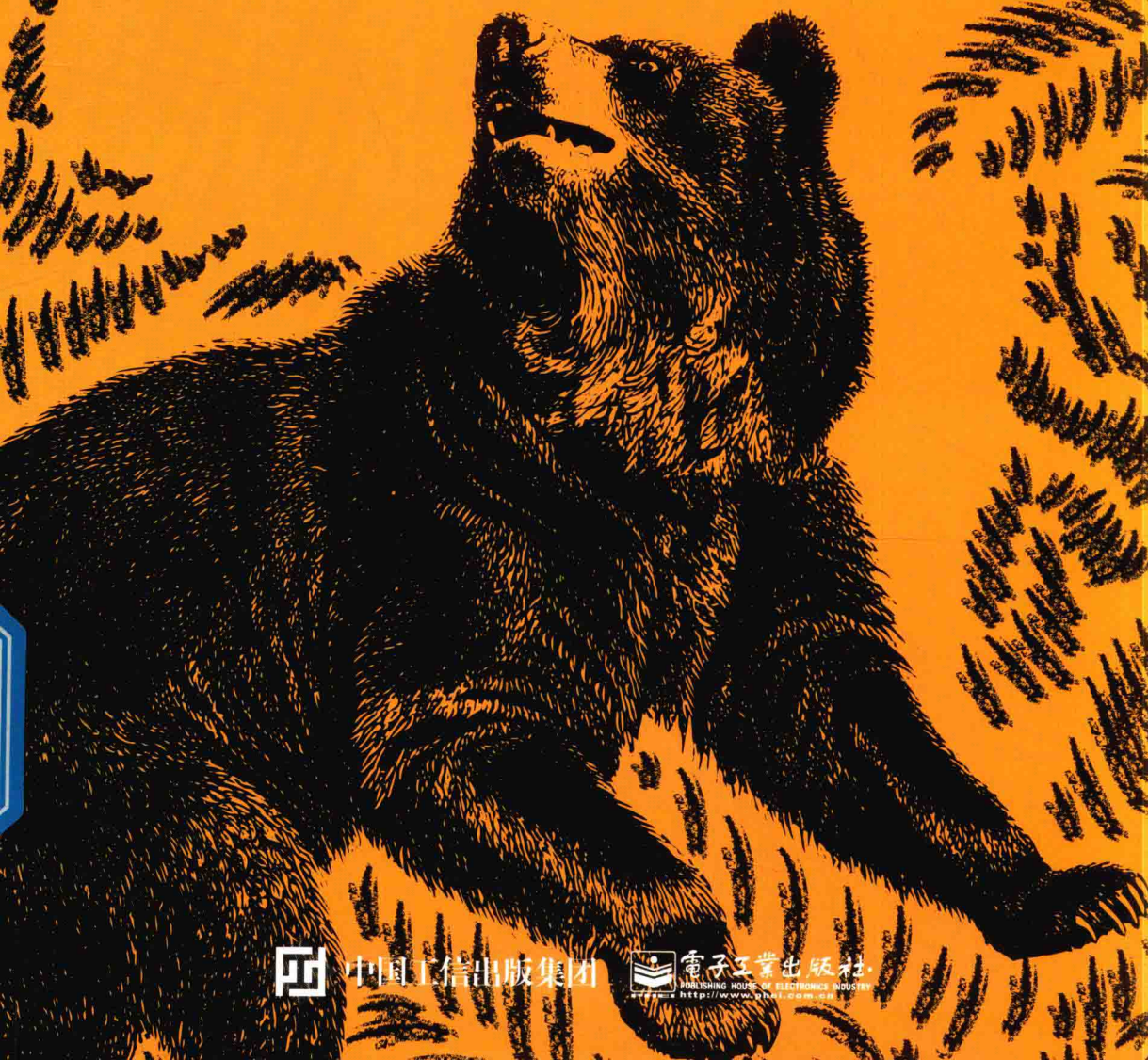
Broadview
www.broadview.com.cn

揭秘 Kotlin 编程原理

封亚飞 著



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

揭秘 Kotlin 编程原理

封亚飞 著

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

内 容 简 介

Kotlin被谷歌宣布为官方语言以来，引发了极大的关注，并成为学习的热点。

本书主要从封装、继承和多态三个方面全面介绍Kotlin面向对象设计的语法特性及其背后的实现方式。全书可分为基础篇、实战篇与提高篇，内容上层层深入，揭示了Kotlin对属性包装、多种形态的函数定义方式以及各种特殊类型的定义等方面的背后实现机制。

本书适合各种编程语言的开发者阅读，不管你是Java开发、Kotlin开发、Android开发，还是PHP、JSP，或者是C、C++、VB、Go语言的爱好者，都可以翻开阅读。因为里面总会有让你感到熟悉的一些语言特性，当你看到Kotlin中也有这样一种特性的时候，你一定会会心一笑！

图书在版编目（CIP）数据

揭秘 Kotlin 编程原理 / 封亚飞著. —北京：电子工业出版社，2018.3
ISBN 978-7-121-33481-8

Ⅰ. ①揭… Ⅱ. ①封… 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 006815 号

策划编辑：刘　皎
责任编辑：牛　勇
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
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：720×1000　1/16　印张：18.75　字数：324 千字
版　　次：2018 年 3 月第 1 版
印　　次：2018 年 3 月第 1 次印刷
定　　价：79.00 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

谷歌作为世界级的科技公司巨头，强悍的技术研发与创新能力使其一直是业界的楷模，其在各个领域的每一次创新，都能够引领一个新的时代！

Kotlin 便是其最新的一个创新力作。

编程语言的历史已经超过了半个世纪，从最初的机器二进制码，到汇编、B 语言，再到 C 语言，再到由 C 语言所开发出的其他若干种编程语言。每一种编程语言都有其特定的用途，例如 C 语言通常用于开发底层系统软件或者驱动程序，而部分更底层的功能则必须要由汇编甚至是直接的机器指令去完成。再如 C++、Delphi 曾经统治了 PC 桌面软件的开发领域，而应用服务器端的开发则长期被 PHP、ASP、JSP 垄断，浏览器端的嵌入式脚本则几乎由 JavaScript 语言一统天下。

这几年互联网领域先后经历了几次大革命，包括物联网、大数据、云计算等，如今则处于人工智能的火热时代。在这个时代，人们极其努力地开启机器智慧，在大数据样本下，通过算法，让机器进行一定的模糊识别，从而解决很多传统办法解决不了的棘手问题。伴随其中的一个重要的编程语言便是 Java，因为 Java 的口号是“write once, run anywhere”（即：一次编写，到处运行）。Java 由于其强大的跨平台（主要指操作系统）能力，而备受各种中间件组件开发人员的钟爱。而 Java 之所以能够跨平台，主要归功于 JVM 虚拟机。

JVM 虚拟机内部针对不同的底层平台进行了通用性抽象，从而可以让 Java 这种

上层编程语言对外提供统一的 API，例如在进行多线程开发时，开发者无须在不同的平台上引入不同的类库，而在开发界面视图时，Java 也提供统一的界面组件类库。平台的差异化工作都交给底层的 JVM 虚拟机进行适配处理，从而让高层业务开发人员可以专心进行业务设计与逻辑实现，不用再关心底层各种纷繁复杂的硬件和平台特性。开发人员唯一需要感知的平台差异性仅仅在于需要在不同的平台上下载平台相关的 JVM 软件而已。

在 Java 刚推出来的几年里，由于 JVM 的性能低下，导致不太被认可。但是后来随着各种黑科技的引入，例如 JIT 即时编译、基于 Java 字节码的栈顶缓存技术、垃圾回收算法的改进、JDK 高性能类库（例如并发包、NIO 等）的发布，等等，JVM 的性能得到长足改进和飞速提升，早已今非昔比，在部分场景下甚至比 C/C++的性能还要高，例如运行期所进行的方法与线程级的逃逸分析以及 C1、C2 分级动态编译等技术。人们再也没有任何理由拒绝使用 Java，所以 Java 得到了飞速发展，多年来稳居服务端应用编程语言使用率第一的宝座。

同时，JVM 是一个开源的产品，在技术体系上也是开放的，当然，并不是无条件的开放，而是在统一的技术规范下，不对实现做任何约束。因此各种基于 JVM 规范的编程语言也得以被发明出来，例如 Scala、Clojure、Groovy 等，甚至 PHP、Ruby 等程序也可以转换到 JVM 规范。不管高级编程语言是 Java 还是 Scala，只要能够被翻译成 Java 字节码，JVM 都能够执行，这便是技术规范的开放性。

虽然 Java 与 JVM 在最近这些年取得了巨大的成功，但是也并非没有缺点。例如 Java 是一种严格的面向对象设计的编程语言，一切编程要素都被严格编写在 Java 类型内部，你不可能像 C 语言那样，直接在源程序中定义一个函数。这种完全的面向对象设计的特性也给 Java 自己造成了很多不便，例如无法对底层类库进行扩展，除非你去继承并实现一个新的类型。

同时，Java 编程语言的语法太过于严格和死板，不像很多其他编程语言那样，有不少让人心动的功能特性，这种死板和严格往往会造成工作效率的低下。

于是，Kotlin 诞生了。

当笔者刚看到 Kotlin 时，并没有多少惊讶。因为 Kotlin 底层仍然是基于 JVM 虚

拟机的，主要是“仍然”哟！因为基于 JVM 的编程语言太多了，它们都有自己的“脾气”和鲜明的“性格”，很难说谁比谁好。更何况，笔者刚刚读完了 JVM 底层的源代码，并汇编成书——《揭秘 Java 虚拟机：JVM 设计原理与实现》（有兴趣的读者可以上淘宝、京东、亚马逊、当当等主流平台上选购），因此笔者并没有觉得 Kotlin 会“玩”出啥新的花样来。然而，随着对 Kotlin 特性了解的加深，笔者越来越发现 Kotlin 真的不是随随便便搞出来的一个全新的编程语言——如果你有多年的编程开发经验，并且熟知很多的编程语言，你会对 Kotlin 感到很惊讶！因为这真的是一门融合了众多编程语言特性的编程语言，并且是在不违反 JVM 规范的基础上，将其他众多语言的特性融入了进来，说其是博采众家之长，一点也不为过。

在惊讶之余，笔者将对 Kotlin 的理解写了下来，并形成了本书。本书着重为你介绍 Kotlin 各种高级特性背后的实现机制，希望我们可以一起探讨 Kotlin 背后的设计哲学。

本书主要从封装、继承和多态这三方面介绍 Kotlin 的面向对象设计的语法特性及其背后的实现方式。

其中详细讲解了 Kotlin 在面向对象封装方面所作出的努力，Kotlin 保留了 Java 封装好的一面，勇敢地摒弃了其不好的一面，例如对静态字段和方法的舍弃与变通。而在方法封装上，Kotlin 更是玩出了新花样，打破了 Java 封装的彻底性，让 Java 开发者可以体验“面向过程”编程的感觉。同时，Kotlin 充分吸收其他编程语言中的好的语言特性，提供了诸如 VB 语言中的“with 语法”。

在继承方面，Kotlin 也有自己的思考，其综合了 Java 和 C++等面向对象编程语言继承的优缺点，设计出自己的一套独特的继承机制。不过 Kotlin 依然保留了 Java 语言中一个类不能同时继承多个类的强制约束。

Kotlin 给人最多的惊艳，都集中在“多态”这一领域。其中，最让笔者惊叹的便是 Kotiln 提供了这样一种能力：不用修改原有类，也无须通过继承的方式，就能为某个类增加新的行为。虽然 Kotlin 仅仅是取巧，仅仅实现了一个语法糖的包装，但是这种小的改变却秀出了“美”的新高度。或许，这都不能算是继承，这里姑且将其与继承混为一谈吧。

另外，操作符重载也是 Kotlin 中一个非常惊艳的功能，给了笔者不小的冲击力——也许是知识的贫乏限制了笔者的想象力。

如果仅仅讲解 Kotlin 的语法，多么无聊。所以，本书并没有只停留于以往内容层面的介绍，作为一名对技术抱有极大热情、凡事喜欢刨根问底的极客（姑且是往自己脸上贴金吧^_^），笔者进一步研究了 Kotlin 各种高级特性背后的实现机制。本书主要揭示了 Kotlin 中属性包装、多种形态的函数定义及各种特殊类型的定义等背后的实现机制。由于 Kotlin 并没有自己的虚拟机，而是完全托管于 JVM 虚拟机，所以 Kotlin 最多只能将技术玩到“Java 字节码”这一层，而笔者对此则是再熟悉不过的。

本书适合各种编程语言的开发者阅读，不管你是使用 Java、Kotlin、Android 开发，还是使用 PHP、JSP 开发，甚至是使用 C、C++、VB、GO 开发，都可以阅读本书。因为你总会从本书中读到一些熟悉的语言特性，当你看到 Kotlin 中也有这样一种特性的时候，你一定会心一笑！

目　录

快速入门

1.1 简介

注：本节部分内容摘自 Kotlin 官网和相关资料。

谷歌在2017年的I/O 开发者大区会上宣布了安卓开发全面支持Kotlin编程语言。虽然谷歌时至今日才支持 Kotlin，但是 Kotlin 的历史却比这个要早很多——

Kotlin 是 JetBrains 在 2010 年推出的基于 JVM 的新编程语言，其主要设计目标如下：

- 兼容 Java。
- 比 Java 更安全，能够静态检测常见的陷阱，如引用空指针。
- 比 Java 更简洁，通过支持变量类型推断、高阶函数（闭包）、构造函数、混合（mixins）和一级委托等来实现。
- 比最成熟的竞争对手 Scala 语言更加简单。

Kotlin 的愿景是在现代应用程序的所有组件中使用单一的表达式，并成为高性能的强类型语言。Kotlin 对单一表达式和高性能的支持基于以下两点：

- 对 JavaScript 提供支持，支持所有 JavaScript 语言特性、大部分标准库及 JavaScript 互操作性。这允许将应用程序的浏览器前端迁移到 Kotlin，同时继续使用现代的 JavaScript 开发框架（如 React）。
- 引入了对协同程序的支持。作为线程的轻量级替代，协同程序支持更多可扩展的应用程序后端，在单个 JVM 实例上支持大量工作负载。除此之外，协同程序是一个非常具有表现力的实现异步行为的工具，这对于在所有平台上构建响应式用户界面很重要。

下面给出两个地址。

Kotlin 官网：http://jetbrains.com/kotlin。

Kotlin 源码：http://github.com/JetBrains/Kotlin。

除了以上这些强大的特性外，Kotlin 还支持以下特性：

- Kotlin 可以自由地引用 Java 的代码，反之亦然。
- Kotlin 可以引用现有的全部 Java 框架和库。
- Java 文件可以很轻松地借助 IntelliJ 的插件转成 Kotlin。

Kotlin 可以做到与 Java 百分之百互通，并具备诸多 Java 尚不支持的新特性。Kotlin 可以使用 Java 所有的 Library，两种代码可以在同一个项目中共存，甚至可以做到双向的一键转换。

正是由于 Kotlin 有这么多新特性，所以才最终得到谷歌的垂青。这些新特性绝非锦上添花，而是能够实实在在解决问题，所以 Kotlin 受到广大程序员的追捧也在情理之中。

Kotlin 虽然十分强大，但是其底层仍然基于 JVM（Java Virtual Machine），这也是 Kotlin 能够与 Java 百分之百兼容的技术基础。不过从这个角度来看， Kotlin 的精华其实是提供了各种语法糖，通过这些语法糖，开发者可以提高编程效率，并提升程序的健壮性。

这里不得不提一下 Kotlin 的开发商——

Kotlin 的开发商 JetBrains 在业界大名鼎鼎，很多人正在使用的 IntelliJ IDEA 就是

该公司所开发。由于该公司开发的是编译器，因而它对各种编程语言有独到的研究，可以这么说，Kotlin 是一门集大成的编程语言。在 Kotlin 中，你可以看到很多其他编程语言中的优秀特性，例如 C++的构造函数继承、C#的函数扩展、Visual Basic 的 with 语法、Python 的 for 循环语法，等等。所以，如果你是其他编程语言的开发者，你一定能够在 Kotlin 中发现你曾经熟悉的语言特性，很多惊喜都在等着你！

```
Let's GO!
```

1.2 编写第一个 Hello World 程序

与其他任何教程一样，在正式介绍 Kotlin 之前，我们先通过一个“Hello World”程序让大家认识一下 Kotlin。程序很简单，跑完后你会发现原来是如此简单。

步骤 1：下载 IntelliJ IDEA

Kotlin 中既有 Eclipse 的插件，也有 IntelliJ IDEA 的插件。不过由于 Kotlin 和 IntelliJ IDEA 都是同一个开发商所开发（即 JetBrains），因此推荐下载 IntelliJ IDEA。下载地址如下：

https://www.jetbrains.com/idea/download/

下载之后在本地完成安装。

步骤 2：安装 Kotlin 插件

IntelliJ IDEA 安装完成之后，打开它，接着安装 Kotlin 插件。单击菜单栏“IDEA”→“Preferences”，在打开的窗口中的搜索框里输入“Kotlin”，并进行搜索。搜索结果通常会包含多条，选择其中带有“Kotlin language support”描述信息的插件，然后单击安装，如图 1-1 所示。

由于从国外站点下载资源，因此安装过程通常会比较慢，请耐心等待。

步骤 3：新建工程

Kotlin 插件安装完成之后，需要重启 IntelliJ IDEA 才会使其生效。

IntelliJ IDEA 重启之后，便可以新建 Kotlin 工程开始编写代码。

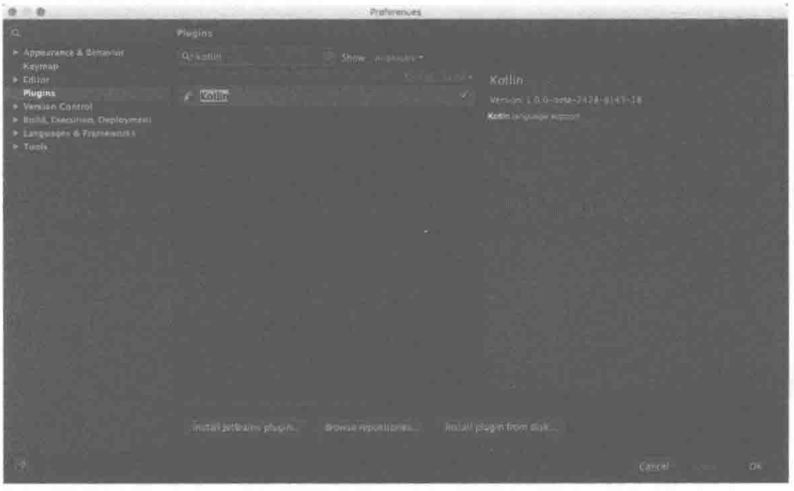


图 1-1　安装 Kotlin 插件

依次单击 IntelliJ IDEA 菜单栏的“File”→“New”→“Project”菜单项，会弹出一个工程模板选择窗口。如果 Kotlin 插件的确安装成功，则会看到其中有一个“Kotlin”选项，如图 1-2 所示。

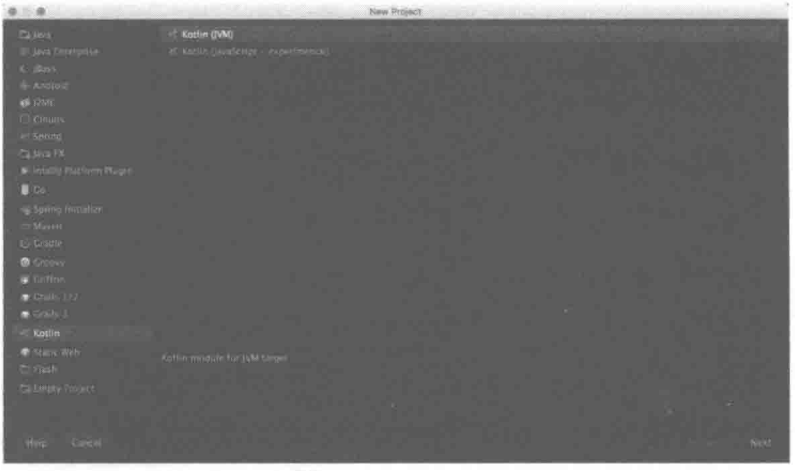


图 1-2　新建工程时选择 Kotlin 工程模板

单击左侧的“Kotlin”菜单项之后，右侧通常会出现两个选项：

- Kotlin(JVM)
- Kotlin(JavaScript - experimental)

之所以会弹出这两个选项，是因为 Kotlin 既可以兼容 Java 语言，也可以兼容 JavaScript 语言。

在这里，选择第一个选项，即 Kotlin(JVM)。

选中之后，单击 Next 按钮，进入下一步。单击之后，出现如图 1-3 所示的窗口。

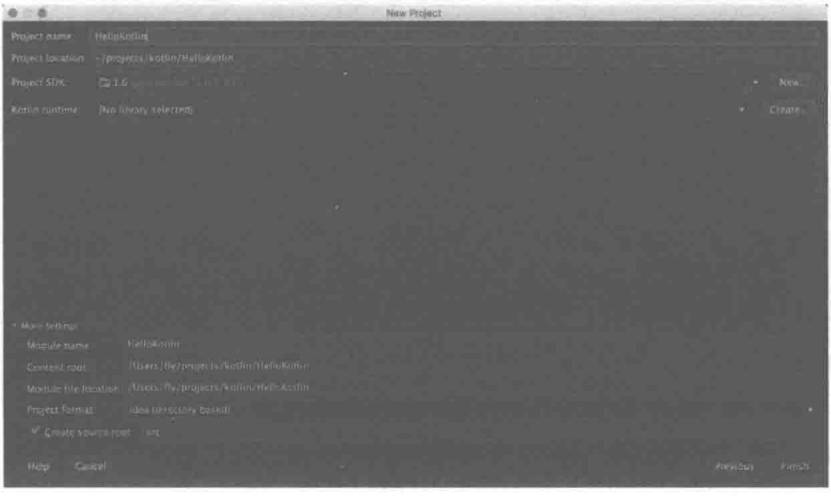


图 1-3 设置 Kotlin 工程配置窗口

在这里设置 Kotlin 工程的基本配置信息。首先需要设置的是工程所在的文件目录，这一步很简单，略过不表。

选择好工程目录并填写好工程名称（本示例所输入的工程名称是 HelloKotlin）后，接下来需要设置 JDK。

步骤 4：选择 JDK

由于本示例选择使用 Kotlin 开发基于 JVM 的程序，因此必须选择对应的 JDK（Java Development Kit）版本，当然前提是本机已经安装了 JDK。

笔者的机器上同时安装了 JDK 6 和 JDK 8，因此在图 1-4 所示的位置便会出现两个选项。这里选择 JDK 6。

步骤 5：设置 Kotlin runtime

设置完 JDK 版本后，接下来最重要的一步便是设置 Kotlin 的运行时环境。