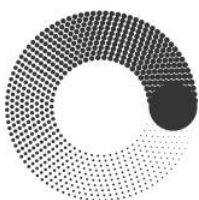


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**Факультет информационных технологий
Кафедра Информатики и информационных технологий**

направление подготовки 09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

Дисциплина: Программирование для мобильных устройств

Тема: Кнопки и обработка событий

Выполнил: студент группы 221-3710

Мелихов Владимир Семёнович

Проверил:

(Оценка)

Дата, подпись _____

(Дата)

(Подпись)

Замечания:

Москва

2025

Ход работы

Видео:

<https://drive.google.com/file/d/1kGubQM6WtXutd5M3qtJ6pJ5P3hEJBTw1/view?usp=sharing>

ИТОГОВЫЙ КОД:

```
<?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:padding="12dp">

    <!-- Row 1: B, I, TextSize: 30, +, - -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:paddingBottom="8dp">

        <ToggleButton
            android:id="@+id/button_b"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:textOn="B"
            android:textOff="B"
            android:textStyle="bold" />

        <ToggleButton
            android:id="@+id/button_i"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:textOn="I"
            android:textOff="I"
            android:textStyle="italic"
            android:layout_marginStart="8dp" />

        <TextView
            android:id="@+id/label_textsize"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/label_textsize"
            android:layout_marginStart="16dp"
            android:paddingEnd="4dp" />

        <TextView
            android:id="@+id/size"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="30"
            android:paddingEnd="12dp" />
    </LinearLayout>
</LinearLayout>
```

```

<Button
    android:id="@+id/button_plus"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:text="+" />

<Button
    android:id="@+id/button_minus"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:text="-"
    android:layout_marginStart="8dp" />
</LinearLayout>

<!-- Row 2: Fonts radio buttons -->
<RadioGroup
    android:id="@+id/rgFonts"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical">

    <RadioButton
        android:id="@+id/button_sans"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SansSerif" />

    <RadioButton
        android:id="@+id/button_serif"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:layout_marginStart="12dp"
        android:text="Serif" />

    <RadioButton
        android:id="@+id/button_monospace"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="12dp"
        android:text="Monospace" />
</RadioGroup>

<!-- Row 3: Big formatted text -->
<EditText
    android:id="@+id/edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Форматируемый текст"
    android:textSize="30sp"
    android:typeface="serif"
    android:textStyle="italic"
    android:paddingTop="16dp"
    android:inputType="textMultiLine"
    android:overScrollMode="ifContentScrolls"
    android:scrollbars="vertical" />
</LinearLayout>

```

```

package com.example.fourcourse

import android.graphics.Typeface
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.RadioButton
import android.widget.TextView
import android.widget.ToggleButton
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), View.OnClickListener {
    private var mTextSize = 30f
    private lateinit var mEdit: EditText
    private lateinit var tSize: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        if (supportActionBar != null) {
            supportActionBar?.hide();
        }
        setContentView(R.layout.activity_main)

        mEdit = findViewById(R.id.edit_text)
        tSize = findViewById(R.id.size)

        val buttonB = findViewById<ToggleButton>(R.id.button_b)
        val buttonI = findViewById<ToggleButton>(R.id.button_i)
        val buttonSans = findViewById<RadioButton>(R.id.button_sans)
        val buttonSerif = findViewById<RadioButton>(R.id.button_serif)
        val buttonMono = findViewById<RadioButton>(R.id.button_monospace)
        val buttonPlus = findViewById<Button>(R.id.button_plus)
        val buttonMinus = findViewById<Button>(R.id.button_minus)

        // Initial UI to match screenshot
        tSize.text = mTextSize.toInt().toString()
        mEdit.textSize = mTextSize
        mEdit.setTypeface(Typeface.SERIF, Typeface.ITALIC)
        buttonI.isChecked = true
        buttonSerif.isChecked = true

        buttonB.setOnClickListener(this)
        buttonI.setOnClickListener(this)
        buttonSans.setOnClickListener(this)
        buttonSerif.setOnClickListener(this)
        buttonMono.setOnClickListener(this)
        buttonPlus.setOnClickListener(this)
        buttonMinus.setOnClickListener(this)
    }

    override fun onClick(p0: View?) {
        when (p0?.id) {
            R.id.button_b -> if (mEdit.typeface.style == Typeface.ITALIC)
                mEdit.setTypeface(mEdit.typeface, Typeface.BOLD_ITALIC)
            else if (mEdit.typeface.style == Typeface.BOLD_ITALIC)
                mEdit.setTypeface(mEdit.typeface, Typeface.ITALIC)
            else if (mEdit.typeface.style == Typeface.BOLD)
                mEdit.setTypeface(Typeface.create(mEdit.typeface, Typeface.NORMAL))
            else mEdit.setTypeface(mEdit.typeface, Typeface.BOLD)

            R.id.button_i -> if (mEdit.typeface.style == Typeface.BOLD)

```

```

        mEdit.setTypeface(mEdit.typeface, Typeface.BOLD_ITALIC)
    else if (mEdit.typeface.style == Typeface.BOLD_ITALIC)
        mEdit.setTypeface(mEdit.typeface, Typeface.BOLD)
    else if (mEdit.typeface.style == Typeface.ITALIC)
        mEdit.setTypeface(Typeface.create(mEdit.typeface, Typeface.NORMAL))
    else mEdit.setTypeface(mEdit.typeface, Typeface.ITALIC)

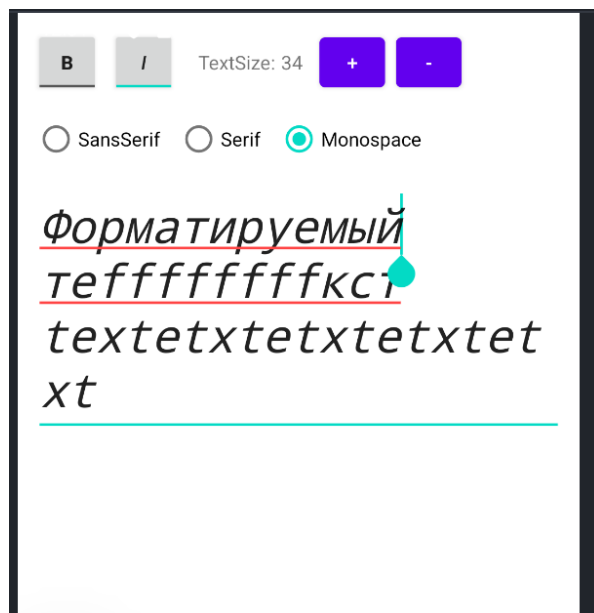
    R.id.button_sans -> mEdit.setTypeface(Typeface.SANS_SERIF, mEdit.typeface.style)
    R.id.button_serif -> mEdit.setTypeface(Typeface.SERIF, mEdit.typeface.style)
    R.id.button_monospace -> mEdit.setTypeface(Typeface.MONOSPACE, mEdit.typeface.style)

    R.id.button_plus -> {
        if (mTextSize <= 72) mTextSize += 2f
        mEdit.textSize = mTextSize
        tSize.text = mTextSize.toInt().toString()
    }

    R.id.button_minus -> {
        if (mTextSize >= 20) mTextSize -= 2f
        mEdit.textSize = mTextSize
        tSize.text = mTextSize.toInt().toString()
    }
}
}
}
}
}

```

Результат



Контрольные вопросы

1. **findViewById** — ищет элемент интерфейса по его id и возвращает объект, чтобы можно было работать с ним в коде.
2. **setTypeface** — задаёт шрифт и стиль текста (обычный, жирный, курсив, моноширинный и т. д.).

3. В `setTypeface` передаются:
4. объект `Typeface` (например, `Typeface.SERIF`),
5. стиль (`Typeface.NORMAL`, `BOLD`, `ITALIC`, `BOLD_ITALIC`).
6. **`setTextSize`** — изменяет размер текста элемента (`TextView`, `EditText`, `Button`).
7. **`setText`** — устанавливает текстовое содержимое элемента.
8. **`getText`** — возвращает текущий текст элемента (часто как `Editable` или `CharSequence`).
9. **`getTypeface`** — возвращает текущий шрифт и стиль текста элемента.
10. **`setOnClickListener`** — назначает обработчик события нажатия (код, который выполнится при клике).
11. **`setContentLayout`** — связывает `Activity` с определённым макетом (`layout`) для отображения интерфейса.
12. В Kotlin знак `?` после типа или переменной означает, что она **`nullable`** — может содержать `null`.