

Практическая работа Intent

Переход между Активностями

Задание 1: Протестируйте работу putExtra и getExtra, передайте не только строковые значения, а к примеру числовые.

MainActivity.kt:

```
val btn: Button = findViewById(R.id.btn_start)
btn.setOnClickListener{
    intent = Intent(this, MainActivity2::class.java)
    intent.putExtra("text", "Привет из MainActivity!")
    intent.putExtra("number", 42)
    intent.putExtra("decimal", 3.14)
    intent.putExtra("isTrue", true)
    intent.putExtra("array", intArrayOf(1, 2, 3, 4, 5))
    startActivity(intent)
}
```

MainActivity2.kt:

```
val textView = findViewById<TextView>(R.id.tv)

val text = intent.getStringExtra("text")
val number = intent.getIntExtra("number", 0)
val decimal = intent.getDoubleExtra("decimal", 0.0)
val isTrue = intent.getBooleanExtra("isTrue", false)
val array = intent.getIntArrayExtra("array")

val resultText = ""
    Полученный текст: $text
    Число: $number
    Дробное число: $decimal
    Логическое значение: $isTrue
    Массив: ${array?.joinToString()}
"".trimIndent()

textView.text = resultText
```

Результат на след странице:



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

MainActivity.kt:

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding

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

        ViewCompat.setOnApplyWindowInsetsListener(binding.main) { v, insets -
>
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }

        binding.btnStart.setOnClickListener {
            intent = Intent(this, MainActivity2::class.java)
            intent.putExtra("name", "Привет!")
            intent.putExtra("age", 20)
            startActivity(intent)
        }
    }
}
```

MainActivity2.kt:

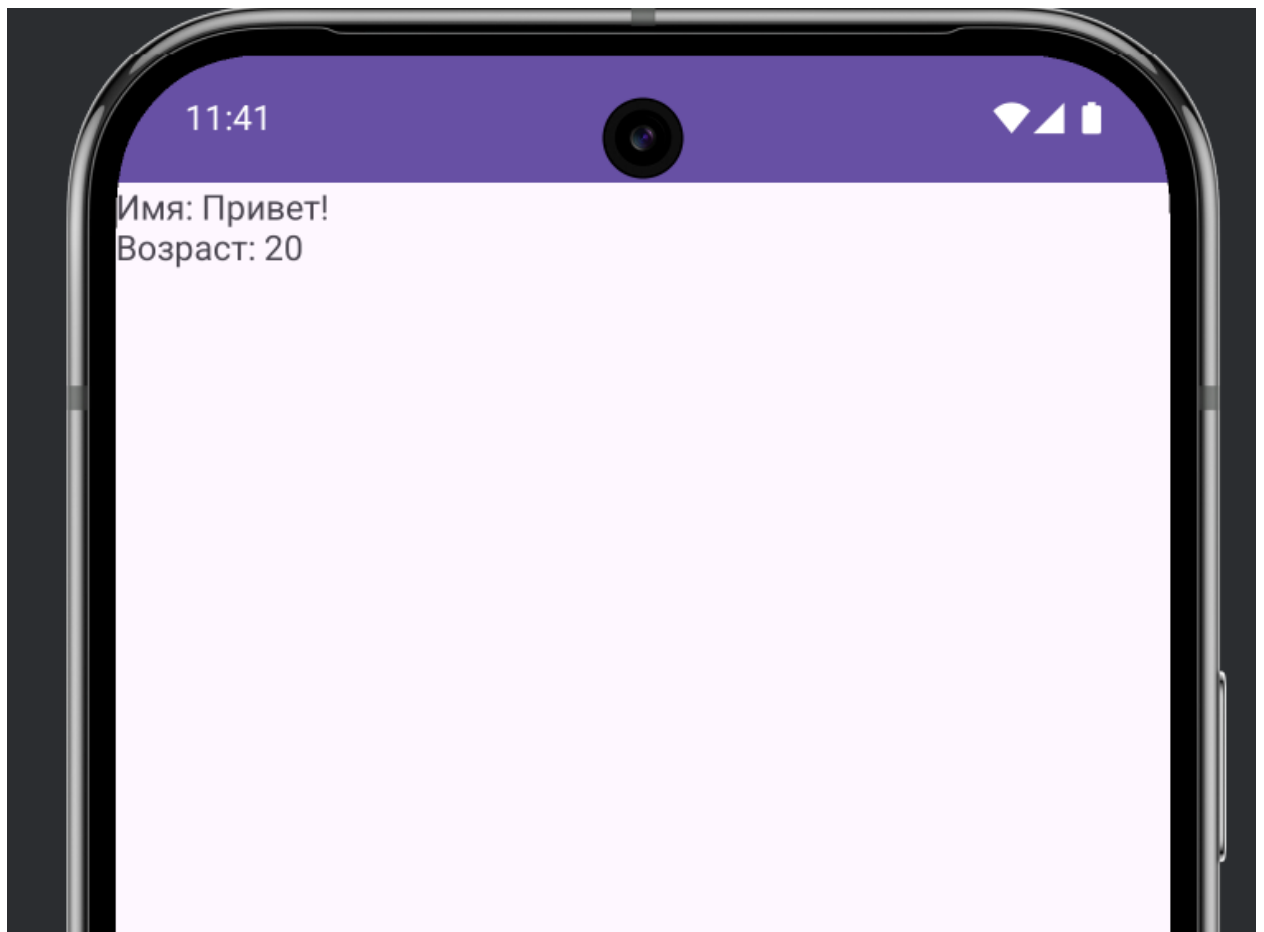
```
class MainActivity2 : AppCompatActivity() {
    private lateinit var binding: ActivityMain2Binding

    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMain2Binding.inflate(layoutInflater)
        setContentView(binding.root)

        val name = intent.getStringExtra("name")
        val age = intent.getIntExtra("age", 0)

        binding.tv.text = "Имя: $name\nВозраст: $age"
    }
}
```

Результат:



Ответ: ViewBinding упрощает разработку, делает код чище и снижает риск ошибок (например, сам автоматически генерирует правильные типы для элементов, предоставляет прямой доступ к элементам через короткие ссылки и т.д.).