**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:** «Кнопки и обработка событий»

**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 с различными классами

кнопок, представленными на Android, а также обработкой событий.

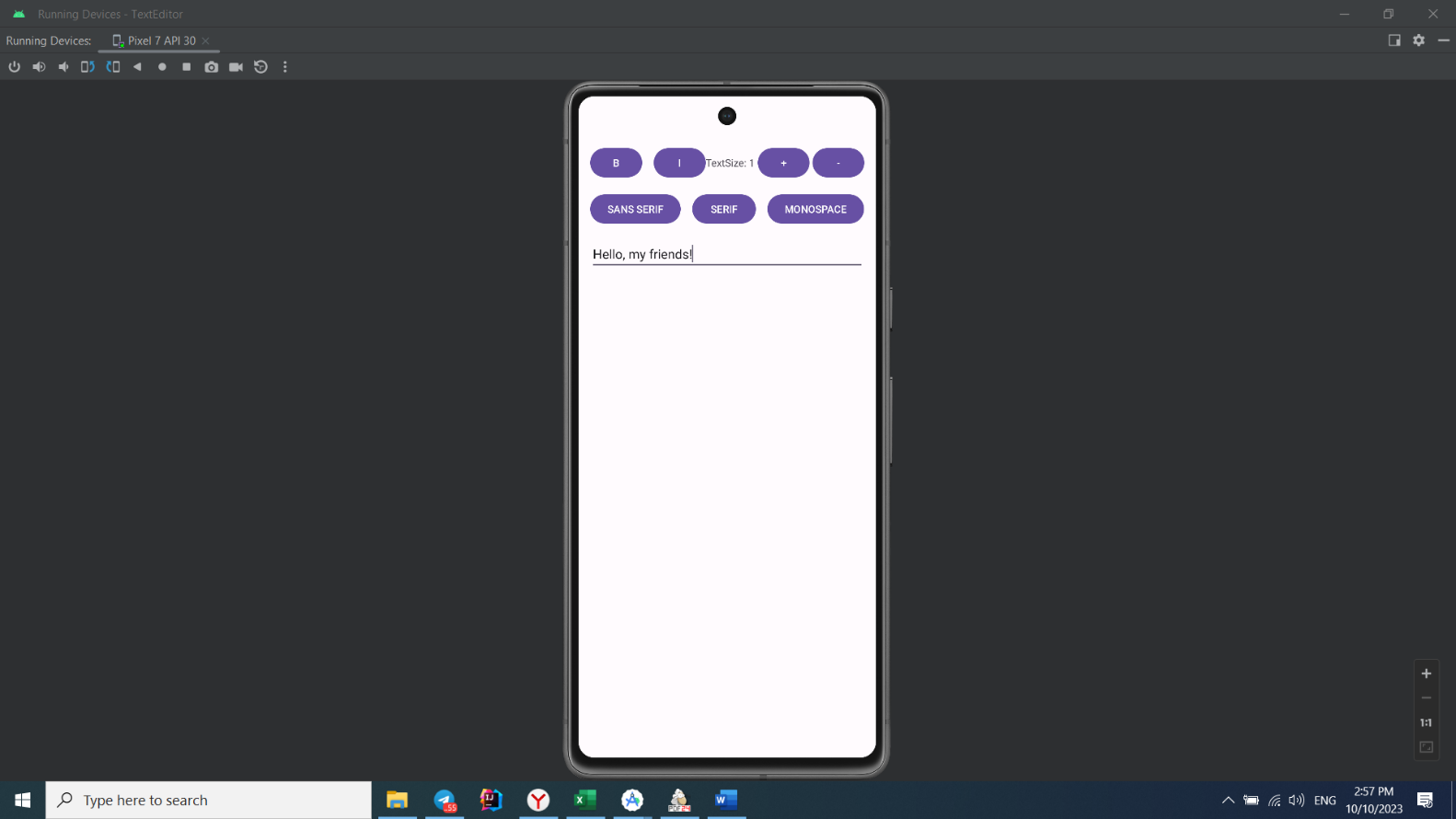
# Реализация activiti\_main.xml

В этом activity\_main.xml разместим названия всех кнопок и полей константно в главном файле:

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">  
     
    <RelativeLayout 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"  
    android:padding="16dp"  
    tools:context=".MainActivity">  
     
    <LinearLayout  
    android:id="@+id/linearLayout1"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:orientation="horizontal"  
    android:layout\_alignParentTop="true"  
    android:layout\_alignParentStart="true">  
     
    <Button  
    android:id="@+id/button\_b"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="B"  
    android:layout\_weight="1"/>  
     
    <Button  
    android:id="@+id/button\_i"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="I"  
    android:layout\_marginStart="16dp"  
    android:layout\_weight="1"/>  
     
    <TextView  
    android:id="@+id/sizeLabel"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="TextSize: " />  
     
    <TextView  
    android:id="@+id/textSizeValue"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="1" />  
     
    <Button  
    android:id="@+id/button\_plus"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="+"  
    android:layout\_marginStart="4dp"  
    android:layout\_weight="1"/>  
     
    <Button  
    android:id="@+id/button\_minus"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="-"  
    android:layout\_marginStart="4dp"  
    android:layout\_weight="1"/>  
    </LinearLayout>  
     
    <LinearLayout  
    android:id="@+id/linearLayout2"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:orientation="horizontal"  
    android:layout\_below="@id/linearLayout1"  
    android:layout\_alignParentStart="true"  
    android:layout\_marginTop="16dp">  
     
    <Button  
    android:id="@+id/button\_sans"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="SANS SERIF" />  
     
    <Button  
    android:id="@+id/button\_serif"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="SERIF"  
    android:layout\_marginStart="16dp" />  
     
    <Button  
    android:id="@+id/button\_monospace"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="MONOSPACE"  
    android:layout\_marginStart="16dp" />  
    </LinearLayout>  
      
    <EditText  
    android:id="@+id/edit\_text"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@id/linearLayout2"  
    android:layout\_marginTop="16dp"  
    android:hint="Enter text here..."  
    android:layout\_alignParentStart="true" />  
     
    </RelativeLayout>  
     
   </androidx.constraintlayout.widget.ConstraintLayout>

# ACTIvityMain.java

Для того, чтобы привести код в работоспособное состояние были добавлены кнопки и обработчики событий.

1. package com.example.texteditor;  
     
     
   import androidx.appcompat.app.AppCompatActivity;  
   import android.graphics.Typeface;  
   import android.os.Bundle;  
   import android.view.View;  
   import android.widget.Button;  
   import android.widget.EditText;  
   import android.widget.TextView;  
   public class MainActivity extends AppCompatActivity implements  
    View.OnClickListener {  
    private float mTextSize = 20;  
    private EditText mEdit;  
    private TextView tSize;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.*activity\_main*);  
    mEdit = findViewById(R.id.*edit\_text*);  
    tSize = findViewById(R.id.*textSizeValue*);  
    Button buttonB = findViewById(R.id.*button\_b*);  
    Button buttonI = findViewById(R.id.*button\_i*);  
    Button buttonSans = findViewById(R.id.*button\_sans*);  
   // \*\*\*  
    Button buttonSerif = findViewById(R.id.*button\_serif*);  
    Button buttonMonospace = findViewById(R.id.*button\_monospace*);  
     
    Button buttonPlus = findViewById(R.id.*button\_plus*);  
    Button buttonMinus = findViewById(R.id.*button\_minus*);  
     
    buttonB.setOnClickListener(this);  
    buttonI.setOnClickListener(this);  
    buttonSans.setOnClickListener(this);  
    buttonSerif.setOnClickListener(this);  
    buttonMonospace.setOnClickListener(this);  
     
    buttonPlus.setOnClickListener(this);  
    buttonMinus.setOnClickListener(this);  
     
     
    }  
     
     
    @Override  
    public void onClick(View v) {  
    int viewId = v.getId();  
    if (viewId == R.id.*button\_plus*) {  
    if (mTextSize < 72) {  
    mTextSize += 2;  
    mEdit.setTextSize(mTextSize);  
    tSize.setText(String.*format*("%.0f", mTextSize));  
    }  
    } else if (viewId == R.id.*button\_minus*) {  
    if (mTextSize > 18) {  
    mTextSize -= 2;  
    mEdit.setTextSize(mTextSize);  
    tSize.setText(String.*format*("%.0f", mTextSize));  
    }  
    } else if (viewId == R.id.*button\_b*) {  
    if (mEdit.getTypeface().getStyle() == Typeface.*ITALIC*)  
    mEdit.setTypeface(mEdit.getTypeface(), Typeface.*BOLD\_ITALIC*);  
    else if (mEdit.getTypeface().getStyle() == Typeface.*BOLD\_ITALIC*)  
    mEdit.setTypeface(mEdit.getTypeface(), Typeface.*ITALIC*);  
    else if (mEdit.getTypeface().getStyle() == Typeface.*BOLD*)  
    mEdit.setTypeface(Typeface.*create*(mEdit.getTypeface(), Typeface.*NORMAL*));  
    else mEdit.setTypeface(mEdit.getTypeface(), Typeface.*BOLD*);  
    } else if (viewId == R.id.*button\_i*) {  
    if (mEdit.getTypeface().getStyle() == Typeface.*BOLD*)  
    mEdit.setTypeface(mEdit.getTypeface(), Typeface.*BOLD\_ITALIC*);  
    else if (mEdit.getTypeface().getStyle() == Typeface.*BOLD\_ITALIC*)  
    mEdit.setTypeface(mEdit.getTypeface(), Typeface.*BOLD*);  
    else if (mEdit.getTypeface().getStyle() == Typeface.*ITALIC*)  
    mEdit.setTypeface(mEdit.getTypeface(), Typeface.*NORMAL*);  
    else mEdit.setTypeface(mEdit.getTypeface(), Typeface.*ITALIC*);  
    } else if (viewId == R.id.*button\_sans*) {  
    mEdit.setTypeface(Typeface.*SANS\_SERIF*);  
    } else if (viewId == R.id.*button\_serif*) {  
    mEdit.setTypeface(Typeface.*SERIF*);  
    } else if (viewId == R.id.*button\_monospace*) {  
    mEdit.setTypeface(Typeface.*MONOSPACE*);  
    }  
    }  
   }



# Контрольные вопросы

1. Интерфейс **View.OnClickListener** содержит один метод, который называется **onClick(View v)**. Этот метод вызывается при щелчке на представлении (**View**), для которого был установлен слушатель **OnClickListener**. В коде выше, в методе **MainActivity**, мы реализовали этот интерфейс и переопределили метод **onClick** для обработки нажатий на различные кнопки.
2. Метод **findViewById** используется для поиска и получения ссылки на представление (компонент интерфейса) по его уникальному идентификатору (ID), который указывается в файле макета XML. Этот метод позволяет вашей активности получать доступ к элементам пользовательского интерфейса и взаимодействовать с ними программно.
3. Метод **setTypeface** используется для установки шрифта текста в представлении (например, **TextView** или **EditText**). Он позволяет задать различные стили шрифта, такие как жирный (**Typeface.BOLD**), курсив (**Typeface.ITALIC**), обычный (**Typeface.NORMAL**) и т. д.
4. В метод **setTypeface** передаются два параметра: первый параметр - шрифт (**Typeface**), который определяет стиль шрифта, а второй параметр - стиль шрифта (например, **Typeface.BOLD**, **Typeface.ITALIC**).
5. Метод **setTextSize** используется для установки размера текста в представлении, таком как **TextView** или **EditText**. Он позволяет задавать размер текста в пикселях или в специальных единицах измерения, таких как **sp** (масштабируемые пиксели).
6. Метод **setText** используется для установки текстового содержимого в представлении, например, **TextView** или **EditText**. Он позволяет устанавливать строку или символы в представлении.
7. Метод **getText** используется для получения текстового содержимого из представления. Он возвращает объект **CharSequence**, который представляет текст в представлении.
8. Метод **getTypeface** используется для получения текущего шрифта текста в представлении, таком как **TextView** или **EditText**. Он возвращает объект **Typeface**, который представляет текущий шрифт.
9. Метод **setOnClickListener** используется для установки обработчика кликов на представление (**View**). Когда пользователь кликает на представление, вызывается метод **onClick**, который был установлен через этот метод.
10. Метод **setContentView** используется для установки макета (XML-файла) как содержимого активности. Он определяет, какие элементы пользовательского интерфейса будут отображаться на экране активности. Этот метод обычно вызывается в методе **onCreate** активности для инициализации пользовательского интерфейса.