Broadview
www.broadview.com.cn

FK 疯狂软件教育 www.fkjava.org 标准教材

疯狂Kotlin讲义

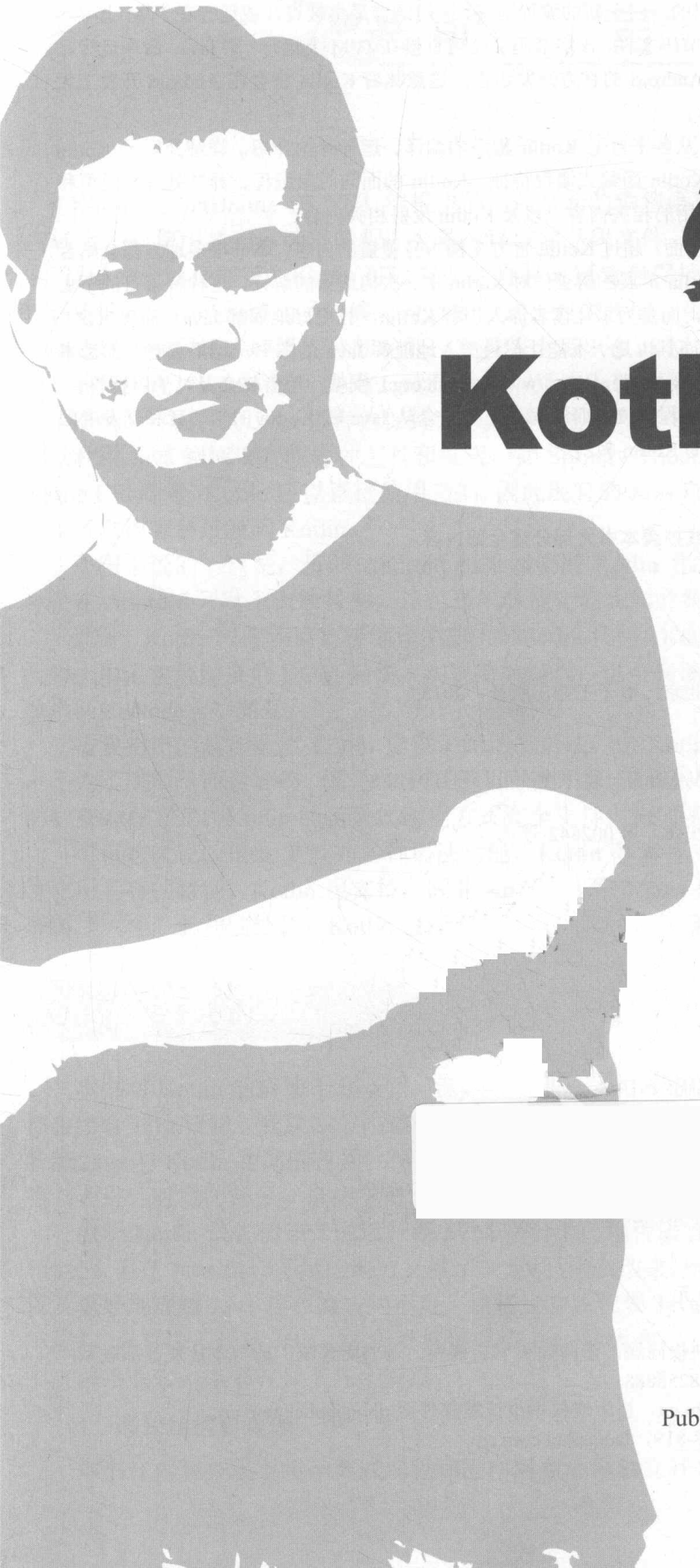
李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想
技术成就辉煌

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



疯狂 Kotlin讲义

李刚 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Kotlin 是 JetBrains 在 2011 年推出的一门全新的编程语言，这门语言最早被设计成运行在 JVM 上——使用 Kotlin 编写的程序会被编译成字节码文件，该字节码文件可直接在 JVM 上运行（用 java，命令运行）。目前 Google 已推荐使用 Kotlin 作为 Android 的官方开发语言，这意味着 Kotlin 将会在 Android 开发上大放异彩。

本书全面介绍了 Kotlin 的语法。从各平台上 Kotlin 程序的编译、运行开始介绍，详细介绍了 Kotlin 的基本语法，Kotlin 的数组和集合，Kotlin 函数式编程特征，Kotlin 的面向对象编程、异常处理、泛型和注解，还介绍了 Kotlin 与 Java 混合调用的相关内容，以及 Kotlin 反射相关内容。

本书对 Kotlin 的解读十分系统、全面，超过 Kotlin 官方文档本身覆盖的内容。本书很多地方都会结合 Java 字节码进行深入解读，比如对 Kotlin 扩展的解读，对 Kotlin 主、次构造器的解读，这种解读目的不止于教会读者简单地掌握 Kotlin 的用法，而是力求让读者深入理解 Kotlin，且更好地理解 Java。简单来说，本书不仅是一本 Kotlin 的学习图书，而且也是一本能让你更深入地理解 Java 的图书。如果读者在阅读本书时遇到了技术问题，可以登录疯狂 Java 联盟（http://www.crazyit.org）发帖，笔者将会及时予以解答。

本书为所有打算深入掌握 Kotlin 编程的读者而编写，尤其适合从 Java 转 Kotlin 的学习者和开发者阅读，也适合作为大学教育、培训机构的 Kotlin 教材。

图书在版编目（CIP）数据

疯狂 Kotlin 讲义 / 李刚编著. —北京：电子工业出版社，2018.2
ISBN 978-7-121-33459-7

Ⅰ. ①疯… Ⅱ. ①李… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

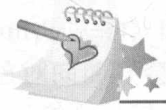
中国版本图书馆 CIP 数据核字（2018）第 002842 号

策划编辑：张月萍
责任编辑：葛　娜
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　1/16　　印张：20.25　　字数：571 千字
版　　次：2018 年 2 月第 1 版
印　　次：2018 年 2 月第 1 次印刷
印　　数：3500 册　　定价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。



前　言

Kotlin 是 JetBrains 在 2011 年推出的一门全新的编程语言，这门语言最早被设计成运行在 JVM 上——使用 Kotlin 编写的程序会被编译成字节码文件，该字节码文件可直接在 JVM 上运行（用 java 命令运行即可）。Kotlin 可以与现有的 Java 语言包保持完全兼容，而且 Kotlin 代码比 Java 代码更简洁。Kotlin 增加了扩展、对象表达式、对象声明、委托等 Java 原本不支持的功能，它们都是现代编程语言广泛支持的功能，并且完全可以在 JVM 上运行。

简单来说，Kotlin 既可利用 Java 的优势，又比 Java 更简洁。

Kotlin 与现有的 Java 语言包保持完全兼容，这意味着 Kotlin 不是一门简单的语言，它完全可以利用 Java 领域现有的各种工具和框架，如 Spring、Hibernate、MyBatis、Lucene、Hadoop、Spring Cloud 等。Kotlin 可以直接使用它们，因此现有的 Java 项目完全可以采用 Kotlin 开发，Java 开发者也很容易过渡到 Kotlin。

不得不说的一点是，目前 Android 已推荐使用 Kotlin 作为官方开发语言，这意味着 Kotlin 将会在 Android 开发上大放异彩，这也是笔者决定向大家介绍这门语言的重要原因之一。

此外，Kotlin 程序还可直接编译生成 JavaScript 代码，Kotlin 程序既可编译成前端 JavaScript 代码，用于实现网页的 DOM 操作，实现前端编程；也可编译成后端 JavaScript 代码，与服务端技术（如 Node.js）交互。

需要指出的是，虽然 Kotlin 提供了简洁的语法，但 Kotlin 的功能并不简单，Kotlin 从来就不是为了更简单而设计的，而是为了更强大而设计的。Kotlin 既支持函数式编程方式，也支持面向对象编程方式。Kotlin 的函数式编程方式完全支持主流的函数和闭包，语法功能非常丰富。

可以这么说：Java 支持的各种语法功能，Kotlin 基本都支持；Java 不支持的很多现代编程语言所具有的特征，Kotlin 也支持，因此 Kotlin 绝不比 Java 更简单。如果读者相信网络上某些所谓“大神”肤浅的结论：Kotlin 很简单，那么我建议你放弃阅读这本书。

本书有什么特点

本书对 Kotlin 的解读十分全面、深入，并非一本简单介绍 Kotlin 语法的图书，在很多地方都会结合 Java 语法、底层字节码进行讲解。如果读者有较好的 Java 功底，阅读本书能更清晰地看清 Java 与 Kotlin 之间的差异，便于快速上手 Kotlin；对于没有 Java 功底的读者，可选择忽略将二者进行对比的部分，直接学习本书也可掌握 Kotlin 语言的编程。

由于 Kotlin 最先被设计成运行在 JVM 平台上的编程语言，因此 Kotlin 具有和 Java 天然的相似性，但在 Java 设计不足的地方又做了大量的补充、改进，所以本书也能让你更好地理解 Java，以及更好地理解 Java 存在的一些不足。这样说并不代表 Java 不优秀，“知其雄，守其雌，为天下谿”，只有深入理解 Java，才能更好地感悟 Java 的优秀。

总结起来，本书有如下几个特点。

1. 逻辑结构更合理

本书在内容体系上将函数式编程和面向对象编程独立开来，先介绍函数式编程部分，再介

绍面向对象编程，更符合 Kotlin 语言本身的知识体系。而不像某些资料一会儿函数，一会儿面向对象，搅得读者晕头转向。实际上，无论是经典的图书如《C++ Primer》，还是 Swift 官方文档（甚至 Kotlin 官方文档），几乎都没见过一会儿函数、一会儿面向对象这样介绍的。

在介绍知识时，本书会先详细讲解各种知识点的理论，然后再通过示例演示 Kotlin 各理论的用法，将知识点融合在示例中，符合读者的认知、学习规律。

另外，整本书的知识具有和《疯狂 Java 讲义》大致相同的脉络，所以《疯狂 Java 讲义》的读者会很容易上手。

2. 讲解深入本质

Kotlin 是 JVM 语言，所以其很多东西其实是受到 Java 的影响的。书中会对一些看似奇怪的语法从字节码文件层次进行剖析，让读者更好地理解 Kotlin 与 Java 的对应关系。

比如，主构造器和次构造器到底是什么？Java 构造器并不区分主次，为什么 Kotlin 搞出这两个东西？主、次构造器为何要委托父类构造器？委托父类构造器时为什么存在区别？主、次构造器生成字节码之后到底对应 Java 的哪个部分？这些知识在本书 7.5 节有深入讲解。

再比如，Java 本身不支持扩展，那么 Kotlin 的扩展是如何在 JVM 上运行的？难道 Kotlin 改造了 JVM 吗？Java 本身不支持扩展，那么 Java 是否可以调用 Kotlin 扩展的成员？这些问题需要从字节码层次进行剖析，本书在 8.1 节有深入讲解。

还有，Java 泛型的上限、下限的本质是什么？Kotlin 泛型的声明处型变和使用处型变的本质是什么？与 Java 的对应关系是怎样的？Java 本身并不支持声明处型变，那为何 JVM 能支持 Kotlin 的声明处型变？这些问题可以在本书第 10 章中找到答案。

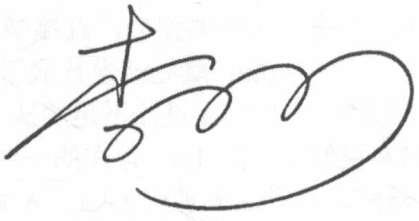
3. 知识内容更全面

本书内容超过 Kotlin 官方文档本身所覆盖的知识，比如介绍反射的章节就超过了 Kotlin 官方文档内容。本书反射部分不仅更详细地介绍如何获得类、函数、属性的引用，而且真正从 API 级别介绍 KClass、KCallable、KFunction、KProperty，KProperty0、KProperty1、KProperty2 的用法，以及它们的内在关联，并实实在在地教读者掌握如何用 Kotlin 反射动态创建对象、动态调用方法。

本书写给谁看

本书为所有打算深入掌握 Kotlin 编程的读者而编写，尤其适合从 Java 转 Kotlin 的学习者和开发者阅读，也适合作为大学教育、培训机构的 Kotlin 教材。

本书程序文件请从 www.broadview.com.cn/33459 下载。



2017-12-3

目 录 CONTENTS

CHAPTER

1

第1章 Kotlin语言与开发环境

本章要点

- Kotlin 语言简介
- 服务端 Kotlin 与客户端 Kotlin
- 下载和安装 Kotlin 的 SDK
- 编写第一个 Kotlin 程序
- 编译、运行 Kotlin 程序
- 使用 IntelliJ IDEA 编译、运行 Kotlin
- 使用 Eclipse 编译、运行 Kotlin

Kotlin 是 JetBrains 在 2011 年推出的一门全新的编程语言，这门语言最早被设计成运行在 JVM（Java 虚拟机）上——使用 Kotlin 编写的程序会被编译成字节码文件，该字节码文件可直接在 JVM 上运行(用 java 命令运行)。Kotlin 可以与现有的 Java 语言包保持 100%的兼容性，而且 Kotlin 代码比 Java 代码更简洁、更富有表现力。简单来说，一句话：Kotlin 既可利用 Java 的优势，又比 Java 更简洁。

此外，Kotlin 程序还可直接编译生成 JavaScript 代码——Kotlin 程序既可编译成前端 JavaScript 代码，用于实现网页的 DOM 操作，实现前端编程；也可编译成后端 JavaScript 代码，与服务端技术（如 Node.js）交互。

不得不说的一点是，目前 Android 已推荐使用 Kotlin 作为官方开发语言，这意味着 Kotlin 将会在 Android 开发中大放异彩，这也是笔者决定向读者介绍这门语言的重要原因之一。

1.1 Kotlin 语言简介

1.1.1 服务端的 Kotlin

Kotlin 程序可以编译成 Java 字节码文件，字节码文件可以直接在 JVM 上运行，因此 Kotlin 非常适合开发后端应用程序。Kotlin 与现有的 Java 语言包能保持完全兼容，这意味着 Kotlin 不是一门简单的语言，它完全可以利用 Java 领域现有的各种技术框架，如 Spring、Hibernate、MyBatis、Lucene 等，因此 Java 开发者非常容易过渡到使用 Kotlin。

总结来看，Kotlin 的显著优势有如下几点。

- **简洁性**：这是我们选择使用 Kotlin 的最大动力。Kotlin 具有大量现代编程语言的简洁性和便捷性，因此 Kotlin 被誉为 Android 平台的 Swift。
- **兼容性**：Kotlin 完全兼容 Java，因此 Kotlin 既是一门新的语言，也不是一门“全新”的语言，Kotlin 可以自由使用 Java 领域的无数框架和库。因此，开发者既可保持熟悉的技术栈，又可获得现代编程语言的优势。
- **迁移性**：Kotlin 支持大型项目从 Java 向 Kotlin 逐步迁移——项目主体部分继续使用 Java，新开发的功能可使用 Kotlin 编写，也可逐步使用 Kotlin 代替部分老旧的 Java 代码。

短期内，Kotlin 不会对 Java 造成巨大的冲击，但 Kotlin 简洁、优雅的语法可以对 Java 形成良好的补充，开发者可根据需要自由选择 Java 或 Kotlin，最终都会生成字节码文件，运行于 JVM 平台上。

1.1.2 使用 Kotlin 开发 Android 应用

Google 官方推荐使用 Kotlin 作为 Android 开发语言，证明了 Kotlin 非常适合开发 Android 应用。使用 Kotlin 开发 Android 应用可充分利用 Kotlin 的简洁性和便捷性。

Kotlin 完全兼容 JDK 1.6，因此保证了基于 Kotlin 开发的 Android 应用完全可以在较老的 Android 设备上运行。

对于广大的 Android 应用开发者而言，大部分时候都是与 Android 应用程序框架层交互的，调用 Android 应用程序框架层的 API，而 Kotlin 可以自由调用 Java 的各种类库，因此使用 Kotlin 调用 Android 应用程序框架层来开发应用程序甚至无须额外学习，开发者可以无缝地过渡为使用 Kotlin 开发。

对于性能方面来说，Kotlin 编译的字节码与 Java 原生字节码极为相似。随着 Kotlin 对内联函数的支持，使用 Lambda 表达式的代码通常比用 Java 写的代码运行得更快。

正是基于以上两点主要优势，Google 官方推荐使用 Kotlin 作为 Android 开发语言。

1.1.3 Kotlin 用于 JavaScript

Kotlin 程序还可以编译成 JavaScript 代码，Kotlin 程序会生成遵守 ECMAScript 规范的 JavaScript 代码。当选择生成 JavaScript 目标时，不仅会包括开发者自己写的 Kotlin 代码，也会包括 Kotlin 附带的标准库，它们都会转换为 JavaScript。

Kotlin 既可生成前端使用的 JavaScript 代码，也可生成后端使用的 JavaScript 代码。

当生成前端 JavaScript 代码时，Kotlin 可实现如下功能。

- Kotlin 提供了大量 API 来操作 DOM（文档对象模型），允许通过 DOM API 来动态创建和更新页面。
- Kotlin 也提供了支持 WebGL 的 API，因此可以在网页上用 WebGL 创建图形元素。
- Kotlin 也可使用现有的前端库和框架，如 jQuery 和 ReactJS 等。

当生成后端 JavaScript 代码时，Kotlin 完全可以与后端 JavaScript（如 Node.js）进行交互。

本书重点介绍 Kotlin 生成 JVM 字节码，这种 Kotlin 程序完全可以兼容 Java 程序，这也是 Kotlin 开发 Android 应用的基础。

1.2 使用命令行编译、运行 Kotlin

下面先介绍使用命令行环境编译、运行 Kotlin。与其他编程语言类似，Kotlin 同样需要下载并安装对应的 SDK，接下来即可使用该 SDK 提供的命令来编译、运行 Kotlin 程序。

1.2.1 下载和安装 Kotlin 的 SDK

下载和安装 Kotlin SDK 请按如下步骤进行。

① 登录 https://github.com/JetBrains/kotlin/releases，看到 Kotlin SDK 的最新发布版本，即可看到如图 1.1 所示的页面。本书成书之时，Kotlin SDK 的最新版本是 1.1.4-2，本书所有的案例也是基于该版本 SDK 的。

图 1.1 下载 Kotlin SDK 的页面

② 单击如图 1.1 所示页面中的链接，即可下载得到一个 kotlin-compiler-1.1.4-2.zip 压缩包文件。

③ 将该压缩包文件解压缩到任意目录下，即可看到如下文件路径。

- bin：该路径下存放了 Kotlin SDK 的各种工具命令，常用的 kotlinc、kotlin 等命令就放在该路径下。
- lib：该目录下包含了 Kotlin 的各种工具 JAR 包。
- license：存放与 Kotlin 项目相关的各种授权文档。

在上面路径中，bin 是一个非常有用的路径，在这个路径下包含了编译和运行 Kotlin 程序的 kotlinc 和 kotlin 两个命令。除此之外，还包含了 kotlinc-jvm、kotlinc-js 等工具命令。其中 kotlinc 和 kotlinc-jvm 是一样的，都用于将 Kotlin 程序编译成适用于 JVM 的字节码文件，而 kotlinc-js 则用于将 Kotlin 程序编译成 JavaScript 代码。

④ 将解压缩路径下的 bin 目录添加到系统的 PATH 环境变量中，这样即可保证操作系统能通过 PATH 环境变量找到 kotlinc、kotlin 等工具命令。

⑤ 将解压缩路径下的 lib 目录中的 kotlin-stdlib.jar 文件（该文件与 kotlin-runtime.jar 文件相同，都代表了 Kotlin 的运行时环境库）添加到 CLASSPATH 环境变量中，这样即可保证运行 Kotlin 程序时能正常加载 Kotlin 的运行时环境库。

提示：

本书假设读者已有最基本的 Java 编程基础，因此书中不会逐步介绍环境变量配置这些基础内容。如果读者需要学习环境变量配置这些基础知识，则建议参考《疯狂 Java 讲义》一书。

1.2.2 第一个 Kotlin 程序

编辑 Kotlin 源代码可以使用任何无格式的文本编辑器，在 Windows 操作系统上可使用记事本（NotePad）、EditPlus 等程序，在 Linux 平台上可使用 VI 工具等。

提示：

编写 Kotlin 程序不要使用写字板，更不可使用 Word 等文档编辑器。因为写字板、Word 等工具是有格式的编辑器，当使用它们编辑一个文档时，这个文档中会包含一些隐藏的格式化字符，这些隐藏字符会导致程序无法正常编译、运行。

在记事本中新建一个文本文件，并在该文件中输入如下代码。

程序清单：codes\01\1.2\helloWorld.kt

```
fun main(args: Array<String>) {
    println("Hello, World!")
}
```

编辑上面的 Kotlin 文件时，注意程序中粗体字标识的单词，Kotlin 程序严格区分大小写。将上面文本文件保存为 helloWorld.kt，该文件就是 Kotlin 程序的源程序，Kotlin 源程序的文件名要求以.kt 结尾。

Kotlin 程序与 Java 程序不同，Kotlin 支持函数式编程，因此 Kotlin 程序只需要一个 main()函数作为程序入口，不需要将该 main()函数放在任何类中。

提示：

仔细看一下上面的main()函数，不难发现这个main函数其实就是Java主类中的main()方法的变体。只不过在Kotlin语言中，函数也是一等公民，因此函数可以独立存在，而Java主类中的main()方法必须放在类中声明。Java主类中的main()方法需要声明一个String[] args形参，而Kotlin的mian()函数同样声明了args: Array<String>形参——它们都是一个字符串数组类型的形参，这表明它们的本质是一样的，只是形式不同而已。至于Kotlin主函数前面的fun关键字，它专门用于声明函数；Swift语言还使用func关键字声明函数呢，其实没有什么特别的，习惯不同而已。

编写好Kotlin程序的源代码后，接下来就应该编译该Kotlin源文件来生成字节码文件了。

1.2.3 编译、运行Kotlin程序

编译Kotlin程序需要使用kotlinc或kotlinc-jvm命令（两个命令完全一样），因为前面已经把kotlinc命令所在的路径添加到了系统的PATH环境变量中，因此现在可以使用kotlinc命令来编译Kotlin程序。

对于初学者而言，先掌握kotlinc命令的如下用法：

```
kotlinc -d destdir srcFile
```

在上面命令中，-d destdir是kotlinc命令的选项，用以指定编译生成的字节码文件的存放路径，destdir只需是本地磁盘上的一个有效路径即可；而srcFile是Kotlin源文件所在的位置，这个位置既可以是绝对路径，也可以是相对路径。

通常，总是将所生成的字节码文件放在当前路径下，当前路径可以用一点（.）来表示。在命令行窗口进入helloWorld.kt文件所在的路径，在该路径下输入如下命令：

```
kotlinc -d . helloWorld.kt
```

运行该命令后，在该路径下生成一个HelloWorldKt.class文件。

与javac编译器类似，使用kotlinc编译文件只需要指定存放目标文件的位置即可，无须指定字节码文件的文件名。因为使用kotlinc编译后生成的字节码文件有默认的文件名：如果Kotlin源程序中包含一个或多个函数，kotlinc会额外生成一个文件，该生成文件的文件名是源文件名首字母大写并添加Kt后缀，以.class作为扩展名；如果Kotlin源程序中包含一个或多个类，kotlinc则会为每个类生成一个字节码文件，文件名以源文件所定义的各个类的类名作为主文件名，以.class作为扩展名。

事实上，指定目标文件存放位置的-d选项也是可以省略的，如果省略该选项，则意味着将生成的字节码文件放在当前路径下。

可能有读者会感觉到：这不就是《疯狂Java讲义》所介绍的内容吗？kotlinc的用法与javac的用法何其相似啊！实际上这也是Kotlin设计者希望实现的效果：Java是一门如此深入人心、影响深远的语言，拥有广泛的程序员基础，而Kotlin不仅借助于JVM作为运行平台，而且保持和Java相似的编译、运行方式，这样Kotlin也更容易被广大程序员所接受。

运行Kotlin程序可使用kotlin或java命令（你没看错，可以用java命令），启动命令行窗口，进入HelloWorldKt.class所在的位置，在命令行输入如下命令：

```
java HelloWorldKt
```

运行上面命令，将看到如下输出：

```
Hello World!
```

这表明 Kotlin 程序运行成功。

当然，Kotlin 也专门提供了一个 kotlin 命令来运行 Kotlin 程序，因此在命令行输入如下命令：

```
kotlin HelloWorldKt
```

运行上面命令，同样可以看到如下输出：

```
Hello World!
```

1.3 使用 IntelliJ IDEA 编译、运行 Kotlin

Kotlin 本身就是 JetBrains 开发的，而 IntelliJ IDEA 则是 JetBrains 自家开发的 IDE 工具，因此 IntelliJ IDEA 必然支持 Kotlin 语言。

IntelliJ IDEA 是一个商业版软件，使用该软件的商业版是要收费的，但 JetBrains 还为 IntelliJ IDEA 提供了一个免费的社区版，因此本书会基于免费的社区版 IntelliJ IDEA 进行介绍。

登录 https://www.jetbrains.com，下载并安装最新的社区版 IntelliJ IDEA。

通过 IntelliJ IDEA 新建一个支持 Kotlin 的 Java 项目，请按如下步骤进行。

① 单击 IntelliJ IDEA 的“File”→“New”→“Project...”新建一个 Java 项目，并为新建的项目勾选“Kotlin/JVM”复选框，如图 1.2 所示。

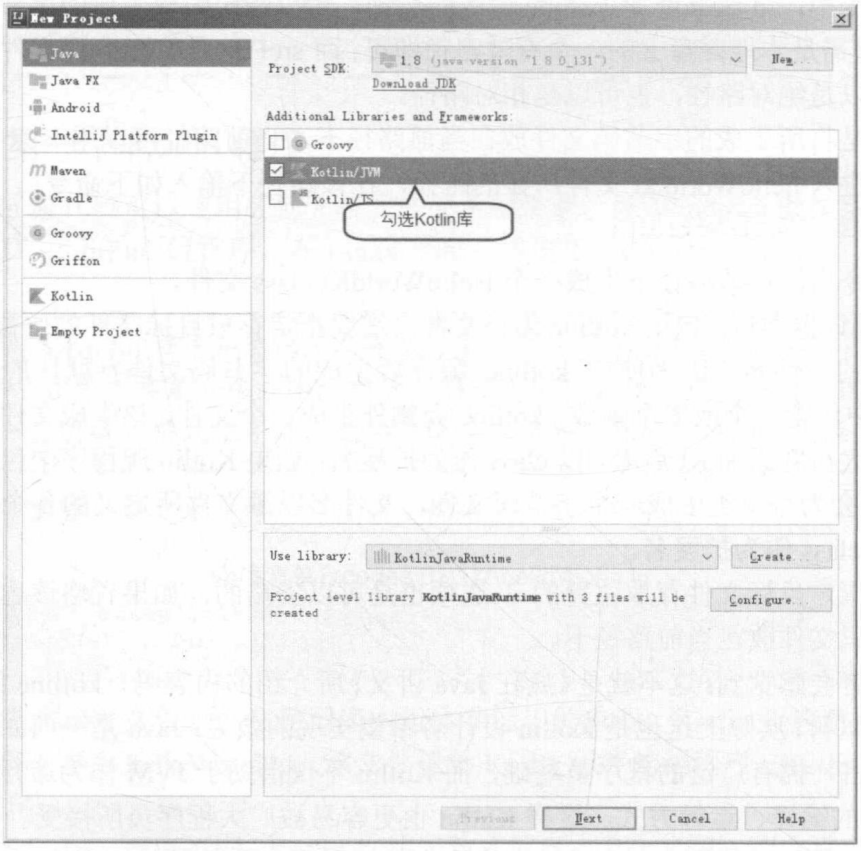


图 1.2 新建支持 Kotlin 的 Java 项目

② 让 Java 项目支持 Kotlin 的关键就是勾选“Kotlin/JVM”复选框，项目创建完成后可看到如图 1.3 所示的项目结构。

其实让 Java 项目支持 Kotlin 也很简单，无非就是需要两个条件：

- 该项目“知道”使用 kotlinc 编译器来编译 Kotlin 程序。
- 该项目包含了 Kotlin 的运行时环境。

正如从图 1.3 所看到的，上面项目中包含了 kotlin-stdlib.jar，这就是 Kotlin 的运行时环境库（即本章 1.2 节让大家添加到 CLASSPATH 环境变量中的 JAR 包），而 kotlin-reflect.jar 是 Kotlin 反射才需要的 JAR 包，kotlin-stdlib-jre7.jar、kotlin-stdlib-jre8.jar 是 Kotlin 为支持 Java 7、Java 8 提供的运行时环境库。此外，由于 IntelliJ IDEA 本身就是 JetBrains 自家的 IDE 工具，让它集成一个 kotlinc 编译器是顺理成章的事——这就是该项目能支持 Kotlin 程序的根本所在。

③ 右键单击图 1.3 所示项目结构中的 src 目录，在弹出的快捷菜单中选择“New”→“Kotlin File/Class”菜单项，如图 1.4 所示。

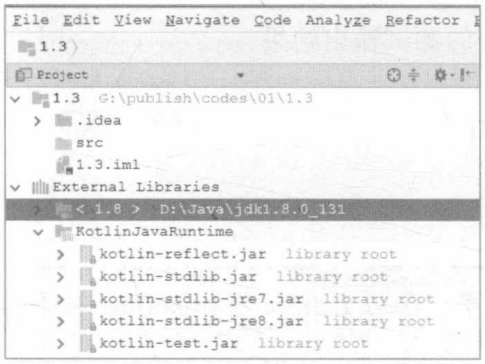


图 1.3 支持 Kotlin 的 Java 项目

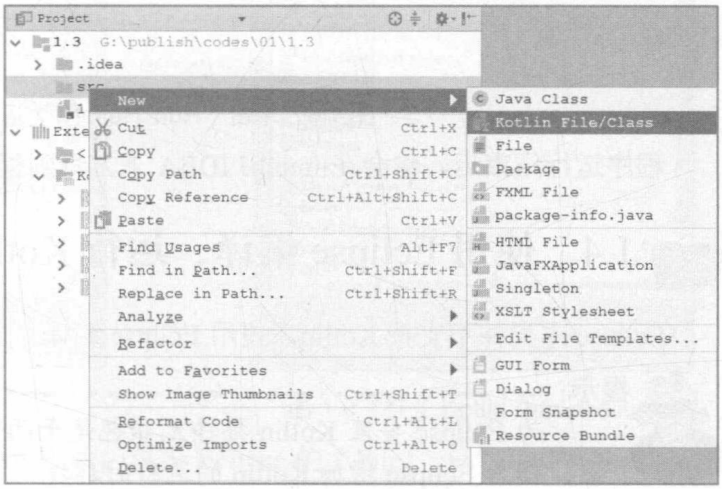


图 1.4 新建 Kotlin 文件

④ 系统弹出如图 1.5 所示的对话框，在该对话框中可以选择 Kotlin 文件的类型。

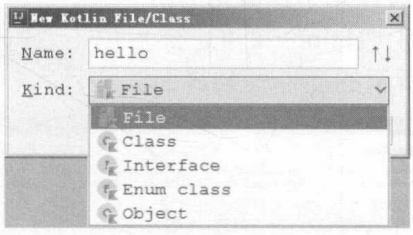


图 1.5 选择 Kotlin 文件的类型

对于熟悉 Java 的读者来说，对类、接口和枚举这些文件类型太熟悉了，它们都是 Java 程序员的老朋友。但在这里不打算选择这些老朋友，而是直接选择 File 类型——这表明将会新建一个普通的 Kotlin 文件。正如前面所介绍的，Kotlin 支持函数式编程，函数也是 Kotlin 的一等公民，因此此处将直接在 Kotlin 程序中新建主函数作为程序入口。

⑤ 在 hello.kt 文件中添加一个主函数，IntelliJ IDEA 提供了一个快速完成此操作的模板，只需输入 main，然后按 Tab 键，即可在编辑界面中看到添加了 main()函数。

在 main()函数中添加一行简单的输出语句用于测试。

⑥ 在 hello.kt 的编辑界面中单击鼠标右键，系统弹出如图 1.6 所示的快捷菜单，单击该菜单中的“Run 'HelloKt'”菜单项即可运行该程序。

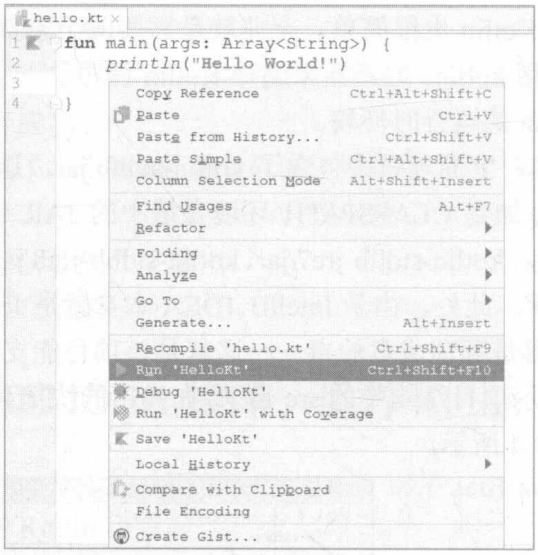


图 1.6 单击“Run 'HelloKt'”运行 Kotlin 程序

程序运行结束后，将会在 IntelliJ IDEA 下方看到程序的测试输出结果。

1.4 使用 Eclipse 编译、运行 Kotlin

Eclipse 本身并不支持 Kotlin，使用 Eclipse 编译、运行 Kotlin 需要安装插件。

提示：

为 Eclipse 安装 Kotlin 插件无非也是干两件事：①为 Eclipse 增加 kotlinc 编译器；②为 Eclipse 增加 Kotlin 的运行时环境。

由于本书假设读者已有一定的 Java 基础，因此关于 Eclipse 的安装步骤本书就不介绍了。为 Eclipse 安装 Kotlin 插件请按如下步骤进行。

① 单击 Eclipse 主菜单中的“Help”菜单，然后单击“Eclipse Marketplace”菜单项，即可看到 Eclipse 弹出的如图 1.7 所示的插件安装界面。

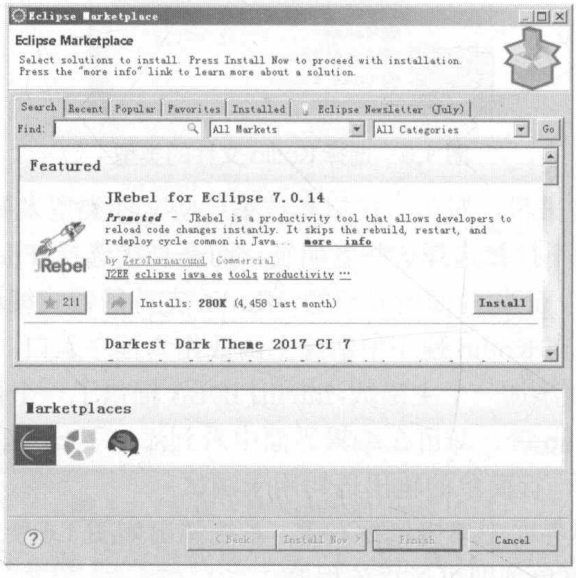


图 1.7 Eclipse 插件安装界面