автономное профессиональное образовательное учреждение

Вологодской области

«Вологодский колледж связи и информационных технологий»

**СОГЛАСОВАНО:**

Заместитель директора по методическому сопровождению и инновационной деятельности

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Потылицына

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ, ОБУЧАЮЩЕГО ПРОГРАММИРОВАНИЮ ДЕТЕЙ ДОШКОЛЬНОГО ВОЗРАСТА**

специальность09.02.07 «Информационные системы и программирование»

группа ИСП-418А

Студент \_\_\_\_\_\_\_\_\_\_\_

*подпись И.О.Фамилия*

\_\_\_\_ \_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись И.О.Фамилия*

\_\_\_\_ \_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_

Консультант по экономической части \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись И.О.Фамилия*

\_\_\_\_ \_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись И.О.Фамилия*

\_\_\_\_ \_\_\_\_\_\_\_\_\_ 2022

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись И.О.Фамилия*

\_\_\_\_ \_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_

Вологда,

2022 г.

**СОДЕРЖАНИЕ**

* ВВЕДЕНИЕ
* РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ [ПРОГРАММНОГО ПРОДУКТА/ ВЕБ-САЙТА/МОБИЛЬНОГО ПРИЛОЖЕНИЯ/ ИГРОВОГО ПРОДУКТА/И ДР.]
* 1.1 Описание предметной области
* 1.2 Моделирование бизнес-процессов
* 1.3 Экономическое обоснование []
* 1.4 Сравнительный анализ систем аналогов
* РАЗДЕЛ 2. РЕАЛИЗАЦИЯ [ПРОГРАММНОГО ПРОДУКТА/ ВЕБ-САЙТА/ МОБИЛЬНОГО ПРИЛОЖЕНИЯ/ИГРОВОГО ПРОДУКТА/И ДР.]
* 2.1 Инструментальные средства разработки
* 2.2 Организация хранения информации
* 2.3 Реализация [программных модулей/игровых процессов]
* 2.4 Разработка [графического пользовательского интерфейса/ локаций и 3D моделей для игры]
* 2.5 [Дополнительный подраздел]
* ЗАКЛЮЧЕНИЕ
* СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ
* ПРИЛОЖЕНИЯ

**ВВЕДЕНИЕ**

На сегодняшний день информационные технологии имеют большую популярность на рынке. Любая компания мечтает об IT профессионале в своей команде. Многие дети еще с малых лет начинают интересоваться программированием и информационными технологиями, чтобы в будущем стать настоящим специалистом в этой сфере. Все современные курсы, а также онлайн курсы для детей, системы дистанционного обучения, которые постоянно все рекламируют, содержат информацию, построенную на принципах одностороннего изложения - выложен видеоматериал занятия или обычный текст. В занятиях очень мало визуализации и интерактивности, поэтому они не привлекают внимание детей. Игровая индустрия постоянно развивается. Существует огромное множество разновидностей жанров видеоигр, поэтому очень много людей в той или иной игре находят что-то интересное для себя. Когда нечем заняться, люди проводят все свое время с гаджетом. Конечно же, особенно сильное влияние мобильные игры и приложения оказывают на молодое поколение, а именно на детей. Поэтому будет отличным решением спроектировать и разработать обучающее мобильное приложение.

Актуальность темыразработки мобильного приложения, обучающего программированию детей дошкольного возраста неоспорима, так как программирование сейчас неотъемлемая часть нашей жизни и является фундаментом информационного будущего.

Предметом выпускной квалификационной работы является приложение, обучающее программированию детей дошкольного возраста.

Объектом исследования выпускной квалификационной работы является мобильное приложение.

Целью выпускной квалификационной работы является разработка мобильного приложения, обучающего программированию детей дошкольного возраста.

Для достижения поставленной цели были определены следующие задачи:

1. Описать технические характеристикипроектирования и разработки мобильного приложения, обучающего программированию детей дошкольного возраста.
2. Исследовать предметную область.
3. Спроектировать инфологическую и ER модель базы данных.
4. Привести базу данных в 3 нормальную форму.
5. Разработать базу данных и разместить ее на СУБД.
6. Разработать мобильное приложение
7. Протестировать разработанное мобильное приложение

Методами исследования данного курсового проекта стали:

….

Выпускная квалификационная работа состоит из введения, двух разделов, заключения и списка использованных источников.

**РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ, ОБУЧАЮЩЕГО ПРОГРАММИРОВАНИЮ ДЕТЕЙ ДОШКОЛЬНОГО ВОЗРАСТА**

## **1.1 Описание предметной области**

Разрабатываемое мобильное приложение служит начальной ступенью погружения детей в мир информационных технологий. Оно позволит определить интерес ребенка к программированию и даст хороший старт в освоении информационных технологий. Задачи в приложении, которые будут стоять перед пользователем, способствуют развитию логического и структурного мышления. Приложение отличается оригинальным дизайном, в корне отличающимся от аналогов. Таким образом, разрабатываемый проект рассчитан на заинтересованность в нем детей. Платформа позволит детям стать активными пользователями технологий, а также получить знания, которые в будущем могут помочь устроиться на хорошую работу в сфере информационных технологий.

Пользователь приложения будет иметь свой собственный аккаунт, при регистрации в приложении необходимо будет ввести ФИО, почту родителя, логин и пароль. После регистрации аккаунта, информация о пользователе записывается в базу данных на сервер. Помимо личных данных на сервер также будет записываться прогресс прохождения курса, чтобы впоследствии можно было вернуться и продолжить урок в описываемой данным точке.

## **1.2 Моделирование бизнес - процессов**

Прежде, чем приступить к разработке приложения, необходимо детально продумать его работу. Чтобы при разработке не упустить важные детали в работе приложения или не забыть о существовании каких-либо функций любой проект проходит этап моделирования.

Моделирование бизнес-процессов является одним из методов улучшения качества и эффективности разрабатываемой программы. Он основан на описании процесса через различные элементы, присущие этому процессу (такие как действия, данные, события и др.). Кроме того, моделирование бизнес-процессов описывает логическую взаимосвязь всех элементов процесса от начала до завершения в рамках разрабатываемой программы.

Для наиболее точного понимания работы системы на этапе моделирования используется несколько методов. Эти методы относятся к разным видам моделирования и позволяют сфокусировать внимание на различных аспектах. Они содержат как графические, так и текстовые инструменты, с помощью которых можно визуализировать основные компоненты процесса и дать точные определения параметров и связей элементов.

UML (Unified Modeling Language) – объектно-ориентированный метод моделирования процессов. Он включает в себя 9 диаграмм, которые позволяют моделировать отдельные статические или динамические аспекты процессов, однако для описания процессов работы разрабатываемого мобильного приложения, обучающего программированию детей дошкольного возраста, будет достаточно 3 UML-диаграммы: диаграмма деятельности, диаграмма классов и диаграмма прецедентов.

Диаграмма деятельности — технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ. На представленной выше диаграмме представлен процесс использования приложения пользователем. Диаграмма имеет три актера: Пользователь, Приложение и Сервер, на дорожках которых изображены операции, выполняемые тем или иным объектом.

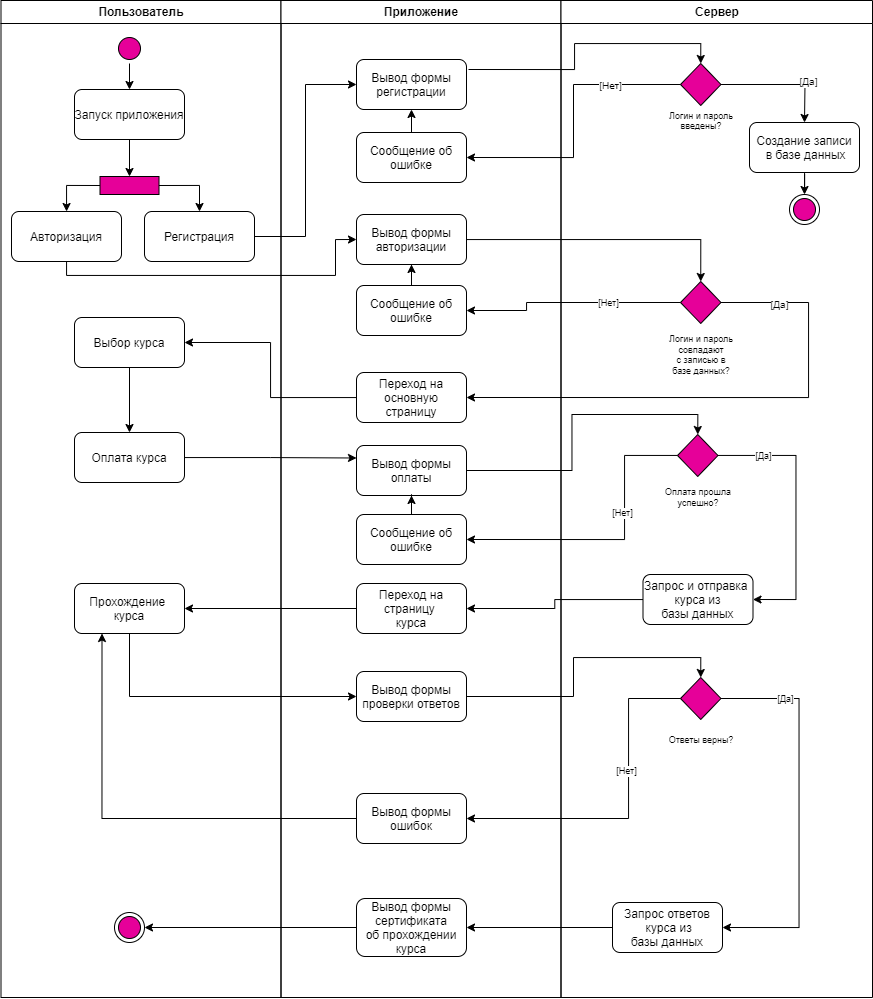


Рисунок 1. UML диаграмма деятельности

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На таких диаграммах отображаются атрибуты и операции классов, а также связи между атрибутами разных классов.

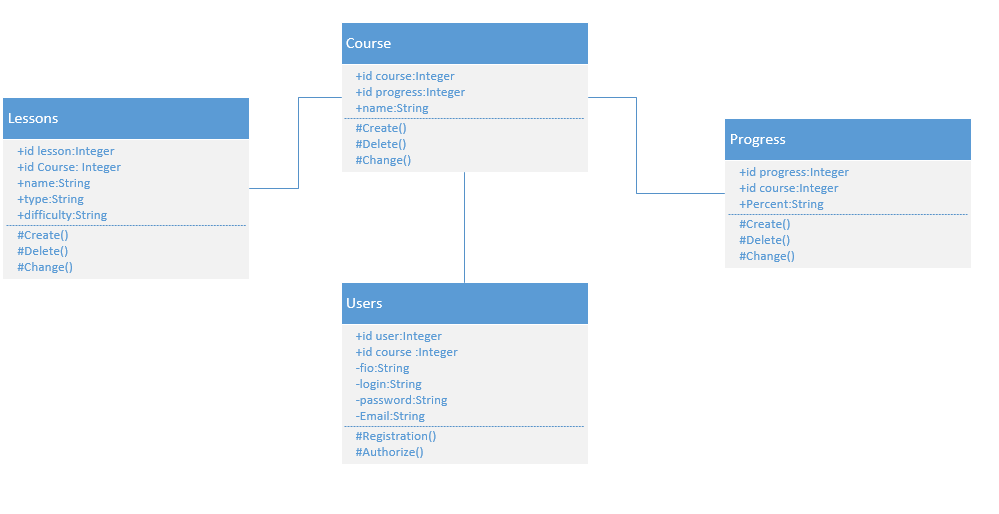


Рисунок 2. UML диаграмма классов

Диаграмма прецедентов описывает основные группы пользователей системы и варианты её использования. Как правило, такая диаграмма состоит из актёров (актёр – это набор ролей пользователя), прецедентов (действий, выполняемых системой) и отношений между прецедентами. На представленной выше диаграмме изображено три актёра, каждый из которых имеет собственные прецеденты.

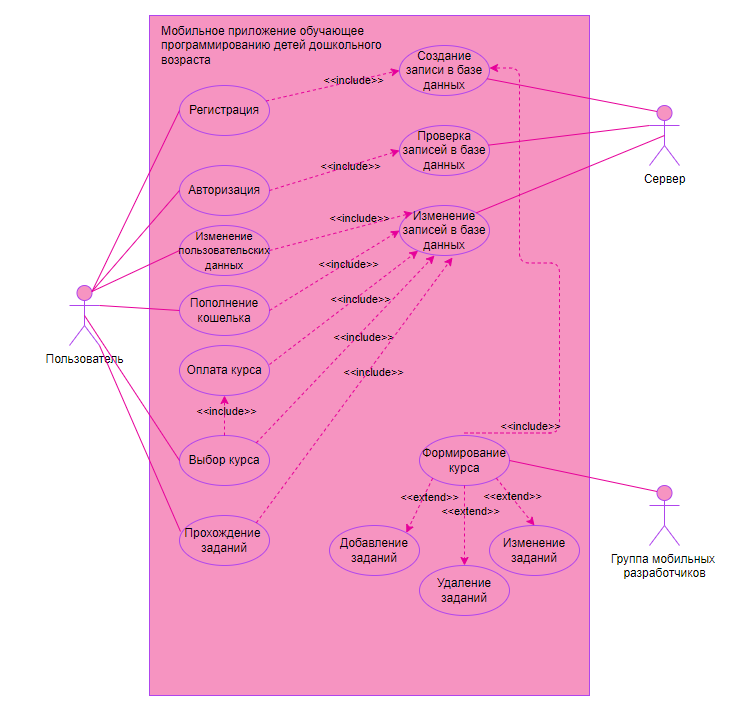


Рисунок 3. UML диаграмма прецедентов

Помимо UML диаграмм, для описания бизнес-процессов системы используется методология IDEF0. Её отличительной особенностью является то, что в IDEF0 рассматриваются логические отношения между процессами, а не их временная последовательность.

Схема обозначений в IDEF0 является достаточно простой: прямоугольник(блок) – это бизнес-процесс, а стрелки – это его связи с другими процессами или элементами внешней среды. Стрелки бывают входящие (вводные, они ставят определённую задачу), исходящие (результат деятельности процесса), управляющие (сверху вниз, являются механизмами управления процессом, например, инструкции, законы и пр.) и механизмы (снизу вверх, показывают, что используется для того, чтобы процесс выполнялся, или кто руководит выполнением данного процесса).

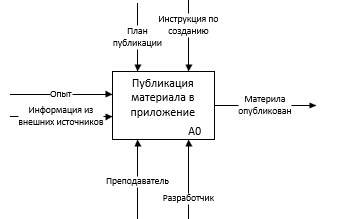


Рисунок 4. Диаграмма описания бизнес-процессов

На представленной выше диаграмме описан процесс публикации материала в разрабатываемое приложение. В выполнении этого процесса задействованы несколько лиц: Преподаватель и Разработчик.

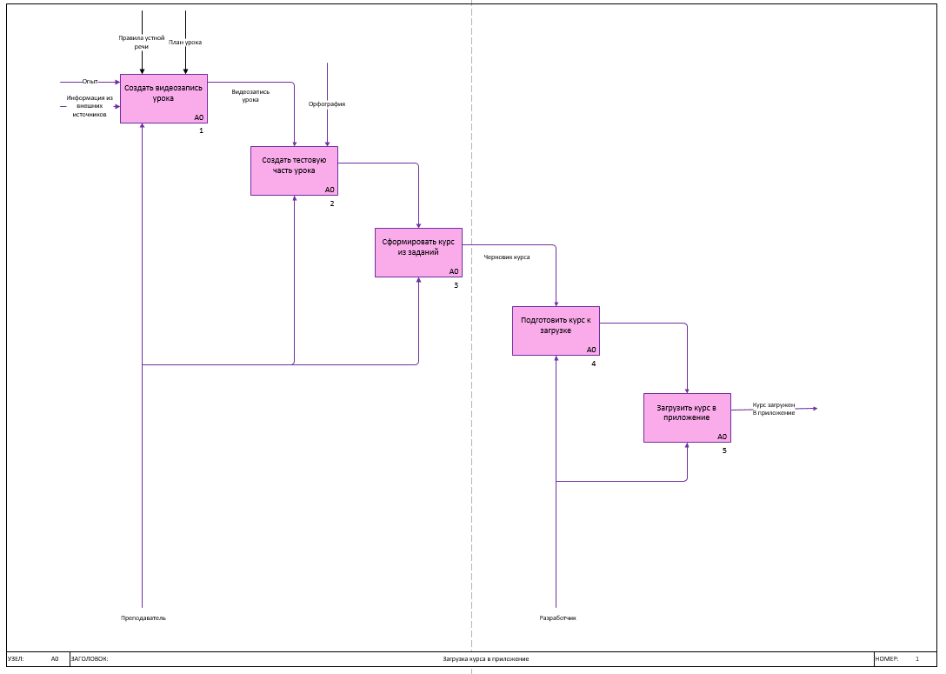


Рисунок 5. Диаграмма описания бизнес-процессов (декомпозиция)

Поскольку публикация материала – сложный процесс, диаграмма IDEF0 переходит на уровень ниже, где данный процесс описывается подробнее. Этот процесс называется декомпозицией. На декомпозиции представлена последовательность действий, необходимых для успешного завершения основного процесса – публикации материала.

**1.3 Экономическое обоснование**

Для экономического обоснования проекта необходимо произвести стоимостную оценку затрат на разработку программного продукта. Для этих целей необходимо составить смету затрат, которая может включать следующие статьи:

1. Расходы, связанные с оплатой труда (основная заработная плата; дополнительная заработная плата; отчисления на социальные нужды);
2. Расходы, связанные с эксплуатацией и содержанием оборудования (амортизация; электроэнергия);
3. Расходы на материалы;
4. Накладные расходы.

Для составления сметы затрат необходимо составить производственный план, в котором должны быть описаны все процессы, связанные с подготовкой и реализацией программного продукта. Необходимо составить таблицу, в которой будет отображены виды работ, выполняемых программистом, и количество используемого для этого времени. Расчет времени работы программиста на ПК производится отдельно.

Таблица 1

Затраты рабочего времени на разработку программного продукта

|  |  |  |
| --- | --- | --- |
| № | Наименование работ | Длительность в человеко-часах |
| 1 | Разработка технического задания | 6 |
| 2 | Анализ технического задания и сбор данных | 20 |
| 3 | Составление алгоритма | 20 |
| 4 | Переложение алгоритма на язык программирования | 10 |
| 5 | Реализация программы на ПК | 11 |
| 6 | Отладка программы на ПК | 10 |
| 7 | Проведение экспериментов на ПК | 13 |
| 8 | Оформление пояснительной записки на ПК | 12 |
| 9 | ВСЕГО | 102 |
| 10 | в том числе с использованием ПК | 102 |

Следующим этапом является определение величины расходов на оплату труда. Величина заработной платы программистов определяется исходя из часовых ставок, действующих на конкретном предприятии (для предприятий, находящихся на бюджетном финансировании, - на основании тарифных ставок), либо исходя из средней заработной платы программистов. В городе Вологда по состоянию на апрель 2021 года средняя заработная плата программиста без опыта работы составляет 35 тыс. рублей. Таким образом, можно рассчитать среднюю часовую тарифную ставку по формуле:

*ЧТС = Средняя заработная плата / (Количество рабочих дней в месяце х 8 часовой рабочий день)*

(1)

ЧТС = 35000 / (22 ∙ 8) = 199 руб.

Таким образом, ЧТС составила 199 рублей.

Далее формируется фонд оплаты труда.

Фонд оплаты труда (ФОТ) – это общая сумма затрат на оплату труда и выплат социального характера.

Фонд оплаты состоит из основной заработной платы программиста, дополнительной заработной платы и обязательные отчисления в социальные фонды.

Основная заработная плата (ОЗП) - включает в себя заработную плату программиста, непосредственно осуществляющего реализацию программного продукта. Она определяется исходя из часовой тарифной ставки и расчетных норм времени, затраченных на реализацию проекта. Рассчитывается по формуле:

*ОЗП = ЧТС х чел/час*

(2)

ОЗП = 199 ∙ 102 = 20298 руб.

Таким образом, ОЗП составила 20298 рублей.

Дополнительная заработная плата (ДЗП) - относят премии, надбавки, расходы, связанные с оплатой отпусков, районные коэффициенты и прочие выплаты. Если величина дополнительных выплат точно неизвестна, то, как правило, при расчете фонда оплаты труда, сумму дополнительной заработной платы принимают равной 20% от величины основной заработной платы. Дополнительную заработную плату можно найти, рассчитав формулу:

*ДЗП = ОЗП х 20%*

(3)

ДЗП = 20298 ∙ 20%= 4059,6 руб.

Таким образом, ДПЗ составила 4059,6 рублей.

Обязательные отчисления в социальные фонды (ООСФ) – это отчисления во внебюджетные социальные фонды, с целью обеспечения работающим социальных гарантий, связанный с пенсионным обеспечением, медицинским обслуживанием, пособий по безработице и т.д.

В 2021 году отчисления составляют:

22% – на пенсионные отчисления;

5,1% – на обязательное медицинское страхование;

2,9% – на социальное страхование.

Таким образом, общая сумма отчислений составляет 30%

*ООСФ = (ДЗП +ОЗП) х 30%*

(4)

ООСФ = (20298 руб.+ 4059,6 руб.) ∙ 30% = 7 307,28 руб.

Следовательно, ООСФ равны 7307,28 рублей.

Полученные данные о составе фонда оплаты труда представлены в таблице:

Таблица 2

Фонд оплаты труда

|  |  |  |
| --- | --- | --- |
| № | Статья | Показатель (руб) |
| 1 | Основная заработная плата (ОЗП) | 20298 |
| 2 | Дополнительная заработная плата (ДЗП) | 4059,6 |
| 3 | Обязательные отчисления в социальные фонды | 7 307,28 |
| 4 | ИТОГО | 31 664,88 |

Следующим этапом производим расчет затрат, связанных с работой оборудования. Основные фонды (внеоборотные фонды, основной капитал) - часть имущества предприятия, используемая в качестве средств труда, при производстве продукции, которые многократно участвуют в производственном процессе, сохраняя при этом свою натуральную форму, постепенно изнашиваясь, переносят свою стоимость по частям на вновь создаваемую продукцию.

Статья «Расходы на содержание и эксплуатацию оборудования» - к этой статье относятся затраты на содержание, амортизацию и текущий ремонт оборудования, транспорта, инструментов и приспособлений. которых погашается путем начисления амортизации.

С 01.01.2017, амортизируемым имуществом считаются объекты, чья стоимость превосходит сто тысяч рублей, а срок службы составляет более одного года. (предусмотрено постановлением Правительства РФ от 07.07.2016 № 640.)

**ПБУ 6 «Основные средства»** устанавливает четыре способа начисления амортизации, один из них – линейный, который широким распространением из-за своей простоты и удобства.

Расчет амортизации за месяц линейным способом:

*Норма А (в%) = 100 : Срок полезного использования в месяцах*

*А= О перв ∙ Норма А (в %),*

(4)

где Норма А – норма амортизационных отчислений за месяц в %;

О перв – первоначальная стоимость объекта основных средств.

Для расчета расходов, связанных с эксплуатацией и содержанием оборудования, необходимо составить перечень основных фондов, занятых на реализации проекта программного продукта. Данные приведены в таблице:

Таблица 3

Перечень основных фондов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование оборудования | Количество  (шт) | Потребляемая мощность  (Ватт) | Цена  (руб) |
| Ноутбук Apple MacBook Air 13 | 1 | 30 Вт | 99990 |
| ИТОГО |  |  | 99990 |

Так как, объекта основных фондов составляет менее ста тысяч рублей, амортизация по нему не начисляется.

Расходы на электроэнергию определяются исходя из действующих в регионе тарифов и потребляемой мощности оборудования.

Предположим расход электроэнергии, потребленной ПК, занятым на реализации проекта программного продукта составляет 0,03 кВт, время эксплуатации составляет 102 часа, тариф для городского населения 4,87 – руб./кВт. час. Тогда общая сумма расходов на электроэнергию за год составит: 102 часов ∙ 0.03 ∙ 4,87= 14,9 руб.

При работе на персональном компьютере в сети Интернет, необходимо учитывать расходы, связанные с оплатой трафика. Данные расходы определяются исходя из действующих тарифов на интернет-трафик, а также времени работы ПК в сети.

Абонентская плата составляет 600 руб. в месяц, при 8-часовом рабочем дне и пятидневной рабочей неделе. Время работы ПК в сети по реализации проекта составило 102 часа. Тогда сумма расходов на интернет – трафик составит: 600 : (8 ∙ 22) ∙ 102 = 347,7 руб.

Таким образом, расходы на содержание и эксплуатацию оборудования можно представить в таблице:

Таблица 4

Расходы, связанные с эксплуатацией оборудования

|  |  |  |
| --- | --- | --- |
| № | Наименование статьи | Показатель (руб.) |
| 1 | Амортизация | 0 |
| 2 | Расходы на электроэнергию | 14,9 |
| 3 | Расходы на Интернет | 347,7 |
| 4 | ИТОГО | 362,6 |

При создании и внедрении программного продукта или информационной технологии, как правило, используют такие материальные носители информации как: RW CD, бумага, канцелярские принадлежности, справочная литература и т.п. Все затраты на материалы сводятся в таблицу:

Таблица 5

Расходы на материалы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование | Единица измерения | Количество | Цена за единицу, руб. | Сумма руб. |
| 1 | Бумага | упаковка | 1 | 240 | 240 |
| 2 | Ручка гелевая | шт. | 2 | 30 | 60 |
| 3 | ИТОГО |  |  |  | 300 |

Далее необходимо произвести расчет накладных расходов. Накладные расходы включают в себя затраты, связанные с общехозяйственными нуждами, которые возникают в ходе реализации проекта. К ним можно отнести расходы на аренду, на общекоммунальные платежи, на содержание аппарата управления и т.д. Величина накладных расходов рассчитывается в процентном отношении от заработной платы и устанавливается в целом по предприятию. При разработке программных продуктов эта величина может составлять 20% от фонда оплаты труда.

Расчет накладных расходов при создании программного продукта 20% от ФОТ

*Накладные расходы = ФОТ х 20%*

(5)

Накладные расходы = 31664,88 ∙ 20% = 6333 руб.

Таким образом, накладные расходы составили 6333 рублей.

Определив суммы расходов по статьям затрат, необходимо рассчитать себестоимость программного продукта.

Себестоимость – это совокупность всех затрат, связанных с производством программного продукта.

Провести калькуляцию себестоимости можно в виде таблицы:

Таблица 6

Калькуляция себестоимости программного продукта

|  |  |  |
| --- | --- | --- |
| № | Наименование статьи затрат | Показатель (руб.) |
| 1 | Расходы на оплату труда (ФОТ) | 31664,88 |
| 2 | Расходы, связанные с эксплуатацией и содержанием оборудования | 362,6 |
| 3 | Расходы на материалы | 300 |
| 4 | Накладные расходы | 6 333 |
| 5 | ИТОГО | 38 660,48 |

Таким образом, себестоимость программного продукта составила 38 660,48 рублей.

Цена **-** это важнейшая экономическая категория, отражающая в денежном выражении ценность товара или предоставляемой услуги.

Важным фактором, влияющим на процесс формирования цены, является конкуренция на рынке, необходимость учета которой совершенно очевидна. В целях повышения конкурентоспособности продукта может возникнуть необходимость снижения его цены на рынке. Наиболее важным моментом для разработчика, с экономической точки зрения, является процесс формирования цены. Очевидно, что программные продукты представляют собой весьма специфический товар со множеством присущих им особенностей. Многие их особенности проявляются и в методах расчетов цены на них. На разработку программного продукта средней сложности обычно требуются весьма незначительные средства. Однако, при этом он может дать экономический эффект, значительно превышающий эффект от использования достаточно дорогостоящих систем.

Цена программного продукта включает:

1. себестоимость программного продукта;
2. предлагаемую прибыль;
3. НДС

Налог на добавленную стоимость (НДС) – это налог, который возникает при продаже предприятием продукции и предоставлении услуг с добавленной стоимостью. В соответствии с налоговым законодательством РФ, НДС уплачивают предприятия, находящиеся на обычной системе налогообложения. НДС не платится при операциях, которые осуществляются в рамках деятельности, переведенной на ЕНВД и патентную систему. НДС также не начисляют налогоплательщики, перешедшие на упрощенную систему (УСН) или на уплату единого сельскохозяйственного налога. Так как, большинство предприятий, которые создают и реализуют программные продукты, относятся к сфере малого бизнеса, чаще всего выбирают специальные налоговые режимы, не предусматривающие уплату НДС.

Цена программного продукта или информационной технологии определяется по формуле:

*Ц = Сз ∙ (1 + N),*

(6)

где Сз– себестоимость программного продукта или информационной технологии, руб.;

N – норма прибыли, %.

Норма прибыли (N) **-** это показатель доходности проекта создания программного продукта или прирост денежных средств, которые были затрачены на его создание. Для обеспечения доходности норму прибыли для программного продукта можно принять за 0,2

Ц = 38 660,48 ∙ (1 + 0,2) = 46 392,57 руб.

Таким образом, цена программного продукта составит 46 392,57 руб.

**1.4 Сравнительный анализ систем аналогов**

Следующим этапом проектирования необходимо провести сравнительный анализ систем аналогов. Изучая рынок мобильных приложений можно отметить, что конкурентов не так много, а с такой же концепцией и вовсе нет. Рассмотрим аналоговые приложения на основе некоторых критериев:

* Наличие обучающего материала.
* Наличие видео лекций.
* Возможность прохождения уровней заново.
* Возможность связи с преподавателем.

Таблица 7

Сравнение мобильных приложений, обучающих программированию детей

|  |  |  |  |
| --- | --- | --- | --- |
| Сравниваемые возможности | Разрабатываемое приложение | LightBot | Code the Robot |
| Наличие обучающего материала | + | - | - |
| Наличие видео лекций | + | - | - |
| Возможность прохождения уровней заново | + | + | + |
| Возможность связи с преподавателем | + | - | - |

**РАЗДЕЛ 2. РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ, ОБУЧАЮЩЕГО ПРОГРАММИРОВАНИЮ ДЕТЕЙ ДОШКОЛЬНОГО ВОЗРАСТА.**

**2.1 Инструментальные средства разработки**

Для того, чтобы спроектировать и разработать базу данных, требуется специальное программное обеспечение, которое позволяет создавать и редактировать БД.

В качестве СУБД для разработки мобильного приложения обучающего программированию детей дошкольного возраста был выбран PostgreSQL. Для упрощения администрирования на сервере postgresql в базовый комплект установки входит такой инструмент как pgAdmin. Он представляет графический клиент для работы с сервером, через который мы в удобном виде можем создавать, удалять, изменять базы данных и управлять ими.

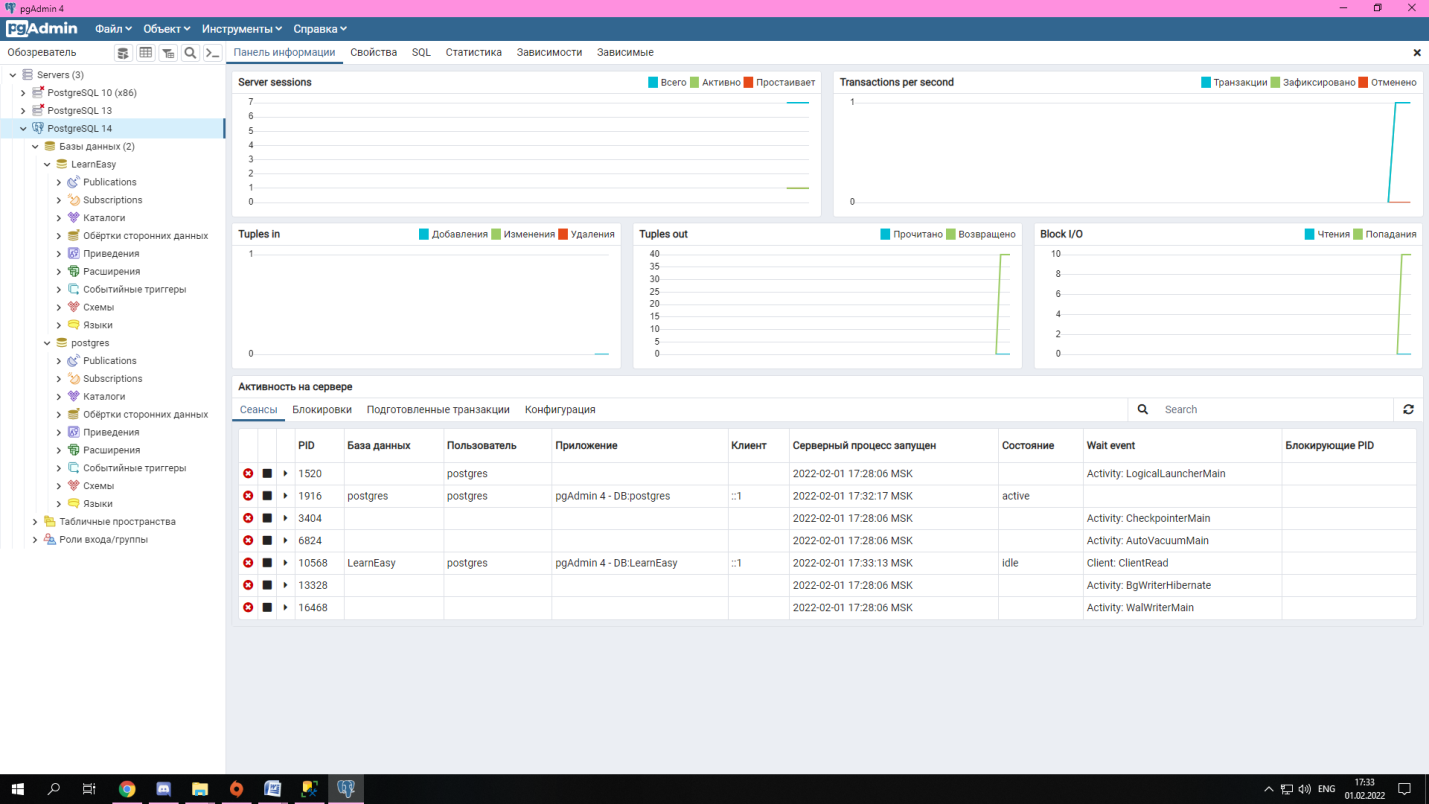


Рисунок 6. Интерфейс pgAdmin

При моделировании бизнес-процессов в качестве программного обеспечения был выбран Microsoft Visio и Draw.io. Visio - это инновационное решение, которое помогает визуализировать бизнес-процессы, которые связаны с определенными данными. Draw.io - это онлайн сервис, предназначенный для формирования диаграмм и схем.

С помощью Microsoft Visio были созданы: Er-модель базы данных, UML диаграмма классов, диаграмма описания бизнес-процессов. С помощью онлайн сервиса Draw.io были созданы: UML диаграмма деятельности, UML диаграмма прецедентов.

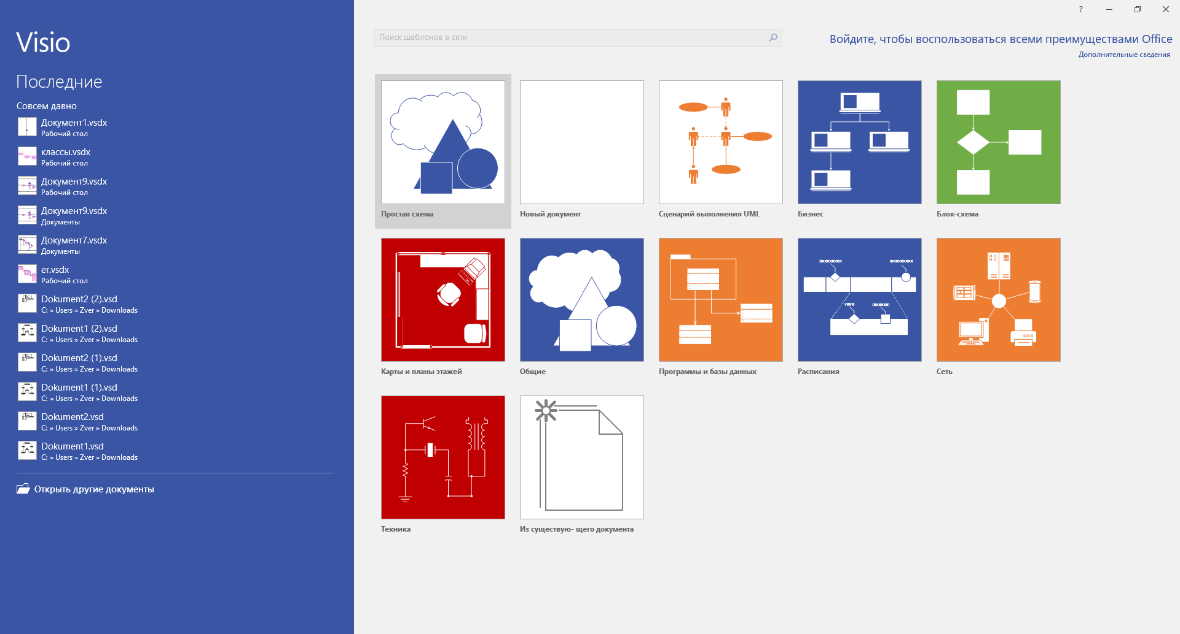


Рисунок 7. Выбор шаблона для создания диаграммы в Visio

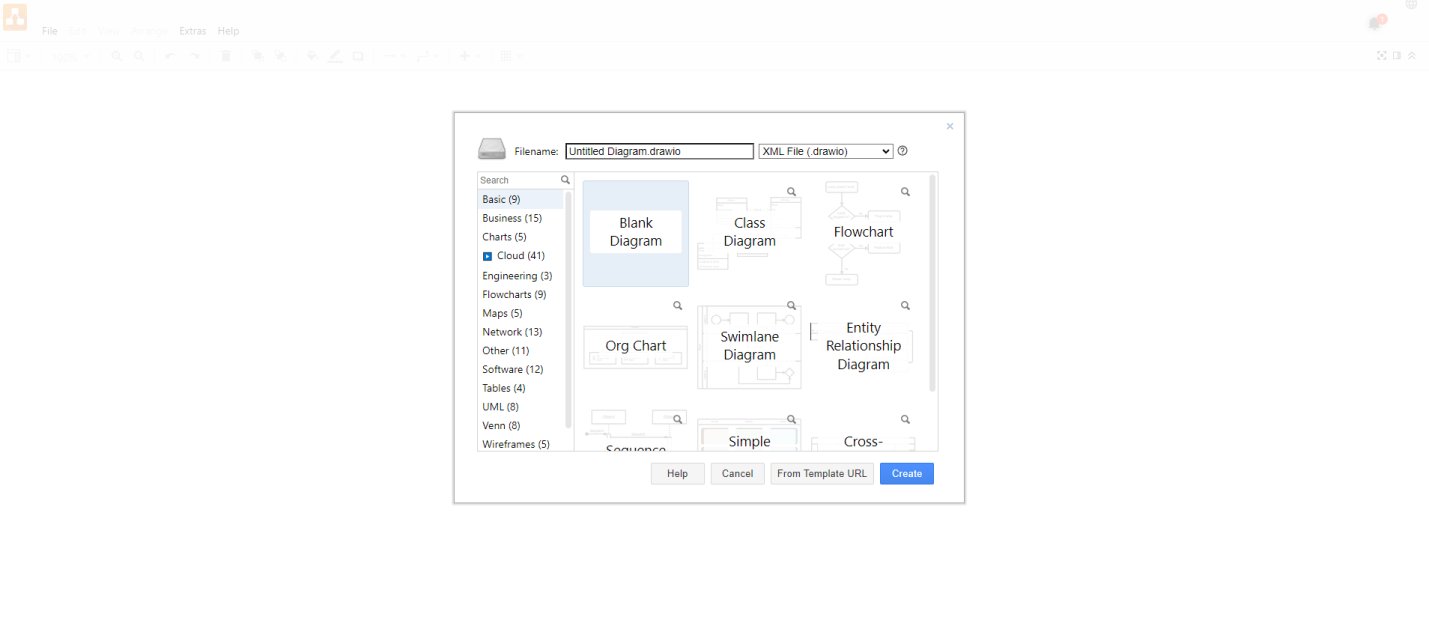


Рисунок 8. Выбор шаблона для создания диаграммы в Draw.io

Для разработки приложения была выбрана операционная система Android 8.1 Oreo. ОC является инновационной и с каждым годом набирает все большую и большую популярность.

В качестве языка программирования выбран Java. Java является объектно-ориентированным языком программирования. Приложения Java могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры.

Язык Java активно используется для создания мобильных приложений под операционную систему Android.

В качестве среды разработки Android приложения был выбран программный продукт Android Studio используя при этом плагины Android Development Tools, Android Drawable Importer, Name That Color. Android Studio является интегрированной средой разработки (IDE) для работы с платформой Android.

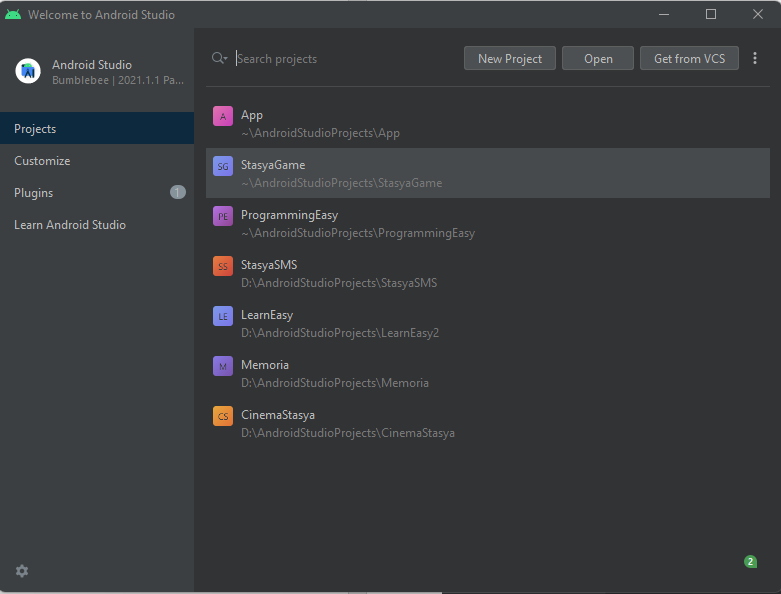
****

Рисунок 9. IDE Android Studio

**2.2 Организация хранения информации**

После установки необходимого программного обеспечения для создания баз данных, подготовки моделей бизнес-процессов и понимания системы работы разрабатываемого приложения, необходимо перейти к этапу, на котором эта база данных будет создаваться.

Перед созданием таблиц в базе данных нужно определить сущности, их атрибуты и все возможные связи между ними. Сущность – это любой объект, который можно выделить при анализе предметной области, для которой разрабатывается база данных. Каждая сущность включает в себя атрибуты, которые служат для подробного описания этой сущности. Между двумя сущностями могут быть установлены связи – при их помощи возникает понимание отношений этих сущностей в разрабатываемой базе данных.Связи создаются с помощью внешних ключей. Внешний ключ устанавливается для столбцов из зависимой, подчиненной таблицы, и указывает на один из столбцов из главной таблицы.

Инфологическая модель служит для представления детальной информации о какой-либо сущности. Модель представляет собой таблицу, состоящую из атрибутов сущности, их параметров (названия, типа, размера и пр.) и связей между атрибутами сущностей.

После исследования предметной области можно выделить как минимум 14 сущностей (будущие таблицы базы данных): Users, Course, Lessons и т.д. Сущность

Users, хранит в себе данные о пользователях, имеет 6 атрибутов, первичный ключ и 1 внешний ключ, подробнее см. в таблице 1.

Таблица 8

Инфологическая модель сущности Users

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Users | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdUser | + | + | integer |  | Primary |  |
| FirstName | + |  | character varying | 30 |  |  |
| LastName | + |  | character varying | 30 |  |  |
| Birthday | + |  | date |  |  |  |
| Login | + |  | character varying | 30 |  |  |
| Password | + |  | character varying | 30 |  |  |
| Email | + |  | character varying | 30 |  |  |
| AccTypeId | + | + | integer |  | Foreign | AccType.Id |

Сущность Course, хранит в себе данные о курсе, имеет 1 атрибут, первичный ключ и 1 внешний ключ, подробнее см. в таблице 2.

Таблица 9

Инфологическая модель сущности Course

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdCourse | + | + | integer |  | Primary |  |
| Name | + |  | character varying | 70 |  |  |
| ResoultsId | + | + | integer |  | Foreign | Resoults.Id |

Сущность Lessons, хранит в себе данные об уроках, которые проходит пользователь, а также о том какие уроки присутствуют в курсе пользователя. Сущность имеет 1 атрибут, первичный ключ и 3 внешних ключа, подробнее см. в таблице 3.

Таблица 10

Инфологическая модель сущности Lessons

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lessons | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdLesson | + | + | integer |  | Primary |  |
| Name | + |  | character varying | 30 |  |  |
| CourseId | + | + | integer |  | Foreign | Course.Id |
| ExerciseId |  | + | integer |  | Foreign | Exercise.Id |
| LessonTypeId | + | + | integer |  | Foreign | LessonType.Id |

Сущность Progress, хранит в себе данные о прогрессе курса, имеет 1 атрибут и 2 внешних ключа, подробнее см. в таблице 4.

Таблица 11

Инфологическая модель сущности Progress

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Progress | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| Percent | + |  | character varying | 30 |  |  |
| CourseId | + | + | integer |  | Foreign | Course.Id |
| UserId | + | + | integer |  | Foreign | User.Id |

Сущность LessonType, хранит в себе данные о типе урока (видеоурок, практическое занятие, итоговые и промежуточные тесты) имеет 2 атрибутa, первичный ключ, подробнее см. в таблице 5.

Таблица 12

Инфологическая модель сущности LessonType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LessonType | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdLessonType | + | + | integer |  | Primary |  |
| Type | + |  | character varying | 30 |  |  |
| Difficulty | + |  | character varying | 30 |  |  |

Сущность Exercise, хранит в себе данные о курсе, имеет 2 атрибута и первичный ключ, подробнее см. в таблице 6.

Таблица 13

Инфологическая модель сущности Exercise

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Exercise | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdExercise | + | + | integer |  | Primary |  |
| Description | + |  | character varying | 150 |  |  |
| Name | + |  | character varying | 30 |  |  |

Сущность ExercisesLesson, имеет 2 внешних ключа, подробнее см. в таблице 7.

Таблица 14

Инфологическая модель сущности ExercisesLesson

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ExercisesLesson | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| ExerciseId | + | + | integer |  | Foreign | Exercise.Id |
| LessonId | + | + | integer |  | Foreign | Lesson.Id |

Сущность Test, хранит в себе данные о тестах, имеет 1 атрибут, первичный ключ и 1 внешний ключ, подробнее см. в таблице 8.

Таблица 15

Инфологическая модель сущности Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdTest | + | + | integer |  | Primary |  |
| Name | + |  | character varying | 30 |  |  |
| ExerciseId | + | + | integer |  | Foreign | Exercise.Id |

Сущность Resoults, хранит в себе данные о результатах пользователя. Сущность имеет 1 атрибут, первичный ключ и 2 внешних ключа, подробнее см. в таблице 9.

Таблица 16

Инфологическая модель сущности Resoults

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Resoults | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdResoults | + | + | integer |  | Primary |  |
| Mark | + |  | character varying | 20 |  |  |
| UserId | + | + | integer |  | Foreign | User.Id |
| CourseId | + | + | integer |  | Foreign | Course.Id |

Сущность AccType, хранит в себе данные о типе аккаунта, имеет 2 атрибута и первичный ключ, подробнее см. в таблице 10.

Таблица 17

Инфологическая модель сущности AccType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AccType | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdAccType | + | + | integer |  | Primary |  |
| Name | + |  | character varying | 30 |  |  |
| Opportunity | + |  | character varying | 50 |  |  |

Сущность Enrollment хранит в себе информацию о записи пользователя на курс, дату записи и дату окончания. Сущность имеет 2 атрибута и 2 внешних ключа, подробнее см. в таблице 11.

Таблица 18

Инфологическая модель сущности Enrollment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enrollment | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| DateStart | + |  | date |  |  |  |
| DateEnd | + |  | date |  |  |  |
| UserId | + | + | integer |  | Foreign | User.Id |
| CourseId | + | + | integer |  | Foreign | Course.Id |

Сущность Subscription, хранит в себе данные о подписке пользователя на курс, дату ее начала и дату окончания. Сущность имеет 2 атрибута, первичный ключ и 3 внешних ключа, подробнее см. в таблице 12.

Таблица 19

Инфологическая модель сущности Subscription

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subscription | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdSubscription | + | + | integer |  | Primary |  |
| DateStart | + |  | date |  |  |  |
| DateEnd | + |  | date |  |  |  |
| SubTypeId | + | + | integer |  | Foreign | SubType.Id |
| PaymentId | + | + | integer |  | Foreign | Payment.Id |
| UserId | + | + | integer |  | Foreign | User.Id |

Сущность Payment, хранит в себе данные об оплате курса, имеет 2 атрибута, первичный ключ и 2 внешних ключа, подробнее см. в таблице 13.

Таблица 20

Инфологическая модель сущности Payment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Payment | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdPayment | + | + | integer |  | Primary |  |
| Price | + |  | money |  |  |  |
| Discount | + |  | integer |  |  |  |
| SubscriptionId | + | + | integer |  | Foreign | Subscription.Id |
| UserId | + | + | integer |  | Foreign | User.Id |

Сущность SubType, хранит в себе данные о типе подписки, имеет 2 атрибута и первичный ключ, подробнее см. в таблице 14.

Таблица 21

Инфологическая модель сущности SubType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SubType | | | | | | |
| Attribute Name | NOT NULL | UNIQUE | TYPE | SIZE | KEY | RELATION |
| IdSubType | + | + | integer |  | Primary |  |
| Name | + |  | character varying | 30 |  |  |
| Duration | + |  | character varying | 30 |  |  |

Инфологические модели подробно представили каждую сущность, однако для лучшего понимания связей между всеми сущностями существует ER-модель. Она содержит в себе сущности, их атрибуты и связи между ними, представленные в виде схемы.

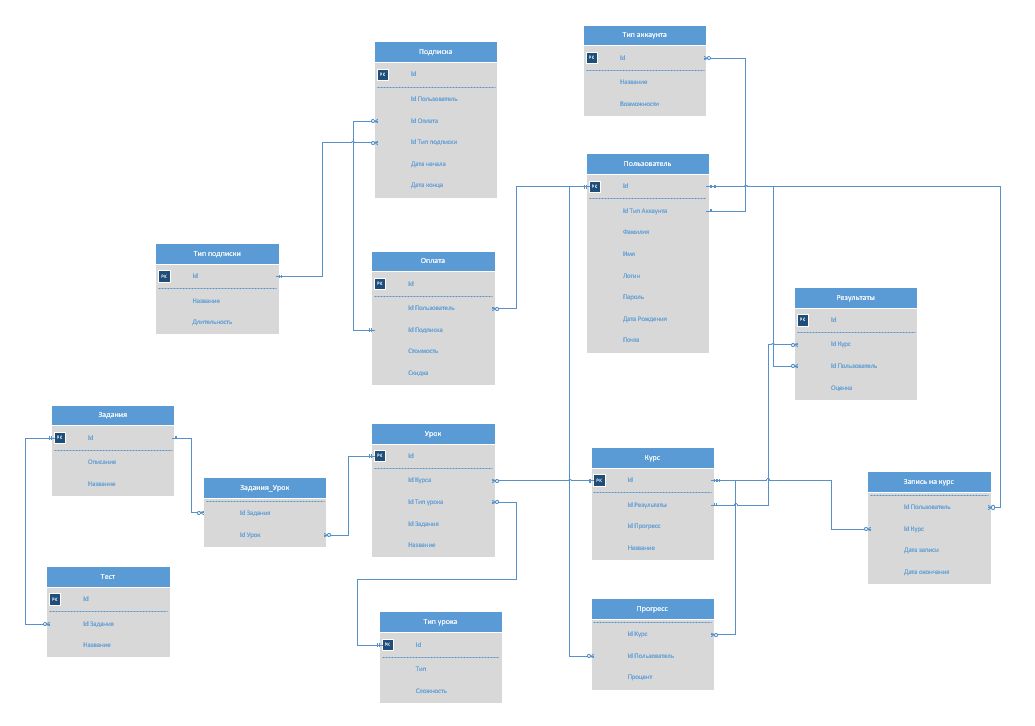


Рисунок 10. ER-модель базы данных мобильного приложения

Основываясь на представленных выше инфологических и ER-модели, создаем таблицы в базе данных при помощи SQL-запроса. Данные запросы написаны на языке SQL.

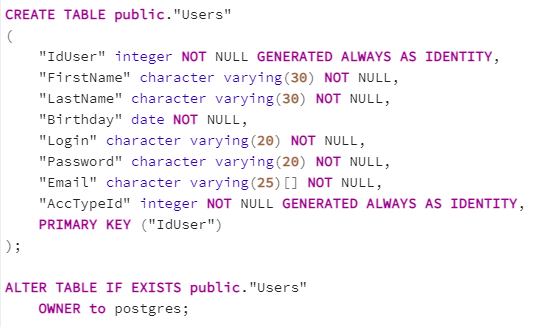


Рисунок 11. Скрипт создания таблицы Users

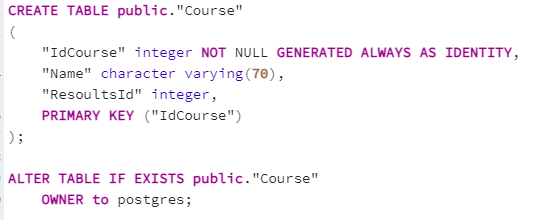


Рисунок 12. Скрипт создания таблицы Course

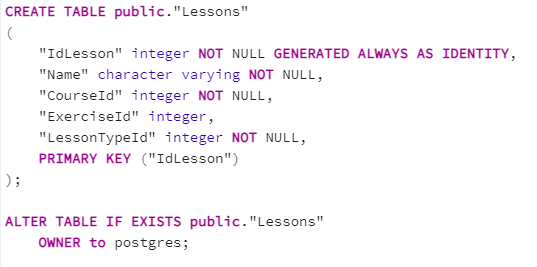


Рисунок 13. Скрипт создания таблицы Lessons

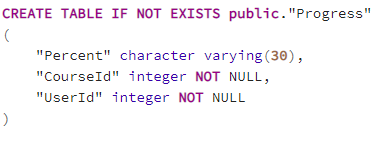


Рисунок 14. Скрипт создания таблицы Progress

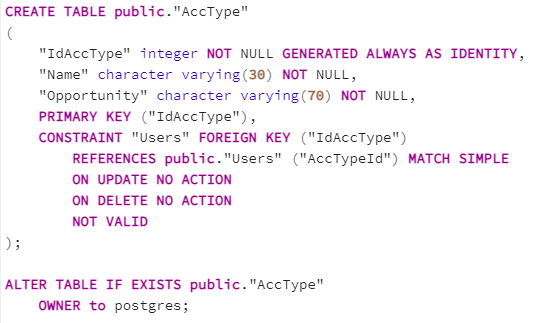


Рисунок 15. Скрипт создания таблицы AccType

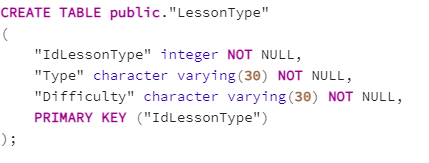


Рисунок 16. Скрипт создания таблицы LessonType

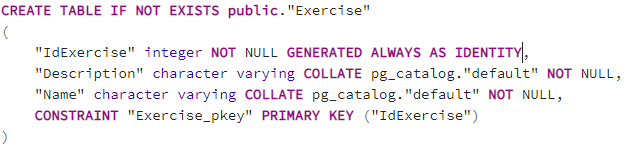


Рисунок 17. Скрипт создания таблицы Exercise

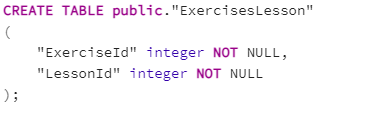


Рисунок 18. Скрипт создания таблицы ExercisesLesson

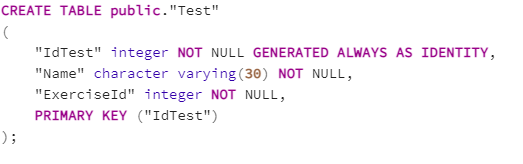


Рисунок 19. Скрипт создания таблицы Test

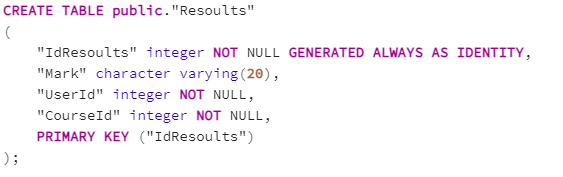


Рисунок 20. Скрипт создания таблицы Results

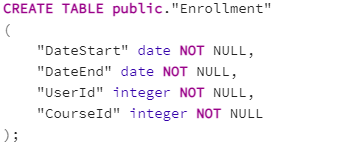


Рисунок 21. Скрипт создания таблицы Enrollment

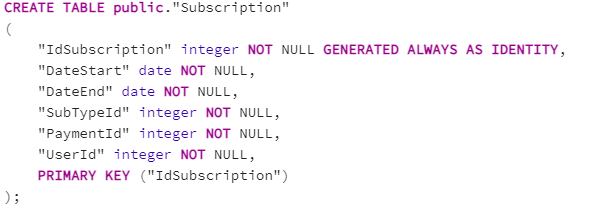


Рисунок 22. Скрипт создания таблицы Subscription

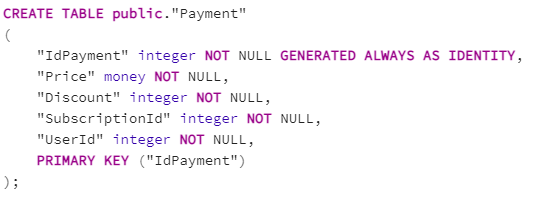


Рисунок 23. Скрипт создания таблицы Payment

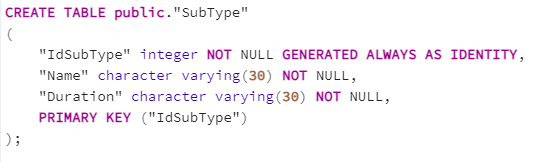


Рисунок 24. Скрипт создания таблицы SubType

После выполнения SQL-запросов в базе данных были созданы таблицы, которые готовы к работе. Для проверки работоспособности базы данных были введенны данные в таблицы и созданы несколько вопросов, выполнение которых прошло успешно.

**2.3 Реализация игровых процессов**

В данном разделе приводится описание программных модулей проекта. Проект состоит из 10 классов. Класс для входа, регистрации, главное меню, выбор уровня(урока), уровней(уроков). Два массива для отображения картинок на уровне(уроке). Для понимания сложные отрезки кода прокомментированы.

Код из класса Entry.Java:

import ndroid.appcompat.app.AppCompatActivity;  
import ndroid.appcompat.widget.AppCompatButton;  
  
import android.content.Intent;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.mysql.jdbc.PreparedStatement;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.ResultSetMetaData;  
import java.sql.Statement;  
  
public class Entry extends AppCompatActivity {  
  
 EditText login;  
 EditText password;  
 String loginUser=””;  
 String passwordUser=””;  
 ndroid state=false;  
 private static final String *url* = “jdbc:mysql://46.229.213.172:3306/evator”;  
 private static final String *user* = “evator”;  
 private static final String *pass* = “rYlSwr2ip1w4”;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 AppCompatButton btn = findViewById(R.id.*button*);  
 login= findViewById(R.id.*editTextTextPersonName*);  
 password = findViewById(R.id.*editTextTextPassword*);  
 btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 loginUser=login.getText().toString();  
 passwordUser = password.getText().toString();  
 ConnectMySql connectMySql = new ConnectMySql();  
 connectMySql.execute(“”);  
 }  
 });  
 }  
  
 public void login(View v){  
  
 }  
  
 public void registration(View v){  
 Intent intent = new Intent(Entry.this, RegisterActivity.class);  
 startActivity(intent);  
 }  
  
 private class ConnectMySql extends AsyncTask<String, Void, String> {  
 String res = “”;  
  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 Toast.*makeText*(MainActivity.this, “Please wait…”, Toast.*LENGTH\_SHORT*)  
 .show();  
  
 }  
  
 @Override  
 protected String doInBackground(String… params) {  
 try {  
 Class.*forName*(“com.mysql.jdbc.Driver”);  
 Connection con = DriverManager.*getConnection*(*url*, *user*, *pass*);  
 System.*out*.println(“Databaseection success”);  
  
 String result = “”;  
 String query =”select \* from stasy”;  
 Statement statement = con.createStatement();  
 ResultSet resultSet = statement.executeQuery(query);  
 int columns = resultSet.getMetaData().getColumnCount();  
 state=false;  
 while (resultSet.next()) {  
 for (int i = 1; i < columns; i++) {  
 if (resultSet.getString(“Login”).equals(loginUser) && resultSet.getString(“Password”).equals(passwordUser)) {  
 System.*out*.println(“Ok”);  
 result = “Вход успешен”;  
 Intent intent = new Intent(MainActivity.this, ScreenActivity.class);  
 startActivity(intent);  
 state=true;  
 break;  
 }  
 }  
 }  
 if(!state) {  
 System.*out*.println(“neok”);  
 result = “Неверный логин или пароль”;  
 }  
  
// while (rs.next()) {  
// result += rs.getString(1).toString() + “\n”;  
// }  
 res = result;  
 } catch (Exception e) {  
 e.printStackTrace();  
 res = e.toString();  
 }  
 return res;  
 }  
  
 @Override  
 protected void onPostExecute(String result) {  
 Toast.*makeText*(Entry.this, result, Toast.*LENGTH\_SHORT*)  
 .show();  
 }  
 }  
}

В данном классе будет осуществляться вход в приложение с помощью введения данных в строчки логина и пароля. После входа пользователь переходит в главное меню приложения. Для незарегистрированных пользователей необходимо нажать на кнопку зарегистрироваться.

Код из класса RegisterActivity.java:

import ndroid.appcompat.app.AppCompatActivity;  
import ndroid.appcompat.widget.AppCompatButton;  
  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.text.format.DateUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.Calendar;  
  
public class RegisterActivity extends AppCompatActivity {  
  
 EditText FirstName,LastName,Login,Password,Email;  
 private static final String *url* = “jdbc:mysql://46.229.213.172:3306/evator”;  
 private static final String *user* = “evator”;  
 private static final String *pass* = “rYlSwr2ip1w4”;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
 FirstName = findViewById(R.id.*editTextTextPersonName2*);  
 LastName = findViewById(R.id.*editTextTextPersonName3*);  
 Login = findViewById(R.id.*editTextTextPersonName4*);  
 Password=findViewById(R.id.*editTextTextPersonName5*);  
 Email = findViewById(R.id.*editTextTextPersonName6*);  
 AppCompatButton btn = findViewById(R.id.*button3*);  
 btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 ConnectMySql connectMySql = new ConnectMySql();  
 connectMySql.execute(“”);  
 }  
 });  
  
 }  
  
 public void registration(View v){  
  
 }  
  
 private class ConnectMySql extends AsyncTask<String, Void, String> {  
 String res = “”;  
  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 Toast.*makeText*(RegisterActivity.this, “Please wait…”, Toast.*LENGTH\_SHORT*)  
 .show();  
  
 }  
  
 @Override  
 protected String doInBackground(String… params) {  
 try {  
 Class.*forName*(“com.mysql.jdbc.Driver”);  
 Connection con = DriverManager.*getConnection*(*url*, *user*, *pass*);  
 System.*out*.println(“Databaseection success”);  
  
 String result = “”;  
 String query =”INSERT INTO stasy (FirstName, LastName, Login, Password, `Email`) VALUES (?, ?, ?, ?, ?);”;  
 PreparedStatement preparedStatement = con.prepareStatement(query);  
 preparedStatement.setString(1, FirstName.getText().toString());  
 preparedStatement.setString(2, LastName.getText().toString());  
 preparedStatement.setString(3, Login.getText().toString());  
 preparedStatement.setString(4, Password.getText().toString());  
 preparedStatement.setString(5, Email.getText().toString());  
 preparedStatement.executeUpdate();  
 result=”Регистрация успешна”;  
 finish();  
  
// while (rs.next()) {  
// result += rs.getString(1).toString() + “\n”;  
// }  
 res = result;  
 } catch (Exception e) {  
 e.printStackTrace();  
 res = e.toString();  
 }  
 return res;  
 }  
  
 @Override  
 protected void onPostExecute(String result) {  
 Toast.*makeText*(RegisterActivity.this, result, Toast.*LENGTH\_SHORT*)  
 .show();  
 }  
 }  
}

В данном классе будет осуществляться регистрация пользователя в приложение, для этого необходимо ввести данные (имя, фамилия, логин, пароль, электронная почта). Для окончания регистрации необходимо нажать на кнопку зарегистрироваться, если регистрация прошла успешна, пользователь перейдет в главное меню приложения.

Код из класса MainActivity.java:

package com.example.stasyagame;  
  
import ndroid.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 private long backPressedTime;  
 private Toast backToast;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Button buttonStart = (Button)findViewById(R.id.*buttonStart*);  
  
 buttonStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try{  
 Intent intent = new Intent( MainActivity.this, Gamelevels.class);  
 startActivity(intent);finish();  
 }catch(Exception e) {  
  
 }  
 }  
 });  
  
 Button buttonlesson = (Button)findViewById(R.id.*buttonlesson*);  
  
 buttonlesson.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try{  
 Intent intent = new Intent( MainActivity.this, Gamelessons.class);  
 startActivity(intent);finish();  
 }catch(Exception e) {  
  
 }  
 }  
 });  
  
 Window w = getWindow();  
 w.setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
 }  
  
 //системная кнопка назад  
  
 @Override  
 public void onBackPressed() {  
  
  
 if(backPressedTime + 2000 > System.*currentTimeMillis*()){  
 backToast.cancel();  
 super.onBackPressed();  
 return;  
 }else{  
 backToast = Toast.*makeText*(getBaseContext(), “Нажмите дважды,чтобы выйти”, Toast.*LENGTH\_SHORT*);  
 backToast.show();  
 }  
 backPressedTime = System.*currentTimeMillis*();  
 }  
}

В данном классе размещено две кнопки. Кнопка уроки и кнопка тесты. При нажатии на первую пользователь переходит на страницу выбора урока, а на вторую, пользователь попадет на страницу выбора уровня.

Код из класса Gamelessons.java:

package com.example.stasyagame;  
  
import ndroid.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class Gamelessons extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*lessons*);  
  
 Window w = getWindow();  
 w.setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
  
 Button buttonback = (Button) findViewById(R.id.*buttonback*);  
  
 buttonback.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //кнопка назад  
 try{  
 Intent intent = new Intent(Gamelessons.this, MainActivity.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
 }  
 });  
  
 TextView textView1 = (TextView) findViewById(R.id.*textView1*);  
 TextView textView2 = (TextView) findViewById(R.id.*textView2*);  
 TextView textView3 = (TextView) findViewById(R.id.*textView3*);  
 TextView textView4 = (TextView) findViewById(R.id.*textView4*);  
 TextView textView5 = (TextView) findViewById(R.id.*textView5*);  
 TextView textView6 = (TextView) findViewById(R.id.*textView6*);  
 TextView textView7 = (TextView) findViewById(R.id.*textView7*);  
 TextView textView8 = (TextView) findViewById(R.id.*textView8*);  
 TextView textView9 = (TextView) findViewById(R.id.*textView9*);  
 TextView textView10 = (TextView) findViewById(R.id.*textView10*);  
 TextView textView11 = (TextView) findViewById(R.id.*textView11*);  
 TextView textView12 = (TextView) findViewById(R.id.*textView12*);  
 TextView textView13 = (TextView) findViewById(R.id.*textView13*);  
 TextView textView14 = (TextView) findViewById(R.id.*textView14*);  
 TextView textView15 = (TextView) findViewById(R.id.*textView15*);  
  
 textView1.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 try{  
 Intent intent = new Intent(Gamelessons.this, Lesson1.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
  
 }  
 });  
  
 //заблокированные уровни  
  
 textView4.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView5.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView6.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView7.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView8.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView9.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView10.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView11.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView12.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView13.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });textView14.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });textView15.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelessons.this, “Урок заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
  
 }  
 //системная кнопка назад  
 @Override  
 public void onBackPressed(){  
 try{  
 Intent intent = new Intent(Gamelessons.this, MainActivity.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
 }  
}

В данном классе пользователю необходимо выбрать урок для прохождения. Для пользователя без подписки доступно всего 3 урока, остальные заблокированы. Если нажать на заблокированный уровень всплывет сообщение:

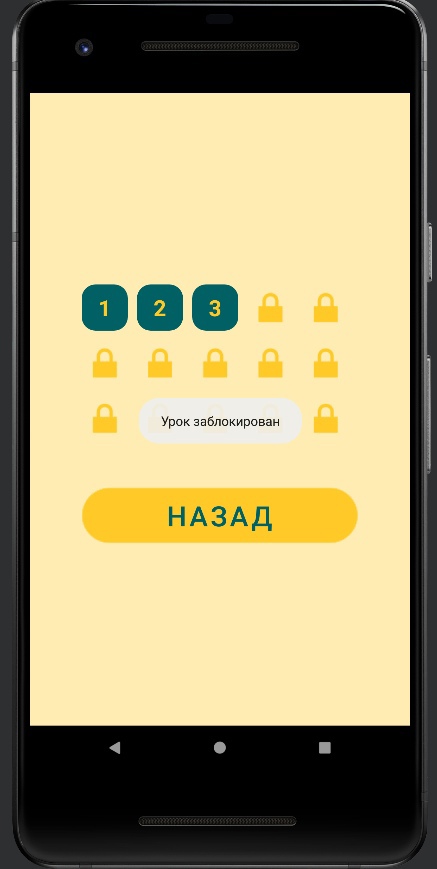


Рисунок 25. Уведомление о заблокированном уровне

Код из класса Gamelevels.java:

package com.example.stasyagame;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Gravity;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import ndroid.appcompat.app.AppCompatActivity;  
  
public class Gamelevels extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*levels*);  
  
 Window w = getWindow();  
 w.setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
  
 Button buttonback = (Button) findViewById(R.id.*buttonback*);  
 buttonback.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 //кнопка назад  
 try{  
 Intent intent = new Intent(Gamelevels.this, MainActivity.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
 }  
 });  
  
 TextView textView1 = (TextView) findViewById(R.id.*textView1*);  
 TextView textView4 = (TextView) findViewById(R.id.*textView4*);  
 TextView textView5 = (TextView) findViewById(R.id.*textView5*);  
 TextView textView6 = (TextView) findViewById(R.id.*textView6*);  
 TextView textView7 = (TextView) findViewById(R.id.*textView7*);  
 TextView textView8 = (TextView) findViewById(R.id.*textView8*);  
 TextView textView9 = (TextView) findViewById(R.id.*textView9*);  
 TextView textView10 = (TextView) findViewById(R.id.*textView10*);  
 TextView textView11 = (TextView) findViewById(R.id.*textView11*);  
 TextView textView12 = (TextView) findViewById(R.id.*textView12*);  
 TextView textView13 = (TextView) findViewById(R.id.*textView13*);  
 TextView textView14 = (TextView) findViewById(R.id.*textView14*);  
 TextView textView15 = (TextView) findViewById(R.id.*textView15*);  
  
 textView1.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 try{  
 Intent intent = new Intent(Gamelevels.this, Level1.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
  
 }  
 });  
 //заблокированные уровни  
  
 textView4.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView5.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView6.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView7.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView8.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView9.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView10.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован”,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView11.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, “Уровень заблокирован»,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });

textView12.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, «Уровень заблокирован»,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });  
 textView13.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, «Уровень заблокирован»,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });textView14.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, «Уровень заблокирован»,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();  
  
 }  
 });textView15.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Toast toast = Toast.*makeText*(Gamelevels.this, «Уровень заблокирован»,Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*CENTER*, 1,1);  
 toast.show();

}  
 });  
 }

//системная кнопка назад  
 @Override  
 public void onBackPressed(){  
 try{  
 Intent intent = new Intent(Gamelevels.this, MainActivity.class);  
 startActivity(intent);finish();  
  
 }catch (Exception e) {  
  
 }  
 }  
}

В данном классе пользователю необходимо выбрать уровень для прохождения. Для пользователя без подписки доступно всего 3 уровня, остальные заблокированы. Если нажать на заблокированный уровень всплывет сообщение «Уровень заблокирован».

Код из класса Level1.java:

package com.example.stasyagame;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.Dialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.Window;  
import android.view.WindowManager;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import java.util.Random;  
  
public class Level1 extends AppCompatActivity {  
  
 Dialog dialog;  
 Dialog dialogEnd;  
  
 public int numLeft; //переменная для левой картинки  
 public int numRight; //переменная для правой картинки  
 Array array = new Array(); //новый объект из класса Array  
  
  
 public int count = 0; //cчетчик правильных ответов  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*universal*);  
  
 //переменная text\_levels  
  
 TextView text\_levels = findViewById(R.id.*textlevels*);  
 text\_levels.setText(R.string.*level1*);  
  
  
 //левая часть  
 final ImageView imgleft = (ImageView) findViewById(R.id.*imgleft*);  
 //скругление углов  
 imgleft.setClipToOutline(true);  
  
 //правая часть  
 final ImageView imgright = (ImageView) findViewById(R.id.*imgright*);  
 //скругление углов  
 imgright.setClipToOutline(true);  
  
 Window w = getWindow();  
 w.setFlags(WindowManager.LayoutParams.*FLAG\_LAYOUT\_NO\_LIMITS*, WindowManager.LayoutParams.*FLAG\_LAYOUT\_NO\_LIMITS*);  
  
 //вызов диалогового окна  
 dialog = new Dialog(this); //создание  
 dialog.requestWindowFeature(Window.*FEATURE\_NO\_TITLE*); //скрытие заголовка  
 dialog.setContentView(R.layout.*previewdialogtest*); //путь к макету диалогового окна  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*)); //прозрачный фон  
 dialog.setCancelable(false); //окно нельзя закрыть кнопкой назад  
  
 //кнопка закрытия  
 TextView btnclose = (TextView) dialog.findViewById(R.id.*btnclose*);  
 btnclose.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try {  
 //Вернуться назад к выбору урока  
 Intent intent = new Intent(Level1.this, Gamelevels.class);  
 startActivity(intent);  
 finish();  
  
 } catch (Exception e) {  
  
 }  
 dialog.dismiss(); //закрытие диалогового окна  
 }  
 });  
  
 //кнопка продолжить  
 Button btncontinue = (Button) dialog.findViewById(R.id.*btncontinue*);  
 btncontinue.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 dialog.dismiss();  
 }  
 });  
  
 dialog.show();  
  
 //\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
 //вызов диалогового окна в конце игры  
 dialogEnd = new Dialog(this); //создание  
 dialogEnd.requestWindowFeature(Window.*FEATURE\_NO\_TITLE*); //скрытие заголовка  
 dialogEnd.setContentView(R.layout.*dialogendtest*); //путь к макету диалогового окна  
 dialogEnd.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*)); //прозрачный фон  
 dialogEnd.getWindow().setLayout(WindowManager.LayoutParams.*MATCH\_PARENT*, WindowManager.LayoutParams.*MATCH\_PARENT*);  
 dialogEnd.setCancelable(false); //окно нельзя закрыть кнопкой назад  
  
 //кнопка закрытия  
 TextView btnclose2 = (TextView) dialogEnd.findViewById(R.id.*btnclose*);  
 btnclose2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try {  
 //Вернуться назад к выбору урока  
 Intent intent = new Intent(Level1.this, Gamelevels.class);  
 startActivity(intent);  
 finish();  
  
 } catch (Exception e) {  
  
 }  
 dialogEnd.dismiss(); //закрытие диалогового окна  
 }  
 });  
  
 //кнопка продолжить  
 Button btncontinue2 = (Button) dialogEnd.findViewById(R.id.*btncontinue*);  
 btncontinue2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 try {  
  
 Intent intent = new Intent(Level1.this, Level2.class);  
 startActivity(intent);  
 finish();  
  
 } catch (Exception e) {  
  
 }  
  
  
 dialogEnd.dismiss();  
 }  
 });  
  
 //\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
 //кнопка назад  
 Button buttonback = (Button) findViewById(R.id.*buttonback*);  
 buttonback.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 try {  
 Intent intent = new Intent(Level1.this, Gamelevels.class);  
 startActivity(intent);  
 finish();  
 } catch (Exception e) {  
  
 }  
 }  
 });  
  
 //массив для прогресса  
 final int[] progress = {  
 R.id.*point1*, R.id.*point2*, R.id.*point3*, R.id.*point4*, R.id.*point5*, R.id.*point6*,  
 };  
  
  
 //анимация  
 final Animation a = AnimationUtils.*loadAnimation*(Level1.this, R.anim.*alpha*);  
  
  
 int i = 0, k = 1;  
  
 numLeft = i;  
 imgleft.setImageResource(array.images1[i]); //достаем из массива картинку  
  
 numRight = k;  
  
 //цикл проверяющий равенство чисел  
  
  
 imgright.setImageResource(array.images1[k]);  
  
 //обработка нажатия на левую картинку  
 imgleft.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View view, MotionEvent event) {  
 //условие касания картинки  
 if (event.getAction() ==MotionEvent.*ACTION\_DOWN*){  
  
 imgright.setEnabled(false); //блокирование правой картинки  
 if (numLeft > numRight){  
 imgleft.setImageResource(R.drawable.*img\_true*);  
 }else{  
 imgleft.setImageResource(R.drawable.*img\_false*);  
 }  
  
 }else if(event.getAction() ==MotionEvent.*ACTION\_UP*){  
  
 if(numLeft>numRight){  
 //если левая картинка больше  
 if(count<6){  
 count=count+1;  
 }  
  
 //заполнение очков  
 for (int i=0; i<6; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points*);  
 }  
  
 //определение правильных ответов и закрашивание  
 for (int i=0; i<count; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points\_green*);  
 }  
  
 }else{  
 //елси левая картинка меньше  
 if(count>0){  
 if(count==1){  
 count=0;  
 }else{  
 count=count-2;  
  
 }  
 }  
  
 //заполнение очков  
 for (int i=0; i<5; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points*);  
 }  
  
 //определение правильных ответов и закрашивание  
 for (int i=0; i<count; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points\_green*);  
 }  
 }  
 if(count==6){  
 //выход из уровня  
 dialogEnd.show();  
  
 }else{  
  
 if(numLeft==10){  
 numLeft=0;  
 }else{  
 numLeft += 2;  
 }  
  
 imgleft.setImageResource(array.images1[numLeft]); //достаем из массива картинку  
 imgleft.startAnimation(a);  
  
 if(numRight==11){  
 numRight=1;  
 }else{  
 numRight += 2;  
 }  
  
 imgright.setImageResource(array.images1[numRight]);  
 imgright.startAnimation(a);  
 imgright.setEnabled(true); //разблокировка правой картинки  
  
 }  
  
 }  
  
  
 return true;  
 }  
 });  
 //обработка нажатия на правую картинку  
 imgright.setOnTouchListener(new View.OnTouchListener() {  
 @Override  
 public boolean onTouch(View view, MotionEvent event) {  
 //условие касания картинки  
 if (event.getAction() ==MotionEvent.*ACTION\_DOWN*){  
  
 imgleft.setEnabled(false); //блокирование левой картинки  
 if (numLeft < numRight){  
 imgright.setImageResource(R.drawable.*img\_true*);  
 }else{  
 imgright.setImageResource(R.drawable.*img\_false*);  
 }  
  
 }else if(event.getAction() ==MotionEvent.*ACTION\_UP*){  
  
 if(numLeft < numRight){  
 //если правая картинка больше  
 if(count<6){  
 count=count+1;  
 }  
  
 //заполнение очков  
 for (int i=0; i<6; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points*);  
 }  
  
 //определение правильных ответов и закрашивание  
 for (int i=0; i<count; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points\_green*);  
 }  
  
 }else{  
 //елси правая картинка меньше  
 if(count>0){  
 if(count==1){  
 count=0;  
 }else{  
 count=count-2;  
  
 }  
 }  
  
 //заполнение очков  
 for (int i=0; i<5; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points*);  
 }  
  
 //определение правильных ответов и закрашивание  
 for (int i=0; i<count; i++){  
 TextView tv = findViewById(progress[i]);  
 tv.setBackgroundResource(R.drawable.*style\_points\_green*);  
 }  
 }  
 if(count==6){  
 //выход из уровня  
 dialogEnd.show();  
  
 }else{  
  
 numLeft += 2;  
 imgleft.setImageResource(array.images1[numLeft]); //достаем из массива картинку  
 imgleft.startAnimation(a);  
 numRight += 2;  
 imgright.setImageResource(array.images1[numRight]);  
 imgright.startAnimation(a);  
 imgleft.setEnabled(true); //разблокировка левой картинки  
  
 }  
  
 }  
  
  
 return true;  
 }  
 });  
  
 }  
 //системная кнопка назад  
 @Override  
 public void onBackPressed(){  
 try {  
 Intent intent = new Intent(Level1.this, Gamelevels.class);  
 startActivity(intent);  
 finish();  
 }catch (Exception e){  
  
 }  
 }  
}

В данном классе пользователю необходимо пройти уровень. Перед началом всплывает диалоговое окно с подсказкой. Для прохождения уровня на максимальное кол-во очков, будет достаточно изучить урок, которому соответствует уровень. Чтобы пройти уровень необходимо выбирать правильный ответ. С каждым правильным ответом игрок зарабатывает одно очко, с неправильным теряет два. Вопросы будут повторяться, если игрок ответит неправильно. После прохождения уровня всплывает диалоговое окно об окончании уровня с интересным фактом по теме урока.

Код из класса Array(массив для картинок):

package com.example.stasyagame;  
  
public class Array {  
  
 //массив для первого уровня  
  
 final int[] images1 = {  
 R.drawable.*levelone\_11*,  
 R.drawable.*levelone\_1*,  
 R.drawable.*levelone\_2*,  
 R.drawable.*levelone\_22*,  
 R.drawable.*levelone\_33*,  
 R.drawable.*levelone\_3*,  
 R.drawable.*levelone\_44*,  
 R.drawable.*levelone\_4*,  
 R.drawable.*levelone\_5*,  
 R.drawable.*levelone\_55*,  
 R.drawable.*levelone\_66*,  
 R.drawable.*levelone\_6*,  
   
 };  
}

**2.4 Разработка графического пользовательского интерфейса**

Приводится описание форм программы, расположение виджетов на экране. Обосновывается выбор того или иного способа отражения информации. Приводятся примеры удобства интерфейса с точки зрения UI/UX.

Код из файла entry.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Entry">  
  
 <EditText  
 android:id="@+id/editTextTextPersonName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="48dp"  
 android:ems="10"  
 android:hint="Логин"  
 android:onClick="login"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toTopOf="@+id/editTextTextPassword"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.921" />  
  
 <EditText  
 android:id="@+id/editTextTextPassword"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:ems="10"  
 android:hint="Пароль"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/buttonshape"  
 android:text="Войти"  
 android:textColor="#FFFFFF"  
 app:layout\_constraintBottom\_toTopOf="@+id/button2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_marginBottom="328dp"  
 android:background="@drawable/buttonshape"  
 android:text="Зарегестрироваться"  
 android:onClick="registration"  
 android:textColor="#FFFFFF"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

В данной активности размещены две кнопки и два поля Edit text для ввода текста. Кнопка войти служит для перехода в активность activity\_main.xml, а кнопка зарегистрироваться для перехода в активность activity\_register.xml.

Рисунок 26. Интерфейс страницы Entry

Код из файла activity\_register.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RegisterActivity">  
  
 <EditText  
 android:id="@+id/editTextTextPersonName2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_marginTop="72dp"  
 android:ems="10"  
 android:hint="Имя"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="TouchTargetSizeCheck" />  
  
 <EditText  
 android:id="@+id/editTextTextPersonName3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:ems="10"  
 android:hint="Фамилия"  
 android:inputType="textPersonName"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName2" />  
  
 <EditText  
 android:id="@+id/editTextTextPersonName4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Логин"  
 android:inputType="textPersonName"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName3" />  
  
 <EditText  
 android:id="@+id/editTextTextPersonName5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_marginTop="8dp"  
 android:ems="10"  
 android:hint="Пароль"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName4" />  
  
 <EditText  
 android:id="@+id/editTextTextPersonName6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:ems="10"  
 android:hint="Электронная почта"  
 android:inputType="textPersonName"  
 android:minHeight="48dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName5" />  
  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:background="@drawable/buttonshape"  
 android:text="Зарегестрироваться"  
 android:textColor="#FDFCFC"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName6" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

В данной активности размещена одна кнопка и пять полей Edit text для ввода текста. Кнопка зарегистрироваться служит для перехода в активность activity\_main.xml.

Рисунок 27. Интерфейс страницы RegisterActivity

Код из файла activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/buttonstart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFECB3"  
 tools:context=".MainActivity">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:layout\_editor\_absoluteY="0dp">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:src="@drawable/logo" />  
  
  
  
 <LinearLayout  
 android:id="@+id/startlayout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:orientation="vertical"  
 android:padding="100dp">  
  
 <Button  
 android:id="@+id/buttonlesson"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:text="Уроки"  
 android:textColor="@color/green"  
 android:textSize="33sp" />  
  
 <Button  
 android:id="@+id/buttonStart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_weight="1"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:text="Тесты"  
 android:textColor="@color/green"  
 android:textSize="33sp" />  
 </LinearLayout>  
  
 </RelativeLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

В данной активности размещены две кнопки и логотип. Кнопка уроки служит для перехода в активность lessons.xml, а кнопка тесты для перехода в активность levels.xml.

Рисунок 28. Интерфейс страницы MainActivity

Код из файла lessons.xml:

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFECB3"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:id="@+id/container1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:id="@+id/container2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp">  
  
 <LinearLayout  
 android:id="@+id/container4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:id="@+id/line1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="1"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="2"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="3"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView6"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView7"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView8"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView9"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView10"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView11"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView12"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView13"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView14"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView15"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
 </LinearLayout>  
 </ScrollView>  
  
 <LinearLayout  
 android:id="@+id/container3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:paddingTop="50dp">  
  
 <Button  
 android:id="@+id/buttonback"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:gravity="center"  
 android:paddingLeft="10dp"  
 android:paddingRight="10dp"  
 android:text="@string/back"  
 android:textColor="@color/green"  
 android:textSize="30sp" />  
 </LinearLayout>  
  
 </LinearLayout>  
 </LinearLayout>  
</RelativeLayout>

В данной активности размещены 15 TextView и кнопка. Кнопка назад служит для перехода в активность activity\_main.xml. TextView являются кнопками, нажатие на которые переносит игрока на уровни. 3 TextView имеют номера уровней, а остальные отображаются картинкой с замком.

Рисунок 29. Интерфейс страницы Gamelessons

Код из файла levels.xml:

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFECB3"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:id="@+id/container1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:id="@+id/container2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp">  
  
 <LinearLayout  
 android:id="@+id/container4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:id="@+id/line1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="1"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="2"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@drawable/button\_gamelevels"  
 android:gravity="center"  
 android:text="3"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView6"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView7"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView8"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView9"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView10"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView11"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView12"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView13"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView14"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView15"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
  
 android:layout\_marginRight="10dp"  
 android:background="@android:drawable/ic\_lock\_lock"  
 android:gravity="center"  
 android:textColor="@color/orange"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
 </LinearLayout>  
 </ScrollView>  
  
 <LinearLayout  
 android:id="@+id/container3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:paddingTop="50dp">  
  
 <Button  
 android:id="@+id/buttonback"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:gravity="center"  
 android:paddingLeft="10dp"  
 android:paddingRight="10dp"  
 android:text="@string/back"  
  
 android:textColor="@color/green"  
 android:textSize="30sp" />  
 </LinearLayout>  
  
 </LinearLayout>  
 </LinearLayout>  
</RelativeLayout>

В данной активности размещены 15 TextView и кнопка. Кнопка назад служит для перехода в активность activity\_main.xml. TextView являются кнопками, нажатие на которые переносит игрока на уровни. 3 TextView имеют номера уровней, а остальные отображаются картинкой с замком. Интерфейс страницы идентичен с lessons.xml

Для того, чтобы приложение отображалось корркетно на разных рамерах экранов, была создана адаптивная верстка.

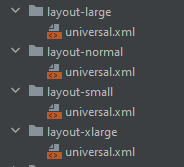


Рисунок 30. Адаптивная верстка проекта

Для уроков и уровней сделан универсальный макет, потому что меняться во время прохождения будут только картинки в layout.

Код из файла universal.xml:

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFECB3"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:id="@+id/container1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:id="@+id/container2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <LinearLayout  
 android:id="@+id/line1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/buttonback"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:layout\_weight="1"  
 android:layout\_marginRight="30dp"  
 android:text="@string/back"  
 android:textSize="40sp"  
 android:paddingLeft="5dp"  
 android:paddingRight="5dp"  
 android:textColor="@color/green"  
 android:textAllCaps="false"/>  
  
 <TextView  
 android:id="@+id/textlessons"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="@string/level"  
 android:textSize="40sp"  
 android:textColor="@color/green"  
 android:gravity="center"  
 android:padding="10dp"/>  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
 <TextView  
 android:id="@+id/point1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
 <TextView  
 android:id="@+id/point2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
 <TextView  
 android:id="@+id/point3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
 <TextView  
 android:id="@+id/point4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
 <TextView  
 android:id="@+id/point5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
 <TextView  
 android:id="@+id/point6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="10dp"  
 android:layout\_weight="1"  
 android:background="@drawable/style\_points" />  
  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
  
 <ImageView  
 android:id="@+id/imgleft"  
 android:layout\_width="match\_parent"  
 android:layout\_height="320dp"  
 android:layout\_marginRight="5dp"  
 android:scaleType="centerCrop"  
 android:background="@drawable/style\_img\_universal"  
 android:layout\_weight="1"  
 android:src="@drawable/imgtest" />  
 <ImageView  
 android:id="@+id/imgright"  
 android:layout\_width="match\_parent"  
 android:layout\_height="320dp"  
 android:layout\_marginRight="5dp"  
 android:scaleType="centerCrop"  
 android:background="@drawable/style\_img\_universal"  
 android:layout\_weight="1"  
 android:src="@drawable/imgtest" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/line4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal">  
  
  
  
 </LinearLayout>  
 </LinearLayout>  
 </LinearLayout>  
</RelativeLayout>

В данной активности размещены 6 TextView, кнопка и два ImageView. Кнопка назад служит для перехода в активность levels.xml. TextView являются очками, при правильных ответах они заполняются. Также отображается название уровня. ImageView служит для размещения картинок, которые необходимо выбирать.

Рисунок 31. Интерфейс универсального шаблона для уровней

Код из файла prewievdialogtest.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id = "@+id/dialogfon"  
 android:background="@drawable/preview"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:layout\_margin="10dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="20dp"  
 android:gravity="center">  
 <TextView  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:id="@+id/btnclose"  
 android:text="X"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:layout\_gravity="end"/>  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:id="@+id/previewimg"  
 android:src="@drawable/prewiewimg"  
 android:layout\_weight="60"  
 android:layout\_marginTop="10dp"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:id="@+id/textdiscription"  
 android:text="@string/levelone"  
 android:layout\_weight="30"  
 android:textColor="@color/green"  
 android:textSize="24sp"  
 android:layout\_marginTop="10dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:id="@+id/btncontinue"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:layout\_weight="15"  
 android:text="@string/textcontinue"  
 android:textColor="@color/green"  
 android:paddingLeft="10dp"  
 android:paddingRight="10dp"  
 android:textAllCaps="true"  
 android:textSize="24sp"  
 android:layout\_marginTop="10dp" />  
 </LinearLayout>  
</LinearLayout>

Данная активность служит диалоговым окном и всплывает перед началом уровня.

Рисунок 32. Диалоговое окно в начале уровня

Перед уроком также всплывает диалоговое окно, код идентичен диалоговому окну уровня.

Рисунок 33. Диалоговое окно в начале урока

Код из файла dialogtest.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id = "@+id/dialogfon"  
 android:background="@drawable/preview"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:layout\_margin="10dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="20dp"  
 android:gravity="center">  
 <TextView  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:id="@+id/btnclose"  
 android:text="X"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:layout\_gravity="end"/>  
  
 <ScrollView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:layout\_weight="90"  
 android:layout\_marginTop="10dp">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/textdiscriptionend"  
 android:text="@string/leveloneEnd"  
 android:textColor="@color/green"  
 android:textSize="24sp"/>  
 </ScrollView>  
  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btncontinue"  
 android:background="@drawable/button\_stroke\_black95\_press\_white"  
 android:layout\_weight="10"  
 android:text="@string/textcontinue"  
 android:textColor="@color/green"  
 android:paddingLeft="10dp"  
 android:paddingRight="10dp"  
 android:textAllCaps="true"  
 android:textSize="24sp"  
 android:layout\_marginTop="10dp" />  
 </LinearLayout>

Данная активность служит диалоговым окном и всплывает после окончания уровня.

Рисунок 34. Диалоговое окно в конце уровня

Приложение готово, основные принципы его работы представлены на рисунках 35-39.

Основная задача пользователя на уровне – выбирать правильный ответ из двух предложенных, это делается касанием выбранного ответа.

Рисунок 35. Интерфейс 1го уровня

После нажатия, приложение анализирует ответ и отображает на экране. При правильном ответе отображается картинка с зеленой галочкой, при неправильном ответе – картинка с красным крестиком.

Рисунок 36. Интерфейс при правильном ответе

Рисунок 37. Интерфейс при неправильном ответе

При правильном ответе, пользователю начисляется 1 балл, при неправильном отнимается 2.

Рисунок 38. Заполнение очков

Чтобы пройти уовень, пользователю необходимо изучить материал на уроке. С помощью нажатия на картинку с материалом, она сменяется на другую. Количество заданий на уровне зависит от количества материала на уроке. Если какая-то информация была плохо усвоена, пользователь может пройти урок еще раз.

Рисунок 39. Интерфейс урока

**2.5 Тестирование приложения**

Таким образом, разработав android приложение можно произвести тестирование.

Unit–тестирование - это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Цель этого тестирования состоит в том, чтобы убедиться, что каждая единица программного кода работает должным образом. При Unit – тестировании возрастает качество разрабатываемого разрабатываемого продукта. Модульное тестирование – прочная основа эффективной стратегии автоматизированного тестирования, суть которой – проверить небольшие изолированные фрагменты ПО. Это непростая задача, которая требует определенной дисциплины. Кроме того, тестирование – это тоже код, к которому нужен не менее тщательный подход, чем к основному коду.

Для тестирования функционала разработанного мобильного приложения, необходимо создать новый класс. Тестриоваться будет функция ввода почты при регистрации.

Код из класса EmailValidatorTest.java:

import org.junit.Test;  
import static org.junit.Assert.\*;  
import org.junit.Assert.assertFalse;  
import org.junit.Assert.assertTrue;  
  
class EmailValidatorTest {  
  
 @Test  
 emailValidator\_CorrectEmailSimple\_ReturnsTrue() {  
 *assertTrue*(EmailValidator.isValidEmail("name@mail.com"));  
 }  
  
 @Test  
 emailValidator\_CorrectEmailSubDomain\_ReturnsTrue() {  
 *assertTrue*(EmailValidator.isValidEmail("name@mail.ru"));  
 }  
  
 @Test  
 emailValidator\_InvalidEmailNoTld\_ReturnsFalse() {  
 *assertFalse*(EmailValidator.isValidEmail("name@mail"));  
 }  
  
 @Test  
 emailValidator\_InvalidEmailDoubleDot\_ReturnsFalse() {  
 *assertFalse*(EmailValidator.isValidEmail("name@mail..com"));  
 }  
  
 @Test  
 emailValidator\_InvalidEmailNoUsername\_ReturnsFalse() {  
 *assertFalse*(EmailValidator.isValidEmail("@mail.com"));  
 }  
  
 @Test  
 emailValidator\_EmptyString\_ReturnsFalse() {  
 *assertFalse*(EmailValidator.isValidEmail(""));  
 }  
  
 @Test  
 emailValidator\_NullEmail\_ReturnsFalse() {  
 *assertFalse*(EmailValidator.isValidEmail(null));  
 }  
}

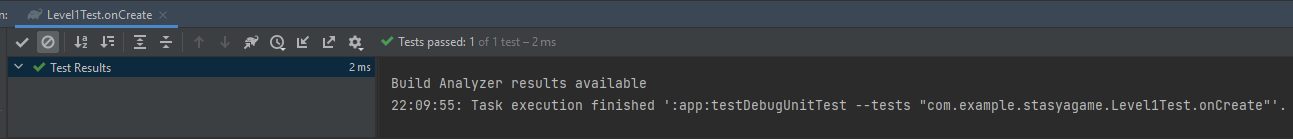


Рисунок 40. Результат тестирования

Таким образом, при тестировании разработанного Android-приложения ошибок не обнаружено. Можно заметить, что заданная информация отображается корректно.

Таблица 22

Тест-кейс тестового случая

|  |  |
| --- | --- |
|  | **Описание** |
| **Название проекта** | Мобильное приложение, обучающее программированию детей дошкольного возраста |
| **Рабочая версия** | Версия проекта 1.0 |
| **Заголовок/название теста** | Проверка функции ввода почты при регистрации |
| **Тестовые данные** | Почта для подтверждения регистрации. |
| **Фактический результат** | Тест сразу запустится, результат удачен. Конфигурация для прогона тестов создалась автоматически. |
| **Статус** | Фактический результат соответствует ожидаемому результату. Тест удачен. |
| **Примечания** | Проверку проходили все очевидные случаи. Допускается, что в тестовом классе проверяются не все возможные Утверждения. |

**ЗАКЛЮЧЕНИЕ**

Мобильное приложение, обучающее программированию детей дошкольного возраста **-** это программное обеспечение, специально разработанное под конкретную мобильную платформу, предназначенное для изучения программирования на смартфонах, планшетах и других мобильных устройствах.

Уникальностью данного разрабатываемого продукта является то, что он создан для детей дошкольного и младшего школьного возраста. Приложение является в первую очередь обучающим, и его главная цель помочь получить знания в легкой игровой форме, а не перепродажа товаров.

В ходе написания выпускной квалификационной работы было проведено:

1. Моделирование бизнесс-процессов
2. Экономическое обоснование разработки приложения
3. Сравнительный анализ систем аналогов
4. Разработка базы данных для приложения
5. Разрабокта мобильного приложения на операционной системе Android в среде Android Studio
6. Разработка графического пользовательского интерфейса приложения
7. Тестирование приложения.

Поставленная цель, а именно разработка мобильного приложения, обучающего программированию детей дошкольного возраста, достигнута, все задачи выполнены.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

Нормативно-правовые акты

1. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ (последняя редакция)
2. ГОСТ 34.321-96 Информационные технологии (ИТ). Система стандартов по базам данных. Эталонная модель управления данными
3. ГОСТ 34.320-96 Информационные технологии (ИТ). Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы
4. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств
5. ГОСТ Р ИСО/МЭК ТО 10032-2007. Национальный стандарт Российской Федерации. Эталонная модель управления данными
6. ГОСТ Р ИСО/МЭК 11179-1-2010. Информационная технология. Регистры метаданных (РМД). Часть 1. Основные положения
7. ГОСТ Р 43.0.11-2014 Информационное обеспечение техники и операторской деятельности. Базы данных в технической деятельности
8. ГОСТ 19.404-79. Единая система программной документации. Пояснительная записка. Требования к содержанию и оформлению. – М.: Издательство стандартов, 1981.
9. ГОСТ 19.402-78. Единая система программной документации. Описание программы. Требования к содержанию и оформлению. – М.: Издательство стандартов, 1980.
10. ГОСТ 19.401-78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению. – М.: Издательство стандартов, 1980
11. ГОСТ 19.504-79 Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению. – М.: Издательство стандартов, 1980.

Основная литература

1. Вязовик, Н. А. Программирование на Java: учебное пособие для СПО / Н. А. Вязовик. — Саратов: Профобразование, 2019. — 604 c. — ISBN 978-5-4488-0365-9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/86206.html
2. Горбовцов Г. Я., Грызина Н. Ю., Мастяева И. Н., Семенихина О. Н. Исследование операций в экономике: учебное пособие: Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2006. — 118 c. — ISBN 5-7764-0272-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/10690>.
3. Джошуа, Блох Java. Эффективное программирование / Блох Джошуа; перевод В. Стрельцов; под редакцией Р. Усманов. — Саратов: Профобразование, 2017. — 310 c. — ISBN 978-5-4488-0127-3. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/64057.html
4. Ковалевская, Е. В. Методы программирования: учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — М.: Евразийский открытый институт, 2011. — 320 c. — ISBN 978-5-374-00356-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/10784.html
5. Описание нотации IDEF0 [Электронный ресурс]. – Режим доступа: https://micro-solution.ru/bp-know/regulation/IDEF0
6. Простое руководство по UML-диаграммам и моделированию баз данных [Электронный ресурс]. – Режим доступа: https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling
7. Савельев А.О. Проектирование и разработка веб-приложений на основе технологий Microsoft [Электронный ресурс]: учебное пособие/ Савельев А.О., Алексеев А.А.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 418 c.— Режим доступа: http://www.iprbookshop.ru/94860.html.— ЭБС «IPRbooks»
8. Сальникова, Н. А. Информатика. Моделирование. Программирование. Часть 2: учебное пособие / Н. А. Сальникова. — Волгоград: Волгоградский институт бизнеса, Вузовское образование, 2009. — 142 c. — ISBN 2227-8397. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: http://www.iprbookshop.ru/11320.html
9. Тарасов С.В. СУБД для программиста. Базы данных изнутри [Электронный ресурс]/ Тарасов С.В.— Электрон. текстовые данные.— Москва: СОЛОН-Пресс, 2018.— 320 c.— Режим доступа: http://www.iprbookshop.ru/90409.html.— ЭБС «IPRbooks»