**LinearLayout**

Контейнер LinearLayout представляет объект ViewGroup, который упорядочивает все дочерние элементы в одном направлении: по горизонтали или по вертикали. Все элемены расположены один за другим. Направление разметки указывается с помощью атрибута **android:orientation.**

Если, например, ориентация разметки вертикальная (**android:orientation="vertical"**), то все элементы располагаются в столбик - по одному элементу на каждой строке. Если ориентация горизонтальная (**android:orientation="horizontal"**), то элементы располагаются в одну строку. Например, расположим элементы в горизонтальный ряд:

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

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

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="horizontal" >

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Hello"

        android:textSize="26sp" />

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Android"

        android:textSize="26sp" />

    <TextView

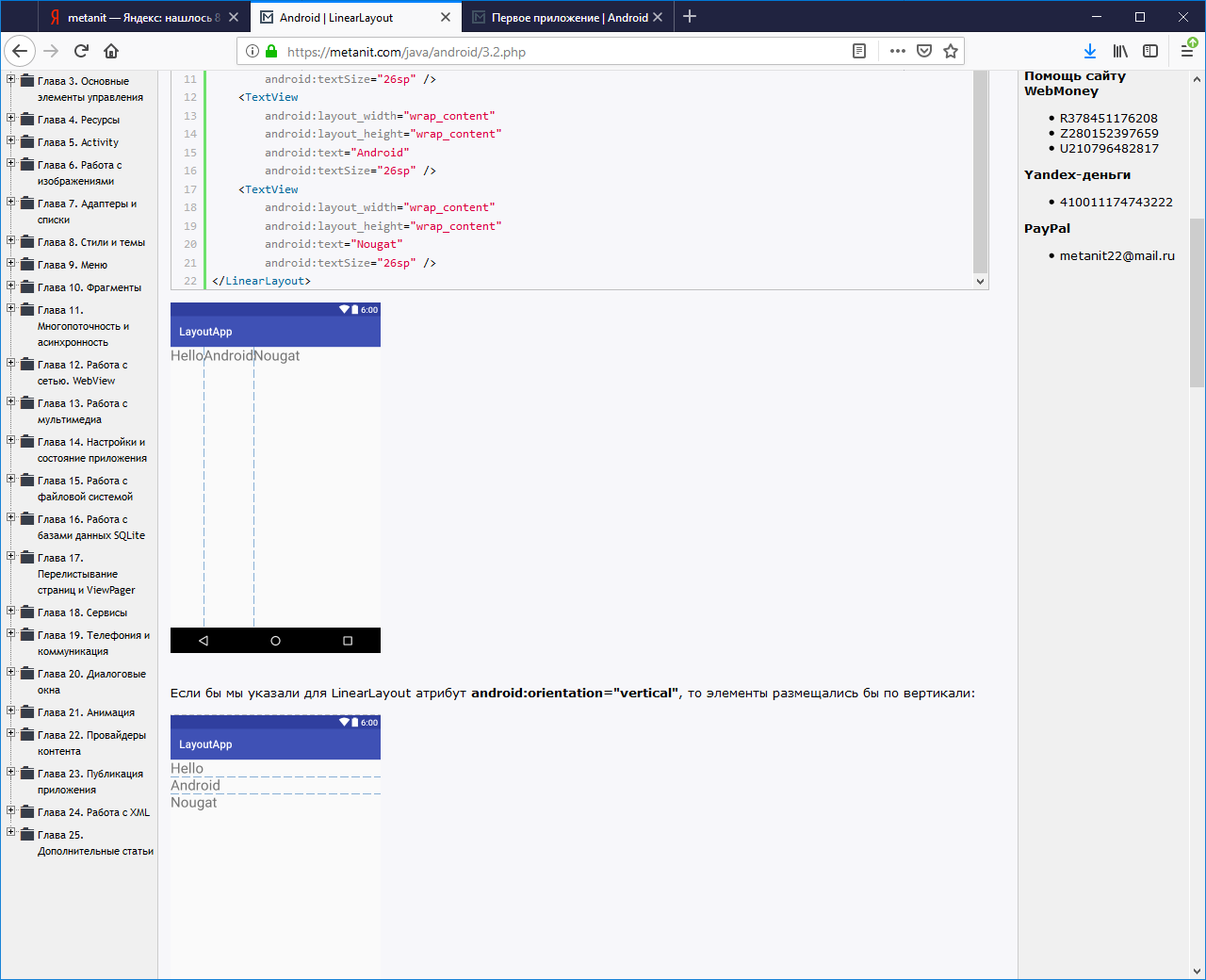
        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

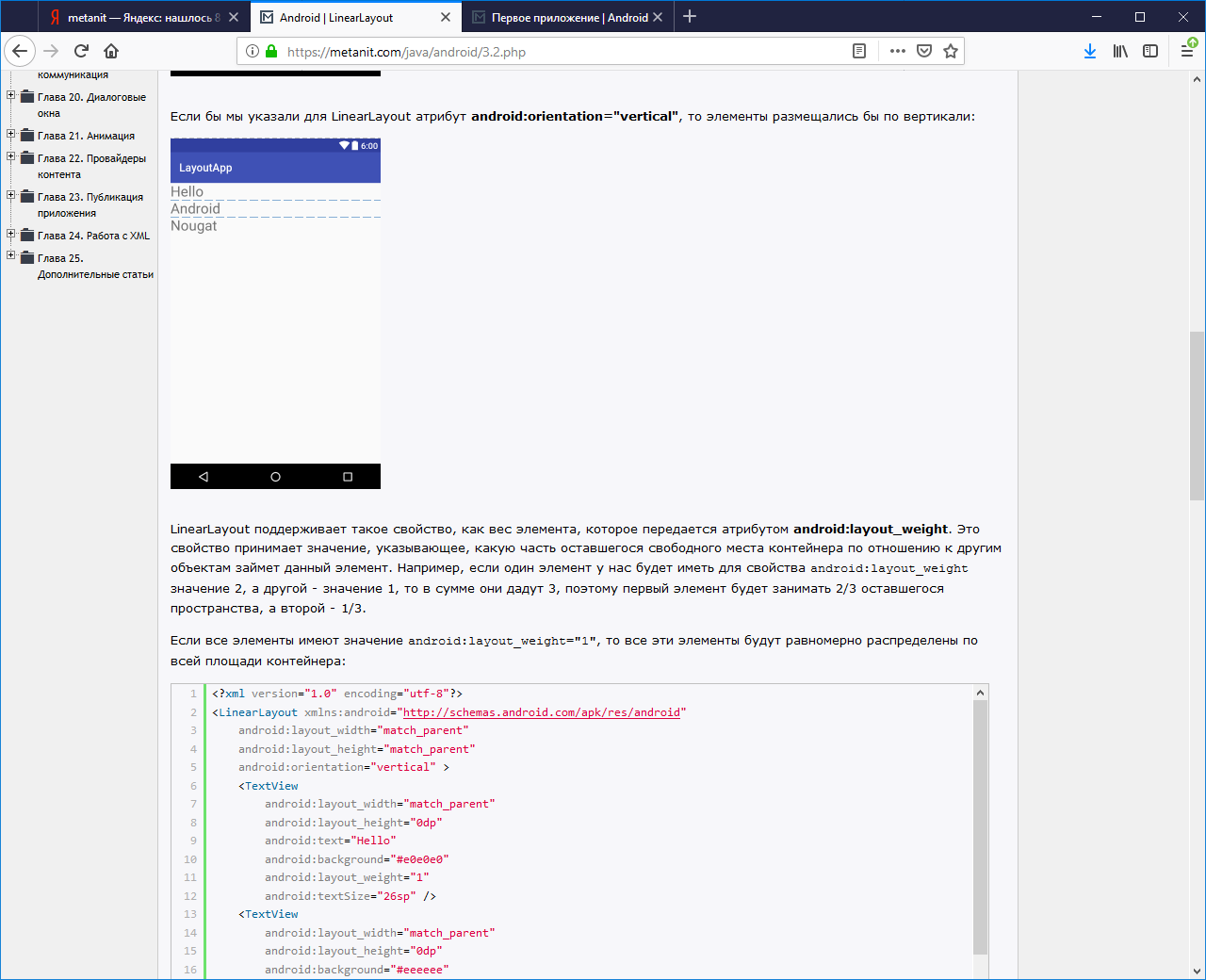
        android:text="Nougat"

        android:textSize="26sp" />

</LinearLayout>



Если бы мы указали для LinearLayout атрибут android:orientation="vertical", то элементы размещались бы по вертикали:



LinearLayout поддерживает такое свойство, как вес элемента, которое передается атрибутом **android:layout\_weight**. Это свойство принимает значение, указывающее, какую часть оставшегося свободного места контейнера по отношению к другим объектам займет данный элемент. Например, если один элемент у нас будет иметь для свойства android:layout\_weight значение 2, а другой - значение 1, то в сумме они дадут 3, поэтому первый элемент будет занимать 2/3 оставшегося пространства, а второй - 1/3.

Если все элементы имеют значение **android:layout\_weight="1",** то все эти элементы будут равномерно распределены по всей площади контейнера:

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

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

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical" >

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:text="Hello"

        android:background="#e0e0e0"

        android:layout\_weight="1"

        android:textSize="26sp" />

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:background="#eeeeee"

        android:text="Android"

        android:layout\_weight="1"

        android:textSize="26sp" />

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:text="Nougat"

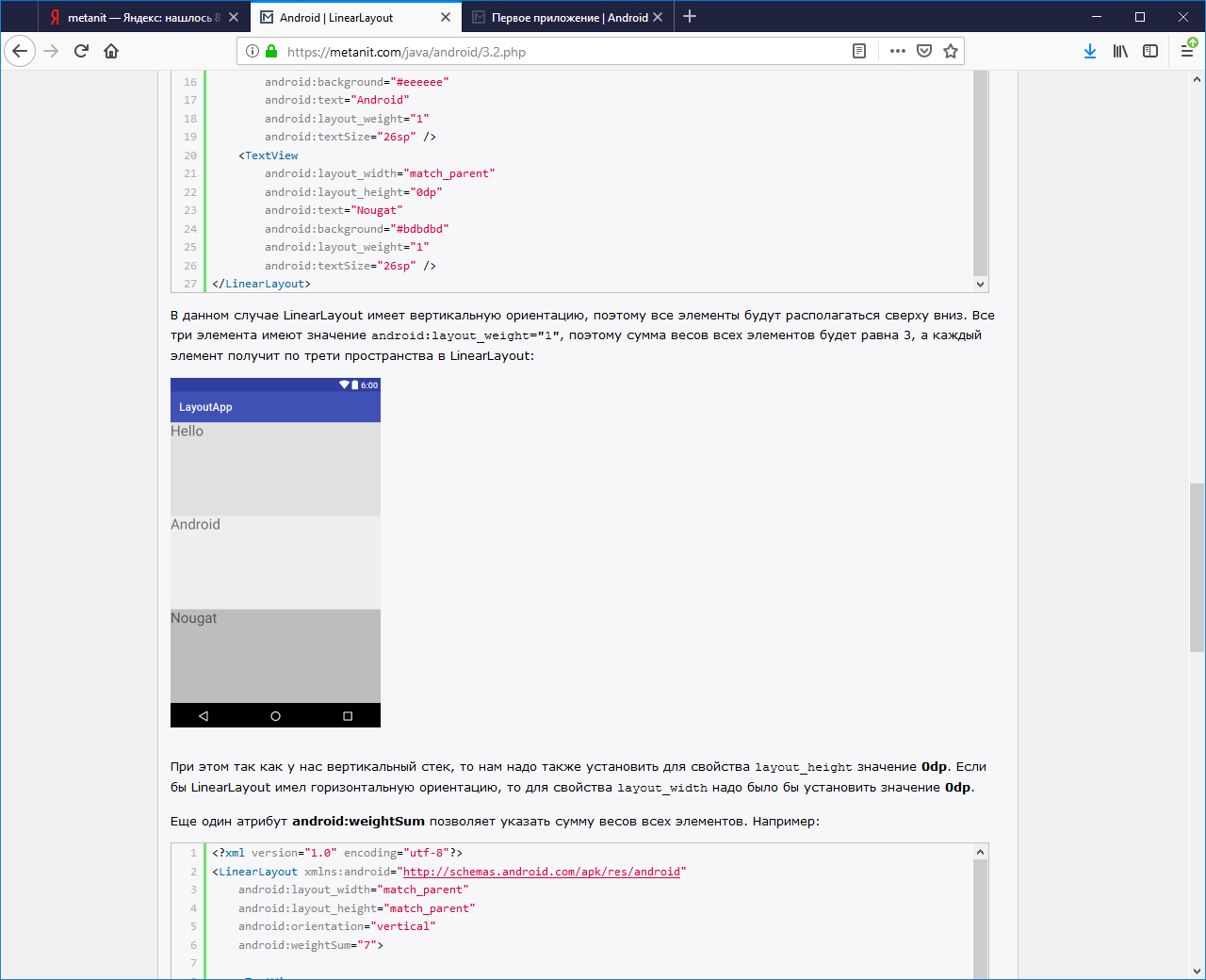
        android:background="#bdbdbd"

        android:layout\_weight="1"

        android:textSize="26sp" />

</LinearLayout>

В данном случае LinearLayout имеет вертикальную ориентацию, поэтому все элементы будут располагаться сверху вниз. Все три элемента имеют значение android:layout\_weight="1", поэтому сумма весов всех элементов будет равна 3, а каждый элемент получит по трети пространства в LinearLayout:



При этом так как у нас вертикальный стек, то нам надо также установить для свойства layout\_height значение 0dp. Если бы LinearLayout имел горизонтальную ориентацию, то для свойства layout\_width надо было бы установить значение 0dp.

Еще один атрибут **android:weightSum** позволяет указать сумму весов всех элементов. Например:

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

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

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical"

    android:weightSum="7">

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:text="Hello"

        android:background="#e0e0e0"

        android:layout\_weight="1"

        android:textSize="26sp" />

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:background="#eeeeee"

        android:text="Android"

        android:layout\_weight="3"

        android:textSize="26sp" />

    <TextView

        android:layout\_width="match\_parent"

        android:layout\_height="0dp"

        android:text="Nougat"

        android:background="#bdbdbd"

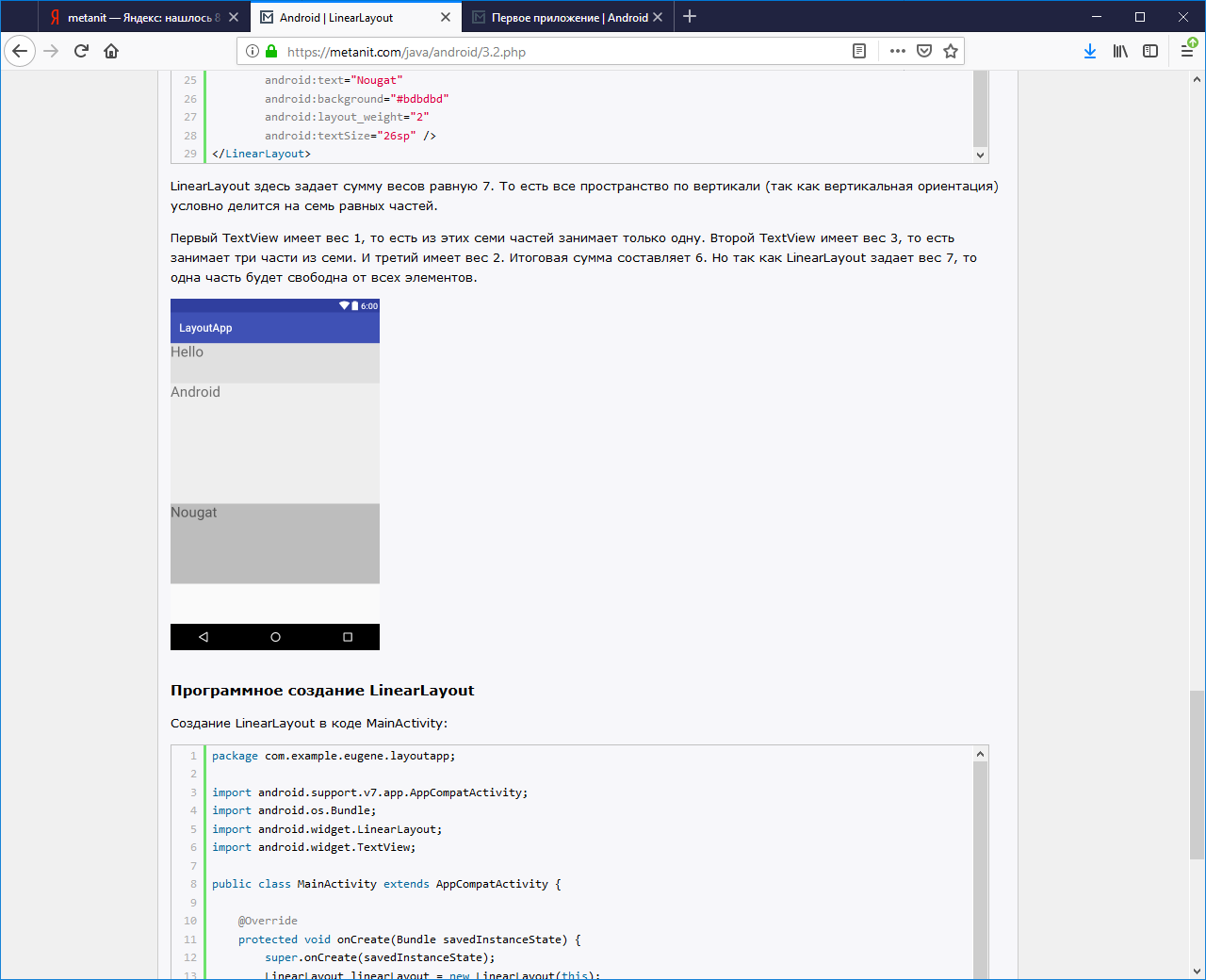
        android:layout\_weight="2"

        android:textSize="26sp" />

</LinearLayout>

LinearLayout здесь задает сумму весов равную 7. То есть все пространство по вертикали (так как вертикальная ориентация) условно делится на семь равных частей.

Первый TextView имеет вес 1, то есть из этих семи частей занимает только одну. Второй TextView имеет вес 3, то есть занимает три части из семи. И третий имеет вес 2. Итоговая сумма составляет 6. Но так как LinearLayout задает вес 7, то одна часть будет свободна от всех элементов.



**Программное создание LinearLayout**

Создание LinearLayout в коде MainActivity:

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.LinearLayout;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        LinearLayout linearLayout = new LinearLayout(this);

        linearLayout.setOrientation(LinearLayout.VERTICAL);

        TextView textView1 = new TextView(this);

        textView1.setText("Hello");

        linearLayout.addView(textView1, new LinearLayout.LayoutParams

                (LinearLayout.LayoutParams.WRAP\_CONTENT, LinearLayout.LayoutParams.WRAP\_CONTENT));

        TextView textView2 = new TextView(this);

        textView2.setText("Android");

        linearLayout.addView(textView2, new LinearLayout.LayoutParams

                (LinearLayout.LayoutParams.WRAP\_CONTENT, LinearLayout.LayoutParams.WRAP\_CONTENT));

        setContentView(linearLayout);

    }

}