**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

**Дисциплина: Программирование для мобильных устройств**

**Тема:** Создание первого Android-приложения.

Использование различных классов макетов.

**Выполнил(а): студент(ка) группы \_\_211-722\_**

Дерендяев Дмитрий Сергеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** 10.10.2023\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2023**

# Введение

**Целью** лабораторной работы является знакомство со средой разработки Android Studio, создание первого Android-приложения, получение навыков создания различных классов компоновок пользовательского интерфейса.

# Реализация различных компоновок

Для всех реализаций мобильного приложения будет использован одинаковый класс MainActivity.java(figure 1):

Figure

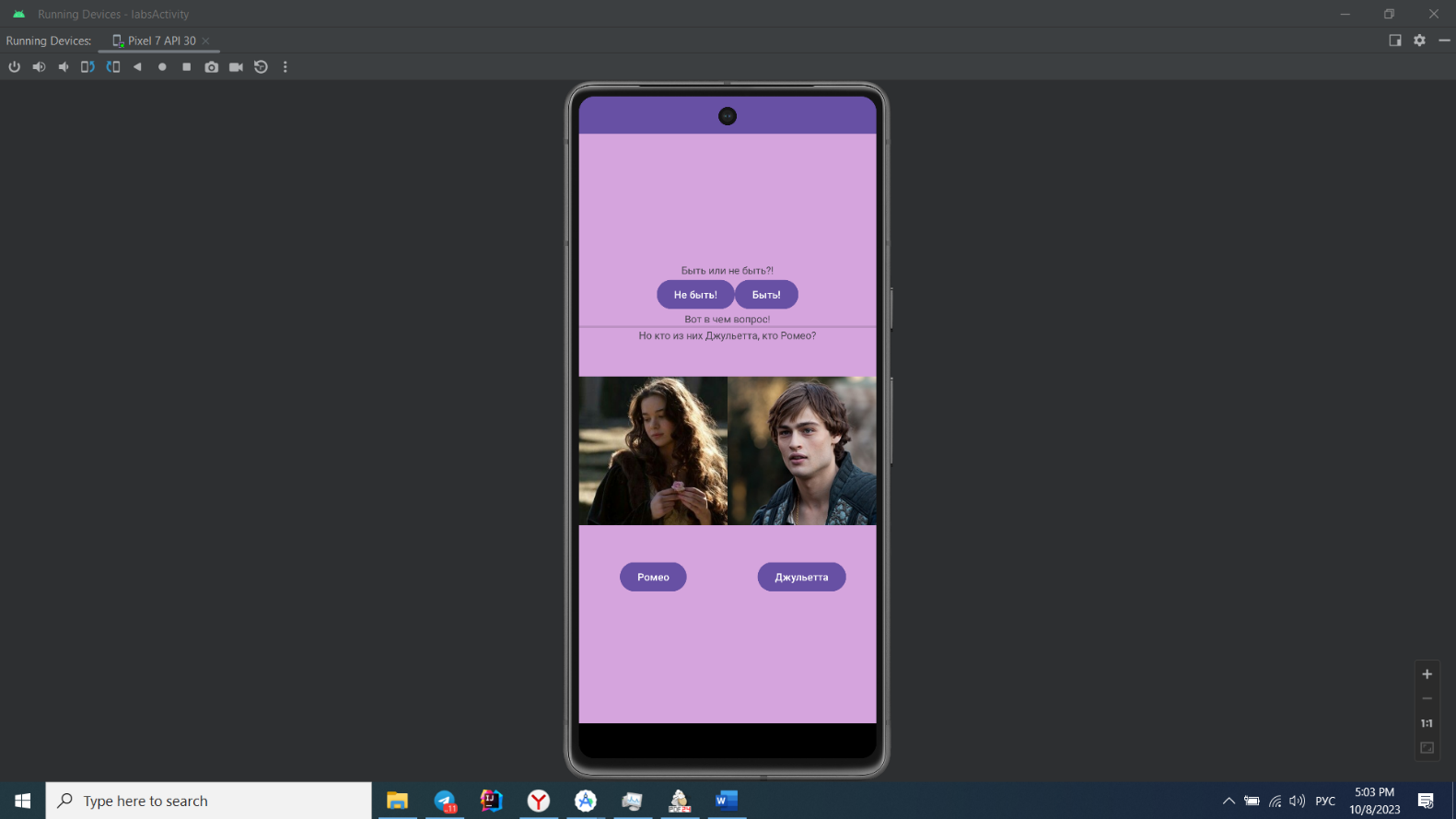
1. package com.example.labsactivity;  
     
   import androidx.appcompat.app.AppCompatActivity;  
     
   import android.os.Bundle;  
     
   public class MainActivity extends AppCompatActivity {  
     
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.*activity\_main*);  
    }  
   }

В этой лабораторной работе мы будем изменять только activity\_main.xml, чтобы применить различные компоновки.

## LinearLayout

Код с использованием LinearLayout выглядит так:

1. <?xml version="1.0" encoding="utf-8"?>  
   <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    tools:context=".MainActivity">  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:background="#D5A4DD"  
    android:gravity="center"  
    android:orientation="vertical">  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView1" />  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center"  
    android:orientation="horizontal">  
     
    <Button  
    android:id="@+id/button1"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button1" />  
     
    <Button  
    android:id="@+id/button2"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button2" />  
    </LinearLayout>  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView2" />  
     
     
    <View  
    android:id="@+id/divider"  
    android:layout\_width="match\_parent"  
    android:layout\_height="3dp"  
    android:background="?android:attr/listDivider"  
    />  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView3" />  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center">  
     
    <ImageView  
    android:id="@+id/imageView"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:src="@drawable/juliette" />  
     
    <ImageView  
    android:id="@+id/imageView2"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:src="@drawable/romeo"  
    android:contentDescription="TODO" />  
    </LinearLayout>  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center"  
    android:orientation="horizontal">  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center"  
    android:layout\_weight="1">  
    <Button  
    android:id="@+id/button3"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button3" />  
    </LinearLayout>  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center"  
    android:layout\_weight="1">  
     
    <Button  
    android:id="@+id/button4"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button4" />  
     
    </LinearLayout>  
     
     
    </LinearLayout>  
     
    </LinearLayout>  
     
   </androidx.constraintlayout.widget.ConstraintLayout>

Результат работы приложения изображен на Figure 2.

Figure

## ConstraintLayout

Во втором варианте код реализован с использованием ConstraintLayout

1. <?xml version="1.0" encoding="utf-8"?>  
   <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    tools:context=".MainActivity">  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:background="#D5A4DD"  
    android:gravity="center"  
    android:orientation="vertical">  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView1" />  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center"  
    android:orientation="horizontal">  
     
    <Button  
    android:id="@+id/button1"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button1" />  
     
    <Button  
    android:id="@+id/button2"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button2" />  
    </LinearLayout>  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView2" />  
     
    <View  
    android:id="@+id/divider"  
    android:layout\_width="match\_parent"  
    android:layout\_height="3dp"  
    android:background="?android:attr/listDivider"  
    />  
     
    <TextView  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/textView3" />  
     
    <LinearLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center">  
     
    <ImageView  
    android:id="@+id/imageView"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:src="@drawable/juliette" />  
     
    <ImageView  
    android:id="@+id/imageView2"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:src="@drawable/romeo"  
    android:contentDescription="TODO" />  
    </LinearLayout>  
     
    <!-- Реализация кнопок 3 и 4 с использованием ConstraintLayout -->  
    <androidx.constraintlayout.widget.ConstraintLayout  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:gravity="center">  
     
    <Button  
    android:id="@+id/button3"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button3"  
    app:layout\_constraintStart\_toStartOf="parent"  
    app:layout\_constraintEnd\_toStartOf="@id/guideline"  
    app:layout\_constraintTop\_toTopOf="parent"  
    app:layout\_constraintBottom\_toBottomOf="parent" />  
     
    <Button  
    android:id="@+id/button4"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/button4"  
    app:layout\_constraintStart\_toEndOf="@id/guideline"  
    app:layout\_constraintEnd\_toEndOf="parent"  
    app:layout\_constraintTop\_toTopOf="parent"  
    app:layout\_constraintBottom\_toBottomOf="parent" />  
     
    <androidx.constraintlayout.widget.Guideline  
    android:id="@+id/guideline"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:orientation="vertical"  
    app:layout\_constraintGuide\_end="205dp" />  
     
    </androidx.constraintlayout.widget.ConstraintLayout>  
     
    </LinearLayout>  
     
   </androidx.constraintlayout.widget.ConstraintLayout>

Теперь кнопки button3 и button4 – выравниваются под картинками по центру используя специальные теги привязки. Результат при этом не изменился и все так же получается, как на Figure 2.

# Контрольные вопросы

1. Компоновка (layout) - это способ организации расположения элементов интерфейса в приложении Android.

2. Некоторые из типов компоновок в Android: LinearLayout, ConstraintLayout, RelativeLayout, FrameLayout, и другие.

3. LinearLayout организует дочерние элементы в строку или столбец, в зависимости от ориентации, которая может быть горизонтальной или вертикальной.

4. ConstraintLayout позволяет создавать сложные и гибкие компоновки, опираясь на ограничения между элементами.

5. RelativeLayout позволяет определять положение дочерних элементов относительно друг друга или родительского контейнера.

6. Атрибут android:layout\_height определяет высоту элемента в компоновке Android.

7. Значение match\_parent означает, что элемент будет занимать всю доступную высоту или ширину родительского контейнера.

8. Значение wrap\_content означает, что элемент будет занимать только столько места, сколько необходимо для содержимого.

9. Атрибут android:gravity используется для управления выравниванием содержимого внутри элемента компоновки.

10. Атрибут android:id присваивает уникальный идентификатор элементу, который можно использовать для обращения к нему из Java-кода или XML-макета.

11. Запись @+id в атрибуте android:id создает новый идентификатор, который добавляется в список доступных идентификаторов в проекте.

12. На платформе Android для указания размеров представлений используются единицы измерения, такие как dp (плотность независимых пикселей), sp (шкала независимых пикселей), px (пиксели), mm (миллиметры) и др.