Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по курсу **«Графические интерфейсы Android и QT»**

по лабораторной работе №1

на тему «**Создание проекта**»

Выполнили  
студенты группы 19ВВ3:  
 Яцков В.В.

Приняли:  
Митрохин М. А.

Пенза 2022

**Ход работы:**

1. Установили Android Studio и создали проект на android 5.
2. Создали класс активности, и наследовали его от системного класса Activity

**public class** HelloActivity **extends** Activity {   
   
}

1. Прописали активность в manifest:

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

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

xmlns:tools="http://schemas.android.com/tools"

package="com.example.lab1">

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.Lab1"

tools:targetApi="31">

<activity android:name=".HelloActivity"

android:exported="true">

</activity>

1. Добавили в созданную активность в манифесте intent-filter и прописали категорию LAUNCHER

<activity android:name=".HelloActivity">

<intent-filter>

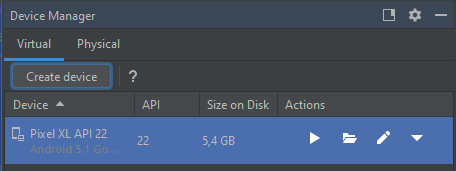
<action android:name="android.intent.action.MAIN"/>

<category android:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

1. Выбрали устройсто для тестирования Pixel XL



1. Протестировали приложение
2. Далее создали файл верстки. В каталоге *res* создали каталог *layout* и в нем файл *activity\_helloact.xml. В код, который автоматически создался, прописали визуальные элементы view. Текст, кнопку.*

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:text="Hello!"/>

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:text="Нажми!"/>

</FrameLayout>

1. Также в классе активности объявили нашу кнопку с использованием id и создали метод setOnClickListener(), который прослушивает нажатия на кнопку и будет обрабатывать события

public class HelloActivity extends Activity {

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_helloact);

Button button1 = (Button)findViewById(R.id.button1);

button1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

button1.setText("НАЖАТО!");

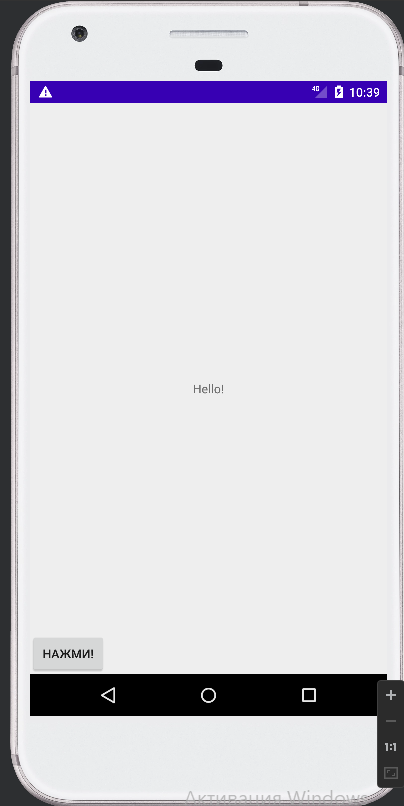
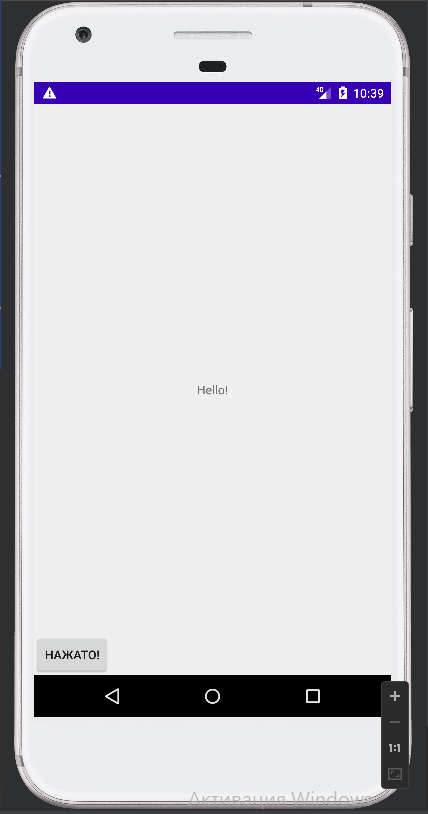
}

});

}

}

1. Запустили и протестировали проект.

Теперь нужно модифицировать программу добавив кликер возрастания и кликер убывания на кнопку, а также отображать кол-во кликов.

Для этого добавили вторую кнопку и изменили Text View. Теперь Activity\_main.xml выглядит так:

<?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=".HelloActivity">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button +"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.748"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.499" />  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button -"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.242"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.499" />  
  
 <TextView  
 android:id="@+id/score"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="0"  
 android:textSize="48sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.313" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Теперь задали функционал в HelloActivity:

package com.example.lab1;

public class HelloActivity extends AppCompatActivity {

int count = 0;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

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

Button b1 = findViewById(R.id.button);

Button b2 = findViewById(R.id.button2);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

count++;

score.setText(String.valueOf(count));

}

});

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

count--;

score.setText(String.valueOf(count));

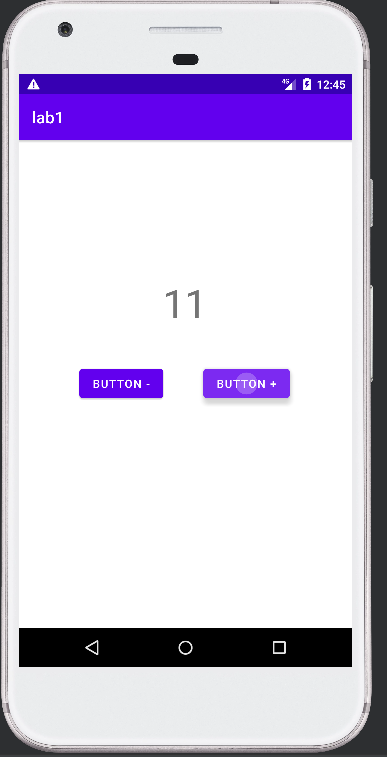
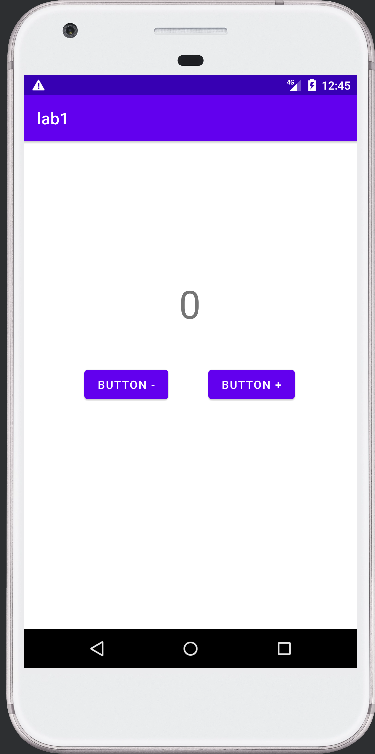
}

});

}

}

Результат работы программы:



Вывод: Научились создавать проект в android studio и получили базовые знания верстки мобильного приложения.