互动出版网 China-Pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

第 1 章 Chapter 1

引　言

1.1 简史

Java 语言是一门通用的、面向对象的、支持并发的程序语言。它的语法与 C 和 C++ 语言非常相似，但隐藏了 C 和 C++ 中许多复杂、深奥及不安全的语言特性。Java 平台最初用于解决基于网络的消费类设备上的软件开发问题，它在设计上就考虑到要支持部署在不同架构的主机上，并且不同组件之间可以安全地交互。面对这些需求，编译出来的本地代码必须解决不同网络间的传输问题，并能够运行在各种客户端上，而且还要使客户端确信这些代码是安全的。

伴随着万维网的盛行发生了一些十分有趣的事情：Web 浏览器允许数以百万计的用户共同在网上冲浪，以及通过很简单的方式访问丰富多样的内容。用户冲浪所使用的设备并不是其中的关键，它们仅仅是一种媒介，无论机器的性能如何，无论使用高速网络还是慢速的 modem，用户总能看到并听到同样的内容。

Web 狂热者很快就发现网络信息的载体——HTML 文档格式对信息的表达有很多限制，HTML 的一些扩展应用，譬如网页表单，让这些限制显得更加明显。显而易见，没有任何浏览器能够承诺它可以提供给用户所需要的全部特性，扩展能力将是解决这个问题的唯一答案。

Sun 公司的 HotJava 浏览器是世界上第一款展现出 Java 语言某些有趣特性的浏览器，它允许把 Java 代码内嵌入 HTML 页面。显示 HTML 页面时，这些 Java 代码也会一并下载至浏览器中。而在浏览器获取这些代码之前，它们已经过严谨地检查以保证它们是安全的。与 HTML 语言一样，这些 Java 代码与网络和主机是完全无关的，无论代码来自哪里，在哪台机器上执行，它们执行时都能表现出一致的行为。

带有 Java 技术支持的网页浏览器将不再受限于它本身所提供的功能。浏览网页的用户可

互动出版网 China-Pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

以放心地假定在他们机器上运行的动态内容不会损害其机器。软件开发人员编写一次代码，程序就可以运行在所有支持 Java 运行时环境的机器之上。

1.2 Java 虚拟机

Java 虚拟机是整个 Java 平台的基石，是 Java 技术用以实现硬件无关与操作系统无关的关键部分，是 Java 语言生成出极小体积的编译代码的运行平台，是保障用户机器免于恶意代码损害的屏障。

Java 虚拟机可以看做一台抽象的计算机。如同真实的计算机那样，它有自己的指令集以及各种运行时内存区域。使用虚拟机来实现一门程序设计语言是相当常见的，业界中流传最为久远的虚拟机可能是 UCSD Pascal 的 P-Code 虚拟机[㊀]。

第一个 Java 虚拟机的原型机是由 Sun Microsystems 公司实现的，它用在一种类似 PDA（Personal Digital Assistant，俗称掌上电脑）的手持设备上，以仿真实现 Java 虚拟机指令集。时至今日，Oracle 已将许多 Java 虚拟机实现应用于移动设备、台式机、服务器等领域。但 Java 虚拟机并不局限于特定的实现技术、主机硬件和操作系统。Java 虚拟机也不局限于特定的代码执行方式，它虽然不强求使用解释器来执行程序，但是也可以通过把自己的指令集编译为实际 CPU 的指令来实现。它可以通过微代码（microcode）来实现，甚至可以直接在 CPU 中实现。

Java 虚拟机与 Java 语言并没有必然的联系，它只与特定的二进制文件格式——`class` 文件格式所关联。`class` 文件包含了 Java 虚拟机指令集（或者称为字节码（bytecode））和符号表，以及其他一些辅助信息。

基于安全方面的考虑，Java 虚拟机在 `class` 文件中施加了许多强制性的语法和结构化约束，凡是能用 `class` 文件正确表达出来的编程语言，都可以放在 Java 虚拟机里面执行。由于它是一个通用的、机器无关的执行平台，所以其他语言的实现者都可以考虑将 Java 虚拟机作为那些语言的交付媒介。

本书所说的 Java 虚拟机与 Java SE 8 平台相兼容，而且支持由本书所定义的 Java 编程语言。

1.3 各章节摘要

本书中其余章节的概述如下：

- 第 2 章概览 Java 虚拟机整体架构。
- 第 3 章介绍如何将 Java 语言编写的程序转换为 Java 虚拟机指令集。
- 第 4 章定义 `class` 文件格式。它是一种与硬件和操作系统无关的二进制格式，用来

㊀ P-Code 虚拟机是由加州大学圣地亚哥分校（UCSD）于 1978 年发布的高度可移植、机器无关的、运行 Pascal 语言的虚拟机。——译者注

互动出版网 China-Pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

表示编译后的类和接口。

- 第 5 章定义了 Java 虚拟机启动以及类和接口的加载、链接和初始化过程。
- 第 6 章定义了 Java 虚拟机指令集，并按照这些指令的指令助记符的字母顺序来表示。
- 第 7 章提供了一张以操作码值为索引的 Java 虚拟机操作码助记符表。

在《 Java 虚拟机规范（第 2 版）》中，第 2 章是 Java 语言概览，这可以使读者更好地理解 Java 虚拟机规范，但它本身并不属于规范的一部分。因此本规范里没有再包含此章节的内容，读者可以参考《 Java 语言规范》（Java SE 8 版）来获取这部分信息，如在本书中有需要引用这些信息的地方，将使用类似于“（JLS § x.y）”的形式来表示。

在《 Java 虚拟机规范（第 2 版）》中，第 8 章用于描述 Java 虚拟机线程和共享内存之间的底层操作，它对应于《 Java 语言规范》（Java SE 8 版）的第 17 章，而那一章又对应于 JSR-133 专家组所发布的《 Java 内存模型和线程规范》[㊀]。本规范中不再包含这部分内容，读者可参考上述规范来获取关于线程与锁的信息。

1.4 说明

本书会用到 Java SE 平台 API 中的类和接口。如果单用一个未经修饰的标识符（比如 N）来指代类或接口，而那个标识符又不是范例中所定义的，那我们指的就是 `java.lang` 包中的类名或接口名（比如 `java.lang.N`）。如果要提到其他包中的类名或接口名，我们则会使用全限定名。

每当提及 `java` 或者它的子包（`java.*`）里的类和接口时，就意味着这个类或接口是由启动类加载器进行加载的（见 5.3.1 节）。

每当提及某个 `java` 包的子包时，就意味着这个包是由类加载器所定义的。

在本书中，*斜体*用于描述 Java 虚拟机中的“汇编语言”，即操作码和操作数，也包括一些 Java 虚拟机运行时数据区中的项目，有时也用来说明一些新的条目和需要强调的内容。

非规范性的信息用于阐明规范中的某些内容，这部分信息以小字缩排的形式来印刷。

这些文本是 Java 虚拟机规范之外的信息。它们用来表示某些直观的内容、阐述某些原理、给出某种建议或演示某个范例等等。

1.5 反馈

读者如发现本书中有错误、遗漏或含义不明之处，可通过 jvms-comments_ww@oracle.com 发送反馈信息。

用 `javac`（Java 编程语言的参照编译器（reference compiler））来生成并操作 `class` 文件时，如果有问题，可与 compiler-dev@openjdk.java.net 联系。

㊀ 《Java Memory Model and Thread Specification》：http://www.jcp.org/en/jsr/summary?id=133。——译者注