Циклы в Kotlin

В Kotlin цикл `while` есть как в варианте с предусловием, так и постусловием. Цикл `for` присутствует только в варианте обхода элементов коллекций, массивов, строк и других объектов, имеющих метод `iterator()`, который создает итерации. Также есть функции `repeat(){}` и `forEach{}`, использование которых может быть более удобно в некоторых случаях. Эти функции принимают лямбда-выражения.

Примеры с циклами `while` и `do-while`:

```
fun main() {
    var a = readln()

    while (a == "") {
        println("Write a string")
        a = readln()
    }

    println(a)
}
```

```
fun main() {
    var a: String

    do {
        println("Write a string")
        a = readln()
    } while (a == "")

    println(a)
}
```

В заголовке цикла `for` нередко используются диапазоны:

```
fun main() {

    for (i in 0..9)
        print(i)
    // вывод: 0123456789

    println()
}
```

Можно воспользоваться свойствами `length` или `size`, чтобы узнать размер строки, массива или коллекции. Далее полученное значение использовать в диапазоне заголовка цикла `for`. Однако бывает удобнее пользоваться свойствами коллекций, которые уже возвращают готовый диапазон или объекты, состоящие из пар индекс-значение:

```
fun main() {
    val c = mutableListOf(3, 5, 7)

    for (i in 0 until c.size)
        c[i] += 2

    println(c) // [5, 7, 9]

    for (i in c.indices)
        c[i] += 2

    println(c) // [7, 9, 11]

    for ((i, item) in c.withIndex())
        c[i] = item + 2

    println(c) // [9, 11, 13]
}
```

В примере выше в одном случае мы сами формируем диапазон. Во втором – его нам возвращает вызов `indices`. В третьем используется функция `withIndex`, возвращающая итерируемый объект, состоящих из пар "индекс-значение".

Функция-метод `forEach` выполняет переданное ей лямбда-выражение для каждого элемента.

```
fun main() {
    val c = listOf(3, 5, 7)
    c.forEach { println(it) }
}
```

Данный код выведет каждый элемент списка с новой строки. Здесь `it` – это очередной элемент списка.

Функция `repeat` повторяет переданный в лямбда-выражении фрагмент кода заданное число раз, которое передается первым аргументом.

```
fun main() {
    val a = listOf(1, 4, 8)

    repeat(3) {
        println(a[it])
    }
}
```

Обратим внимание, что несмотря на то, что аргумент функции – число 3, `it` принимает значения 0, 1 и 2, которые в коде выше используются как индексы для извлечения элементов из списка.

PDF-версия курса с дополнительными уроками