Broadview
www.broadview.com.cn
PEARSON
· 原味精品书系 ·

Java虚拟机规范（英文版）

（Java SE 8版）

The Java Virtual Machine Specification (Java SE 8 Edition)

[美] Tim Lindholm
Frank Yellin
Gilad Bracha
Alex Buckley 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

· 原味精品书系 ·

Java虚拟机规范（英文版）

（Java SE 8版）

The Java Virtual Machine Specification（Java SE 8 Edition）

[美] Tim Lindholm
Frank Yellin
Gilad Bracha
Alex Buckley 著

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

内容简介

本书由Java虚拟机技术创建人撰写，全面而准确地阐释了Java虚拟机各方面的细节，围绕Java虚拟机整体架构、编译器、class文件格式、加载、链接与初始化、指令集等核心主题对Java虚拟机进行全面而深入的分析，深刻揭示Java虚拟机的工作原理。书中完整地讲述了由Java SE 8所引入的新特性，例如对包含默认实现代码的接口方法所做的调用，以及为支持类型注解及方法参数注解而对class文件格式所做的扩展等，还阐明了class文件中各属性的含义及字节码验证的规则。

本书基于Java SE 8，是深度了解Java虚拟机和Java语言实现细节的极佳选择。

图书在版编目（CIP）数据

Java虚拟机规范：Java SE 8版 =The Java Virtual Machine Specification, Java SE 8 Edition：英文 /（美）林霍尔姆（Lindholm,T.）等著. — 北京：电子工业出版社，2016.4
（原味精品书系）
ISBN 978-7-121-27305-6
I. ① J…II. ① 林…III. ① JAVA语言－程序设计－英文 IV. ① TP312
中国版本图书馆CIP数据核字（2015）第231508号

策划编辑：张春雨　刘　芸
责任编辑：徐津平
印　　刷：三河市兴达印务有限公司
装　　订：三河市兴达印务有限公司
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：787×980 1/16　　印张：37.5　　字数：720千字
版　　次：2016年4月第1版
印　　次：2016年4月第1次印刷
定　　价：108.00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。
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。
服务热线：（010）88258888。

向索菲亚和苏珊致以最深的谢意！

前言

本书涵盖了自 2011 年发布 Java SE 7 版之后所发生的所有变化。此外，为了与常见的 Java 虚拟机实现相匹配，本书还添加了大量修订及说明。

本版与前面各版一样，仅仅描述了抽象的 Java 虚拟机，而在实现具体的 Java 虚拟机时，本书仅指出了设计规划。Java 虚拟机的实现必须体现出本书所列规范，但仅在确有必要时才受限。

对 Java SE 8 而言，Java 编程语言里的一些重要变化在 Java 虚拟机中都有相应的体现。为了尽量保持二进制兼容性，我们应该直接在 Java 虚拟机里指定令人满意的默认方法，而不应该依赖于编译器，因为那样做将无法在不同厂商、版本的产品之间移植。此外，那种做法也不可能适用于已有的 class 文件。在设计 JSR 335——*Lambda Expressions for the Java Programming Language*（《Java 编程语言的 lambda 表达式》）时，Oracle 公司的 Dan Smith 向虚拟机实现者咨询了将默认方法集成到常量池和方法结构、方法与接口方法解析算法，以及字节码指令集中的上佳方式。JSR 335 也允许在 class 文件级别的接口里出现 private 方法与 static 方法，而这些方法也同接口方法解析算法紧密地结合起来了。

Java SE 8 的特点之一是：Java SE 平台的程序库也伴随着 Java 虚拟机一起进化。有个小例子可以很好地说明这一特点：在运行程序的时候，Java SE 8 可以获取方法的参数名，虚拟机会把这些名字存放在 class 文件结构中，而与此同时，java.lang.reflect.Parameter 里也有个标准的 API 能够查询这些名字。另外，我们也可以通过 class 文件结构中一项有趣的统计数据来说明这个特点：本规范的第 1 版中定义了 6 个属性，其中有 3 个属性对 Java 虚拟机至关重要，而 Java SE 8 版的规范则定义了 23 个属性，其中只有 5 个属性对 Java 虚拟机很重要。换句话说，在新版规范中，属性主要是为了支持程序库而设计的，其次才是为了支持 Java 虚拟机本身。为了帮助读者理解 class 文件结构，本规范会更为清晰地描述出每项属性的角色及使用限制。

在 Oracle 公司的 Java Platform 团队里，有多位同事都对这份规范提供了很大的支持，他们包括：Mandy Chung、Joe Darcy、Joel Franck、Staffan Friberg、Yuri Gaevsky、Jon Gibbons、Jeannette Hung、Eric McCorkle、Matherey Nunez、Mark Reinhold、John Rose、Georges Saab、Steve Sides、Bernard Traversat、Michel Trudeau 和 Mikael Vidstedt。特别感谢 Dan Heidinga（IBM）、Karen Kinnear、Keith McGuigan 及 Harold Seigel 对常见的 Java 虚拟机实现中的兼容性及安全性问题做出的贡献。

Alex Buckley

于加利福尼亚州圣克拉拉

2014 年 3 月

目录

CHAPTER 1

Introduction

1.1 A Bit of History

The Java® programming language is a general-purpose, concurrent, object-oriented language. Its syntax is similar to C and C++, but it omits many of the features that make C and C++ complex, confusing, and unsafe. The Java platform was initially developed to address the problems of building software for networked consumer devices. It was designed to support multiple host architectures and to allow secure delivery of software components. To meet these requirements, compiled code had to survive transport across networks, operate on any client, and assure the client that it was safe to run.

The popularization of the World Wide Web made these attributes much more interesting. Web browsers enabled millions of people to surf the Net and access media-rich content in simple ways. At last there was a medium where what you saw and heard was essentially the same regardless of the machine you were using and whether it was connected to a fast network or a slow modem.

Web enthusiasts soon discovered that the content supported by the Web's HTML document format was too limited. HTML extensions, such as forms, only highlighted those limitations, while making it clear that no browser could include all the features users wanted. Extensibility was the answer.

The HotJava browser first showcased the interesting properties of the Java programming language and platform by making it possible to embed programs inside HTML pages. Programs are transparently downloaded into the browser along with the HTML pages in which they appear. Before being accepted by the browser, programs are carefully checked to make sure they are safe. Like HTML pages, compiled programs are network- and host-independent. The programs behave the same way regardless of where they come from or what kind of machine they are being loaded into and run on.

A Web browser incorporating the Java platform is no longer limited to a predetermined set of capabilities. Visitors to Web pages incorporating dynamic content can be assured that their machines cannot be damaged by that content. Programmers can write a program once, and it will run on any machine supplying a Java run-time environment.

1.2 The Java Virtual Machine

The Java Virtual Machine is the cornerstone of the Java platform. It is the component of the technology responsible for its hardware- and operating system-independence, the small size of its compiled code, and its ability to protect users from malicious programs.

The Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and manipulates various memory areas at run time. It is reasonably common to implement a programming language using a virtual machine; the best-known virtual machine may be the P-Code machine of UCSD Pascal.

The first prototype implementation of the Java Virtual Machine, done at Sun Microsystems, Inc., emulated the Java Virtual Machine instruction set in software hosted by a handheld device that resembled a contemporary Personal Digital Assistant (PDA). Oracle's current implementations emulate the Java Virtual Machine on mobile, desktop and server devices, but the Java Virtual Machine does not assume any particular implementation technology, host hardware, or host operating system. It is not inherently interpreted, but can just as well be implemented by compiling its instruction set to that of a silicon CPU. It may also be implemented in microcode or directly in silicon.

The Java Virtual Machine knows nothing of the Java programming language, only of a particular binary format, the `class` file format. A `class` file contains Java Virtual Machine instructions (or *bytecodes*) and a symbol table, as well as other ancillary information.

For the sake of security, the Java Virtual Machine imposes strong syntactic and structural constraints on the code in a `class` file. However, any language with functionality that can be expressed in terms of a valid `class` file can be hosted by the Java Virtual Machine. Attracted by a generally available, machine-independent platform, implementors of other languages can turn to the Java Virtual Machine as a delivery vehicle for their languages.