## Кликер

### Цель работы

Познакомиться с основными элементами управления приложения и научиться их использовать согласно логике приложения.

### Задания для выполнения

1. Создайте простое приложение-кликер. Главное и единственное окно приложения должно демонстрировать кнопку с надписью “Нажми меня” и надпись “Вы нажали кнопку 0 раз”. При каждом нажатии на кнопку, надпись соответствующим образом изменяется.

### Методические указания

Создадим новый проект так же, как на прошлой работе. Назовем его clicker.

Откроем XML-файл с активностью. Добавим текстовое поле:

<TextView

android:id="@+id/mainTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кликов: 0"

Под ним добавим кнопку:

<Button

android:id="@+id/main\_btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/click\_me"/>

Перейдем в файл с кодом активности. Для начала введем переменные для наших элементов управления:

TextView mainText;

Button mainBtn;

Еще на понадобится переменная, содержащая значение количества тапов по кнопке. Она может быть целого или длинного типа.

private long score = 0;

Эти переменные будут являться членами класса.

Затем в методе при загрузке приложения найдем по идентификатору ссылки на соответствующие элементы управления:

mainText = (TextView) findViewById(R.id.*mainTxt*);

mainBtn = (Button) findViewById(R.id.*main\_btn*);

Теперь создадим обработчик события нажатия на кнопку. В этой работе мы попробуем создать новый класс для обработки события. Это более правильный подход, чтобы не загромождать главный класс активности всеми возможными событиями приложения. В Java можно создавать класс сразу при его объявлении по месту, объединив объявление класса с вызовом конструктора:

View.OnClickListener clickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

}

};

Обратите внимание на точку с запятой в конце объявления класса - это все-таки оператор присваивания.

Таким образом, мы создали анонимный класс-потомок класса View.OnClickListener() с одним переопределенным методом - onClick() и сразу же создали экземпляр этого класса в переменную clickListener. Это полезно, если Вы планируете использовать определение класса только один раз.

Метод onClick() должен будет вызываться при нажатии на кнопку. В определении этого метода нам нужно произвести следующие действия:

Увеличиваем значение счетчика нажатий:

score ++;

Составляем нужную строку:

String s = "Кликов: " + score;

Выводим эту строку в текстовое поле:

mainText.setText(s.toCharArray(),0, s.length());

Теперь нам осталось только назначить обработчик события нужному нам элементу управления:

mainBtn.setOnClickListener(clickListener);

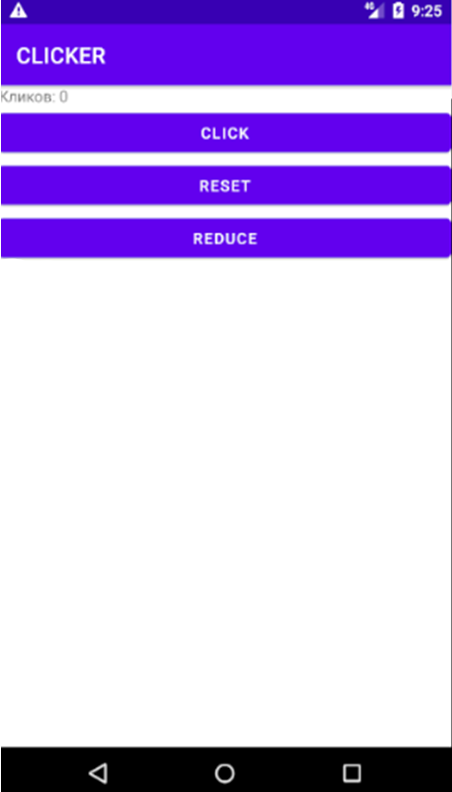
Запустим наше приложение. Мы должны увидеть работающую кнопку и счетчик нажатий:

|  |  |
| --- | --- |
|  |  |

При желании Вы можете изменить оформление или позиционирование этих элементов.

### Дополнительные задания

1. Добавьте в приложение кнопку, которая наоборот, уменьшает счетчик нажатий.
2. Добавьте кнопку, сбрасывающую счетчик.
3. Модифицируйте надпись так, чтобы она гласила “Кнопка нажата N раз”. Реализуйте правильное склонение слова “раз”.
4. Используйте вместо кнопки произвольную картинку.
5. (\*) Создайте приложение-калькулятор. Приложение должно предоставлять пользователю возможность вводить числа тапами по кнопкам и с клавиатуры.



activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 >  
  
 <TextView  
 android:id="@+id/mainTxt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Кликов: 0"  
 />  
  
 <Button  
 android:id="@+id/main\_btn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/click"  
 />  
  
 <Button  
 android:id="@+id/reset"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/reset"  
 />  
  
 <Button  
 android:id="@+id/reduce"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/reduce"  
 />  
  
 <ImageButton  
 android:id="@+id/imageButton"  
 android:layout\_width="82dp"  
 android:layout\_height="111dp"  
 android:src="@drawable/pug"  
 />

MainActivity.java

package com.example.clicker;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import com.example.clicker.R;  
  
public class MainActivity extends AppCompatActivity {  
 private long score = 0;  
 String click;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 TextView mainText;  
 Button mainBtn;  
 mainText = (TextView) findViewById(R.id.*mainTxt*);  
 mainBtn = (Button) findViewById(R.id.*main\_btn*);  
 View.OnClickListener clickListener = new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 score ++;  
 String click = "Кликов: " + score;  
 mainText.setText(click.toCharArray(),0, click.length());  
 }  
 };  
 mainBtn.setOnClickListener(clickListener);  
 View.OnClickListener reduce = new View.OnClickListener() {  
 @Override  
 public void onClick(View V) {  
 score --;  
 mainText.setText(click.toCharArray(),0, click.length());  
 }  
 };  
 mainBtn.setOnClickListener(reduce);  
 View.OnClickListener reset = new View.OnClickListener() {  
 @Override  
 public void onClick(View V) {  
 score = 0;  
 mainText.setText(click.toCharArray(),0, click.length());  
 }  
 };  
 mainBtn.setOnClickListener(reset);  
  
 }

}

Strings.xml

<resources>  
 <string name="app\_name">CLICKER</string>  
 <string name="click">click</string>  
 <string name="reset">reset</string>  
 <string name="reduce">reduce</string>  
</resources>