

*Практическая работа Intent*  
*Переход между Активностями*

Выполнил: Рыжевский Е.ИС233

**Задание 1**

**MainActivity.kt:**

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val btn: Button = findViewById(R.id.btn_intent)

        btn.setOnClickListener {
            val intent = Intent(this, MainActivity2::class.java)
            intent.putExtra("text", "Изменённый текст")
            intent.putExtra("number", 10000)
            startActivity(intent)
        }
    }
}
```

**activity\_main.xml:**

```
<?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:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/btn_intent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Открыть вторую активность" />

</LinearLayout>
```

**MainActivity2.kt:**

```
package com.example.myapplication

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity2 : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)

        val textView: TextView = findViewById(R.id.tv)

        val message = intent.getStringExtra("text")
        val number = intent.getIntExtra("number", 0)
    }
}
```

```

        textView.text = "$message\nПолучено число: $number"
    }
}

```

#### activity\_main2.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mainLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Текст из MainActivity"
        android:textSize="18sp" />

</LinearLayout>

```

#### build.gradle.kts:

```

plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.myapplication"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.myapplication"
        minSdk = 24
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    buildFeatures {
        viewBinding = true
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
    kotlinOptions {

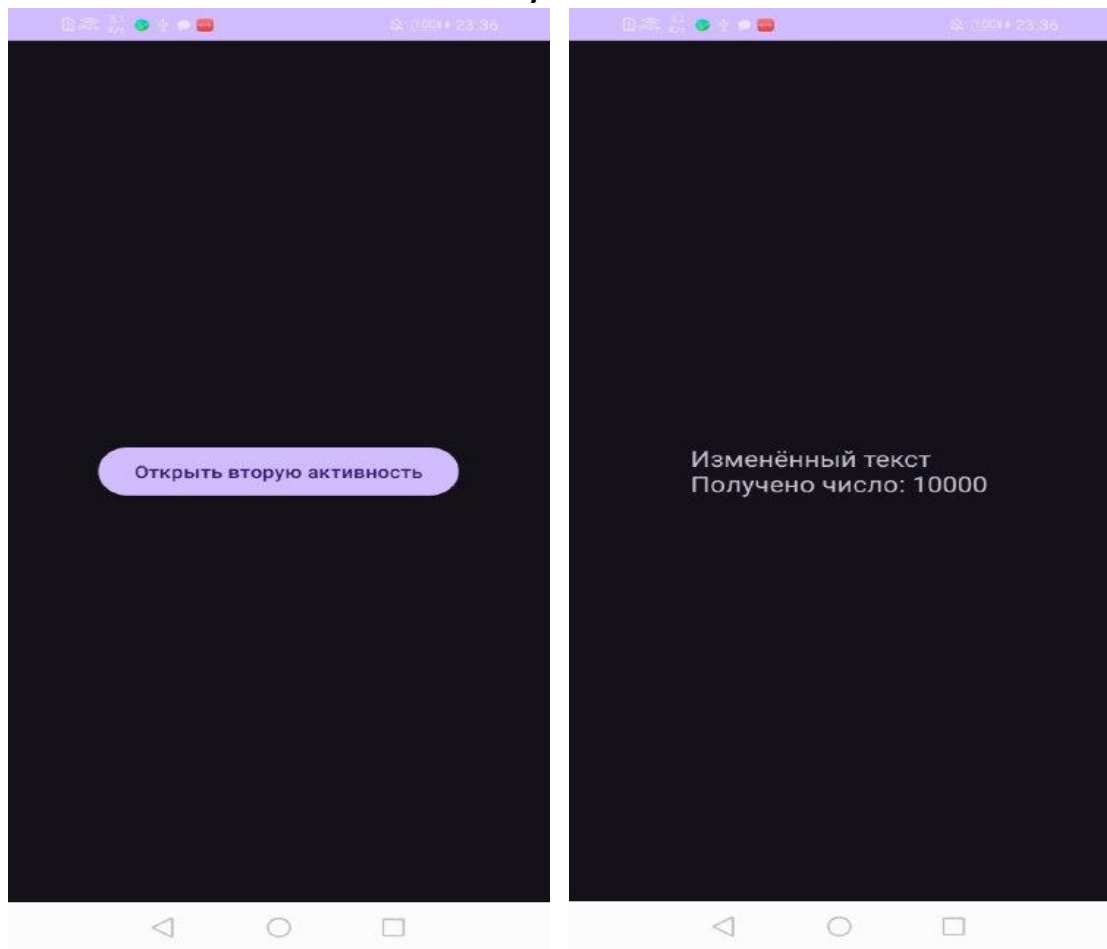
```

```

        jvmTarget = "11"
    }
}
dependencies {
    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}

```

### Результат:



**Задание 2:** Поработав с привязкой переменных `findViewById`, используйте `ViewBinding` и напишите, как вы лично считаете в чём разница и зачем нужен данный способ привязки.

**activity\_main.xml:**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#B3E5FC"
    android:gravity="center"

```

```

        android:orientation="vertical"
        android:padding="16dp">

        <Button
            android:id="@+id/btn_intent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Открыть вторую активность" />
    </LinearLayout>

```

### activity\_main2.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_main2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#B3E5FC"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ожидание..."
        android:textSize="18sp"
        android:textColor="#000000" />
</LinearLayout>

```

### MainActivity2.kt:

```

package com.example.myapplication

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.myapplication.databinding.ActivityMain2Binding

class MainActivity2 : AppCompatActivity() {

    private lateinit var binding: ActivityMain2Binding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMain2Binding.inflate(layoutInflater)
        setContentView(binding.root)

        val message = intent.getStringExtra("text")
        val number = intent.getIntExtra("number", 0)
        val price = intent.getDoubleExtra("price", 0.0)
        val isActive = intent.getBooleanExtra("isActive", false)

        binding.tv.text = """
            Сообщение: $message
            Число: $number
            Цена: $price
            Статус: $isActive
            """.trimIndent()
    }
}

```

### MainActivity.kt:

```

package com.example.myapplication

```

```

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.myapplication.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnIntent.setOnClickListener {
            val intent = Intent(this, MainActivity2::class.java).apply{
                putExtra("text", "Привет из MainActivity!")
                putExtra("number", 100)
                putExtra("price", 9.99)
                putExtra("isActive", true)
            }
            startActivity(intent)
        }
    }
}

```

#### build.gradle.kts:

```

plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.myapplication"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.myapplication"
        minSdk = 24
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    buildFeatures {
        viewBinding = true
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

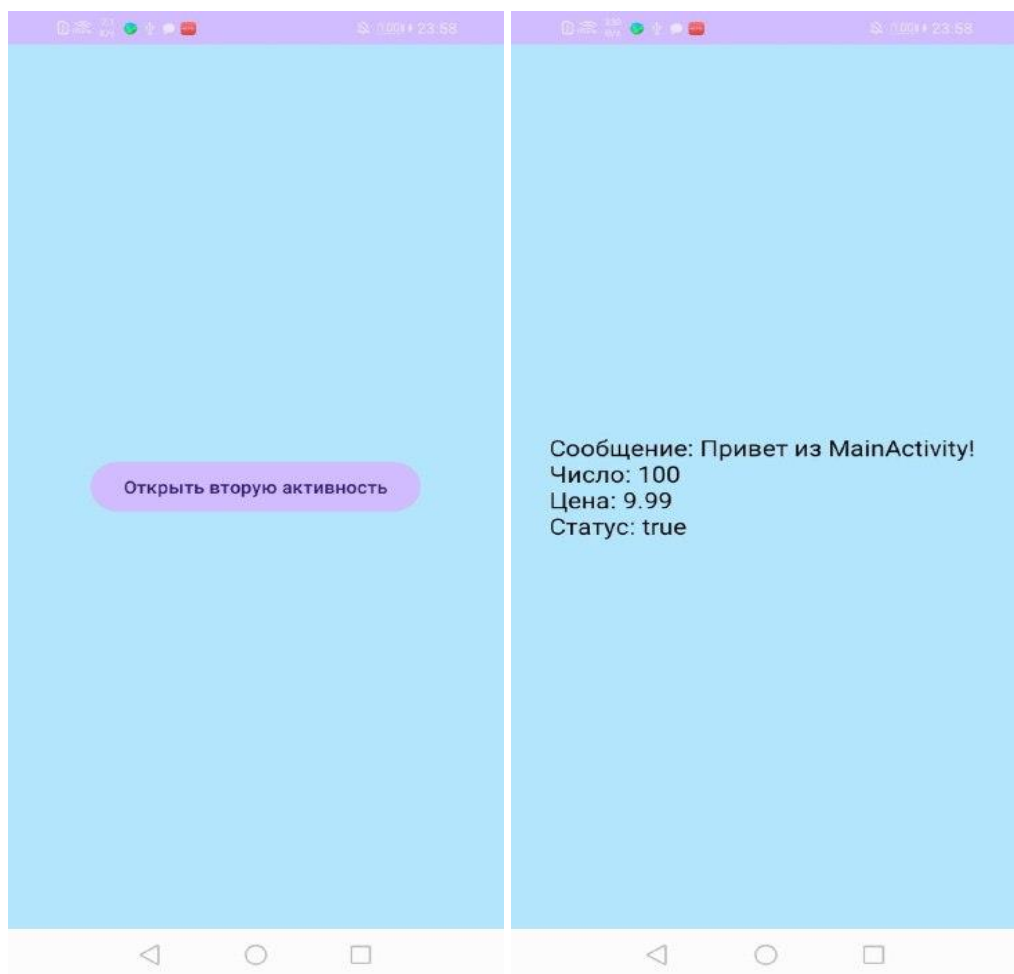
```

```

kotlinOptions {
    jvmTarget = "11"
}
dependencies {
    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}

```

## Результат:



Работая сначала с `findViewById`, я заметил, что этот способ требует много ручной работы: каждый элемент нужно искать по ID и указывать его тип. Это легко может привести к ошибкам — например, если указать неправильный тип или ID, приложение может упасть. Когда я начал использовать `ViewBinding`, стало намного удобнее. Android сам создаёт класс, в котором уже есть все элементы интерфейса с правильными типами. Доступ к ним происходит через объект `binding`, и ничего дополнительно искать не нужно. По моему мнению, `ViewBinding` делает код более читаемым и безопасным. К тому же он помогает сократить количество лишних строк и ускоряет процесс разработки. В целом, это

современный и надёжный способ привязки, который я бы выбрал вместо findViewById в любом проекте.