Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 1-3**

**по дисциплине**

**«Технологии разработки мобильных приложений»**

Тема работы: «Знакомство с операционной системой Android», «Изучение компонентов пользовательского интерфейса и обработчиков событий», «Переход между активностями. Работа с Intent»

Выполнил:

ст. гр. ПРИ-120

Ляпанов А.А.

Принял:

пр. каф. ИСПИ  
Петрова А.И.

Владимир, 2023

**Цель работы**

Изучить архитектуру платформы Androidи получить практические навыки создания мобильного приложения в среде разработки Android Studio.

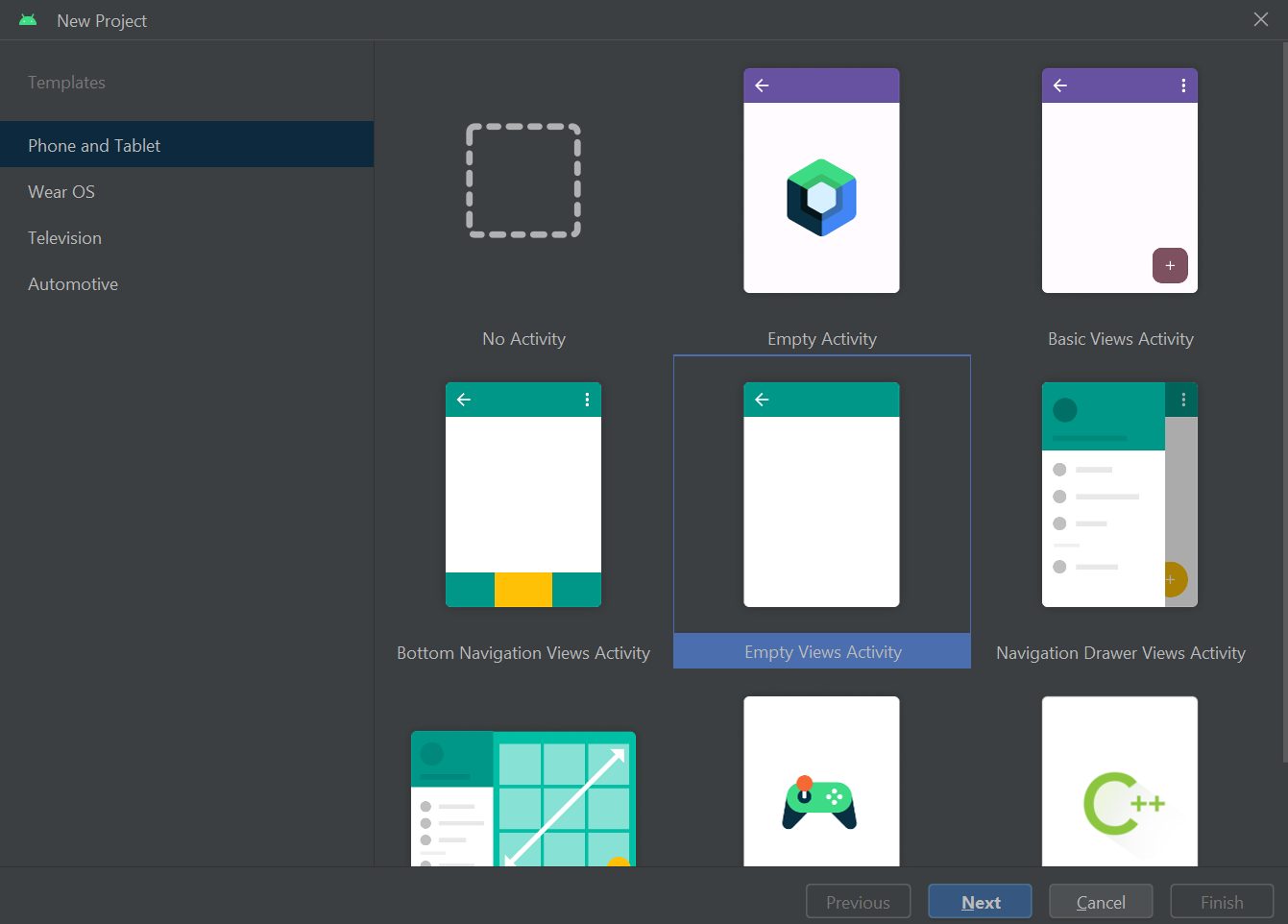
**Задание**

1. Выполнить пример из лабораторной работы.
2. Выполнить следующие задания:
   1. Изменить цвет текста.
   2. Переместить текст из центра в верхнюю часть экрана.

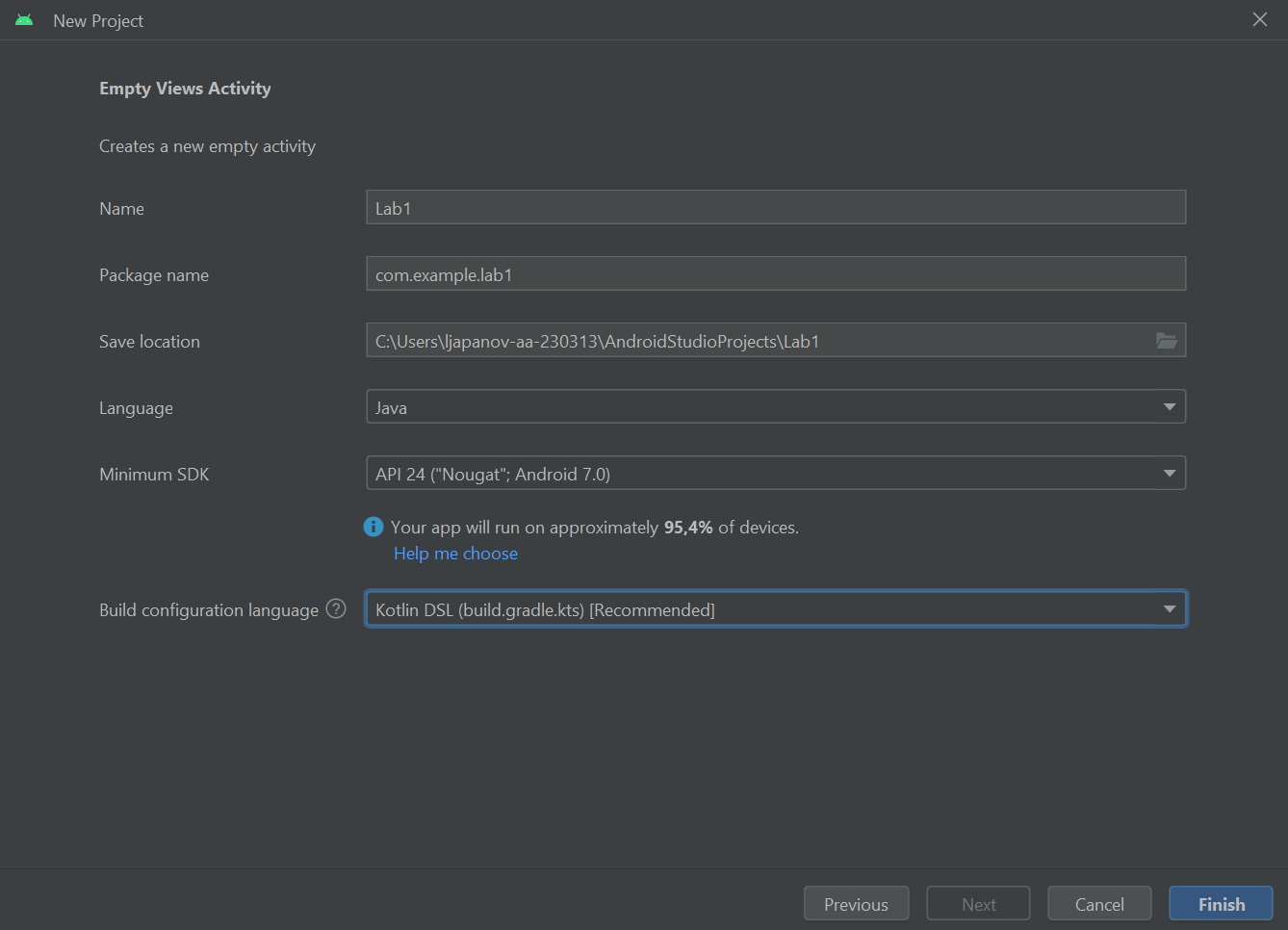
**Ход работы**

**Описание выполнения примера**

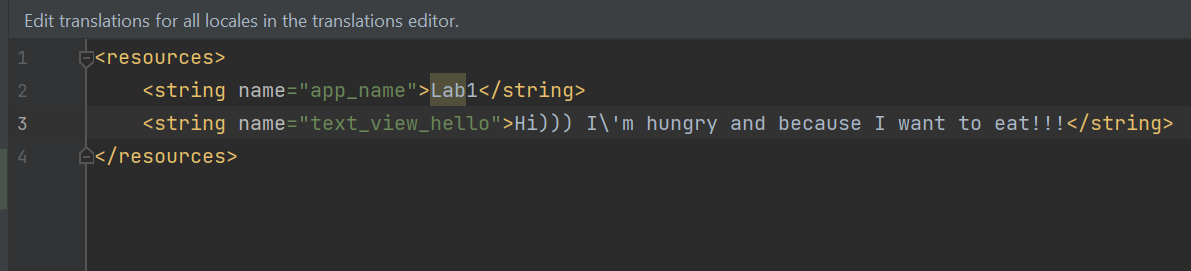
1. Скачиваем с официального сайта среду разработки мобильных приложений Android Studio. Выполняем установку необходимых компонентов.
2. Создаем новый проект, что продемонстрировано на рисунке 1.

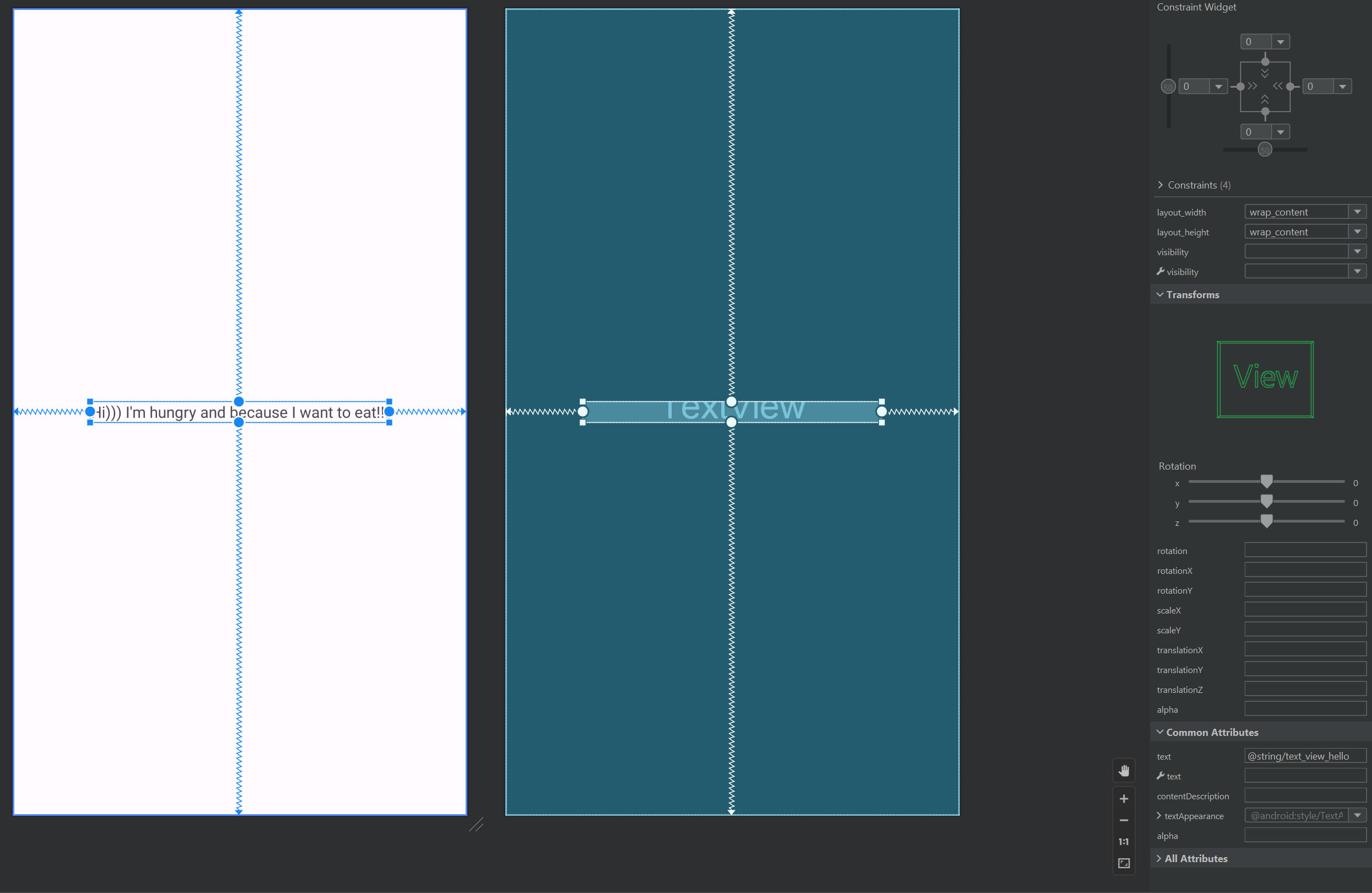
  
Рисунок 1 (Создание проекта и выбор шаблона)

1. Заполняем свойства проекта, в которые входит название проекта, имя пакета, место сохранения проекта, язык разработки, минимальный уровень API. Данный шаг продемонстрирован на рисунке 2.

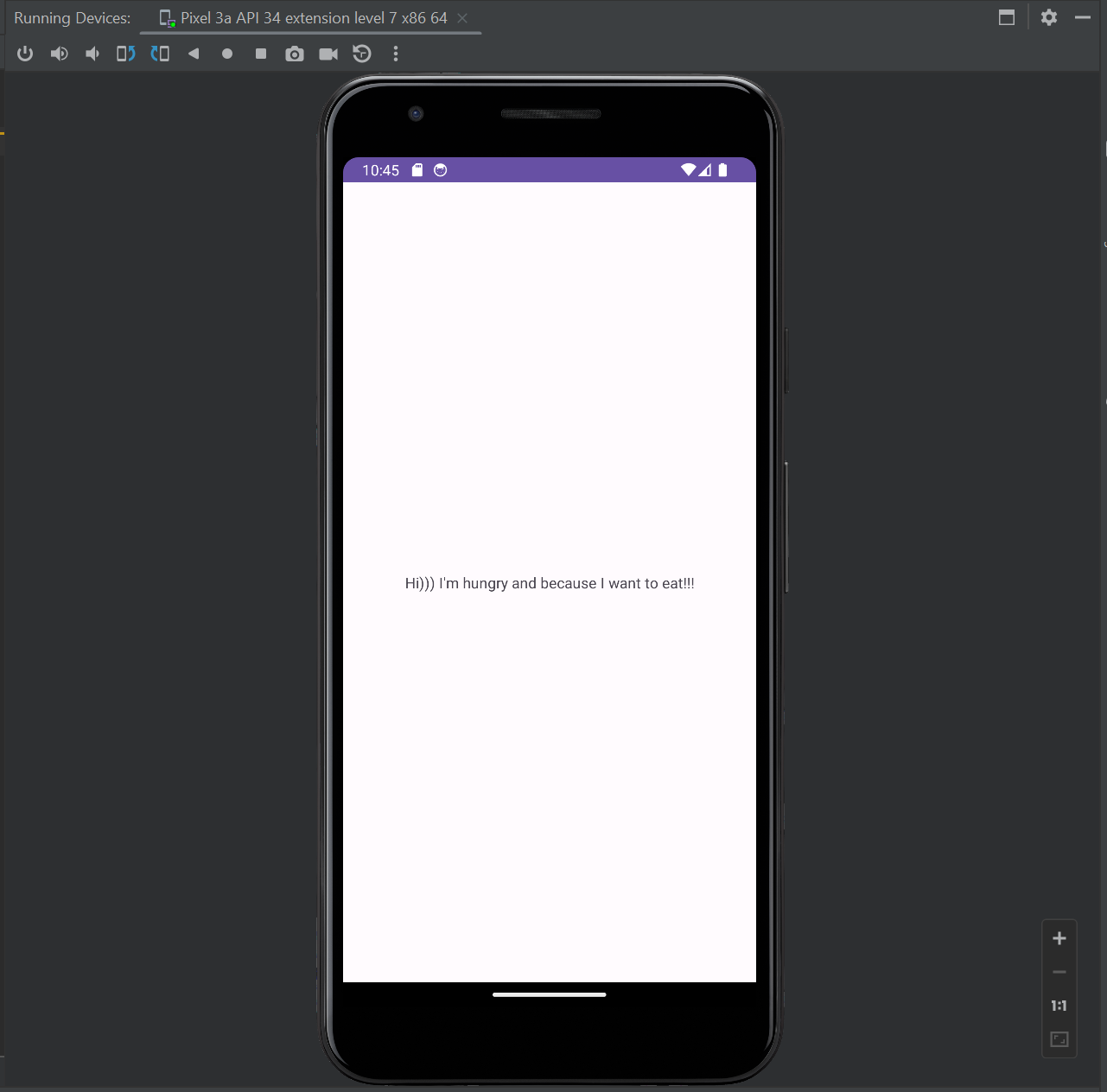
  
Рисунок 2 (Заполнение свойств проекта)

1. После сборки проекта выполним несколько преобразований с нашим исходным кодом. Вынесем в отдельный xml файл значение нашей выводимой строки, чтобы избежать дальнейшего дублирования текста в приложении, после чего в поле android:text для объекта TextView укажем ссылку на заданную нами хранимую строку в файле strings.   
   Данный шаг продемонстрирован на рисунках 3, 4.

  
Рисунок 3 (Скриншот файла strings.xml)

  
Рисунок 4 (Скриншот, демонстрирующий результат)

1. Выполняем запуск через эмулятор, после чего перед нами отображается результат, который продемонстрирован на рисунке 5.

  
Рисунок 5. (Скриншот, демонстрирующий результат работы программы)

**Описание выполнения индивидуального задания**

1. В листинге 1 представлена xml конфигурация, отвечающая за отображение элементов на дисплее.

**Листинг 1**

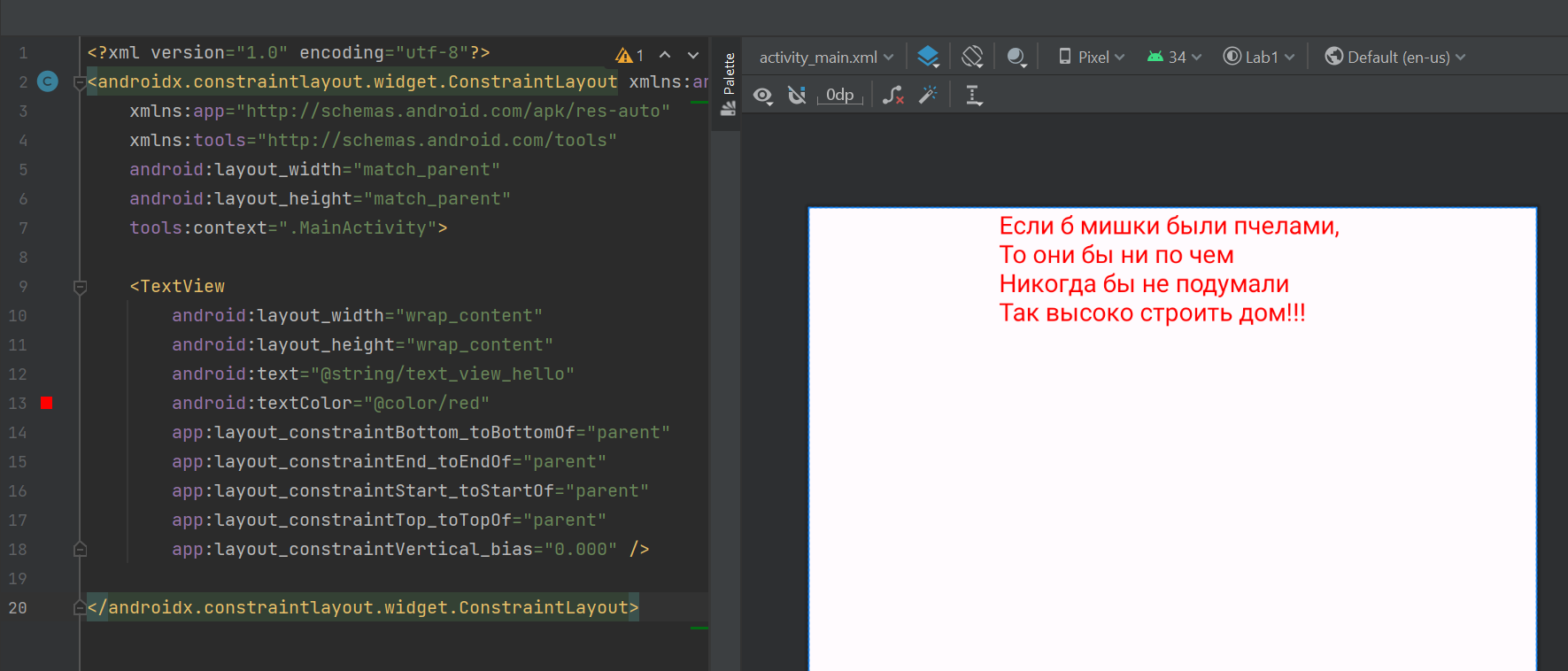
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/text\_view\_hello"  
 android:textColor="@color/red"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.000" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

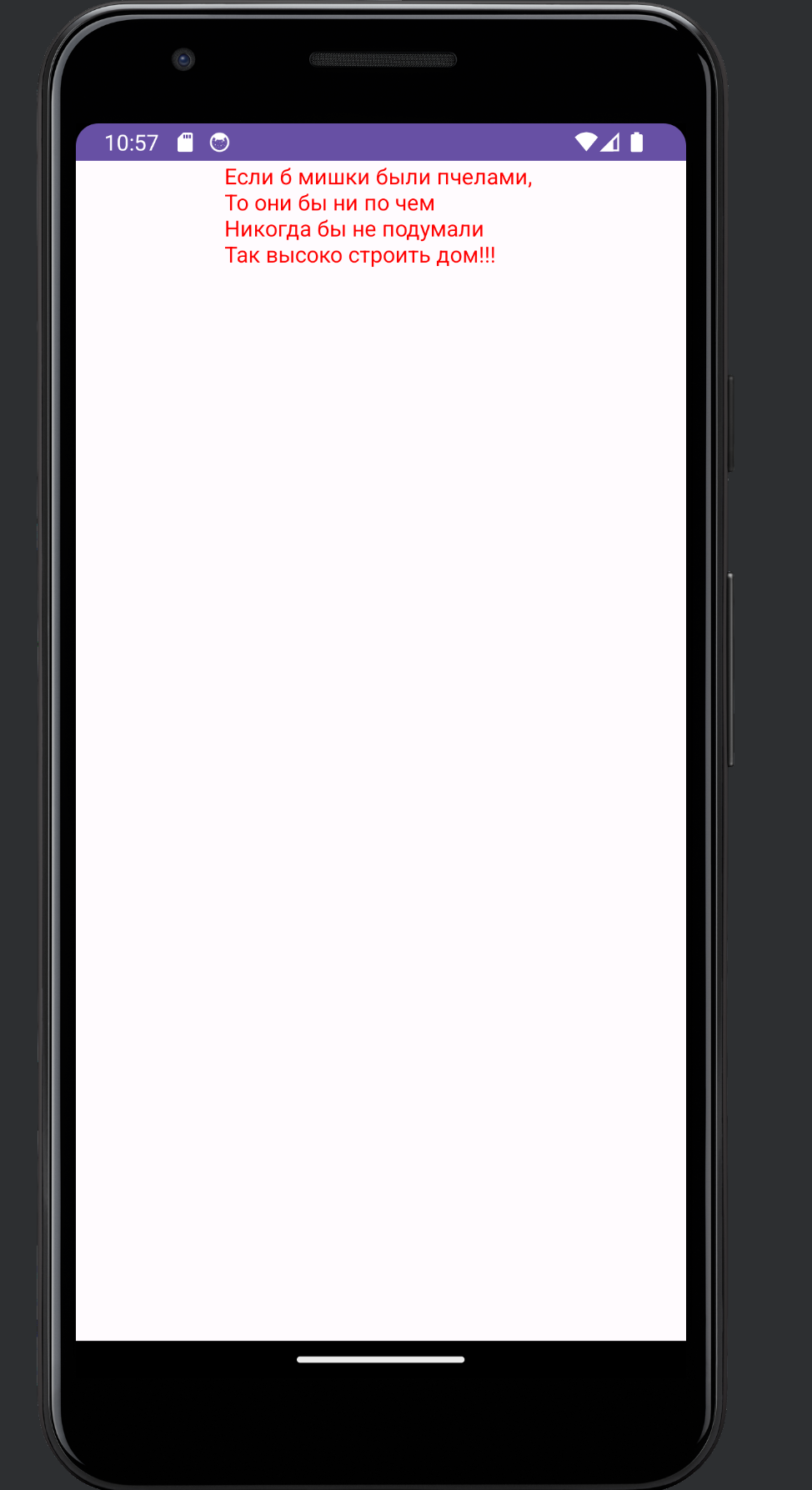
1. В листинге 2 представлены значения для хранимых строк.

**Листинг 2**

<resources>  
 <string name="app\_name">Lab1</string>  
 <string name="text\_view\_hello">Если б мишки были пчелами, \nТо они бы ни по чем \nНикогда бы не подумали \nТак высоко строить дом!!!</string>  
</resources>

1. На рисунке 6 представлен скриншот, демонстрирующий результат изменения цвета и расположения текста.

  
Рисунок 6 (Скриншот, демонстрирующий результат выполнения индивидуального задания)

  
Рисунок 7 (Скриншот, демонстрирующий итоговый запуск эмулятора)

**Вывод по лабораторной работе 1**

В ходе выполнения лабораторной работы была изучена архитектура платформы Androidи, а также удалось получить практические навыки создания мобильного приложения в среде разработки Android Studio.

**Лабораторная работа 2**

**Цель работы**

Изучить основные графические компоненты мобильного приложения и механизмы организации обработки событий.

**Задание**

Выполнить пример из лабораторной работы.

Изменить калькулятор по следующим заданиям.

1. Изменить логику таким образом, чтобы было возможно производить операции над двумя и больше числами.
2. Добавить операции взятия корня числа и возведения в квадрат.
3. Добавить возможность сохранять промежуточный результат и очищать его путем реализации кнопок M и MC.

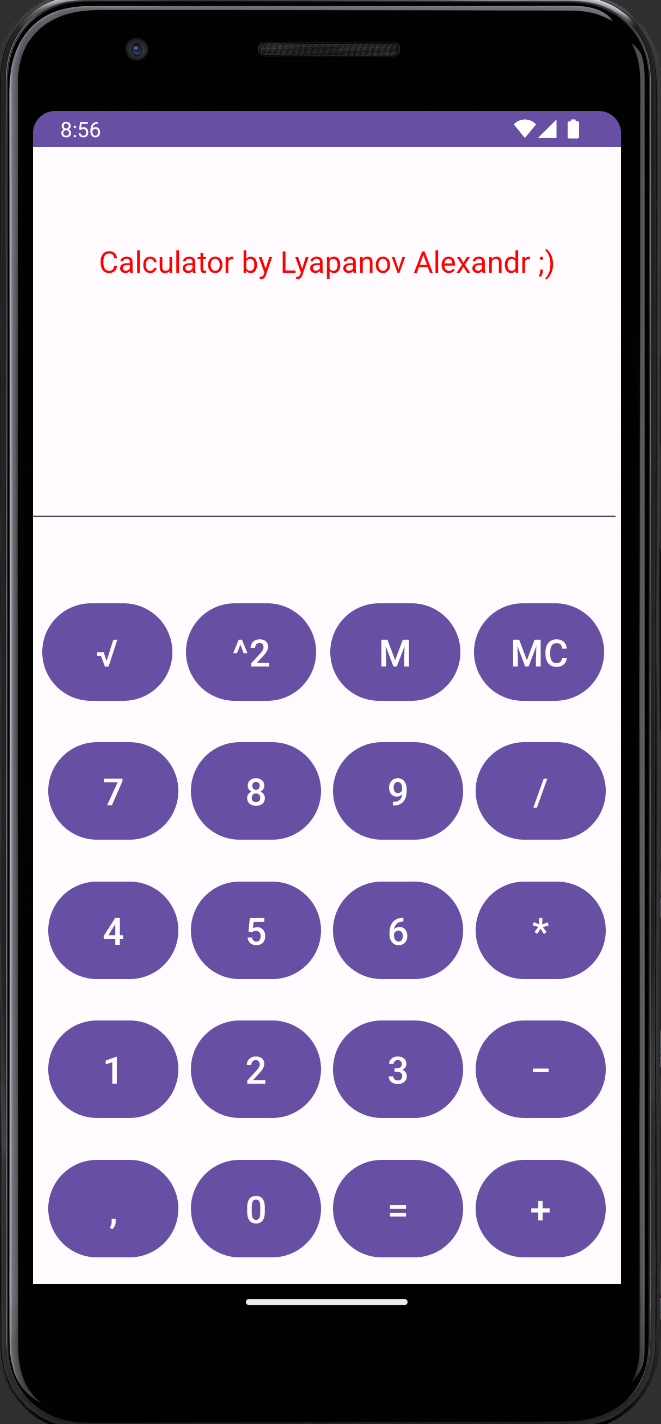
**Ход работы**

1. Создан новый Android Application Project.
2. На экран были добавлены необходимые пользовательские элементы интерфейса для реализации калькулятора. В листинге 3 представлен xml код, демонстрирующий некоторые элементы разметки экрана.

**Листинг 3**

<?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=".MainActivity">  
  
 <Button  
 android:id="@+id/buttonSqrt"  
 android:layout\_width="87dp"  
 android:layout\_height="73dp"  
 android:layout\_marginStart="3dp"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginEnd="3dp"  
 android:layout\_marginBottom="3dp"  
 android:text="&#8730;"  
 android:textSize="25dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/button7"  
 app:layout\_constraintEnd\_toStartOf="@+id/button2Deg"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText" />  
  
 <Button  
 android:id="@+id/button2Deg"  
 android:layout\_width="87dp"  
 android:layout\_height="73dp"  
 android:layout\_marginStart="3dp"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginEnd="3dp"  
 android:layout\_marginBottom="3dp"  
 android:text="^2"  
 android:textSize="25dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/button8"  
 app:layout\_constraintEnd\_toStartOf="@+id/buttonM"  
 app:layout\_constraintStart\_toEndOf="@+id/buttonSqrt"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText" />

1. На рисунке 8 представлен скриншот, демонстрирующий экран калькулятора.

  
Рисунок 8 (Скриншот экрана калькулятора)

1. Далее был написан обработчик для нажатий на кнопки. В листинге 4 представлен код, демонстрирующий обработчик события для кнопки =.

**Листинг 4**

mButtonEq.setOnClickListener(new View.OnClickListener() {  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void onClick(View view) {  
 mValueTwo = Float.*parseFloat*(mEditText.getText() + "");  
 if (mAddition) {  
 mEditText.setText(mValueFour + mValueTwo + "");  
 mAddition = false;  
 }  
 if (mSubtract) {  
 mEditText.setText(mValueFour - mValueTwo + "");  
 mSubtract = false;  
 }  
 if (mMultiplication) {  
 mEditText.setText(mValueFour \* mValueTwo + "");  
 mMultiplication = false;  
 }  
 if (mDivision) {  
 mEditText.setText(mValueFour / mValueTwo + "");  
 mDivision = false;  
 }  
 }  
});

1. В листинге 5 представлен код, демонстрирующий метод для возможности производить операции над двумя и больше числами.

**Листинг 5**

public void calculate(String exp) {  
 if (!mAddition && !mDivision && !mSubtract && !mMultiplication) {  
 mValueOne = Float.*parseFloat*(mEditText.getText() + "");  
 mValueFour = Float.*parseFloat*(mEditText.getText() + "");  
 if (exp.equals("addition")) {  
 mAddition = true;  
 } else if (exp.equals("sub")) {  
 mSubtract = true;  
 } else if (exp.equals("division")) {  
 mDivision = true;  
 } else if (exp.equals("multiplication")) {  
 mMultiplication = true;  
 }  
 mEditText.setText(null);  
 } else if (mAddition) {  
 mValueTwo = Float.*parseFloat*(mEditText.getText() + "");  
 mValueFour = mValueFour + mValueTwo;  
 mEditText.setText(null);  
 } else if (mSubtract) {  
 mValueTwo = Float.*parseFloat*(mEditText.getText() + "");  
 mValueFour = mValueFour - mValueTwo;  
 mEditText.setText(null);  
 } else if (mMultiplication) {  
 mValueTwo = Float.*parseFloat*(mEditText.getText() + "");  
 mValueFour = mValueFour \* mValueTwo;  
 mEditText.setText(null);  
 } else if (mDivision) {  
 mValueTwo = Float.*parseFloat*(mEditText.getText() + "");  
 mValueFour = mValueFour / mValueTwo;  
 mEditText.setText(null);  
 }  
}

1. В листинге 6 представлен обработчик событий для кнопки M (memory)

**Листинг 6**

mButtonM.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (mEditText == null || mEditText.getText().equals("")) {  
 mEditText.setText(mValueThree + "");  
 } else {  
 mValueThree += Float.*parseFloat*(mEditText.getText() + "");  
 mEditText.setText(mValueThree + "");  
 }  
 }  
});

1. В листинге 7 представлен обработчик событий для кнопки MC

**Листинг 7**

mButtonMC.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mEditText.setText(null);  
 mValueOne = 0.0f;  
 mValueTwo = 0.0f;  
 }  
});

1. В листинге 8 представлен обработчик событий для кнопки взятия корня.

**Листинг 8**

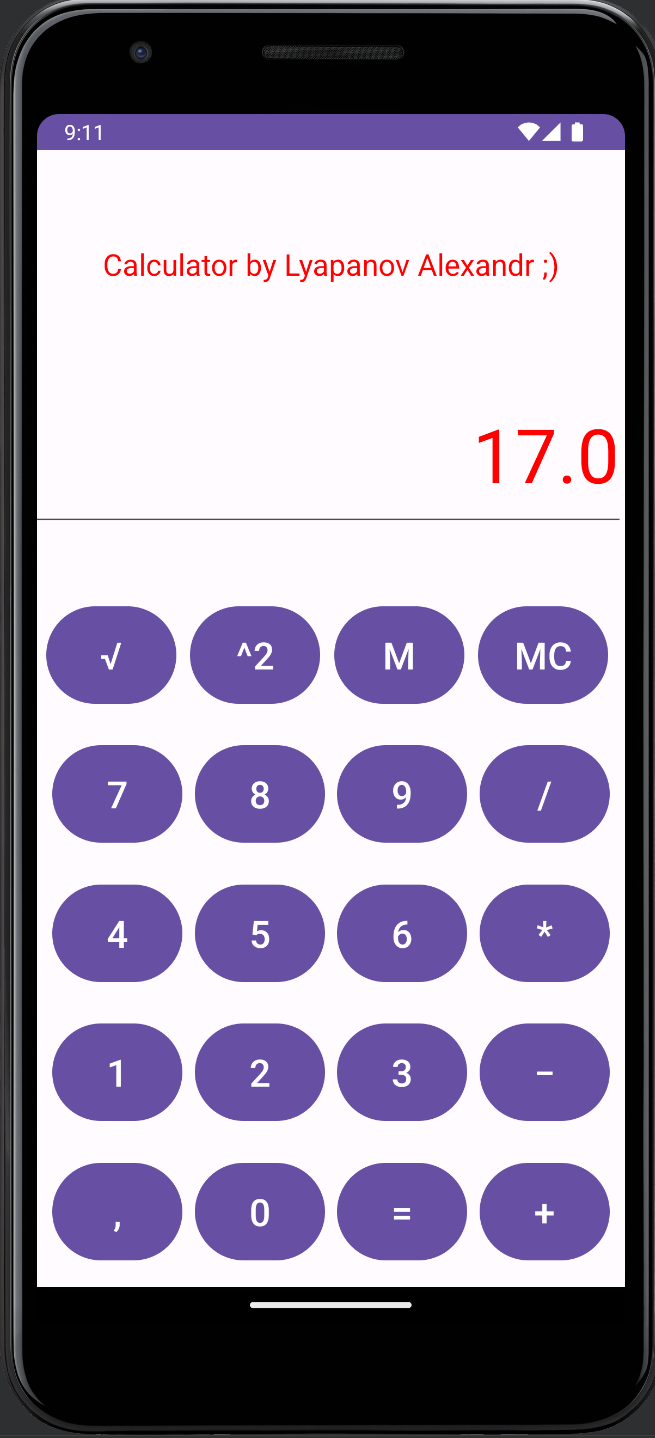
mButton2Deg.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (mEditText == null || mEditText.getText().equals("")) {  
 mEditText.setText("");  
 } else {  
 mValueOne = Float.*parseFloat*(mEditText.getText() + "");  
 mEditText.setText((mValueOne \* mValueOne) + "");  
 }  
 }  
});

1. В листинге 9 представлен обработчик событий для кнопки возведения во 2 степень.

**Листинг 9**

mButton2Deg.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if (mEditText == null || mEditText.getText().equals("")) {  
 mEditText.setText("");  
 } else {  
 mValueOne = Float.*parseFloat*(mEditText.getText() + "");  
 mEditText.setText((mValueOne \* mValueOne) + "");  
 }  
 }  
});

1. На рисунке 9 представлен скриншот, демонстрирующий работу приложения.

  
Рисунок 9 (Скриншот работы приложения)

**Вывод по лабораторной работе 2**

В ходе выполнения лабораторной работы были изучены основные графические компоненты мобильного приложения и механизмы организации обработки событий.

**Лабораторная работа 3**

**Цель работы**

Изучить функционал, позволяющий осуществлять переход между активностями приложения и научиться работать с Intent.

**Задание**

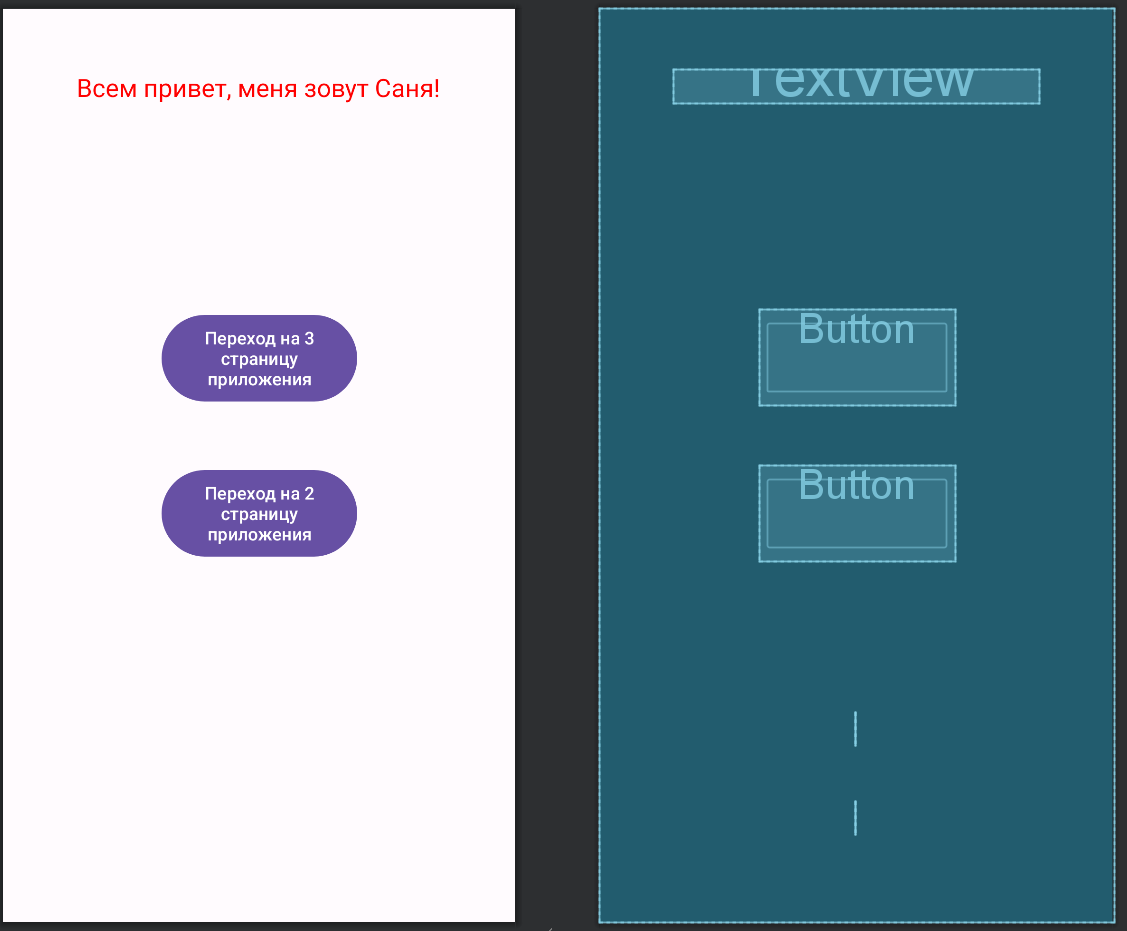
1. Выполнить пример из ЛР.
2. Выполнить след. задания:

2.1) Добавить переход на 3 активность. Добавить переход с 3 активности на 1.

2.2) Реализовать передачу данных при переходе с активности 1 на активность 2.

**Ход работы**

1. Был создан новый Android Application Project, который изначально содержит 1 Activite – «MainActivity».
2. Через xml файл была сконфигурирована первая Activity. Использовались такие виджеты, как: TextView и Button.
3. На рисунке 10 представлен скриншот, демонстрирующий стартовую страницу приложения.

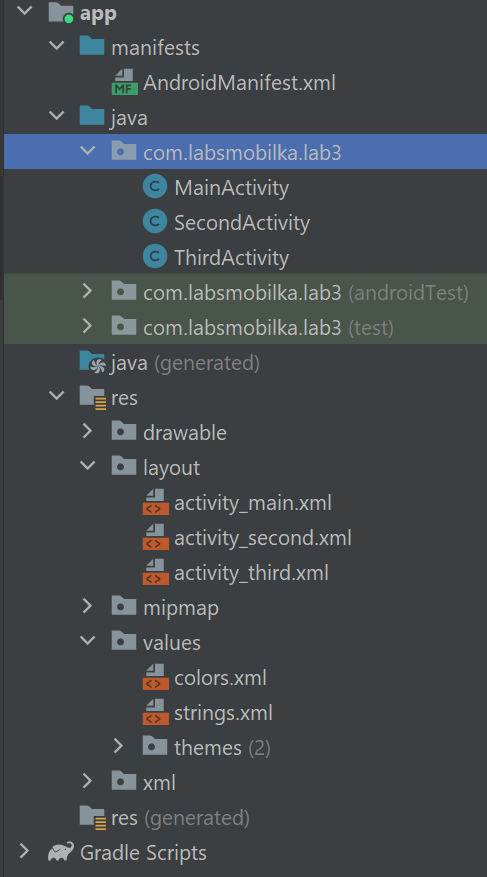
  
Рисунок 10 (Стартовая страница приложения)

1. Далее была прописана логика работы первой Activity в классе MainActivity.java, код которой представлен в листинге 10.

**Листинг 10**

package com.labsmobilka.lab3;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Bundle arguments = getIntent().getExtras();  
 Button button1 = findViewById(R.id.*button1*);  
 Button button2 = findViewById(R.id.*buttonToThird*);  
 TextView textView1 = findViewById(R.id.*text3*);  
 TextView textView2 = findViewById(R.id.*text4*);  
 if (arguments != null) {  
 String personName = arguments.getString("name");  
 String resultOfCalculation = arguments.getString("result");  
 textView1.setText(personName);  
 textView2.setText(resultOfCalculation);  
 }  
 button1.setOnClickListener(this);  
 button2.setOnClickListener(this);  
  
 }  
 @Override  
 public void onClick(View view) {  
 if (view.getId() == R.id.*button1*) {  
 Intent intent = new Intent(this, SecondActivity.class);  
 startActivity(intent);  
 }  
 if (view.getId() == R.id.*buttonToThird*) {  
 Intent intent = new Intent(this, ThirdActivity.class);  
 startActivity(intent);  
 }  
 }  
}

1. Далее в проект были добавлены еще две Activity: (SecondActivity, ThirdActivity). Добавление данных Activity производилось следующим образом (п.к.м. по папке, где находится MainAcitvity и далее (New -> Activity -> Empty View Activity)). В результате проект приобрел структуру, которая продемонстрирована на рисунке 11.

  
Рисунок 11 (Структура итогового проекта)

1. После проделанных действий в файлах activity\_second и activity\_third была выполнена разметка соответствующих Activity. Необходимый код можно просмотреть в следующем ресурсе: <https://github.com/SashaLyapanov/Mobile_development/tree/main>.
2. На рисунке 12 представлен скриншот, демонстрирующий экран второй Activity.

  
Рисунок 12 (Скриншот второй Activity)

1. На рисунке 13 представлен скриншот, демонстрирующий экран третьей Activity.

  
Рисунок 13 (Скриншот третьей Activity)

1. В файле SecondActivity прописана логика для второй Acitvity. В листинге 11 представлен код класса SecondActivity.java.

**Листинг 11**

package com.labsmobilka.lab3;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class SecondActivity extends AppCompatActivity implements View.OnClickListener{  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_second*);  
 Button button = findViewById(R.id.*button2*);  
 button.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View view) {  
 EditText name = findViewById(R.id.*input\_name*);  
 if (view.getId() == R.id.*button2*) {  
 Intent intent = new Intent(this, MainActivity.class);  
 intent.putExtra("name", name.getText().toString());  
 startActivity(intent);  
 }  
 }  
  
}

1. В файле ThirdActivity прописана логика для третьей Acitvity. В листинге 12 представлен код класса ThirdActivity.java, позволяющий взаимодействовать данной Activity c MainActivity.

**Листинг 12**

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_third*);  
 Button goBack = findViewById(R.id.*go\_to\_main*);  
 goBack.setOnClickListener(this);

…………

}

@Override  
public void onClick(View view) {  
 if (view.getId() == R.id.*go\_to\_main*) {  
 EditText result = findViewById(R.id.*editText*);  
 Intent intent = new Intent(this, MainActivity.class);  
 intent.putExtra("result", result.getText().toString());  
 startActivity(intent);  
 }  
}

1. На риунке 14 – 15 представлены скриншоты работы приложения.

  
Рисунок 13 (Работа приложения)

  
Рисунок 14 (Работа приложения)

**Вывод по лабораторной работе 3**

В ходе выполнения лабораторной работы был изучен функционал, позволяющий осуществлять переход между активностями приложения и получен навык работы с объектом Intent.

**Вывод по всей работе**

В ходе выполнения были изучены основы мобильной Android разработке в среде Android Studio. Освоены основные элементы интерфейса, работа с Activity и Intent. Код лабораторной работы представлен на github в публичном репозитории <https://github.com/SashaLyapanov/Mobile_development/tree/main>.