

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", "edit text")
    intent.putExtra("number", 63)
    intent.putExtra("price", 66.66)
    startActivity(intent)
}
```

MainActivity2.kt:

```
val textView: TextView = findViewById(R.id.tv)
val message = intent.getStringExtra("text")
val number: Int = intent.getIntExtra("number", 0)
val price = intent.getDoubleExtra("price", 0.0)
textView.text = "Text: $message, Number: $number, Price: $price"
```

Скрин:

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:

```
private val binding: ActivityMainBinding by lazy {
    ActivityMainBinding.inflate(layoutInflater)
}

...

setContentView(binding.root)
binding.btnStart.setOnClickListener{
    val intent = Intent(this, MainActivity2::class.java)
    intent.putExtra("text", "It's my live!!!")
```

```
        startActivity(intent)
    }
MainActivity2.kt:
private val binding: ActivityMain2Binding by lazy {
    ActivityMain2Binding.inflate(layoutInflater)
}
...
setContentView(binding.root)
    val textMessage = intent.getStringExtra("text")
    val number = intent.getIntExtra("number", 1000)
    binding.tv.text = "Текст: $textMessage\nЧисло: $number"
}
```

Скрин:

На след.странице



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.).