Коллекции в Kotlin - списки, словари и множества

В Kotlin есть три основных разновидности коллекций – список (List), словарь (Map) и множество (Set). Множества содержат только уникальные значения, в них не может быть двух элементов с одинаковыми значениями, а порядок элементов не важен.

Коллекции – это классы-дженерики, поэтому при создании объектов явно указывается или компилятор сам выводит тип составляющих коллекцию объектов. Примеры объявлений переменных типов коллекций без связывания их с объектами:

```
fun main() {
    val c: List<Int>
    val d: Map<String, Int>
    val e: Set<Byte>
}
```

Здесь список и множество состоят из целых чисел, в словаре ключами являются строки, а значениями числа. Тип может быть родительским. Тогда коллекция может содержать любой дочерний тип данных, однако необходимость в подобном возникает редко.

Обычно коллекции создаются не с помощью конструкторов, а вызовом специальных функций языка Kotlin – `listOf()`, `mapOf()`, `setOf()`.

```
fun main() {
    val c = listOf(1, 4, 6, 1)
    val d = mapOf("apple" to 10, "orange" to 5)
    val e = setOf(3, 4, 5, 0)
}
```

Типы переменных можно опустить, так как компилятор способен их вывести сам из объектов.

Список можно создать с помощью конструктора. Это удобно, когда список длинный или заполняется одинаковыми значениями.

```
val c = List(10){0}
```

В примере будет создан список из десяти элементов, каждый элемент будет равен нулю. В фигурных скобках находится лямбда-выражение, которое может быть сложнее.

В Kotlin интерфейсы **`List`**, **`Map`** и **`Set`** относятся к неизменяемым коллекциям. Другими словами, все, что приведено выше, не допускает изменения своих элементов.

Если нужны изменяемые коллекции, создавать их следует от **`MutableList`**, **`MutableMap`** и **`MutableSet`**. Соответственно используются функции `mutableListOf()`, `mutableMapOf()`, `mutableSetOf()`. Изменяемые интерфейсы наследуют от своих неизменяемых интерфейсов + интерфейс **`MutableCollection`** (только для **`MutableList`** и **`MutableSet`**).

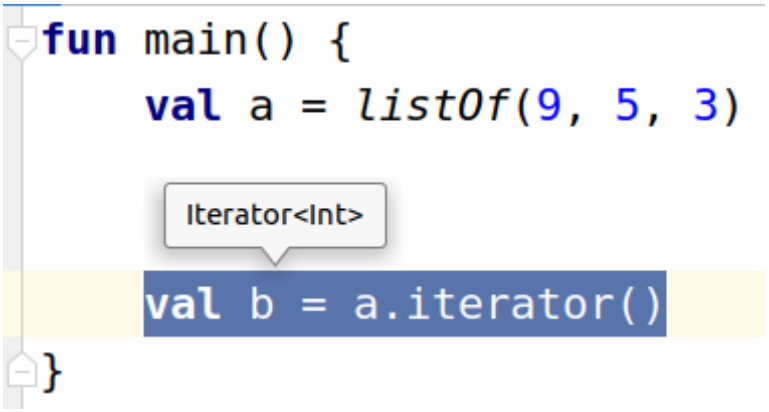
```
fun main() {
    val c = MutableList(10){0}
    c[0] = 109

    val a = mutableListOf<Int>()
    a.add(40)

    val b = mutableMapOf('a' to 10, 'b' to 5)
    b['c'] = 12

    for ((key, value) in b)
        println("$key = $value")
}
```

Коллекции относятся к итерируемым объектам, то есть у них есть метод `iterator()`, вызов которого создает объект-итератор. Этот метод вызывается неявно в заголовке цикла **`for`**.



Класс `ArrayList` – конкретная реализация изменяемого списка. Используется по умолчанию. Его конструктор несколько иной, лямбда ему не передается.

```
fun main() {
    val a = ArrayList<String>()
    a.add("Hello")
    a.add("World")

    val b = setOf("hi", "oh")
    val c = ArrayList(b)
}
```

PDF-версия курса с дополнительными уроками