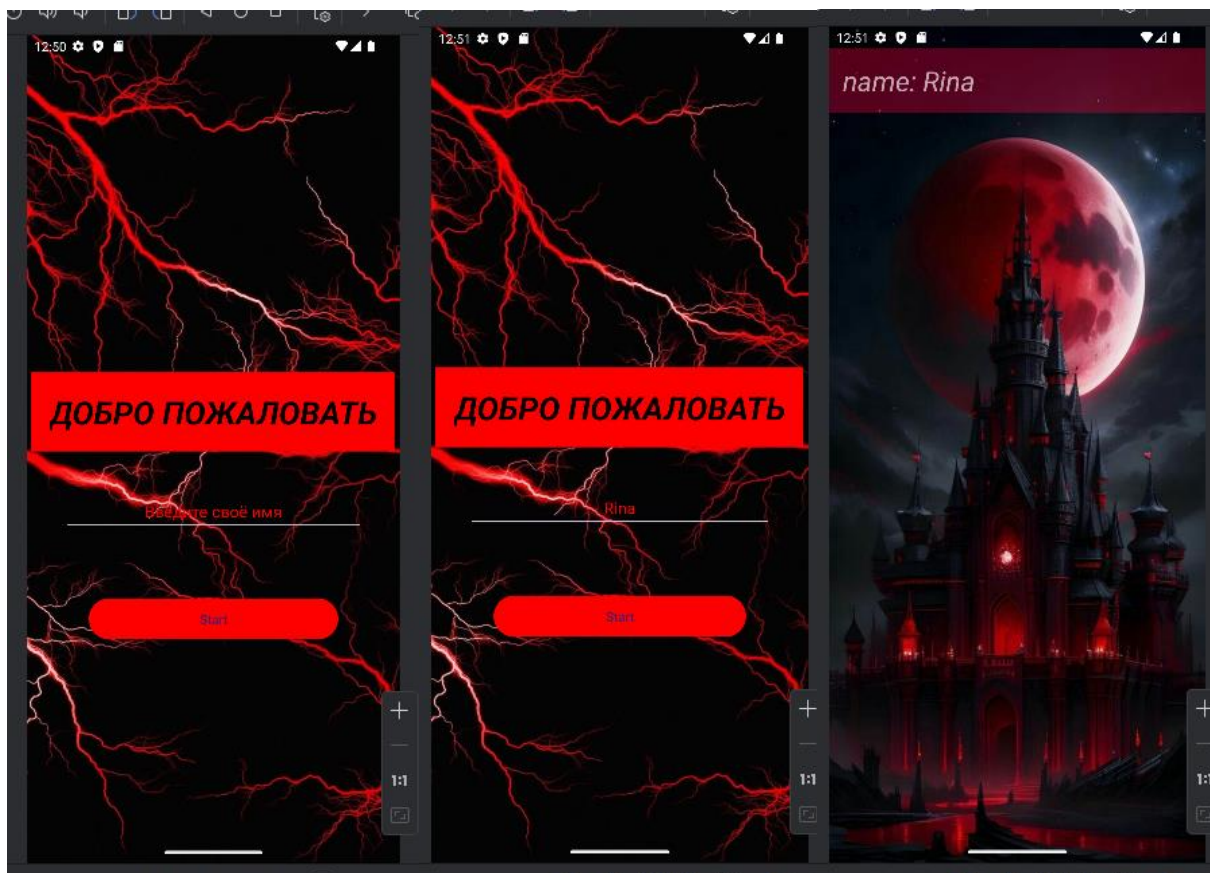


Практическая работа Intent

Задание 1



Ход работы:

1. В коде кнопки добавляю переменную текста, который вводит пользователь и присваиваю имя

```
val btn : Button = findViewById(R.id.btn_1)
val editText : EditText = findViewById(R.id.editText2)

btn.setOnClickListener{
    val inputText = editText.text.toString()
    val intent = Intent( packageContext: this, MainActivity2::class.java)
    intent.putExtra( name: "name", inputText)
    startActivity(intent)
}
```

2. Во втором xml файле добавляю текстовое окно, в которое будет помещаться текст пользователя

```
<TextView
    android:id="@+id/tv_Name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#56FF0040"
    android:padding="15dp"
    android:textColor="#A7FFFFFF"
    android:textSize="30dp"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

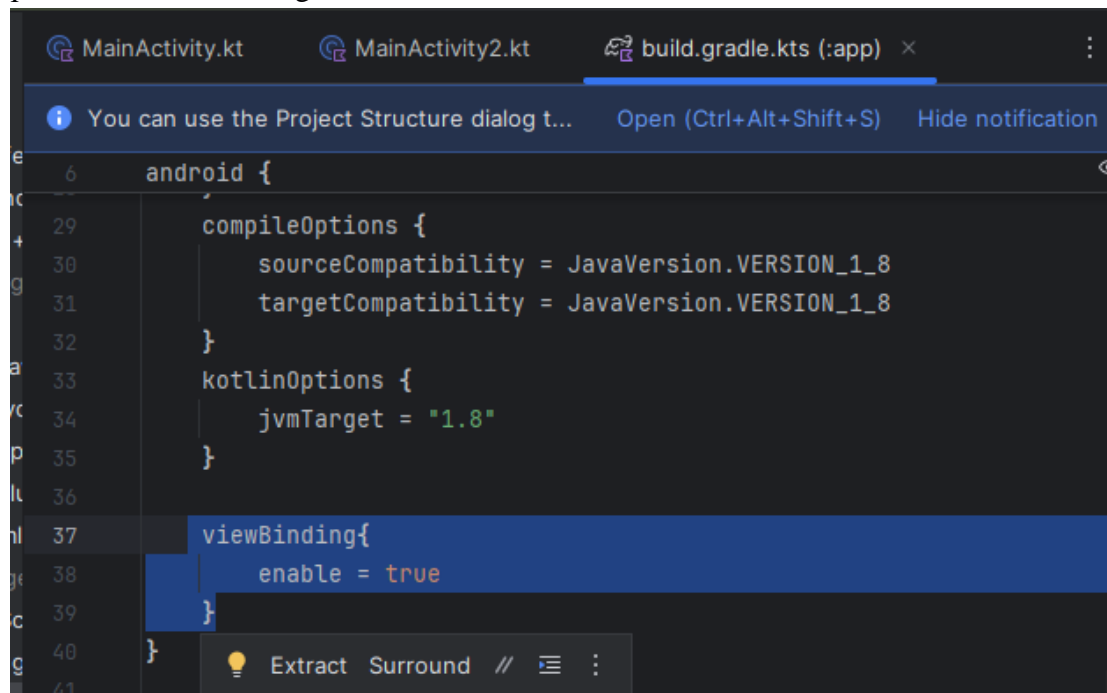
3. И также прописываю код вывода на экран имени пользователя со словом "name: " для конкретизации

```
val receivedText = intent.getStringExtra(name: "name")
val textView = findViewById<TextView>(R.id.tv_Name)
textView.text = "name: " + receivedText
```

4. Таким образом, всё, что введёт пользователь в первом окне, будет транслироваться во втором

Задание 2

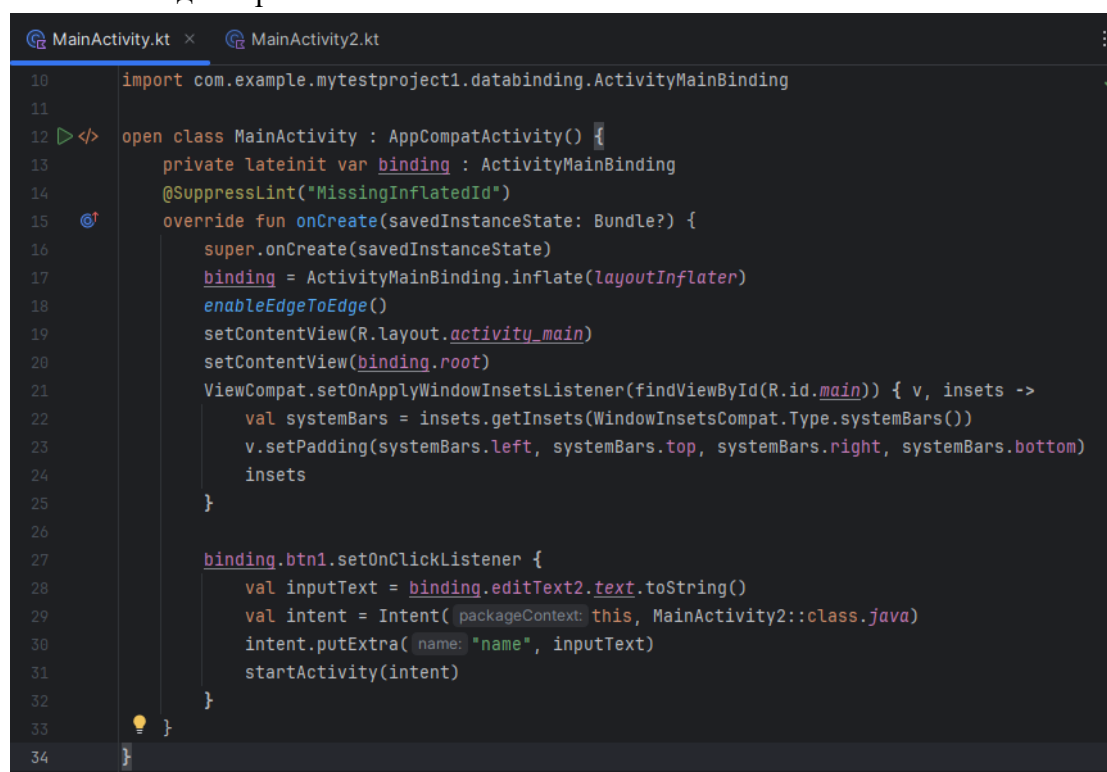
5. В Gradle Scripts открываю файл build.gradle и прописываю следующий код для работы с ViewBinding



The screenshot shows an IDE window with the file 'build.gradle.kts (:app)' open. The code is in Kotlin and configures the Android build system. A notification bar at the top says 'You can use the Project Structure dialog t... Open (Ctrl+Alt+Shift+S) Hide notification'. The code includes the 'android' block with 'compileOptions' and 'kotlinOptions' for Java 1.8 compatibility. The 'viewBinding' block is highlighted in blue, showing 'enable = true'. A context menu is visible at the bottom of the code block with options like 'Extract', 'Surround', and '//'.

```
6 android {
29     compileOptions {
30         sourceCompatibility = JavaVersion.VERSION_1_8
31         targetCompatibility = JavaVersion.VERSION_1_8
32     }
33     kotlinOptions {
34         jvmTarget = "1.8"
35     }
36
37     viewBinding{
38         enable = true
39     }
40 }
41
```

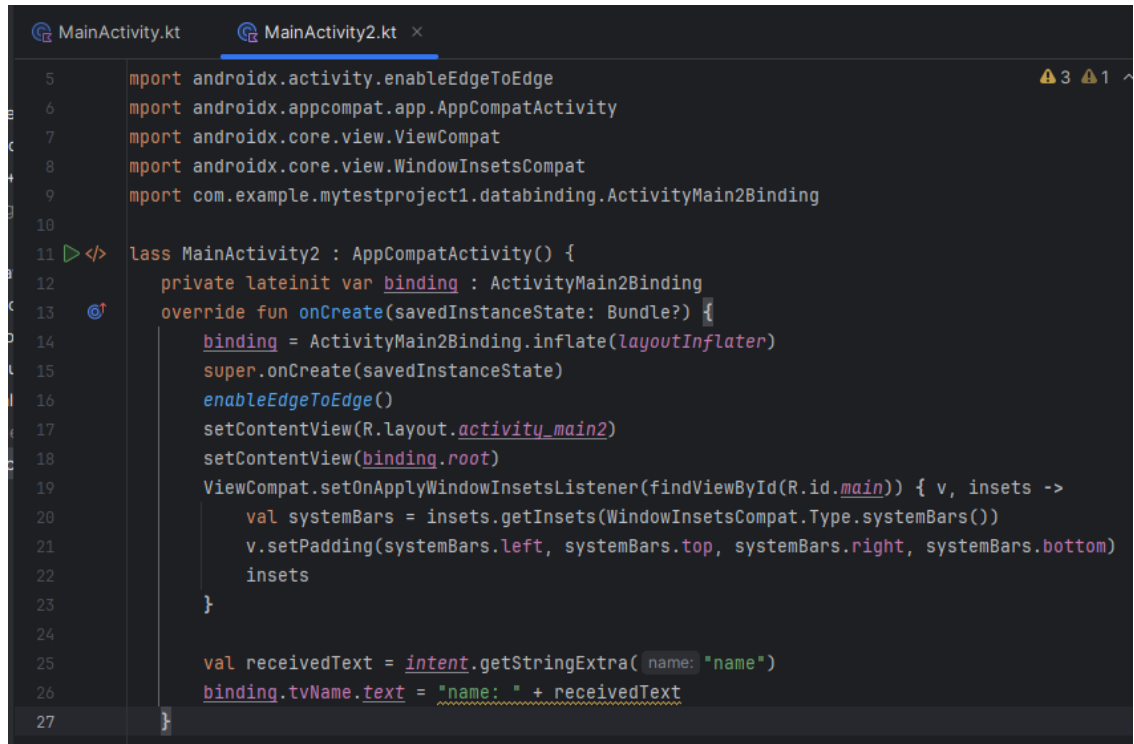
6. Изменяю код в первом классе



The screenshot shows an IDE window with the file 'MainActivity.kt' open. The code imports 'ActivityMainBinding' and implements the 'MainActivity' class. It uses ViewBinding to inflate the layout, set the content view, and handle a button click. The code is as follows:

```
10 import com.example.mytestproject1.databinding.ActivityMainBinding
11
12 open class MainActivity : AppCompatActivity() {
13     private lateinit var binding : ActivityMainBinding
14     @SuppressWarnings("MissingInflatedId")
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         binding = ActivityMainBinding.inflate(layoutInflater)
18         enableEdgeToEdge()
19         setContentView(R.layout.activity_main)
20         setContentView(binding.root)
21         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
22             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
23             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
24             insets
25         }
26
27         binding.btn1.setOnClickListener {
28             val inputText = binding.editText2.text.toString()
29             val intent = Intent(packageContext, MainActivity2::class.java)
30             intent.putExtra(name, "name", inputText)
31             startActivity(intent)
32         }
33     }
34 }
```

7. То же делаю и со вторым



```
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 com.example.mytestproject1.databinding.ActivityMain2Binding
10
11 class MainActivity2 : AppCompatActivity() {
12     private lateinit var binding : ActivityMain2Binding
13     override fun onCreate(savedInstanceState: Bundle?) {
14         binding = ActivityMain2Binding.inflate(layoutInflater)
15         super.onCreate(savedInstanceState)
16         enableEdgeToEdge()
17         setContentView(R.layout.activity_main2)
18         setContentView(binding.root)
19         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
20             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
21             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
22             insets
23         }
24
25         val receivedText = intent.getStringExtra(name: "name")
26         binding.tvName.text = "name: " + receivedText
27     }
28 }
```

8. В целом работа программы не изменилась, мы видим тот же результат, но, думаю, для работы с масштабными приложениями этот способ будет наиболее подходящим, так как он безопаснее и быстрее, но пока мы работаем с основами, проще для восприятия, понимания и написания будет первый способ