Создание контекстного меню

Контекстное меню появляется, когда мы долго жмем на каком-то пункте списка или любом элементе экрана (если, конечно, для этого элемента добавлен обработчик). В этом уроке мы создадим небольшое контекстное меню, которое будет появляться при «затяжном» нажатии на строку текста. В меню будет три опции выбора размера текста данного текста. После выбора какого-то пункта контекстное меню должно исчезнуть, а размеру текста должно присвоиться выбранное значение (22, 24 или 26).

Создаем проект (здесь я его назвал «First») и добавляем Layout типа RelativeLayout с одним текстовым TextView. Визуально это будет выглядеть так:



А код данного Layout-а такой (файл activity\_main.xml):

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

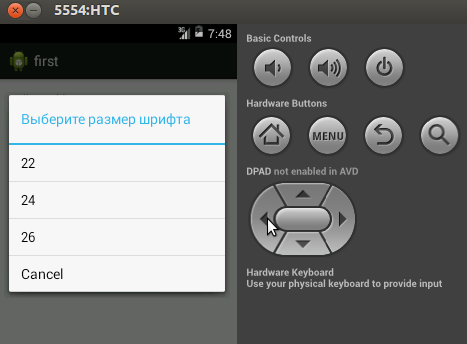
Как видите, вместо того, чтобы прописывать напрямую содержимое текста, мы используем строковый ресурс @string/hello\_world. Как вы уже знаете, это — правильный подход.

Теперь рассмотрим код MainActivity.java, в котором, собственно, и разыгрывается все представление.

|  |
| --- |
| package com.example.first;    import android.os.Bundle;  import android.app.Activity;  import android.view.ContextMenu;  import android.view.ContextMenu.ContextMenuInfo;  import android.view.MenuItem;  import android.view.View;  import android.widget.TextView;    public class MainActivity extends Activity {    final int ITEM\_SIZE\_22 = 22;  final int ITEM\_SIZE\_24 = 24;  final int ITEM\_SIZE\_26 = 26;  final int ITEM\_CANCEL = 0;    TextView txt;    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    txt = (TextView) findViewById(R.id.text\_field);  registerForContextMenu(txt);  }    /\*\*  \* Метод, вызывающийся при построении контекстного меню  \*/  @Override  public void onCreateContextMenu(ContextMenu menu, View v,  ContextMenuInfo menuInfo) {  menu.setHeaderTitle("Выберите размер шрифта");  menu.add(0, ITEM\_SIZE\_22, 0, "22");  menu.add(0, ITEM\_SIZE\_24, 0, "24");  menu.add(0, ITEM\_SIZE\_26, 0, "26");  menu.add(0, ITEM\_CANCEL, 0, "Cancel");  }    /\*\*  \* Метод, осуществляющий обработку выбора пункта меню  \*/  @Override  public boolean onContextItemSelected(MenuItem item) {  switch (item.getItemId()) {  case ITEM\_SIZE\_22:  txt.setTextSize(22);  txt.setText("Размер текста 22");  break;  case ITEM\_SIZE\_24:  txt.setTextSize(24);  txt.setText("Размер текста 24");  break;  case ITEM\_SIZE\_26:  txt.setTextSize(26);  txt.setText("Размер текста 26");  break;  case ITEM\_CANCEL:  closeContextMenu();  break;  }    return true;  }  } |

Пояснения. В самом начале класса мы объявили несколько констант, которые будут использоваться далее при создании пунктов контекстного меню (их значение будет использоваться как идентификатор отдельного пункта, чтобы мы могли понять, какая же опция выбрана).

В методе onCreate мы добавляем «регистрируем» для нашего текста контекстное меню (с помощью метода registerForContextMenu). Метод onCreateContextMenu всегда вызывается перед созданием контекстного меню, и здесь мы можем задать заголовок для него методом setHeaderTitle.



Метод [onContextItemSelected](http://developer.android.com/reference/android/app/Activity.html" \l "onContextItemSelected(android.view.MenuItem)" \t "_blank) вызывается при выборе какого-либо пункта (в качестве параметра передается сам пункт). С помощью конструкции switch — case мы проверяем, какой именно пункт был выбран и в зависимости от этого задаем размер для нашего текста (setText).

Также я добавил в меню кнопку Cancel, при нажатии на которую мы закрываем программно наше контекстное меню методом closeContextMenu().

