## ВВЕДЕНИЕ

С каждым годом интерес к здоровому образу жизни растет. Огромное множество людей стараются следить за собой, люди отказываются от вредных привычек, начинают правильно питаться и, конечно же, увлекаются спортом. Фитнес – образ жизни, ведущий к физическому и ментальному здоровью человека, включающий в себя и занятия в тренажерном зале, йогу и растяжку. Однако, далеко не все имеют возможность посещать спортивные клубы, но очень хотят заниматься тренировками. Благодаря использованию компьютерных информационных технологий в фитнес-индустрии, множество людей получили возможность тренироваться и получать советы профессиональных тренеров, будучи дома.

**Целью** курсовой работы является разработка мобильного фитнес-приложения, которое будет предоставлять необходимый функционал для проведения тренировок дома.

**Задачи** курсовой работы, которые необходимо выполнить для достижения поставленной цели следующие:

1. провести анализ предметной области;
2. определить требования к разрабатываемому модулю программного обеспечения (ПО);
3. составить техническое задание (ТЗ) на разработку модуля ПО;
4. обосновать выбор конкретных программных средств реализации;
5. разработать структуру модуля ПО;
6. спроектировать логику взаимодействия модуля с пользователем;
7. описать используемые алгоритмы и структуры данных;
8. реализовать программный модуль;
9. выполнить тестирование разработанного модуля ПО;
10. выполнить установку модуля и его внедрение в разрабатываемое ПО.

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Описание предметной области

В данной предметной области уже существует довольно много решений. Однако в виду того, что почти все они имеют схожую структуру, в данной главе будут рассмотрены наиболее известные из них.

Первое приложение – это «ПРО Фитнес», в котором собраны упражнения и тренировки, а также журнал тренировок (рисунок 1).

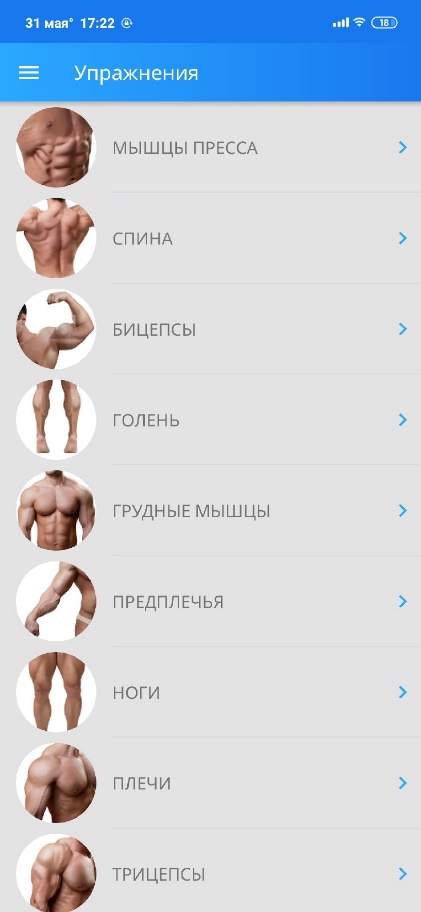


Рисунок 1 – приложение «ПРО Фитнес»

В результате анализа были выявлены следующие достоинства приложения:

* большая база различных тренировок и упражнений на все группы мышц;
* наличие таймеров во время тренировок, которые помогают контролировать время выполнения упражнения;
* видео-представления техники выполнения упражнений.

Среди недостатков приложения можно выделить следующее:

* отсутствие разбиения упражнений по типам тренировок (кардио, силовые и т.д.);
* непродуманный макет интерфейса в режиме тренировки;
* непрезентабельный и несоответствующий современным представлениям дизайн;
* неудачно выбран размер таймера, его трудно заметить во время выполнения упражнения.

Следующее приложение – это «FitProSport», которое включает множество полезных функций помимо упражнения и тренировок, например, отслеживание результатов (рисунок 2).

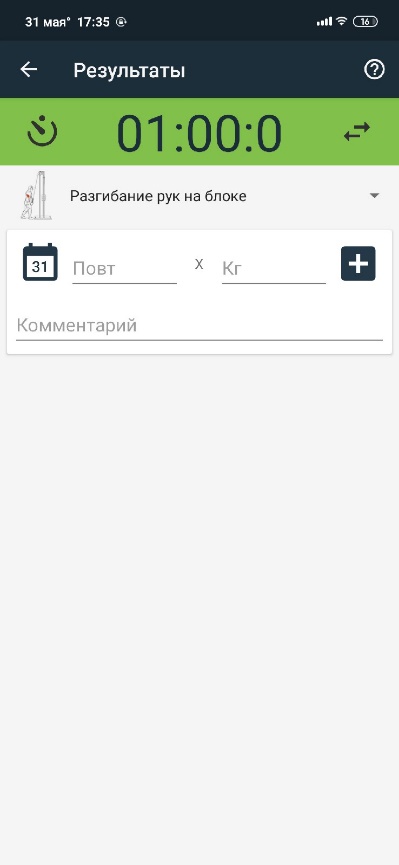
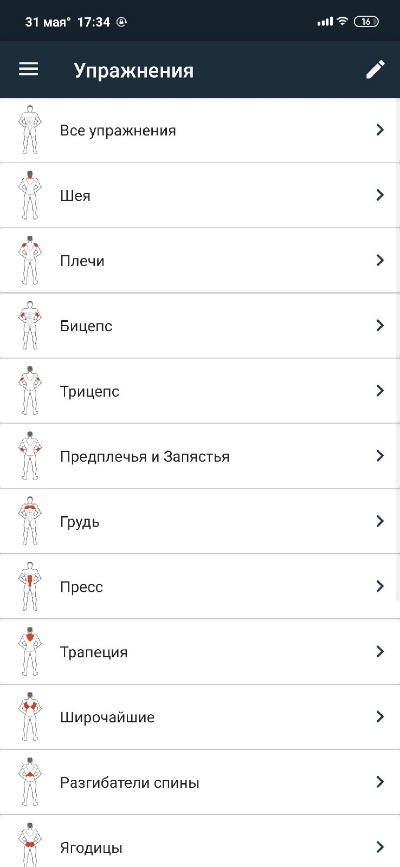


Рисунок 2 – приложение «FitProSport»

Приложение обладает следующими достоинствами:

* огромная база упражнений, как для домашних, так и для профессиональных тренировок в зале;
* возможность самому определять нагрузку и время выполнения упражнения.

Недостатки:

* в приложении отсутствуют готовые программы тренировок;
* узкая направленность (подходит больше для профессионалов). Упражнения в тренировке определяются пользователем, что идеально подходит для профессиональных спортсменов, но может вызывать серьезные трудности у новичков;
* негармоничная цветовая гамма, элементы интерфейса сливаются воедино, что мешает общему восприятию приложения.

Последнее приложение, которое было рассмотрено в ходе анализа – это «Тренировки для дома».

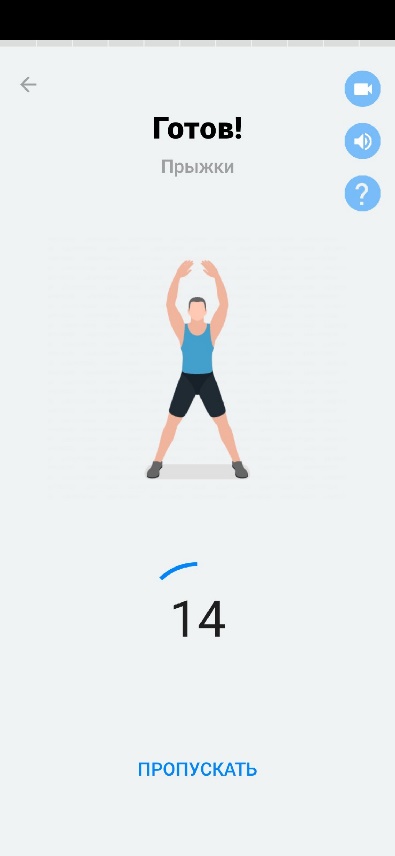
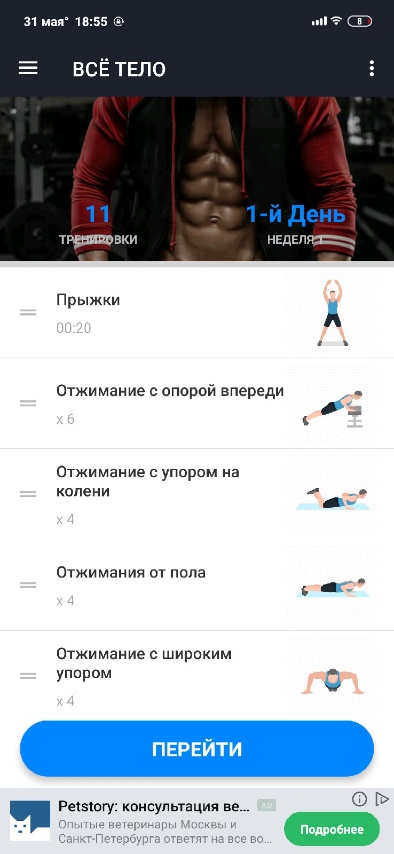
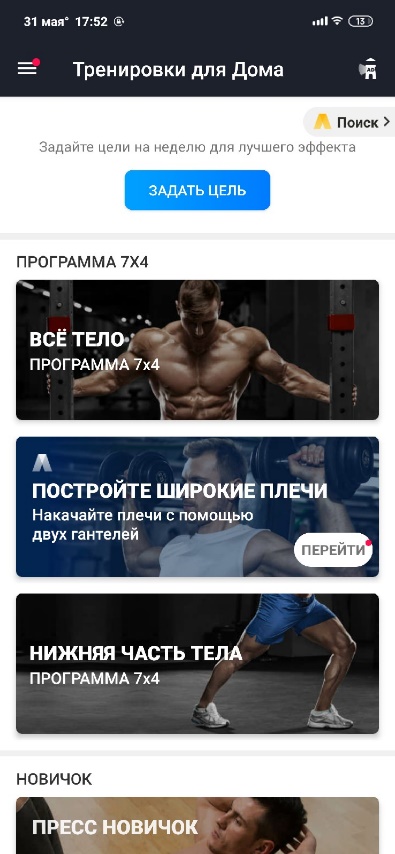


Рисунок 3 – приложение «Тренировки для дома»

Были выявлены следующие достоинства приложения:

* продуманная организация элементов интерфейса, современный стиль иконок;
* наличие огромной базы готовых программ тренировок;
* формирование отчета и истории тренировок;
* возможность подбирать тренировки по уровню сложности.

Но также были выявлены некоторые недостатки:

* узкая направленность (упражнения и программы больше ориентированы на мужчин);
* узкая направленность (доступные программы и тренировки относятся только к категории силовых).

Таким образом, существующие решения не способны в полной мере обеспечить весь необходимый функционал для проведения тренировок дома, что свидетельствует об актуальности данной работы.

В результате проведенного анализа доступных приложений для занятий спортом было принято решение разработать фитнес-приложение, который будет представлять собой совокупность многих из выведенных достоинств (таблица 1).

Таблица 1 - сравнительный анализ особенностей аналогов и разрабатываемого фитнес приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Приложение  Особенности | «ПРО Фитнес» | «FitProSport» | «Тренировки для дома» | Фитнес-приложение |
| Большая база упражнений | + | + | + | + |
| Таймер | + | - | - | + |
| Выбор нагрузки | - | + | + | + |
| Формирование отчета | - | - | + | - |
| Разнообразие типов тренировок | - | - | + | + |
| Ориентация на широкий круг пользователей | + | - | - | + |
| Продуманная организация интерфейса | - | + | + | + |
| Современный дизайн | - | - | + | + |
| Видео-представление техники выполнения | + | - | - | - |

## Определение требований к программному обеспечению

При разработке мобильного приложениянеобходимо учитывать, что в дальнейшем оно будетиспользоваться по своему прямому назначению, а такжевозможно будет дополнено рядом функций и новыхвозможностей касательно логики и интерфейса.

Исходя из исходных требований проектирования мобильного приложения, сначала были сформулированы основные задачи и основные требования к приложению.

Общие требования ко всему мобильному приложению:

* предоставление пользователю полный доступ ко всем функциям приложения;
* реализация системы всплывающих окон (уведомлений);
* полная совместимость со всеми версиями Android, начиная с 6.0.;
* возможность выбора типа тренировки (кардио, растяжка, йога, силовая);
* предоставление описания техники выполнения упражнения;
* возможность изменения времени выполнения упражнения;
* интерфейс, выполненный в стиле Material-UI.

## . Разработка технического задания, выбор программных средств реализации

При проектировании приложения было составлено техническое задание со следующей структурой:

* наименование системы;
* назначение и цели создания системы;
* требования к функционалу;
* требования к особенностям программы;
* требования к аппаратной части.

Полное наименование системы: тренировки дома и в зале.

Назначение: фитнес-приложение для проведения домашних тренировок и тренировок в зале по всем основным направлениям.

Целью создания системы является разработка приложения для того, чтобы сделать удобнее проведение домашних тренировок и тренировок в зале, предоставить возможность новичкам работать по определенно сформированным программам, а профессионалам – подстраивать программы под себя.

Система должна обладать следующим функционалом:

* просмотр основных категорий тренировок, таких как кардио, растяжка, силовые, йога;
* увеличение и уменьшение нагрузки по определенным упражнениям;
* выполнение упражнений временными интервалами;
* отображение оставшегося времени выполнения упражнения на экране;
* возможность пропускать упражнения во время тренировки.

Система должна обладать следующими особенностями:

* система запоминает последние изменения нагрузки по упражнениям.

Требования к аппаратной части:

* приложение должно корректно выполняться на различных устройствах на базе Android версии 6.0 и выше. Вне зависимости от размеров и разрешения экрана интерфейс должен отображаться корректно и должны быть доступны все функциональные возможности.

Перейдем к выбору программных средств реализации. В качестве интегрированной среды разработки выбрана Android Studio. Особенности разработки в выбранной IDE:

* расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана;
* сборка приложений, основанная на Gradle;
* различные виды сборок и генерация нескольких apk. файлов;
* шаблоны основных макетов и компонентов Android [1].

В программе должна храниться вся необходимая информация о тренировках. Существует несколько способов хранения данных в Android-приложениях:

* Хранение с помощью обычных файлов. Это могут быть внутренние файлы или внешние (расположенные на SD-карте).
* Использование файла настроек (preferences). Значения сохраняются в виде пары <имя, значение>. Но, несмотря на простоту, данный способ достаточно ограничен и неудобен для хранения большого количества структурированных данных.
* Использование базы данных SQLite. Каждое приложение для Android может создавать и использовать базы данных SQLite для хранения больших объемов структурированных данных. Данный инструментарий поддерживает стандартные возможности SQL (синтаксис, транзакцииидр.) и способен эффективно использовать память SD-карты [2].

Поскольку использование БД в данной работе является наиболее логичным и удобным решением, выбор был сделан в пользу SQLite.

## Разработка структуры программного обеспечения

Для разработки данного мобильного приложения необходимо разработать структуру программного обеспечения. Особенности программной реализации данного приложения во многом зависит от правильного взаимодействия компонентов c общими элементами интерфейса и логики приложения. Структура проекта может быть представлена в виде диаграммы компонентов, представленной на рисунке 4.

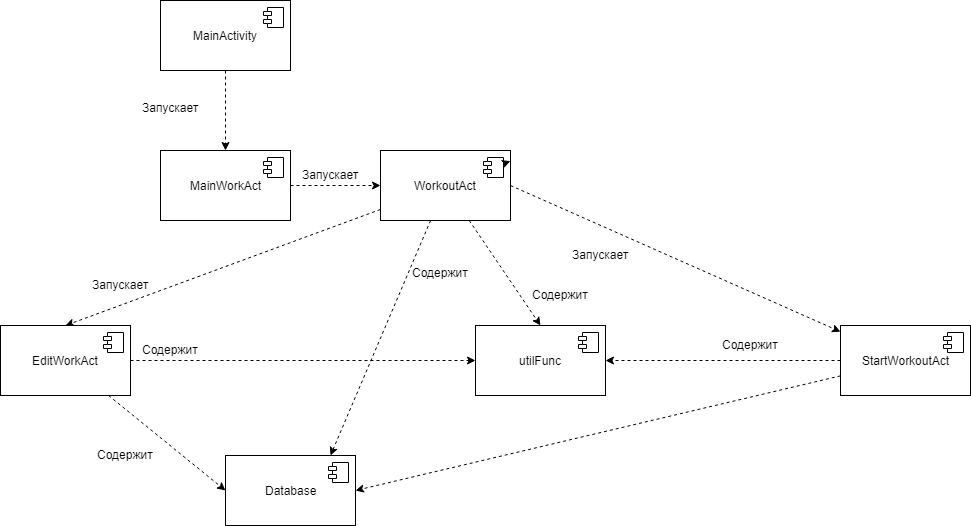


Рисунок 4 – Диаграмма компонентов

Блок хранения данных представляет собой БД.

«MainActivity» *–* основной класс, экземпляр которого создается при запуске приложения. Данный класс ответственен за представление стартового окна приложения с приглашением начать тренировку.

«Database» *–* класс для работы с БД приложения, содержит основные методы создания и обновления БД, необходимые в приложении.

«utilFunc» –класс, который содержит в себе метод склонения числительных (например, «минута», «минут») для корректного отображение времени в приложении.

«MainWorkAct» *–* класс, который ответственен за представление каталога доступных для пользователя программ тренировок.

«WorkoutAct» –класс, который принимает информацию от класса «MainWorkAct» о типе выбранной тренировки и предоставляет пользователю список упражнений по выбранной программе. Для получения списка осуществляется запрос к БД с помощью экземпляра класса «Database». Благодаря методу из класса «utilFunc» корректно отображается время, выделенное на каждое из упражнений. Кроме того, из данного класса мы можем перейти либо к редактированию нагрузки, либо сразу приступить к тренировке.

«EditWorkAct» –класс, который принимает информацию от класса «WorkoutAct» о типе выбранной тренировки и предоставляет пользователю список упражнений по выбранной программе с возможность редактирования нагрузки. Для получения списка и сохранения изменений осуществляется запрос к БД с помощью экземпляра класса «Database». Благодаря методу из класса «utilFunc» корректно отображается время, выделенное на каждое из упражнений.

«StartWorkoutAct» –класс, который принимает информацию от класса «EditWorkAct» о типе выбранной тренировки и предоставляет интерфейс упражнения из тренировки, в котором содержится информация о технике выполнения упражнения, а также таймер. Для получения информации об упражнении осуществляется запрос к БД с помощью экземпляра класса «Database».

Для хранения данных тренировок была разработана база данных, схема которой представлена на рисунке 5.

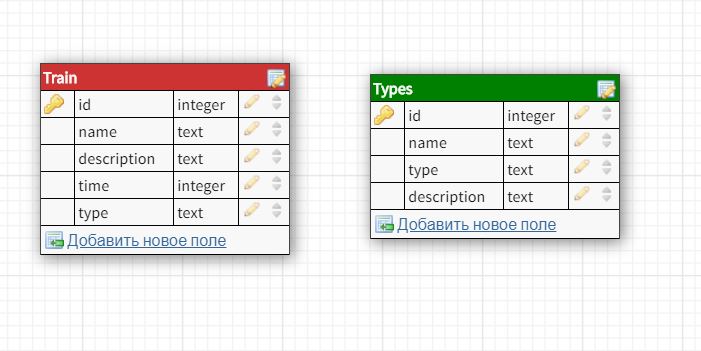


Рисунок 5 – Схема БД

Рассмотрим базу данных более подробно. Для хранения информации о категориях тренировок была разработана таблица «Types» со следующими полями:

* id – идентификатор;
* name – полное название категории;
* type – тип тренировки;
* description – описание программы по данной категории.

Для хранения информации о доступных упражнениях была разработана таблица «Train» со следующими полями:

* id – идентификатор;
* name – полное название упражнения;
* type – тип тренировки;
* time – время выполнения упражнения;
* description – описание техники выполнения упражнения.

## Проектирование логики взаимодействия с пользователем

При запуске приложения предоставляется стартовое окно с приглашением начать тренировку. Пользователь выбирает одну из доступных ему программ тренировок и переходит в окно со списком всех доступных упражнений в соответствии с выбором. В данном окне содержится информацию не только о самих упражнениях, но и о размере нагрузке, выделенной на каждое из них. В том случае, если пользователя не устраивают эти параметры, имеет возможность перейти в режим изменения нагрузки. В данном режиме пользователь редактирует время выполнения каждого упражнения, подстраивая нагрузку под себя. При сохранении изменений, он возвращается в окно со списком упражнений для продолжения работы. После этого пользователь начинает свою тренировку и попадает в окно упражнения, где содержится информация о технике выполнения упражнения, а также запускается таймер выполнения, по окончании которого выводится предложение перейти к следующему упражнению. По окончании тренировки пользователь переходит в окно со всеми доступными программами.

Для приведенного выше описания логики взаимодействия с пользователем была разработана диаграмма вариантов использования, приведенная на рисунке 6.

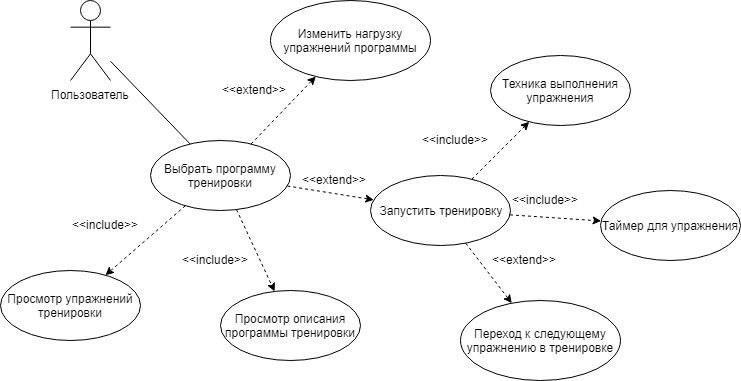


Рисунок 6 – Диаграмма вариантов использования логики взаимодействия с пользователем

## ПРАКТИЧЕСКАЯ ЧАСТЬ

## Описание алгоритмов и структур данных

Т.к. в качестве среды разработки выбрана Android Studio, то реализация будет производиться с помощью средств языка программирования Java, который является официальным языком выбранной среды разработки.

Для хранения последовательностей однотипных данных выбор сделан в пользу массива, т.к. они обеспечивают максимальную производительность за счет индексной адресации [3].

Библиотеки Android имеют набор классов для работы с XML-документами с произвольной структурой и содержанием, поэтому для хранения начальных данных для БД целесообразно использовать именно этот тип файла.

Ключевым компонентом для создания визуального интерфейса в приложении Android является Activity – базовый класс Android-приложений. С помощью activities происходит взаимодействие с пользователем. Все объекты activity, которые есть в приложении, управляются системой в виде стека activity. При запуске новой activity она помещается поверх стека и выводится на экран устройства, пока не появится новая activity. Когда текущая activity заканчивает свою работу (например, пользователь уходит из приложения), то она удаляется из стека, и возобновляет работу та activity, которая ранее была второй в стеке [4].

Предметная область подразумевает работу с большими объемами данных, поэтому для стабильной работы приложения необходимо оптимизировать процесс обработки данных. Для оптимизации запросов в БД SQLite будут использованы следующие стандартные методы:

* использование простых конструкций в операторе where, что ускоряет выполнение запросов в 5-7 раз;
* последовательная работа Activity с БД с целью исключения создания двух независимых соединений на запись;
* открытие БД для записи в режиме редактирования нагрузки, во всех остальных случаях – только для чтения;
* хранение начальных данных в отдельном файле с возможностью восстановления при повреждении БД;
* очистка памяти после совершения запросов [5].

## Реализация программных модулей

Первым делом перед запуском система должна проверить файл под названием AndroidManifest.xml, данные файл включает в себя важную информацию о самом приложении, которая непосредственно требуется система Android. После получения данной информации, если все требования манифест файла выполнены, система сможет начать выполнить какой-либо код мобильного приложения [6]. Данный файл cоздает имя пакета для мобильного приложения, в данном приложении код имеет следующий вид:

|  |
| --- |
|  |
| <manifest xmlns:android="http://schemas.android.com/apk/res/android" | |
| package="com.example.myapplication">  … … …  </manifest> |  |
|  |  |

Так же в файле манифеста необходимо объявлять активность (<activity>):

|  |
| --- |
|  |
|  | <activity android:name=".MainWorkAct"></activity> |
|  | <activity android:name=".EditWorkAct" /> |
|  | <activity android:name=".StartWorkoutAct" /> |
|  | <activity android:name=".WorkoutAct" /> |
|  | <activity android:name=".MainActivity"> |
|  | <intent-filter> |
|  | <action android:name="android.intent.action.MAIN" /> |
|  |  |
|  | <category android:name="android.intent.category.LAUNCHER" /> |
|  | </intent-filter> |
|  | </activity> |

Элемент Activity один из основных элементов приложения, он содержит в себе множество атрибутов, определяющих разрешения, ориентацию экрана и т.д. Обязательным является реализация метода создания *onCreate().* Наиболее важно именно в этом методе вызватьфункцию *setContentView()* для определения макета пользовательского интерфейса операции. Ниже представлен метод создания **MainActivity** – стартового окна приложения. Помимо перечисленного, в данном методе импортируются элементы интерфейса и к ним применяются стили и анимации.

|  |
| --- |
|  |
| protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState); |
| setContentView(R.layout.activity\_main); |
| DataBase dbHelp = new DataBase(this); |
| SQLiteDatabase db = dbHelp.getWritableDatabase(); |
| dbHelp.close(); |
| db.close(); |
| animimgpage = AnimationUtils.loadAnimation(this, R.anim.animimgpage); |
| btnone = AnimationUtils.loadAnimation(this, R.anim.btnone); |
| btntwo = AnimationUtils.loadAnimation(this, R.anim.btntwo); |
| btnthree = AnimationUtils.loadAnimation(this, R.anim.btnthree); |
| ltr = AnimationUtils.loadAnimation(this, R.anim.ltr); |
|  |
| Typeface MLight = Typeface.createFromAsset(getAssets(), "fonts/MLight.ttf"); |
| Typeface MMedium = Typeface.createFromAsset(getAssets(), "fonts/MMedium.ttf"); |
| Typeface Vidaloka = Typeface.createFromAsset(getAssets(), "fonts/Vidaloka.ttf"); |
|  |
| titlepage = (TextView) findViewById(R.id.titlepage); |
| subtitlepage =(TextView) findViewById(R.id.subtitlepage); |
| btnexercise = (TextView) findViewById(R.id.btnexercise); |
| imgpage = (ImageView) findViewById(R.id.imgpage); |
| bgprogres = (View) findViewById(R.id.bgprogress); |
| bgprogresstop = (View)findViewById(R.id.bgprogresstop); |
| titlepage.setTypeface(Vidaloka); |
| subtitlepage.setTypeface(MLight); |
| btnexercise.setTypeface(MMedium); |
|  |
| imgpage.startAnimation(animimgpage); |
| titlepage.startAnimation(btnone); |
| subtitlepage.startAnimation(btnone); |
| btnexercise.startAnimation(btnthree); |
| bgprogres.startAnimation(btntwo); |
| bgprogresstop.startAnimation(ltr); |
| } |
| } |

Для запуска другого activity достаточно вызвать метод *startActivity()*, передав в него объект Intent, который описывает запускаемую операцию. В намерении указывается activity, которое необходимо запустить [4]. Намерение также может содержать небольшой объем данных, которые будут использоваться запущенным activity. Данные передаются с помощью метода *putExtra()*. Ниже представлен пример запуска **EditWorkAct** из **WorkoutAct**. Среди параметров передается информация о типе тренировки.

|  |
| --- |
|  |
| btnchange.setOnClickListener(new View.OnClickListener() {  @Override | |
| public void onClick(View v) { | |
| Intent a = new Intent(WorkoutAct.this,EditWorkAct.class); | |
| a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION); | |
| a.putExtra("type", type); | |
| startActivity(a); | |
| } | |
| }); | |

По аналогичному принципу работают все Activity в приложении:

* MainActivity – стартовое окно приложение с предложением приступить к тренировке;
* MainWorkAct – окно с выборов категории тренировок;
* WorkoutAct – окно со списоком доступных упражнений в соответствии с категорией тренировки;
* EditWorkAct – окно редактирования нагрузки упражнений;
* StartWorkoutAct – окно выполнения упражнения.

Одно из основных составляющих мобильного приложения, являются ресурсы. Они хранятся в папке res, там находятся различные ресурсы, layout файлы, файла с картинками. Обратить внимание требует на себя подпапка values, которая предназначена для хранения ресурсов (констант) причем различных типов [6]. В случае данного приложения в подпапке присутствуют файлы следующих ресурсов:

* сolors.xml;
* strings.xml;
* styles.xml.

Ресурсы типа – colors.xml представляют собой основной набор цветов, используемых в приложении, это цвет текста, цвет ошибок, цвет выделений и т.д.:

|  |
| --- |
|  |
| <?xml version="1.0" encoding="utf-8"?>  <resources> | |
| <color name="colorPrimary">#008577</color> | |
| <color name="colorPrimaryDark">#0D122C</color> | |
| <color name="colorAccent">#D81B60</color> | |
| </resources> | |

Одним из основных файлов ресурсов является файл с именем strings.xml, который включает в себя названия всех переменных. Файл содержит начальные данные, необходимые для создания БД.

В библиотеке Android есть класс **SQLiteOpenHelper** для создания базы данных. Класс **SQLiteOpenHelper** содержит два абстрактных метода [7]:

* *onCreate()* — вызывается при первом создании базы данных;
* *onUpgrade()* — вызывается при модификации базы данных.

Для его использования, в программе создал класс Database, расширенный из класса SQLiteOpenHelper. Код класса представлен ниже. Для начальной инициализации БД импортируем массивы данных из ресурсов с помощью функций *getResources()* и *getStringArray().*

|  |
| --- |
|  |
| public class DataBase extends SQLiteOpenHelper {  private final Context fContext; |
| public DataBase(Context context) |
| { |
| super(context, "db", null, 1); |
| fContext = context; |
| } |
| @Override |
| public void onCreate(SQLiteDatabase db) { |
| db.execSQL("create table train (id INTEGER, name TEXT, description TEXT, time INTEGER, img TEXT, type TEXT);"); |
| ContentValues values = new ContentValues(); |
| Resources res = fContext.getResources(); |
| String[] names = res.getStringArray(R.array.name); |
| String[] imgs = res.getStringArray(R.array.img); |
| String[] desc = res.getStringArray(R.array.desc); |
| int length = names.length; |
| for (int i = 0; i < length; i++) { |
| String type = imgs[i].substring(0, imgs[i].length() - 1); |
| values.put("id", i); |
| values.put("name", names[i]); |
| values.put("description", desc[i]); |
| values.put("time", 1); |
| values.put("img", imgs[i]); |
| values.put("type", type); |
| db.insert("train", null, values); |
| } |
| db.execSQL("create table types (id INTEGER, type TEXT, name TEXT, description TEXT);"); |
| ContentValues values1 = new ContentValues(); |
| String[] types = res.getStringArray(R.array.type); |
| String[] types\_name = res.getStringArray(R.array.type\_name); |
| String[] types\_desc = res.getStringArray(R.array.type\_desc); |
| length = types.length; |
| for (int i = 0; i < length; i++) { |
| values1.put("id", i); |
| values1.put("type", types[i]); |
| values1.put("name", types\_name[i]); |
| values1.put("description", types\_desc[i]); |
| db.insert("types", null, values1); |
| } |
| } |
| @Override |
| public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {} |
|  |
| } |

## Тестирование программных модулей

В процессе разработки приложения производилось поэтапное тестирование с целью выявления программных ошибок и несоответствий ТЗ (техническому заданию). Для этого были созданы эмуляторы смартфона с разными диагоналями экрана для разных версий Android, начиная с 6.0. Тестируемый программный продукт последовательно запускался на этих эмуляторах с целью выявления особенностей работы приложения, запущенного на разных операционных системах.

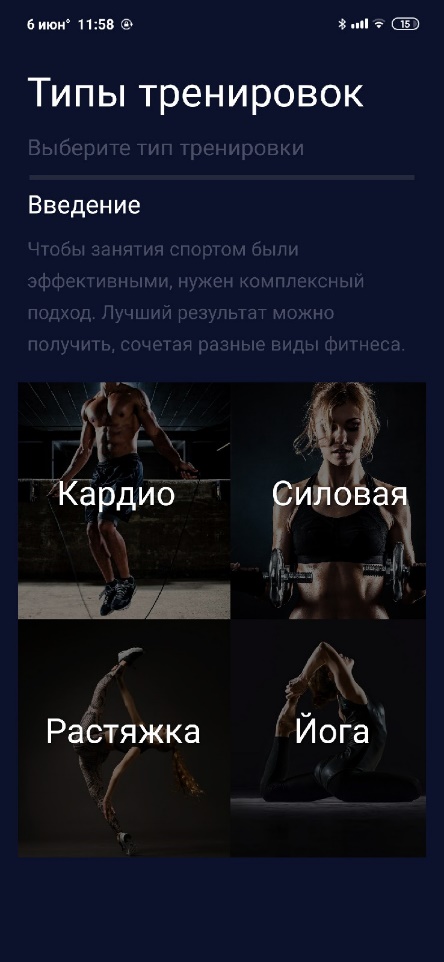
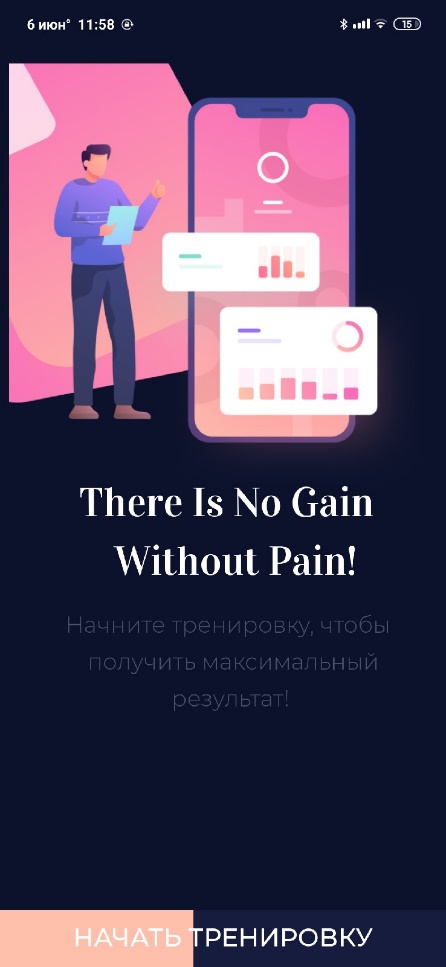
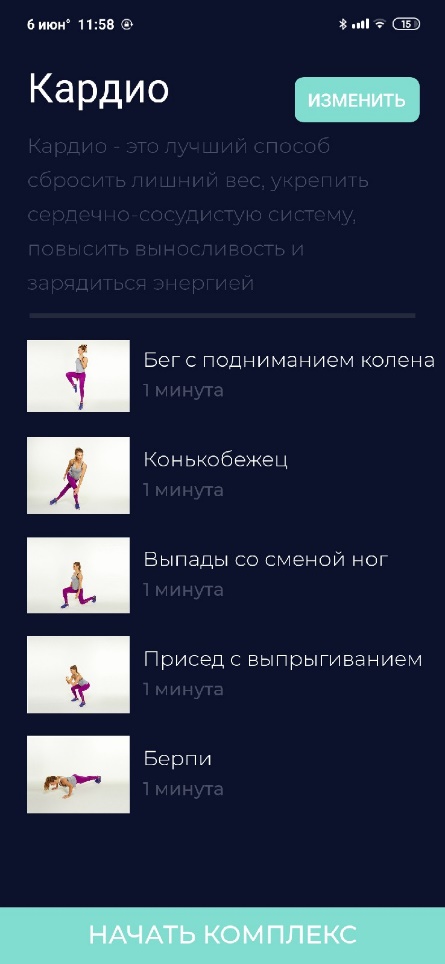
В результате проведенного тестирования аномального поведения компонентов не выявлено и можно сделать вывод, что элементы на разных версиях устройств отображаются корректно, переходы между активностями работают без торможений. Примеры отображения окон представлены на рисунке 7. Тестирование выполнялось на устройстве Xiaomi Mi 9 с ОС Android 9.0 Pie и разрешением экрана 2340х1080.

Рисунок 7 – различные окна фитнес-приложения

Для тестирования отдельных модулей работы c базой данных в текст программы были внесены специальные функции, позволяющие анализировать базу данных и, при подозрении на ошибку, выводящие сообщение в системный журнал. Они также известны как unit-тесты.

Ниже представлен исходный код unit-теста для проверки создания БД с использованием библиотеки Junit [8].

|  |
| --- |
| **public class DBTestinng extends** AndroidTestCase**{**  **@Test** |
| **public** void testCreateDB(){ |
| Database dbHelper = **new** Database(mContext); |
| SQLiteDatabase db = db.getWritableDatabase(); |
| allertTrue(db.isOpen());  }  } |

Приведенный тест выполняется успешно, о чем свидетельствует сообщение в Android Studio.

## Установка и использование программного обеспечения

Т.к. приложение реализовано с помощью программных средств Android Studio, то для сборки APK воспользуемся встроенным мастером. Для того, чтобы начать процесс подписания APK необходимо пройти шаги, описанные ниже.

1. Открыть окно мастера (Build->Generate Signed APK);

2. Создать Keystore. Для этого необходимо заполнить форму всеми необходимыми данными и нажать на кнопку «ОК»;

3. Заполнить данными из прошлого шага Generate Signed APK и нажать кнопку «NEXT»;

4. Ввести мастер-пароль для дальнейшего использования приложения и нажать кнопку «ОК»;

5. Выбрать Build Type тип «release» и нажать кнопку «Finish»;

6. Подождать завершения создания APK, после чего в папке появится готовый файл сборки.

## ЗАКЛЮЧЕНИЕ

На основании того, что предметная область курсового проекта определена и исследована, требования к разрабатываемому модулю ПО определены, техническое задание на разработку модуля ПО составлено, программные средства реализации изучены, выбраны и выбор обоснован, структура разрабатываемого модуля ПО и его логика взаимодействия с пользователем спроектированы, используемые алгоритмы и структуры данных описаны, программный модуль реализован и выполнено его тестирование, а также внедрение в проект, поставленную цель курсовой работы можно считать достигнутой.

Исходя из того, что все поставленные на курсовой проект задачи были выполнены и мобильное фитнес-приложение реализовано, курсовой проект можно считать выполненным.

*.*

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Свободная энциклопедия Википедия, статья «Android Studio» [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Android\_Studio, свободный. Дата обращения: 31.05.2019
2. Сайт TUTSPLUS, статья «Android с нуля: как хранить данные приложения локально» [Электронный ресурс]. – Режим доступа: https://code.tutsplus.com/ru/tutorials/android-from-scratch-how-to-store-application-data-locally--cms-26853, свободный. Дата обращения: 31.05.2019
3. Сайт Хабр, статья «Алгоритмы и структуры данных JDK» [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/182776/, свободный. Дата обращения: 27.05.2019
4. Официальный сайт разработчиков Android, статья «Операции» [Электронный ресурс]. – Режим доступа: <https://developer.android.com/guide/components/activities.html?hl=RU>, свободный. Дата обращения: 17.05.2019
5. Сайт Enterra, статья «Android. Особенности работы с SQLite» [Электронный ресурс]. – Режим доступа: http://www.enterra.ru/blog/android\_issues\_with\_sqlite/, свободный. Дата обращения: 16.05.2019
6. Сайт ИНТУИТ, статья «Структура Android приложения» [Электронный ресурс]. – Режим доступа: https://www.intuit.ru/studies/courses/12643/1191/lecture/21983?page=5, свободный. Дата обращения: 01.06.2019
7. Сайт STARTANDROID, статья «Хранение данных. SQLite» [Электронный ресурс]. – Режим доступа: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/74-urok-34-hranenie-dannyh-sqlite.html, свободный. Дата обращения: 03.06.2019
8. Сайт JAVASTUDY, статья «Junit-введение в юнит тесты» [Электронный ресурс]. – Режим доступа: http://javastudy.ru/junit/junit-hello-world/, свободный. Дата обращения: 05.06.2019

## ПРИЛОЖЕНИЕ А

**(обязательное)**

## ЛИСТИНГ КОДА ОСНОВНЫХ ФУНКЦИОНАЛЬНЫХ МОДУЛЕЙ

DataBase.java

package com.example.myapplication;

import android.content.ContentValues;

import android.content.Context;

import android.content.res.Resources;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DataBase extends SQLiteOpenHelper {

private final Context fContext;

public DataBase(Context context)

{

super(context, "db", null, 1);

fContext = context;

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table train (id INTEGER, name TEXT, description TEXT, time INTEGER, img TEXT, type TEXT);");

ContentValues values = new ContentValues();

Resources res = fContext.getResources();

String[] names = res.getStringArray(R.array.name);

String[] imgs = res.getStringArray(R.array.img);

String[] desc = res.getStringArray(R.array.desc);

int length = names.length;

for (int i = 0; i < length; i++) {

String type = imgs[i].substring(0, imgs[i].length() - 1);

values.put("id", i);

values.put("name", names[i]);

values.put("description", desc[i]);

values.put("time", 1);

values.put("img", imgs[i]);

values.put("type", type);

db.insert("train", null, values);

}

db.execSQL("create table types (id INTEGER, type TEXT, name TEXT, description TEXT);");

ContentValues values1 = new ContentValues();

String[] types = res.getStringArray(R.array.type);

String[] types\_name = res.getStringArray(R.array.type\_name);

String[] types\_desc = res.getStringArray(R.array.type\_desc);

length = types.length;

for (int i = 0; i < length; i++) {

values1.put("id", i);

values1.put("type", types[i]);

values1.put("name", types\_name[i]);

values1.put("description", types\_desc[i]);

db.insert("types", null, values1);

}

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

EditWorkAct.java

package com.example.myapplication;

import android.content.ContentValues;

import android.content.Context;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.graphics.Typeface;

import android.view.animation.AnimationUtils;

import android.widget.Toast;

public class EditWorkAct extends AppCompatActivity {

TextView titlepage, subtitlepage, btnexercise;

Button btnadd, btnremove, btnadd2, btnremove2, btnadd3, btnremove3,

btnadd4, btnremove4, btnadd5, btnremove5;

View divpage, bgprogress;

LinearLayout fitone, fittwo, fitthree, fitfour, fitfive;

Animation bttone, bttwo, bttfour, bttfive, bttsix, bttseven;

String type;

Context context = this;

int[] img = new int[]{R.id.img1, R.id.img2, R.id.img3, R.id.img4, R.id.img5 };

int[] title = new int[]{R.id.fitonetitle, R.id.fittwotitle, R.id.fitthreetitle,

R.id.fitfourtitle, R.id.fitfivetitle};

int[] workvalue = new int[]{R.id.workvalue, R.id.workvalue2, R.id.workvalue3,

R.id.workvalue4, R.id.workvalue5};

int [] sumworkout = new int[] {0, 0, 0, 0, 0};

int [] id = new int[] {0, 0, 0, 0, 0};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_work);

type = getIntent().getStringExtra("type");

bttone = AnimationUtils.loadAnimation(this, R.anim.btnone);

bttwo = AnimationUtils.loadAnimation(this, R.anim.btntwo);

bttfour = AnimationUtils.loadAnimation(this, R.anim.btnfour);

bttfive = AnimationUtils.loadAnimation(this, R.anim.btnfive);

bttsix = AnimationUtils.loadAnimation(this, R.anim.btnsix);

bttseven = AnimationUtils.loadAnimation(this, R.anim.btnseven);

titlepage = findViewById(R.id.titlepage);

subtitlepage = findViewById(R.id.subtitlepage);

fitone = findViewById(R.id.fitone);

fittwo = findViewById(R.id.fittwo);

fitthree = findViewById(R.id.fitthree);

fitfour = findViewById(R.id.fitfour);

fitfive = findViewById(R.id.fitfive);

divpage = findViewById(R.id.divpage);

bgprogress = findViewById(R.id.bgprogress);

titlepage = (TextView) findViewById(R.id.titlepage);

subtitlepage =(TextView) findViewById(R.id.subtitlepage);

btnexercise = (TextView) findViewById(R.id.btnexercise);

divpage = (View) findViewById(R.id.divpage);

btnadd = findViewById(R.id.btnadd);

btnremove = findViewById(R.id.btnremove);

btnadd2 = findViewById(R.id.btnadd2);

btnremove2 = findViewById(R.id.btnremove2);

btnadd3 = findViewById(R.id.btnadd3);

btnremove3 = findViewById(R.id.btnremove3);

btnadd4 = findViewById(R.id.btnadd4);

btnremove4 = findViewById(R.id.btnremove4);

btnadd5 = findViewById(R.id.btnadd5);

btnremove5 = findViewById(R.id.btnremove5);

btnexercise = findViewById(R.id.btnexercise);

Typeface MLight = Typeface.createFromAsset(getAssets(), "fonts/MLight.ttf");

Typeface MMedium = Typeface.createFromAsset(getAssets(), "fonts/MMedium.ttf");

Typeface Vidaloka = Typeface.createFromAsset(getAssets(), "fonts/Vidaloka.ttf");

DataBase dbHelp = new DataBase(this);

SQLiteDatabase db = dbHelp.getWritableDatabase();

String where = "type" + " LIKE '%" + type + "%'";

String[] whereArgs = new String[]{where};

String sqlSelect = "SELECT \* FROM train WHERE type='" + type + "';";

Cursor query = db.rawQuery(sqlSelect, null);

int i = 0;

if (query.moveToFirst()){

do {

TextView myView = (TextView) findViewById(title[i]);

TextView myDesc = (TextView) findViewById(workvalue[i]);

ImageView myImg = (ImageView) findViewById(img[i]);

String name = query.getString(query.getColumnIndex("name"));

String img = query.getString(query.getColumnIndex("img"));

String desc = Integer.toString(query.getInt(query.getColumnIndex("time")));

id[i] = query.getInt(query.getColumnIndex("id"));

sumworkout[i] = Integer.parseInt(desc);

int resID = getResources().getIdentifier(img, "drawable", getPackageName());

myView.setText(name);

myImg.setImageResource(resID);

myDesc.setText(desc);

myView.setTypeface(MLight);

myDesc.setTypeface(MMedium);

i++;

}

while (query.moveToNext());

}

db.close();

btnadd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sumworkout[0]+=1;

TextView myView = (TextView) findViewById(workvalue[0]);

myView.setText(sumworkout[0]+"");

}

});

btnremove.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(sumworkout[0] <= 0){

Toast.makeText(getApplicationContext(), "Ошибка!", Toast.LENGTH\_SHORT).show();

} else {

sumworkout[0]-=1;

TextView myView = (TextView) findViewById(workvalue[0]);

myView.setText(sumworkout[0]+"");

}

}

});

btnadd2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sumworkout[1]+=1;

TextView myView = (TextView) findViewById(workvalue[1]);

myView.setText(sumworkout[1]+"");

}

});

btnremove2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(sumworkout[1] <= 0){

Toast.makeText(getApplicationContext(), "Ошибка!", Toast.LENGTH\_SHORT).show();

} else {

sumworkout[1]-=1;

TextView myView = (TextView) findViewById(workvalue[1]);

myView.setText(sumworkout[1]+"");

}

}

});

btnadd3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sumworkout[2]+=1;

TextView myView = (TextView) findViewById(workvalue[2]);

myView.setText(sumworkout[2]+"");

}

});

btnremove3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(sumworkout[2] <= 0){

Toast.makeText(getApplicationContext(), "Ошибка!", Toast.LENGTH\_SHORT).show();

} else {

sumworkout[2]-=1;

TextView myView = (TextView) findViewById(workvalue[2]);

myView.setText(sumworkout[2]+"");

}

}

});

btnadd4.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sumworkout[3]+=1;

TextView myView = (TextView) findViewById(workvalue[3]);

myView.setText(sumworkout[3]+"");

}

});

btnremove4.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(sumworkout[3] <= 0){

Toast.makeText(getApplicationContext(), "Ошибка!", Toast.LENGTH\_SHORT).show();

} else {

sumworkout[3]-=1;

TextView myView = (TextView) findViewById(workvalue[3]);

myView.setText(sumworkout[3]+"");

}

}

});

btnadd5.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sumworkout[4]+=1;

TextView myView = (TextView) findViewById(workvalue[4]);

myView.setText(sumworkout[4]+"");

}

});

btnremove5.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(sumworkout[4] <= 0){

Toast.makeText(getApplicationContext(), "Ошибка!", Toast.LENGTH\_SHORT).show();

} else {

sumworkout[4]-=1;

TextView myView = (TextView) findViewById(workvalue[4]);

myView.setText(sumworkout[4]+"");

}

}

});

btnexercise.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

DataBase dbHelp = new DataBase(context);

SQLiteDatabase db = dbHelp.getWritableDatabase();

ContentValues args = new ContentValues();

for (int i = 0; i < 5; i++){

String strFilter = "id=" + id[i];

args.put("time", sumworkout[i]);

db.update("train", args, strFilter, null);

}

db.close();

Intent a = new Intent(EditWorkAct.this,WorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", type);

startActivity(a);

}

});

titlepage.startAnimation(bttone);

subtitlepage.startAnimation(bttone);

divpage.startAnimation(bttone);

fitone.startAnimation(bttwo);

fittwo.startAnimation(bttfour);

fitthree.startAnimation(bttfive);

fitfour.startAnimation(bttsix);

fitfive.startAnimation(bttseven);

btnexercise.startAnimation(bttfive);

bgprogress.startAnimation(bttfive);

}

}

MainActivity.java

package com.example.myapplication;

import android.content.Intent;

import android.database.sqlite.SQLiteDatabase;

import android.graphics.Typeface;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.ImageView;

import android.widget.TextView;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity {

TextView titlepage, subtitlepage, btnexercise;

ImageView imgpage;

Animation animimgpage, btnone, btntwo, btnthree, ltr;

View bgprogres, bgprogresstop;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

DataBase dbHelp = new DataBase(this);

SQLiteDatabase db = dbHelp.getWritableDatabase();

dbHelp.close();

db.close();

//Загружаем анимацию

animimgpage = AnimationUtils.loadAnimation(this, R.anim.animimgpage);

btnone = AnimationUtils.loadAnimation(this, R.anim.btnone);

btntwo = AnimationUtils.loadAnimation(this, R.anim.btntwo);

btnthree = AnimationUtils.loadAnimation(this, R.anim.btnthree);

ltr = AnimationUtils.loadAnimation(this, R.anim.ltr);

//Импортируем шрифты

Typeface MLight = Typeface.createFromAsset(getAssets(), "fonts/MLight.ttf");

Typeface MMedium = Typeface.createFromAsset(getAssets(), "fonts/MMedium.ttf");

Typeface Vidaloka = Typeface.createFromAsset(getAssets(), "fonts/Vidaloka.ttf");

//Импортируем элементы

titlepage = (TextView) findViewById(R.id.titlepage);

subtitlepage =(TextView) findViewById(R.id.subtitlepage);

btnexercise = (TextView) findViewById(R.id.btnexercise);

imgpage = (ImageView) findViewById(R.id.imgpage);

bgprogres = (View) findViewById(R.id.bgprogress);

bgprogresstop = (View)findViewById(R.id.bgprogresstop);

//Применяем к titlepage шрифт

titlepage.setTypeface(Vidaloka);

subtitlepage.setTypeface(MLight);

btnexercise.setTypeface(MMedium);

//Экспорт анимации

imgpage.startAnimation(animimgpage);

titlepage.startAnimation(btnone);

subtitlepage.startAnimation(btnone);

btnexercise.startAnimation(btnthree);

bgprogres.startAnimation(btntwo);

bgprogresstop.startAnimation(ltr);

btnexercise.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(MainActivity.this, MainWorkAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

startActivity(a);

}

});

}

}

MainWorkAct.java

package com.example.myapplication;

import android.content.Context;

import android.content.Intent;

import android.database.Cursor;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.database.sqlite.SQLiteDatabase;

import android.widget.LinearLayout;

import android.widget.TextView;

public class MainWorkAct extends AppCompatActivity {

ImageView cardio, power, stretch, yoga;

TextView startpage, titlepage, subtitlepage, starttext, cardiotxt,

powertxt, stretchtxt, yogatxt;

LinearLayout layout1, layout2;

View divpage;

Animation bttone, bttwo, bttfour, bttfive, bttsix, bttseven;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main\_work);

startpage =(TextView) findViewById(R.id.startpage);

titlepage =(TextView) findViewById(R.id.titlepage);

subtitlepage =(TextView) findViewById(R.id.subtitlepage);

starttext =(TextView) findViewById(R.id.starttext);

cardiotxt =(TextView) findViewById(R.id.cardiotxt);

powertxt =(TextView) findViewById(R.id.powertxt);

stretchtxt =(TextView) findViewById(R.id.stretchtxt);

yogatxt =(TextView) findViewById(R.id.yogatxt);

divpage = (View) findViewById(R.id.divpage);

layout1 = (LinearLayout)findViewById(R.id.linearLayout);

layout2 = (LinearLayout)findViewById(R.id.linearLayout1);

bttone = AnimationUtils.loadAnimation(this, R.anim.btnone);

bttwo = AnimationUtils.loadAnimation(this, R.anim.btntwo);

bttfour = AnimationUtils.loadAnimation(this, R.anim.btnfour);

bttfive = AnimationUtils.loadAnimation(this, R.anim.btnfive);

bttsix = AnimationUtils.loadAnimation(this, R.anim.btnsix);

bttseven = AnimationUtils.loadAnimation(this, R.anim.btnseven);

cardio = (ImageView) findViewById(R.id.cardio);

power = (ImageView) findViewById(R.id.power);

stretch = (ImageView) findViewById(R.id.stretch);

yoga = (ImageView) findViewById(R.id.yoga);

titlepage.startAnimation(bttone);

subtitlepage.startAnimation(bttwo);

divpage.startAnimation(bttfour);

startpage.startAnimation(bttfive);

starttext.startAnimation(bttsix);

layout1.startAnimation(bttseven);

layout2.startAnimation(bttseven);

cardiotxt.startAnimation(bttseven);

powertxt.startAnimation(bttseven);

stretchtxt.startAnimation(bttseven);

yogatxt.startAnimation(bttseven);

cardio.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(MainWorkAct.this, WorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", "cardio");

a.putExtra("id", 0);

startActivity(a);

}

});

power.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(MainWorkAct.this, WorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", "power");

a.putExtra("id", 5);

startActivity(a);

}

});

stretch.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(MainWorkAct.this, WorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", "stretch");

a.putExtra("id", 10);

startActivity(a);

}

});

yoga.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(MainWorkAct.this, WorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", "yoga");

a.putExtra("id", 15);

startActivity(a);

}

});

}

}

StartWorkoutAct.java

package com.example.myapplication;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.graphics.Color;

import android.graphics.Typeface;

import android.os.CountDownTimer;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.LinearInterpolator;

import android.view.animation.RotateAnimation;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import java.util.Locale;

import java.util.concurrent.TimeUnit;

public class StartWorkoutAct extends AppCompatActivity {

TextView startpage, subtitlepage, fitonetitle, fitonedesc, timerValue, btnexercise, starttext;

View bgprogress;

ImageView img, img1;

RotateAnimation rotate;

private static final long START\_TIME\_IN\_MILLIS = 4000;

private CountDownTimer countDownTimer;

private boolean mTimerRunning;

private long mTimeLeftInMillis = START\_TIME\_IN\_MILLIS;

String type;

int id, cnt, minute;

boolean flag = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_start\_workout);

type = getIntent().getStringExtra("type");

id = getIntent().getIntExtra("id", 0);

cnt = getIntent().getIntExtra("cnt", 0);

// Импортируем шрифты

Typeface MLight = Typeface.createFromAsset(getAssets(), "fonts/MLight.ttf");

Typeface MMedium = Typeface.createFromAsset(getAssets(), "fonts/MMedium.ttf");

Typeface Vidaloka = Typeface.createFromAsset(getAssets(), "fonts/Vidaloka.ttf");

startpage = (TextView) findViewById(R.id.startpage);

starttext = (TextView) findViewById(R.id.starttext);

subtitlepage = (TextView) findViewById(R.id.subtitlepage);

btnexercise = (TextView) findViewById(R.id.btnexercise);

fitonetitle = (TextView) findViewById(R.id.fitonetitle);

fitonedesc = (TextView) findViewById(R.id.fitonedesc);

timerValue = (TextView) findViewById(R.id.timerValue);

btnexercise = (TextView) findViewById(R.id.btnexercise);

img = (ImageView) findViewById(R.id.imgtimer);

bgprogress = (View) findViewById(R.id.bgprogress);

startpage.setTypeface(Vidaloka);

btnexercise.setTypeface(MMedium);

timerValue.setTypeface(MMedium);

fitonetitle.setTypeface(MMedium);

DataBase dbHelp = new DataBase(this);

SQLiteDatabase db = dbHelp.getWritableDatabase();

String where = "type" + " LIKE '%" + type + "%'";

String[] whereArgs = new String[]{where};

String sqlSelect = "SELECT \* FROM train WHERE id=" + id + ";";

Cursor query = db.rawQuery(sqlSelect, null);

int i = 0;

if (query.moveToFirst()){

do {

String name = query.getString(query.getColumnIndex("name"));

String img\_n = query.getString(query.getColumnIndex("img"));

int time = query.getInt(query.getColumnIndex("time"));

utilFunc func = new utilFunc();

String desc = func.choosePluralMerge(time, "минута", "минуты", "минут");

String description = query.getString(query.getColumnIndex("description"));

ImageView myImg = (ImageView) findViewById(R.id.img);

int resID = getResources().getIdentifier(img\_n, "drawable", getPackageName());

minute = time;

fitonetitle.setText(name);

myImg.setImageResource(resID);

fitonedesc.setText(desc);

starttext.setText(description);

fitonetitle.setTypeface(MLight);

fitonedesc.setTypeface(MMedium);

i++;

}

while (query.moveToNext());

}

db.close();

//fitonetitle.setText(Integer.toString(id));

rotate = new RotateAnimation(0, 360, Animation.RELATIVE\_TO\_SELF, 0.5f,Animation.RELATIVE\_TO\_SELF,0.5f);

rotate.setDuration(5000);

rotate.setRepeatCount((int)(TimeUnit.MINUTES.toMillis(minute)/5000));

rotate.setInterpolator(new LinearInterpolator());

btnexercise.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (!flag)

{

startTimer();

img.startAnimation(rotate);

btnexercise.setText("СЛЕДУЮЩЕЕ УПРАЖНЕНИЕ");

bgprogress.setBackgroundColor(Color.parseColor("#25293E"));

flag = true;

} else

{

id++;

cnt++;

if (cnt == 5)

{

Intent a = new Intent(StartWorkoutAct.this, MainWorkAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

countDownTimer.cancel();

startActivity(a);

}

else

{

Intent a = new Intent(StartWorkoutAct.this,StartWorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", type);

a.putExtra("cnt", cnt);

a.putExtra("id", id);

countDownTimer.cancel();

finish();

startActivity(a);

}

}

}

});

}

private void startTimer(){

countDownTimer = new CountDownTimer(mTimeLeftInMillis, 1000) {

@Override

public void onTick(long millisUntilFinished) {

mTimeLeftInMillis = millisUntilFinished;

updateCountDowntText();

}

@Override

public void onFinish() {

String timeLeft = String.format(Locale.getDefault(), "%02d:%02d", 0, 0);

timerValue.setText(timeLeft);

Toast.makeText(getApplicationContext(), "СТАРТ", Toast.LENGTH\_SHORT).show();

mTimeLeftInMillis = TimeUnit.MINUTES.toMillis(minute) + 1000;

continueTimer();

}

}.start();

mTimerRunning = true;

}

private void continueTimer(){

countDownTimer = new CountDownTimer(mTimeLeftInMillis, 1000) {

@Override

public void onTick(long millisUntilFinished) {

mTimeLeftInMillis = millisUntilFinished;

updateCountDowntText();

}

@Override

public void onFinish() {

String timeLeft = String.format(Locale.getDefault(), "%02d:%02d", 0, 0);

timerValue.setText(timeLeft);

Toast.makeText(getApplicationContext(), "ПОЗДРАВЛЯЕМ!", Toast.LENGTH\_SHORT).show();

bgprogress.setBackgroundColor(Color.parseColor("#FF8892"));

}

}.start();

mTimerRunning = true;

}

private void updateCountDowntText(){

int minutes = (int) (mTimeLeftInMillis / 1000) / 60;

int seconds = (int) (mTimeLeftInMillis / 1000) % 60;

String timeLeft = String.format(Locale.getDefault(), "%02d:%02d", minutes, seconds);

timerValue.setText(timeLeft);

}

}

WorkoutAct.java

package com.example.myapplication;

import android.content.Intent;

import android.content.SharedPreferences;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.graphics.Typeface;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

public class WorkoutAct extends AppCompatActivity {

TextView titlepage, subtitlepage, startpage, starttext, btnexercise,

fitonetitle, fitonedesc, fittwotitle, fittwodesc, fitthreetitle,

fitthreedesc, fitfourtitle, fitfourdesc, fitfivetitle, fitfivedesc;

Animation bttone, bttwo, bttfour, bttfive, bttsix, bttseven;

Button btnchange;

View divpage, bgprogress;

LinearLayout fitone, fittwo, fitthree, fitfour, fitfive;

String type;

int id;

int[] img = new int[]{R.id.img1, R.id.img2, R.id.img3, R.id.img4, R.id.img5 };

int[] title = new int[]{R.id.fitonetitle, R.id.fittwotitle, R.id.fitthreetitle,

R.id.fitfourtitle, R.id.fitfivetitle};

int[] desc = new int[]{R.id.fitonedesc, R.id.fittwodesc, R.id.fitthreedesc,

R.id.fitfourdesc, R.id.fitfivedesc};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_workout);

type = getIntent().getStringExtra("type");

id = getIntent().getIntExtra("id", 0);

//Загружаем анимацию

bttone = AnimationUtils.loadAnimation(this, R.anim.btnone);

bttwo = AnimationUtils.loadAnimation(this, R.anim.btntwo);

bttfour = AnimationUtils.loadAnimation(this, R.anim.btnfour);

bttfive = AnimationUtils.loadAnimation(this, R.anim.btnfive);

bttsix = AnimationUtils.loadAnimation(this, R.anim.btnsix);

bttseven = AnimationUtils.loadAnimation(this, R.anim.btnseven);

//Импортируем шрифты

Typeface MLight = Typeface.createFromAsset(getAssets(), "fonts/MLight.ttf");

Typeface MMedium = Typeface.createFromAsset(getAssets(), "fonts/MMedium.ttf");

Typeface Vidaloka = Typeface.createFromAsset(getAssets(), "fonts/Vidaloka.ttf");

//Импортируем элементы

titlepage = (TextView) findViewById(R.id.titlepage);

subtitlepage =(TextView) findViewById(R.id.subtitlepage);

btnexercise = (TextView) findViewById(R.id.btnexercise);

bgprogress = (View) findViewById(R.id.bgprogress);

divpage = (View) findViewById(R.id.divpage);

btnchange = (Button) findViewById(R.id.btnchange);

fitone = (LinearLayout) findViewById(R.id.fitone);

fittwo = (LinearLayout) findViewById(R.id.fittwo);

fitthree = (LinearLayout) findViewById(R.id.fitthree);

fitfour = (LinearLayout) findViewById(R.id.fitfour);

fitfive = (LinearLayout) findViewById(R.id.fitfive);

DataBase dbHelp = new DataBase(this);

SQLiteDatabase db = dbHelp.getWritableDatabase();

String where = "type" + " LIKE '%" + type + "%'";

String[] whereArgs = new String[]{where};

String sqlSelect = "SELECT \* FROM train WHERE type='" + type + "';";

Cursor query = db.rawQuery(sqlSelect, null);

int i = 0;

if (query.moveToFirst()){

do {

TextView myView = (TextView) findViewById(title[i]);

TextView myDesc = (TextView) findViewById(desc[i]);

ImageView myImg = (ImageView) findViewById(img[i]);

String name = query.getString(query.getColumnIndex("name"));

String img = query.getString(query.getColumnIndex("img"));

int time = query.getInt(query.getColumnIndex("time"));

utilFunc func = new utilFunc();

String desc = func.choosePluralMerge(time, "минута", "минуты", "минут");

int resID = getResources().getIdentifier(img, "drawable", getPackageName());

myView.setText(name);

myImg.setImageResource(resID);

myDesc.setText(desc);

myView.setTypeface(MLight);

myDesc.setTypeface(MMedium);

i++;

}

while (query.moveToNext());

}

sqlSelect = "SELECT \* FROM types WHERE type='" + type + "';";

query = db.rawQuery(sqlSelect, null);

i = 0;

if (query.moveToFirst()){

do {

String type\_name = query.getString(query.getColumnIndex("name"));

String desc = query.getString(query.getColumnIndex("description"));

titlepage.setText(type\_name);

subtitlepage.setText(desc);

i++;

}

while (query.moveToNext());

}

db.close();

//Применяем шрифт

titlepage.setTypeface(Vidaloka);

subtitlepage.setTypeface(MLight);

btnexercise.setTypeface(MMedium);

btnexercise.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(WorkoutAct.this,StartWorkoutAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", type);

a.putExtra("cnt", 0);

a.putExtra("id", id);

startActivity(a);

}

});

btnchange.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent a = new Intent(WorkoutAct.this,EditWorkAct.class);

a.addFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

a.putExtra("type", type);

startActivity(a);

}

});

//применяем анимацию

titlepage.startAnimation(bttone);

btnchange.startAnimation(bttone);

subtitlepage.startAnimation(bttone);

divpage.startAnimation(bttone);

fitone.startAnimation(bttwo);

fittwo.startAnimation(bttfour);

fitthree.startAnimation(bttfive);

fitfour.startAnimation(bttsix);

fitfive.startAnimation(bttseven);

btnexercise.startAnimation(bttfive);

bgprogress.startAnimation(bttfive);

}

}

utilFunc.java

package com.example.myapplication;

import static java.lang.Math.abs;

public class utilFunc {

public static String choosePluralMerge(long number, String caseOne, String caseTwo, String caseFive) {

String str = Long.toString(number) + " ";

number = abs(number);

if (number % 10 == 1 && number % 100 != 11) {

str += caseOne;

} else if (number % 10 >= 2 && number % 10 <= 4 && (number % 100 < 10 || number % 100 >= 20)) {

str += caseTwo;

} else {

str += caseFive;

}

return str;

}

}