Touch. Обработка касания и получение координат касания

Создадим простой проект, где будем ловить касания до экрана. Кроме того, будем в **TextView** выводить координаты места касания — параметры **x** (по горизонтали) и **y** (по вертикали).

Используемые строковые ресурсы в проекте **res/values/strings.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <resources>      <string name="app\_name">Touch</string>      <string name="coords">Coords</string>      <string name="action\_settings">Settings</string>  </resources> |

Файл layout-а — **res/layout/activity\_main.xml**

|  |
| --- |
| <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:id="@+id/rl"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      android:paddingBottom="@dimen/activity\_vertical\_margin"      tools:context=".MainActivity">        <TextView          android:id="@+id/tv"          android:text="@string/coords"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content" />    </RelativeLayout> |

Код java **MainActivity.java**

|  |
| --- |
| package ru.androiddocs.touch;    import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.view.MotionEvent;  import android.view.View;  import android.widget.TextView;      public class MainActivity extends ActionBarActivity implements View.OnTouchListener {        TextView tv;      float mX;      float mY;      String mCoords = "";      View rl;        @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            // инициализируем нужные элементы          tv = (TextView) findViewById(R.id.tv);          rl = (View) findViewById(R.id.rl);          // назначаем слушателя касания для Layout-а          rl.setOnTouchListener(this);      }        @Override      public boolean onTouch(View v, MotionEvent event) {          // получаем координаты касания          mX = event.getX();          mY = event.getY();            // переключатель в зависимости от типа события          switch (event.getAction()) {              case MotionEvent.ACTION\_DOWN: // нажатие              case MotionEvent.ACTION\_MOVE: // движение                  mCoords = "Coords: x = " + mX + ", y = " + mY;                  break;              case MotionEvent.ACTION\_UP: // отпускание              case MotionEvent.ACTION\_CANCEL:                  // ничего не делаем                  break;          }          tv.setText(mCoords);          return true;      }  } |

Комментарии есть в самом коде. Тут у нас всего два метода: **onCreate** и **onTouch**. В первом инициализируем нужные элементы , сеттим layout и обработчик касания для него — onTouch.

Метод onTouch() на входе получает объект типа **MotionEvent**, из которого мы извлекаем координаты положения касания соответствующим методом **getX()** или **getY()**. Также методом **getAction()** мы получаем тип события:

**ACTION\_DOWN** — простое нажатие  
**ACTION\_MOVE** — движение по элементу  
**ACTION\_UP** — поднятие пальца от экрана  
**ACTION\_CANCEL** — движение прервано (согласно документации) — поймать его непросто.

Нас интересуют, прежде всего, первые два. Здесь мы формируем строку из координат и выводим в нашем TextView.

Выглядит все примерно так.

