МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Особенности управления разработкой проекта мобильного приложения на Android»



Обучающегося 3 курса

Стрижова Егора Дмитриевича

Научный руководитель:

Кандидат физико-математических наук, ассистент

Жуков Николай Николаевич

Санкт-Петербург

2019

Оглавление

[Оглавление 2](#_Toc30078875)

[Введение 3](#_Toc30078876)

[1. теоретические аспекты управления программным проектом по разработки мобильного приложения 4](#_Toc30078877)

[1.1. Выбор языка программирования 4](#_Toc30078878)

[1.2. Сравнительный анализ гибких методологий разработки 5](#_Toc30078879)

[1.3. Техническое задание 8](#_Toc30078880)

[2. реализация проекта 12](#_Toc30078881)

[2.1. Диаграмма Ганта 12](#_Toc30078882)

[2.2. Trello 13](#_Toc30078883)

[2.3. Диаграмма прецедентов 14](#_Toc30078884)

[2.4. Тестирование сборки всех модулей 15](#_Toc30078885)

[Заключение 16](#_Toc30078886)

[СПИСОК ЛИТЕРАТУРЫ 17](#_Toc30078887)

[Приложения 19](#_Toc30078888)

Введение

Актуальность темы исследования**:** обусловлена тем, что в настоящее время android индустрия, сочетающая в себе огромное количество технологий, ставящая разные задачи и требующая различные знания и является одной из самых передовых в IT и умение управлять и ориентироваться в нём очень важно.

Объект исследования: создание продуктов в android-индустрии.

Предмет исследования: управление проектом по разработки android приложения.

Цель исследования: анализ процесса разработки и самого итогового продукта, полученного при помощи Android Studio.

Для достижения указанной цели в процессе выполнения работы были сформулированы следующие задачи:

* исследовать и проанализировать гибкие методологии разработки;
* разработать техническое задание продукта;
* визуализировать план разработки при помощи диаграммы Ганта;
* применить инструменты управления проектом к процессу разработки Android Studio (Trello);
* визуализировать процесс разработки при помощи диаграммы последовательностей;
* провести тестирование приоритетных модулей продукта;

1. теоретические аспекты управления программным проектом по разработки мобильного приложения

1.1. Выбор языка программирования

В ходе анализа информации по теме было выявлено, что наиболее подходящим для разработки приложения под OS Android является язык Java.

Также было выявлено, что одним из лучших IDE для разработки данного приложения является Android Studio.

Для наиболее точного понимания поставленных задач и выявления способов их решения было составлено техническое задание.

1.2. Сравнительный анализ гибких методологий разработки

Таблица 1.1 – Сравнительный анализ гибких методологий разработки

|  |  |  |
| --- | --- | --- |
| Методологии | Преимущества | Недостатки |
| Scrum | Простые практики и артефакты менеджмента | Минимальное руководство во всех дисциплинах кроме менеджмента |
| Самоуправляемая команда, решающая свои проблемы |
| Эволюционный подход к требованиям и разработке, адаптивное поведение |
| Активное участие клиента |
| Сосредоточенность, прозрачность | Нет четко определенных проектных документов |
| Легко сочетается с другими методами |
| Коммуникация, взаимное обучение, общее построение ценностей |
| Укрепление команды на ежедневных митингах |
| ЕР | Много полезных и легко усваиваемых техник | Требуется заказчик в команде |
| Участие заказчика | Зависимость от устного общения |
| Эволюционный подход | Сильная связанность EР практик, они эффективны только в совокупности |
| Кодировщики оценивают трудозатраты, расписание следует этим оценкам | Отсутствие стандартного способа описания дизайна |
| Важность взаимодействия | Некоторые разработчики не любят парного программирования |
| Важность качества | Отсутствие определенных артефактов |
| Уточнение требований к системе на основе приемочных тестов от заказчика | Простой дизайн может приводить к отсутствию архитектурной проработки |
| Четкие метрики |
| Частые ревизии |
| Agile | Гибко отслеживать часто меняющиеся требования к ПО без выхода за рамки бюджета проекта | Может привести к низкому качеству продукта |
| Максимально эффективно вносить изменения и новые идеи заказчика в разрабатываемый продукт |
| В короткие сроки внедрить работающий функционал и оптимизировать его в процессе использования | Риск никогда не достигнуть завершения проекта |
| Создать поле для постоянного взаимодействия команды разработки и пользователей | Могут возникнуть проблемы с расширяемостью продукта |
| Обеспечить постоянный контроль за разработкой и внедрением со стороны заказчика |

На основе таблицы 1 было принято решение воспользоваться методологией SCRUM, так как она обеспечивает максимальную гибкость в разработке, что очень важно при темпах развития технологий, а также обеспечивает постоянный контакт команд, занимающихся разработкой.

1.3. Техническое задание

1 Назначение разработки

Разработка программного проекта приложения на android для тренировок в тренажерном зале, предполагает выполнения комплекса задач, связанных с различными аспектами информационных технологий. Разработка данного продукта предполагает организацию взаимодействий приложения с пользователем, для его мотивации заниматься спортом.

Приведём фрагмент технического задания для описания функционала разрабатываемого приложения.

2. Требования к программе:

2.1. Требования к функциональным характеристикам

Данный проект является актуальным для тренировок

Программа должна обладать следующим функционалом:

1. Модуль с тренировками
2. Модуль с цитатами
3. Модуль для фотографий

2.1.1. Модуль с тренировками

Создание единого макета для дальнейшего добавления видов упражнений. (см. Рис. 1.1, 1.2)

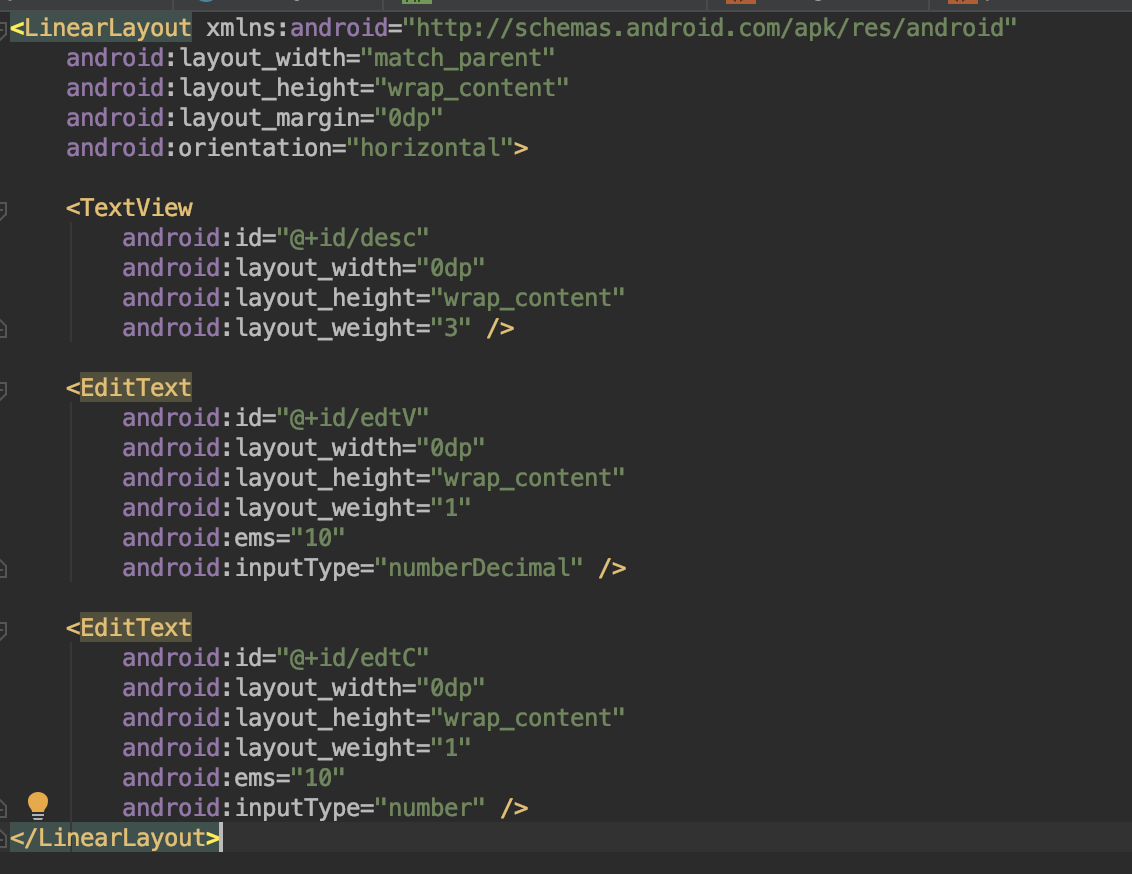


Рисунок 1.1 – Демонстрация макета, по которым создаются упражнения

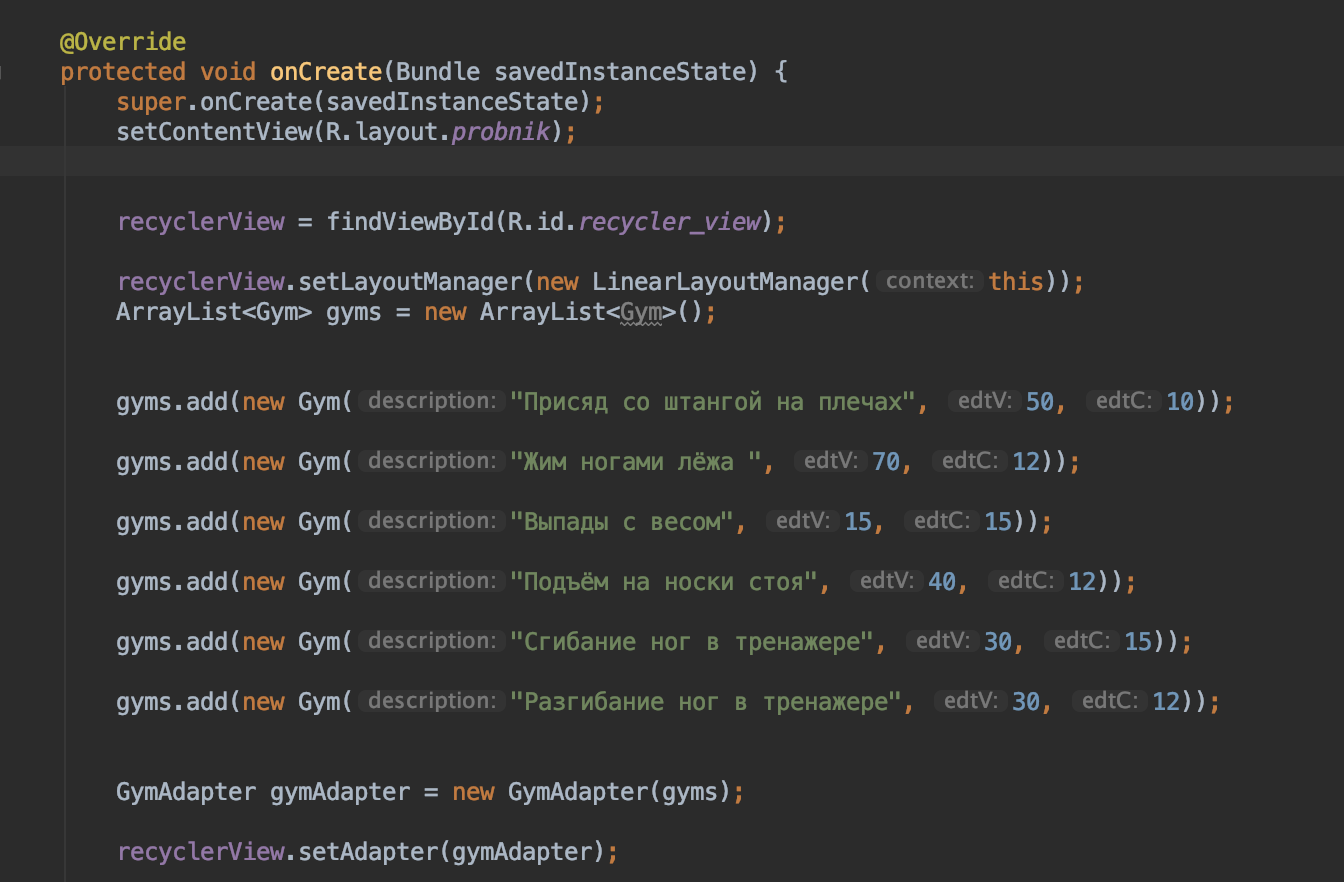


Рисунок 1.2 – Демонстрация реализации единого макета для упражнений

Более подробная информация показана в приложение 1

2.1.2 Модуль с цитатами

2.1.3 Модуль для фотографий

Реализация кнопок для подключения камеры. (см. Рисунок 2)

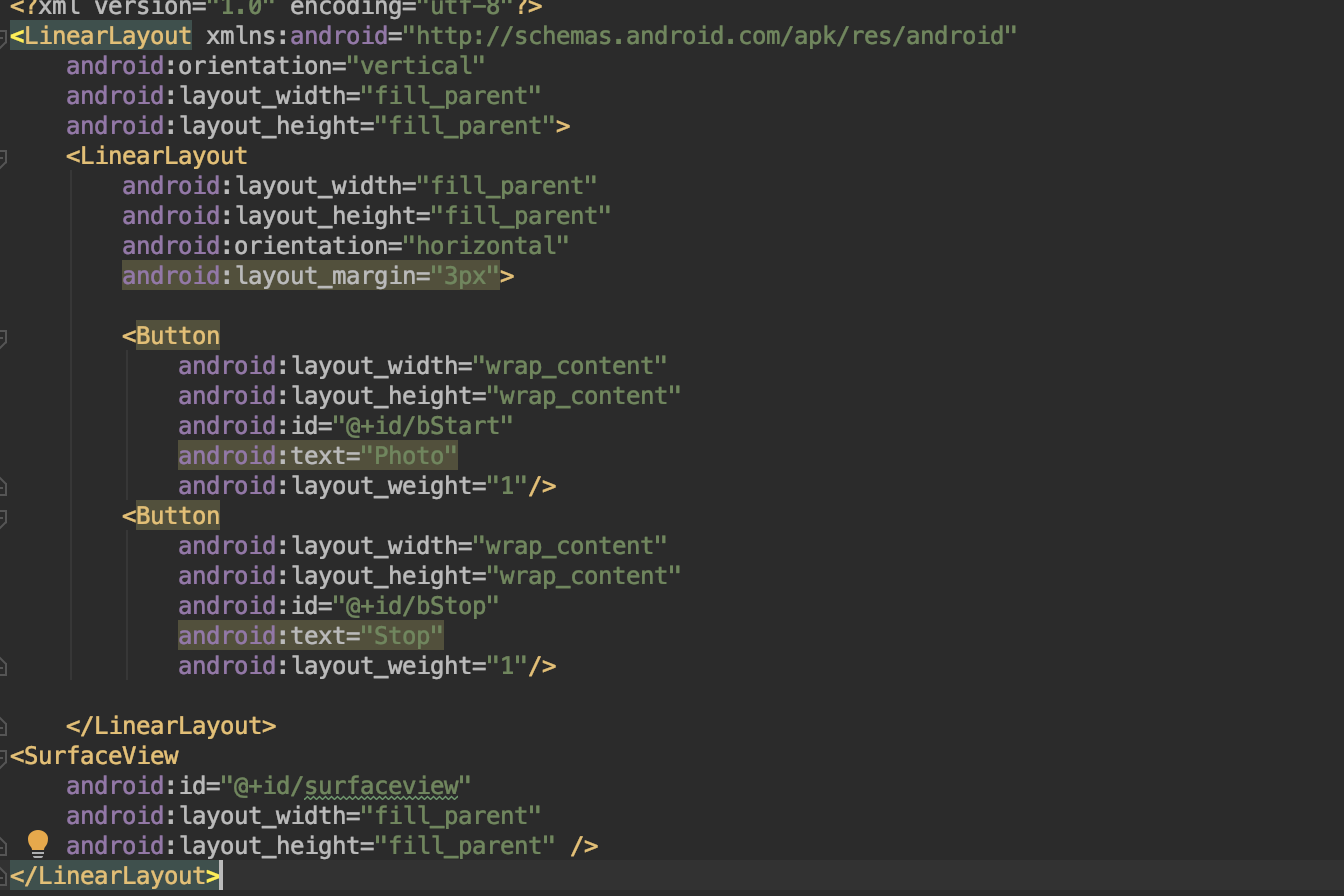


Рисунок 2 – Демонстрация макета для вызова камеры

2.2. Требования к входным и выходным данным

Входными данными являются текст виде десятичных чисел для веса упражнений и виде целых чисел для количество повторений. Также входными данными является фотография.

2.3. Требования к надежности

В программе должна присутствовать проверка входной информации на соответствие формальному вводу информации пользователем для восприятия форм на числа.

2.4. Требования к составу и параметрам технических средств

Минимальные системные требования:

1. Смартфон на операционной системе android, на минимальной версии 8.0.
2. Минимальная память для приложения 2 Мб.

2.5. Требования к информационной и программной совместимости

Программа должна взаимодействовать с android смартфонами не ниже версии android 8.0.

2.6. Требования к программной документации

Программная документация должна быть представлена окнами приложения.

3 Графический функционал.

3.1. Основное меню (см. Рис. 3.1, 3.2)



Рисунок 3.1, Рисунок 3.2 – Демонстрация окна основного меню

3.2. Меню с упражнениями для тренировок (см. Рис. 4.1, 4.2)

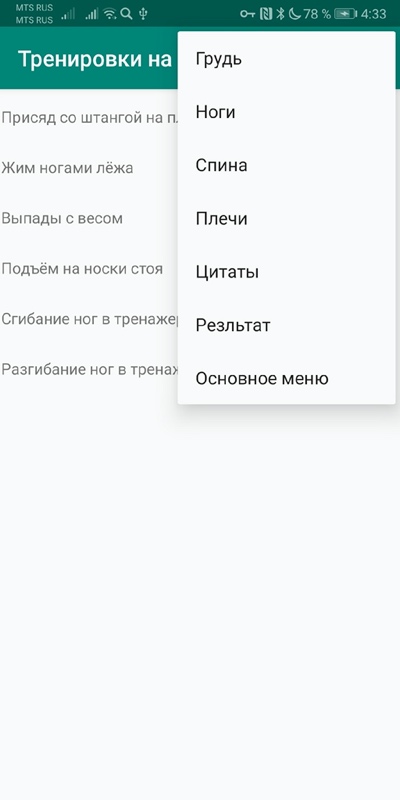


Рисунок 4.1, Рисунок 4.2 – Демонстрация основного меню с упражнениями

3.3. Меню с цитатами (см. Рисунок 5)

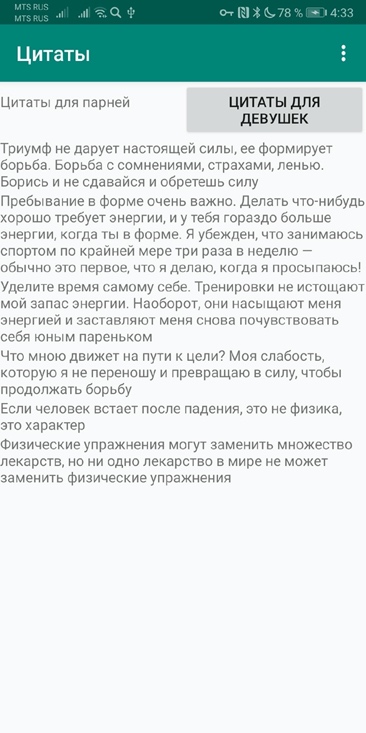


Рисунок 5 – Демонстрация окна с цитатами

Реализация поставленных задач представленны в следующей главе.

2. реализация проекта

2.1. Диаграмма Ганта

Для реализации графика этапов разработки была выбрана диаграмма Ганты.

Диаграмма Ганта — это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами.

На данной диаграмме Ганта можно хорошо рассмотреть этапы разработки и график работы по данному проекту. (см. Рисунок 6)

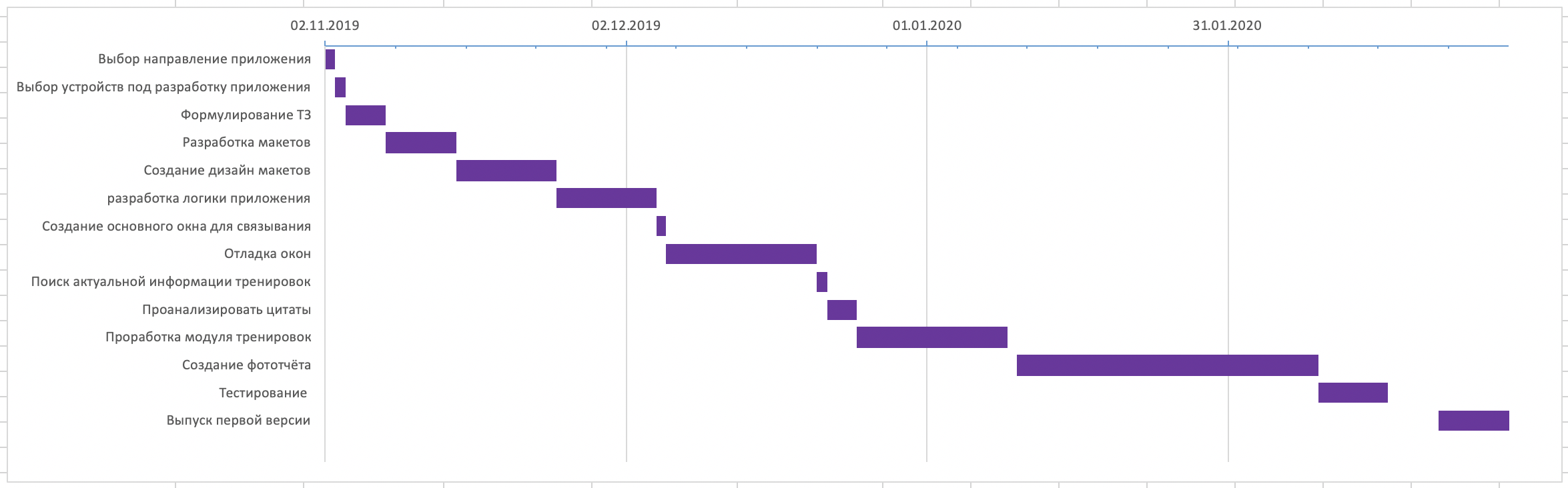


Рисунок 6 – Демонстрация Диаграммы Ганты

2.2. Trello

На основе выбора методологии «SCRUM», была выбрана доска канбан реализуемая в Trello, так как в ней удобно следить за ходом создания проекта и управлять командой для его лучшей реализации. Канбан - система организации производства и снабжения, позволяющая реализовать принцип «точно в срок». (см. Рисунок 7)

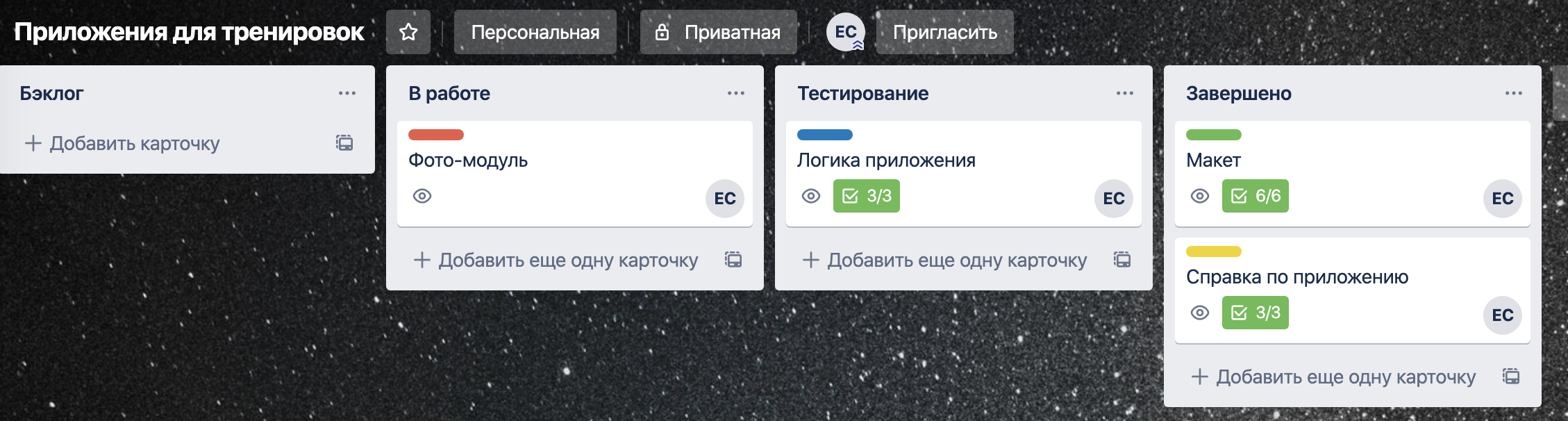


Рисунок 7 – Демонстрация доски Канбан

2.3. Диаграмма прецедентов

Для лучшей реализации управления приложения виде схемы было принято решение использовать диаграмму прецедентов.

Диаграмма вариантов использования (прецедентов) отражает отношения между пользователем и прецедентами. Позволяющей описать систему на концептуальном уровне, где видны этапы управления приложением.

(см. Рисунок 8)

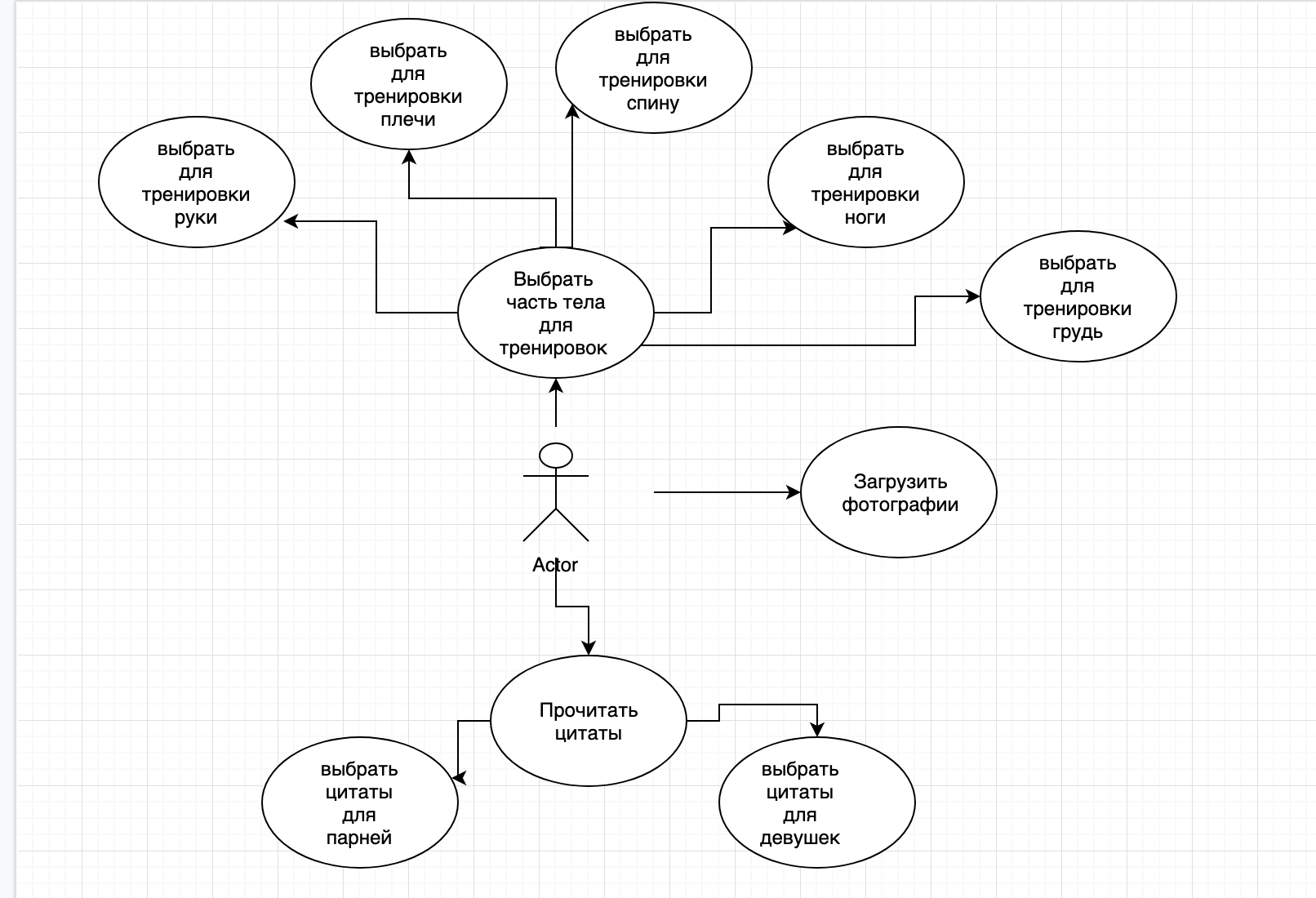


Рисунок 8 – Демонстрация диаграммы прецедентов

2.4. Тестирование сборки всех модулей

На этом этапе была проедена проверка на ошибки сборки всех модулей.

(см. Рисунок 9)

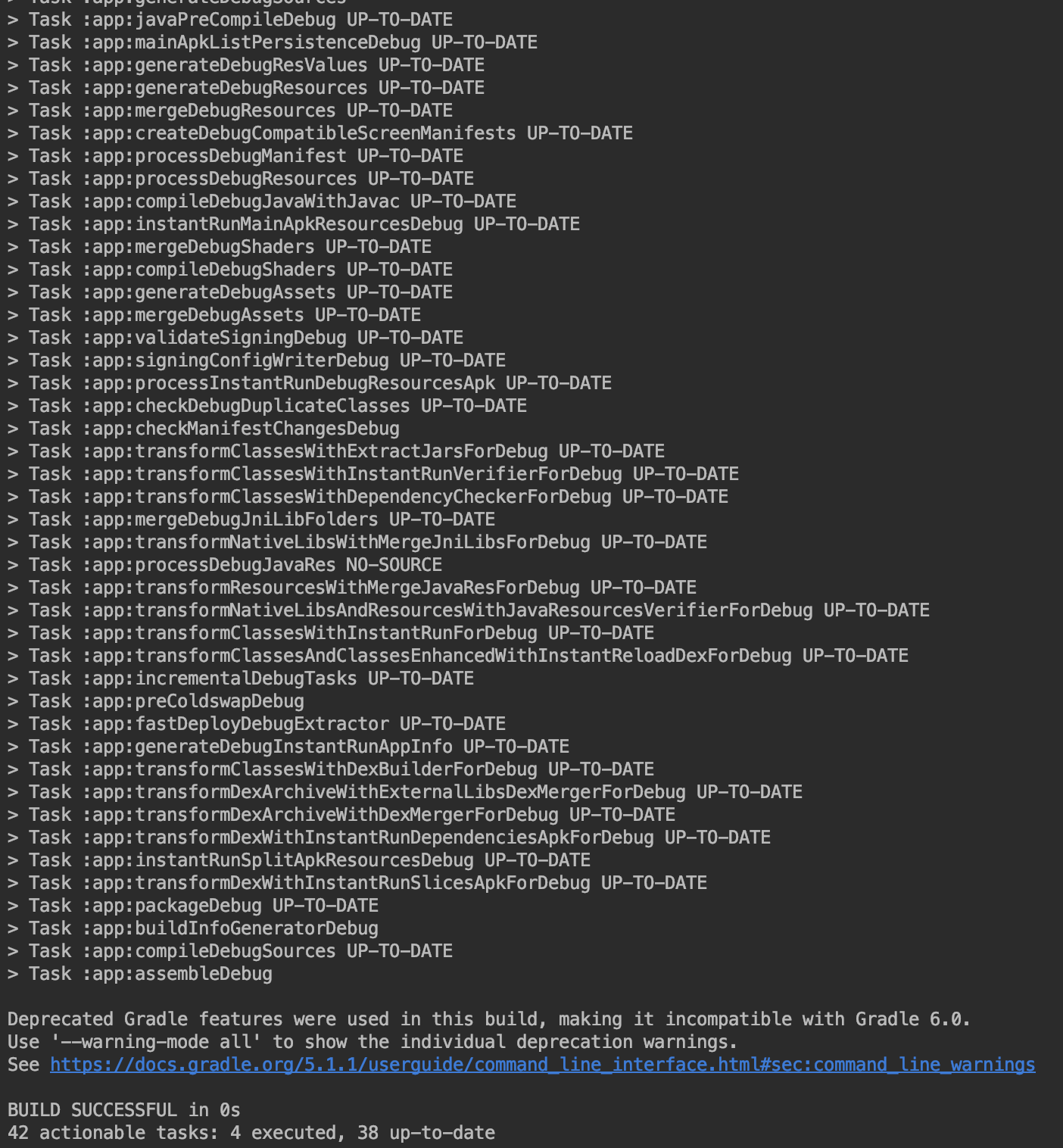


Рисунок 9 – Демонстрация сборки всех модулей

Заключение

При написании данной курсовой работы решались поставленные задачи посредством изучения различных способов их решения с помощью использования технологий компьютерного моделирования.  
В результате выполненной работы были смоделированы модели разработки приложения на android, вывели порядок управления разработки и разработали программный код на языке программирования Java.   
После пройдённой работы были улучшены макет и дизайн приложения, которые можно увидеть в Приложение 2.  
Получившееся приложение поможет пользователю в мотивации для тренировок и самих тренировок в спортивном зале.

СПИСОК ЛИТЕРАТУРЫ

1. Акопов А. С. Компьютерное моделирование: учебник и практикум для среднего профессионального образования / А. С. Акопов. — Москва: Издательство Юрайт, 2019. — 389 с.
2. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 689 c.
3. Берд, Барри Java для чайников / Барри Берд. - М.: Диалектика / Вильямс, 2013. - 521 c.
4. Булавин, Л.А., Выгорницкий Н.В., Лебовка Н.И. Компьютерное моделирование физических систем. - Долгопрудный: Издательский Дом «Интеллект», 2011. - 352 c.
5. Гарнаев, Андрей WEB-программирование на Java и JavaScript / Андрей Гарнаев, Сергей Гарнаев. - Москва: СПб. [и др.] : Питер, 2017. - 718 c.
6. Голощапов А.Л. Google Android Создание приложений для смартфонов и планшетных ПК. – БХВ-Петербург, 2013. – 831с
7. Гонсалвес, Энтони Изучаем Java EE 7 / Энтони Гонсалвес. - М.: Питер, 2016. - 640 c.
8. Градов В.М. Компьютерное моделирование: Учебник / В.М. Градов, Г.В. Овечкин, П.В. Овечкин и др. - М.: Инфра-М, 2016. - 784 c.
9. Гупта, Арун Java EE 7. Основы / Арун Гупта. - М.: Вильямс, 2014. - 336 c.
10. Королев А.Л. Компьютерное моделирование / А.Л. Королев. - М.: Бином. ЛЗ, 2013. - 230 c.
11. Марио Ц. Программирование Игр под Android. – СПБ.: Питер, 2013 – 688с
12. Михеева Е. В. Практикум по информационным технологиям в профессиональной деятельности / Е. В. Михеева. – Академия: Москва, 2015. – 256 c.
13. Монахов, В. Язык программирования Java и среда NetBeans (+ CD-ROM) / В. Монахов. - М.: БХВ-Петербург, 2012. - 720 c.
14. Орлова И.В. Экономико-математические методы и модели: компьютерное моделирование: Учебное пособие / И.В. Орлова. - М.: Вузовский учебник, НИЦ Инфра-М, 2013. - 389 c.
15. Сафонов В.И. Компьютерное моделирование: учебное пособие / В.И. Сафонов; Мордов. гос. пед. ин-т. – Саранск, 2009. – 92 с.
16. Сирота А.А. Анализ и компьютерное моделирование информационных процессов и систем / Э.К. Алгазинов, А.А. Сирота; под общ. ред. проф. д.т.н. Э.К. Алгазинов. - М.: ДИАЛОГ-МИФИ, 2009. - 416 c.
17. Тарасевич Ю.Ю. Математическое и компьютерное моделирование. Вводный курс: Учебное пособие / Ю.Ю. Тарасевич. - М.: ЛИБРОКОМ, 2013. - 152 c.
18. Тупицына А.И. Методы компьютерного моделирования физических процессов и сложных систем. Учебное пособие– СПб: Университет ИТМО, 2014. – 48 с.
19. Эккель, Брюс Философия Java / Брюс Эккель. - М.: Питер, 2016. - 809 c.

Приложения

Приложение 1

Создания специального view – RecyclerView для создания генерации списков

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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=".GymR"

tools:layout\_editor\_absoluteY="81dp">

<android.support.v7.widget.RecyclerView

android:id='@+id/recycler\_view'

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="1dp">

</android.support.v7.widget.RecyclerView>

</android.support.constraint.ConstraintLayout>

Создание макета списка

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="0dp"

android:orientation="horizontal">

<TextView

android:id="@+id/desc"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="3" />

<EditText

android:id="@+id/edtV"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:ems="10"

android:inputType="numberDecimal" />

<EditText

android:id="@+id/edtC"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:ems="10"

android:inputType="number" />

</LinearLayout>

Создание класса для списков

package com.example.myapplication;

public class Gym {

private float edtV;

private float edtC;

private String description;

public Gym(String description, int edtV, int edtC) {

this.description = description;

this.edtV = edtV;

this.edtC = edtC;

}

public String getDesc(){

return this.description;

}

public float getEdtV(){

return this.edtV;

}

public float getEdtC(){

return this.edtC;

}

}

Программное реализация создания списков

public class GymAdapter extends RecyclerView.Adapter<GymAdapter.GymViewHolder> {

ArrayList<Gym> gyms;

public GymAdapter (ArrayList<Gym> gyms){

this.gyms = gyms;

}

public class GymViewHolder extends RecyclerView.ViewHolder{

EditText edtV;

EditText edtC;

TextView desc;

public GymViewHolder (View view){

super(view);

edtV = (EditText) view.findViewById(R.id.edtV);

edtC = (EditText) view.findViewById(R.id.edtC);

desc = (TextView) view.findViewById(R.id.desc);

}

}

@NonNull

@Override

public GymViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.list\_gym, parent, false);

return new GymViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull GymViewHolder holder, int position) {

holder.desc.setText(gyms.get(position).getDesc());

holder.edtV.setText(String.valueOf(gyms.get(position).getEdtV()));

holder.edtC.setText(String.valueOf(gyms.get(position).getEdtC()));

}

@Override

public int getItemCount() {

return gyms.size();

}

}

Связующая программная реализация для объединения всего вышеперечисленного

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.probnik);

recyclerView = findViewById(R.id.recycler\_view);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

ArrayList<Gym> gyms = new ArrayList<Gym>();

gyms.add(new Gym("Присяд со штангой на плечах", 50, 10));

gyms.add(new Gym("Жим ногами лёжа ", 70, 12));

gyms.add(new Gym("Выпады с весом", 15, 15));

gyms.add(new Gym("Подъём на носки стоя", 40, 12));

gyms.add(new Gym("Сгибание ног в тренажере", 30, 15));

gyms.add(new Gym("Разгибание ног в тренажере", 30, 12));

GymAdapter gymAdapter = new GymAdapter(gyms);

recyclerView.setAdapter(gymAdapter);

Приложение 2

Улучшенные макеты приложения после пройденной работы.

