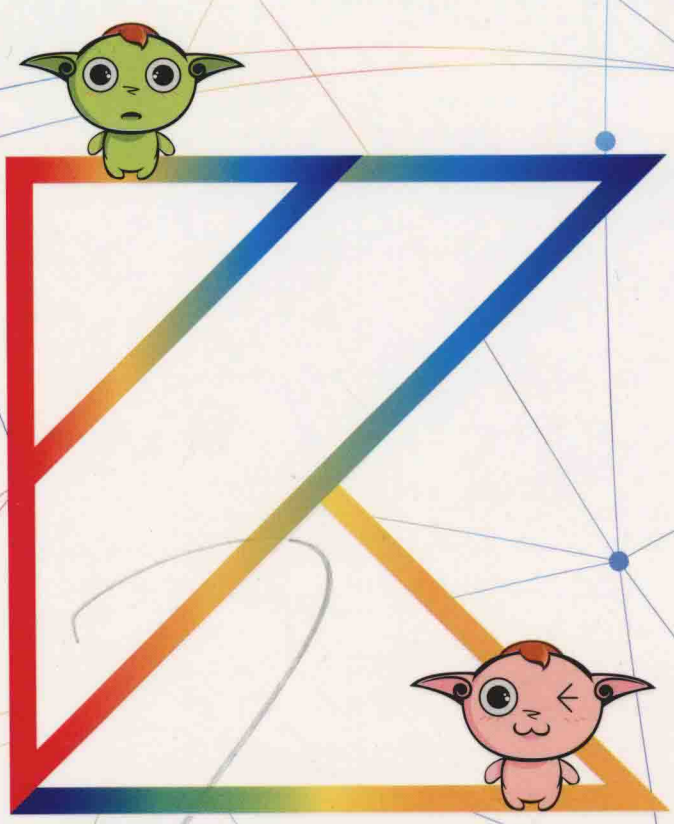
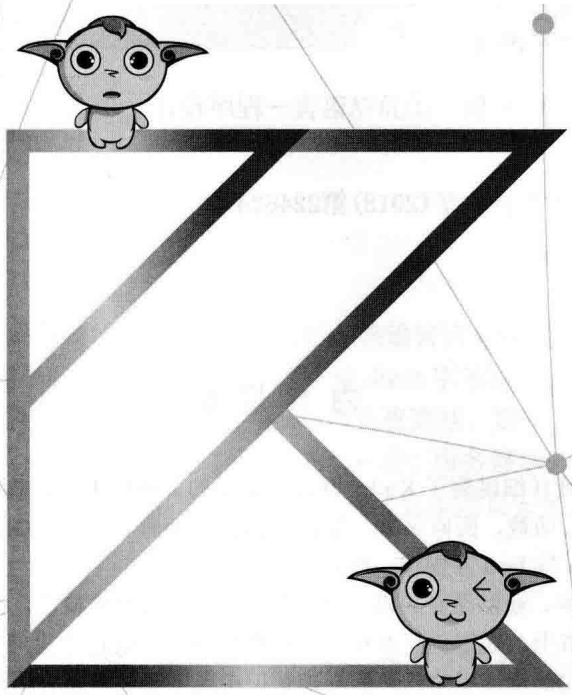


Kotlin

从基础到实战

黑马程序员 编著

中国工信出版集团 人民邮电出版社
POSTS & TELECOM PRESS



Kotlin
从基础到实战

黑马程序员 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

Kotlin从基础到实战 / 黑马程序员编著. -- 北京 : 人民邮电出版社, 2019.3
ISBN 978-7-115-49440-5

Ⅰ. ①K… Ⅱ. ①黑… 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224624号

内 容 提 要

本书从初学者的角度详细讲解了 Kotlin 开发中常用的多种技术。全书共 13 章，内容包括 Kotlin 入门、Kotlin 编程基础、函数、面向对象、集合、lambda 编程、泛型、Gradle、协程、“坦克大战”游戏开发、DSL、Kotlin 与 Java 互操作、时钟。

本书通过典型的案例、通俗易懂的语言阐述面向对象中的抽象概念，在集合、Lambda 编程、泛型、Gradle、协程等章节中，通过剖析案例、分析代码结构、解决常见问题等方式，帮助初学者培养良好的编程习惯。第 10 章运用前几章的基础知识实现了一个坦克大战的游戏案例开发。第 11～13 章分别介绍了 DSL、Kotlin 与 Java 进行互操作以及通过 Kotlin 语言实现一个 JavaScript 语言的时钟项目，帮助初学者掌握 Kotlin 语言与 Java 语言、JavaScript 语言的互操作。

本书既可作为高等院校本、专科计算机相关专业的教材，也可作为社会培训教材，是一本适合广大编程爱好者参考和学习的书籍。为了帮助编程者更好地学习本书中的内容，本书还提供了配套的源代码与视频等资源，方便读者学习。

◆ 编　　著　黑马程序员
责任编辑　范博涛
责任印制　马振武

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：18.25　　2019 年 3 月第 1 版
字数：458 千字　　2019 年 3 月北京第 1 次印刷

定价：59.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前言 FOREWORD

为什么要学习 Kotlin

Kotlin 是 JetBrains 公司开发的基于 JVM 的语言。该语言完全兼容 Java 的特性，并且已经正式成为 Android 官方支持的开发语言。它可以编译成 Java 字节码，也可以编译成 JavaScript 字节码，方便在没有 JVM 的设备上运行。它比 Java 语言更简洁、更安全、易扩展，能够静态检测常见陷阱，也可以应用于 Android 开发、JavaScript 开发、服务器端开发的程序中。由于从实际使用效果来说，Kotlin 语言比 Java 语言的开发效率高并且使用更安全，因此 Kotlin 语言的应用越来越广泛。

如何使用本书

本书是一本 Kotlin 基础入门书籍，使用 IntelliJ IDEA 作为开发工具。作为一种技术的入门图书，最重要也最难办的一件事是将一些非常复杂、难以理解的思想和问题简单化，让初学者能够轻松理解并快速掌握。本书先对每个知识点都进行了深入的分析，并针对每个知识点精心设计了相关案例，然后模拟这些知识点在实际工作中的运用，真正做到了学习过程的由浅入深、由易到难。初学者在使用本书时，建议从头开始循序渐进地学习，并且反复练习书中案例，以达到熟能生巧。

本书共分为 13 章，接下来分别对每章进行简单的介绍，具体内容如下。

- 第 1 章主要讲解 Kotlin 语言的特性与 IntelliJ IDEA 工具的安装。通过对本章的学习，初学者能够掌握 IntelliJ IDEA 的安装过程，动手实现属于自己的第一个 Kotlin 程序。
- 第 2 章主要讲解 Kotlin 语言的基本语法，不论任何一门语言，其基本语法都是最重要的内容。在学习基本语法时，一定要做到认真学习每一个知识点，切忌走马观花。
- 第 3 章主要讲解函数，包括函数的分类与使用。通过对本章的学习，初学者可以了解函数的定义以及如何使用不同类型的函数。
- 第 4 章主要讲解 Kotlin 语言最重要的特征——面向对象。本章内容以编程思想为主，初学者需要花费很大的精力来理解本章所讲解的内容。
- 第 5～7 章主要讲解 Kotlin 中的集合、Lambda 编程以及泛型，包括集合中的 List 接口、Set 接口、Map 接口、Lambda 表达式、高阶函数、内联函数、泛型的约束、协变与逆变等，这几章的内容非常重要，在后续 Kotlin 程序开发中会经常用到，因此，要求初学者一定要熟练掌握这部分内容。
- 第 8 章主要讲解 Gradle，包括如何创建 Gradle 程序、Gradle 的任务、Gradle 的依赖、Gradle 的扩展。通过对本章的学习，初学者可以完成简单的 Gradle 程序开发。
- 第 9 章主要讲解协程，包括协程的挂起、主协程、协程取消、管道等。通过对本章的学习，初学者可以掌握协程的基本操作与使用。
- 第 10 章主要讲解坦克大战游戏的开发，该游戏总结了第 1～9 章的知识点。通过对本章

的学习，初学者可以熟练运用 Kotlin 中的基础开发技术。

- 第 11～13 章主要讲解 DSL、Kotlin 与 Java 互操作以及时钟项目，包括 DSL 的使用、Kotlin 与 Java 相互调用、Kotlin 与 Java 互操作对比、使用 Kotlin 语言实现一个 JavaScript 语言的时钟项目。通过对这 3 章的学习，初学者可以掌握 Kotlin 与 Java 代码如何进行相互调用以及如何运用 Kotlin 语言实现一个 JavaScript 语言的项目。

在上面所提到的 13 章中，第 1～3 章主要是针对 Kotlin 的一些比较基础的知识进行详细的讲解，这些知识多而细，要求初学者深入理解，奠定好学习后面知识的基础。第 4～9 章中每个小节的知识点后都会提供一个实用的案例，并在案例后面对其进行详细的分析，初学者可以结合案例后的分析对案例进行学习，每一个案例都需要动手实践。第 10 章主要是总结第 1～9 章的知识点，实现了一个坦克大战游戏的开发。初学者学习本章时要求动手实现该游戏的全部效果。第 11 章的内容了解即可，第 12～13 章主要讲 Kotlin 与 Java 语言的交互以及如何运用 Kotlin 语言编写一个 JavaScript 语言的时钟项目，这两章的内容比较重要，需要初学者掌握并可以灵活运用其中的知识。

在学习本书时，首先要做到对知识点理解透彻，其次一定要亲自动手去练习书中所提供的案例，因为在学习软件编程的过程中动手实践是非常重要的。对于一些非常难以理解的知识点也可以选择通过案例的练习来学习。如果实在无法理解书中所讲解的知识，建议初学者不要纠结于某一个知识点，可以先往后学习，通常来讲，看到后面对知识点的讲解或者其他小节的内容后，前面看不懂的知识点一般就能理解了。

致谢

本书的编写和整理工作由传智播客教育科技有限公司完成，主要参与人员有吕春林、陈欢、柴永菲、闫文华、高美云、张泽华、吴通、肖琦、伍碧林、马伟奇等，研发小组全体成员在这近一年的编写过程中付出了很多辛勤的汗水，在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力，但书中难免会有不妥之处，欢迎各界专家和读者来信给予宝贵意见，我们将不胜感激。您在阅读本书时，如果发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至：itcast_book@vip.sina.com

黑马程序员

2018 年 10 月于北京

目 录 CONTENTS

1 Chapter

第 1 章 Kotlin 入门

学习目标

- 了解什么是 Kotlin
- 掌握 Kotlin 开发环境的搭建方法
- 掌握 Kotlin 程序的编写方法

Kotlin 是由 JetBrains 公司开发的，用于多平台应用的静态编程语言。2017 年谷歌 I/O 大会上 Android 团队宣布 Kotlin 成为其官方头等支持语言。它可以被编译成 Java 字节码，100%兼容 Java 语言，也可以被编译成 JavaScript，方便在没有 JVM 的设备上运行。它比 Java 更简洁、更安全，能够静态检测常见的陷阱。本章将针对 Kotlin 语言的前景、特性、开发环境以及如何编写 Kotlin 程序等内容进行详细讲解。

1.1 Kotlin 简介

1.1.1 Kotlin 的前景

Kotlin 语言由 JetBrains 公司开发，是一个基于 JVM 的新编程语言，它的语法格式比 Java 更加简洁，现在已经正式成为 Android 官方支持的开发语言，并且 100%兼容 Java 语言。目前 Kotlin 语言主要用于以下几个领域。

1. 服务端开发

Kotlin 语言非常适合开发服务端应用程序，并且与 Java 技术保持良好的兼容性，之前用 Java 语言做的服务端程序都可以使用 Kotlin 语言来代替。Kotlin 的革新式语言功能有助于构建强大而易于使用的程序。Kotlin 语言对协程的支持有助于构建服务器端程序，伸缩到适度的硬件要求以应对大量的客户端。Kotlin 语言与所有基于 Java 语言的框架完全兼容，可以让你保持熟悉的技术栈，同时获得更现代化的语言优势。

2. Android 开发

Kotlin 语言也适合开发 Android 程序。在兼容性方面，Kotlin 语言与 JDK 6 完全兼容，保证了 Kotlin 应用程序可以在较旧的 Android 设备上运行。在性能方面，由于 Kotlin 支持内联函数，

使用 Lambda 表达式的代码通常比使用 Java 的代码运行速度快，因此 Kotlin 应用程序的运行速度比 Java 快。在互操作性方面，Kotlin 与 Java 可进行 100%的互操作，在 Kotlin 应用程序中可以使用所有现有的 Android 库。在编译时长方面，Kotlin 支持高效的增量编译，所以对于清理构建会有额外的开销，增量构建通常与 Java 一样快或者更快。

3. JavaScript 开发

Kotlin 提供了 JavaScript 作为目标平台的能力。这种能力通过将 Kotlin 转换为 JavaScript 来实现，目前的实现目标是 ECMAScript 5.1。当选择 JavaScript 为目标时，作为项目部分的 Kotlin 代码以及 Kotlin 附带的标准库都会转换为 JavaScript。但不包括使用的 JDK、任何 JVM、Java 框架或库。所有非 Kotlin 文件在编译期间会被忽略掉。Kotlin 编译器遵循以下目标：提供最佳大小的输出；提供可读的 JavaScript 输出；提供与现有模块系统的互操作性；在标准库中提供相同的功能。

1.1.2 Kotlin 的特性

1. 简洁

在开发程序时，通常情况下开发人员会花费更多的时间去阅读现有代码。例如，需要在当前项目上添加新的功能，此时就需要阅读与当前功能相关的代码，而阅读代码的时间长短取决于代码量的多少。在 Kotlin 程序中，由于代码简洁，从而大大减少了样板代码的数量，因此在后续阅读代码时会更加简便，这就提高了工作效率，进而可以更快地完成任务。在许多情况下，IDEA 工具将自动检测到可以用更简洁的结构替换公共的代码模式，并提供修复代码的方法，通过研究这些自动修复所使用的语言特性，可以在开发程序时灵活应用这些特性。

2. 安全

一般情况下，为了保证编程语言的安全性，在设计程序时会尽量避免出错的可能，当然这并不能保证程序绝对不会出现问题。防止错误的发生通常以牺牲成本为代价，需要给编译器更多关于程序的预期操作信息，这样编译器就可以验证与程序所做的匹配信息是否一致。

Kotlin 试图用较小的成本获取比 Java 更高级别的安全性。在 JVM 上运行的程序已经提供了许多安全机制，例如，防止内存泄露、防止缓冲区溢出以及由于不正确使用动态分配内存造成的其他问题等。Kotlin 作为一种静态语言，在 JVM 上也保证应用程序的类型安全，不必指定所有类型的声明。很多情况下，编译器会自动推断类型。此外，Kotlin 允许定义可空类型变量，并提供了多种方式对空数据进行处理，这样可以避免程序的空指针异常（NullPointerException），从而大大降低了程序崩溃的可能性。

3. 互操作性

Kotlin 与 Java 的互操作性，表现在 Kotlin 程序可以调用 Java 中的方法、扩展 Java 类、实现 Java 中的接口以及使用 Java 语言来注释 Kotlin 程序等。Kotlin 中的类和方法可以完全像普通的 Java 类和方法一样去调用，这样 Java 代码与 Kotlin 代码可以在项目中的任何地方进行互调。其重点体现在使用现有的 Java 标准库扩展 Java 中的功能，使 Kotlin 程序使用起来更方便。

Kotlin 的开发工具 IDEA 还提供了跨语言项目的全力支持，它不仅可以编译 Java 源文件，而且还可以使 Java 与 Kotlin 进行任意的组合。IDEA 工具的跨语言功能，允许程序执行如下操作。

- 自由组合 Kotlin 语言与 Java 源文件。

- 调试混合语言项目，并在不同语言编写的程序之间进行互操作。

1.2 Kotlin 开发环境搭建

1.2.1 Kotlin 常用开发工具

在 Kotlin 的官方文档（www.kotlincn.net）上可以看到，Kotlin 语言的开发工具有 4 种类型，分别是 IntelliJ IDEA、Android Studio、Eclipse 以及 Compiler。这 4 种工具的简单介绍如下。

1. IntelliJ IDEA

IntelliJ IDEA 是 JetBrains 公司开发的，是 Kotlin 官方推荐使用的开发工具。在 Kotlin 官网上下载最新版本的 IntelliJ IDEA，已经默认安装了 Kotlin 插件。如果下载的 IntelliJ IDEA 没有 Kotlin 插件，则可以打开 IntelliJ IDEA 的插件安装界面，完成插件的安装或升级。

2. Android Studio

Android Studio 是谷歌公司基于 IntelliJ IDEA 开发的一个工具，主要用于 Android 程序的开发。Android Studio 从 3.0 版本开始内置安装 Kotlin 插件。如果使用的是 3.0 之前的版本，则可以通过 Android Studio 的插件安装界面完成 Kotlin 插件的安装，插件安装完成后需要重新启动 Android Studio。

3. Eclipse

Eclipse 是一款经典的开发工具，虽然它是由 Java 语言开发的，但它不仅支持 Java 语言，而且还支持 C/C++、COBOL、PHP、Android 等编程语言，现在还支持 Kotlin 语言。如果想要在 Eclipse 工具中开发 Kotlin 语言程序，则需要安装 Kotlin 插件。

4. Compiler

Compiler 是一个命令行的编译器，在 Kotlin 官网上也可以下载这个工具，然后通过命令行来编译 Kotlin 程序。

以上 4 种工具中，Android Studio 是在 IntelliJ IDEA 工具的基础上添加了一些针对 Android 开发的插件，这些插件在开发 Kotlin 语言的程序中是用不到的；Compiler 工具用起来不太方便；Eclipse 与 IntelliJ IDEA 工具开发 Kotlin 语言都比较方便。由于本书主要讲解 Kotlin 语言的开发，因此选择 Kotlin 官方推荐的工具 IntelliJ IDEA。

1.2.2 IntelliJ IDEA 的安装

1. 下载 IntelliJ IDEA

首先打开 JetBrains 公司官网，单击界面上的【DownLoad】按钮进入到下载界面，在这个界面上有两个选项，分别为 Ultimate 企业版（免费试用）与 Community 社区版（免费开源），这里选择 Community 版本的 IntelliJ IDEA，点击【下载】按钮进行下载，此处下载的是最新版本的 IDEA，如图 1-1 所示。

需要注意的是，由于目前最新版本 ideaIC-2018.1.5 相对来说没有 ideaIC-2017.3.5 版本稳定，因此在图 1-1 所示界面中选择【Previous Version】链接，跳转到 IDEA 版本页面，选择 ideaIC-2017.3.5 版本进行下载，也可以根据个人习惯下载不同版本，如图 1-2 所示。

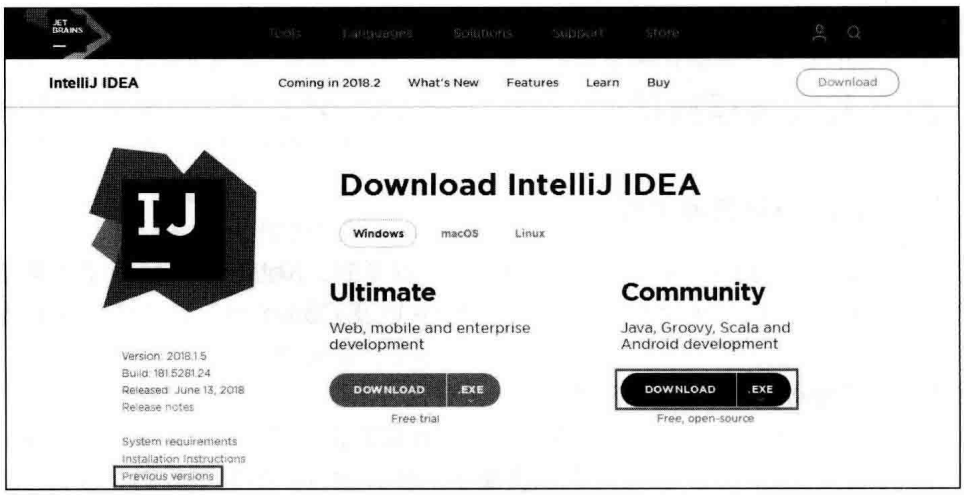


图1–1　下载界面

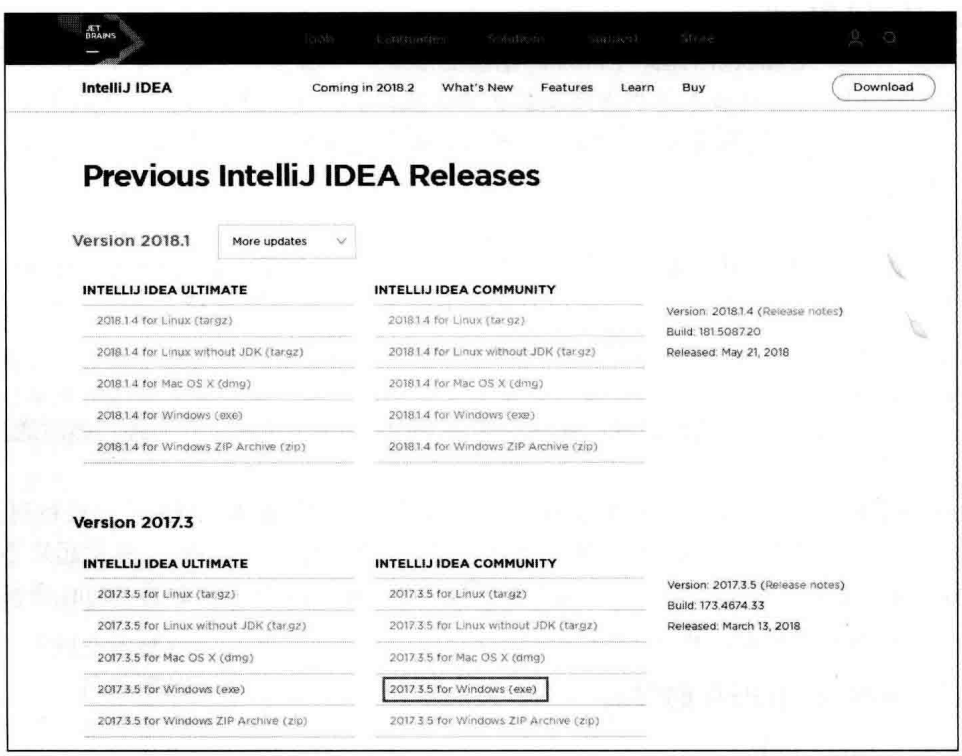


图1–2　下载界面

2. 安装 IntelliJ IDEA

在安装 IntelliJ IDEA 工具的过程中，可根据个人喜好选择程序的安装位置，如图 1–3 所示。

在图 1–3 所示界面中，单击【Next】按钮，进入安装设置界面。在该界面 Create Desktop Shortcut 下方有两个复选框，用于选择计算机系统位数，分别是【32–bit launcher】和【64–bit launcher】，根据相应的系统位数（右键单击【我的电脑】，单击【属性】，查看系统位数）选择即可，如图 1–4 所示。

在图 1–4 所示界面中，单击【Next】按钮，等待程序进行安装，最后弹出一个安装完成的对话框。在这个对话框上单击【Finish】按钮即可完成 IntelliJ IDEA 工具的安装，安装完成的对

话框如图 1-5 所示。

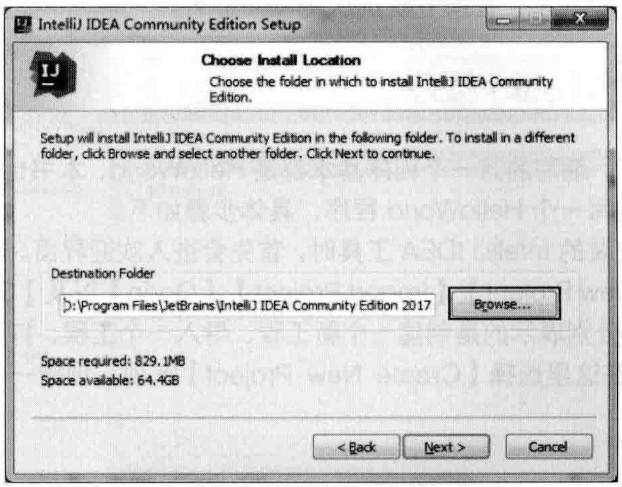


图1-3 选择安装位置

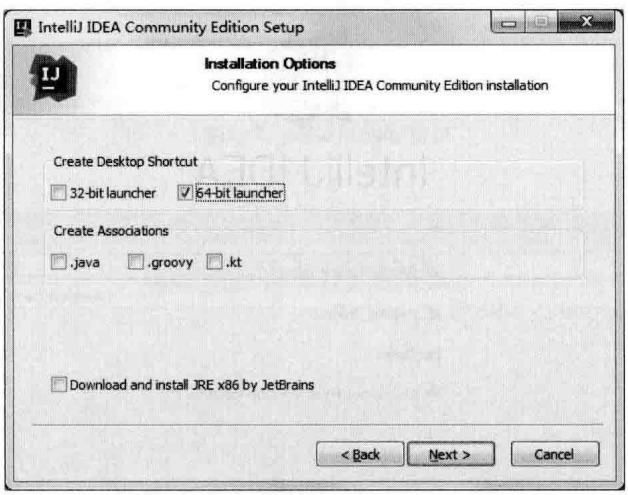


图1-4 选择操作系统版本

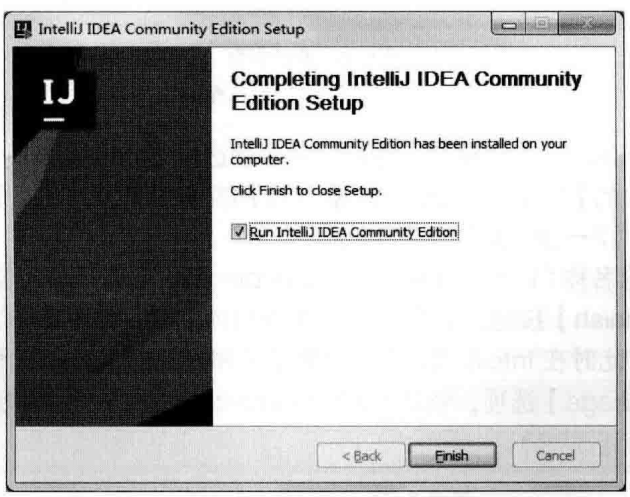


图1-5 安装完成

需要注意的是，安装完 IntelliJ IDEA 工具之后，还需要安装 1.6 以上版本的 JDK，在这里就不一一截图显示安装的过程了，直接下载 JDK 并安装即可。

1.3 开发第一个 Kotlin 程序

接触一门新语言时，编写的第一个程序基本都是 HelloWorld，本书也不例外。本小节就教大家如何用 Kotlin 语言编写一个 HelloWorld 程序，具体步骤如下。

当第一次打开新安装的 IntelliJ IDEA 工具时，首先会进入欢迎界面。在这个界面上有 4 个选项，分别是【Create New Project】、【Import Project】、【Open】以及【Check out from Version Control】。这 4 个选项分别表示的是创建一个新工程、导入一个工程、打开文件夹以及从 svn 或 git 上获取一个工程。在这里选择【Create New Project】选项，创建一个新的工程，如图 1-6 所示。

图1-6 欢迎界面

接着会弹出一个 New Project 窗口，在窗口的左侧选中【Java】选项，在 Project SDK 对应的选项框中，点击后边的【New...】按钮，选择 JDK 的安装位置，勾选上【Kotlin/JVM】复选框，单击【Next】按钮进入下一步，如图 1-7 所示。

最后设置该项目的名称（Project name）为 Chapter01，项目存放的位置（Project location）可自行设置，单击【Finish】按钮完成 Chapter01 项目的创建，如图 1-8 所示。

项目创建完成了，此时在 IntelliJ IDEA 中会显示创建好的 Chapter01 程序，右键单击【src】，选择【New】→【Package】选项，创建 com.itheima.chapter01 包，如图 1-9 所示。

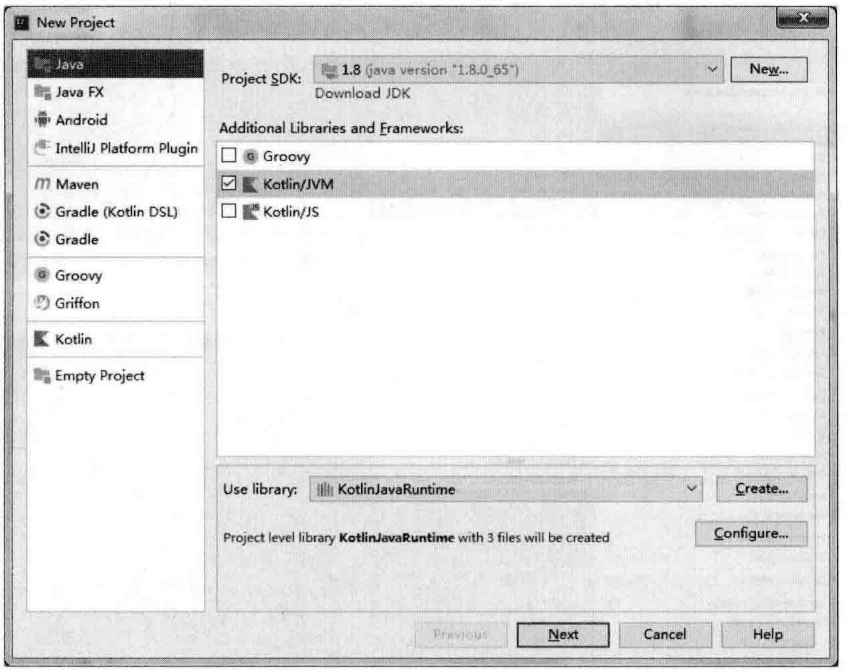


图1-7 New Project窗口

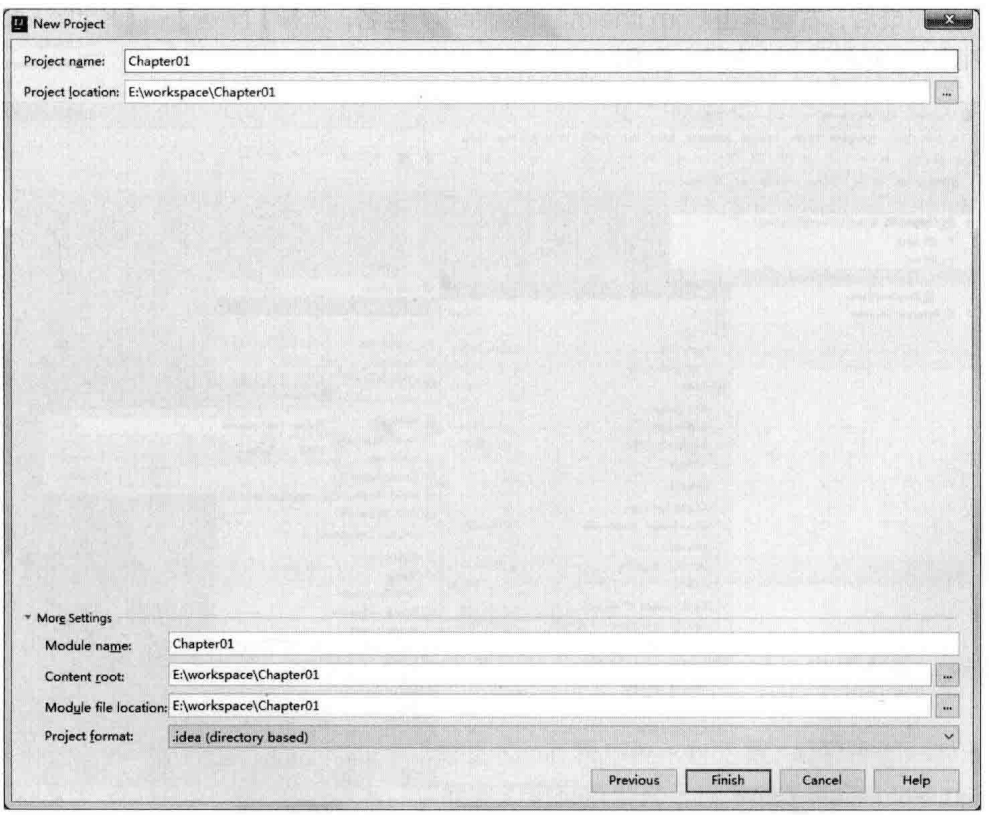


图1-8 设置项目名称与位置

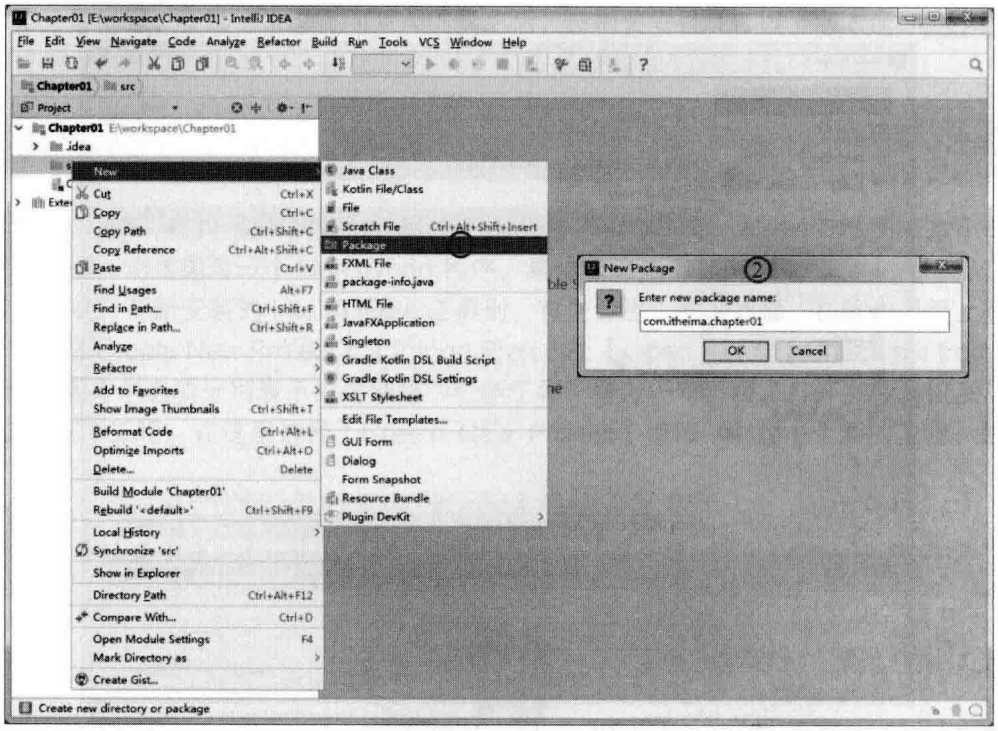


图1-9　创建包名

包创建完成后，右键单击 com.itheima.chapter01 包名，选择【New】→【Kotlin File/Class】选项，创建 HelloWorld.kt 文件，如图 1-10 所示。

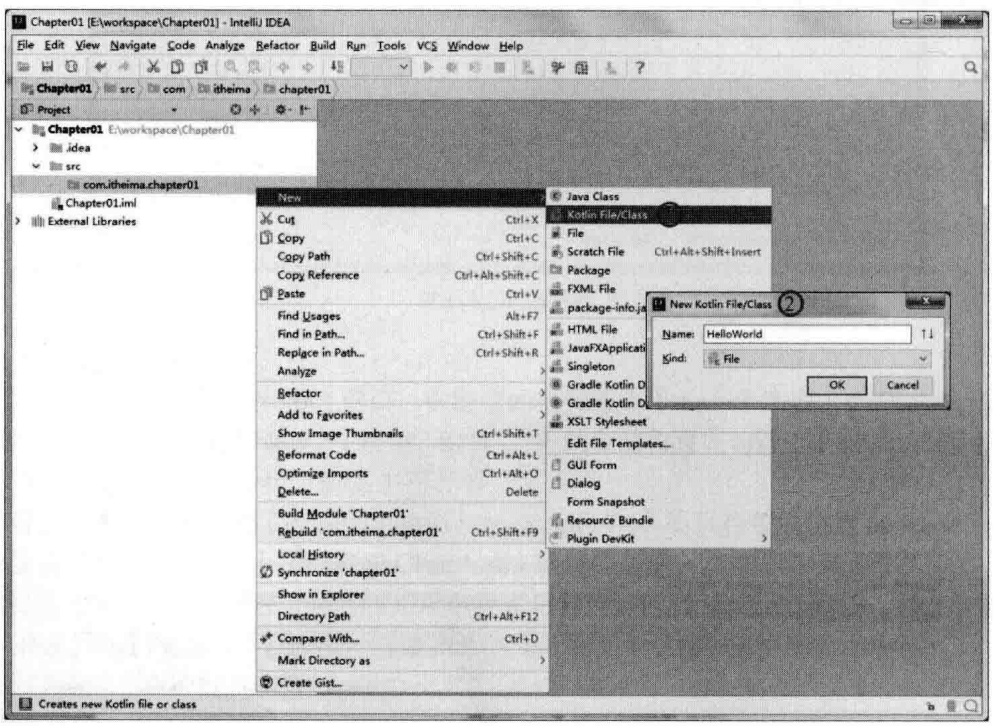


图1-10　创建HelloWorld.kt文件