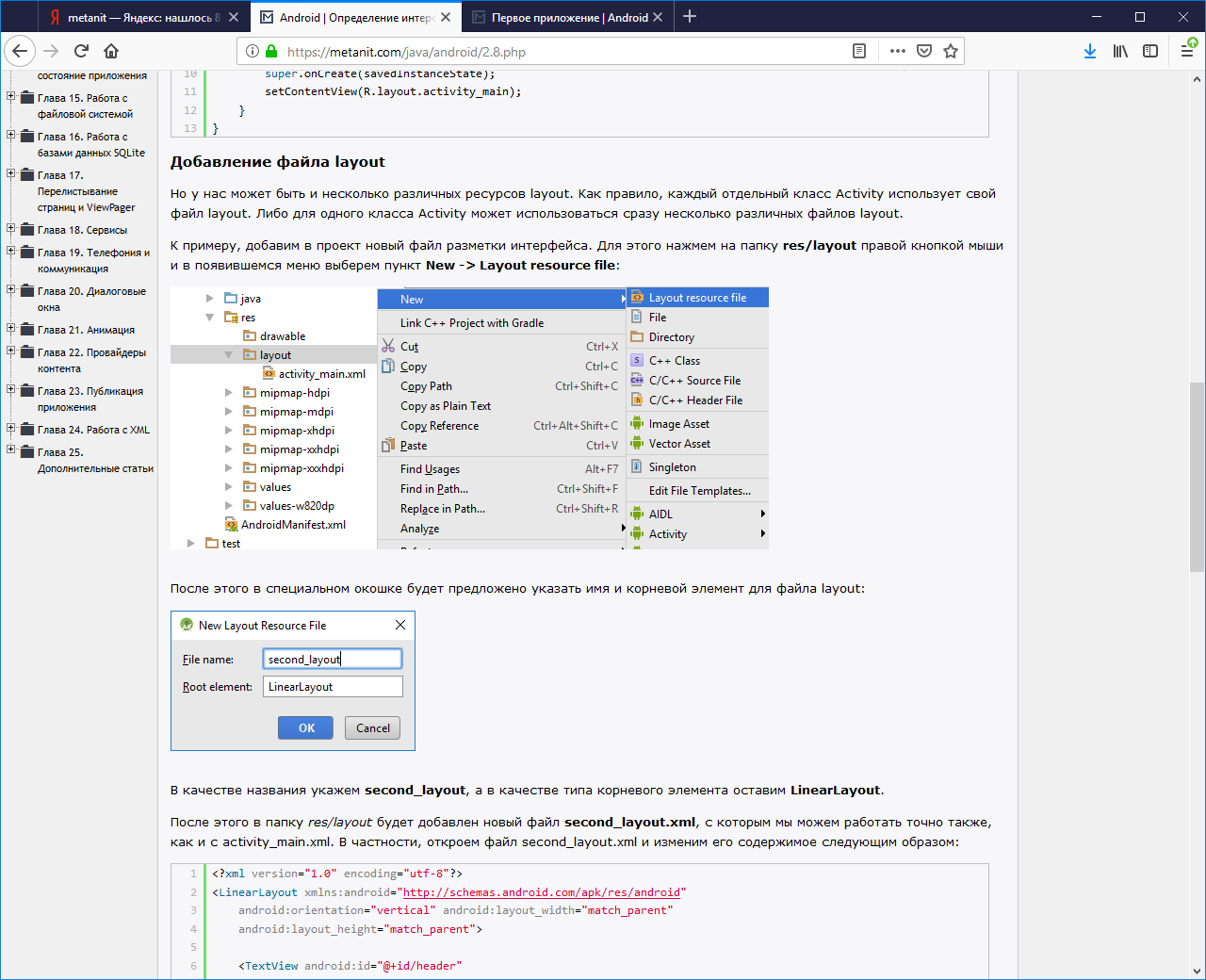
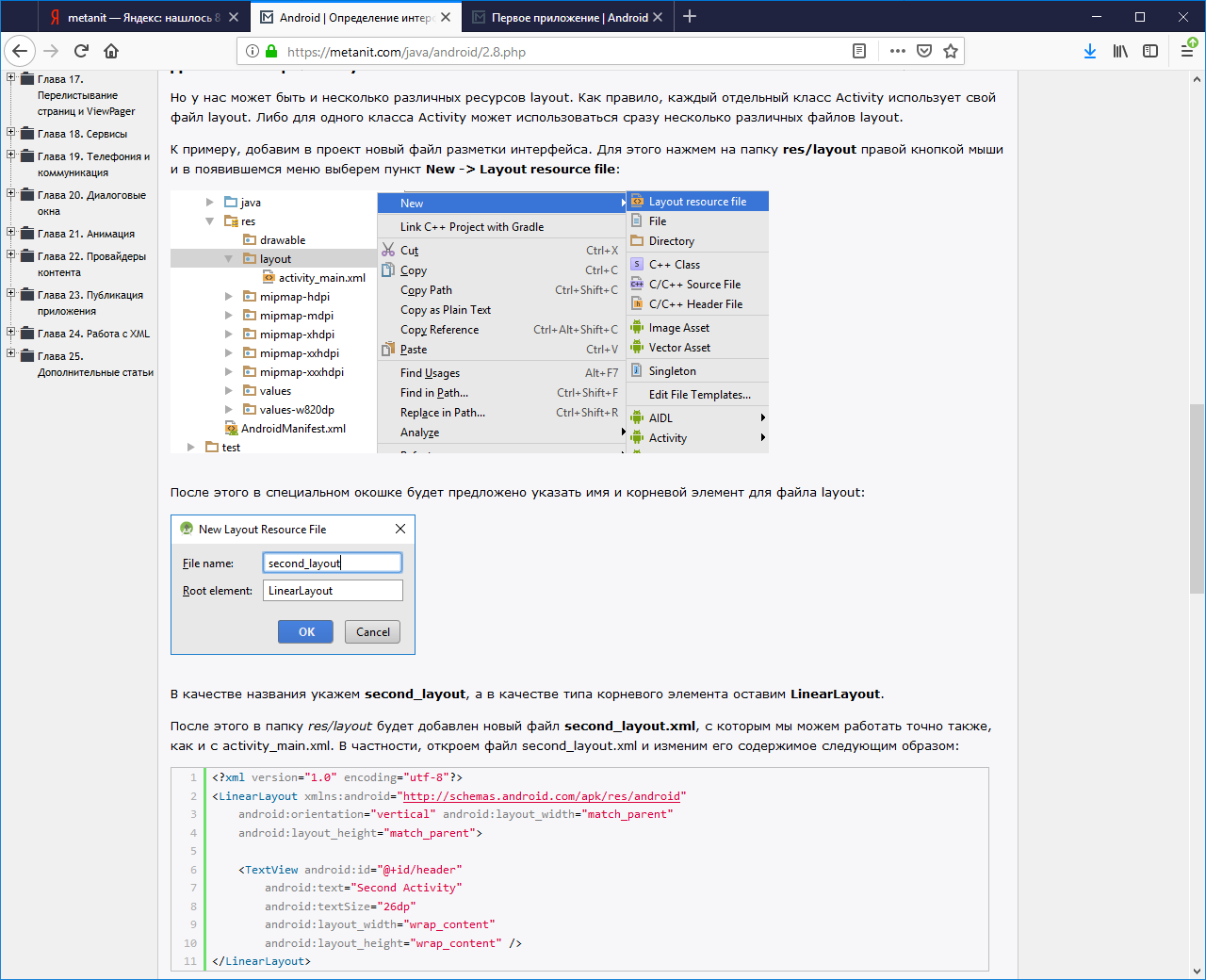
**Добавление файла Layout**

У нас может быть и несколько различных ресурсов layout. Как правило, каждый отдельный класс Activity использует свой файл layout. Либо для одного класса Activity может использоваться сразу несколько различных файлов layout.

К примеру, добавим в проект новый файл разметки интерфейса. Для этого нажмем на папку res/layout правой кнопкой мыши и в появившемся меню выберем пункт New -> Layout resource file:



После этого в специальном окошке будет предложено указать имя и корневой элемент для файла layout:



В качестве названия укажем second\_layout, а в качестве типа корневого элемента оставим LinearLayout.

После этого в папку res/layout будет добавлен новый файл second\_layout.xml, с которым мы можем работать точно также, как и с activity\_main.xml. В частности, откроем файл second\_layout.xml и изменим его содержимое следующим образом:

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

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

    android:orientation="vertical" android:layout\_width="match\_parent"

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

    <TextView android:id="@+id/header"

        android:text="Second Activity"

        android:textSize="26dp"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content" />

</LinearLayout>

**Получение элементов в коде и их идентификаторы**

Кроме текста, ширины и высоты текстовое поле устанавливает еще один важный атрибут - id. Знак + в записи android:id="@+id/header" означает, что если для элемента не определен id со значением header, то его следует определить.

Чтобы использовать этот файл в качестве основного интерфейса, перейдем к MainActivity и изменим ее код:

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        // устанавливаем в качестве интерфейса файл second\_layout.xml

        setContentView(R.layout.second\_layout);

        // получаем элемент textView

        TextView textView = (TextView) findViewById(R.id.header);

        // переустанавливаем у него текст

        textView.setText("Hello Android 7!");

    }

}

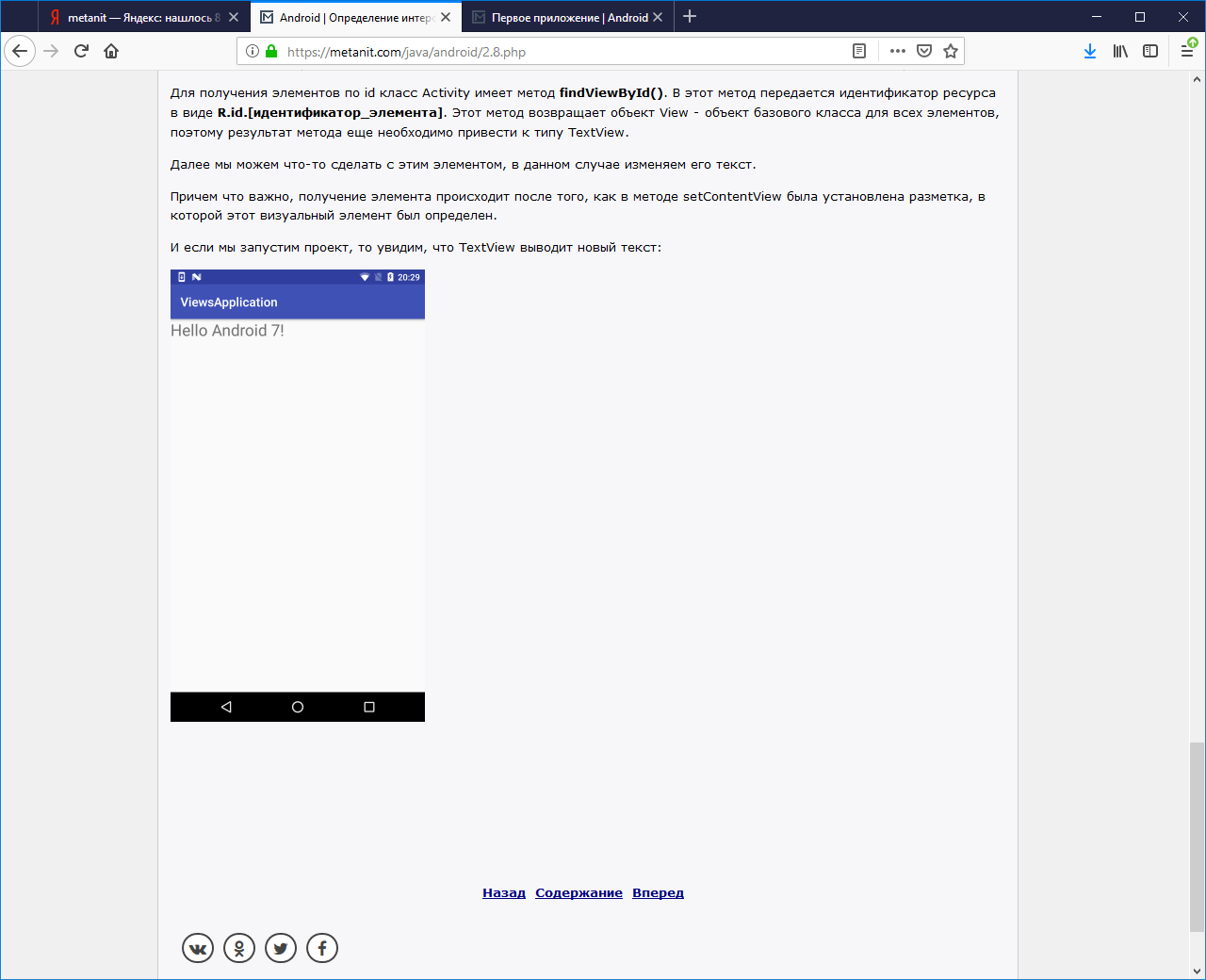
С помощью метода **setContentView()** устанавливается разметка из файла second\_layout.xml.

Другой важный момент, который стоит отметить - получение визуального элемента TextView. Так как в его коде мы определили атрибут android:id, то через этот id мы можем его получить.

Для получения элементов по id класс Activity имеет метод **findViewById()**. В этот метод передается идентификатор ресурса в виде **R.id.[идентификатор\_элемента]**. Этот метод возвращает объект View - объект базового класса для всех элементов, поэтому результат метода еще необходимо привести к типу TextView.

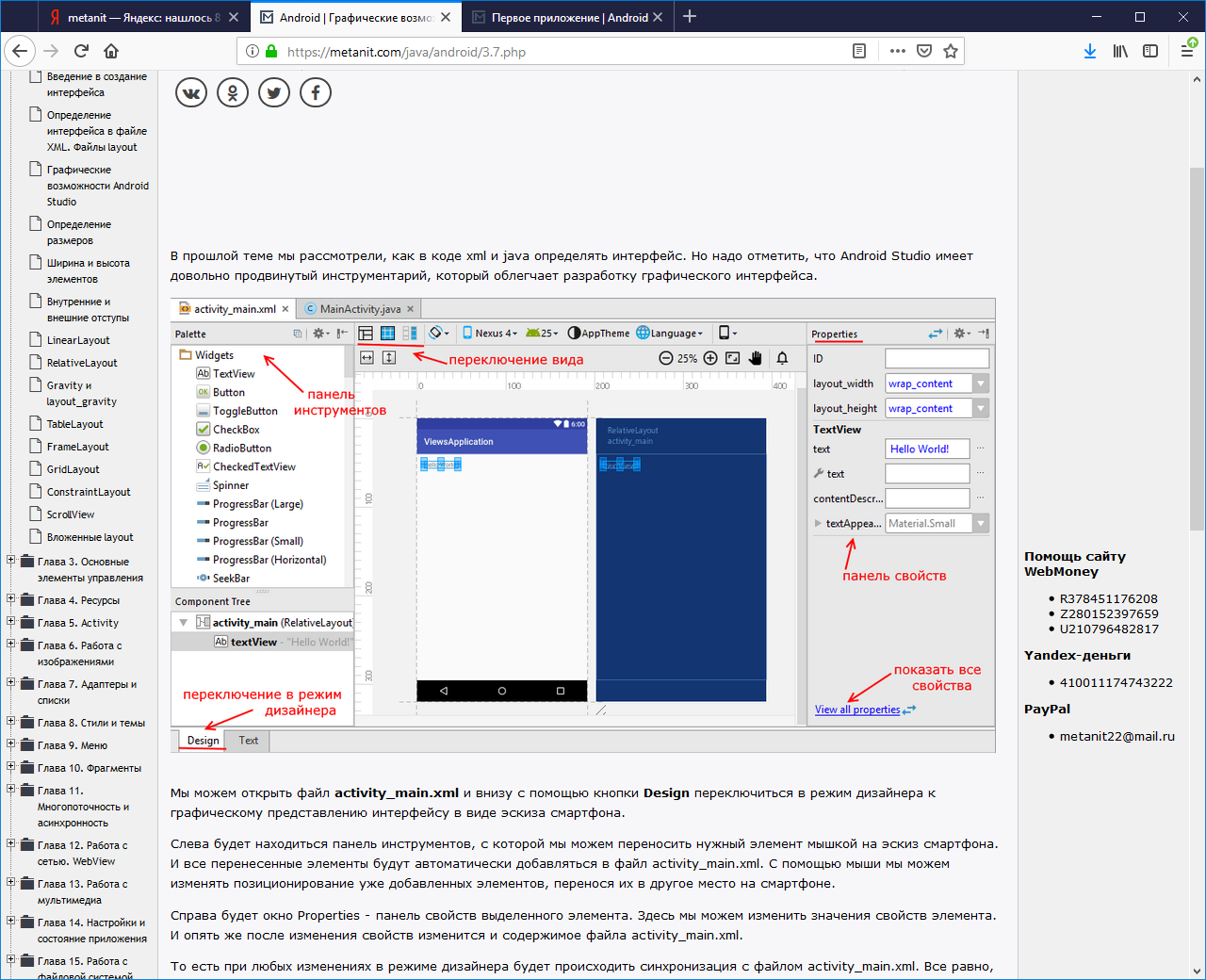
Причем что важно, получение элемента происходит после того, как в методе **setContentView** была установлена разметка, в которой этот визуальный элемент был определен.

И если мы запустим проект, то увидим, что TextView выводит новый текст:



**Графические возможности Android Studio**

Надо отметить, что Android Studio имеет довольно продвинутый инструментарий, который облегчает разработку графического интерфейса.



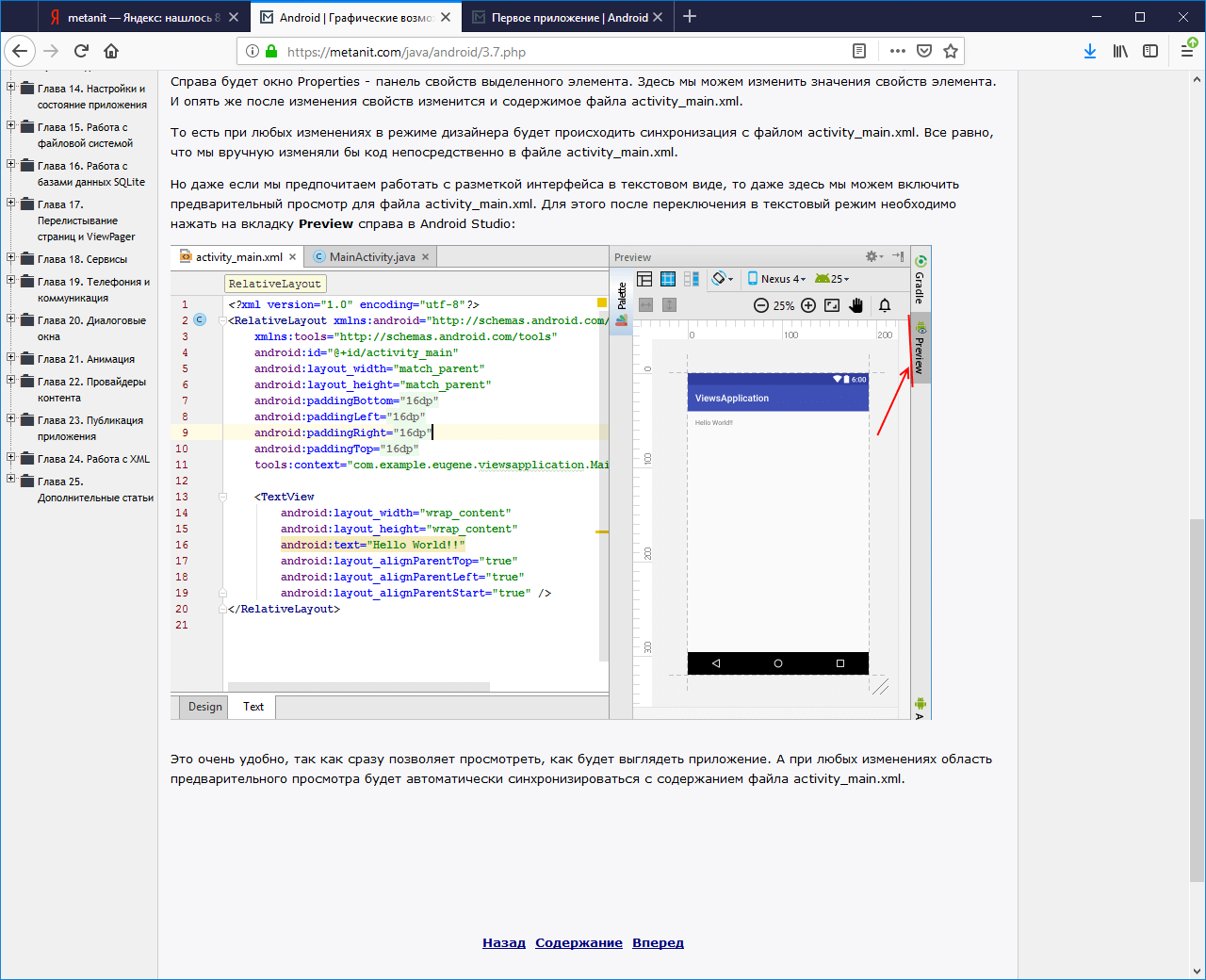
Мы можем открыть файл activity\_main.xml и внизу с помощью кнопки Design переключиться в режим дизайнера к графическому представлению интерфейсу в виде эскиза смартфона.

Слева будет находиться панель инструментов, с которой мы можем переносить нужный элемент мышкой на эскиз смартфона. И все перенесенные элементы будут автоматически добавляться в файл activity\_main.xml. С помощью мыши мы можем изменять позиционирование уже добавленных элементов, перенося их в другое место на смартфоне.

Справа будет окно Properties - панель свойств выделенного элемента. Здесь мы можем изменить значения свойств элемента. И опять же после изменения свойств изменится и содержимое файла activity\_main.xml.

То есть при любых изменениях в режиме дизайнера будет происходить синхронизация с файлом activity\_main.xml. Все равно, что мы вручную изменяли бы код непосредственно в файле activity\_main.xml.

Но даже если мы предпочитаем работать с разметкой интерфейса в текстовом виде, то даже здесь мы можем включить предварительный просмотр для файла activity\_main.xml. Для этого после переключения в текстовый режим необходимо нажать на вкладку Preview справа в Android Studio:



Это очень удобно, так как сразу позволяет просмотреть, как будет выглядеть приложение. А при любых изменениях область предварительного просмотра будет автоматически синхронизироваться с содержанием файла activity\_main.xml.