可以看到，这里同时使用了 filter 和 map 函数，并通过 Lambda 表示式将水果单词长度限制在 5 个字母以内。重新运行一下代码，结果如图 2.27 所示。

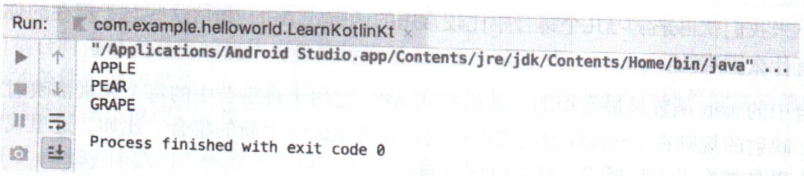

图 2.27　对水果单词长度进行过滤

另外值得一提的是，上述代码中我们是先调用了 filter 函数再调用 map 函数。如果你改成先调用 map 函数再调用 filter 函数，也能实现同样的效果，但是效率就会差很多，因为这样相当于要对集合中所有的元素都进行一次映射转换后再进行过滤，这是完全不必要的。而先进行过滤操作，再对过滤后的元素进行映射转换，就会明显高效得多。

接下来我们继续学习两个比较常用的函数式 API——any 和 all 函数。其中 any 函数用于判断集合中是否至少存在一个元素满足指定条件，all 函数用于判断集合中是否所有元素都满足指定条件。由于这两个函数都很好理解，我们就直接通过代码示例学习了：

```
fun main() {
    val list = listOf("Apple", "Banana", "Orange", "Pear", "Grape", "Watermelon")
    val anyResult = list.any { it.length <= 5 }
    val allResult = list.all { it.length <= 5 }
    println("anyResult is " + anyResult + ", allResult is " + allResult)
}
```

这里还是在 Lambda 表达式中将条件设置为 5 个字母以内的单词，那么 any 函数就表示集合中是否存在 5 个字母以内的单词，而 all 函数就表示集合中是否所有单词都在 5 个字母以内。现在重新运行一下代码，结果如图 2.28 所示。

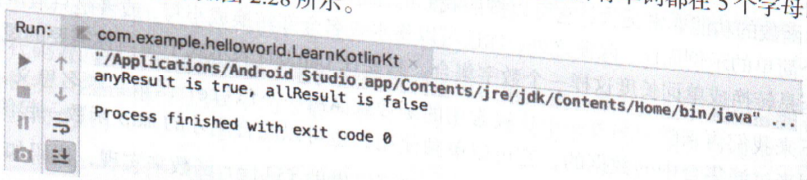

图 2.28　any 和 all 函数的执行结果

这样我们就将 Lambda 表达式的语法结构和几个常用的函数式 API 的用法都学习完了，虽然集合中还有许多其他函数式 API，但是只要掌握了基本的语法规则，其他函数式 API 的用法只要看一看文档就能掌握了，相信这对你来说并不是难事。