Базовый Kotlin

Матвей Попов

План

- Как и где можно писать на Kotlin
- Переменные, арифметические операции
- Интерполяция, операторы сравнения, условные операторы
- Циклы, массивы, коллекции
- Функции
- Классы, ООП, интерфейсы
- ENUM, лямбды, Extensions

Где можно писать Kotlin код?

- IntelliJ IDEA
- VS code
- KPlay: https://play.kotlinlang.org/
- Terminal + Kotlinc + JDK

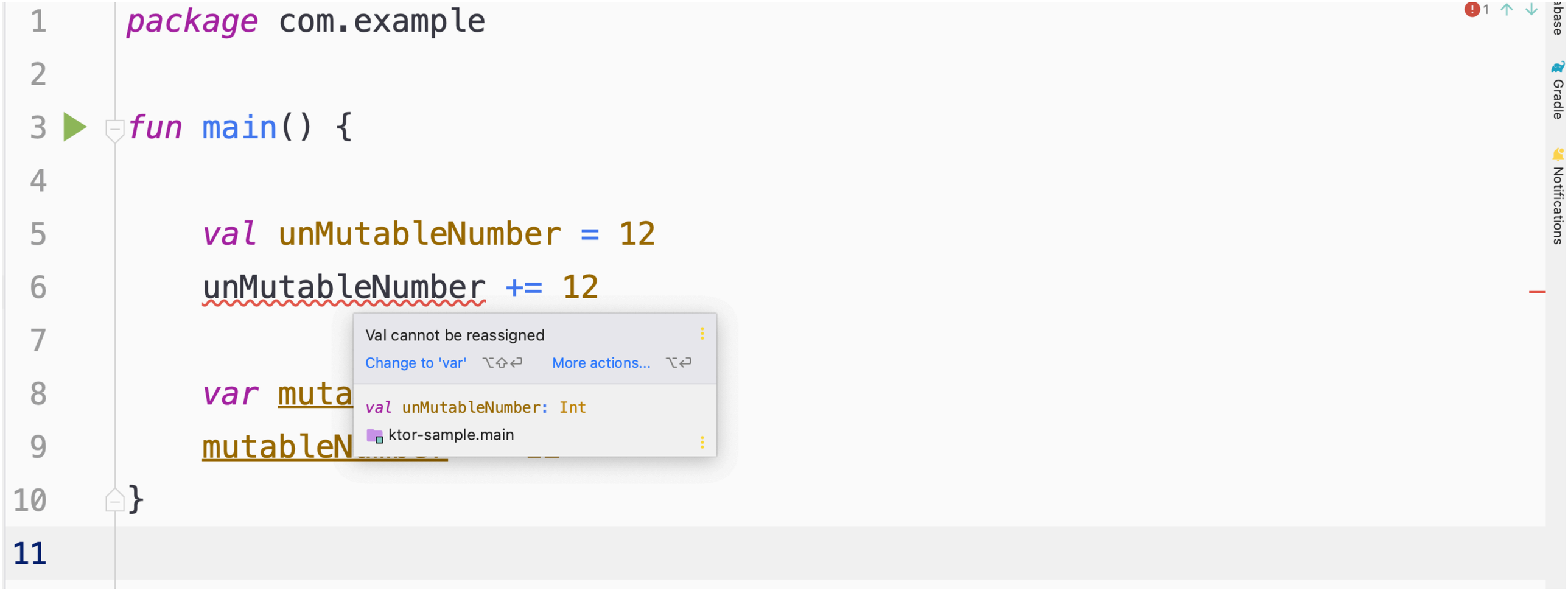
Переменные


fun main() {
val number = 12
val text = "Some random text"
}
Объект
Ключевое слово
Название переменной

Виды переменных

- Изменяемые
- Не изменяемые

```
1  package com.example
2
3  fun main() {
4
5      val unMutableNumber = 12
6      unMutableNumber += 12
7
8      var mutableNumber = 10
9      mutableNumber += 12
10 }
11
```


1
package com.example
2
3
fun main() {
4
5
val unMutableNumber = 12
6
unMutableNumber += 12
7
Val cannot be reassigned
Change to 'var'
More actions...
8
var muta
val unMutableNumber: Int
9
ktor-sample.main
10
}
11
1
Gradle
Notifications

А где указывается тип данных?

```
1  package com.example
2
3  fun main() {
4
5      val myText = "my text"
6
7      val anotherMyText: String = "another my text"
8  }
9
```


1 package com.example
2
3 fun main() {
4
5 val myText = "my text"
6
7 val ano nother my text"
8 }
3
Variable 'myText' is never used
Remove variable 'myText' ⌥⇧⏎ More actions... ⌥⏎
val myText: String
ktor-sample.main

Типы данных

- Целочисленные
- Вещественные
- Строковые
- Логические
- Объекты

```kotlin
package com.example

import com.example.example.MyClass

fun main() {

    val myShort: Short = 1
    val myInt: Int = 12
    val myLong: Long = 89643
    val myUnsignedByte: UByte = 1u

    val myDouble: Double = 213.34
    val myFloat: Float = -12.3f

    val myString: String = "my string"
    val myChar: Char = 'A'

    val myBool: Boolean = false

    val myClass: MyClass = MyClass()

```

Арифметические операции

```kotlin
package com.example

fun main() {

    val firstNumber = 5
    val secondNumber = 7

    println(firstNumber + secondNumber) // 12
}

```

```kotlin
package com.example

fun main() {

    val first = 10
    val second = 3
    println(first / second) // 3

    val firstDouble: Double = 10.0
    val secondDouble: Double = 3.0
    println(firstDouble / secondDouble) // 3.3333333333333335
}

```

```kotlin
package com.example

fun main() {

    val intNumber: Int = 5
    val floatNumber: Float = 13.2f

    println(intNumber + floatNumber) // 18.2
    println(floatNumber / intNumber) // 2.6399999
}

```

```
1  package com.example
2
3  fun main() {
4
5      val intNumber: Int = 5
6      val floatNumber: Float = 13.2f
7
8      println(intNumber > floatNumber) // false
9      println(floatNumber == intNumber)
10 }
11
```

1 4

Operator '==' cannot be applied to 'Float' and 'Int'

```
val floatNumber: Float
```

ktor-sample.main

Интерполяция

Интерполяция строк

Процесс вычисления строкового литерала, содержащего один или несколько заполнителей, в результате чего заполнители заменяются соответствующими значениями

```kotlin
package com.example

fun main() {

    // Конкатинация строк

    val greeting = "Добрый день"
    val delimiter = ", "

    println(greeting + delimiter + "уважаемые студенты.")
}

```

```kotlin
package com.example

fun main() {

    // Интерполяция строк

    val greeting = "Добрый день"
    val delimiter = ","

    println("$greeting$delimiter Звездный Лорд")
}

```

```
1  package com.example
2
3  fun main() {
4
5      // Интерполяция строк
6
7      val leftValue = 12
8      val rightValue = -6
9
10     println("Left value is bigger than right value: ${leftValue > rightValue}")
11 }
12
```

```kotlin
package com.example

fun main() {

    // Многострочный текст

    val multiLineText = """
        first_line
        second_line
        third_line
        fourth_line
    """.trimIndent()

    println(multiLineText)
}
```

Операторы сравнения

```kotlin
package com.example

fun main() {
    // Логические операторы
    // Операторы сравнения

    val variable = 1 + 4
    val boolVariable = (5 == (6 - 1))

    // <, >, <=, >=, ==, != (операторы сравнения)
    // ===, !== (операторы ссылочного сравнения)

    // &&, ||, ! (логические операторы)
    val result = (variable == 5) && boolVariable
    val anotherResult = (variable != 2) and !boolVariable
}

```

```kotlin
package com.example

import kotlin.random.Random

fun main() {

    val leftSide = 15
    val rightSize = 30

    val inputVariable = Random( seed: 12).nextInt()

    println(inputVariable in leftSide ≤.. ≤ rightSize)
}

```

Условные операторы

Или операторы ветвления

```
1  package com.example
2
3  fun main() {
4
5      val includeParams = true
6
7      if (includeParams) {
8          println("Some text with params: $includeParams")
9      } else {
10         println("Something went wrong...")
11     }
12 }
13
```

Конструкции if/else могут возвращать значения

```kotlin
package com.example

import kotlin.random.Random

fun main() {

    val inputAge = Random( seed: 12).nextInt()

    val resultText = if (inputAge >= 18) {
        "You are adult!"
    } else {
        "You are child!"
    }

    println(resultText)
}

```

```kotlin
package com.example

import kotlin.random.Random

fun main() {

    val randomInt = Random( seed: 12).nextInt()

    when (randomInt) {
        1 -> println("Your value is one")
        2 -> println("Your value is two")
        3 -> {
            val randomLong = Random( seed: 11).nextLong()
            println("Your long value is $randomLong")
        }
        else -> println("Your value is $randomInt")
    }
}

```

Все о циклах

Цикл

Разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкций

```
1  package com.example
2
3  fun main() {
4
5      var counter = 5
6
7      while (counter > 0) {
8          println("Current step number: $counter")
9          counter--
10     }
11 }
12
```

Массивы

Коллекции

Функции

Классы

Null safety

ООП

ENUM

Лямбда функции

Extensions

А теперь про домашнее задание

Материалы

Вопросы?

Спасибо за уделенное время!