



图 2.16 使用 downTo 遍历降序区间

另外，降序区间也是可以结合 `step` 关键字跳过区间中的一些元素的，这里我就不进行演示了，你可以自己动手试一试。

`for-in` 循环除了可以对区间进行遍历之外，还可以用于遍历数组和集合，关于集合这部分内容，我们在本章后面的部分就会学到，到时候再延伸 `for-in` 循环的相关用法。

如果让我总结一下的话，我觉得 `for-in` 循环并没有传统的 `for-i` 循环那样灵活，但是却比 `for-i` 循环要简单好用得多，而且足够覆盖大部分的使用场景。如果有一些特殊场景使用 `for-in` 循环无法实现的话，我们还可以改用 `while` 循环的方式来进行实现。

好了，关于 Kotlin 的循环部分就先讲这么多吧。

## 2.5 面向对象编程

和很多现代高级语言一样，Kotlin 也是面向对象的，因此理解什么是面向对象编程对我们来说就非常重要了。关于面向对象编程的解释，你可以去看很多标准化、概念化的定义，但是我觉得那些定义只有本来就懂的人才能看得懂，而不了解面向对象的人，即使看了那些定义还是不明白什么才是面向对象编程。

因此，这里我想用自己的理解来向你解释什么是面向对象编程。不同于面向过程的语言（比如 C 语言），面向对象的语言是可以创建类的。类就是对事物的一种封装，比如说人、汽车、房屋、书等任何事物，我们都可以将它封装一个类，类名通常是名词。而类中又可以拥有自己的字段和函数，字段表示该类所拥有的属性，比如说人可以有姓名和年龄，汽车可以有品牌和价格，这些就属于类中的字段，字段名通常也是名词。而函数则表示该类可以有哪些行为，比如说人可以吃饭和睡觉，汽车可以驾驶和保养等，函数名通常是动词。

通过这种类的封装，我们就可以在适当的时候创建该类的对象，然后调用对象中的字段和函数来满足实际编程的需求，这就是面向对象编程最基本的思想。当然，面向对象编程还有很多其他特性，如继承、多态等，但是这些特性都是建立在基本的思想之上的，理解了基本思想之后，其他的特性我们可以在后面慢慢学习。