИЙ ДЛЯ ЯЗЫКА KOTLIN

Литералы коллекций являются лаконичным, понятным, и, главное, привычным для программистов способом задания коллекций. Их лаконичность весьма полезна в области анализа данных, кроме того, наличие отдельной конвенции в языке позволит компилятору дополнительно оптимизировать использование литералов коллекций. Принятый в языке дизайн литералов должен давать пользователю возможности для упрощения кода при использовании литералов и для оптимизации создания различных коллекций. В языке Kotlin в настоящее время есть ограниченные литералы коллекций, которые доступны только в аннотациях и всегда имеют тип *Array*. Целью данной работы является дизайн полноценных литералов коллекций для языка Kotlin, а именно — дизайн синтаксиса и конвенции, которые соответствуют философии Kotlin.

Сначала были рассмотрены аналоги. В сообществе уже предпринимались попытки предложить дизайн литералов коллекций для Kotlin. Самым полным и широко обсуждаемым является предложение дизайна из КЕЕР (Kotlin Evolution and Enhancement Process) (<https://clck.ru/dYUtu>). Далее мы не будем рассматривать ограниченные литералы коллекций, которые присутствуют в Kotlin на данный момент, так как они не доступны для использования в произвольном коде и являются эквивалентом (синтаксическим сахаром) для конструкции *arrayOf()* в аннотациях.

КЕЕР-литералы (дизайн литералов, предложенный в КЕЕР) вводят новый оператор, на вызов которого заменяется литерал коллекций в коде. Выбор оператора предлагается выполнять в зависимости от типа аргументов литерала и типа ожидаемого результата, а для передачи аргументов в коллекцию используется механизм *vararg*.

Важно отметить, что КЕЕР-литералы позволяют пользователям определять свои типы для литералов коллекций, через тип результата оператора. Но при этом этот дизайн не является эффективным с точки зрения производительности из-за того, что для передачи аргументов задействуется *vararg*, который является неэффективной конструкцией. Поэтому необходимо разработать новый дизайн, учитывающий следующие требования:

- Тип литерала должен определяться ожидаемым типом, как это происходит, например, для лямбд;
- Должна быть возможность определить произвольный пользовательский тип для литерала коллекции;
- У литерала должен быть стандартный тип в случае, когда нам не дано никаких подсказок к возможному типу;
- Литералы коллекций не должны использовать механизм *vararg*

Результаты дизайна литералов описаны в тезисах «Дизайн литералов коллекций для языка Kotlin» [1, с. 115]. Здесь же подробнее разберем, почему были приняты данные

решения. Сначала был проведен обзор литералов в других языках, далее был продуман дизайн синтаксиса и соответствующей конвенции.

Для анализа литералов были рассмотрены следующие языки программирования: Groovy, Python, C#, F# и так далее (рис. 1). Во многих языках присутствуют литералы двух видов: для последовательностей и для словарей. В Kotlin также было решено использовать два вида литералов: для последовательностей (list-литералы) и для списка пар ключ-значение (map-литералы).

```
• Groovy:  
def list = [5, 6, 7, 8]  
def map = [name: 'Gromit', likes: 'cheese', id: 1234]  
• Python:  
a = [66.25, 333, 333, 1, 1234.5]  
d = {'short': 'dict', 'long': 'dictionary'}  
• C#:  
var digits = new List<int> { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };  
var dic = new Dictionary<int, int> { { 1, 2 }, { 3, 4 } };  
• F#:  
let aList = [ 1; 2; 3; 4 ] // list literal  
let aSet = set [ 1; 2; 3; 4 ] // set( list )
```

Рис. 1. Примеры литералов коллекций

Синтаксис литералов состоит из двух основных частей: символы для задания границ литерала и символы для разделения пар ключ-значение в map-литералах. В рассмотренных языках для литералов коллекций используются скобки двух видов «[]» и «{}». Было решено не изменять синтаксис для литералов последовательностей и оставить квадратные скобки, так как в Kotlin уже присутствует синтаксис для них. Для литералов словарей чаще всего используются фигурные скобки, но для map-литералов было решено также использовать квадратные скобки. Такое решение было принято, так как у фигурных скобок в Kotlin уже есть сложившаяся семантика (фигурные скобки внутри блока кода используются для задания лямбд) и добавление второго смысла фигурным скобкам создало бы сильную путаницу для пользователя, например, при чтении кода. Пары ключ-значение было решено разделять с использованием двоеточия «:». Альтернативным вариантом было использование символа «=>», как например при использовании именованных параметров функций $foo(a = 1, b = 2)$. От этого варианта было решено отказаться. Основная причина такая же, как и у фигурных скобок — у знака «=>» есть сложившаяся семантика. А именно это присваиванием имени значения, когда в случае литерала коллекции это было бы связывание значения со значением.

Одним из требований к дизайну является возможность явно указать тип литерала. Было решено специфицировать тип, задавая его перед литералом, например, $Set<Int> [1, 2, 3]$ (рис. 2). Стоит отметить, что данное решение создает конфликт между оператором доступа по индексу и литералом коллекции со специфицированным типом. Например, если объявить оператор *get/set* на компаньоне класса *Set*, то выражение $Set<Int> [1]$ может быть как вызовом оператора *get*, так и литералом коллекции из одного элемента. Данная проблема решается двухэтапным резолвом, который в случае двусмысленности предпочитает *get/set* (для сохранения обратной совместимости).

List literal	Map literal
• []	• [:]
• [1, 2, 3]	• ["a" : 1, "b" : 2, "c" : 3]
• Set<Int> [1, 2, 3]	• Map<String, Int> ["a" : 1, "b" : 2, "c" : 3]

Рис. 2. Синтаксис литералов коллекции

Еще одним требованием является то, что у литерала должен быть стандартный тип в случае, если не получается его (тип) вывести, как в выражении `val a = [1, 2, 3]`. Стандартными типами было решено выбрать *List* для list-литералов и *Map* для map-литералов. В случае если у литерала есть аргументы, то типовой параметр выводится на их основе. Если литерал пустой, то сначала типовой параметр выводится на основе ожидаемого типа; если информации недостаточно, то тип аргумента приравнивается к *Nothing*.

Литералы коллекций, как и многие другие конструкции языка Kotlin, было решено реализовывать как синтаксическую конвенцию. Семантика литерала коллекций будет определяться через развертывание синтаксиса (<https://clck.ru/eszeX>). Аналогичным образом в Kotlin работают такие конструкции как цикл *for*, арифметические операции и делегированные свойства. Дизайн конвенции для литералов коллекций пережил несколько итераций. Самым первым вариантом была конвенция, некоторое время назад предложенная командой языка Kotlin. На объекте коллекции объявляется оператор *add* с одним аргументом для list-литералов и с двумя аргументами для map-литералов. На компаньоне соответствующего типа объявляется оператор *build*. В таком случае литерал вида `Set<Int> [1, 2, 3]` можно развернуть в `Set.build<Int>(3) { add(1); add(2); add(3); }` (рис. 3). В данном варианте конвенция является весьма негибкой и позволяет работать только с изменяемыми объектами.

```
class MyList<T> {
    operator fun add(element: T);

    companion object {
        operator fun build(size: Int, MyList<T>.() -> Unit = {});
    }
}

val a = MyList<Int> [1, 2]  →  val a = MyList.build(3) {
                               add(1)
                               add(2)
}
```

Рис. 3. Первый вариант конвенции

Второй вариант конвенции было решено сделать более гибким, за счет вынесения логики построения коллекции из ее типа в отдельный интерфейс. Такой интерфейс должен обладать следующими функциями:

- `init(size: Int)` — инициализирует создание коллекции и задает ее размер (что может позволить провести необходимые оптимизации);

- `add(element: E)/add(key: K, value: V)` — добавляет элемент в создаваемую коллекцию;
- `build(): T` — возвращает итоговую коллекцию.

На типе коллекции необходимо реализовать оператор `buildList(size: Int, conf: ListCollectionLiteralBuilder<T, E>().() -> Unit): T` для list-литералов или `buildMap(size: Int, conf: MapCollectionLiteralBuilder<T, K, V>().() -> Unit)` для map-литералов, в вызов которого и будет развернут литерал (рис. 4).

```
interface ListCollectionLiteralBuilder<T, E> {
    operator fun init(size: Int) { ... } // how to initialize a collection
    operator fun add(element: E) { ... } // how to add a new element
    operator fun build(): T { ... } // how to get the final collection
}

operator fun <T> buildList(
    size: Int,
    conf: ListCollectionLiteralBuilder<SomeClass, T>().() -> Unit = {}
): SomeClass {
    return LCLBImpl<SomeClass, T>().apply {
        init(size)
        conf()
    }.build()
}

val a = SomeClass [1, 2, 3] → val a: SomeClass = SomeClass.buildList<Int>(3) {
    add(1)
    add(2)
    add(3)
}
```

Рис. 4 Конвенция с использованием специального интерфейса

Данный вариант дает большую гибкость для пользователя, но после анализа этой конвенции было выявлено, что большая часть операторов `buildList/buildMap` будет выглядеть одинаково. Для избавления пользователя от написания одного и того же кода был предложен третий вариант конвенции. В третьем варианте работа по реализации оператора `buildList/buildMap` была переложена на компилятор, то есть оператор генерировался синтетически на основе интерфейса. Однако позже было отмечено, что такие неявные синтетические конструкции не являются идиоматичными для Kotlin. В итоге было решено, что, хотя реализация оператора `buildList/buildMap` и является определенным boilerplate-кодом, но её не нужно писать большое количество раз (литералы коллекций намного чаще используются, чем реализуются). Поэтому было решено вернуться ко второму варианту и оставить его как финальный.

В итоге был разработан дизайн литералов коллекций, удовлетворяющий всем поставленным требованиям. Полученный дизайн действительно позволяет пользователям самим контролировать процесс создания коллекции. Также конвенция не использует механизм `vararg`, что позволяет избавиться от создания временного объекта и получить более эффективное инициализирование коллекции. В будущем планируется продолжить развивать данный дизайн и параллельно вести его реализацию в компиляторе Kotlin.

Литература

1. Райш. А.В., Ахин М.Х., Беляев М.А. Дизайн литералов коллекций для языка Kotlin // Наука. Технологии. Инновации: сб. научных трудов. Новосибирск, 2021. Часть 1. С. 115-118.

© Райш А.В., 2022

БЕЗОПАСНОСТЬ И КОНФИДЕНЦИАЛЬНОСТЬ FACEAPP И ТИКТОК

Приложения для смартфонов могут вызывать привыкание у пользователей из-за их уникальности, простоты использования, актуальности и растущей популярности. Добавление искусственного интеллекта в их функциональность быстро завоевало популярность у пользователей. Возникает стремительный рост сферы разработки мобильных приложений, которые облегчают жизнь [1, с. 400]. На протяжении многих лет некоторые приложения для смартфонов быстро завоевывали огромную популярность, такие как FaceApp и TikTok. FaceApp может похвастаться использованием искусственного интеллекта для преобразования фотографий человеческих лиц, используя свои мощные возможности распознавания. FaceApp стал популярным из-за бурной реакции пользователей, которая привела к тому, что рынок ряда других подобных, но менее известных клонов-приложений занял первые места в магазинах Google Play Market и AppStore. TikTok предлагает редактирование видео и обмен короткими видеоклипами, делая их очаровательными, забавными, вызывающими привыкание у молодого поколения. Видео могут попасть в рекомендации другим пользователям, тем самым абсолютно каждый может набрать достаточно большую аудиторию, которая будет реагировать на его творчество. Видеоролики могут быть сняты на абсолютно любую тематику: юмор, политика, искусство, образование, программирование и т.д. FaceApp и TikTok стали мишенями средств массовой информации, правительств из-за опасений по поводу конфиденциальности, неправильного использования данных, антикриминалистики и безопасности. В этой статье предоставлен анализ приложения FaceApp и TikTok на предмет владения, управления данными, обеспечения конфиденциальности, использования стеганографии и общей безопасности.

Глобальный рост использования мобильных данных на протяжении многих лет имел впечатляющую траекторию из-за требований пользователей. Годовой отчет Ассоциации индустрии сотовой связи (CTIA) за 2019 год (<https://clck.ru/at8dS>) продемонстрировал показатели роста с 2017 по 2018 год с беспрецедентным увеличением на 82%. В отчете также показано, что количество используемых смартфонов в США выросло с 273,2 млн до 284,7 млн, что на 4% больше. Исследовательский центр Интернета и технологий сообщает, что по состоянию на июнь 2019 года 81% всех американцев владели смартфонами (<https://clck.ru/VYdsE>). Каждая разновидность мобильного телефона требует определенного набора инструментов и методов сбора доказательств. Кроме того, у всех них есть различные мобильные приложения, которые собирают важные данные, которые могут быть использованы судебными экспертами по телефону. По мере усложнения смартфонов и их приложений должны расти и инструменты, используемые для извлечения из них доказательных данных.

Электронная или цифровая криминалистика может быть определена как процесс идентификации, наблюдения, сохранения и анализа цифровых доказательств путем соблюдения стандартных операционных процедур (<https://clck.ru/at8rD>). С другой стороны, анти-криминалистика очень похожа на сокрытие данных, скрывая доказательства или сообщения в устройствах, чтобы помешать обнаружению и незаметности [2, с. 1392-1403]. Одним из способов сокрытия потенциальными преступниками информации или улик является использование техники, называемой стеганографией. Эта практика скрывает секретную информацию внутри чего-то другого. Например, скрывание секретного файла или сообщения внутри цифровой фотографии, в результате чего измененная фотография будет выглядеть почти так же, как оригинал. Несмотря на то, что существует множество разновидностей цифровых криминалистических инструментов, сохраняются ограничения на успешное обнаружение скрытых данных с помощью стеганографических методов. Существует множество свободно доступных инструментов, помогающих новичкам и злоумышленникам в использовании методов стенографии. Такие инструменты могут быть использованы для сокрытия информации о мультимедийных данных и передачи ее через смартфоны. Часто текстовые сообщения, голосовые сообщения и приложения для социальных сетей на смартфонах выступают в качестве средства такой передачи. При разработке руководств и рекомендаций по судебно-медицинской экспертизе для смартфонов или мобильных устройств правоохранительные органы должны проявлять большую осторожность, когда речь заходит о стеганографии на мобильных устройствах [3, с. 1-2].

Самые популярные на сегодняшний день смартфоны работают с операционными системами Apple iOS или Android, все из которых требуют взаимодействия с клиентами через приложения. Извлечение доказательств из приложения для смартфонов Android требует обширных знаний и опыта работы с мобильными операционными системами и их файловыми структурами. «Android использует пакеты, известные как Android Package Kit (APK), для размещения кода в приложении, которое также управляет этим приложением. Файл APK - это пакет всех частей определенной программы. Он содержит программный код, ресурсы, ресурсы, сертификат и файл манифеста. В общем, существует заранее определенная структура файлов для программного кода и его ресурсов, организованных в несколько папок, среди которых src/, gen/, libs/, res/, и assets/».

В последнее время появились приложения для смартфонов, работающие на базе искусственного интеллекта. Искусственный интеллект не просто повторяет одно и то же; он умен и может делать больше, приспосабливаясь к ситуациям. Хотя искусственный интеллект управляется алгоритмами, он более «похож на человека», чем традиционное машинное обучение. Операционная система Android от Google использует искусственный интеллект для улучшения приложений, таких как Google Assistant, который обеспечивает двустороннюю связь. Приложения на базе искусственного интеллекта на смартфонах часто изучают использование устройств пользователей, например, выбор чтения новостей, поведение и интересы, чтобы изучить и точно настроить свои алгоритмы. Однако существуют проблемы и

проблемы с конфиденциальностью (<https://clck.ru/at9Db>) связанные с тем, что искусственный интеллект в наши дни является технологическим бредом.

Например, такие приложения, как FaceApp, которые используют искусственный интеллект для обработки изображения вашего лица (редактирование фотографий, изменение вашего лица на противоположный пол, изменение черт лица или фотопшоп), связаны с проблемами конфиденциальности и безопасности в отношении настроек разрешений приложений, места хранения данных (на серверах, расположенных в другой стране или облаке) и часто безотзывных прав компании на пользовательские данные, которые указаны где-то в их пользовательских соглашениях. Приложения на базе искусственного интеллекта, хотя и полезны, возможно, слишком умны для нашего же блага.

Популярные приложения, такие как FaceApp и TikTok, пользуются популярностью во всем мире. TikTok и FaceApp были загружены миллионами пользователей, в магазинах Android и Apple iOS. FaceApp предлагает преобразования изображений, популярность которых растет, а TikTok предлагает делиться короткими видеороликами. Большая часть западной пользовательской базы TikTok изначально принадлежала Musical.ly, в то время как его восточная аудитория все еще использует отдельную версию приложения в Китае под названием Douyin. TikTok использует искусственный интеллект для отображения персонализированного контента пользователю путем анализа интересов и предпочтений пользователей посредством их взаимодействия с контентом в приложении (<https://clck.ru/at9QS>). Хотя за последние три года их популярность значительно возросла, были высказаны опасения по поводу их влияния на пользователей и общество (<https://clck.ru/at9RU>). Зависимость от TikTok также становится проблемой (<https://clck.ru/at9Qt>). Многие пользователи тратят по несколько часов в день на просмотр этих видеороликов, что не очень полезно для здоровья. В процессе просмотра именно коротких видео, происходит быстрый выброс гомонов радости и удовольствия, один из которых дофамин. Именно такой вид зависимости возникает у пользователей данного приложения. Если человек продолжит получать «быстрый» дофамин, то постепенно мозг научится получать его искусственно, при этом сам будет производить в меньших количествах. Снижается также способность получения дофамина посредством другой деятельности, так как на это требуется больше времени.

Социальные сети доминируют в технологическом мире. Из-за этого становится все более важным, чтобы эти платформы были не только надежными, но и безопасными. Одной из самых последних платформ социальных сетей, появившихся на рынке, является TikTok, популярность которого растет рекордными темпами, что делает его одним из наиболее часто используемых приложений в 2022 году. В октябре 2018 года это было самое загружаемое приложение для фотографий и видео в Apple Store по всему миру. Приложение ежемесячно набирает более 500 миллионов активных пользователей. Многие пользователи скачали это приложение, чтобы занять себя во время пандемии Covid 19. Несмотря на популярность, TikTok получил пристальное внимание американских законодателей. Угрозы о национальной безопасности побудили правительство США рассмотреть вопрос о запрете TikTok. Другие

страны, такие как Индия, уже ввели национальные запреты по аналогичным причинам. Проблемы национальной безопасности, связанные с TikTok, относятся к материнской компании и к их сомнительным отношениям с данными пользователей. В последнее время многие поделились своими исследованиями, полученными в результате анализа приложения. Эти результаты подчеркивают проблемы безопасности в приложении, уделяя особое внимание разрешениям приложений, механизмам криптографии и дополнительным проблемам, которые представляют угрозу для персональных данных пользователей. TikTok принадлежит китайской интернет-технологической компании «ByteDance», и недавно власти Великобритании провели расследование в связи с возможным неправильным использованием детских данных (<https://lnnk.in/hHcw>). Несмотря на то, что ByteDance внесла некоторые изменения с тех пор, как была привлечена к различным судам в разных странах, проблемы конфиденциальности продолжают возникать, поскольку ее бизнес опирается на пользовательский контент.

FaceApp, приложение для редактирования фотографий на основе искусственного интеллекта, стало вирусной сенсацией. Приложение завоевало популярность, потому что оно трансформировало «селфи» человека, чтобы он выглядел намного старше или моложе. Приложение действительно взлетело, когда знаменитости начали делиться фотографиями в Instagram о том, как они будут выглядеть в более зрелом возрасте. FaceApp был загружен и использован более 100 миллионов раз. Это было незадолго до того, как использование FaceApp стало тревожным. Люди начали понимать, что они не знают, кому именно они дали доступ к своей фотопленке. Распространились слухи, что российская компания, создавшая FaceApp, вела учет загруженных фотографий своих пользователей и имела доступ к библиотекам фотографий. FaceApp принадлежит российской компании «Wireless Lab» и вызвало критику в связи с возможным неправильным использованием данных и нарушениями конфиденциальности [4, с. 1-7]. Дело в том, что были высказаны опасения, что FaceApp обрабатывает фотографии путём отправки их в облако, т.е. редактирование не происходит локально на устройстве пользователей. Разработчики подтверждают, что большая часть обработки, необходимой для усиления эффектов, изменения пола или возраста выполняется в облаке. Однако они утверждают, что приложение загружает только те фотографии, которые пользователи выбрали сами для редактирования. FaceApp утверждает, что он может хранить фотографии, которые пользователи решили загрузить, в облаке в течение короткого периода для «производительности трафика». Например, чтобы убедиться, что приложение не загружает одну и ту же фотографию повторно. Также разработчики добавляют, что большинство изображений удаляются с серверов в течение 48 часов с даты загрузки.

Открытость рынка приложений для Android и отсутствие надлежащего тестирования безопасности разработчиками приложений привели к тому, что авторы вредоносных программ нацелили эти приложения на создание и распространение вредоносных программ для смартфонов.

Такие приложения, как FaceApp и TikTok, в настоящее время пользуются огромным успехом и очень популярны среди людей всех возрастов благодаря огромному росту их числа загрузок. Их сильные стороны заключаются в простоте использования или веселье, которые они приносят общедоступному пользователю. В новом исследовании FaceApp и TikTok авторы обнаружили, что FaceApp действительно удаляло внедренные метаданные, а также удаляло скрытые сообщения на изображениях при загрузке для обработки приложением [5, с. 38-59]. Анализ стеганографии помог определить, что приложения удаляют скрытые тексты из загруженных изображений, что не позволяет злоумышленникам делиться стеганографической информацией. Программный код приложения был проверен для обоих приложений на наличие любой вредоносной логики, которая может поставить под угрозу конфиденциальность пользователя. Проанализировав files.apk, авторы обнаружили шестнадцатеричные подписи двух приложений, их сертификаты RSA, разрешения, назначенные пользователем, и сетевые действия, чтобы назвать некоторые из них. Анализ этих приложений также включал анализ сетевого трафика, который показал, что FaceApp использует шифрование для всего трафика, однако TikTok не шифрует видео при передаче их от отправителей получателям, что делает его уязвимым для сетевых анализаторов. Авторы рекомендуют TikTok шифровать весь трафик независимо от типа пользователя (платный или бесплатный). Авторы рассмотрели опубликованные политики конфиденциальности FaceApp и TikTok из Интернета и представили основные моменты, согласившись с тем, что они, по-видимому, охватывают достаточно вопросов для пользователя. Однако рекомендуется периодически проверять их уровень безопасности, конфиденциальность, права собственности, авторские права и места хранения этих приложений, поскольку изменения в них с помощью более новых версий или исправлений безопасности могут изменить их архитектуру и дизайн. В качестве наилучшей практики авторы предлагают всем пользователям при использовании таких приложений, чтобы в фотографии и в видео не включалась конфиденциальная или личная информация пользователя. В связи с тем, что распознавание лиц вызывает споры в сочетании с технологией глубокой подделки, злоумышленники могут использовать общедоступные фотографии для создания потрясающе реалистичных видеороликов о человеке, использующем только одно изображение своего лица, тем самым увеличивая вероятность неправильного использования.

В этой статье приведены оценки приложения FaceApp и TikTok с разных точек зрения, чтобы определить их безопасность, конфиденциальность, управление данными и криминалистическую позицию. Следует отметить, что, хотя важно получать удовольствие от современных технологий и социальных сетей, пользователи приложений FaceApp и TikTok должны тщательно продумать меры предосторожности, которые они принимают для защиты своей личности. Существует огромное количество социальных сетей и не только эти два приложения считаются небезопасными. Каждый человек должен ответственно пользоваться приложениями и не фотографировать, снимать или выкладывать конфиденциальную информацию. Фотографии, видео, личные данные — всё это принадлежит к тщательному сокрытию для безопасности в интернете. Использование безопасности в социальных сетях в

настоящее время является темой, которая широко обсуждается пользователями. Люди по-прежнему, не беспокоятся о безопасности социальных сетей. Все приложения соединяются друг с другом и обмениваются информацией, поэтому пользователи должны помнить о недостатках и последствиях использования социальных сетей. Благодаря всем этим исследовательским работам становится совершенно ясно, что социальные сети сопряжены с высокими рисками безопасности, а также с рисками для конфиденциальности. Экспертам по информационной безопасности, сотрудникам правительства, полиции, разведки, национальной безопасности необходимо разработать новые стратегии борьбы с возникающими будущими рисками и угрозами и адаптации к ним.

Литература

1. Алемасов Е.П., Зарипова Р.С. Тенденции развития сферы мобильных приложений в современном обществе // Социальная онтология России. Сб. научных статей по докладом XIV Всероссийских Копыловских чтений. Новосибирск, 2020. С. 399-402.
2. Сун Х.М., Венг К.Ю., Ли К.Ф., Янг К.Х. Анти-криминалистика с внедрением стеганографических данных в цифровые изображения // Журнал IEEE по отдельным областям связи. 2011. № 29(7). С. 1392-1403.
3. Burrows C., Zadeh P.B. A mobile forensic investigation into steganography // 2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security). IEEE, 2016. P. 1-2.
4. Гонсалес Х., Кадир А.А., Стаханова Н., Аль-Захрани А.Дж., Горбани А.А. Изучение симптомов обратного инжиниринга в приложениях для Android // Восьмой Европейский семинар по системной безопасности. 2016. С. 1-7.
5. Neyaz A., Kumar A., Krishnan S., Placker J., Liu Q. Security, privacy and steganographic analysis of FaceApp and TikTok // International Journal of Computer Science and Security (IJCSS). 2020. Т. 14. № 2. P. 38-59.

© Сиразева А.Л., Зарипова Р.С., 2022

ВЛИЯНИЕ ИНФОРМАЦИОННОГО ОБЩЕСТВА НА РАЗНЫЕ СФЕРЫ ДЕЯТЕЛЬНОСТИ

Трансформация рынков капитала сыграла важную роль в развитии всей новой экономики. Развитие финансовых рынков через компьютерные сети и их новые правила обеспечили финансирование интернет-экономики. В электронном бизнесе существует механизм, с помощью которого инновации финансируются на рынках капитала. Например, этот рабочий цикл в Силиконовой долине в конце 1990-х годов состоял из создания бизнес-плана и продажи его фонду венчурного капитала. Независимые инвесторы также могут вкладывать частные средства в перспективные бизнес-проекты. Однако многие проекты так и не дошли до стадии реализации или выхода на рынок. Если проект оказывался достаточно успешным, у компании была возможность получить дополнительный капитал за счет положительной оценки проекта инвесторами и заняться серьезным бизнесом: она могла стать конкурентоспособной компанией или быть поглощенной более богатой компанией. Его дальнейшее существование зависело от способности компании приносить доход и зарабатывать деньги. Оценка инвесторов на финансовом рынке основана на способности блеснуть в финансовом мире, технических инновациях и предпринимательском творчестве.

В конце 1990-х годов – начале 2000-х происходил такой процесс, во время которого наблюдались усиление глобализации и, поддерживаемая сетью компьютерных сетей, взаимозависимость между финансовыми рынками. М. Кастельс пишет об этом следующее: «... финансовые колебания на каком-либо рынке в любой точке земного шара в принципе могут перекинуться на другие рынки вне зависимости от различий между характером национальных экономик и уровнями рыночной стоимости». [1, с. 103]. Регулировать деятельность финансовых рынков так же становилось все труднее и труднее.

Равным образом видоизменяются финансовые рынки и под влиянием электронной торговли. Фондовая биржа NASDAQ, специализирующаяся на продаже акций, к примеру, не имеет определенного места для проведения торгов, так как является электронным рынком. Поскольку NASDAQ являлась средством для публичного размещения акций, она сыграла важную роль в развитии новой экономики.

Под трансформацию попали и валютные рынки, которые постепенно становятся электронными. В 2000 году имел место быть проект совместного предприятия Лондонской и Франкфуртской фондовыми биржами с NASDAQ и Токийской фондовой биржей, который привел к созданию глобальной ассоциации NASDAQ.

Технология осуществления сделок является основой для усиления взаимозависимости глобальных финансовых рынков, поскольку она может сократить объемы транзакционных издержек как минимум на 50%, а этот факт, в свою очередь, способствует увеличению количества инвесторов и сделок. Инвестиции в режиме онлайн имеют следующие

результаты: увеличение объемов рынка за счет мобилизации сбережений и их инвестирования в разных местах; онлайн-информация помогает инвесторам в принятии тех или иных решений; посредничество перестает быть важным фактором в установлении взаимоотношений (к примеру, индивидуальные инвесторы обходятся без традиционных брокеров и инвестиционных фирм).

В эпоху Интернета становятся легко доступными каждому желающему различные слухи и новости, зачастую конфиденциальные. Имеют место быть и объявления, поднимающие престиж фирмы, или сообщения о финансовых спекуляциях, что вызывает информационную неопределенность. В такой период частные инвесторы должны быть внимательны к поступающей информации и способны быстро реагировать на всё быстрое действие рынка [2, с. 138].

В электронной экономике основным источником повышения производительности труда является рабочая сила, тогда творческие способности рабочей силы и эффективная организация бизнеса обуславливаются уже новаторством. Свободный доступ к онлайн-информации способствует генерации знаний, которая, в свою очередь, является функцией новаторства.

Новаторство является не только продуктом умственного труда, но и оказывается результатом работы коллективного интеллекта, поскольку процесс создания инновации сопровождается коллективной работой сети. М. Кастельс полагает, что «процесс новаторства в условиях электронной экономики постепенно дрейфует в сторону информационно-открытых сетевых сообществ, состоящих не только из свободных индивидуумов, но и из предпринимателей и служащих компаний, поскольку фирмы заинтересованы в содействии новаторству, ибо они самыми первыми извлекают выгоду из результатов совместной работы». [1, с. 125].

В 2000-х годах многие начинающие Интернет-фирмы оказались нежизнеспособны. В это же время финансовые рынки так же не обошли стороной и крупные технологические компании, которые показывали огромную эффективность, большую прибыль, но считались причинами испытываемого спада. Рассмотрим несколько таких компаний:

1. Акции Nokia, несмотря на хорошие показатели успешности их деятельности, упали в цене в августе 2000 года из-за задержки поставки новых моделей мобильных телефонов и предупреждения снижения доходов в следующем квартале;

2. Из-за того, что ожидания по поводу доходов Dell и Intel, ведущего производителя ноутбуков и лидера в области микроэлектроники, не оправдались, компании потеряли 50% своей стоимости.

3. Потеря 80% стоимости акций в марте 2001 года компании Yahoo!, являющейся популярной поисковой системой, вынудила уйти в отставку исполнительного директора.

4. Корпорация Microsoft была под угрозой дробления, но держалась за монополию на исчезающем рынке персональных компьютеров, поэтому понесла меньшие убытки по сравнению с другими компаниями, попадавшими в такую же ситуацию. Этот факт поспособствовал повышению стоимости акций компании в первом квартале 2001 года.

5. Товарооборот компании Amazon к концу 2000 года достиг объема сбыта в 3 миллиарда долларов, но несмотря на это ее акции еще летом упали в цене на 60%.

Факторами, замедляющими падение стоимости акций, оказались способности к раскрутке и умелое создание бизнес-имиджа организации. Ярким примером использования таких факторов была Nokia, которая после падения стоимости ее акций 19 октября 2000 года объявила о многообещающих доходах ее компании, что поспособствовало увеличению стоимости ее акций на 27% за один день и даже поднятию индекса NASDAQ, хотя Nokia не торговала через NASDAQ.

В новых условиях в экономике, где возрастает потребность в умении находить, обрабатывать и использовать информацию, возникает огромное давление на рабочую силу. Электронная экономика не может функционировать без специалистов, которые умеют работать с этим морем информации. Качество определяется характером обучения сотрудников и самостоятельностью работающих в электронной экономике, которая заключается в способности адаптироваться к условиям постоянно меняющейся бизнес-среды.

Основным фактором производства в электронном бизнесе являются талантливые люди, умеющие использовать потенциал Интернета и обладающие собственным интеллектуальным капиталом. Компании вкладывают много энергии в эффективное привлечение, удержание и использование этих сотрудников в своей команде. Делают они это с помощью различных уловок: материальных благ, премий, подарков и т.д.

В самых быстрорастущих отраслях вскоре возникла нехватка рабочей силы из-за того, что компании осознали важность и потребность в самопрограммируемой рабочей силе для электронного бизнеса. Однако потребность в электронном финансировании была быстро удовлетворена за счет увеличения числа женщин, получающих высшее образование и устраивающихся на высокооплачиваемую работу. В 1990-е годы, несмотря на гендерную дискриминацию в бизнесе и промышленности, разрыв в оплате труда женщин и их коллег-мужчин сократился [3, с. 356].

Иммиграция также стала основным источником предложения талантов. Столкнувшись с возможностью того, что рынки труда не смогут предоставить рабочую силу в области информационных технологий, Европа признала важность привлечения профессиональных рабочих-мигрантов. Таким образом, многие иммигранты не только поселились в важном техническом или экономическом центре, но и впоследствии вернулись на родину и основали собственные компании.

В электронном бизнесе не всякая рабочая сила является самопрограммирующейся. Существует и общая рабочая сила. М. Кастельс считает, что «олицетворением общей рабочей силы являются работники, не имеющие специальных навыков или не обладающие способностью приобретения таковых в ходе производственного процесса, и использующие лишь те навыки, которые требуются для выполнения указаний руководства». [1, с. 117]. Факт существования общей рабочей силы несколько не зависит от качеств отдельной личности. Это вытекает из отсутствия вложений интеллектуального капитала в данного человека.

А. Турен говорит, что такие действия «не всегда принимают форму сознательной воли, воплощенной в личности или даже группе людей. Вот почему такое общество должно называться программируемым обществом, обозначение, которое ясно указывает на его способность создавать модели управления производством, организацией, распределением и потреблением; поэтому такого рода общество появляется на операциональном уровне не в результате естественных законов или специфических культурных характеристик, а скорее как результат производства, благодаря действию общества само на себя, его собственным системам социального действия». [4, с. 415].

В сегодняшнем мире, когда знания и информация распространяются по всему миру, вся рабочая сила имеет возможность программировать себя. Однако вся рабочая сила останется неизменной, пока модели неравенства, социальные институты и профессиональные приоритеты будут идти рука об руку. Гибкость — одно из важнейших проявлений трансформации условий труда. Например, М. Кастелл процитировал исследование Криса Беннера (2001), которое показало, что «гибкая практика найма, заложенная в агентствах по трудоустройству, и гибкая политика найма являются отличительной чертой экономики Силиконовой долины». [1, с. 119].

Человеческое сознание в информационном обществе будет важнейшим пластом культурного развития в этом обществе. Для овладения основами информационной культуры необходимо уметь анализировать и обрабатывать полученную информацию. Необходимо приобрести такие навыки, которые позволяют выбрать и правильно определить цель, самостоятельно сформулировать задачу, найти и интерпретировать информацию.

Непрекращающийся интерес к этой информации еще до того, как она генерируется или воздействует на людей, доминирует в информационной культуре современного общества. Знания, полученные благодаря интеллекту и глубокому знанию процессов, отражающих эти знания, определяют уровень эффективного участия человека в новом информационном обществе. Образование, направленное на вовлечение людей в общественную жизнь, предполагает взаимосвязанные процессы - воспитание и обучение. Этот процесс имеет свою цель в формировании и развитии личности человека. В процессе обучения осваивается личностный и/или иной общественно значимый способ обращения с миром, характерный для контекста его окружения. В процессе обучения вы приобретаете общие и специальные знания о современном мировоззрении. Л. Лоу считает, что «смысл передачи знания есть нечто большее, чем мотивация передачи, поскольку желание овладеть знанием и способности обучаемого являются решающим фактором, если не сказать более. Для культуры передачи знания, его восприятия и накопления интеллектуального капитала необходимы личностные способности, особые модели ментальности, системное мышление и некоторые другие ингредиенты» [5, с. 192].

Скачок производительности труда позволяет утверждать, что это новая экономика, а не просто технологическая революция. Благодаря инвестициям в информационные технологии можно ожидать значительного повышения производительности. В целом появление экономики, способной к стабильно высокому росту почти до полной занятости, с низкой

инфляцией и низким ростом доходов, можно объяснить повышением производительности труда.

Новаторы и создатели новых технологий меняют организационную структуру, потому что они первыми используют ее и готовят к ней работников, получая таким образом выгоду от роста производительности труда. Инвестиции США в ИТ-оборудование составляли 50% от общего объема инвестиций бизнеса в новую экономику в 2000 г. во второй половине 1990-х гг. [1, с. 123].

Таким образом, Интернет — это общество, поскольку оно выражает интересы, процессы, институты и социальные и политические ценности. Уникальность Интернета заключается в том, что он формирует специфическую материальную и технологическую основу сетевого общества. Интернет повторно использует виртуальность и меняет нашу реальность, в результате чего возникает сетевое общество — общество, в котором мы живем. Электронная коммерция помогает трансформировать финансовые рынки. Стоит обобщить некоторые факторы, связанные с изменением финансовых рынков. Электронная коммерция увеличивает количество инвесторов с различными стратегиями (инвесторы ищут возможность заработать деньги, а интернет-технологии представляют собой заманчивые и многообещающие проекты). Валютные рынки становятся электронными (это поощряет онлайн-инвестиции, увеличение объема торгов и агрегацию фондового рынка). Объем рынка растет за счет мобилизации сбережений и их инвестирования в разные сферы. Информация, размещенная в Интернете, легкодоступна для всех. Экономический спад переживают не только интернет-стартапы, но и крупные технологические компании.

Литература

1. Кастельс М. Галактика Интернет: Размышления об Интернете, бизнесе и обществе. М.: У-Фактория, 2004. 328 с.
2. Делягин М.Г. Мировой кризис. Общая теория глобализации. М.: Юнити-Дана, 2005. 768 с.
3. Почепцов Г.Г. Теория коммуникации. М.: «Рефл-бук»; К.: «Ваклер», 2001. 656 с.
4. Турен А. От обмена к коммуникации: рождение программированного общества // Новая технократическая волна на Западе. М.: Прогресс, 1986. С. 410-430.
5. Лоу Л. Образование и развитие человеческих ресурсов: движущая сила следующего столетия // Постиндустриальный мир и Россия. М.: Эдиториал УРСС, 2001. С. 179-196.

© Соколова А.И., 2022

СОЗДАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ УМНЫМ ДОМОМ НА ПЛАТФОРМЕ ANDROID

«Умный дом» относится к удобной домашней установке, где приборы и устройства могут автоматически управляться удаленно из любого места с подключением к Интернету с помощью мобильного или другого сетевого устройства. Устройства в умном доме взаимосвязаны через Интернет, что позволяет пользователю удаленно управлять такими функциями, как доступ к дому, температура, освещение и домашний кинотеатр. Другими целями являются повышение безопасности и более эффективное использование энергии благодаря подключенным устройствам с дистанционным управлением (<https://clk.ru/epuyD>). Условно все устройства умного дома можно разделить на несколько типов (рис. 1):



Рис. 1. Типы устройств умного дома

Контроллер – это «мозг» умного дома. Этот прибор контролирует работу сети и всех входящих в нее устройств, хранит в своей памяти сложные сценарии (определенные наборы действий) и обеспечивает связь системы умного дома с вашим смартфоном, планшетом или компьютером.

Различные датчики, определяющие движения, задымления, протечки, – это «органы чувств» умного дома. Благодаря им система непрерывно получает информацию о том, что происходит в доме.

Исполнители – группа устройств, которые выполняют команды системы, отдаваемые на основании данных от датчиков. Например, шаровой кран мгновенно перекрывает воду, если датчик обнаруживает протечку. Реле выключают и включают свет, а диммеры меняют яркость. Устройства управления – приборы для комфортного использования системы умного

дома. Это пульты дистанционного управления или, например, выключатели света на батарейках, которые можно разместить там, где вам удобно. Прочее интегрируемое оборудование и веб-сервисы – видеокамеры, оборудование и сервисы, имеющие открытый API (мультимедиа-оборудование, различная современная техника, проекторы и т.д.) или управляемые ИК пультами дистанционного управления (<https://clck.ru/epuz5>).

Приложение построено на клиент-серверной архитектуре, представляющей собой иерархическую сеть, которая состоит из узлов-клиентов (рис. 2) (их может быть от одного и до неограниченного количества) и центрального сервера, через который выполняется хранение и обработка данных, а также передача их в обоих направлениях.

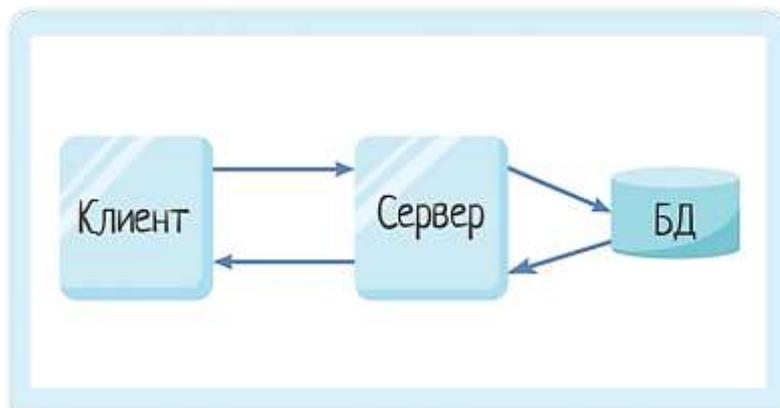


Рис. 2. Клиент-серверная архитектура

Основной экран приложения представлен на рисунке 3 и включает в себя: статус подключения Sip-аккаунта; кнопка вызова бокового меню; спинер выбора группы камер; вкладка «Недавние» – список недавних звонков с датой и фото; вкладка «Подсмотреть» – выводит видеопоток с домофона; вкладка «Все камеры» – список доступных для просмотра камер; вкладка «Открыть» – открывает домофонную дверь.

Каждый пункт меню содержит свой фрагмент, заменяемый в контейнере. Статус-бар и кнопка бокового меню остаются на прежнем месте. При авторизации пользователя ему выдается sip-аккаунт на основе его адреса проживания, для взаимодействия с домофоном и камерами видеонаблюдения. Sip – это протокол передачи данных, описывающий способ установления и завершения пользовательского сеанса связи, включающего обмен мультимедийным содержимым.

При звонке с домофона вызов осуществляется одновременно на стандартную домофонную панель и мобильное приложение. С домофона на сервер передается адрес квартиры, на нем осуществляется поиск всех устройств, зарегистрированных на логин с данным адресом, и отправляется им уведомление о входящем вызове и изображение с домофона.

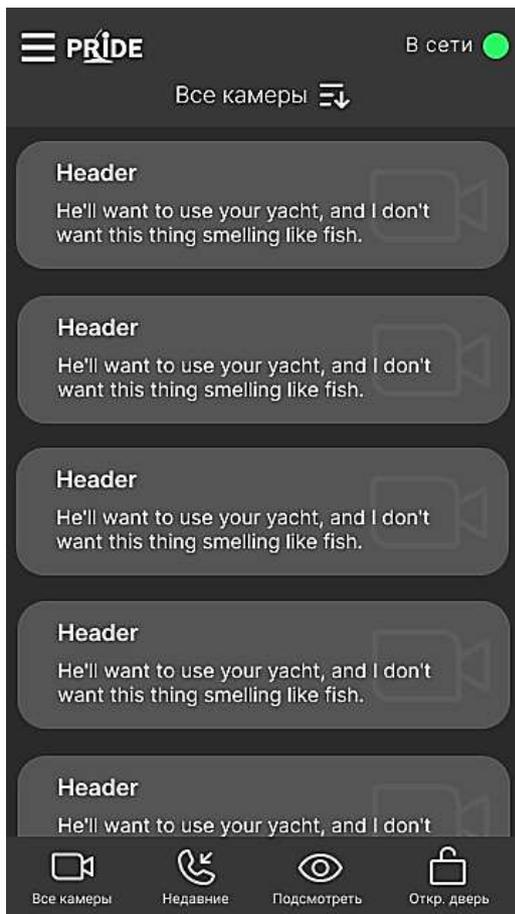


Рис. 3. Главный (домашний) фрагмент

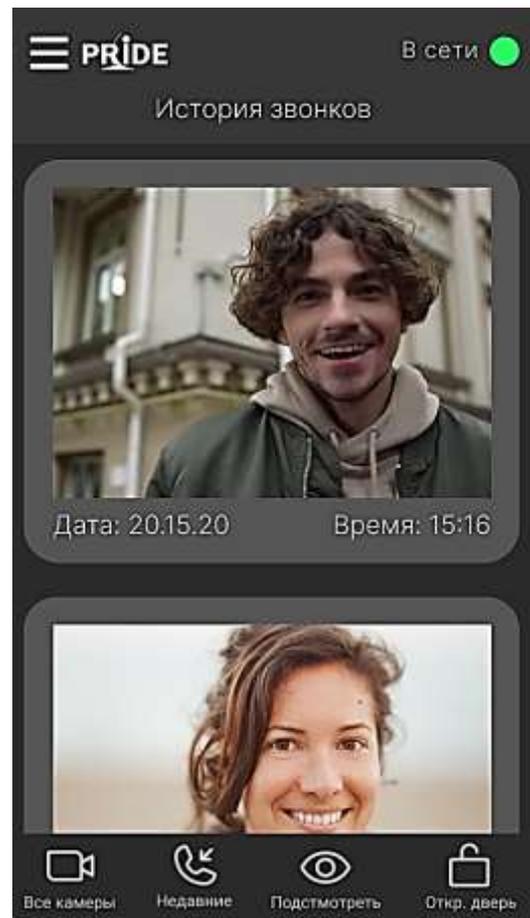


Рис. 4. Фрагмент «История звонков»

При совершении звонка на сервере сохраняется дата, время и скриншот с домофонной панели. При переходе на вкладку «Недавние» (рис. 4) передается на устройство пользователя и сохраняется в памяти устройства.

© Сорокин А.О., Слива М.В., 2022

СОВРЕМЕННЫЕ ТРЕНДЫ ТРАНСФОРМАЦИЙ ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКОГО ПРОСТРАНСТВА

Современный мир непрерывно развивается, информационные технологии уже стали неотъемлемой частью нашей жизни. Стремительная цифровизация, охватившая все сферы жизни, набирает обороты. Вместе с тем фундаментально меняются потребности людей и общества в целом, на что, в том числе, оказал влияние кризис, связанный с распространением коронавирусной инфекции. Все это послужило новым толчком к развитию ИТ-сектора и обозначило необходимость формирования альтернативной экономической модели и иной организации общества. Так, с учетом вышеуказанных и глобальных вызовов, начали появляться новые тренды социально-экономических трансформаций.

В условиях пандемии, связанной с распространением коронавируса, возникает необходимость взаимодействия людей друг с другом и с окружающим миром преимущественно посредством мониторов компьютеров и экранов смартфонов, что является, в свою очередь, вызовом для компаний и разработчиков в ИТ-сфере. Ведь инновационные решения в области информационных технологий являются основой для налаженной и эффективной работы как малых и крупных компаний в частности, так и всей экономической системы в целом [1, с. 46].

Новым вектором тренда цифровизации в этом году становится концепция «digital first», главной стратегией которой является максимальное развитие бизнеса в электронной среде, что значительно расширяет спектр возможностей людей. Особое влияние повсеместное использование информационно-коммуникационных технологий оказывает на изменение структуры занятости, вызывая коренные изменения процесса трудоустройства. Развитие глобальной сети Интернет расширило возможности взаимодействия между участниками трудовых отношений и стало толчком к переосмыслению самого подхода к работе. Свои функциональные обязанности представители многих профессий могут выполнять в режиме удаленного доступа. После окончания периода вынужденной удаленной работы, который был связан с распространением вируса COVID-19, какие-то организации вернулись к привычному режиму работы в офисе, некоторые планируют полный переход на дистанционный формат, а есть и такие, которые остановились на смешанном варианте занятости. Исследование, проведенное компанией «Future Forum» (<https://futureforum.com/>), показало, что 93% работников интеллектуального труда предпочли бы иметь гибкий график работы, а 76% – возможность работать удаленно.

«Внедрите в работу цифровые инструменты общения, например, документы совместного пользования или рабочие чаты, и подтолкните сотрудников к более гибкому рабочему графику, нацеленному на максимальную продуктивность», – эти слова Мегана

Райбштайна, вице-президента по организационным операциям в компании «Zillow» (<https://www.zillow.com/>), указывают на то, что важными аспектами для эффективной самореализации работников и их высокой конкурентоспособности на современном рынке труда являются навыки работы с информацией и навыки коммуникации посредством современных технических средств, требующие определенного уровня технической культуры. Профессионалы, обладающие должными в информационной и технической области знаниями, – именно тот необходимый штат сотрудников, с помощью эффективного функционирования которого организация сможет добиться наилучшего графика работы и наивысшего уровня производительности [2, с. 398].

Информационные технологии кардинально изменили содержание коммуникации, придав информационной среде социальный характер. Уже сейчас происходит формирование новой модели взаимодействия объектов социальной реальности при помощи использования электронного оборудования и информационных сетей. В связи с этим все большую популярность набирают работные сайты, где осуществление подбора сотрудников и поиска вакансий возможно в онлайн-формате, что способствует ускорению и упрощению социального взаимодействия субъектов трудовых отношений.

В текущем году рекрутинговая компания «Antal» (<https://antalrussia.ru/>) провела опрос среди организаций с целью выяснения, какие работные сайты они используют для поиска кандидатов в 2021 году. Процентная статистика ответов графически представлена на рис. 1. По приведенным данным, 87% организаций используют сайт «HeadHunter», 21% осуществляют подбор персонала на онлайн-портале «Superjob», а другие 15% пользуются «Avito», при этом 12% компаний вовсе не используют сайты для поиска работников. Однако, такие платформы как «Rabota.ru», «Linkedin», «Amazing Hiring», «Indeed», «Worki», «ТрудВсем» и «Trud.com» являются менее популярными среди работодателей.

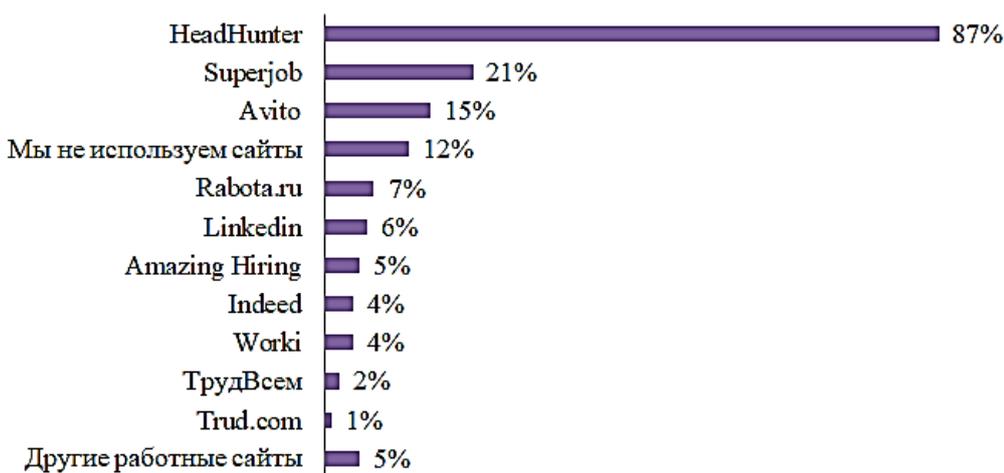


Рис. 1. Статистика использования сайтов для поиска кандидатов в 2021 году

Стоит заметить, что Интернет-коммуникация не является способом замещения традиционного социального взаимодействия, а трансформирует его и воспроизводит по-новому на просторах глобальной сети. Как по своему содержанию, так и по последствиям

онлайн-взаимодействие реально и затрагивает все сферы жизни общества, начиная с развитых сервисов автоматизированной доставки товаров, популяризации дистанционного обучения и заканчивая роботизацией производственных процессов и медицинского оборудования [3, с. 120]. Сегодня заказывать доставку еды из ресторана, расположенного на другом конце города, или приобретать обувь из других стран на сайте бренда производителя – не заоблачная мечта, а настоящая реальность.

Продуктивно функционирует, быстро развивается и совершенствуется новая концепция сети – Интернет вещей (IoT), способная передавать данные между физическими объектами («вещами»), в свою очередь оснащенными встроенными средствами и технологиями связи друг с другом или с внешней средой. IoT применяется во всех сферах нашей жизни и призван помочь нам создать более удобный, «умный» и безопасный мир. По данным РБК (<https://www.rbc.ru/>), в 2021 году к нему будет подключено уже 25 миллиардов устройств по всему миру.

IoT используется в различных отраслях для достижения поставленных целей при помощи таких выполняемых им функций, как отслеживание и анализ потребительского поведения в реальном времени, совершенствование и диагностика работы машин и систем, поиск новых методов работы в условиях глобальной цифровизации и многих других. Можно выделить такие основные области использования и функционирования IoT, как производство, розничная торговля, медицина и энергетика [4, с. 109].

Невозможно переоценить влияние интернета вещей на одну из самых востребованных на сегодняшний день сфер – на медицину. Врачи могут более точно ставить диагнозы, максимально эффективно выстраивать планы лечения и реабилитации, что обеспечивает возможность непрерывного мониторинга состояния пациентов в тяжелом состоянии и повышает уровень обслуживания и безопасности пациентов. Все это достигается с помощью данных, полученных с использованием технологий Интернета медицинских вещей, которые способны в режиме реального времени осуществлять сбор информации о физиологических процессах и физическом состоянии пациентов, считывая ее с носимых устройств или медицинских сетей.

В последние годы можно проследить стремительное ускорение процесса роботизации, который упрощает жизнь людей и повышает показатели эффективности бизнеса. Однако повышение производительности, сокращение продолжительности производственного цикла, снижение производственных затрат, а также повышение качества и увеличение объема продукции затмевает все возможные неблагоприятные последствия роботизации, ведущими из которых являются уменьшение количества рабочих мест, потребность обучения и переобучения персонала, необходимость реконструкции помещений и дополнительных затрат на установку и настройку оборудования.

По данным «Research and Markets» (<https://www.researchandmarkets.com/>), в 2018 году объем мирового рынка промышленной робототехники в целом составил 48,7 миллиардов долларов, а в 2024 году прогнозируется увеличение его объема до 75,6 миллиардов с темпом роста 9,2%. Аналитики считают, что одним из ключевых трендов социально-экономических

трансформаций станет именно рост автоматизации и популяризация коллаборативных роботов. Это произойдет за счет сокращения кадрового состава на производствах [5, с. 49].

Большую роль в экономике, образовании и науке, медицине и даже искусстве играет искусственный интеллект (ИИ). Данный тренд современных социально-экономических трансформаций в сфере торговли безоговорочно упростит процесс покупки, сделав его быстрее и удобнее, а также будет способствовать улучшению клиентского опыта. Крупные ИТ-компании уже разрабатывают сервисы голосового и визуального поиска товаров и услуг. Одними из главных преимуществ новых систем станет возможность предсказания спроса за счет сбора и анализа более точной информации о предпочтениях покупателей. Итак, компания вызывает рост продаж, предлагая покупателям товары на основе их предпочтений и по лучшим ценам, при этом полностью удовлетворяя своих клиентов и выстраивая хорошие взаимоотношения с целевой аудиторией [6, с. 81].

Многие из нас до сих пор не знают о существовании так называемого Интернета поведения. У истоков этого тренда стоит искусственный интеллект, который постоянно собирает сведения о каждом из нас. Получение информации с помощью поисковых систем, мессенджеров, голосовых помощников, диктофонов и геолокации – способ, которым нейросети изучают и учатся понимать наши потребности с целью предложения лучшего контента. За 2020 год прирост Интернет-пользователей составил 7,3%, на 316 миллионов больше, чем за 2019 год, а на январь 2021 года к интернету подключены 4,66 миллиардов человек во всем мире. Аналитическая компания «Gartner» (<https://www.gartner.com/>) считает, что к 2023 году уже 40% людей будет находиться под постоянным наблюдением своих устройств. Такие перспективы наталкивают на негативные размышления. Однако конечная цель развития информационных технологий – облегчать жизнь людям, и чем лучше развиты ИТ, тем полезнее они становятся. Сегодня даже размещаемая реклама становится настолько индивидуально подобранной, что предлагает действительно необходимые товары. Таким образом, взаимодействие человека с компьютером становится все продуктивнее, что неоспоримо является плюсом.

Искусственный интеллект может в равной степени как восхищать нас своими обширными возможностями и «умениями», так и пугать. Электронные устройства «знают» огромное количество информации о своих владельцах: номера телефонов, данные банковских карт, сведения о паспортах и даже точное местоположение [6, с. 257]. И тут возникает вопрос обеспечения безопасности личной информации – потребность в кибербезопасности. Данный тренд продиктован временем, когда границы личного пространства сильно размыты. Только за прошедший год, по данным информационного портала «Security Lab» (<https://www.securitylab.ru/>), количество киберпреступлений возросло на 80%. Цифры статистики неутешительны, и теперь множество людей с недоверием и подозрением относятся даже к надежным поставщикам товаров и услуг, предпочитая дополнительно совершить проверку их достоверности или же вовсе отказаться от покупок.

Новым толчком к развитию кибербезопасности стало многократное увеличение числа сотрудников, выполняющих свои обязанности при помощи удаленного доступа.

Распространение дистанционного формата работы требует усиления корпоративных систем безопасности, что приводит к повсеместному внедрению мультифакторной аутентификации. Возрастающая популярность удаленной работы влечет за собой непрерывное совершенствование систем безопасного доступа как к корпоративным, так и к личным данным.

Несмотря на специализированные социальные программы, направленные на повышение грамотности населения в области информационных технологий и обеспечения безопасности и защиты данных, риск подверженности граждан и организаций взломам и кибератакам остается высоким. Одной из причин может быть названа недостаточно высокая степень совершенствования систем безопасности по сравнению со скоростью развития системных продуктов взлома, компьютерных вирусов и тому подобного. Однако существующие и применяемые методы обеспечения и правила безопасности пользования электронными устройствами, Интернетом и мобильной сетью в большинстве своем все-таки оказываются эффективными.

Формирование новых трендов современных социально-экономических трансформаций непосредственно оказывает влияние на затраты на ИКТ. По данным информационного агентства «IDC» (<https://www.idc.com/cis/russia>), начиная с 2020 года, прогнозируется ежегодный рост мировых затрат на информационно-коммуникационные технологии (рис. 2), что обусловлено новыми и быстро растущими потребностями общества, а также возможностями ИТ-компаний удовлетворить их.

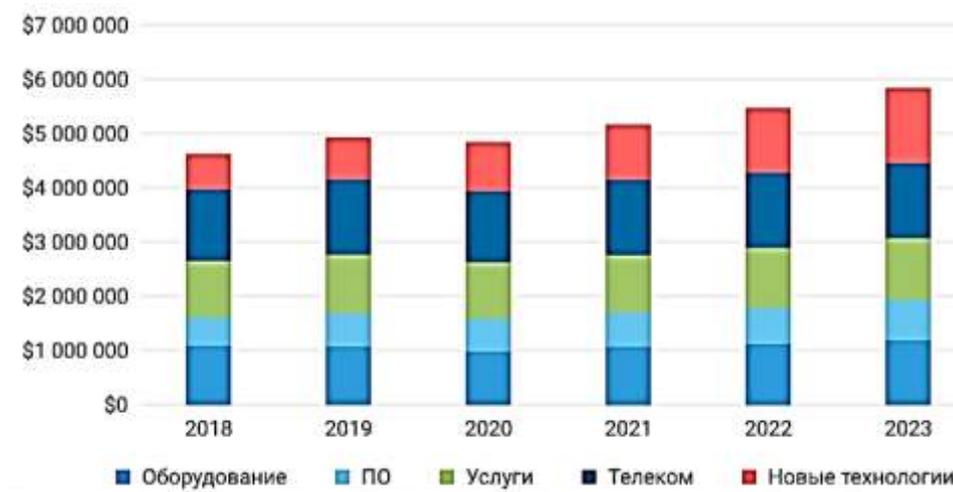


Рис. 2. Мировые расходы на ИКТ 2018-2023

Уже в ближайшем будущем человечество сможет совершить качественный прорыв в своем существовании, развивая и совершенствуя информационные технологии во всех направлениях, начиная с экологического производства продуктов питания и заканчивая разработкой новых лекарственных препаратов и даже вакцин. С помощью использования цифровых технологий медицинское обслуживание и исследования в этой области станут менее затратными, но при этом более эффективными за счет обращения ко всему человеческому опыту и всемирным источникам информации.

Несомненно, есть задачи, с которыми гарантированно может справиться только человек, например, точная постановка диагноза пациентам с неоднозначными симптомами и показаниями анализов или оказание экстренной медицинской помощи. Но выполнение рутинных стандартизированных задач разумнее доверить плодам технологических решений.

Существующий уровень технического прогресса несет в себе большой потенциал для создания инновационных прогрессивных способов и методов взаимодействия и функционирования сфер общественной жизни и всего мирового сообщества в целом. Таким образом, быстрые темпы развития всей отрасли информационных технологий и постоянно меняющиеся условия внешней среды можно назвать одними из важнейших факторов, влияющих на формирование трендов современных трансформаций экономики и общества. С течением времени качество жизни людей будет улучшаться за счет постоянного расширения спектра возможностей современных технологий, которые с каждым годом становятся все доступнее.

Литература

1. Зараменских Е.П. Основы бизнес-информатики. М.: Юрайт, 2017. 407 с.
2. Томашевский К.Л. Цифровизация и ее влияние на рынок труда и трудовые отношения (теоретический и сравнительно-правовой аспекты) // Вестник Санкт-Петербургского государственного университета. Право. 2020. № 11(2) С. 398-413. <https://doi.org/10.21638/spbu14.2020.210>
3. Кравченко С.А. Социология цифровизации. М.: Юрайт, 2021. 236 с.
4. Бриньолфсон Э., Макафи Э. Вторая эра машин: работа, прогресс и процветание в эпоху новейших технологий. М.: АСТ, 2016. 384 с.
5. Бондарева Н.Н. Состояние и перспективы развития роботизации: в мире и России // МИР (Модернизация. Инновации. Развитие). 2016. Т.7. № 3(27). С. 49–57. <https://doi.org/10.18184/2079-4665.2016.7.3.49.57>
6. Баррат Дж. Последнее изобретение человечества: Искусственный интеллект и конец эры Homo sapiens. М.: Альпина нон-фикшн, 2015. 304 с.

© Тарасова А.Ю., 2022

АРХИТЕКТУРА WEB-ПРИЛОЖЕНИЙ

21 век – это век информационных технологий. Но большинство людей, которые пользуются компьютерами, смартфонами, планшетами, смарт-часами, смарт-телевизорами все еще не понимают как они устроены и как работают, для некоторых это все является магией, а для кого-то смартфон вообще сравним с человеком, который имеет свой мозг и живет своей жизнью. И сейчас я говорю не про то, как в телефон поступает заряд от батареи, или как процессор обрабатывает информацию. Я говорю о том, что происходит, когда вы пользуетесь приложениями на смартфоне, смарт-телевизоре, или когда на сайте нажимаете какую-нибудь кнопку, а в ответ каким-то чудом получаете результат. И на самом деле все не так сложно для понимания, особенно становится легче если знать, что все эти устройства работают одинаково независимо от того, если приложение это установлено на телефоне, смарт-часах или используется сайт в браузере компьютера. Цель статьи простыми словами, не используя сложных определений рассказать о том: Что такое web-приложение; Из чего состоит web-приложение; О зонах ответственности за каждый компонент web-приложения; И о том что вообще происходит в каждом из этих компонентов.

Приложения на телефоне делятся на две категории: мобильные приложения и web-приложения. Мобильные приложения – это приложения схожи с десктопными приложениями, которые устанавливаются на компьютеры, т. е. это те приложения, которым достаточно один раз установиться на устройство для полной работы. Для них главное версию обновлять раз в полгода и больше им ничего не нужно.

Web-приложения – это клиент-серверные приложения, в которых есть клиент, который использует пользователь и есть сервер, к которому обращается клиент для предоставления пользователю информации [1, с. 144].

Основное отличие – это то, что web-приложению обязательно нужен доступ в интернет, а обычному мобильному приложению не нужен. Например, калькулятор, он не имеет доступа в интернет и может спокойно работать без него, поэтому калькулятор – это не web-приложение. Таким образом, все устройства имеют два типа приложений – это мобильные или десктопные и web-приложения. Верхнеуровневая архитектура приложений в основном всегда делится на 3 части [2, с. 33].

С первой частью сталкивался и взаимодействовал каждый человек, который имел компьютер или смартфон. И называется эта часть – уровень представления (view layout). В определении web-приложений было сказано, что это клиент-серверное приложение. Так вот еще одно понятие уровня представления – это клиент. Уровень представления – это интерфейс, с которым взаимодействует пользователь. Именно в эту часть от пользователя передаются текстовые, аудио и видео данные.

Первые user интерфейсы появились очень давно и до сих пор существуют, например, командная строка (CMD) на компьютерах. User интерфейс (UI) – это еще одно понятие, которое относится к уровню представлению. Консоль, браузер, приложение на телефоне – все это является клиентами, user интерфейсами, уровнем представления. Но мы же не просто так передаем информацию разным видам интерфейсам? Мы же хотим что-то получить взамен. И на самом деле мы и получаем от них взамен тоже информацию. Например, в поисковик браузера вводится запрос, а в ответ браузер возвращает результат, поэтому большинство пользователей считают, что браузер является самым умным. И это заблуждение. В приложениях клиент (тот же браузер) играет роль связующего звена. Пользователь заполняет данными web-форму, браузер собирает эти данные в одну корзинку и передает эту корзинку следующим компонентам приложения. На самом деле уровень представления не простая часть приложений, она проста только для понимания, потому что почти каждый человек взаимодействовал с ней. Но разработка user интерфейсов — это не простое дело, для этой работы придумали даже целое отдельное направление, называется – Front-end.

Второй ключевой элемент верхнеуровневой архитектуры приложений – это уровень бизнес-логики. Именно эта часть отвечает за валидацию, модификацию, трансформацию, анализ, поиск и передачу данных, которыми обмениваются пользователи с интерфейсами.

Например, человек хочет начать пользоваться новой соц. сетью, но для ее использования необходимо зарегистрироваться. Он заполняет данными web-форму, нажимает кнопку зарегистрироваться, браузер просит подтвердить его мейл и все. Но так думает большинство людей, потому что только это и видно им. Но на самом деле это не все.

Во всех user интерфейсах для каждого элемента могут быть заданы правила, после которых они могут начать отправлять запрос от пользователя. Например, такими правилами могут быть: нажать на кнопку, скролить ленту или навести курсор мыши на конкретный элемент. Для лучшего понимания запрос можно сравнить с вопросом. В нашей жизни не каждый человек знает все. Например, задайте кому-нибудь вопрос о химии, так вам не каждый сможет ответить. Поэтому чтобы получить ответ на ваш вопрос необходимо знать конкретного человека, который знает химию. Также и у элементов интерфейса. Правила говорят не только после каких действий делать запрос, но и говорят кому именно направлять данный запрос. Вспомним, что web-приложения – это клиент-серверные приложения. Что такое клиент уже разобрали и запомнили. Но что такое сервер. А сервер – это именно та часть, которая содержит бизнес-логику и к которой направляются запросы. Сервер – это именно тот человек, который знает ответы на вопросы.

Для web-приложений сервер физически представляет из себя обычный компьютер. Многие представляют огромные черные шкафы с множеством проводов. Потому что, если в поисковик ввести слово «сервер», то в результате можно увидеть те самые шкафы. Но на самом деле на каждой полке этого шкафа тоже хранится обычный компьютер, они просто так выглядят, чтобы их можно было компактно хранить. А большое их количество характеризуется тем, что известные web-приложения имеют большое количество пользователей и нужно быстро и часто отвечать на их вопросы, точнее выдавать результат на

запросы пользователей. И один компьютер не сможет разом ответить на тысячу запросов. Поэтому для таких приложений увеличивают вычислительную мощность, т. е. добавляют еще такие же компьютеры. Таким образом получается, что компьютеры и ноутбуки тоже могут быть серверами. Но они таковыми не являются, потому что на компьютерах, которые являются серверами, работают специальные программы. И все эти программы разные, каждая создана под конкретное приложение. Эти программы написаны на определенном языке программирования определенными людьми, а именно Back-end разработчиками.

Таким образом можно сказать, что есть уровень представления, который разрабатывают Front-end разработчики и есть уровень бизнес-логики, который разрабатывают Back-end разработчики.

В итоге полный процесс регистрации в соц. сети будет выглядеть так:

1. Когда пользователь заполнил web-форму и нажал кнопку «Зарегистрироваться».
2. Браузер собирает его данные в корзину и отправляет на сервер.
3. Сервер валидирует переданные данные, например, чтобы указанный номер телефона имел таковой формат и состоял из цифр.
4. Сервер создает у себя информацию о том, что пользователь теперь имеет профиль этого приложения.
5. Сервер отправляет сообщение для подтверждения на мейл.

Сервер возвращает клиенту результат запроса, которое состоит из сообщения «Подтвердите ваш мейл».

6. Браузер отображает, полученное сообщение от сервера пользователю.

Теперь можно сказать, что магический браузер на самом деле не магический и выполняет всего 2 пункта из 7. А основная работа лежит на сервере приложений, т. е. на уровне бизнес-логики. Но что именно понимается под бизнес-логикой, а именно – все. Этот уровень отвечает за весь функционал (логику) приложения, который предоставлен пользователю. Бизнес-логика отвечает:

Во-первых, за все математические вычисления.

Во-вторых, за валидацию, актуальность и целостность данных приложения. Например, чтобы данные о профиле пользователя всегда сохранялись и не терялись, и имели правильный формат, чтобы вместо паспортных данных не хранился адрес проживания.

В-третьих, за интеграцию с другими системами. Например, мы в своей соц. сети хотим отображать погоду за текущую неделю, но у нас нет своей метеостанции, да и строить ее ради одного виджета в приложение это очень затратно. Поэтому мы на нашем сервере делаем интеграцию с Гисметео. И теперь наше приложение будет иметь еще одно правило – пользователь должен зайти в свой профиль соц. сети. После чего будет запускаться следующий процесс:

1. Браузер передает данные о месте нахождения пользователя серверу.
2. Сервер на основе этих данных запрашивает у Гисметео погоду за неделю для этого места. Так как сервер — это компьютер, и он не нуждается в user интерфейсе, то наш сервер делает запрос напрямую к серверу Гисметео.

3. Сервер Гисметео получает наш запрос, начинает его валидировать, также он может, например, переданные ему наши данные преобразовывать в другой формат, потому что ему с другим форматом удобнее работать, он так настроен. Он так же может иметь свои интеграции и много другое. Но наше приложение это не волнует, нам главное получить требуемую информацию.

4. Наш сервер получает ответ от сервера Гисметео. Может тоже начать проверять полученные данные или сразу же вернуть их клиенту.

5. Браузер получает информацию о погоде за текущую неделю и отображает ее пользователю.

В-четвертых, за аутентификацию и авторизацию пользователя в системе приложения.

В-пятых, в целом за правильно настроенную логику работы всего функционала приложения [3, с. 52].

На уровне бизнес-логики можно делать с данными пользователей что угодно, а самое интересное, что никто об этом не узнает, так как для пользователей представлен только уровень представления.

Итого, уровень представления – это браузер, консоль, приложение, то есть интерфейсы для пользователей. Уровень бизнес-логики отвечает за весь функционал приложений и за взаимодействие с другими системами. Но, а как приложение хранит данные о пользователях, о переводах. Может на том же компьютере, где и бизнес-логика находится, а сможет ли этот компьютер обеспечивать безопасность, актуальность, атомарность данных? Много ответственности тогда будет лежать на уровне бизнес-логики, что за собой приведет множество потенциальных угроз и поддерживать эту часть будет очень тяжело. Для решения этой проблемы существует уровень доступа к данным (data access layer). Он отвечает за хранение данных приложения. А физически, это будет другой компьютер (сервер), где будет работать не программа приложения, а на этом сервере будет поднята база данных, и именно она будет хранить в себе все данные приложения.

Базы данных делятся на два типа: Реляционные БД (SQL БД); Не реляционные БД (No SQL БД). А не реляционные БД в свою очередь делятся на еще несколько видов, где каждый решает отдельную задачу:

-БД на основе пар «ключ-значение» - такая база хранит данные как совокупность пар «ключ-значение». Ключ служит уникальным идентификатором. Такие БД хорошо подходят для хранилищ сессий или корзин интернет-магазина;

-Документно-ориентированная БД – важно запомнить, что под документом имеется структура. Такая база никакие файлы не хранит. А хранит она JSON или XML формат данных. Хорошо подходят для хранения данных о каталогах или о профилях;

-Графовые БД – предназначены для хранения взаимосвязей. Такие БД хорошо работают в соц. сетях. Имея данные о людях и взаимосвязях между ними, можно узнать, кто является «друзьями друзей» конкретного человека;

-Колоночные БД – с виду похожи на реляционные базы, только взаимодействие с данными происходит не по строкам, а по колонкам. Хорошо решают задачу, если нам

понадобиться, например, получить список нескольких миллионов наименований продуктов. В реляционной базе пришлось бы проходить по каждой записи. А в колоночной БД можно получить с помощью одной операции. Эти четыре вида не реляционной БД являются основными, но на самом деле этих видов намного больше. Как понять какой тип БД выбрать? Если нет тех проблем, которые решают не реляционные базы, то спокойно можно использовать реляционные БД [4, с. 99].

Таким образом, получается следующая схема верхнеуровневой архитектуры приложений, представленная на рисунке.



Рис. Схема архитектуры приложений

В итоге можно прийти к следующему выводу, что необходимо разбираться в архитектуре приложений хотя бы на верхнем уровне, так как это позволит снять «магическую» маску с user-интерфейсов и лучше понимать, как устроены сами приложения и на что они способны. Это так же поможет людям меньше попадаться на фишинговые атаки. А для тех, кто хочет связать свою жизнь с web-разработкой, поможет проще понять один из первых обязательных паттернов разработки – MVC.

Литература

1. Pollentine В. Appcelerator Titanium Smartphone App Development Cookbook. Packt Publishing Ltd, 2011. 308 с.
2. Горюнова М. П. Архитектурные стили в разработке WEB-приложений и область их применения // Проблемы современной науки и образования. 2017. № 12(94). С. 32–34.
3. Базаревский В. Э. Архитектура мобильного веб-приложения для обработки сигнальных данных // Доклады Белорусского государственного университета информатики и радиоэлектроники. 2013. № 1(71). С. 51–56.
4. Fowler М., Sadalage P.J. NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Pearson Education, 2013. 192 с.

© Торопов В.В., 2022

УДК 004.42

Тухватуллина Э.Э.

Южно-Уральский государственный гуманитарно-педагогический университет
г. Челябинск, Россия

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРИЕМА ЗАЯВОК ДЛЯ ФОРМИРОВАНИЯ РАСПИСАНИЯ УЧЕБНЫХ ЗАНЯТИЙ

Одной из важнейших проблем качественной организации учебного процесса в высшем учебном заведении является задача создания автоматизированного учебного расписания. Правильно и точно составленное расписание обеспечивает равномерную загрузку студенческих групп и профессорско-преподавательского состава. Составление расписания учебных занятий и связанная с этим задача управления аудиторным фондом традиционно являются трудоемким и сложным процессом. Решить данную проблему может помочь автоматизированная система составления расписания. Цель нашей работы: создание программного обеспечения для формирования расписания учебных занятий. Для достижения цели были поставлены следующие задачи: рассмотреть особенности составления расписания занятий в высших учебных заведениях; сформулировать возможности информационной системы «Формирование расписания»; разработать систему приема заявок для формирования расписания учебных занятий; написать руководство пользователя.

Расписание учебных занятий — это документ, определяющий педагогически целесообразную последовательность учебных занятий в образовательном учреждении на каждый день учебной недели и конкретизирующий таким образом учебный план. Расписание обеспечивает оптимальное распределение учебной нагрузки учащихся, сохранение их работоспособности и условий для их разносторонней деятельности. При составлении расписания все дисциплины учебного плана распределяются по дням недели в строгой привязке к аудиторному фонду [1]. Расписание учебных занятий составляется в точном соответствии с утвержденными учебными планами.

Проведем анализ аналогов систем, предназначенных для формирования расписания в высших учебных заведениях. Программа «1С: Автоматизированное составление расписания. Университет» для составления расписания предназначена для решения задач автоматизированного составления учебных расписаний и оперативного управления помещениями в ВУЗах. С его помощью составлять расписание можно в автоматическом, ручном и смешанном режимах с учетом многих ограничений и условий. При этом можно построить как допустимое расписание, так и оптимизированное, в котором сокращено количество окон или количество используемых помещений.

Экспресс-расписание ВУЗ – программа для автоматизации составления расписания учебных занятий в институтах, университетах, академиях. Программа полностью информирует диспетчера об имеющихся нагрузках, занятости выбранного преподавателя, группы, аудитории, остатке невыполненных часов, рассчитывает и рекомендует, сколько необходимо назначать занятий в день с учетом будущего отсутствия преподавателей, чтобы выполнить учебный план в срок.

ХроноГраф 3.0 Мастер – это программа составления учебного расписания, недельных сеток часов. ХроноГраф 3.0 Мастер также помогает формировать списки учащихся и педагогического коллектива. Во многих университетах расписание формируется в ручном режиме, составляется оно на картонных планшетах, мелким почерком карандашом на них наносится сетка расписания. В таком же виде они распространяются по факультетам и кафедрам. Найти нужную информацию или сделать определенную выборку в такой ситуации крайне затруднительно. Поэтому как минимум необходимо автоматизировать ввод и хранение информации о расписании занятий [1].

Программные продукты позволяют достаточно успешно формировать расписание занятий в автоматическом режиме, но при этом теряются опыт и устойчивая привычная структура расписания, полученные при ручном составлении. Кроме того, такие программы чаще всего используют локальный подход, т.е. автоматизацию только одного отдела, ответственного за составление расписания. Сотрудникам этого отдела требуется провести трудоемкий процесс ввода исходной информации в единую базу данных. Разрабатываемая нами информационная система «Формирование расписания» предназначена для решения следующих задач:

1) учёт данных каждого пользователя: фамилия, имя, отчество преподавателя; название группы; название дисциплины; нагрузка преподавателя. Вся входящая информация и готовое расписание хранится в базе данных.

2) защита от несанкционированного доступа к материалам регистрируемого (данные находятся на сервере и к ним есть доступ только у руководителя/администратора).

Система должна работать в однопользовательском режиме, при этом авторизация пользователя предусмотрена, для избежание доступа случайных лиц. Кроме того, в системе должна быть предусмотрена возможность резервного копирования и восстановления данных, а также система должна иметь возможность самовосстановления после сбоев в операционной системе или отключения электропитания.

К виду автоматизируемой деятельности относятся процессы сбора (получения), обработки и санкционированного предоставления (отображения) информации посредством АИС «Формирование расписания», включая:

- 1) создание, редактирование и прочие виды обработки информации;
- 2) организацию хранения информации;
- 3) авторизация пользователя;
- 4) ввод данных;
- 5) возможность повторного создания расписания со внесенными изменениями;

Структурная схема АИС «Формирования расписания» представлена на (рис. 1).

Разработка информационной системы происходила в интегрированной среде разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений (<https://clck.ru/Nj4w9>).

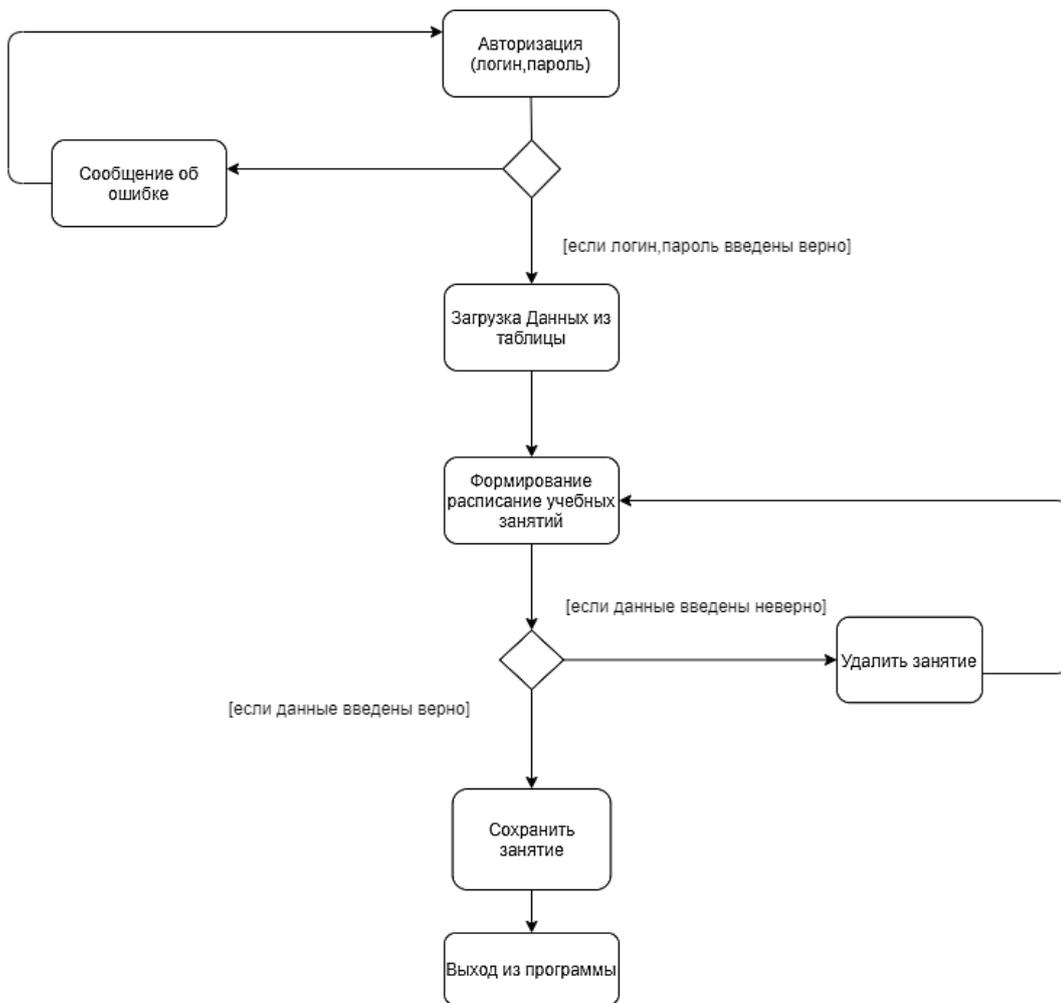


Рис. 1. Структурная схема

После запуска программы администратор вводит логин и пароль. Если логин и пароль введены верно, то открывается окно «Расписание» (рис. 2), в противном случае-выводится сообщение об ошибке.

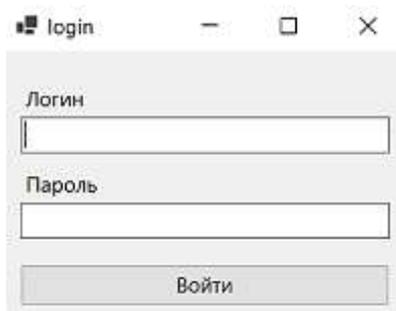


Рис. 2. Авторизация

Загрузка данных происходит из таблиц SQLiteStudio. В блоке «Расписание из ниспадающего списка осуществляется выбор данных для каждого преподавателя (группа, ФИО преподавателя, дата проведения занятий, аудитория). Система дает возможность во всех полях вносить изменения при необходимости (рис. 3).

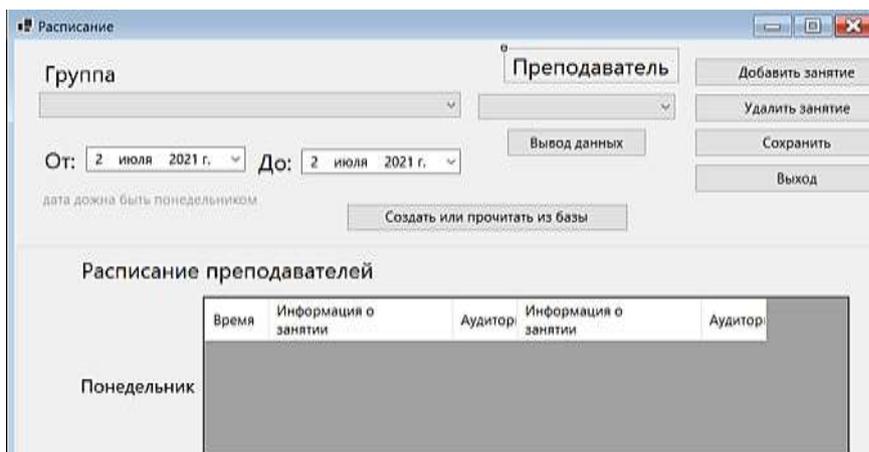


Рис. 3. Заполнение формы «Расписание»

Программа составляет расписание учебных занятий с учетом пожеланий преподавателей. Система не пропускает одновременное проведение занятия преподавателям в одной аудитории и в одно тоже время в определенной группе (выводится сообщение «Время занято»). Сформированное расписание необходимо сохранить в формате изображения. При необходимости (при ошибочном вводе данных) можно внести коррективы в расписание (удалить занятие, заменить преподавателя и т.д.) и снова сохранить. После входа выбираем академическую группу, в которой будет проводиться учебное занятие (рис. 4).

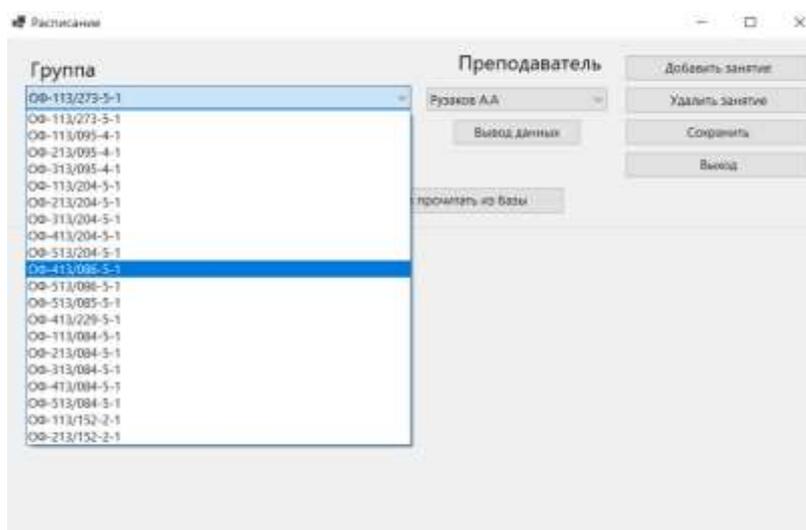


Рис. 4. Выбор группы

Выбираем преподавателя, который будет вести учебное занятие (рис. 5).

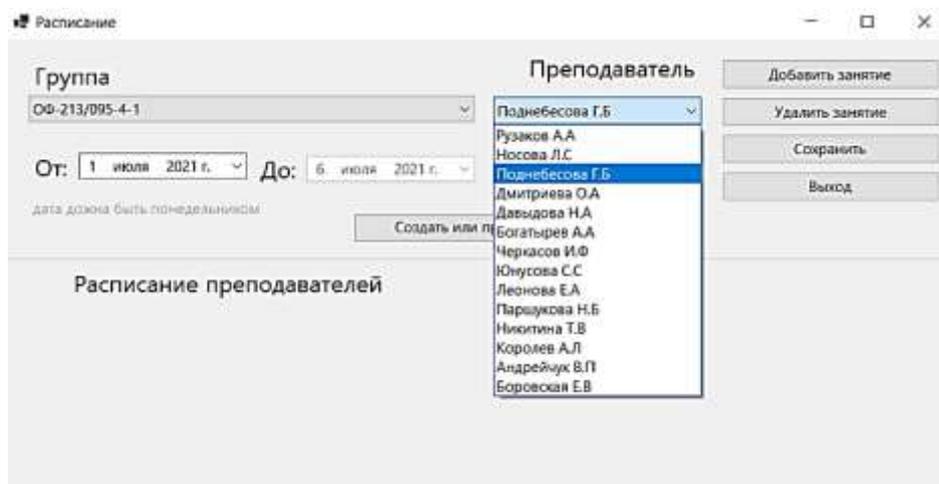


Рис. 5. Выбор преподавателя

Выбираем в календаре дату занятия, срок расписания должен начинаться с понедельника, так как расписание формируется на неделю. Если это условие не соблюдается, выводится сообщение об ошибке. После введение всех данных правильно, открывается основное окно (рис. 6).

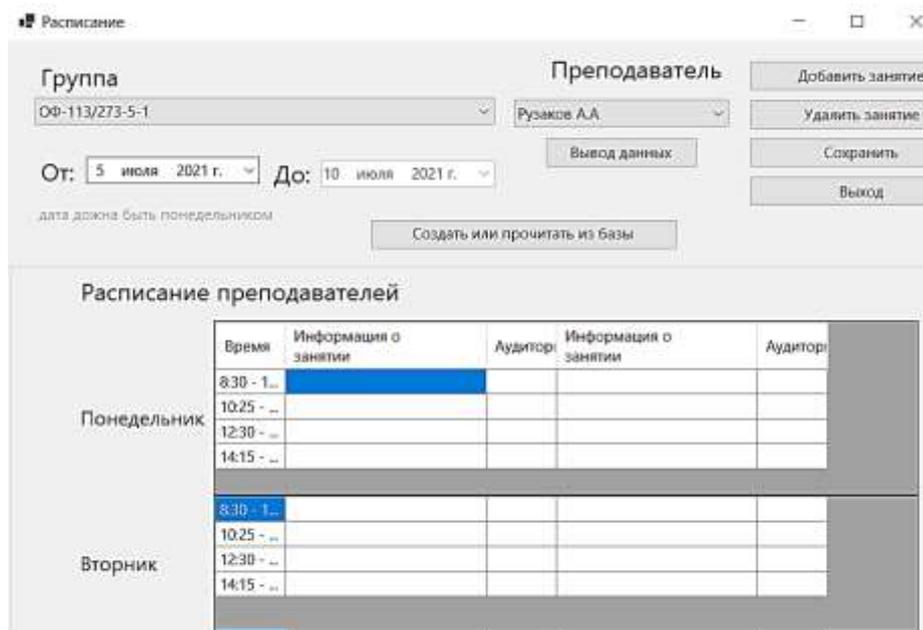


Рис. 6. Окно расписания

Далее можно добавить занятие, удалить или сохранить расписание в виде изображения, чтобы его можно было распечатать или отправить преподавателям. Если допустили ошибку при заполнении данных, то выводится ошибка. Если больше нет необходимости добавлять занятия, то выходим из программы.

Таким образом нами были рассмотрены общие положения при составлении расписания учебных занятий, аналоги систем, предназначенных для формирования расписания в высших учебных заведениях, подходы к автоматизации составления расписания.



Также разработана информационная система для приема заявок для формирования расписания учебных занятий «Формирование расписания» на языке программирования Microsoft Visual Studio. Данная система имеет удобный пользовательский интерфейс, позволяющий легко освоить работу в программе, гибкость же программного кода в случае необходимости позволит удовлетворить растущие требования к системе. Преследуемая изначально цель автоматизации процесса расписания занятий достигнута.

Литература

1. Клеванский Н.Н. Алгоритмы формирования расписания занятий высших учебных заведений // Фундаментальные исследования. 2017. № 10-3. С. 454-458.

© Тухватуллина Э.Э., 2022

УДК 004

Тырышкина С.Ю.
Куйбышевский филиал
Новосибирского государственного педагогического университета
г. Куйбышев, Россия

МЕДИАПРОЕКТЫ КАК СРЕДСТВО ФОРМИРОВАНИЯ ИНФОРМАЦИОННОЙ ГРАМОТНОСТИ ОБУЧАЮЩИХСЯ СОВРЕМЕННОЙ ШКОЛЫ

Важно уметь представлять информацию, чтобы она была достоверной и «цепляющей». Мало овладеть только техническими средствами. Даже самое высококлассное представление ничего не значащего содержания может свести все ваши попытки на «нет». Представление информации является одним из важнейших медиаобразовательных умений наряду с умениями поиска, получения, распространения, обработки и преобразования информации. Во всех образовательных учреждениях важно развивать информационную грамотность посредством медиаобразовательных умений. Актуальность данной реализации имеет значение для всех участников учебного процесса – учащихся и преподавательского состава. Требования к представлению и оформлению с каждым днем меняются и развиваются, как и современные инновационные технологии. Информационную грамотность необходимо включать в образовательный процесс, кроме того, это считается базовым навыком, поэтому ее не надо выделять в отдельный предмет. Решить проблему включения такого навыка можно несколькими способами:

1. Введение медиаобразовательных умений в учебный план предмета «Информатика»;
2. Развитие информационной грамотности может проходить путем включения данного навыка в каждый учебный предмет;
3. Освоение грамотности может реализовываться на внеурочных мероприятиях.

Одним из интересных навыков по представлению графической информации является создание медиапроектов. В своей статье «Медиапроекты как средство формирования информационной грамотности» Иванова Д.С. дает следующее определение: «Медиапроект – проект в сфере деятельности средств массовой информации» [2]. Термин «проект» происходит от латинского слова *projectus*, что в переводе означает «брошенный вперед». Таким образом, можно сказать, что медиапроект — это авторское решение творческой задачи, которое представлено с помощью средств современных информационных и коммуникационных технологий. Реализация медиапроектов поможет развить информационную грамотность и приблизиться к так называемому медиамиру.

Вследствие постоянного развития информационного общества, у школ появляется потребность в расширении медиапространства, например, создание пресс-центров, медиахолдингов и так далее. Создание таких пространств включает в себя не только издание стенных газет или других видов печатных источников распространения информации, но также реализация различных медиапроектов. Рассмотрим несколько вариантов разработки простых медиапроектов.

Самый простой и быстрый способ – это разработка презентации, можно использовать при этом такие программы как PowerPoint, Canva, Prezi и другие.

Второй вариант – это создание сайта. Для специальностей технических профильных направлений это может быть реализовано с помощью языков программирования, таких как, HTML и другие. Если речь идёт о специалистах гуманитарного профиля, то рациональнее будет использовать различные редакторы и сервисы, такие как uKit, Craftum, Wix, Ucoz и другие [2, с. 47].

Третий вариант – создание видеоролика.

В этой статье мы поговорим о представлении информации в качестве видеоролика для повышения информационной грамотности среди обучающихся школ, студентов и преподавателей.

Итак, как же создать видеоролик? Для того, чтобы создать качественный видеоролик не обязательно иметь дорогую аппаратуру и профессиональное образование, реализация продукта может быть выполнена обычным человеком, используя любительскую камеру, микрофон и любой персональный компьютер.

Прежде всего надо подготовить сценарий. Это самый важный, трудоемкий, требующий больших временных затрат, этап. От правильно придуманного и составленного сценария очень многое зависит. Не обязательно продумывать каждый шаг до мелочей, однако, как и в любом фильме, должен быть прописан сюжет. Можно включать импровизацию, но лучше не сочинять на ходу, потому что впоследствии могут возникнуть проблемы.

Следующий этап – съемки. Тоже немаловажный шаг, потому что зрители, включая ваш видеоролик, надеются наблюдать хорошую и качественную картинку. Поэтому важно соблюдать некоторые правила.

1. Правильно держите камеру. Нужно держать камеру двумя руками, а лучше использовать штатив, чтобы избежать дрожания и резких рывков.

2. Следите за кадром и ракурсом. Для удобства можно использовать сетку на камере. Стоит помнить, что смена кадра должна наступать спустя 5 секунд, ведь резкая смена кадров только утомляет зрителя. Не стоит часто использовать кнопку zoom (кнопка для приближения или отдаления изображения во время съемки). Важно снимать фрагментами (общий план, крупный план, съемка деталей и т.д.).

3. Освещение. При естественном освещении не стоит снимать на фоне окна или других светящихся предметов. Нельзя резко переходить из темного освещения в яркое и наоборот. Лучше использовать искусственный свет, чтобы его настраивать под данный кадр [4, с. 99].

Заключительным этапом является редактирование отснятого материала – монтаж. Подготовив материал, вы должны выбрать для себя программу, которая предназначена для монтажа и редактирования видео. Среди наиболее известных можно выделить Windows Movie Maker, Adobe Premiere или Movavi.

Гулевский Н.А. дает определение монтажа видеороликов: «Основная задача видеомонтажа – это удаление участков, стыковка отдельных частей видеофайла, оформление переходов между частями, интеграция спецэффектов и надписей в готовый видеоматериал»

[1]. Следует отметить, что монтаж является не только методом, позволяющим грамотно донести до зрителя мысль, заложенную автором. Монтаж, прежде всего, один из методов создания художественного образа. Существует три вида монтажа: линейный, нелинейный, гибридный. Рассмотрим их по порядку.

Линейный монтаж. Данный метод являет собой перезапись видео с нескольких источников на записывающее устройство, с удалением и добавлением видео частей, а также спецэффектов. Монтаж выполняется в реальном времени, что вносит некоторые неудобства.

Нелинейный монтаж. Этот метод обработки видео можно описать как цифровая сборка видеоряда. Данный метод подразумевает работу с видео в любом положении и итерациях. Поскольку современные программы для монтажа предоставляют мгновенный доступ к видеоряду. Данный монтаж является более эргономичным поскольку основан на банальном методе «перетаскивание». То есть берется фрагмент видео и переносится в любую часть видео, то же самое происходит и со спецэффектами и наложением текста.

Гибридный монтаж. Данный метод впитал с все положительные и отрицательные стороны обоих методов. Однако сверх этого возникает проблема повышенной сложности обработки видео поскольку монтаж должен происходить в реальном времени для трансляции [1, с. 1280]. Но у каждого вида монтажа есть свои плюсы и минусы (табл.).

Таблица

Преимущества и недостатки видов монтажа

Критерий	Монтаж		
	Линейный	Нелинейный	Гибридный
Мгновенный доступ к любой части видеотрекка	Нет	Да	Да
Возможность редактирования видеотрека	Да	Да	Да
Возможность создавать новые части видеотрека	Нет	Да	Да
Возможность корректировки качества видео	Нет	Нет	Нет
Получение информации о видеотреке и его свойствах	Нет	Да	Да

Выполним монтаж видеотрека на примере социальной рекламы в программе Movavi Video Editor Plus.

1. Для начала отснятый материал (весь или его часть) или другие медиафайлы загружаются на «монтажный стол», а затем производится «склеивание» фрагментов в нужном порядке (рис. 1).

Монтажный стол – это элемент программы видеоредактора в виде полосы (ленты), на котором визуально расположены все видео- и звуковые дорожки и где производится сам монтаж видеоряда.



Рис. 1. Подготовка материала и их склейка

2. Чтобы смена картинки при переходе от одного кадра к другому не выглядела слишком резкой, нужно добавить плавные переходы между фрагментами (рис. 2).



Рис. 2. Добавление переходов между фрагментам

3. Наложим музыку, эффекты и титры для динамичного и эффектного видеоролика. Также можно создать интро и аутро. Интро – это сокращение от Introduction, которое переводится, как вступление. Интро – это короткая заставка к видеоролику, которая готовит зрителя к просмотру материала. Английское слово «outro» образовано по аналогии с «intro». [3, с. 37] Аутро – завершающая часть видеоролика. Данные заставки можно сделать с помощью различных сервисов таких как Panzoid, Renderforest, Canva и др. (рис. 3).



Рис. 3. Накладывание музыки, титров, эффектов

4. Сохраняем видеоролик.

Таким образом, школьники, проводя большое количество времени в Интернете, сталкиваются с рядом проблем. Способом решения этих проблем является повышение их медиаграмотности. В этом поможет школа, вводя уроки информационной грамотности. Каждый ученик может иметь возможность реализовать свои навыки и умения посредством создания видеороликов.

Литература

1. Гулевский Н.А. Методы обработки видеофайлов для проведения анализа // Инновации. Наука. Образование. 2021. № 32. С. 1280-1284.
2. Иванова Д.С. Медиапроекты как средство формирования информационной грамотности // Информатика и прикладная математика. 2019. № 25. С. 46-48.
3. Лихошва, И. С. Анализ терминологии в сфере аудио и видео монтажа // Молодой ученый, 2016. С. 35-38.
4. Якушина Е.В. Представление информации как одно из важнейших медиобразовательных умений. Создаём видеопоздравление // Медиа. Информация. Коммуникация. 2015. № 12. С. 93-104.

© Тырышкина С.Ю., 2022

РАЗРАБОТКА СЕРВЕРНОГО МОДУЛЯ ИОТ-ПЛАТФОРМЫ НА ОСНОВЕ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПОДХОДА

В своём современном состоянии Интернет вещей уже стал неотъемлемой частью жизни множества людей. Благодаря появлению беспроводных сетей, постоянному росту пропускного объема интернет-соединения и внедрению новых подключенных устройств человек окружил себя сетевой инфраструктурой, помогающей ему и решающей задачи, которые до этого приходилось решать самостоятельно. Согласно множеству отчетов и прогнозов, подключенных устройств будет становиться всё больше (<https://clck.ru/et9u8>). Вместе с этим будет расширяться и концепция «Интернета вещей». Уже сейчас, спустя 15 лет после рождения Internet of Things (IoT), Интернет вещей стал одним из главных трендов высоких технологий [1]. Однако несмотря на стремительный рост Интернета вещей за последние несколько лет, процесс разработки в данной области все еще остается трудоемким. Многообразие компонентов, библиотек для работы с ними и настройка сетевого окружения являются одними из основных трудностей, с которыми сталкиваются пользователи при создании автоматизированных систем. Производители готовых девайсов предоставляют возможности для конфигурации и объединения своих устройств в единую систему в мобильном приложении для создания собственных сценариев автоматизации, однако проблема данного подхода заключается в несовместимости продуктов от разных производителей, а также невозможность подключить самодельные устройства [2]. Для снижения порога вхождения в область Интернета вещей разрабатывается IoT-платформа, которая позволит создавать автоматизированные системы на основе событийно-ориентированного подхода. Целью данной работы является разработка одного из компонентов IoT-платформы, а именно серверного модуля. В ходе работы был проведен анализ предметной области, разработаны архитектура модуля и интерфейс для взаимодействия с клиентской частью платформы, а также реализован механизм генерации прошивки устройства.

Создание сценариев автоматизации в области Интернета вещей в общем случае сводится к описанию поведения системы при наступлении определенного события. События можно разделить на две группы. К первой группе относятся события, связанные с датчиками. Это может быть обнаружение движения, изменение температуры или уровня освещенности в помещении. Ко второй группе относятся события, которые возникают по прошествии какого-либо заданного промежутка времени или согласно определенному расписанию. На самом высоком уровне не требуется знать о внутреннем устройстве датчика, каким образом значение с датчика будет прочитано и отправлено в систему, с помощью какой библиотеки это можно реализовать. Все, что требуется в данной ситуации, - это правильно описать

реакцию системы на то или иное событие, которое возникло внутри системы или пришло извне.

Однако процесс разработки на данный момент устроен сложнее. В первую очередь устройство необходимо собрать: соединить между собой датчики, микроконтроллер и остальные компоненты, например, устройство вывода информации. Вторым шагом является имплементация опроса датчиков. Под этим действием подразумевается поиск необходимых библиотек, изучение и использование их Application Programming Interface (API) для считывания и обновления данных датчика. Следующим шагом необходимо настроить взаимодействие устройства с внешним миром. На сегодняшний день в проектах, направленных на создание решений для Интернета вещей, основным протоколом передачи данных является протокол Message Queuing Telemetry Transport (MQTT) (<https://mqtt.org/>). MQTT - это протокол обмена сообщениями по шаблону издатель-подписчик. С самого начала протокол MQTT разрабатывался для потоковой передачи данных между устройствами с ограниченной мощностью CPU и/или временем автономной работы, а также для сетей с дорогой или низкой пропускной способностью, нестабильностью или высокой задержкой. Именно поэтому MQTT является идеальным транспортом для Интернета вещей. Обмен сообщениями происходит через центральный сервер, называемый брокером. Брокер ответственен за получение всех сообщений, фильтрацию сообщений, принятие решений о том, кто в них заинтересован, и отправку этих сообщений всем подписанным клиентам (<https://clck.ru/et9vH>). Для функционирования даже самой простой системы необходимо создавать MQTT-сервер либо подключаться к уже существующему. Таким образом, третий шаг включает в себя подключение микроконтроллера к домашней Wi-Fi сети и MQTT-брокеру. После того, как пользователь «научил» свое устройство опрашивать датчики и взаимодействовать с внешним миром, остается описать логику работы устройства, т.е. что устройство должно выполнять. Последним шагом является загрузка пользовательского кода на устройство.

Данное описание является поверхностным и, безусловно, подойдет не для всех автоматизированных систем, однако в общем случае пользователь совершает именно эти действия. Как можно заметить из описания, перед тем, как приступить к программированию логики работы устройства, пользователь занимается поиском и изучением библиотек компонентов, настраивает собственный сервер сообщений. Все это является барьером, который порой не может преодолеть и опытный разработчик, не говоря уже о тех, кто не знаком с программированием вовсе. Концепция разрабатываемой платформы состоит в том, чтобы минимизировать или убрать вовсе трудности этапов 2 и 3 и сконцентрировать внимание пользователя на пункте 4 – описании логики работы устройства. В первой версии IoT-платформы создаются устройства на базе микроконтроллера ESP8266 (<https://clck.ru/et9w2>). ESP8266 является одной из самых популярных в сфере IoT, так как несмотря на невысокую стоимость и энергопотребление, плата позволяет организовать связь по Wi-Fi, в том числе с возможностью загрузки новой версии прошивки как напрямую через USB, так и по беспроводному соединению с использованием технологии Over the air (OTA).

Для IoT-платформы была разработана следующая архитектура (рис.). На стороне клиента (соответствует блоку Frontend) в редакторе пользователь задает конфигурацию своего устройства - какие датчики подключены, каким выводам на плате соответствуют; подключает устройство к платформе и описывает логику работы устройства. Затем код пользователя отправляется на сервер (блок Backend). На основе данного кода создается прошивка, которое затем загружается на микроконтроллер. Для работы с датчиками разработана библиотека компонентов, позволяющая пользователям считывать значения и управлять датчиками на более высоком уровне, подключение необходимых низкоуровневых библиотек инкапсулировано в серверном модуле платформы. Для разработки серверной части был выбран язык C++, поскольку он является основным в области Интернета вещей и робототехники. Также реализация сервера на C++ позволяет повторно использовать существующие библиотеки как для работы с Wi-Fi и брокером сообщений, так и для работы с датчиками.

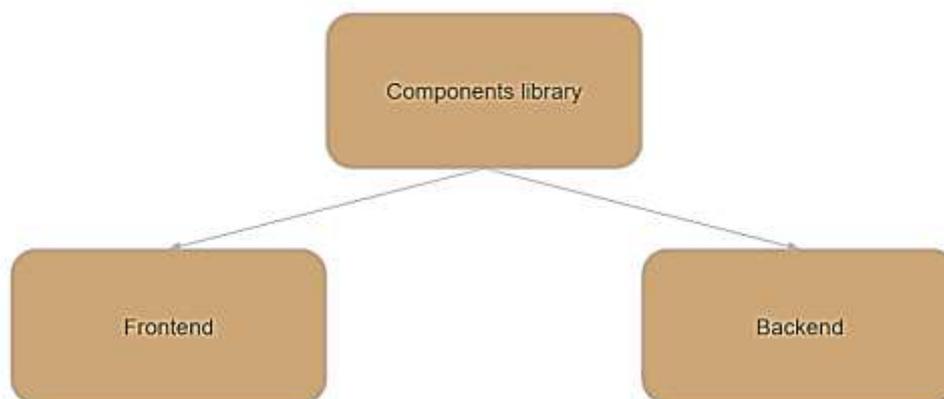


Рис. Архитектура IoT-платформы

Рассмотрим более подробно компоненты серверного модуля IoT-платформы. В ходе процесса разработки автоматизированных систем работа с сетью является неотъемлемой частью программного кода. К таким действиям относятся подключение микроконтроллера к Wi-Fi сети, брокеру сообщений, обработка нестандартных ситуаций: потеря соединения, установка повторного подключения и т.д. В общем случае данный набор операций не претерпевает никаких изменений и зависит исключительно от данных сети и брокера, к которым происходит подключение. В IoT-мире появляются библиотеки, в том числе и от самих производителей микроконтроллеров, предоставляющие API для простого взаимодействия с Wi-Fi и брокером, однако при этом не решается вопрос создания и настройки брокера и перенос кода из одного проекта в другой. Для серверного модуля был разработан компонент для взаимодействия платы с внешним миром. На вход подается только пользовательская информация для подключения к сети Wi-Fi. Брокер сообщений находится на сервере и в первой версии платформы взаимодействие с ним будет реализовано с настройками по умолчанию, поэтому пользователю не придется заниматься изучением установки сервера в домашних условиях и разбираться в тонкостях протокола MQTT. В будущем планируется предоставить пользователю возможность подключаться к

собственному серверу сообщений, передавая адрес и порт. Таким образом, пользователь, передавая данные сети для подключения своего устройства, больше не совершает никаких дополнительных действий, все происходит внутри системы. Третий этап из описанного ранее процесса разработки при работе с IoT-платформой происходит без пользовательского участия.

В мировом сообществе IoT уже давно обратили внимание на проблему большого количества библиотек компонентов и отсутствие стандартов при разработке данных библиотек. Библиотека может быть создана не только производителем датчика. Автором может стать и рядовой разработчик Интернета вещей, которому предложенное производителем решение показалось неоптимальным. В сообществе Интернета вещей подобная практика только приветствуется и находящиеся в свободном доступе библиотеки могут оказаться популярнее так называемых официальных библиотек. Отсюда вытекает вторая озвученная в начале проблема – отсутствие стандартов. Стиль написания кода и/или уровень компетенций отдельно взятых разработчиков может сильно отличаться, что впоследствии сказывается как на внутреннем устройстве кода, так и на публичных интерфейсах, которые предоставляет библиотека. Все это ведет к проблеме для пользователя при поиске библиотек для собственного набора компонентов и необходимости отличать «хорошую» библиотеку от «плохой».

Данная проблема не является чем-то новым в мире Интернета вещей и различные системы пытаются внедрять собственные техники, которые направлены на решение данной проблемы. Основная идея существующих решений основана на том, чтобы создать так называемую «обертку» над датчиком, которая внутри себя содержит необходимые низкоуровневые библиотеки и предоставляет более высокоуровневый интерфейс для работы с датчиком. В основе библиотеки компонентов разрабатываемой IoT-платформы лежит данный подход. На данный момент времени создана поддержка нескольких популярных датчиков для ESP8266. Для микроконтроллера ESP8266 примером подхода классов-обертки является система ESPHome (<https://esphome.io/>). Данная система с открытым исходным кодом позволяет создавать сценарии автоматизации внутри одного конфигурационного файла на языке YAML, концептуально близкий к языкам разметки. В исходном коде ESPHome для каждого поддерживаемого системой датчика реализован класс-обертка на языке C++ и на сегодняшний день ESPHome поддерживает более 500 датчиков различных производителей, поэтому в будущем требуется обеспечить совместимость разрабатываемой IoT-платформы с классами ESPHome, поскольку самостоятельная разработка и поддержка библиотеки компонентов займет очень много сил и времени.

В первой версии серверной части платформы осуществляется простая генерация файла прошивки. Используются модули, описанные в предыдущих разделах, помимо них создаются другие объекты. Для генерации файла прошивки в платформе определены три основные сущности – Sensors, Timers и State. В редакторе пользователю эта информация будет доступна в соответствующих разделах, где он сможет добавлять информацию о подключенных датчиках, редактировать таймеры и создавать новые переменные в State.

Sensors содержит в себе информацию о датчиках, указанные пользователем. Например, если к микроконтроллеру подключен датчик BME280 (<https://clck.ru/et9zC>), пользователь добавляет в Sensors переменную bme с типом BME280. После этого в коде пользователь может обращаться к датчику через конструкцию `Sensors.bme.<название метода>`. Информация о доступных методах текущего датчика запрашивается с сервера отдельным запросом.

Системные события можно разделить на две категории: события датчиков и события по времени. Для событий по времени определена абстракция `Timer`, которая доступна пользователю в редакторе. Пользователь может установить расписание, по которому таймер будет выполнять определенное действие, которое описывается в функции таймера. Внутри этой функции пользователь описывает логику работы своего устройства, что должно произойти по прошествии времени. Например, считать значение с датчика и вывести его на дисплей.

Сущность `State` содержит набор переменных, использующиеся внутри программы. При этом переменная может либо `local`, либо `shared`. Если переменная имеет тип `shared`, то при обновлении ее значения создается событие об изменении данной переменной, новое значение отправляется в брокер сообщений. На такие события могут подписываться другие устройства, для этого необходимо просто читать значения из переменных, имеющие область видимости `shared`. Если переменная имеет тип `local`, то изменение значения такой переменной не порождает никаких событий.

С клиентской части в формате JSON на сервер передаются сведения о `Sensors`, `Timers` и `State`. На основе данной информации происходит генерация файла прошивки.

Результатом работы является модуль, который позволяет на основе пользовательской информации создавать прошивку умного устройства. В будущем планируется провести доработку библиотеки компонентов, чтобы увеличить количество поддерживаемых датчиков внутри платформы. Полученные результаты планируется интегрировать в разрабатываемую IoT-платформу.

Литература

1. Khan R., Khan S.U., Zaheer R., Khan S. Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges // 2012 10th International Conference on Frontiers of Information Technology (FIT): Proceedings. 2012. P. 257-260. <https://doi.org/10.1109/FIT.2012.53>
2. Yue Z., Sun W., Li P., Rehman M.U., Yang X. Internet of things: Architecture, technology and key problems in implementation // 8th International Congress on Image and Signal Processing (CISP). 2015. P. 1298-1302. <https://doi.org/10.1109/CISP.2015.7408082>

© Фархутдинов Д.Д., 2022

РАЗРАБОТКА МЕТОДА ОТСЛЕЖИВАНИЯ ЗАВИСИМОСТЕЙ ДЛЯ КЭША ВЫВОДА ТИПОВ СТАТИЧЕСКОГО АНАЛИЗАТОРА КОДА СРЕДЫ РАЗРАБОТКИ PYCHARM

В настоящее время для создания программ на различных языках программирования повсеместно используются интегрированные среды разработки (IDE, Integrated Development Environment). Среда разработки выявляет ошибки в коде, позволяет производить рефакторинг, автодополнение и другие действия на основе информации, полученной после обработки текста программы. Данный процесс анализа программы вне ее выполнения называется статическим анализом. Возможности, которые предоставляет анализ кода в средах разработки, позволяют перенести фокус внимания пользователя с поддержания корректного синтаксиса программы на ее семантику.

PyCharm – среда разработки на основе платформы IntelliJ, предназначенная для программирования на языке Python. IntelliJ Platform — это платформа для создания IDE, а также инструментов разработчика, адаптированных под особенности различных языков программирования. Платформа IntelliJ представляет собой кроссплатформенное приложение на основе JVM с возможностью построения окон, всплывающих меню и диалогов. С точки зрения функциональности, платформа предоставляет инфраструктуру, необходимую для поддержки разнообразных языковых инструментов (<https://clck.ru/dYUqv>).

Современные среды разработки, в том числе и PyCharm, предполагают поддержку большинства новых возможностей языка и быстрый отклик подсветки кода на изменения пользователя. Однако с появлением новых стандартов языка, для них необходимо разрабатывать и новую функциональность в статическом анализаторе кода. В основном это новые варианты дополнения кода и инспекции – проверки, которые находят разного рода ошибки (рис. 1).

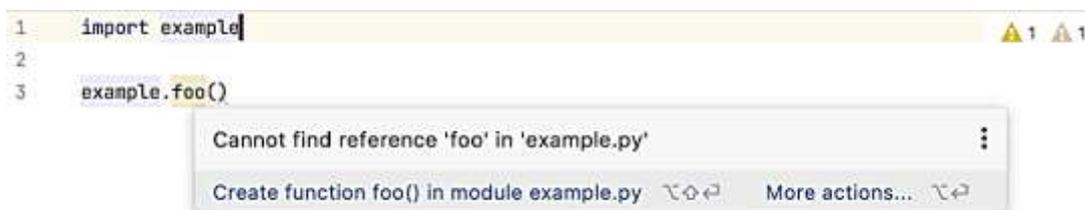


Рис. 1. Подсветка ошибки разрешения имени в PyCharm

Как результат, в данный момент в PyCharm существует уже более сотни инспекций. Растущее количество операций значительно замедляет анализ и является причиной высокой нагрузки на процессор. Данный эффект особенно заметен на файлах размером от нескольких сотен строк. Из-за того, что после внесения изменения в файлы происходит проверка кода

всеми инспекциями, на определенном объеме кода это влечет за собой задержку подсветки на несколько секунд. Таким образом, работа пользователя в IDE замедляется. Более того, в течение некоторого времени после изменения кода можно наблюдать несогласованность подсветки и обновлённого текста программы.

Для выявления причины несогласованности был проведен анализ существующего алгоритма работы статического анализатора кода и механизма кэширования в нем. Было установлено, что выполнение статического анализа в PyCharm подразумевает многократное использование результатов разрешения имен и вывода типов (рис. 2).

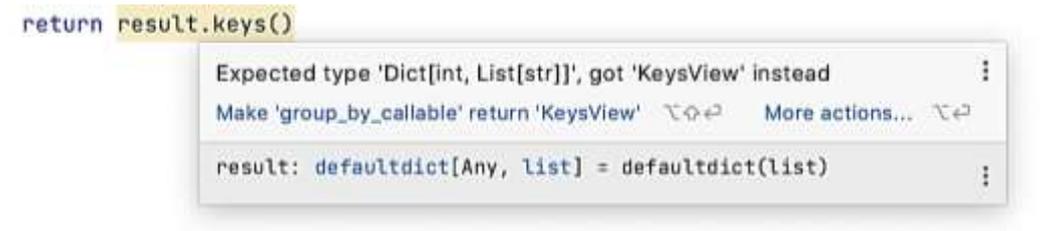


Рис. 2. Подсветка предупреждения об ошибке в коде как результат вывода типов в PyCharm

Во избежание их пересчета в рамках одного прохождения инспекций, то есть между двумя изменениями в коде, применяется кэширование. Данные сохраняются в кэш разрешения имен и кэш выводов типа соответственно (рис. 3). Основной недостаток такого подхода – инвалидация кэшей при любых изменениях в программе, что обеспечивает консистентность данных в анализаторе, но в то же время делает кэширование крайне неустойчивым и, как следствие, малоэффективным. В то же время, небольшие изменения в коде, как правило, не могут повлиять на большинство значений в кэше. Причиной инвалидации всех данных является отсутствие индикатора, отражающего связь конкретного изменения в коде со значением в кэше.

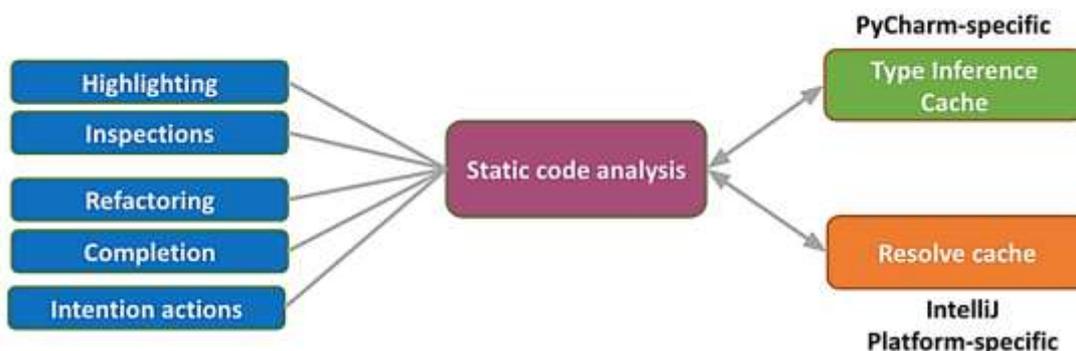


Рис. 3 Использование кэшей вывода типов и разрешения имен для функционирования различных частей среды разработки

Таким образом, задача сводится к разработке метода отслеживания зависимостей между данными в кэше анализатора и изменениями в коде. Метод поможет ускорить работу статического анализатора кода в среде разработки PyCharm. Это также позволит

минимизировать создавшуюся несогласованность и снизит нагрузку на процессор при использовании IDE. Особое внимание отведено увеличению скорости прохождения инспекций после локализованных изменений в коде программы.

Для оценки потенциала применения метода отслеживания зависимостей был поставлен ряд экспериментов и проведено профилирование. Была собрана статистика наиболее распространенных действий пользователя в PyCharm. Затем было записано несколько снимков данных профилирования среды разработки в течение 20 минут с применением наиболее распространенных действий. Важно отметить, что из результатов были исключено время, относящееся к платформенным операциям, т. е. рассматривались операции, специфичные для среды разработки PyCharm и ее статического анализатора. Результаты профилирования показали следующее:

- вывод типов занимает около 50% времени выполнения кода для PyCharm;
- разрешение имен занимает около 30% времени выполнения кода для PyCharm;
- вывод типов и разрешение имен вызывают друг друга во время выполнения статического анализа;

Таким образом, в наилучшем случае, имея 100% кэш-попаданий, мы можем сократить до 80% времени выполнения кода статического анализатора в PyCharm.

Были рассмотрены существующие подходы к отслеживанию зависимостей между данными, в частности, с использованием инкрементальных вычислений. Один из подходов состоит в том, чтобы написать алгоритм с нуля и затем превратить его в инкрементальный. Одной из библиотек, реализующей такой подход, является salsa-rc, механизм инкрементальных вычислений, используемый внутри анализатора языка Rust (<https://clck.ru/dYUtE>). Для использования библиотеки весь код нужно организовать в запросы — чистые функции, преобразующие «параметр» в «результат». Запрос может вызывать другие запросы во время своего выполнения и кэшировать результаты вычислений. Библиотека позволяет изменять части кода, называемые «входами». «Входы» также являются запросами. При изменении «входов» вызываются запросы, зависящие от этих «входов» и далее транзитивно. Важным элементом является сохранение зависимостей при выполнении запроса. Например, если запрос А вызывает запрос В, то В будет зависеть от А. Таким образом, получаем граф зависимостей для программы, где узлами являются «входы» и запросы. Слабая сторона такого подхода в том, что весь код алгоритма, а в нашем случае статического анализатора, должен быть написан на специальном языке. Кроме того, на построение графа зависимостей для всех выражений в языке потенциально может быть затрачено время, сопоставимое с выводом типа. Из-за таких ограничений было решено отказаться от данного варианта инкрементального подхода.

В дополнение был рассмотрен подход с применением статических «срезов» программ (program slice), впервые введенный Марком Вейзером в 1984 году [2, с. 352]. С формальной точки зрения, определение среза производится на основе понятия критерия среза. Согласно этой концепции, критерием среза является пара $\langle p, V \rangle$, где p — это определенная точка программы, а V — подмножество переменных. В последующих работах определяются два

вида срезов. Forward slice – набор операторов программы и предикатов, на которые влияет вычисление значения переменной V в точке программы p . Backward slice – набор операторов программы и предикатов, которые прямо или косвенно влияют на расчет переменной V в точке p [1, с. 30]. Как и предыдущем подходе, на основе данных понятий происходит построение графа зависимостей программы. Узлы графа представляют собой операторы и предикаты, а ребра несут информацию о передаче управления и зависимостей по данным.

После рассмотрения существующих подходов и сопоставления их с архитектурой анализатора был сделан вывод, что построение графов зависимостей для выражений или даже функций программы влечет за собой слишком много расходов, так как это должно происходить каждый раз при изменении структуры исходного кода. Однако, можно взять за основу понятие «среза» и рассмотреть подмножества переменных, на которые влияет не вычисление, а изменение в определенной точке программы. В качестве кандидата для такого «среза» при изменении локальных переменных в языке Python выступает функция. На этой основе был разработан подход, позволяющий ассоциировать значения кэша вывода типов с функциями программы. В разработанном методе кэш вывода типов разделяется на две части, которые заполняются и инвалидируются независимо:

- глобальный кэш;
- кэш функций, где для каждой функции существует локальное хранилище значений.

Основные вопросы, которые необходимо рассмотреть при создании данного разделения:

- Условия попадания значения в один из разделов кэша;
- Условия инвалидации значений в разделах.

Рассмотрим необходимые условия, при выполнении которых значение типа для определенного элемента (выражения языка) распределяется в локальный кэш функции.

1. Элемент, для которого вычисляется тип, находится в теле полностью аннотированной функции. Статический анализатор в PyCharm при вычислении типа основывается прежде всего на stub-файлах и аннотациях в них. Если такой файл отсутствует, анализатор пытается обратиться непосредственно к аннотациям в коде программы. Таким образом, только при отсутствии аннотаций происходит полноценное вычисление типа. Кроме того, результаты опроса программистов на языке Python (Python Developers Survey 2020) (<https://clck.ru/Y8gVf>) показывают, что более 70% пользователей периодически используют хотя бы частичные аннотации типа (type hints). Взяв за основу эти данные и существующие погрешности, можно предположить, что более половины объема кодовой базы на Python аннотировано. Следовательно, элементов, находящихся внутри полностью аннотированных функций достаточно, чтобы сделать оптимизацию, основанную на этом факторе и получить значительный прирост по скорости анализа.

2. При выполнении первого условия элемент лежит в теле функции, внутри которой нет переменных с модификаторами `global` и `nonlocal`. Эти модификаторы изменяют связывание глобальных переменных, чем значительно расширяют варианты возможных сценариев написания программ. В то же время, использование данных модификаторов является скорее

частным случаем, чем повсеместной практикой, поэтому это условие не отбрасывает большой процент существующей кодовой базы на Python.

Все остальные типы для выражений из функций, не удовлетворяющих данным условиям, а также элементы в глобальной области видимости распределяются в глобальный кэш.

Для рассмотрения вариантов инвалидации кэшей разделим виды модификации программы на две категории:

1. Модификация внутри тела функции, удовлетворяющей двум условиям выше (полностью аннотированная функция, отсутствуют `global` и `nonlocal` переменные). При этом мы можем сохранить большинство значений кэша вывода типов: глобальный кэш и локальные кэши других функций. Инвалидируем только локальный кэш измененной функции. Из-за того, что функция полностью аннотирована, а ее аннотации не изменились, то все ее типы вовне, в вызовах и ссылках на возвращаемое значения, остаются прежними, т.к. выводятся из аннотаций.

2. Модификация любой другой функции, класса, глобальной переменной. В этом случае придется, как и раньше, сбросить весь кэш: и для локальных функций, и глобальный. Это связано с тем, что даже в полностью аннотированных функциях могут использоваться функции, классы, поля, значения типов которых лежат в глобальном кэше. При потенциальном изменении типов из глобального кэша придется поддержать консистентность и жертвовать частью вычисленных данных.

Принимая во внимание, что большинство изменений программ в профессиональной среде — это поддержка существующей кодовой базы функций, а также факт, что многие функции аннотированы, можно предположить, что максимальное потенциальное ускорение лежит в районе 60-70% (по результатам опроса разработчиков) от 80% возможных (по результатам профилирования). В результате имеем максимальный потенциальный прирост в скорости анализатора ~50% за счет использования не самого сложного метода отслеживания зависимостей без постоянного поддержания графа зависимостей для всех выражений.

В следующих работах планируется рассмотреть различные варианты реализации описанного метода на базе платформы IntelliJ и кодовой базы PyCharm, а также представить детальную апробацию, варьируя такие параметры, как размер файла с исходным кодом, количество аннотаций типов, количество используемых типов. В качестве метрик планируется использовать:

- время анализа для начального запуска с незаполненным кэшем;
- время обновления состояния IDE после локализованного изменения;
- потребляемая IDE оперативная память.

В рамках данной работы изложены результаты анализа статического анализатора кода в среде разработки PyCharm. Выявлены слабые стороны в работе анализатора и в существующем методе кэширования. Изучены методики отслеживания зависимостей, повышающие эффективность кэширования. На их основе разработан базовый метод отслеживания зависимостей, который позволяет сохранить значительную часть данных в

кэше между локализованными изменениями пользователя без потери консистентности. Как следствие, это снизит избыточную нагрузку на процессор и сделает программирование в среде разработки более быстрым и комфортным для пользователей.

В целом, на базе представленного метода есть большой потенциал для разного рода оптимизаций и более глубокой проработки зависимостей. Однако, рассматривая различные варианты модификаций метода, стоит помнить о компромиссе между глубиной анализа и экономией ресурсов. Из-за того, что типов в среднестатистическом файле тысячи, а изменения в текст программы вносятся каждую секунду, цена одной лишней проверки остается довольно весомой.

Литература

1. Horwitz S., Reps T., Binkley D. Interprocedural slicing using dependence graphs // ACM Transactions on Programming Languages and Systems (TOPLAS). 1990. V. 12. № 1. P. 26-60.
2. Weiser M. Program slicing // IEEE Transactions on software engineering. 1984. №4. P. 352-357. <https://doi.org/10.1109/TSE.1984.5010248>

© Федяева И.А., 2022

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВИЗУАЛИЗАЦИИ ПУТЕШЕСТВИЙ

В современном мире в путешествие в различные места отправляется множество людей, во время и после путешествий остается память, которую хочется сохранить и поделиться. Визуализация путешествий позволит показать те места, в которых путешествовали, включая фотографии и мысли об этом месте.

Картографические сервисы – это сервисы, которые содержат карты местности, данные об организациях и местах. Во всём мире таких сервисов очень много. Преимущество таких сервисов в том, что на них можно разместить сколько угодно объектов с их подробными описаниями. Картографические сервисы предоставляют другим пользователям созданную вами карту, находящуюся на сервере. Картографические сервисы спроектированы таким образом, что могут работать с большим количеством сценариев в Интернете и интранете. Прорисовка динамических карт осуществляется в тот момент, когда на них поступает запрос от пользователя. У них низкое быстродействие, но они подходят для высокопроизводительных интранет-приложений и приложений, для которых необходимо отображение данных «в реальном времени». Одними из таких картографических сервисов являются Google Maps, Яндекс Карты и 2ГИС.

Существуют различные приложения для визуализации путешествий, одним из таких приложений является «Veeen». Veeen (рус. «побывал» или «где я был») – это приложение, которое визуализирует на карте мира страны, которые пользователь успел посетить. Функционирует все довольно просто. В приложении надо выбрать одну из частей света, а в открывшемся списке государств отметить нужные страны. После этого они выделяются так, как если бы их закрасили оранжевым цветом на контурных картах (<https://clck.ru/epwGy>).

«Достижения» также выражаются в процентах. Этот показатель выводится для каждой части света отдельно. Посетить один заграничный город, конечно, не значит основательно изучить всю страну. Но для формальной отметки этого достаточно. Особенно, если вы, к примеру, задались целью объехать все страны как можно раньше. Поэтому отмечать отдельные города в veeen нельзя. Небольшая привилегия есть на карте США. Там пользователи приложения могут отдельно отмечать штаты.

Наглядная карта поможет сформулировать желание относительно следующих поездок, обратить внимание пользователя на незаметные ранее страны и помочь ему снова отправиться в незабываемое путешествие.

В процессе написания статьи изучены особенности и возможности различных картографических сервисов, рассмотрены существующие программные решения. В результате проведённого анализа определены инструментарий и среда разработки для создания программного обеспечения.

При рассмотрении особенностей и возможностей вышеописанных картографических сервисов, было принято решение разработать программное обеспечение для визуализации путешествий с использованием картографического сервиса Google Maps. Программное обеспечение будет проектироваться и работать на основании подготовленной UML-диаграммы (рис. 1).

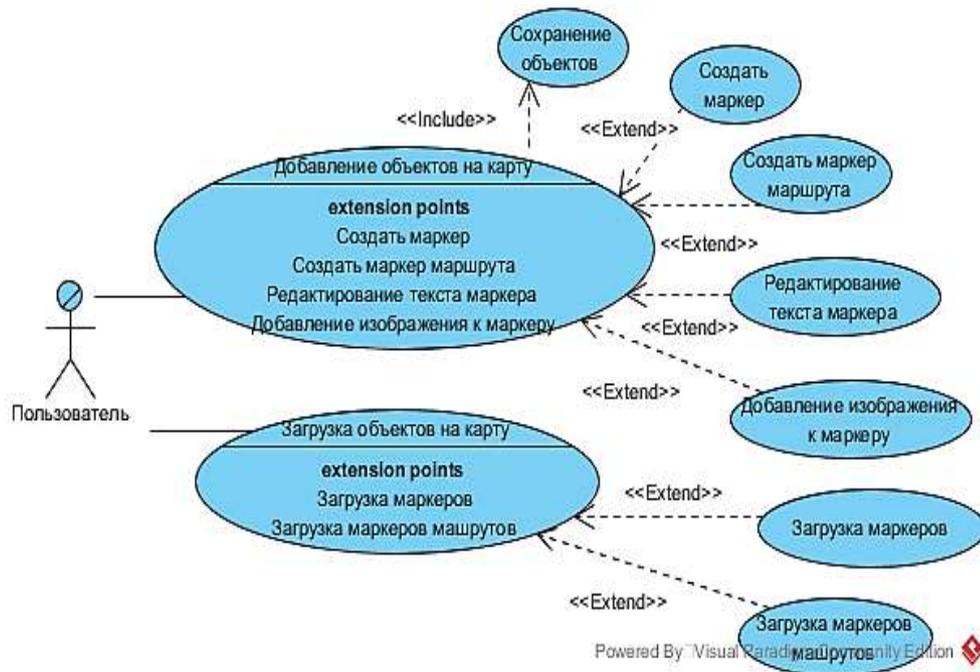


Рис. 1. UML-диаграмма ПО

При запуске программы появляется окно, в центре – web-браузер, загружающий карту, а в нижней части окна находится панель управления, состоящая из текстового поля и кнопок (рис. 2).

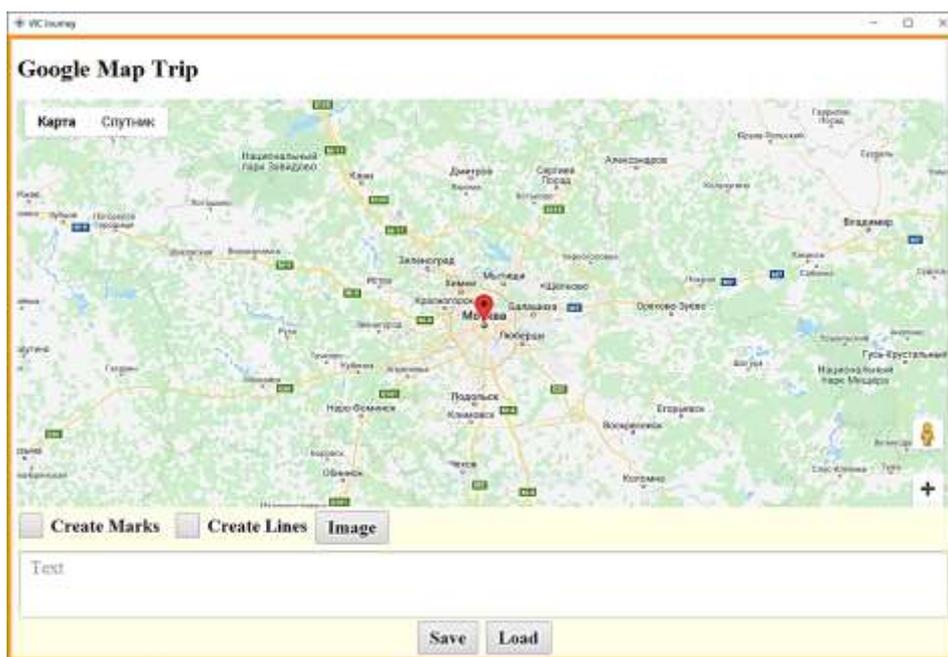


Рис. 2. Окно программы

Для создания маркера на карте требуется нажать кнопку-флажок «Create Marks», пока кнопка активна, маркеры будут создаваться на карте при нажатии левой кнопкой мыши на выбранных пользователем позициях (рис. 3), для прекращения создания маркеров требуется снова нажать на кнопку-флажок «Create Marks». При нажатии на созданный маркер появляется информационное окно с текстом, для изменения этого текста требуется ввести текст в текстовое поле и нажать «Enter» (рис. 4). Для добавления изображения к окну маркера требуется нажать на маркер и нажать кнопку «Image», появится диалоговое окно выбора изображения, выбранное изображение появится в информационном окне маркера (рис. 5).

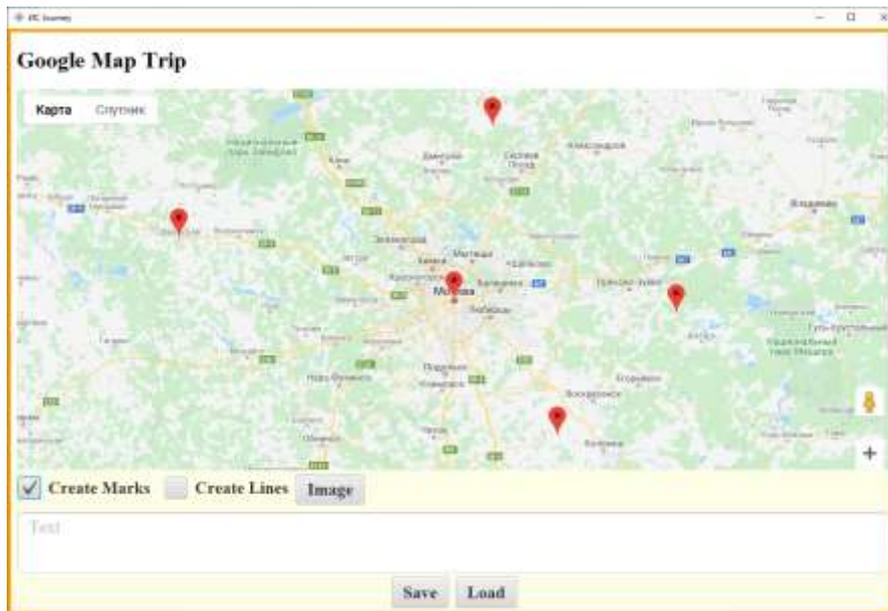


Рис. 3. Создание маркеров

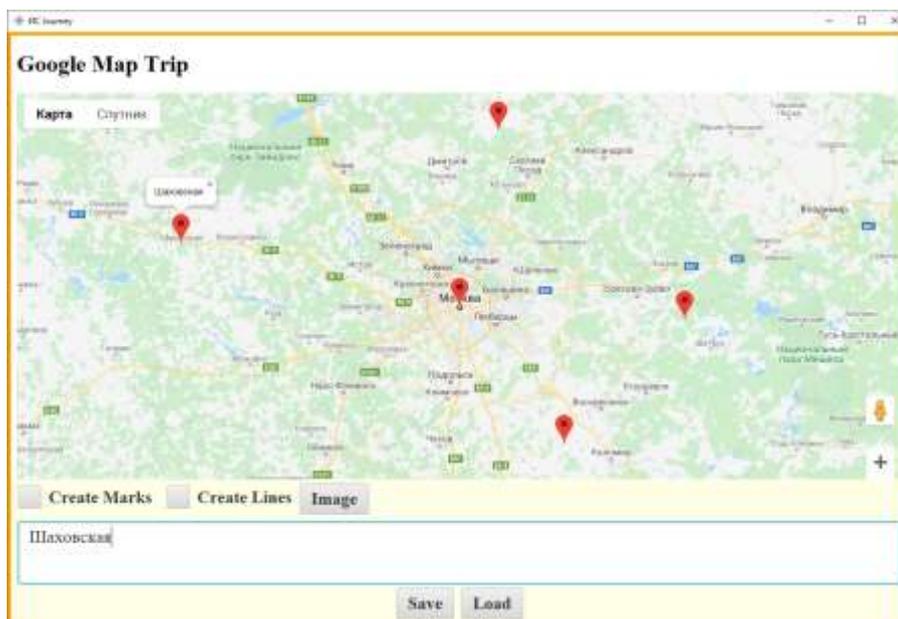


Рис. 4. Изменение текста информационного окна

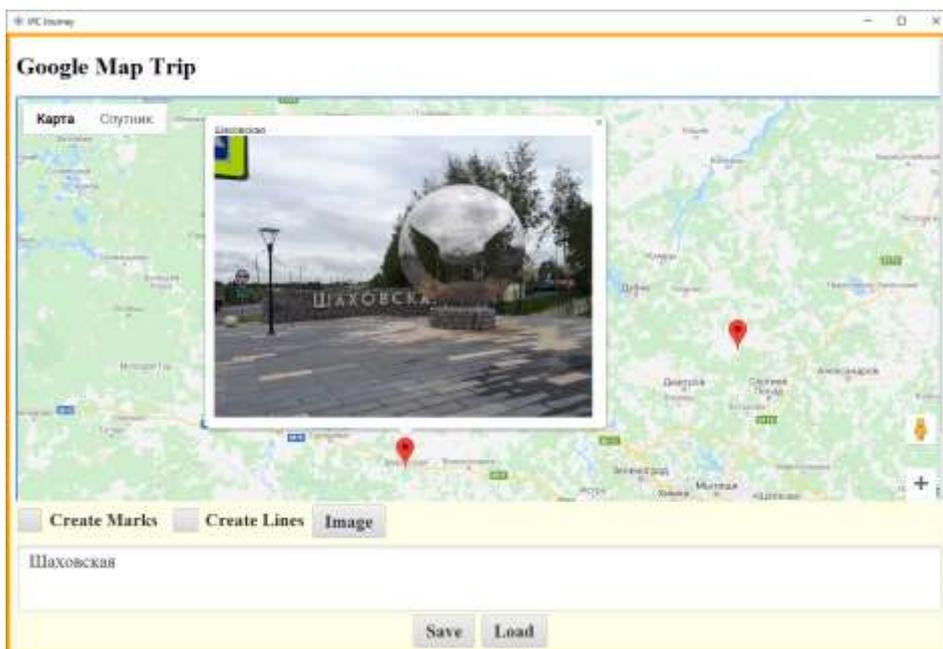


Рис. 5. Добавление изображения к информационному окну

Для создания маршрутов на карте требуется нажать кнопку-флажок «Create Lines», пока кнопка активна, при нажатии левой кнопкой мыши на выбранных пользователем позициях на карте будут появляться маркеры, а между ними поочередно будет прорисовываться маршрутная линия (рис. 6). При нажатии на каждый маркер маршрута также появляется его информационное окно, в котором можно изменить текст и добавить изображение (рис. 7).

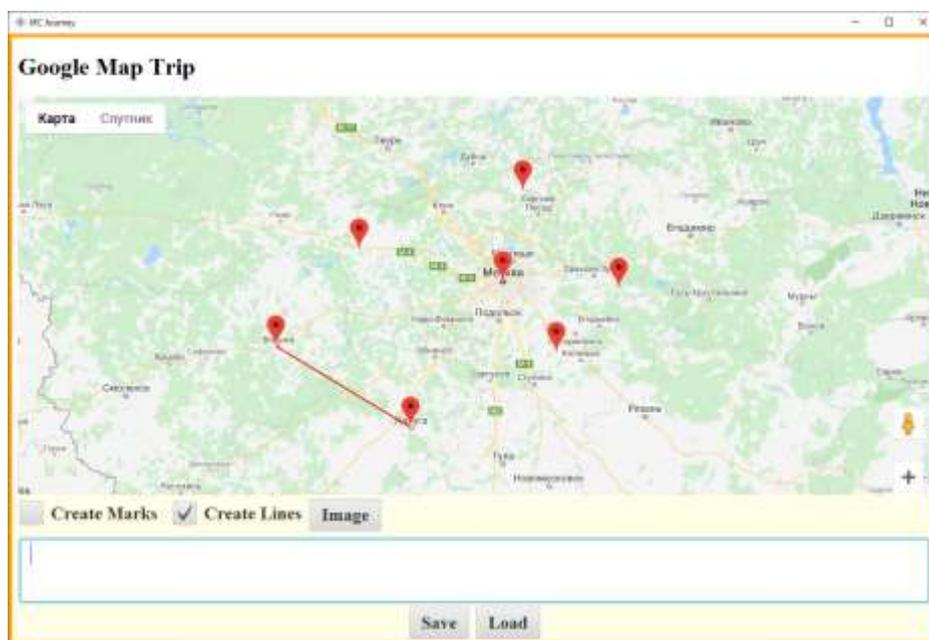


Рис. 6. Создание маршрутов

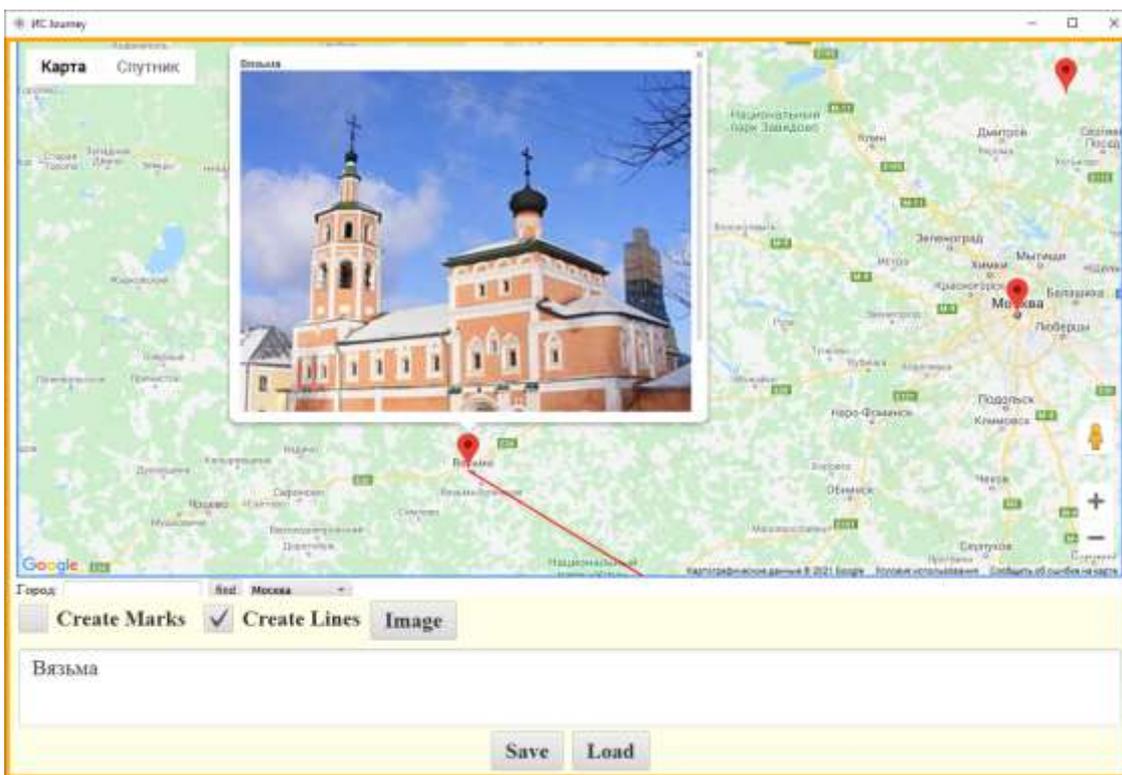


Рис. 7. Изменение текста и добавление изображения

Для сохранения всех созданных маркеров и маршрутов требуется нажать кнопку «Save». При новом запуске программы, для загрузки на карту ранее созданных маркеров и маршрутов, требуется нажать кнопку «Load».

© Филатов М.Е., 2022

СОВЕРШЕНСТВОВАНИЕ ТЕХНОЛОГИИ СИСТЕМЫ РАСПОЗНАВАНИЯ ЛИЦ ДЛЯ ДВЕРНЫХ ЗАМКОВ

Система распознавания лиц – это одна из самых популярных биометрических технологий в настоящее время. Многие люди используют её в своей повседневной жизни: для снятия телефона/ноутбука с блокировки, в банковских системах, социальных сетях, в офисах для доступа в помещение. Технология распознавания человека путем 3D-сканирования лица расширяет возможности систем безопасности, делая её применение быстрым, удобным и надежным. В отличие от пластиковых карт и механических ключей, лицо нельзя забыть дома или передать другому человеку [1].

Распознавание лиц признано одним из самых безопасных способов биометрической аутентификации, например, в сравнении с технологией сканирования отпечатков пальцев, т.к. подделать лицо невозможно. Однако данная технология для дверных замков имеет определенные недостатки: точность считывания лица не всегда достигает 100%, при возникновении проблем с доступом в помещение, невозможно заменить свое лицо другим. Это подтверждает наличие определенных проблем в распознавании лиц [5].

Таким образом, проблему, на решение которой направлен данный проект, можно сформулировать так: каким образом необходимо улучшить технологию системы распознавания лиц для дверных замков? Гипотеза исследования: создание улучшенной технологии распознавания лиц для дверных замков необходимо осуществлять на основе использования зависимого контролера системы контроля и управления доступом (СКУД), входящего в центральную сеть, управляемую центральными программами, с учётом обработки полученной информации по основным параметрам (антропометрические точки, исправление искажений, конвертация изображения в данные) и анализе различных изменений во внешности человека (шрамы, косметика, очки, маска и т.д.). Цель работы: совершенствование технологии системы распознавания лиц для дверных замков. Ресурсы: искусственный интеллект со считыванием и распознаванием биометрических данных, компьютерное оборудование и ПО, ресурсы, затраченные на электроэнергию, обучение пользователей.

Основной функцией продукта является распознавание лиц сотрудников организации. Фотографии всех сотрудников вносятся в систему. Дверной замок при обнаружении совпадений лиц с фото открывается и пропускает сотрудника. Система учитывает различные изменения в человеке (шрамы, косметика, очки, маска и т.д.). Помимо этого, система хранит информацию о том, кто из сотрудников входил и в какое время. Если кто-то другой пытается пройти, то система отправляет сигнал на компьютер в отдел безопасности. Для начала, рассмотрим несколько аналогов системы, которые помогут с разработкой собственного продукта [2].

Amazon Rekognition. Совпадение лиц устанавливается на основании их визуальной геометрии, в том числе на зависимостях между глазами, носом, бровями, ртом и прочими характеристиками лица. Когда сервис анализирует изображение, вокруг лица накладывается граничная рамка, определяющая единственную часть для анализа. Затем в ходе анализа определяются числа объектной нотации для изображения, обозначающие местонахождение главных элементов лица. Когда клиенты осуществляют поиск по лицам, технология сравнивает данные исходного изображения с каждым изображением, участвующим в поиске. Далее сервис присваивает оценку схожести каждому лицу на изображении.

Kairos. Компания искусственного интеллекта, специализирующаяся на распознавании лиц. Благодаря компьютерному зрению и машинному обучению Kairos может распознавать лица на видео, фотографиях и в реальном мире. Алгоритмы находят лица, обнаруживают характерные ориентиры, преобразуют их в уникальные строки чисел, которые используются для сравнения лиц. Установив Kairos можно добиться максимального контроля над своими личными данными и повысить уровень конфиденциальности. Также, можно использовать сервис для поиска, сопоставления и сравнения лиц или измерения таких характеристик, как возраст и пол.

Sigur. Возможность распознавания лиц уже встроена в Sigur. Для ее использования вам не потребуется какое-либо специфическое оборудование: для большинства случаев подойдет практически любая IP-камера и обычный современный компьютер. В Sigur функция распознавания основана на использовании технологии 2D-распознавания на базе нейронных сетей. В системе генерируется событие «Ожидание лица». В случае, если лицо не распознано или сотрудник не появился в кадре в течение 5 сек, система запрещает доступ. После предъявления системе первичного признака верификация считается пройденной, если лицо совпало с лицом любого объекта доступа из той же базы данных, что и объект доступа, идентифицированный по первичному признаку. Если лицо не было распознано, будет выведено соответствующее событие в интерфейсе наблюдения [3].

Signa. Американская страховая компания в сфере здравоохранения, позволяет клиентам подавать заявки на медицинское страхование, подписанные с использованием фотографии, а не рукописной подписи, чтобы сократить количество случаев мошенничества.

Из всех рассмотренных сервисов самым близким к нашему случаю является Amazon Rekognition, так как система проводит анализ лица в 3D формате, что позволяет получить более точные результаты. Благодаря определенному подходу Amazon Rekognition не получает информации об определенном человеке, а лишь информацию о потенциальной схожести этого лица с другим [4].

Функциональная модель информационной системы. ИТ-инфраструктура – это сложная многокомпонентная интегрированная система, которая является комплексом информационных технологий (программных и аппаратных средств) и обеспечивает деятельность организации.

Компьютерное оборудование, программное обеспечение, сетевые службы, сервисы, электронная почта, мониторинговые системы, политики информационной безопасности, периферийная техника и т.д. – всё это составляющие ИТ-инфраструктуры предприятия.

Рассмотрим подробнее минимальные требования для поддержания инфраструктуры компании. СКС – структурированные кабельные системы. Для работы организации требуется локальная сеть, объединяющая компьютеры, телефоны, периферийное оборудование. СКС – это универсальная телекоммуникационная инфраструктура офиса, обеспечивающая передачу сигналов всех типов. СКС может быть установлена прежде, чем станут известны требования пользователей, скорость передачи данных, тип сетевых протоколов. СКС создает основу компьютерной сети, интегрированной с телефонной сетью. Совокупность телекоммуникационного оборудования здания/комплекса зданий, соединенного с помощью структурированной кабельной системы, называют локальной сетью. Офисная АТС – Офисная автоматическая телефонная станция. Устройство, которое обеспечивает внутреннюю и внешнюю связь предприятия, прежде всего, телефонную. Все внешние и внутренние линии связи заводятся в мини АТС. Далее автоматическая телефонная станция в соответствии с программными установками обеспечивает и выход на внешние линии, и прием входящих звонков, и связь внутренних абонентов между собой.

Центральный ИБП – Центральный источник бесперебойного электропитания. Источник вторичного электропитания, автоматическое устройство, назначение которого обеспечить подключенное к нему электрооборудование бесперебойным снабжением электрической энергией в пределах нормы. Использование ИБП связано с обеспечением бесперебойной работы техники, позволяющее подключенному к ИБП оборудованию при неполадках с электрическим током некоторое непродолжительное время продолжить работу.

Сервисные шкафы. Металлическая закрытая конструкция для размещения и хранения всевозможного компьютерного оснащения. Для малого бизнеса довольно сложно оправдать дополнительные затраты, но, если решение является частью комплексного проекта модернизации, сделать это становится проще. Целями, которые преследует хорошо спроектированный стоечный корпус, являются работоспособность, организация, система проводов, физическая безопасность, эффективность охлаждения, простота распределения питания и профессионализм.

Периферийная техника. Это любые дополнительные и вспомогательные устройства, которые подключаются к ПК для расширения его функциональных возможностей. От мыши и клавиатуры до 3D-принтера и факса.

Стационарные и мобильные пользователи. Сотрудники, подключенные к локальной сети через программное обеспечение, с возможностью получать доступ к информации в офисе или вне офиса.

Как пользователь работает с системой? В первую очередь, необходимо запустить систему распознавание просто подойдя к терминалу, после чего система начнет поиск лица.

Далее, для того, чтобы лицо распознать необходимо хорошее качество освещения в зоне распознавания, важна ориентация лица относительно камеры и размер лица в кадре,

качество фотографий и видеопотока. В течение процесса распознавания нежелательно двигаться, для получения максимально точного считывания. В случае успешного распознавания, пользователь услышит соответствующий звуковой сигнал и звук открытия замка. При выявлении несоответствия система также выдаст соответствующий звуковой сигнал, после чего пользователь снова может пройти процесс считывания просто отстранившись от терминала и подойдя снова. Прототип системы распознавания лиц для дверных замков. Технология распознавания путем 3D-сканирования лица расширяет возможности систем безопасности, делая её применение быстрым, удобным и надежным.

Была улучшена технология, позволяющая открывать дверь при помощи своего лица. Для проведения точного сканирования, 3D-технология использует встроенную камеру и инфракрасный датчик, считывающий уникальные данные лица пользователя, что позволяет делать это даже в плохих условиях освещения. При приближении к замку срабатывает датчик движения и автоматически включается сканер. Доступ владельца в помещение производится менее чем за 1 секунду. Технология распознавания лиц – удобный и надежный способ открытия электронного замка. Сотрудник не будет беспокоиться о доступе в помещение, если потерял ключи/RF-карту/забыл код. Разблокировка двери осуществится, даже если внешность человека претерпела некоторые изменения: возрастные изменения, очки, медицинская маска, смена стиля, часть лица прикрыта каким-либо предметом.

Этапы работы улучшенной системы распознавания лиц.

1 этап. Обнаружение лица. В первую очередь, для того, чтобы лицо распознать, надо его сначала обнаружить. Когда человек находится на расстоянии 5-3 метров от двери, срабатывает датчик движения и система ищет лицо для обнаружения. Для обнаружения лица используется метод Виолы-Джонса, разработанный в 2001 году. Этот алгоритм просто сканирует изображение при помощи черно-белых прямоугольников, которые называются примитивами Хаара. Задача этих объектов - находить более светлые и темные области на изображении, характерных конкретно для человеческих лиц.

Сначала находится первый признак, система понимает: «В этой области может быть лицо». Тогда она начинает там же искать второй признак, а потом третий.

И если в одной области найдено 3 признака, уже можно уверенно сказать – да, это лицо. После чего система получает область изображения, в котором есть только лицо.

Камера может обнаружить лицо человека ростом 120-190 см на расстоянии 1 метра от двери. Чтобы понять обнаружила камера лицо или нет, можно посмотреть на экран на замке, который показывает область видимости. Если рост человека выше 190 см, то можно наклонить немного голову или отойти подальше на 10-20 см.

2 этап. Получение данных. Контроллерами называют электронику, которая призвана обработать входящую информацию и создать команду для замка сработать или нет.

Существует 2 вида таких систем: независимые и зависимые.

В нашем случае контроллер будет являться зависимым, так как он входит в большую сеть, управляется центральными программами и соединен с другими предметами.

Когда происходит обнаружение лица, данные изображения поступают в контролер СКУД, который обрабатывает входящую информацию и отправляет ее по IP-сети в сервер ПО для распознавания лиц.

3 этап. Обработка информации

После получения изображения, сервер ПО для распознавания лиц начинает обработку полученной информации, которая состоит из нескольких шагов.

1. Антропометрические точки. Каждое лицо составлено из различных ориентиров или узловых точек. Каждое человеческое лицо имеет 80 узловых точек. Программа производит анализ узловых точек, расставляет на лице антропометрические точки, по которым впоследствии и будут вычисляться индивидуальные характеристики человека: разрез глаз, форма носа, подбородка, расстояние между ними и прочее. Таких признаков может быть много, вплоть до нескольких тысяч.

2. Исправление искажений. Система производит дополнительное преобразование изображения: устраняется поворот и наклон головы. А также проводится 3D-реконструкция лица из 2D-изображения. Таким образом, даже если человек на изображении смотрел вбок, мы всё равно можем получить четкий фронтальный снимок, что существенно повышает качество распознавания.

3. Конвертация изображения в данные. Анализ лица превращается в математическую формулу. Черты лица становятся числовым кодом, который складывается из суммы характеристик лица: расстояний между опорными точками, текстуры определенных областей на лице и прочее. Такой числовой код называется отпечатком лица. Каждый человек имеет свой собственный отпечаток лица. Таких характеристик может быть множество. Основное правило: они должны описывать лицо независимо от посторонних факторов: макияжа, причёски, возрастных изменений.

После обработки изображения и перевода его в данные, информация поступает в сервер СКУД.

4 этап. Идентификация. На сервере СКУД технология определяет соответствия точных данных тем, что представлены в базе данных.

Если совпадения найдены, то информация о человеке и времени записывается в базу данных на компьютере, а информация о доступе в контролер СКУД, иначе на компьютер поступает информация о человеке, который хотел пройти, но не смог, и в контролер СКУД поступает информация об отмене доступа (табл.).

Таблица

Методы проверки выполнения функций тест-кейсами

Действие	Ожидаемый результат
Распознавание лица, не включенного в базу данных, в любых условиях.	Замок не открывается
Распознавание лица, включенного в базу данных в идеальных условиях (на лице нет маски, аксессуаров и других возможных помех, человек смотрит в камеру, стоит прямо, неподвижно, на небольшом расстоянии от двери, освещение хорошее)	Замок открывается
Распознавание лица, включенного в базу данных, при плохом освещении	Замок не открывается

Действие	Ожидаемый результат
Распознавание лица, включенного в базу данных, стоящего далеко от двери	Замок не открывается
Распознавание лица, включенного в базу данных, стоящего слишком близко к двери	Замок не открывается
Распознавание лица, включенного в базу данных, находящегося в слабом движении (человек совершает незначительные движения головой, возможно несильно поворачивает голову, туловище, шевелит руками)	Замок открывается
Распознавание лица, включенного в базу данных, находящегося в активном движении (быстро, далеко перемещающегося относительно камеры)	Замок не открывается
Распознавание лица, включенного в базу данных, стоящего неподвижно, при несильном наклоне головы, туловища	Замок открывается
Распознавание лица, включенного в базу данных, в медицинской маске (закрывающей рот и нос, как правило, не больше половины лица)	Замок открывается
Распознавание лица, включенного в базу данных, в маске, закрывающей большую часть лица	Замок не открывается
Распознавание лица, включенного в базу данных, в очках с прозрачными стеклами.	Замок открывается
Распознавание лица, включенного в базу данных, в очках с прозрачными стеклами и медицинской маске.	Замок открывается
Распознавание лица, включенного в базу данных, в очках с непрозрачными стеклами (солнечных).	Замок открывается
Распознавание лица, включенного в базу данных, в очках с непрозрачными стеклами и медицинской маске.	Замок не открывается
Распознавание лица, включенного в базу данных, при нанесении косметики, получении небольших травм лица (синяки, ссадины) и т.д.	Замок открывается
Распознавание лица, включенного в базу данных, в случае серьезных изменений во внешности (получение сильной травмы лица, результат пластической операции и т.д.)	Замок не открывается

5 этап. Открытие двери. Получив информация о доступе, контролер СКУД посылает сигнал в электронный замок для открытия двери.

В ходе работы над проектом «Улучшенная технология системы распознавания лиц для дверных замков» была осуществлена следующая работа:

- изучена информация о технологии распознавания лиц;
- проанализированы положительные и отрицательные стороны системы распознавания лиц;
- проанализированы способы улучшения системы распознавания лиц, учитывая непостоянность внешнего вида человека;
- разработана улучшенная технология распознавания лиц для дверных замков.
- протестирован получившийся интерфейс приложения технологии системы распознавания лиц для дверных замков.

Создание улучшенной технологии распознавания лиц для дверных замков было осуществлено на основе использования зависимого контролера СКУД, входящего в центральную сеть, управляемую центральными программами, с учётом обработки полученной информации по основным параметрам (антропометрические точки, исправление искажений, конвертация изображения в данные) и анализе различных изменений во внешности человека (шрамы, косметика, очки, маска и т.д.)

Литература

1. Брилюк Д.В., Старовойтов В.В. Распознавание человека по изображению лица нейросетевыми методами. Минск, 2002. 54 с.
2. Кулябичев Ю.П., Пивторацкая С.В. Обзор методов идентификации людей на основе изображений лиц с учетом особенностей визуального распознавания. М., 2008. 33 с.
3. Рогозин О.В., Кладов С.А. Сравнительный анализ алгоритмов распознавания лиц в задаче визуальной идентификации // Инженерный журнал наука и инновации. 2013. № 6(18).
4. Шерстобитов А.И., Федосов В.П., Приходченко В.А., Тимофеев Д.В. Распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации // Известия ЮФУ. Технические науки. 2013. № 11(148) С. 66-73.
5. Файтс Ф., Джонстон Т., Кратц М., Шноль В.Г., Мартынюк В.В. Компьютерный вирус: проблемы и прогноз. М.: Мир, 2013. 175 с.

© Фролова Т.А., Шевелева О.Г., 2022

РАЗРАБОТКА GUI-ПРИЛОЖЕНИЯ ДЛЯ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ С ПОМОЩЬЮ C++/CLI WINDOWS FORMS

В современных учебных заведениях невозможно обеспечить учебный процесс без использования информационных технологий. Чаще всего автоматизация принятия решений в области оценки знаний обеспечивается средствами тестирования, и наибольшее распространение получило компьютерное тестирование [3]. Компьютерное тестирование является наиболее оперативной, современной и объективной формой контроля знаний учащихся. К его положительным чертам можно отнести: экономию времени на проверку (что особенно ценно при работе с группами тестируемых) и быструю выдачу тестовых баллов, исключение ошибки при обработке результатов, возможность самоконтроля и самоподготовки учащихся, возможность использования любых изображений, компьютерных моделей, аудио- и видеофайлов [3; 5], и т.д. Тесты используются не только в образовательных учреждениях. Различные предприятия и организации осуществляют аттестацию и сертификацию своих сотрудников посредством компьютерного тестирования. Так, проведение викторин позволяет разработчикам учебных материалов и профессионалам L&D (learning and development) понять объем и уровень знаний и опыта сотрудников, чтобы они могли лучше удовлетворять потребности своих сотрудников в обучении. Это также позволяет им оценивать успеваемость учащихся, определять пробелы в знаниях и получать обратную связь, которая помогает улучшить программы обучения.

Большую роль в успешности прохождения теста играет пользовательский интерфейс, так как здесь закладываются функции и возможности движения по заданиям теста, элементы размещения информации на экране, а также общий визуальный стиль системы тестирования и методы представления информации. Хороший интерфейс пользователя должен обладать ясностью и корректностью логической последовательности взаимодействия с испытуемым, отражая общие принципы дизайна графической информации [3].

Язык C++/CLI является привязкой языка программирования C++ к платформе Microsoft .NET [6]. C++/CLI интегрирует C++ стандарта ISO с Объединенной системой типов (Unified Type System – UTS), рассматриваемой как часть общей языковой Инфраструктуры (Common Language Infrastructure – CLI). Приложение, написанное на любом, поддерживаемым платформой .NET Framework, языке программирования, переводится компилятором в промежуточный код – «высокоуровневый ассемблер» виртуальной машины .NET – Common Intermediate Language (CIL). Созданный код исполняется в среде Common Language RunTime (CLR) [4; 2]. Помимо общезыковой среды выполнения CLR, другим важнейшим компонентом среды .NET является библиотека классов .NET Framework, включающая

интерфейс визуального программирования приложений с графическим интерфейсом пользователя Windows Forms [1].

Несмотря на то, что на сегодняшний день существует множество онлайн и офлайн сервисов для проведения интерактивных викторин, опросов и разнообразных квизов, задача разработки приложения для реализации компьютерного тестирования всё же остается актуальной. Создадим приложение «С++ тестирование», в котором вопрос появляется в виде изображения, доступны 4 варианта ответа, есть возможность листать вопросы вперед и назад, а также в любой момент завершить тестирование. При нажатии на кнопку «Завершить тестирование» программа должна оценить тестируемого и дать некоторые рекомендации.

Немаловажным этапом является разработка юзабилити приложения, которая базируется на принципе «чем проще – тем лучше». Дизайн должен быть минималистичным, чтобы было просто ориентироваться без предварительного обучения пользованию, и визуально красивым. Итак, для начала на форму необходимо поместить следующие компоненты: три компонента Button (кнопки для перехода на следующий и предыдущий вопрос, а также кнопка для подведения итогов тестирования), четыре компонента RadioButton с подписями, соответствующими вариантам ответа (a, b, c и d), а также компонент Label, который будет отображать номер вопроса. Сам вопрос и варианты ответа будут находиться на картинке, поэтому помещаем на форму также компонент PictureBox (рис. 1).

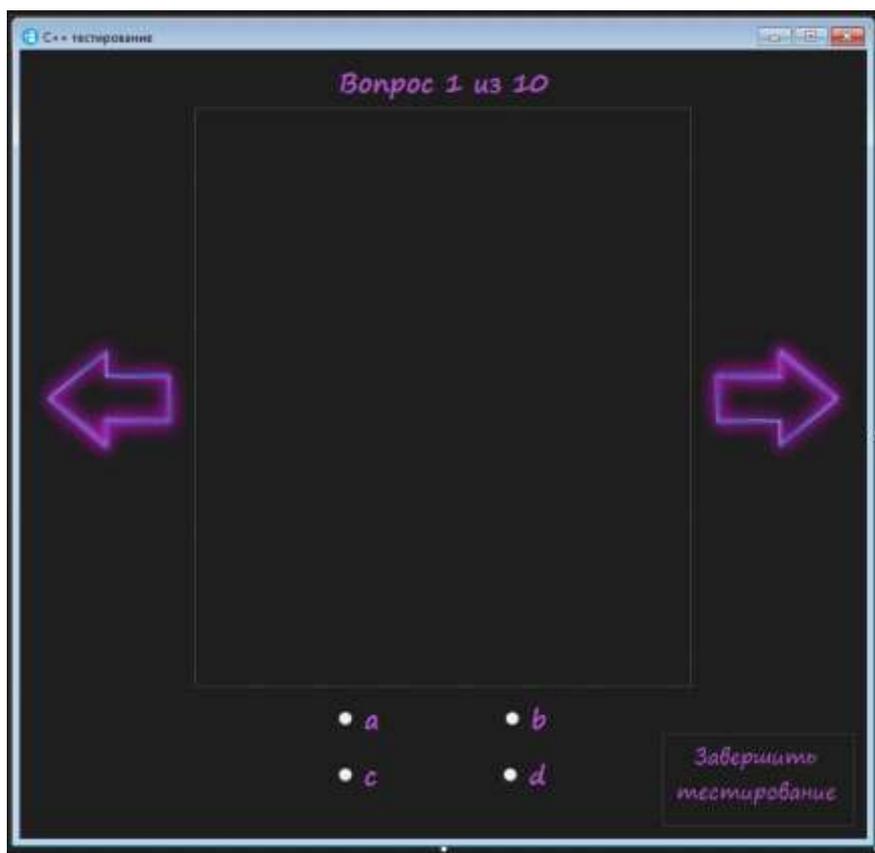


Рис. 1. Скриншот окна конструктора формы

Так как дизайн минималистичен и содержит немного текста, можно задать в форме темную тему, набирающую сейчас популярность во всех сферах разработки и добавляющую

стиль, – в качестве фонового цвета установим транспортный черный (в палитре RGB – 30; 30; 30). Свойству формы ForeColor присвоим значение MediumOrchid для обеспечения должной контрастности, кнопки для пролистывания вопросов сделаем в виде изображений стрелок. В качестве основного был установлен шрифт Segoe Print 16, (рис. 1). Свойству MouseOverBackColor кнопки «Завершить тестирование» был присвоен цвет Indigo, чтобы при наведении на нее курсора мыши она меняла цвет, предупреждая, что нужно обратить внимание при совершении нажатия на неё.

Сразу после загрузки формы в ней будет отображаться первый вопрос теста. Для обеспечения пролистывания вопросов в тесте сделаем активными кнопки навигации между вопросами. Щелчки по ним должны обрабатываться таким образом, чтобы соответствующему клику соответствовала картинка теста (рис. 2). Для начала инициализируем переменную для хранения номера картинки: $n = 0$. При щелчке по кнопке со стрелкой право (переход на следующий вопрос) происходит прибавление номера n на единицу, а по кнопке со стрелкой влево (переход к предыдущему вопросу) – уменьшение значения n на единицу. Пусть тест состоит из 10 вопросов, тогда, если $n < 0$, то $n = 0$; и, если $n > 9$, то $n = 9$. Далее вызывается функция отображения вопроса, которой в качестве входного параметра передается номер вопроса.

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
    n--;  
    if (n < 0) n = 0;  
    Shows(n);  
}  
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {  
    n++;  
    if (n > 9) n = 9;  
    Shows(n);  
}
```

Рис. 2. Методы обработки щелчков
для перехода к предыдущему и следующему вопросу

Метод отображения вопроса описывается следующим образом. В зависимости от номера текущего задания n в компонент Label помещается строка, отображающая номер вопроса $n+1$ для удобной навигации, а в компонент PictureBox загружается соответствующая картинка-вопрос (рис. 3).

```
pictureBox1->Image = gcnew Bitmap("C:\\faust\\v" + (n + 1).ToString() + ".png");
```

Рис. 3. Фрагмент функции отображения задания – отображение картинки

Далее необходимо добавить возможность отвечать на вопросы, то есть сделать активными радиокнопки. Для этого сначала нужно создать пустой массив answer для хранения ответов пользователя. При выборе варианта «а» в соответствующую номеру вопроса ячейку массива answer[n] запишется 1, при выборе варианта «b» запишется 2, при

выборе «с» – 3 и при выборе варианта «d» – 4. Нам нужно также сохранять ответы на конкретные вопросы в тесте и посматривать предыдущие ответы, поэтому нужно установить свойство Checked для всех случаев нажатия на какую-то RadioButton. Удобно воспользоваться switch-переключателем, в качестве переключающего выражения в котором выступает ответ answer[n], который может быть равен 1, 2, 3 или 4 в зависимости от того, какой вариант был выбран (рис. 4). По умолчанию (default) все радиокнопки установлены со значением false.

```
switch (answer[n]) {  
case 1:  
    radioButton1->Checked = true;  
    radioButton2->Checked = false;  
    radioButton3->Checked = false;  
    radioButton4->Checked = false;  
break;
```

Рис. 4. Фрагмент функции отображения вопроса – отображение выбора ответа

На рисунке 5 представлен результат работы программы для реализации тестирования.

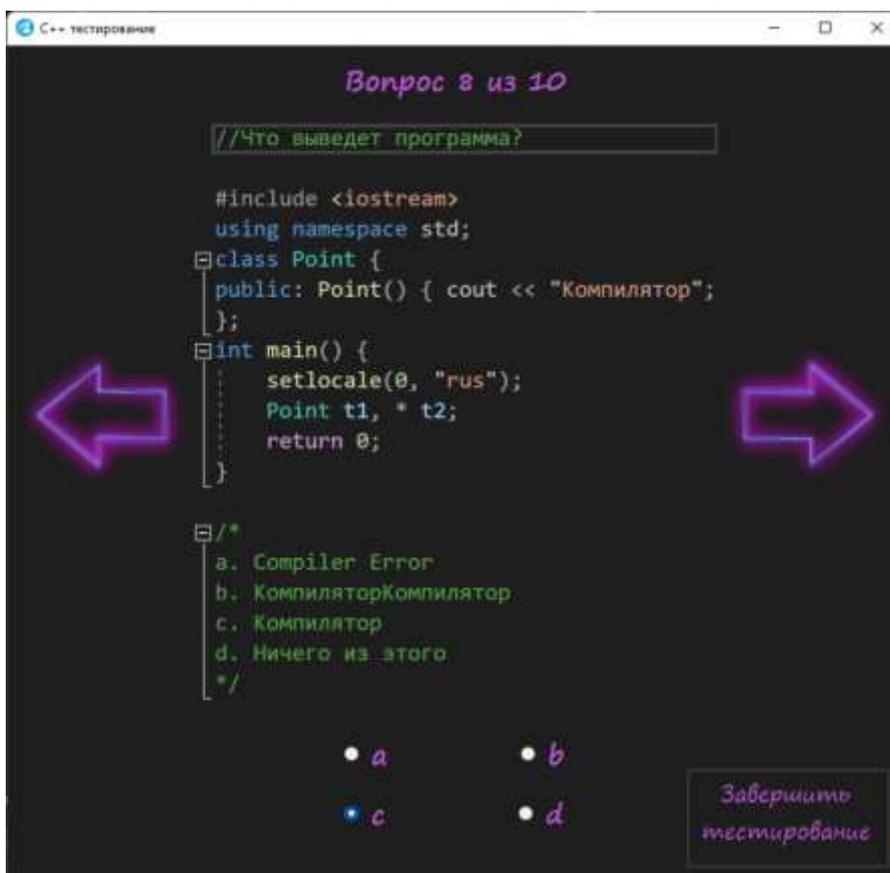


Рис. 5. Скриншот окна приложения

Кнопка «Завершить тестирование» обрабатывает результаты прохождения теста следующим образом. Выполняется поэлементное сравнение двух массивов: пользовательских ответов с правильными ответами. За каждый правильный ответ начисляется 1 балл (рис. 6).

```
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {  
    int otvet[10] = { 3, 2, 3, 1, 3, 2, 1, 3, 3, 1 }; //массив правильных ответов  
    int correct = 0;  
    for (int i = 0; i < 10; i++)  
        if (answer[i] == otvet[i]) correct++;  
    int prcnt = correct * 100 / 10;  
    String^ msg = "\nВы не прошли тест, пройдите его еще раз внимательно!";  
    if (prcnt >= 50) msg = "\nПоздравляем! Вы прошли тест!";  
    MessageBox::Show("Вы ответили на " + correct + " вопросов верно. Это составляет "  
        + prcnt + " процентов. " + msg);  
}
```

Рис. 6. Метод обработки щелчка по кнопке «Завершить тестирование»

Затем вычисляется процент правильного выполнения теста и сравнивается с некоторым «пороговым значением». Все результаты тестирования выводятся в диалоговое окно (рис. 7).

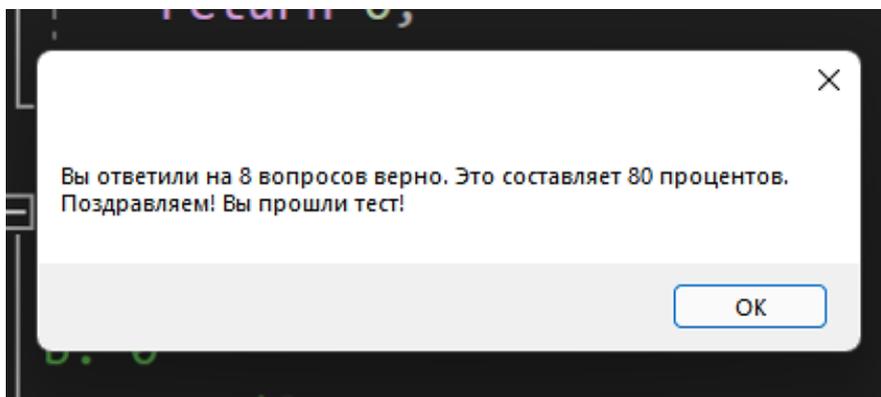


Рис. 7. Скриншот диалогового окна с результатами тестирования

Таким образом, было разработано GUI-приложение для реализации компьютерного тестирования. Разработанное приложение является базовым и планируется быть использовано для добавления в электронное учебное пособие.

Литература

1. Зиборов В.В. MS Visual C++ 2010 в среде .NET. Библиотека программиста. СПб.: Питер, 2012. 320 с.
2. Лазарева А.Б., Эварт Т.Е., Прис Н.М., Глухова А.Ф. Разработка в VISUAL C++ приложений с графическим интерфейсом для решения инженерных задач. Н. Новгород: Нижегород. гос. техн. ун-т им. Р.Е. Алексеева. 2020. 101 с.
3. Погуда А.А. Модели и алгоритмы контроля знаний по гуманитарным дисциплинам: Дисс. ... канд. техн. наук. Томск, 2016. 174 с.
4. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. СПб.: Питер, 2013. 896 с.
5. Соколова И. И. Компьютерное тестирование как наукоемкая педагогическая технология // Известия Российского государственного педагогического университета им. А.И. Герцена. 2004. Т. 4. № 9. С. 77–86.



6. Хогенсон Г. С++/CLI: язык Visual С++ для среды .NET = С++/CLI Primer. М.: Вильямс, 2007. 464 с.

© Харитонов Л.С., 2022

Хвостова А.В.

Южно-Уральский государственный
гуманитарно-педагогический университет
г. Челябинск, Россия

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ СОЗДАНИЯ И ПРОВЕДЕНИЯ КВИЗОВ

Квизы или викторины как вид досуга прочно вошли в повседневную жизнь еще со времен сканвордов в газетах и журналах, однако на этом их развитие не остановилось. В эпоху цифровизации и развития интернет-сетей, популярность всевозможных тестов, опросов и викторин только возрастает.

Основное свое применение они находят, конечно же, в образовании: проведение экзаменов и зачетов, прохождение всевозможных тестов и непосредственно процесс обучения в виде викторин или ситуационных задач. Зачастую, тестовыми опросами пользуются в конце занятий, чтобы подкрепить только изученный материал и определить степень вовлеченности учащихся.

При этом, несмотря на популярность викторин как явления и обилие сайтов, функционалом которых служит создание тестов, существует дефицит продуктов для офлайн создания тестовых опросов. Помимо этого, многие из существующих программных продуктов не имеют поддержки русского языка. Это и послужило нашей задачей и целью работы: разработать программный продукт, позволяющий составлять и проводить викторины. Викторина – игра в ответы на вопросы (устная или письменная) из разных областей знания. Слово «викторина» появилось в 1928-х г. в журнале «Огонёк». Известный советский журналист и писатель Михаил Кольцов так озаглавил подборки, включающие в себя вопросы, шарады, ребусы и т.п.

Классифицировать викторины по видам достаточно сложно, поскольку, так или иначе, признаки, по которым их можно разделить на группы, зачастую пересекаются друг с другом. Однако все викторины можно разделить на три большие группы по типу деятельности: интеллектуальные, спортивные, творческие [1].

Независимо от вида викторины, условий проведения, правила должны отвечать ряду требований [1]: правила должны быть просты; викторина должна охватывать всех; викторина должна быть интересна для всех, это требование тесно связано со следующим; викторина должна быть доступна для всех предполагаемых участников; задания, содержащиеся в викторине, должны быть одинаковыми или равными по содержанию и сложности для всех. Равенства требует не только задание, но и способ привлечения к его выполнению.

Данная работа посвящена интеллектуальным викторинам. Для анализа были выбраны следующие представители систем для создания викторин:

Revision Quiz Maker – онлайн инструмент для создания тестов и викторин. В распоряжении разработчика различные варианты вопросов (от простейшего Да/Нет до

вопросов со множественным выбором, с самостоятельным вводом ответа и т.п.), доступ через Facebook, e-mail или по ссылке, а также получение разнообразной статистики.

Недостатками данного сервиса является необходимость стабильного интернет-соединения, отсутствие возможности выгрузки викторины в файл, а также отсутствие поддержки русского языка. Quizinator — мощный инструмент для создания интерактивных тестов, викторин, экзаменов. Поддерживает различные типы вопросов, использование графики и видеоматериалов, различную специфическую информацию, такую как геометрические эскизы, электрические схемы, карты, химические элементы, формулы и многое другое. Можно задавать произвольную выборку вопросов из банка данных, их случайную сортировку, ограничивать время, на прохождение теста и получать различные виды отчетов.

Недостатки данного сервиса в точности повторяют минусы предыдущего продукта, к тому же, присутствует перегруженность интерфейса. Исходя из анализа данных и других популярных сервисов для создания викторин, можно выделить следующее: большинство программ для создания викторин являются онлайн-сервисами, которые не позволяют выгружать либо работать с викторинами при плохом или отсутствующем интернет-соединении; большинство программ не имеют поддержки русского языка; многие программы и сервисы требуют платную подписку. Благодаря чему и было принято решение разрабатывать подобный продукт самостоятельно. Средой разработки для данного проекта выбрана Microsoft Visual Studio 2017. Программа написана на языке C# [2].

Для реализации поставленных задач были разработаны классы, обеспечивающие программное представление данных, необходимых для работы программы, а также методы для работы с ними (рис. 1).

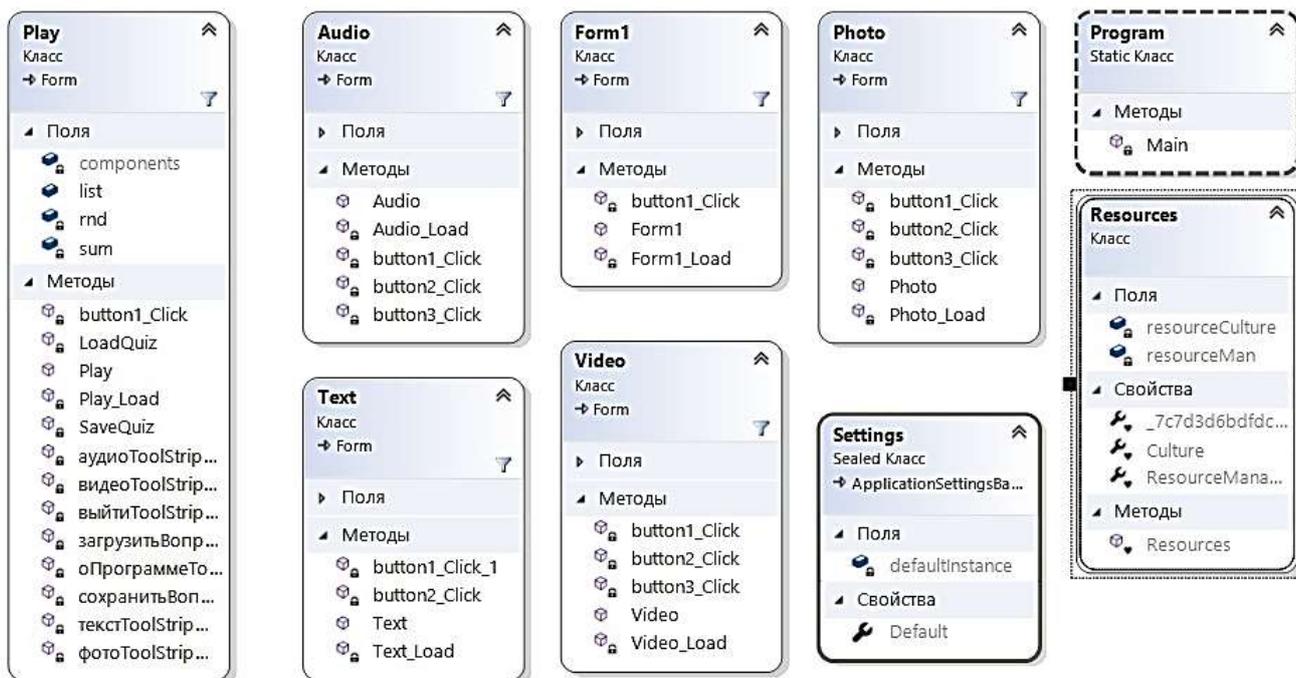


Рис. 1. Диаграмма классов

При запуске программы демонстрируется следующее окно (рис. 2). При нажатии кнопки «Далее», пользователь переходит к основному окну программы (рис. 3).

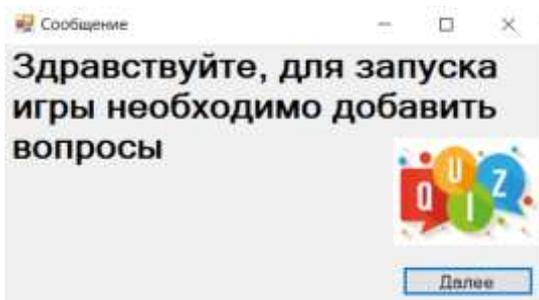


Рис. 2. Окно запуска

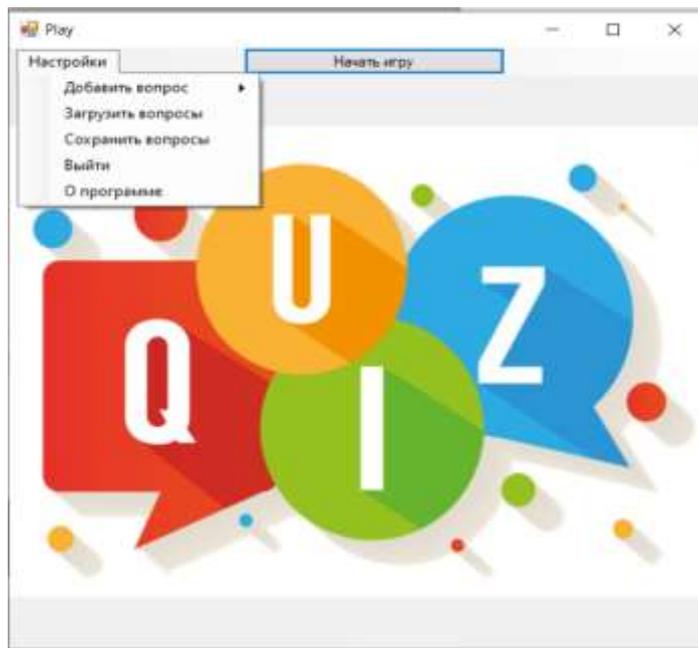


Рис. 3. Основное окно программы

Основное окно содержит в себе:

1) меню настроек, позволяющее добавить вопросы в текущую викторину или сохранить ее, загрузить предварительно созданную в программе викторину, получить справку или выйти из приложения;

2) кнопку «начать игру», которая позволяет перейти к прохождению викторины.

На выбор можно добавить в викторину четыре вида вопросов (рис. 4): на основе изображения, видеофрагмента, аудио-фрагмента и обычный текстовый вопрос.

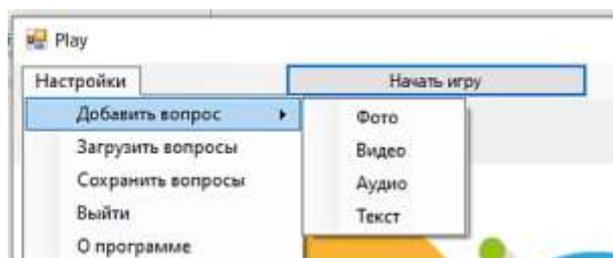


Рис. 4. Меню Настройки

При выборе типа вопроса, появляется окно-конструктор (рис. 5), в котором, в случае, если это вопрос на основе изображения, видео- или аудио-фрагмента, можно выбрать соответствующий типу и вопросу файл, а также непосредственно задать сам вопрос и ответы к нему (рис. 6).

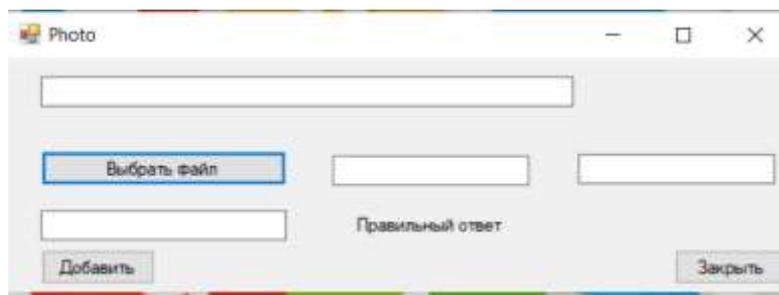


Рис. 5. Конструктор вопроса

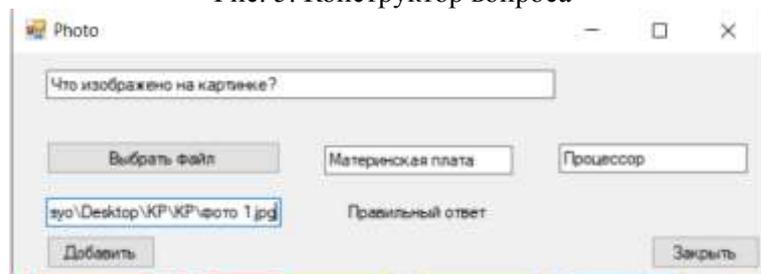


Рис. 6. Заполненный конструктор вопроса

Для того чтобы сохранить созданную викторину, необходимо в меню «Настройки» выбрать пункт «Сохранить викторину». Откроется стандартное окно сохранения файла. В случае, если сохранение прошло успешно, программа покажет уведомление об успешном сохранении. В случае ошибки либо отказа пользователем сохранять викторину, программа также покажет уведомление. Викторина сохраняется в текстовый файл, в котором находится вся информация, необходимая для корректной работы приложения (рис. 7).

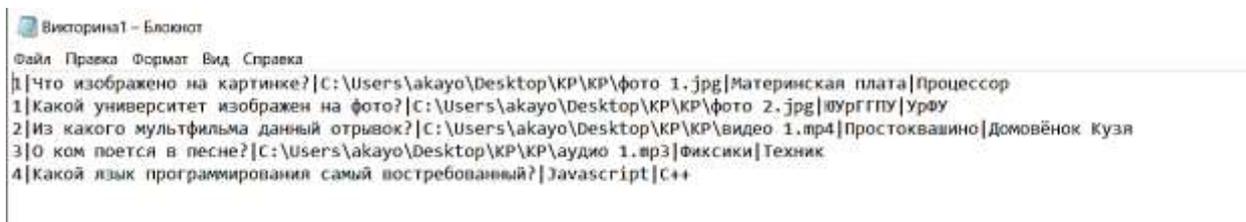


Рис. 7. Содержание текстового файла

После нажатия кнопки «Начать игру», программа будет случайным образом выдавать вопросы, созданные либо загруженные пользователем, до тех пор, пока не будут продемонстрированы все вопросы (рис. 8). После того, как будут даны ответы на все вопросы, программа покажет результат пользователя в виде набранных баллов. Для того чтобы загрузить заранее созданную викторину, необходимо выбрать пункт «Загрузить викторину» в меню «Настройки». В стандартном диалоговом окне пользователь может выбрать нужный файл и загрузить. В случае успешной загрузки программа покажет уведомление об успешной загрузке. Таким образом, в ходе выполнения работы был проведен анализ приложений для создания викторин, и выбрано средство разработки на языке C#. Разработан программный продукт, позволяющий создавать и проводить викторины с таким функционалом как сохранение и загрузка готовых викторин. Реализована поддержка разных видов мультимедиа в дополнение к обычным текстовым вопросам. Тестирование программного продукта прошло успешно.

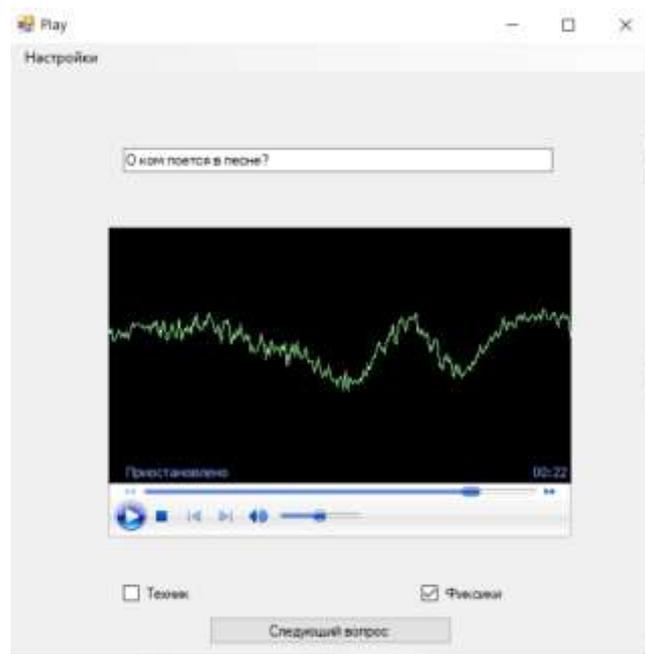


Рис. 8. Интерфейс викторины

Данная программа может быть повсеместно включена в инструментарий для проведения интерактивных занятий благодаря поддержке разных типов вопросов. Для последующего расширения функционала может быть добавлено ведение логов пользователей, проходивших ту или иную викторину, шифрование и дешифровка данных готовых викторин для исключения вмешательства со стороны. Также можно добавить таймер, ограничивающий время прохождения викторины, либо засекающий, сколько времени было потрачено на вопросы.

Литература

1. Данильченко М.А., Мухина М.В., Булаева М.Н., Воронина И.Р. Возможности использования игровых интерактивных методов в образовательном процессе вуза // Азимут научных исследований: педагогика и психология. 2020. Т. 9. №4 (33). С. 183-187. <https://doi.org/10.26140/anip-2020-0904-0038>
2. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. М.: Вильямс, 2015. 1312 с.

© Хвостова А.В., 2022

ГЕОИНФОРМАЦИОННАЯ СИСТЕМА КАК СПОСОБ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ И ЭФФЕКТИВНОСТИ РАСПРЕДЕЛЕННЫХ ЭЛЕКТРИЧЕСКИХ КОМПЛЕКСОВ В СИРИЙСКОЙ АРАБСКОЙ РЕСПУБЛИКЕ

Геоинформационная система, как способ повышения безопасности и эффективности распределенных электрических комплексов в Сирийской Арабской Республике, который заключается в снижении вероятности аварий и рисков, таких как пожары и нарушение операций, выполняемых электрическими комплексами, такие как передача, распределение и многие другие операции, путем мониторинга данных о погоде, температуре и прогнозировании риска землетрясений из Международного центра приложений наук о Земле (SEDAC) НАСА. Целью исследования является создание системы предоставления пространственной информации обо всех комплексах сирийских электрических сетей [2], идентификации электростанций и потребителей, мониторинга метеорологической обстановки, изучаемой глобальными навигационными приборами, что способствует выполнению данной работы по оцифровке сирийской электросети [1]. Для достижения поставленной цели решались следующие задачи:

1. Для кодирования различных структур географических данных использовался формат GeoJSON. Которая использует географическую систему отсчета координат, Всемирную геодезическую систему 1984 года и единицы десятичных градусов. Были добавлены файлы GeoJSON и векторные слои, размещенные из файла GeoJSON.

Все файлы GeoJSON поддерживают следующие типы объектов:

- Пункт (включая адреса и местоположения)
- Серия линий (включая улицы и автомагистрали, соединяющие электропарки)
- Полигоны — это составные группы точек, ряды линий или контуры полигонов.

Так характеристики и координаты электроподстанции в городе Банияс были получены с использованием геокодирования данных о местоположении с помощью GeoJSON (рис. 1).

2. В качестве платформы для размещения пространственных данных был использован NextGIS Web (<https://clck.ru/eq4wv>). Данный сервис позволяет размещать «географические данные», осуществлять навигацию, управлять содержимым карты через веб-интерфейс; гибко назначать права доступа к слоям, группам слоев (рис. 2) и картам.

3. Служба NextGIS Connect использовалась в настольном программном обеспечении GIS для интеграции QGIS (<https://clck.ru/eq4yv>) и NextGIS.com, путем создания соединения, введя данные учетной записи, созданной на сайте NextGIS.com. Что позволяет получать информацию с сайта и отображать все ресурсы облачного сервиса на экране (рис. 3).

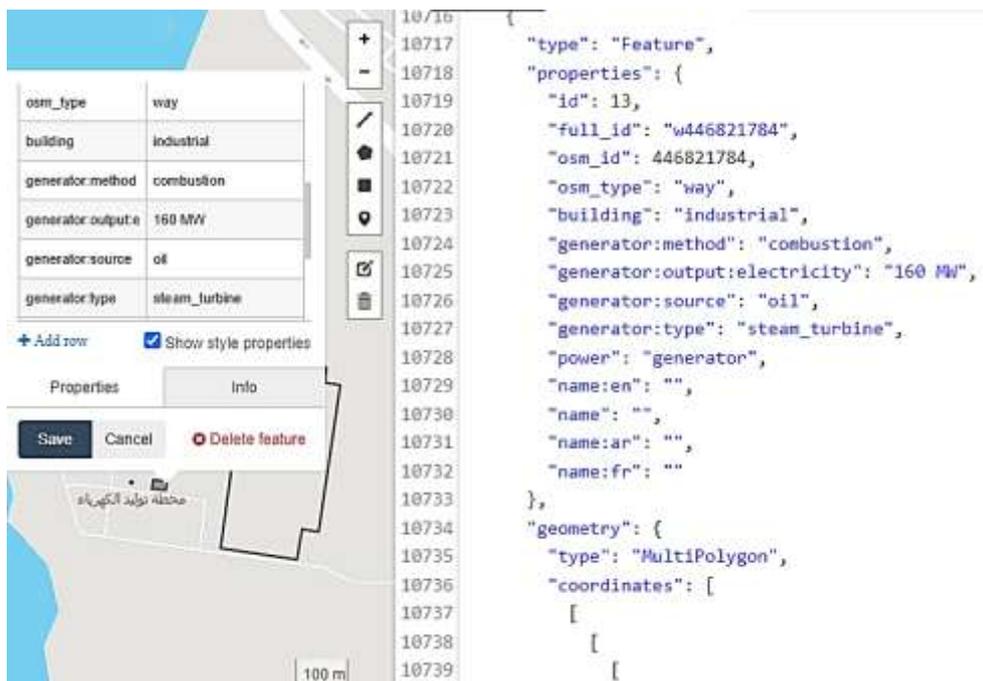


Рис. 1. Пример данных о местоположении в формате GeoJSON

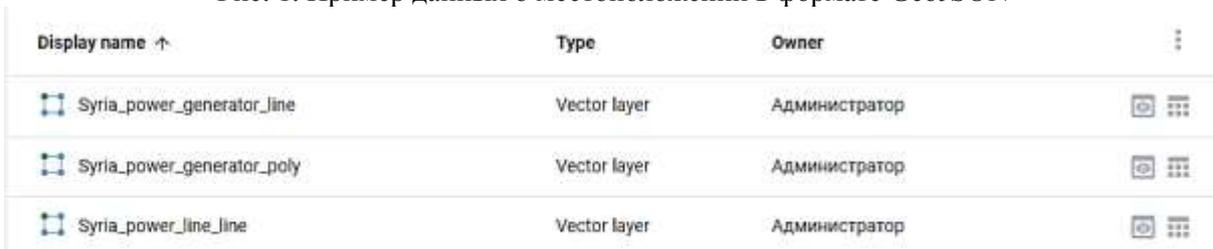


Рис. 2. Создание слоев в приложении NextGIS

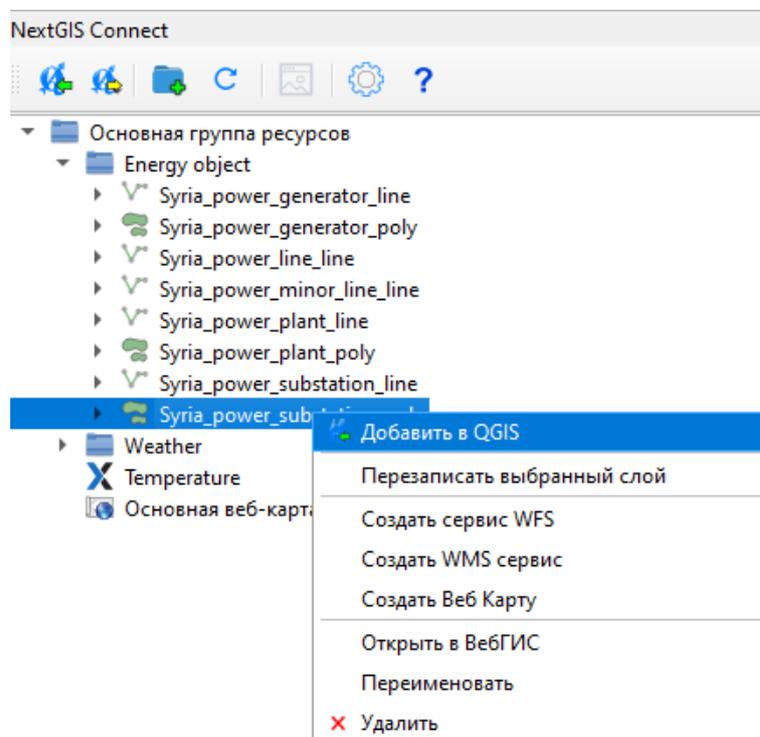


Рис. 3. Интеграция QGIS и NextGIS

4. NextGIS Mobile — это мобильная ГИС для операционной системы Android. что также позволяет: связывать растровые и векторные слои с nextgis.com и NextGIS Web; Таким образом, он накапливается и передается в фоновом режиме в режиме онлайн. Ранее созданные векторные геоданные и атрибуты векторного слоя могут быть созданы и изменены непосредственно с мобильного устройства.

Основная карта Сирии после добавления восьми слоев главных и второстепенных электрических и тепловых электростанций и распределительных станций представлена на рисунке 4.



Рис. 4. Основная веб-карта

Данные прогноза риска землетрясений были получены из Международного информационно-прикладного центра наук о Земле (SEDAC) НАСА и загружены в виде слоя в приложение Nextgis.

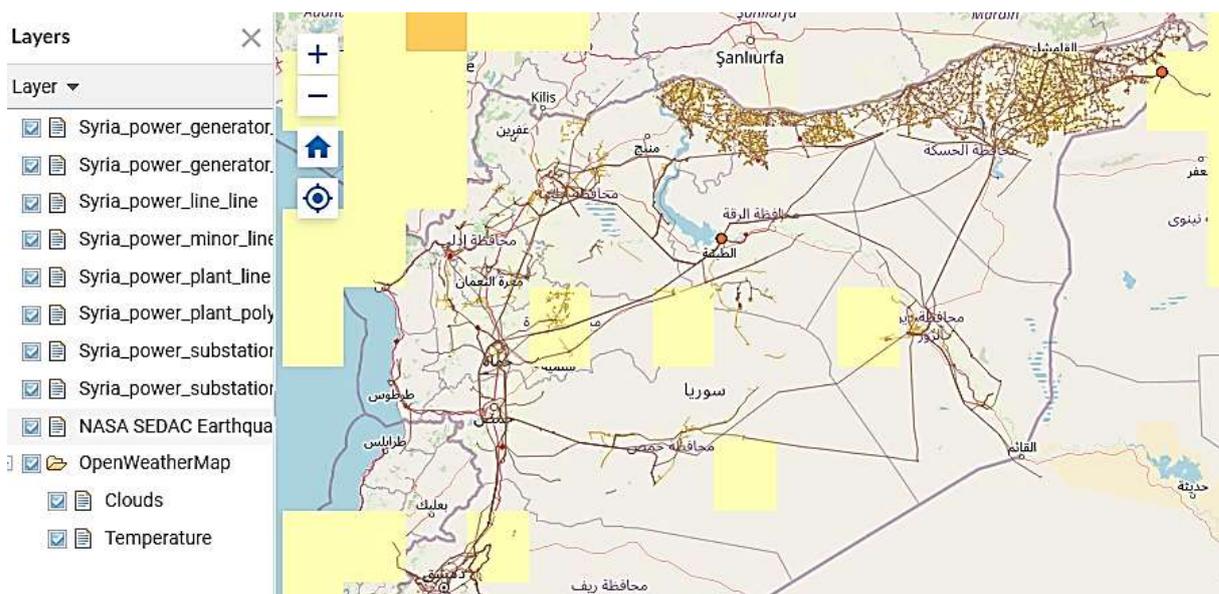


Рис 5. После применения данных прогноза риска землетрясений

Из карты видно, что можно предсказать возможность землетрясений, возможность повреждения сети производства электроэнергии, расположенной на дальнем востоке Сирии, в дополнение к возможности повреждения сети электропередачи.

Представленная геоинформационная система, может использоваться как один из способов повышения безопасности и эффективности распределенных электрических комплексов в Сирийской Арабской Республике, который заключается в снижении вероятности аварий и рисков, таких как пожары и нарушение операций, выполняемых электрическими комплексами, такие как передача, распределение и многие другие операции, путем мониторинга данных о погоде, температуре и прогнозировании риска землетрясений.

Литература

1. Донгак С.Б.Б., Хамитов Р.М. Особенности разработки информационных модулей для предприятия // Современные проблемы цивилизации и устойчивого развития в информационном обществе: Сб. материалов VI Международной научно-практической конференции (г. Москва, 30 декабря 2021 года). М., 2021. С. 174-177.

2. Хджейра М. Разработка интеллектуальной системы учета электроэнергии в Сирийской Арабской Республике // Приоритетные направления инновационной деятельности в промышленности: Сб. научных статей VII международной научной конференции (г. Казань, 30–31 июля 2021 года). Казань, 2021. С. 115-118.

© Хджейра М., Хамитов Р.М., 2022

УДК 004.852

Челышев Э.А., Раскатова М.В., канд. техн. наук
Национальный исследовательский университет «МЭИ»
г. Москва, Россия

МЕТРИКИ В ЗАДАЧАХ КЛАССИФИКАЦИИ

Одним из важнейших этапов в задачах машинного обучения является оценка качества обученных моделей машинного обучения. Именно на этом этапе удается установить, насколько удачно удалось построить модель. Без выполнения этапа оценки качества решение какой-либо задачи машинного обучения просто невыполнимо. Стоит сразу отметить, что в задачах классификации и в задачах восстановления регрессии используются абсолютно разные метрики, так как эти два типа задач имеют также абсолютно разную природу: в задачах классификации множество возможных ответов является конечным, в то время как в задачах восстановления регрессии оно бесконечно.

Пусть имеется некоторое множество объектов X , содержащее в себе вектора некоторого n -мерного векторного пространства, а также конечное множество возможных ответов $Y = \{1, 2, \dots, M\}$. Пусть также имеется обученная модель классификации, то есть задана функция $f: X \rightarrow Y$. Требуется оценить качество построенной модели классификации.

Как известно, обучить модель классификации так, чтобы она не совершала ошибок, является задачей невозможной. Для любого класса помимо верных результатов, когда объект, принадлежащий классу, был классифицирован как относящийся к нему, или объект, к классу не принадлежащий, был классифицирован как относящийся к другому классу, возможны ложноположительные и ложноотрицательные результаты, или ошибки I и II рода соответственно. В случае ложноположительного результата модель классификации отнесла к некоторому классу объект, который к нему не принадлежит. В случае ложноотрицательного результата объект, принадлежащий данному классу, к нему отнесен моделью классификации не был. Количество случаев, когда объект был верно отнесен к классу или, наоборот, верно отнесен к другому классу, обозначим TP (англ. true positive) и TN (англ. true negative) соответственно. Количество ложноотрицательных результатов обозначим FN (англ. false negative), а ложноположительных – FP (англ. false positive)

Все введенные выше величины в задаче бинарной классификации для классов с обозначениями 0 и 1 можно свести в матрицу ошибок (англ. confusion matrix) наподобие таблицы, где y – истинная метка класса, а \hat{y} – ответ, данный моделью классификации.

Таблица

Матрица ошибок

	$y = 1$	$y = 0$
$\hat{y} = 1$	TP	FP
$\hat{y} = 0$	FN	TN

Для случая мультиклассификации (количество классов больше, чем два) матрица ошибок разрастается и строится по такому правилу: на пересечении i -той строки и j -того

столбца стоит количество объектов, которые относятся к j -тому классу, но моделью классификации были отнесены к i -тому классу [5, с. 602].

Очевидно, что матрица ошибок является несовершенным способом оценки качества классификации. Во-первых, при достаточно большом количестве классов матрица ошибок становится большой и неудобной для анализа. Во-вторых, она не дает никакой количественной оценки качества, а именно такая оценка является необходимой для того, чтобы можно было однозначно сравнить две модели классификации.

Accuracy, precision, recall, F-мера

Одной из самых простых и интуитивно понятных метрик для задач классификации является доля правильных ответов, данных классификатором. В англоязычной литературе такая метрика получила название *accuracy*. Она определяется по формуле (1) [1, с. 93].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

При этом важно отметить одну особенность данной метрики: в случаях, когда классы не сбалансированы, данная метрика дает трудно интерпретируемые результаты. Рассмотрим пример: пусть имеется два класса, обозначаемых 0 и 1. Общее количество объектов в тестовой выборке равно 1000, причем на класс 0 приходится 900 объектов, а оставшиеся 100 объектов принадлежат классу 1. Предположим, что имеется некоторая модель классификации, которая отнесла все объекты тестовой выборки к классу 0. Тогда значение метрики *accuracy* равняется 0,9. Полученное при этом достаточно высокое значение метрики никоим образом не отражает того факта, что по сути имеющаяся модель классификации классификацию не выполняет, а всего лишь относит все объекты к самому многочисленному классу. Иными словами, результат обучения модели классификации признать удачным нельзя, а высокое значение метрики *accuracy* дает беспочвенное основание полагать, что обучение было выполнено успешно. Эта особенность делает данную метрику практически неиспользуемой в реальных задачах классификации, где, как правило, классы являются несбалансированными. Для решения проблемы оценки качества классификации в условиях несбалансированной выборки используются метрики *precision* и *recall*, определяемые формулами (2) и (3) соответственно. Данные метрики вычисляются отдельно для каждого класса [3, с. 179].

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

При этом указанные метрики можно понятным образом интерпретировать. Так, например, метрика *precision* – это доля объектов, которые действительно принадлежат классу, из числа объектов, которые были к этому классу отнесены моделью классификации. Метрику же *recall* можно интерпретировать как долю объектов, которые были отнесены к данному классу из числа всех объектов, на самом деле данному классу принадлежащих. То есть, иными словами, *precision* оценивает способность модели классификации отличить объекты, принадлежащие к классу от объектов, принадлежащих другим классам, а метрика *recall* оценивает способность модели максимально полно опознать данный класс.

Нетрудно видеть, что в примере выше для класса 0 $precision = 0,9$ и $recall = 1$, а для класса 1 $precision = 0$ и $recall = 0$. Анализируя эти результаты, можно заключить, что модель классификации хорошо работает с классом 0 и при этом абсолютно не различает класс 1. То есть, используя указанные выше метрики, можно получить результат, благодаря которому можно действительно правильно оценить качество построенной модели классификации. Такой результат объясняется тем, что метрики $precision$ и $recall$ не зависят от соотношения числа элементов в классах и поэтому применимы в том числе для несбалансированных выборок.

В разных задачах более важное значение может иметь либо $precision$, либо $recall$. Например, если задача состоит в определении лиц, инфицированных некоторым опасным вирусом, то более важной здесь представляется метрика $recall$, так как мы хотим определить максимальное число инфицированных лиц. В ряде других случаев стремятся минимизировать ложноположительные срабатывания, для чего наибольшее внимание уделяют метрике $precision$. Комбинированной метрикой, совмещающей в себе метрики $precision$ и $recall$, является F-мера, определяемая по формуле (4), где параметр β имеет смысл веса метрики $precision$. Частным случаем F-меры является F1-мера, для которой $\beta = 1$ [4, с. 14].

$$F_{\beta} = (1 + \beta^2) \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \quad (4)$$

AUC-ROC и AUC-PR

Ряд классификаторов дают вещественный ответ, а именно вероятность принадлежности к некоторому классу. Так, например, логистическая регрессия показывает значение вероятности принадлежности объекта к классу с меткой 1. При этом необходимо предварительно задаться некоторым порогом, для которого значения вероятности, превышающие порог, показывают принадлежность элемента к классу 1, а значения меньшие порога свидетельствуют о принадлежности к классу 0.

Наиболее простым решением при этом представляется выбрать в качестве порога значение 0,5. Однако это не всегда удачное решение, особенно для задач с несбалансированными классами. Возникает вопрос: как можно оценить модель классификации, не привязываясь к конкретному значению порога? Для этого может использоваться метрика AUC-ROC (площадь под кривой ошибок, англ. area under curve - receiver operating characteristic curve). Она представляет из себя кривую в координатах $x = FPR$ (англ. false positive rate) и $y = TPR$ (англ. true positive rate), которые вычисляются с использованием формул (5) и (6) соответственно [2, с. 94]. Данная кривая всегда соединяет точку с координатами (0,0) и точку с координатами (1,1). TPR совпадает с метрикой $recall$, в FPR равна доле объектов из другого класса, которые были предсказаны неверно. Обе данные величины увеличиваются при увеличении порога.

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

Так, например, если бы модель классификации не допускала ошибок, $FPR = 0$ и $TPR = 1$. В таком случае площадь под кривой оказалась бы равной 1. Если же модель классификации случайным образом выдает значения вероятностей принадлежности объектов (так называемое «случайное гадание»), площадь под кривой ошибок будет стремиться к значению 0,5. На рисунке 1 представлена кривая ошибок для классификатора, обладающего достаточно высоким качеством классификации, а на рисунке 2 – кривая ошибок для «случайного гадания» (источник изображений: <https://clck.ru/dCex3>).

Таким образом, чем выше оказывается качество классификатора, тем выше оказывается площадь под кривой ошибок. Каждая точка на графике кривой ошибок соответствует некоторому порогу. AUC-ROC более устойчив в случае несбалансированных классов, так как нечувствителен к масштабу. Площадь под кривой ошибок можно также интерпретировать как вероятность того, что случайно выбранный объект класса 1 получит более высокую вероятность принадлежности к классу 1, чем случайно выбранный объект класса 0. Иногда, впрочем, нечувствительность к масштабу является нежелательной, из-за чего AUC-ROC нельзя признать совершенной.

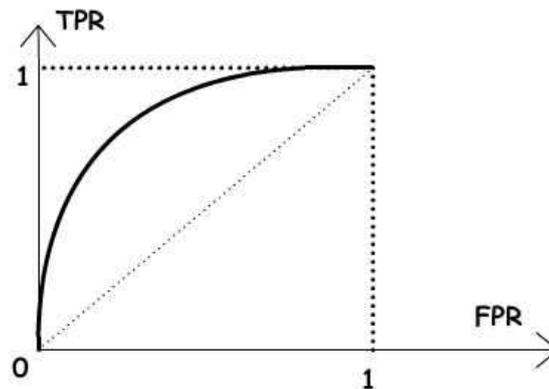


Рис. 1. Кривая ошибок для классификатора с достаточно высоким качеством классификации (<https://clck.ru/dCex3>)

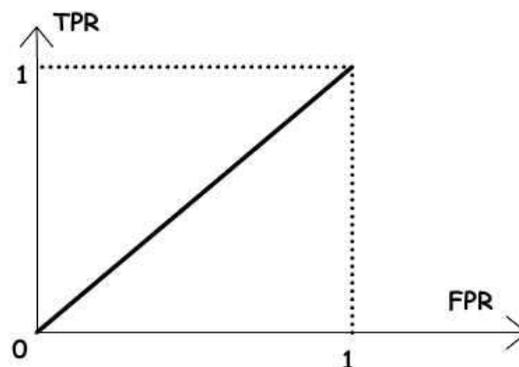


Рис. 2. Кривая ошибок для «случайного гадания» (<https://clck.ru/dCex3>)

Всякая задача машинного обучения требует этапа оценки качества, без которого решение задачи не имеет смысла. Помимо того, что задачи классификации и восстановления регрессии допускают совершенно разные наборы метрик, каждая задача требует индивидуального подбора метрик. Выбор метрик для конкретной задачи должен основываться на глубоком анализе предметной области и учете специфики используемых

данных. В данной работе был дан анализ основных метрик классификации. При этом, безусловно, данный анализ нельзя считать исчерпывающим. Однако именно обозначенные выше метрики классификации являются наиболее широко употребляемыми на практике.

Литература

1. Батура Т.В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30. №1. С. 85–99.
2. Богданов Л. Ю. Оценка эффективности бинарных классификаторов на основе логистической регрессии методом ROC-анализа // Вестник Саратовского государственного технического университета. 2010. Т. 4. №2(50). С. 92-97.
3. Чельшев Э.А., Оцоков Ш.А., Раскатова М.В. Автоматическая рубрикация текстов с использованием алгоритмов машинного обучения // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2021. №4. С. 175-182. <https://doi.org/10.25586/RNU.V9187.21.04.P.175>
4. Чельшев Э.А., Оцоков Ш.А., Раскатова М.В. Разработка информационной системы для автоматической рубрикации новостных текстов // Международный журнал информационных технологий и энергоэффективности. 2021. Т. 6. №3(21). С. 11-17.
5. Vujovic Z. Classification Model Evaluation Metrics // International Journal of Advanced Computer Science and Applications. 2021. Vol. 12. P. 599-606.

© Чельшев Э.А., Раскатова М.В., 2022

УДК 004.728.8

Челышев Э.А., Шибитов Д.В., Оцоков Ш.А., д-р техн. наук
Национальный исследовательский университет «МЭИ»
г. Москва, Россия

ПРОТОКОЛ MQTT И ЕГО ПРИМЕНЕНИЕ В ИНТЕРНЕТЕ ВЕЩЕЙ

В последнее время все большую популярность приобретают технологии автоматизации различных процессов, имеющих место как на производстве, так и в быту. Например, такое понятие, как «умный дом», все больше проникает в нашу жизнь. Полноценная автоматизация быта или производства невозможна без использования достаточно большого количества разнообразных устройств (датчики, измерители, манипуляторы и т.д.).

При этом возникает проблема обеспечения возможности взаимодействия данных устройств между собой или с сервером, который может управлять их работой (так называемое межмашинное взаимодействие). Для решения данной проблемы была разработана концепция интернета вещей (англ. IoT, Internet of Things). Данная концепция предполагает, что ряд устройств объединяются в единую сеть, после чего происходит объединение таких сетей и так далее в сторону увеличения размера сети.

Для обеспечения взаимодействия территориально распределенных устройств может использоваться сеть Интернет, однако очевидно, что помимо стандартного стека протоколов TCP/IP необходимо использование специфичных протоколов прикладного уровня, обеспечивающих передачу сообщений от устройства к устройству. Каковы требования к этим протоколам?

Прежде всего, стоит учесть, что сообщения, передаваемые между устройствами обычно невелики по объему и содержат, как правило, измерения или настройки. Во-вторых, важно понимать, что такие сообщения могут передаваться часто и в большом количестве. В-третьих, количество устройств, особенно, если говорить о промышленном интернете вещей, достаточно велико, а устройства могут быть различными по своему типу (датчики температуры, давления, метана; манипуляторы и т.д.). В-четвертых, канал связи может иметь существенные ограничения по пропускной способности, так как создание канала с высокой пропускной способностью для каждого устройства оказывается экономически нецелесообразным. Наконец, важно осознавать, что отдельные устройства могут приходить в негодность или временно оставаться неработоспособными из-за сбоев в программном обеспечении, отключения питания и иных неблагоприятных условий. Кроме того, могут иметь место неполадки в канале передачи данных. Таким образом, хороший протокол межмашинного взаимодействия должен учитывать все вышеуказанные особенности.

Существуют различные протоколы межмашинного взаимодействия. К ним можно отнести такие протоколы, как AMQP, MQTT, REST, XMPP и другие [6, с. 18]. В данной работе подробно будет рассмотрен протокол MQTT.

Протокол MQTT и его схема работы

MQTT (англ. Message Queuing Telemetry Transport) – протокол прикладного уровня для обмена данными, подходящий для передачи небольших сообщений по каналу с ограниченной

пропускной способности. Схема работы протокола MQTT представлена на рисунке 1. В процессе передачи участвуют три субъекта: издатель, брокер и подписчик. Издатель – устройство, передающее некоторые данные, предназначенные для приема подписчиком. Брокер – некоторый сервер со статическим IP-адресом [2, с. 487]. На данный момент можно как арендовать платные брокеры, так и использовать бесплатные.



Рис. 1. Схема протокола MQTT (<https://clck.ru/dWStB>)

Передача сообщения от издателя брокеру происходит в три этапа. На первом этапе издатель отправляет брокеру запрос на подключение. По мере готовности принять сообщение брокер отправляет издателю подтверждение подключения. В ответ издатель может направлять брокеру сообщения, указывая при этом конкретную тему сообщения – топик (англ. topic). Данный процесс называется публикация. Взаимодействие брокера и подписчика происходит в 4 этапа: подписчик направляет брокеру запрос на подключение, брокер отвечает подтверждением подключения. Затем подписчик направляет брокеру указание, сообщения какого топика он планирует получить. Этот процесс называется подписка. В ответ брокер осуществляет публикацию сообщений выбранного топика, то есть передает подписчику адресованные ему сообщения, которые помечены как относящиеся к данному топика [7].

Топик представляет собой набор символов в кодировке UTF-8. Топики имеют иерархическую организацию в виде деревьев, что делает работу с ними и доступ к данным проще. Топик может включать в себя один или несколько уровней, разделенных символом «/». Кроме того, подписчик может получать сообщения, относящиеся сразу к нескольким топикам. За счет вышеуказанной схемы подписчик может подписаться на конкретный топик и получать сообщения данного топика от множества издателей. Сообщение протокола MQTT состоит из трех составляющих: фиксированный заголовок, переменный заголовок (может отсутствовать) и данные. Формат фиксированного заголовка протокола MQTT представлен на рисунке 2.

Бит	7	6	5	4	3	2	1	0
Байт 1	Тип сообщения				DUP	QoS		Retain
Байт 2	Длина оставшейся части							

Рис. 2. Формат фиксированного заголовка протокола MQTT

Поле тип сообщения состоит из 4 бит и указывает принадлежность сообщения к некоторому типу, например, подключение (англ. CONNECT), публикация (англ. PUBLISH),

подписка (англ. SUBSCRIBE) и др. Всего в протоколе MQTT существует 15 типов сообщений, еще один тип дополнительно зарезервирован.

Следующие три поля являются специальными флагами. Флаг DUP, когда клиент или брокер направляют сообщение повторно. Флаг QoS равен значению уровня обслуживания, используемому в данной транзакции. Подробнее уровни обслуживания рассмотрены в этой статье ниже. Установленный флаг Retain сигнализирует брокеру о необходимости сохранения сообщения. При следующей подписке подписчика на тот же топик брокер направит сохраненное сообщение. Поле длины оставшейся части указывает длину сообщения, включающую в себя размер переменного заголовка и полезной нагрузки. Переменный заголовок включается в сообщение по необходимости и включает в себя идентификатор сообщения (англ. Packet ID), название протокола (англ. Protocol name) и его версию (англ. Protocol version), которые используются только в сообщениях типа CONNECT, а также ряд флагов: флаг наличия имени пользователя, флаг наличия пароля и другие.

Формат той части сообщения, которая содержит полезную нагрузку, протоколом не регламентируется и определяется в приложении. То есть разработчик сам в праве хранить в этой части столько байт, сколько посчитает нужным, и в таком формате, который наиболее удобен для приложений, работающих с сообщениями данного протокола. Также стоит отметить, что размер данных может быть вычислен как разность значения длины оставшейся части и длины переменного заголовка.

Особенности протокола MQTT

Данный протокол имеет ряд особенностей, а именно: асинхронность, способность работать в условиях нестабильной связи, поддержка нескольких уровней качества обслуживания и простота интеграции новых устройств. Асинхронность. Как уже было сказано, передача сообщений происходит не напрямую, а через брокера. При этом между отправкой сообщения и его приемом может пройти существенное время. Именно асинхронное взаимодействие позволяет осуществлять надежную передачу в условиях нестабильной связи. Связь может оказаться нестабильной по разным причинам. Во-первых, сам канал связи может не справиться с нагрузкой либо сетевое оборудование может прийти в негодность. Во-вторых, сами устройства, участвующие в обмене сообщениями, могут оказаться временно неработоспособными в силу ряда причин.

Так, например, если по какой-либо причине (отключение питания или выход из строя) подписчик не может принимать сообщения, то брокер будет аккумулировать адресованные сообщения в очереди, а затем, после того, как штатная работа подписчика восстановится, передаст сообщения из очереди подписчику. Аналогично, если в данное время подписчик подписан на некоторый топик, то сообщения, приходящие к нему и имеющие другие топики, также будут собраны на стороне брокера, ожидая соответствующей подписки [3, с. 77-76].

Поддержка нескольких уровней качества обслуживания. Протокол MQTT поддерживает три уровня качества обслуживания (англ. QoS, Quality of Service). На уровне 0, который еще называют At most once (рус. «максимум единожды») издатель единожды направляет сообщение брокеру и после этого не ожидает какого-либо подтверждения от брокера.

Уровень обслуживания 1 (At least once, рус. «по крайней мере единожды») предполагает, что брокер, получив сообщение, ответит издателю подтверждением. Однако, если подтверждение брокера не дойдет до издателя, то в свою очередь направит сообщение повторно, что приведет к дублированию. В ряде случаев дублирование может оказаться нежелательным. Уровень обслуживания 2 (Exactly once, рус. «ровно единожды») гарантирует доставку сообщений и исключает их дублирование. На этом уровне обслуживания передача сообщения от издателя брокеру состоит из 4 этапов (рис. 3).

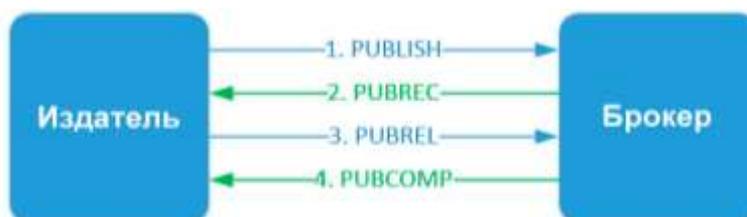


Рис. 3. Передача сообщений на уровне обслуживания 2 (<https://clck.ru/dWStB>)

При отправке сообщения (этап 1) брокеру издатель указывает в сообщении уникальный идентификатор Packet ID. Получив сообщение, брокер отправляет издателю подтверждение получения сообщения PUBREC с указанием Packet ID. Получив данное сообщение, издатель отправляет брокеру сообщение PUBREL с тем же Packet ID, после получения которого брокер отправляет издателю сообщение PUBCOMP о завершении транзакции. Простота интеграции новых устройств. Добавление в сеть нового устройства не требует существенных изменений в сети или ее перенастройки.

Также стоит отметить, что протокол MQTT предполагает также некоторые методы защиты данных, а именно аутентификацию клиентов с использованием имени пользователя и пароля, контроль доступа клиентов через Client ID и шифрование соединения с брокером при помощи протоколов TLS и SSL.

Применение протокола MQTT

На данный момент протокол MQTT поддерживают многие устройства, задействованные в промышленном интернете вещей. Это контроллеры, измерители угарного и углекислого газа, устройства термометрии и др. Уже сегодня ведутся разработки по применению данного протокола в нефтегазовой отрасли с целью создания эффективной системы управления технологическими процессами в производстве [4, с. 90-91]. Протокол MQTT используется также для мониторинга параметров объектов и устройств инфраструктуры железных дорог [1, с. 59]. Протокол MQTT также нашел применение в автоматизации энергетической отрасли [5].

Благодаря своим достоинствам протокол MQTT находит применение в автоматизации производственных процессов и быта. Данный протокол обеспечивает надежный и быстрый обмен сообщениями даже в условиях нестабильной связи, что делает его весьма полезным в межмашинном взаимодействии в рамках интернета вещей. Повсеместное быстрое развитие технологий интернета вещей делает протокол MQTT перспективным направлением информационных технологий.

Литература

1. Базакин И.А., Годяев А.И., Меркулов А.В. Использование беспроводных технологий LTE в режиме Nb-IoT в системах мониторинга параметров объектов и устройств железнодорожной инфраструктуры // Вестник транспорта Поволжья. 2019. № 5(77). С. 59-64.
2. Жарлыкасова А.Н., Жарлыкасов Б.Ж., Муслимова А.З. Модель удаленного управления с использованием протокола MQTT // Наука. Информатизация. Технологии. Образование: Мат-лы XI международной научно-практической конференции (г. Екатеринбург, 26 февраля – 02 марта 2018 года). Екатеринбург, 2018. С. 485-491.
3. Жарлыкасова А.Н., Муслимова А.З. Сравнительный анализ протоколов передачи данных и преимущества протокола MQTT // Тенденции развития современного естествознания и технических наук: Сб. научных трудов по мат-ам Международной научно-практической конференции (г. Белгород, 30 октября 2017 года). Белгород, 2017. С. 75-78.
4. Мирошников Р.С., Цамаев А.Р. Возможности применения протокола MQTT в нефтегазовой отрасли // Будущее науки-2019: сб. научных статей 7-й Международной молодежной научной конференции (г. Курск, 25–26 апреля 2019 года). Курск, 2019. С. 89-92.
5. Оцоков Ш.А., Челышев Э.А., Шибитов Д.В., Раскатова М.В. Разработка программно-аппаратного комплекса сбора и хранения данных термометрии // Инженерный вестник Дона. 2022. № 2. <http://www.ivdon.ru/ru>
6. Селезнев С.П., Яковлев В.В. Архитектура промышленных приложений IoT и протоколы AMQP, MQTT, JMS, REST, CoAP, XMPP, DDS // International Journal of Open Information Technologies. 2019. Т. 7. № 5. С. 17-28.
7. Mishra B., Kertész A. The Use of MQTT in M2M and IoT Systems: A Survey // IEEE Access. 2020. Vol. 8. pp. 201071-201086. doi: 10.1109/ACCESS.2020.3035849

©Челышев Э.А., Шибитов Д.В., Оцоков Ш.А., 2022

УДК 004.415

Школина А.В

Ростовский государственный экономический университет (РИНХ)
г. Ростов-на-Дону, Россия

РАЗРАБОТКА ANDROID ПРИЛОЖЕНИЯ ПРИ ПОМОЩИ ФРЕЙМВОРКА JETPACK COMPOSE

Инновации в разработке мобильных приложений часто приходят волнами или тенденциями. Для разработчика очень важно идти в ногу с тенденциями, независимо от того, связаны ли они с улучшением производительности платформы или безопасностью и стабильностью языка, так как некоторые из них способны кардинально изменить подходы к разработке приложений. Последние пару лет наблюдается переход от классического императивного подхода MVC/P (Model-View-Controller/Presenter) к более новому подходу организации кода MVVM (Model-View-ViewModel). Это связано с тем, что в декларативном программировании более наглядное и простое формулирование задач, что позволяет полностью отказаться от разметки и писать единый код без разделения на разметку и основной код [2, с. 112].

Данный подход реализуется с помощью Jetpack Compose – разработки компании Google для ОС Android, которую впервые анонсировали в 2019 году. Jetpack Compose — их ответ на тенденцию декларативной среды пользовательского интерфейса, которую разрабатывает команда Android. С помощью данного фреймворка можно коренным образом изменить способ создания пользовательского интерфейса разработчиками, упрощая и ускоряя его написание и повышая производительность в работе. Он является частью набора библиотек Jetpack и, как таковой, должен обеспечивать совместимость со всеми версиями платформы, устраняя необходимость избегать определенных функций при работе с более дешевыми устройствами или более старыми версиями Android.

Целью данной статьи является знакомство и разбор основных концепций Jetpack Compose в процессе создания пользовательского интерфейса приложения для обмена сообщениями (<https://clck.ru/dYfQ3>).

При создании проекта в интегрированной среде разработки Android Studio проект по умолчанию будет содержать файл под названием MainActivity.kt, который содержит одноименный класс MainActivity. Данный класс определяет интерфейс, который можно будет увидеть при запуске проекта либо на виртуальном устройстве – эмуляторе, либо на физическом устройстве.

MainActivity наследуется от встроенного класса ComponentActivity. ComponentActivity обеспечивает построение интерфейса из визуальных компонентов и для этого предоставляет минимальный функционал. В частности, предоставляет метод onCreate(), который вызывается при запуске приложения и создает интерфейс этого приложения. В метод передается объект Bundle, который хранит состояние MainActivity - некоторые значения, которые хранят связанные с классом данные. А в самом методе onCreate() вначале вызывается реализация этого метода из базового класса ComponentActivity. Для создания

интерфейса в `onCreate()` вызывается другой унаследованный от `ComponentActivity` метод `setContent()`. Этот метод и определяет, какой интерфейс мы увидим на экране устройства. В этот метод передается объект `@Composable`.

Ключевой концепцией данного фреймворка является `composable`-функция. Это стандартная функция, содержащая аннотацию `@Composable`, которая дает понять компилятору, что используется декларативный подход. Такие функции представляют некоторые части визуального интерфейса, из которых строится приложение. Это упрощает построение и обновление сложных интерфейсов, тестирование и поддержку самих компонентов [1, с. 19].

Сразу после создания проекта файл `MainActivity.kt`, помимо одноименного класса и определенных в нем методов, будет по умолчанию содержать `composable`-функции `Greeting()` и `DefaultPreview()`. Первая функция вызывает функцию `Text()`, которая определяется библиотекой `Compose UI` — данная функция вызывается для объявления текстовый элемент в приложении. Функция `DefaultPreview()` имеет аннотацию `@Preview`, что позволяет предварительно просматривать составные функции прямо в IDE, вместо того, чтобы загружать приложение на устройство `Android` или в эмулятор, а также вызывает функцию `Greeting` с соответствующим параметром. Определение функций и их синтаксис представлены на рисунке 1:

```
@Composable
fun Greeting(name: String) {
    Text(text = "Hello $name!")
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    DuplicateComposeTheme {
        Greeting(name = "Android")
    }
}
```

Рис. 1. Функции по умолчанию

После сборки проекта приложение не поменяется, так как функция `DefaultPreview()` нигде не вызывается, но появится окно предварительного просмотра, представленное на рисунке 2. В данном окне отображаются элементы пользовательского интерфейса, созданные `composable`-функцией, отмеченной аннотацией `@Preview`. Для обновления предварительного просмотра достаточно нажать кнопку обновления в верхней части окна.

Для отображения в приложении имени отправителя и содержания сообщения изменим имя функции `Greeting()` и `DefaultPreview()` на `MessageCard` и `PreviewMessageCard` соответственно, а также сигнатуру функции, которая теперь в качестве параметра будет принимать объект `Message`. В функцию `MessageCard()` добавляется второй текстовый элемент. Теперь будут отображаться два текстовых элемента, но так как не было предоставлено никакой информации об их расположении, то элементы начнут накладываться друг на друга, что делает текст нечитаемым.

В целях решения данной проблемы в Jetpack Compose используются контейнеры компоновки, которые позволяют описать расположение элементов. Контейнер Column позволяет размещать объекты вертикально. Добавим его в функцию MessageCard(), чтобы текстовые элементы располагались последовательно в столбик.

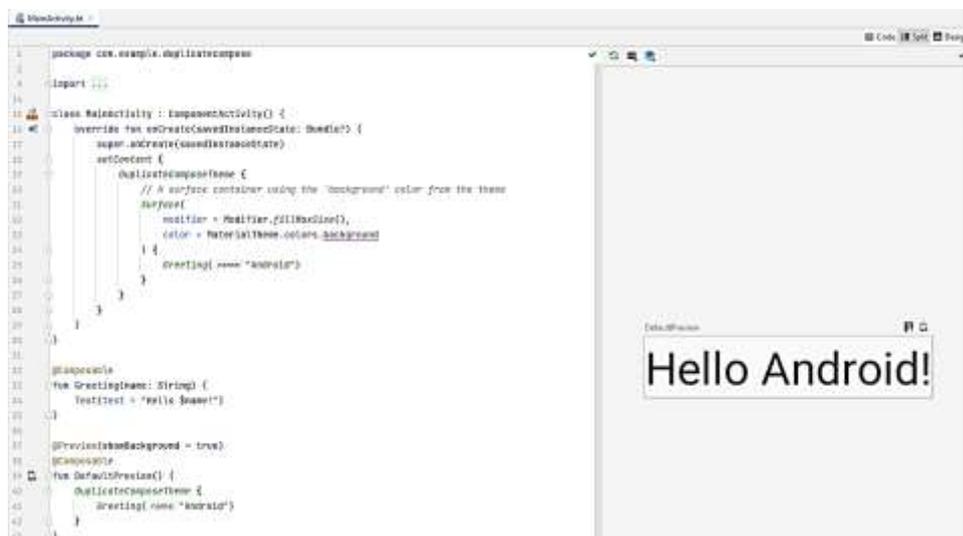


Рис. 2. Окно предварительного просмотра

Для добавления картинки в качестве фотографии собеседника используется компонент Image. В данном случае определяется два параметра – painter и contentDescription. Первый параметр определяет имя используемого изображения, а второй представляет строковое описание для изображения. Добавим компонент Image и контейнер Column в контейнер компоновки Row, чтобы элементы располагались последовательно в строку.

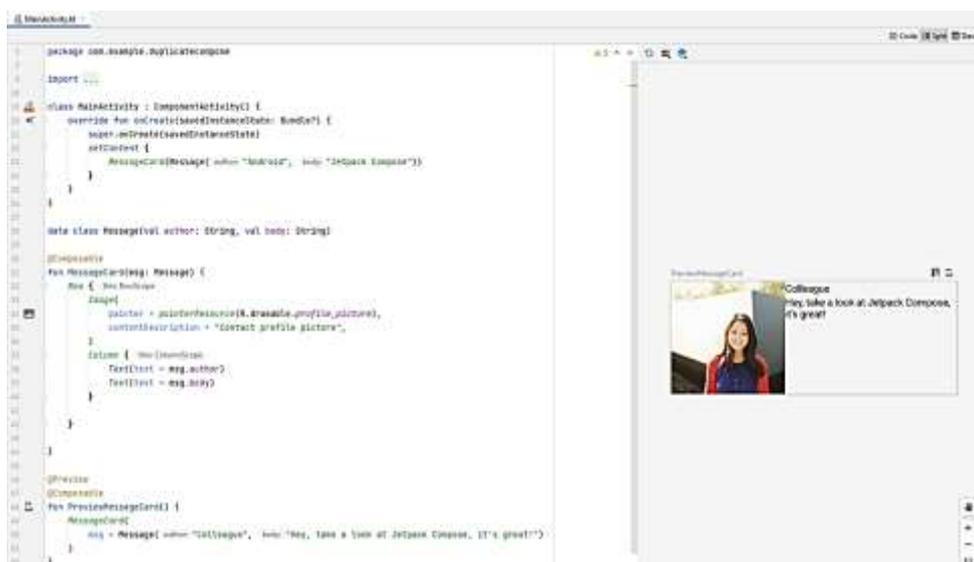


Рис. 3. Первая версия UI приложения

Для настройки внешнего вида всех компонентов в Jetpack Compose применяются модификаторы. Модификаторы представляют собой функции, которые задают определенный аспект для компонентов или иными словами «модифицируют» внешний вид компонента.

Например, установка размеров компонента или его фонового цвета. Определим модификаторы для картинки, изменив ее размер и форму, а также отступы между элементами.

Jetpack Compose обеспечивает реализацию Материального дизайна и его элементов пользовательского интерфейса. Для улучшения внешнего вида тестового приложения можно использовать стилизацию Материального дизайна, которая состоит из трех основных компонентов: цвет, типографика и форма. Изменение элементов интерфейса происходит за счет оберты функции `MessageCard` в `Material theme`, который был создан в проекте, и определения для каждого из них необходимых параметров.

В целях добавления в интерфейс приложения большего количества сообщений создан класс `Conversation`. Для более эффективного отображения элементов используются компоненты `LazyColumn` и `LazyRow`, которые предоставляют пользователю только те элементы, которые отображаются в окне просмотра. В качестве источника сообщений создан новый файл `SampleData.kt`, содержащий список из всех текстовых элементов.

Поскольку Jetpack Compose применяет декларативный подход, то единственным способом обновить визуальный компонент является повторный вызов функции этого компонента, в которую передаются новые значения. Чтобы упростить обновление компонентов и визуального интерфейса в целом, Jetpack Compose предоставляет концепцию состояния. Состояние представляет собой некоторое значение, которое хранится в приложении и в процессе его работы может изменяться. Чтобы контролировать изменение состояния, используются функции `remember` и `mutableStateOf`. `Composable`-функции могут сохранять локальное состояние в памяти с помощью `remember`, и отслеживать изменения значения, переданного `mutableStateOf`. При использовании API состояния фреймворка Jetpack Compose любые изменения состояния автоматически обновляют пользовательский интерфейс.

Для того чтобы при раскрытии сжатого сообщения менялся фон, используется модификатор `clickable` для обработки события нажатия на элемент. Функция `animateColorAsState` позволяет анимировать цвет фона, переключаясь между `MaterialTheme.colors.surface` и `MaterialTheme.colors.primary`, а функция `animateContentSize` позволяет плавно изменять размер контейнера сообщения. Финальный результат разработанного интерфейса представлен на рисунке 4:

В результате проделанной работы были исследованы основные инструменты Jetpack Compose для разработки Android приложения, изучена ключевая концепция данного фреймворка – `composable`-функции, а также понятие состояния, контейнеры компоновки, Материальный дизайн и другие инструменты для разработки пользовательского интерфейса приложения для обмена сообщениями. Всего за 100 строчек кода удалось создать лаконичный интерфейс, эффективно отображающий большое количество анимированных сообщений, содержащих изображение и текст.

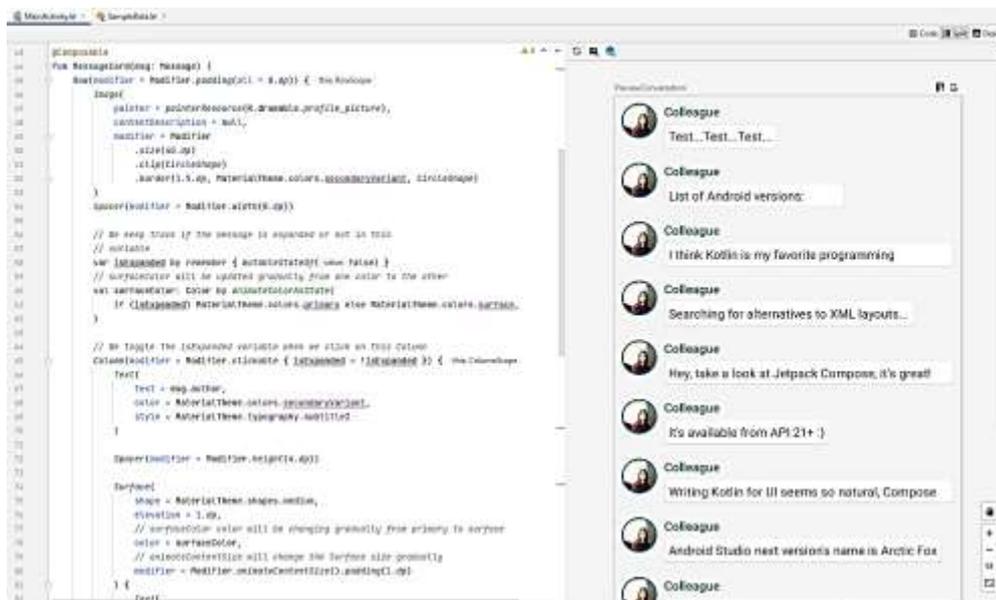


Рис. 4. Финальная версия UI приложения

Jetpack Compose предоставляет современный подход к созданию UI, позволяя эффективно распределять ответственность в коде, упрощает и ускоряет разработку пользовательского интерфейса на Android с помощью меньшего количества кода, мощных инструментов и интуитивно понятных API-интерфейсов Kotlin. Фреймворк совместим с любым уже существующим кодом. Многие библиотеки, такие как Jetpack Navigation, ViewModel и Coroutines адаптированы под Compose, что позволяет достаточно быстро внедрить его в проект. Чтобы начать работу с Jetpack Compose и использовать новейшие инструменты, которые предлагает Google, достаточно обновить новую версию Android Studio “Arctic Fox”, в которой теперь доступны такие функции, как Layout Inspector, Background Task Inspector, Wear OS Pairing и другие (<https://clck.ru/dYfJS>).

Исходный код приложения, размещенный на платформе GitHub (<https://clck.ru/etAWX>).

Литература

1. Jorge Castillo. Jetpack Compose internals: Leanpub, 2021. 121 с.
2. Иваненко К.М., Иванова Е.А. Сравнительный анализ императивного и декларативного программирования // Информационное общество: современное состояние и перспективы развития: сб. материалов X международного студенческого форума (г. Краснодар, 25–29 декабря 2017 года). Краснодар, 2018. С. 111-113.

© Школина А.В., 2022

ОСНОВНЫЕ ВОПРОСЫ РАЗРАБОТКИ СИСТЕМ ХРАНЕНИЯ СВЯЗЕЙ МЕЖДУ СЕРВИСАМИ

Современный рынок постоянно наполняется большим количеством IT-проектов. Для корректной работы внедряемых проектов необходимо интегрироваться друг с другом. С целью удобного взаимодействия проектов создан такой инструмент, как API. API – это веб-интерфейс для программного обеспечения. API, независимо от типа, прежде всего является интерфейсом: точкой, где две системы, субъекта, организации и т.д. встречаются и взаимодействуют [1, с. 31]. Например, когда пользователь в приложении на смартфоне выбирает функцию «узнать курс доллара» - мобильное приложение совершает API запрос на сервис Центрального банка. Так же приложение может обращаться к сервису, где хранятся данные, которые визуализируются в приложении. В данном случае, приложение так же будет выполнять API запрос. Схема взаимодействия приложения с сервисами представлена ниже на рисунке 1.

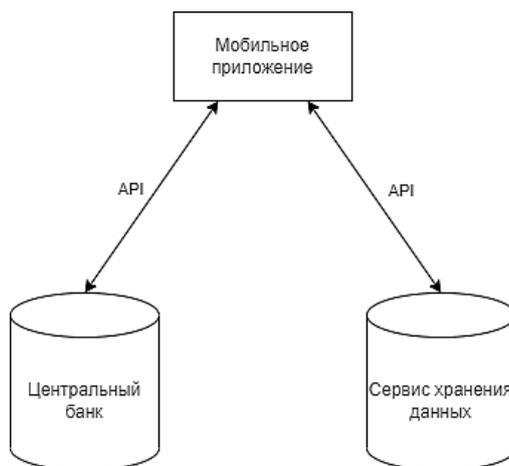


Рис. 1. Пример схемы взаимодействия приложения с сервисами

С помощью API сервисы могут обмениваться данными независимо от того, какой язык программирования использует тот или иной сервис. При обмене данные могут состоять из нескольких полей. Кроме того, полученные данные из одного сервиса могут преобразовываться и передаваться в другой, образуя цепочку, в которой изменение первого элемента повлечет за собой изменение остальных. Опишем на простом примере. Допустим, имеется у нас начальная точка «Мобильное приложение». В нем пользователь заполняет свои персональные данные. Эти данные, включая идентификатор пользователя отправляются в сервис «Автовладельцы», где создается карточка на пользователя и присваивается Идентификатор автовладельца. Сервис «Автовладельцы» совершает запрос и передает данные, включая связку «Идентификатор автовладельца и идентификатор пользователя» в WMS (систему управления складом). Используя данное приложение охранник на складе

может идентифицировать данного человека и автовладельца, которого он представляет. Но, в случае, когда вместо идентификатора пользователя будет использоваться другой способ идентификации пользователя, скажем уже не строка, число, то придется менять всю цепочку. Пример схемы передачи данных представлена на рисунке 2 ниже.



Рис. 2. Пример передачи данных по цепочке

Порой данная цепочка может оказаться настолько длинной и не одиночной, что проектировщик может упустить звено, что приведет к критическим ошибкам (например, задержка сдачи проекта). Следующим шагом в развитии использования сервисов является прогнозирование изменений на этапе проектирования при замене того или иного поля. Это во многом ускорит проектирование схемы взаимодействия с сервисами и позволит уменьшить количество ошибок. Так, например, если исключить (или изменить свойства поля), то необходимо, чтобы система сообщала пользователю о нарушении цепочки передачи данных, как показано на рисунке 3 ниже.



Рис. 3. Пример нарушения передачи данных по цепочке

Практическая значимость заключается во внедрении системы для работы сотрудников компании ООО «Умная Логистика» при проектировании разрабатываемых систем, использующих интеграцию с другими сервисами. Существуют системы, предоставляющие функции API Management. API Management — это процесс создания и публикации программных интерфейсов веб-приложений (API), обеспечения соблюдения их политик использования, контроля доступа, поддержки сообщества подписчиков, сбора и анализа статистики использования и отчетности о производительности. С другой стороны, API Management — это набор инструментов и сервисов, которые позволяют разработчикам и

компаниям создавать, анализировать, применять и масштабировать интерфейсы API в безопасных средах.

Среди достоинств данных систем можно отметить:

- аутентификация - система позволяет пройти аутентификацию, в том числе и через сторонние службы;
 - управление трафиком - система регулирует входящий и исходящий трафик API;
 - мониторинг API — система может помочь в мониторинге запросов/ответов клиента;
- преобразования – система позволяет преобразовать запросы/ответы API.

К минусам можно отнести:

- увеличение Latency — шлюзу необходимо время для обработки запросов/ответов согласно настроенным политикам.

Данные системы достаточно дорогостоящие и в комплекте поставки имеют множество других функций. Предпосылками для разработки новой системы хранения связей явились недостатки существующих систем, а именно: системы хранения встраиваются в проект, что ведет к дополнительным затратам и использованию лишних вычислительных мощностей; подключение избыточных функций, которые не используются, но их существование ведет к дополнительным затратам (<https://clck.ru/esyуM>). Перечисленные выше факторы способствовали принятию решения о разработке новой системы. Принцип работы «как должно быть» данной системы описан ниже в нотации IDEF0 на рисунке 4.

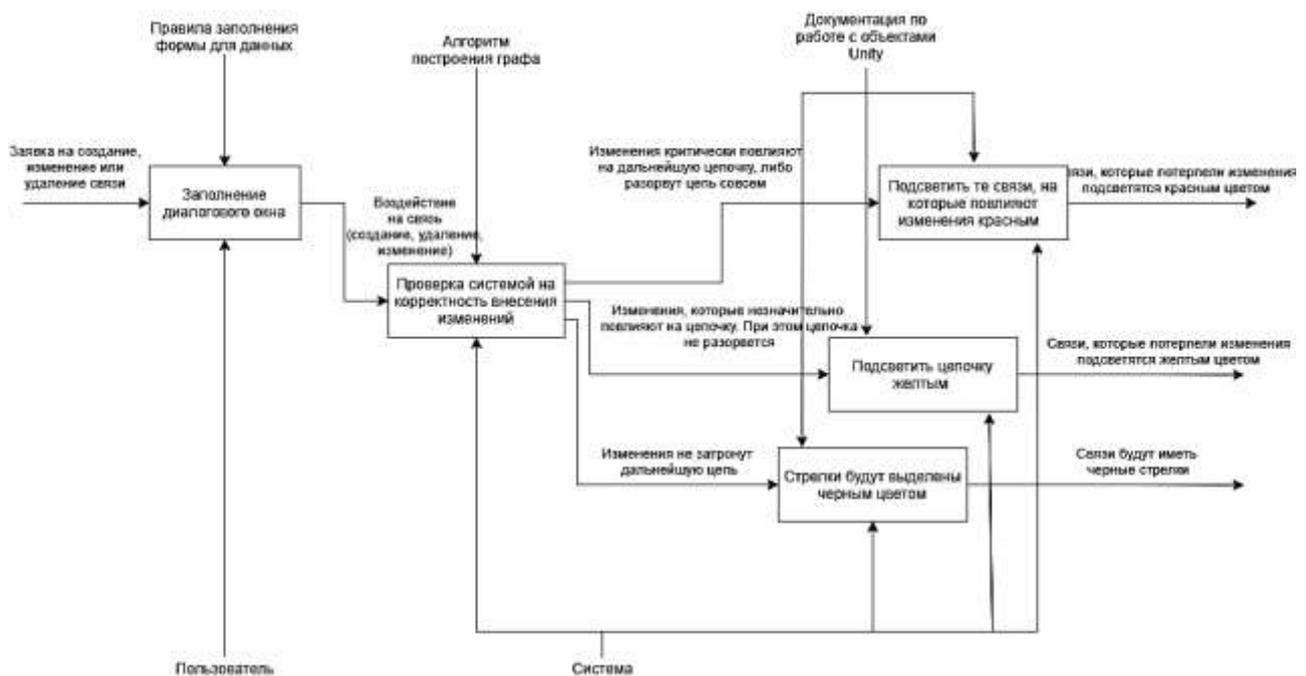


Рис. 4. Контекстная диаграмма работы системы для хранения связей API между сервисами

Опишем данную диаграмму. В качестве входного параметра выступает заявка на создание, изменение или удаление связи. В первом процессе участвует Пользователь. Используя полученную заявку и правила заполнения формы для данных, он настраивает

информацию в диалоговом окне свойства связи. Далее эту информацию принимает Система. Она, используя алгоритм проверки связей, проверяет цепочку связей на влияние внесенных изменений. В случае, если изменения критически повлияют на цепочку связей и дальнейшее использование будет невозможным, то эти связи, которые будут нарушены, выделятся красным цветом. В случае, если изменения не критически повлияют на цепочку, и данные будут передаваться без ошибок, то связи, которые потерпят изменения выделятся желтым цветом. В случае, если изменения никак не повлияют на цепочку – связи будут иметь черный цвет. На выходе будем иметь графическое представление схемы связей API между сервисами. Перед началом разработки заказчиком были поставлены следующие требования: возможность работы нескольких сотрудников одновременно над одним проектом в облаке; интуитивно понятный интерфейс; наличие тонкого клиента, т.е. для работы с системой не устанавливается ПО; разработанная система не должна занимать много места. Так же был предоставлен прототип того, как должна выглядеть разрабатываемая система, которая представлена на рисунках 5 и 6 ниже.

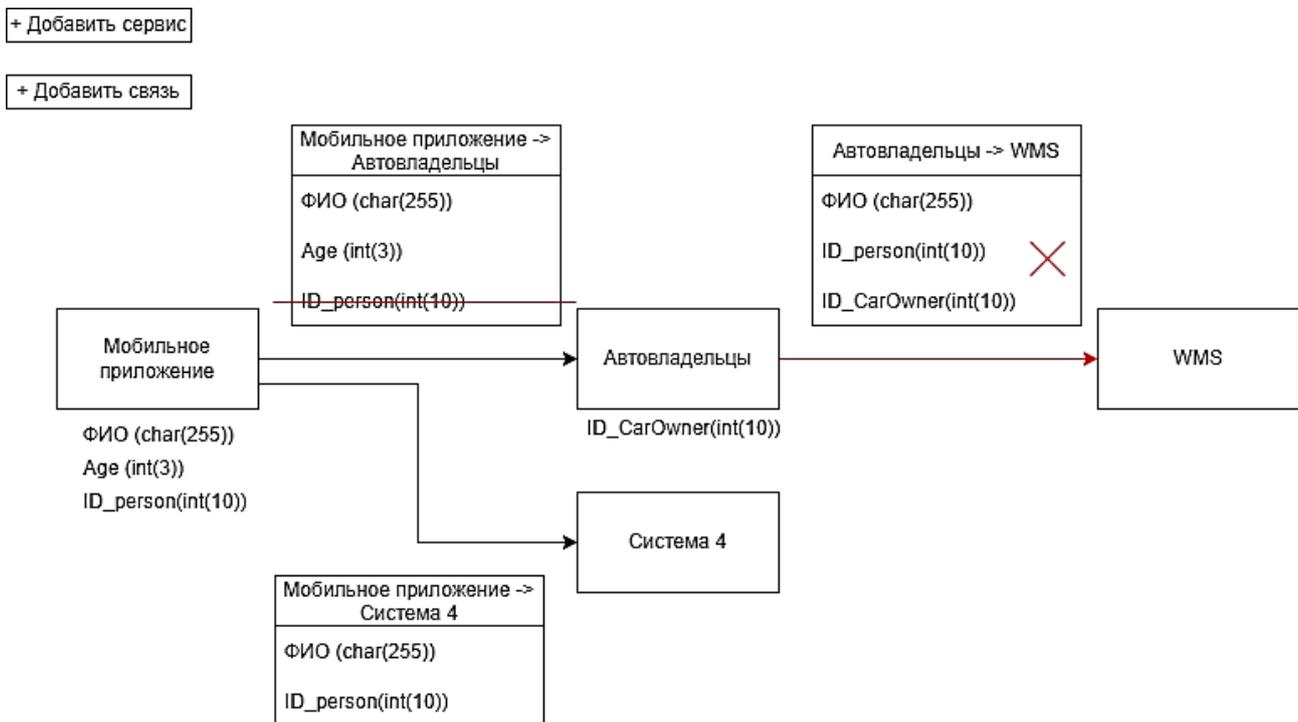


Рис. 5. Прототип интерфейса разрабатываемой системы

При нажатии на кнопку «Добавить связь» должна открываться форма заполнения данных о связи, которая изображена на рисунке 6. При нажатии на кнопку «Добавить сервис» будет открываться форма, где будет содержаться только одно поле для заполнения – Наименование сервиса.

Рис. 6. Прототип интерфейса меню добавления / изменения связи

Исходя из перечисленных требований, для разработки была выбрана платформа Unity с использованием языка программирования C#. Выбранная платформа позволит реализовать такие возможности, как: скомпилированный проект можно разместить на серверах, куда пользователи смогут подключиться для работы без установки дополнительного ПО; готовое приложение на Unity будет легче, по сравнению с Unreal Engine.

Литература

1. Арно Лоре. Проектирование веб-API. М.: ДМК Пресс, 2020. 440 с.

© Шмелев П.Ю., Прозорова Л.Ю., 2022

ОРГКОМИТЕТ

Председатель оргкомитета – Горлов Сергей Иванович, д-р физ.-мат. наук, профессор, ректор;

Заместитель председателя – Погоньшев Денис Александрович, канд. биол. наук, доцент, первый проректор, проректор по научной работе.

Члены оргкомитета

Шульгин Олег Валерьевич – канд. экон. наук, доцент, начальник управления научных исследований;

Долгина Екатерина Станиславовна – канд. культурологии, доцент, декан гуманитарного факультета;

Иванов Вячеслав Борисович – канд. пед. наук, доцент, декан факультета экологии и инжиниринга;

Павловская Анастасия Анатольевна – канд. пед. наук, доцент, декан факультета искусств и дизайна;

Худжина Марина Владимировна – канд. пед. наук, доцент, декан факультета информационных технологий и математики;

Истомина Ирина Павловна – канд. психол. наук, доцент, декан факультета педагогики и психологии;

Давыдова Светлана Александровна – канд. пед. наук, доцент, декан факультета физической культуры и спорта.

СОДЕРЖАНИЕ

Аверьянова И.Г., Лосева А.В., Башарина С.О.

**ПЛАТФОРМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ:
ПОПУЛЯРНОСТЬ И ФУНКЦИОНАЛ..... 4**

Андреева А.Ю.

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕРВИСОВ
ПО СОЗДАНИЮ ИНТЕРАКТИВНОГО ВЕБ-КОНТЕНТА..... 10**

Антонов Д.Г.

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: СЕГОДНЯ И ЗАВТРА,
МЕЧТЫ И РЕАЛЬНОСТЬ. АНАЛИЗ ФИЛОСОВСКИХ ПРОБЛЕМ
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА..... 15**

Байкова К.Д., Медведева Т.А.

**АНАЛИЗ ЭФФЕКТИВНЫХ СПОСОБОВ СОЗДАНИЯ
TELEGRAM-БОТОВ ДЛЯ ВНЕДРЕНИЯ В БИЗНЕС-ПРОЦЕССЫ..... 22**

Бакулин М.А.

УЗВИМОСТИ БИОМЕТРИЧЕСКИХ СИСТЕМ..... 22

Байкова К.Д., Медведева Т.А.

**АНАЛИЗ ЭФФЕКТИВНЫХ СПОСОБОВ СОЗДАНИЯ
TELEGRAM-БОТОВ ДЛЯ ВНЕДРЕНИЯ В БИЗНЕС-ПРОЦЕССЫ..... 27**

Бедняк С.Г., Захарова О.И., Федулова А.А., Кузнецова А.А.

ИСПОЛЬЗОВАНИЕ ЧАТ-БОТОВ..... 33

Борисова Е.В.

РАЗРАБОТКА КОМПЬЮТЕРНОЙ ЛОГИЧЕСКОЙ ИГРЫ «ДЕНЬ СТУДЕНТА» 37

Брагин К.А., Кушев В.О.

**МЕТОДИКА ВНЕШНЕГО ТЕСТИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ..... 41**

Витченко В.А.

**ИССЛЕДОВАНИЕ СПОСОБА РАСШИРЕНИЯ WEB-IDE
ПРОЦЕСС-ОРИЕНТИРОВАННОГО ЯЗЫКА REFLEX..... 46**

Гафетдинова Г.Х., Рахматуллина А.М., Федорова О.В.

**ПОЛЬЗА И ВРЕД ЗАЩИТЫ ИНФОРМАЦИИ
В СОЦИАЛЬНЫХ СЕТЯХ И МЕССЕНДЖЕРАХ..... 51**

Ермишина Е.В., Шарыпова Т.Н.

**МЕТОДИКА ВНЕШНЕГО ТЕСТИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ..... 59**

Захаров Д.М., Слива М.В.

**ИСПОЛЬЗОВАНИЕ ДВИЖКА ROASSAL
ДЛЯ ВИЗУАЛИЗАЦИИ СТАТИСТИЧЕСКИХ ДАННЫХ..... 64**

Заякина А.М. ПЕРСОНАЛИЗАЦИЯ ОБУЧЕНИЯ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ЧАТ-БОТОВ	68
Иванова В.Р., Шарифуллин Б.Р. ИНСТРУМЕНТЫ ИНТЕРНЕТА ВЕЩЕЙ ДЛЯ АВТОМАТИЗАЦИИ ВНУТРЕННИХ ПРОЦЕССОВ ОРГАНИЗАЦИЙ.....	76
Исаева Е.М., Моисеева Т.В., Поляева Н.Ю., Точильникова К.А. СТРУКТУРА СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО РАЗРЕШЕНИЮ ПРОБЛЕМНЫХ СИТУАЦИЙ	81
Карпова Т.С., Малышева С.Ю., Мун Д.Ч. АДАПТИВНАЯ ЧЕЛОВЕКО-МАШИННАЯ СИСТЕМА ФОРМИРОВАНИЯ ТЕХНИЧЕСКИХ ЗАДАНИЙ НА ДОРАБОТКУ И ИЗМЕНЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	85
Коноваленко О.В., Сиволобов С.В. УСТРОЙСТВО ДЛЯ ГОЛОСОВОГО УПРАВЛЕНИЯ БЫТОВОЙ ТЕХНИКОЙ.....	90
Коновской А.С., Нестеров С.В. МОДЕЛИРОВАНИЕ ВЫСОКОГОРНОЙ ОПТИКО-ЭЛЕКТРОННОЙ ЛИНИИ СВЯЗИ.....	95
Кривогузова М.В., Ерошенко И.А., Низовцева Ю.Я. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СФЕРЕ СПОРТА.....	101
Кучинская А.В. ЦИФРОВИЗАЦИЯ БОРЬБЫ С ЭКОНОМИЧЕСКИМИ ПРЕСТУПЛЕНИЯМИ.....	106
Лац Е.В., Кетько Н.В. ПОВЫШЕНИЕ ЭКОНОМИЧЕСКОЙ ГРАМОТНОСТИ ПОТЕНЦИАЛЬНЫХ ПРЕДПРИНИМАТЕЛЕЙ, С ИСПОЛЬЗОВАНИЕМ ОБУЧАЮЩЕЙ ЭКОНОМИЧЕСКОЙ ИГРЫ-ЭМУЛЯТОРА.....	112
Летон Г., Глазько П.Е. АНАЛИЗ ПРИЧИН ОТКАЗА ОТ ИСПОЛЬЗОВАНИЯ VIRTUAL DOM НА ПРИМЕРЕ SVELTE	119
Ликсина Е.В., Вьюнов Д.А. РАЗРАБОТКА МОБИЛЬНОГО НАВИГАТОРА ПО ЗДАНИЮ	124
Маляров Д.А., Волкова С.С. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ПРОГНОЗИРОВАНИЯ ИЗМЕНЕНИЯ ЦЕН АКЦИЙ НА МОСКОВСКОЙ БИРЖЕ РАЗЛИЧНЫМИ МОДЕЛЯМИ МАШИННОГО ОБУЧЕНИЯ.....	128
Массеров Д.Д. БЕЗОПАСНОСТЬ «УМНОГО ГОРОДА» В ПРОЦЕССЕ ЦИФРОВИЗАЦИИ ГОРОДСКОЙ СРЕДЫ.....	135

Муртазина Р.Р. АВТОМАТИЗАЦИЯ ПРОВЕРКИ ДОКУМЕНТОВ ФИЛИАЛА ОАО «РОССИЙСКИЕ ЖЕЛЕЗНЫЕ ДОРОГИ»	142
Новикова Э.А. КЛАССИФИКАЦИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	147
Платонова В.А. РАЗРАБОТКА ИНТЕРАКТИВНОГО WEB-ПРИЛОЖЕНИЯ «КОНСТРУКТОР САЙТА ПРЕПОДАВАТЕЛЯ»	152
Попова А.С. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ РАБОТЫ ТЕХНИЧЕСКОЙ СЛУЖБЫ	157
Райш А.В. ДИЗАЙН ЛИТЕРАЛОВ КОЛЛЕКЦИЙ ДЛЯ ЯЗЫКА KOTLIN	164
Сиразева А.Л., Зарипова Р.С. БЕЗОПАСНОСТЬ И КОНФИДЕНЦИАЛЬНОСТЬ FACEAPP И ТИКТОК	168
Соколова А.И. ВЛИЯНИЕ ИНФОРМАЦИОННОГО ОБЩЕСТВА НА РАЗНЫЕ СФЕРЫ ДЕЯТЕЛЬНОСТИ	174
Сорокин А.О., Слива М.В. СОЗДАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО УПРАВЛЕНИЮ УМНЫМ ДОМОМ НА ПЛАТФОРМЕ ANDROID	179
Тарасова А.Ю. СОВРЕМЕННЫЕ ТРЕНДЫ ТРАНСФОРМАЦИЙ ИНФОРМАЦИОННО- ТЕХНОЛОГИЧЕСКОГО ПРОСТРАНСТВА	182
Торопов В.В. АРХИТЕКТУРА WEB-ПРИЛОЖЕНИЙ	188
Тухватуллина Э.Э. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРИЕМА ЗАЯВОК ДЛЯ ФОРМИРОВАНИЯ РАСПИСАНИЯ УЧЕБНЫХ ЗАНЯТИЙ	188
Торопов В.В. АРХИТЕКТУРА WEB-ПРИЛОЖЕНИЙ	193
Тырышкина С.Ю. МЕДИАПРОЕКТЫ КАК СРЕДСТВО ФОРМИРОВАНИЯ ИНФОРМАЦИОННОЙ ГРАМОТНОСТИ ОБУЧАЮЩИХСЯ СОВРЕМЕННОЙ ШКОЛЫ	199
Фархутдинов Д.Д. РАЗРАБОТКА СЕРВЕРНОГО МОДУЛЯ IOT-ПЛАТФОРМЫ НА ОСНОВЕ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПОДХОДА	204

Федяева И.А. РАЗРАБОТКА МЕТОДА ОТСЛЕЖИВАНИЯ ЗАВИСИМОСТЕЙ ДЛЯ КЭША ВЫВОДА ТИПОВ СТАТИЧЕСКОГО АНАЛИЗАТОРА КОДА СРЕДЫ РАЗРАБОТКИ RUSCHARM.....	209
Филатов М.Е. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВИЗУАЛИЗАЦИИ ПУТЕШЕСТВИЙ.....	215
Фролова Т.А., Шевелева О.Г. СОВЕРШЕНСТВОВАНИЕ ТЕХНОЛОГИИ СИСТЕМЫ РАСПОЗНАВАНИЯ ЛИЦ ДЛЯ ДВЕРНЫХ ЗАМКОВ.....	220
Харитонов Л.С. РАЗРАБОТКА GUI-ПРИЛОЖЕНИЯ ДЛЯ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ С ПОМОЩЬЮ C++/CLI WINDOWS FORMS	227
Хвостова А.В. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ СОЗДАНИЯ И ПРОВЕДЕНИЯ КВИЗОВ	233
Хджейра М., Хамитов Р.М. ГЕОИНФОРМАЦИОННАЯ СИСТЕМА КАК СПОСОБ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ И ЭФФЕКТИВНОСТИ РАСПРЕДЕЛЕННЫХ ЭЛЕКТРИЧЕСКИХ КОМПЛЕКСОВ В СИРИЙСКОЙ АРАБСКОЙ РЕСПУБЛИКЕ	238
Чельшев Э.А., Раскатова М.В. МЕТРИКИ В ЗАДАЧАХ КЛАССИФИКАЦИИ	242
Чельшев Э.А., Шибитов Д.В., Оцоков Ш.А. ПРОТОКОЛ MQTT И ЕГО ПРИМЕНЕНИЕ В ИНТЕРНЕТЕ ВЕЩЕЙ	247
Школина А.В. РАЗРАБОТКА ANDROID ПРИЛОЖЕНИЯ ПРИ ПОМОЩИ ФРЕЙМВОРКА JETPACK COMPOSE	252
Шмелев П.Ю., Прозорова Л.Ю. ОСНОВНЫЕ ВОПРОСЫ РАЗРАБОТКИ СИСТЕМ ХРАНЕНИЯ СВЯЗЕЙ МЕЖДУ СЕРВИСАМИ.....	257

Научное издание

**XXIV Всероссийская студенческая
научно-практическая конференция
Нижевартовского государственного университета**

Часть 3

Информационные технологии

Нижевартовск, 5-6 апреля 2022 г

ISBN 978-5-00047-646-8



Под общей редакцией: *Д.А. Погоньшева*
Редакторы: *Е.С. Овечкина, И.С. Анцева, Е.В. Вилявина*
Технический редактор: *Д.В. Вилявин*
Обложка: *Д.В. Вилявин*

Подписано в печать 15.04.2021
Формат 60×84/8
Гарнитура Times New Roman. Усл. печ. листов 13,45
Электронное издание. Объем 35,7 МБ. Заказ 2241

Издательство НВГУ
628615, Тюменская область, г. Нижневартовск, ул. Маршала Жукова, 4
Тел./факс: (3466) 24-50-51, E-mail: red@nvsu.ru, izdatelstvo@nggu.ru