习一些 Kotlin 的进阶知识，等全部学完本书之后，你将能同时熟练地掌握 Kotlin 和 Android 这两门技术。

如果你还想学习如何使用 Java 来开发 Android 应用程序，那么请参阅本书的第 2 版。

2.1 Kotlin 语言简介

我想大多数人听说或知道 Kotlin 的时间并不长，但其实它并不是一门很新的语言。Kotlin 是由 JetBrains 公司开发与设计的，早在 2011 年，JetBrains 就公布了 Kotlin 的第一个版本，并在 2012 年将其开源，但在早期，它并没有受到太多的关注。

2016 年，Kotlin 发布了 1.0 正式版，这代表着 Kotlin 已经足够成熟和稳定了，并且 JetBrains 也在自家的旗舰 IDE 开发工具 IntelliJ IDEA 中加入了对 Kotlin 的支持，自此 Android 开发语言终于有了另外一种选择，Kotlin 逐渐受到广泛的关注。

接下来的事情你已经知道了，2017 年 Google 宣布 Kotlin 正式成为 Android 一级开发语言，Android Studio 也加入了对 Kotlin 的支持，Kotlin 自此开始大放异彩。

看到这里，或许你会产生一些疑惑：Android 操作系统明明是由 Google 开发的，为什么 JetBrains 作为一个第三方公司，却能够自己设计出一门编程语言来开发 Android 应用程序呢？

想要搞懂这个问题，我们得先来探究一下 Java 语言的运行机制。编程语言大致可以分为两类：编译型语言和解释型语言。编译型语言的特点是编译器会将我们编写的源代码一次性地编译成计算机可识别的二进制文件，然后计算机直接执行，像 C 和 C++都属于编译型语言。解释型语言则完全不一样，它有一个解释器，在程序运行时，解释器会一行行地读取我们编写的源代码，然后实时地将这些源代码解释成计算机可识别的二进制数据后再执行，因此解释型语言通常效率会差一些，像 Python 和 JavaScript 都属于解释型语言。

那么接下来我要考你一个问题了，Java 是属于编译型语言还是解释型语言呢？对于这个问题，即使是做了很多年 Java 开发的人也可能会答错。有 Java 编程经验的人或许会说，Java 代码肯定是要先编译再运行的，初学 Java 的时候都用过 javac 这个编译命令，因此 Java 属于编译型语言。如果这也是你的答案的话，那么恭喜你，答错了！虽然 Java 代码确实是要先编译再运行的，但是 Java 代码编译之后生成的并不是计算机可识别的二进制文件，而是一种特殊的 class 文件，这种 class 文件只有 Java 虚拟机（Android 中叫 ART，一种移动优化版的虚拟机）才能识别，而这个 Java 虚拟机担当的其实就是解释器的角色，它会在程序运行时将编译后的 class 文件解释成计算机可识别的二进制数据后再执行，因此，准确来讲，Java 属于解释型语言。

了解了 Java 语言的运行机制之后，你有没有受到一些启发呢？其实 Java 虚拟机并不直接和你编写的 Java 代码打交道，而是和编译之后生成的 class 文件打交道。那么如果我开发了一门新的编程语言，然后自己做了个编译器，让它将这门新语言的代码编译成同样规格的 class 文件，Java 虚拟机能不能识别呢？没错，这其实就是 Kotlin 的工作原理了。Java 虚拟机不关心 class 文