

Министерство образования, науки и молодежной политики
Краснодарского края
Государственное бюджетное профессиональное образовательное учреждение
Краснодарского края «Ейский полипрофильный колледж»

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

ЛАБОРАТОРНАЯ РАБОТА №5

по теме:

**Компоненты TextView, editTextNumber и Button.
Обработчики событий**

Выполнил:

студент ЕПК, группа
ФИО

Проверил:

преподаватель дисциплины
«Разработка мобильных
приложений»
Фомин А. Т.

1 Обработчики событий

В Android есть целый ряд методов, которые обеспечивают перехват событий взаимодействия пользователя с приложением. Класс View содержит набор вложенных интерфейсов с обратными вызовами, которые можно определить. Эти интерфейсы называются прослушивателями событий.

Например, если кнопка должна отвечать на событие нажатия, она должна зарегистрироваться для прослушивателя событий `onClickListener` и реализовать соответствующий метод обратного вызова `onClick()`.

В коде ниже показаны два варианта регистрации прослушивателей для кнопок `button` и `button2` в файле `MainActivity.kt`. Последний вызывает метод обработчика событий.

Листинг 1.

```
1. package com.example.project_1
2.
3. import android.os.Bundle
4. import android.view.View
5. import androidx.activity.enableEdgeToEdge
6. import androidx.appcompat.app.AppCompatActivity
7. import androidx.core.view.ViewCompat
8. import androidx.core.view.WindowInsetsCompat
9. import android.widget.Button
10. import android.widget.TextView
11. class MainActivity : AppCompatActivity() {
12.     override fun onCreate(savedInstanceState: Bundle?) {
13.         super.onCreate(savedInstanceState)
14.         enableEdgeToEdge()
15.         setContentView(R.layout.activity_main)
16.         val textView: TextView = findViewById(R.id.textView)
17.         // Регистрация прослушивателя для button
18.         val button: Button = findViewById(R.id.button)
19.         button.setOnClickListener {
20.             textView.text = "Нажата кнопка 1"
21.         }
22.         //-----
23.         // Регистрация прослушивателя для button2
```

```

24.         findViewById<Button>(R.id.button2).setOnClickListener {
25.             displayTextOnButtonClick(it)
26.         }
27.         //-----
28.         ViewCompat.setOnApplyWindowInsetsListener
29.             (findViewById(R.id.main)) { v, insets ->
30.             val systemBars =
31.                 insets.getInsets(WindowInsetsCompat.Type.systemBars())
32.                 v.setPadding(systemBars.left, systemBars.top,
33.                     systemBars.right, systemBars.bottom)
34.                 insets
35.             }
36.         }
37.         // Метод обработчика событий
38.         private fun displayTextOnButtonClick(view: View) {
39.             val text = findViewById<TextView>(R.id.textView)
40.             text.text = "Нажата кнопка 2"
41.         }
42.     }

```

2 Ход работы

1. Создайте новый проект Empty Views Activity. Выполните сборку запустите данный проект в виртуальном устройстве.
2. В пути app/res/layout/ откройте activity_main.xml и откройте представление Design
3. Перетащите на форму компоненты: TextView и Button так, как показано на рис.1
4. Перейдите в представление Code и отредактируйте код в файле activity_main.xml, согласно листингу 2
5. Перейдите к файлу MainActivity.kt и внесите изменения в классе MainActivity, согласно листингу 1
6. Вновь перейдите к файлу activity_main.xml и перейдите к представлению Design. В секции Attributes. Внесите изменения внешнего вида размещенных компонент. Кликните на компоненте TextView. В секции Attributes станут доступны атрибуты этого компонента. Измените следующие значения:

1. gravity (чек на center, center_horizon, center_vertical)
2. textAlignment
3. textColor
4. textSize

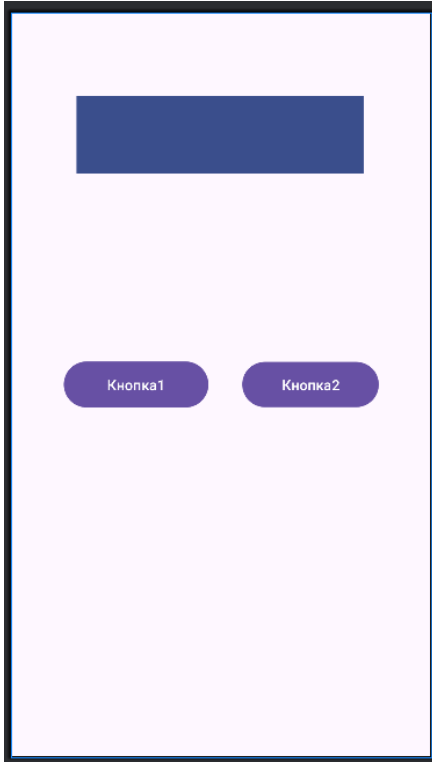


Рис. 1

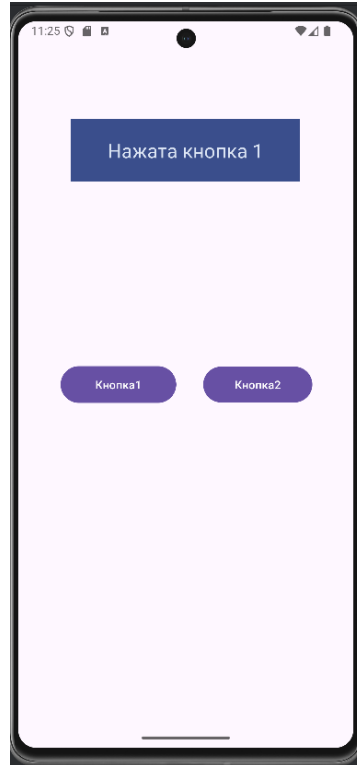


Рис. 2

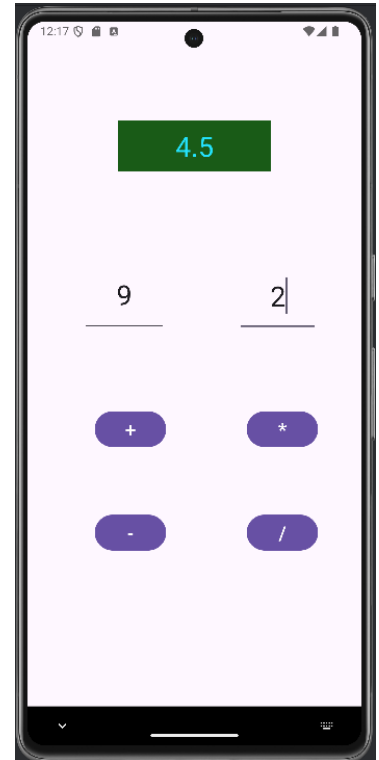


Рис. 3

7. Запустите приложение и проверьте правильно ли размещены компоненты на форме, правильно ли работает обработчик событий. При нажатии на кнопку в текстовом поле отображается номер кнопки так, как показано на рис.2.
 8. Сохраните результат работы.
 9. Создайте еще один проект, опираясь на код предыдущего проекта. Данный проект должен реализовать форму содержащую следующие компоненты:
 - два текстовых поля ввода (EditTextNumber)
 - одна метка (TextView)
 - 4 кнопки (Button)
- Выведите эти компоненты на форму и скомпонуйте так, как показано на рис.3
10. Оформите окно приложения, используя указания данные выше.
 11. Измените код класса MainActivity, так, как показано в листинге 3. Данный проект реализует калькулятор, выполняющий простейшие арифметические операции.

12. Текст и цвета, используемые в приложении, должны быть вынесены в ресурсы проекта по пути app/res/values. В коде программы должны быть ссылки на эти ресурсы.
13. Измените стиль приложения, установив фон окна.
14. Оформите отчет по выполненной лабораторной работе, включив в него листинги кода, скриншоты окон.

Листинг 2

```
1. <TextView
2.     android:id="@+id/textView"
3.     android:layout_width="282dp"
4.     android:layout_height="77dp"
5.     android:background="#3A4E8C"
6.     android:fontFamily="sans-serif"
7.     android:gravity="center|center_horizontal|
   center_vertical"
8.     android:text=""
9.     android:textAlignment="center"
10.    android:textColor="#DCE5F4"
11.    android:textSize="24sp"
12.    app:layout_constraintBottom_toBottomOf="parent"
13.    app:layout_constraintEnd_toEndOf="parent"
14.    app:layout_constraintHorizontal_bias="0.496"
15.    app:layout_constraintStart_toStartOf="parent"
16.    app:layout_constraintTop_toTopOf="parent"
17.    app:layout_constraintVertical_bias="0.125" />
18.
19. <Button
20.     android:id="@+id/button"
21.     android:layout_width="142dp"
22.     android:layout_height="53dp"
23.     android:text="Кнопка1"
24.     app:layout_constraintBottom_toBottomOf="parent"
25.     app:layout_constraintEnd_toEndOf="parent"
26.     app:layout_constraintHorizontal_bias="0.193"
27.     app:layout_constraintStart_toStartOf="parent"
28.     app:layout_constraintTop_toTopOf="parent" />
```

```

29.
30.     <Button
31.         android:id="@+id/button2"
32.         android:layout_width="134dp"
33.         android:layout_height="52dp"
34.         android:text="Кнопка2"
35.         app:layout_constraintBottom_toBottomOf="parent"
36.         app:layout_constraintEnd_toEndOf="parent"
37.         app:layout_constraintHorizontal_bias="0.819"
38.         app:layout_constraintStart_toStartOf="parent"
39.         app:layout_constraintTop_toTopOf="parent" />

```

Листинг 3

```

1.  package com.example.project2
2.
3.  import android.os.Bundle
4.  import android.view.View
5.  import androidx.activity.enableEdgeToEdge
6.  import androidx.appcompat.app.AppCompatActivity
7.  import androidx.core.view.ViewCompat
8.  import androidx.core.view.WindowInsetsCompat
9.  import android.widget.Button
10. import android.widget.TextView
11. import android.widget.EditText
12.
13. class MainActivity : AppCompatActivity() {
14.     override fun onCreate(savedInstanceState: Bundle?) {
15.         super.onCreate(savedInstanceState)
16.         enableEdgeToEdge()
17.         setContentView(R.layout.activity_main)
18.         val textView: TextView = findViewById(R.id.textView)
19.         findViewById<Button>(R.id.button).setOnClickListener {
20.             displayTextOnButtonClick_plus(it)
21.         }
22.         findViewById<Button>(R.id.button2).setOnClickListener {
23.             displayTextOnButtonClick_minus(it)
24.         }

```

```

25.         findViewById<Button>(R.id.button3).setOnClickListener {
26.             displayTextOnButtonClick_pow(it)
27.         }
28.         findViewById<Button>(R.id.button4).setOnClickListener {
29.             displayTextOnButtonClick_div(it)
30.         }
31.
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)
    ) { v, insets ->
32.         val systemBars =
    insets.getInsets(WindowInsetsCompat.Type.systemBars())
33.         v.setPadding(systemBars.left, systemBars.top,
    systemBars.right, systemBars.bottom)
34.         insets
35.     }
36. }
37. private fun displayTextOnButtonClick_plus(view: View) {
38.     val text = findViewById<TextView>(R.id.textView)
39.     val d1 =
    findViewById<EditText>(R.id.editTextNumber).text.toString()
    .toInt()
40.     val d2 =
    findViewById<EditText>(R.id.editTextNumber2).text.toString()
    .toInt()
41.     text.text = (d1 + d2).toString()
42. }
43. private fun displayTextOnButtonClick_minus(view: View) {
44.     val text = findViewById<TextView>(R.id.textView)
45.     val d1 =
    findViewById<EditText>(R.id.editTextNumber).text.toString()
    .toInt()
46.     val d2 =
    findViewById<EditText>(R.id.editTextNumber2).text.toString()
    .toInt()
47.     text.text = (d1 - d2).toString()
48. }
49. private fun displayTextOnButtonClick_pow(view: View) {
50.     val text = findViewById<TextView>(R.id.textView)

```

```
51.         val d1 =
            findViewById<EditText>(R.id.editTextNumber).text.toString()
            .toInt()
52.         val d2 =
            findViewById<EditText>(R.id.editTextNumber2).text.toString()
            .toInt()
53.         text.text = (d1 * d2).toString()
54.     }
55.     private fun displayTextOnButtonClick_div(view: View) {
56.         val text = findViewById<TextView>(R.id.textView)
57.         val d1 =
            findViewById<EditText>(R.id.editTextNumber).text.toString()
            .toDouble()
58.         val d2 =
            findViewById<EditText>(R.id.editTextNumber2).text.toString()
59.         .toDouble()
60.         text.text = (d1 / d2).toString()
61.     }
62. }
```