

高 等 学 校 计 算 机 课 程 规 划 教 材

Java SE与面向对象编程

孟双英 刘海燕 赵洋 编著



清华大学出版社



高等学校计算机课程规划教材

Java SE与面向对象编程

孟双英 刘海燕 赵洋 编著

清华大学出版社
北 京

内 容 简 介

本书主要包括 Java SE 基础语法,面向对象的核心思想(封装、继承、多态),异常处理和断言机制,I/O,文件操作,多线程编程,网络编程,图形界面编程,反射机制,JDBC 等内容。

本书涵盖了 Java SE 所有的知识点,从 Java 的基础语法到面向对象的设计思想,到具体 Java 项目的设计和编码,讲解由浅入深,覆盖全面,既有理论知识的深入剖析,又有实际应用的举例说明,能够带领读者完全掌握书中所讲知识在现实中的应用。

通过学习本教材及本系列其他教程,能够学习 Java 的各大应用领域核心技术,全面并系统地理解 Java,使读者能够对 Java 领域的应用由陌生到熟悉,进而精通,达到满足企业要求的水平。

本书可作为高等院校、示范性软件学院、高职高专院校的计算机相关课程和软件工程专业教材,也可作为各大软件培训机构的培训教程,同时也可供从事软件开发及测试工作的人员,以及对软件测试有兴趣的读者参考与学习。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java SE 与面向对象编程/孟双英,刘海燕,赵洋编著. —北京:清华大学出版社,2014

高等学校计算机课程规划教材

ISBN 978-7-302-34708-8

I. ①J… II. ①孟… ②刘… ③赵… III. ①JAVA 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 292373 号

责任编辑:汪汉友

封面设计:傅瑞学

责任校对:白 蕾

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:20.5

字 数:493 千字

版 次:2014 年 3 月第 1 版

印 次:2014 年 3 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:055444-01

序

面向对象程序设计(OOP)方法,以对象为基础,以消息驱动对象间的交互,从而实现“抽象对象”对虚拟世界的改造(即问题解决)。由于其设计思想与现实世界距离非常近,使面向对象程序设计语言备受欢迎。就面向对象程序设计语言的教学而言,不少老师一直纠结于“知识的传授”与“思想的传授”的关系问题上,即应该更重视语法本身?还是更重视思想?或是两者兼而并重?教师对这一问题的解决方式会直接体现在教学效果上。可见,无论是编写教材还是课堂教学,从根本内容上的定位和思考是很重要的。古人所云“授人以鱼”与“授人以渔”也是基于定位和思考而言的。

写一本好的面向对象程序设计语言教程,也是我——一个多年在一线从事教学工作的老师的渴望。无奈总是提笔见拙,每每不得满意,故一再搁置。曾经花费很长时间,与几位C++爱好者共同执笔,写了几百页的教材,因未尽期望,所以未付诸于铅字。所谓期望,一来希望能为读者真正提供有用的信息和快速掌握OOP设计思想的方法。二来希望书能在时间和空间中有久远或广泛的流传。

在河北师范大学软件学院工作的日子里,我认识了一群有志于教学和技术进步的年轻人。孟双英、刘海燕、赵洋就是其中的3位。他们一直致力于研究面向对象程序设计思想和Java语言,有丰富的项目经验和教学经验。这几位老师撰写的《Java SE与面向对象程序设计》一书,其优点体现在两个方面:其一在于定位,它定位于让学生入门,进入到Java SE编程的殿堂,通过阅读全书,我认为这个目的是能达到的;其二在于教学经验的总结,这样的内容安排、这样的组织结构、这样的知识深度、这样的阐述方式都是实践中检验了的。尽管这样的经验从不同角度看,有这样或那样的瑕疵;但事实上,无论我们自认为有多高明,也无法否定一个被实践多次检验了的经验或真理。

正因为如此,我又重新燃起了出版C++课程教材的想法。感谢这3位年轻人,他们帮助我重新认识了“适合的就是有用的”。谨希望这本来自于实践的书,对读者有用。

李文斌

2014年2月

前 言

本系列教程为“河北师范大学软件学院 Java 教研室”通过长期 Java 项目实践及 5 年实际教学经验的不断积累,多次讨论、修改、精心设计后,形成的一套成熟可行的 Java 课程体系,从中提取精华形成了 Java 应用开发的系列教材。其中包括 Java SE 与面向对象设计思想,Java EE 企业应用开发核心技术(Servlet 规范、JSP 规范),Android 手机应用开发 3 方面内容,其中《Java SE 与面向对象编程》是该系列教材中的基础和核心。

本书主要内容包括 Java SE 基础语法,面向对象的核心思想(封装、继承、多态),异常处理和断言机制,I/O,文件操作,多线程编程,网络编程,图形界面编程,反射机制,JDBC 等。

本书涵盖了 Java SE 所有的知识点,从 Java 的基础语法到面向对象的设计思想,到具体 Java 项目的设计和编码,讲解由浅入深,覆盖全面,既有理论知识的深入剖析,又有实际应用的举例说明,能够带领读者完全掌握书中所讲知识在现实中的应用。

本书的一大特色是理论、实践并存,讲练结合。讲解配备同步训练或上机训练,可让读者进一步巩固、加深和拓展知识。

目前 Java 相关书籍中讲语法的居多,本教材是结合教师最真实的上课授课情况总结而来的,所以更注重的是使初学者学习循序渐进,有理论学习,也有实践练习,能够轻松合理安排理论和实践课程时间,体现理论和实践的完美结合。

通过学习本教材及本系列其他教程,能够学习 Java 的各大应用领域核心技术,全面并系统地理解 Java,使读者对 Java 领域的应用由陌生到熟悉,进而精通,达到满足企业需求的水平。

编 者

2014 年 2 月

目 录

基 础 篇

第 1 章 Java 概述	3
1.1 Java 的历史	3
1.2 Java 语言的特点	4
1.3 Java 平台和主要应用方向	5
1.4 Java 开发环境的搭建	6
1.4.1 JVM	6
1.4.2 JRE 与 JDK	6
1.4.3 JVM、JRE 和 JDK 的关系	7
1.4.4 JDK 的下载和安装	7
1.5 Java 程序的基本结构	13
1.6 Java 注释	14
1.7 Java JDK 版本的更替	15
1.8 练习	15
第 2 章 Java 基础	16
2.1 标识符和关键字	16
2.1.1 标识符	16
2.1.2 关键字	16
2.2 数据类型	17
2.2.1 基本数据类型	17
2.2.2 引用数据类型	18
2.2.3 基本数据类型的类型转换	19
2.3 运算符和表达式	22
2.3.1 算数运算符和算数表达式	22
2.3.2 关系运算和条件运算	23
2.3.3 逻辑运算和位运算	25
2.3.4 赋值运算符	28
2.4 流程控制	28
2.4.1 流程控制概述	28
2.4.2 条件分支	28
2.4.3 循环控制	32
2.4.4 无条件分支	36

2.5	练习	38
第3章	数组	41
3.1	数组概述	41
3.2	一维数组	41
3.2.1	数组的声明	41
3.2.2	数组初始化	42
3.2.3	数组元素的访问	43
3.2.4	数组复制	44
3.3	Arrays 类	45
3.4	二维数组	48
3.4.1	二维数组的定义和初始化	48
3.4.2	Java 中二维数组的实质	50
3.4.3	二维数组应用举例	50
3.5	练习	52
第4章	类和对象	54
4.1	类与对象概述	54
4.1.1	为什么需要类	54
4.1.2	类的定义	55
4.1.3	实例化对象	55
4.1.4	访问对象中的成员变量	56
4.1.5	对象的引用性质	57
4.2	类的成员方法	58
4.2.1	成员方法的定义和使用	58
4.2.2	方法的参数传递	61
4.2.3	方法的重载	63
4.3	构造方法	64
4.4	垃圾回收机制	67
4.5	包的使用	69
4.5.1	为什么需要包	69
4.5.2	访问包中的类	70
4.6	练习	73
第5章	类的封装	75
5.1	封装的基本概念	75
5.2	封装的实现	77
5.3	类的特殊成员	79
5.4	练习	81

第 6 章 类的继承	83
6.1 继承的基本概念	83
6.1.1 继承的实现	84
6.1.2 继承的本质	84
6.2 继承带来的方法重写	87
6.2.1 方法重写的原则	87
6.2.2 方法重写的意义	89
6.2.3 方法的重写与重载的比较	90
6.3 抽象类与接口	91
6.3.1 抽象类的继承	91
6.3.2 抽象类的意义	93
6.3.3 接口	93
6.4 Object 类	95
6.5 练习	101
第 7 章 多态	103
7.1 多态的基本概念	103
7.2 多态的应用	107
7.2.1 引用变量的转型	107
7.2.2 接口回调	110
7.2.3 动态绑定	111
7.3 多态性总结	112
7.4 练习	114
第 8 章 包装器类	116
8.1 包装器类型	116
8.2 自动装箱	117
8.3 练习	122
第 9 章 内部类	123
9.1 内部类的基本使用	123
9.2 局部方法内部类	127
9.3 匿名内部类	128
9.4 静态嵌套类(静态内部类或者嵌套类)	130
9.5 练习	131
第 10 章 枚举	133
10.1 枚举的基本使用	133
10.2 练习	136

第 11 章 异常和断言	137
11.1 异常	137
11.1.1 JDK 中异常的层次结构	137
11.1.2 异常的分类	137
11.2 Java 异常处理机制	139
11.2.1 捕获异常	139
11.2.2 声明(传播)异常	141
11.3 自定义异常类	143
11.4 断言	145
11.5 练习	147
第 12 章 容器和泛型	149
12.1 Java 中容器的分类	149
12.2 集合类型	150
12.2.1 Set	150
12.2.2 List	155
12.2.3 Queue	156
12.3 Map 类型	160
12.4 迭代器(Iterator)	162
12.5 Collections 工具类	163
12.6 泛型	169
12.7 练习	172
第 13 章 流与文件	173
13.1 文件处理	173
13.2 I/O 和流	176
13.3 字节流	177
13.3.1 InputStream	178
13.3.2 OutputStream	178
13.3.3 FilterInputStream 和 FilterOutputStream	179
13.3.4 字节文件处理(FileInputStream 和 FileOutputStream)	180
13.4 字符流	181
13.4.1 Reader	182
13.4.2 Writer	182
13.4.3 字符文件的处理(FileReader 和 FileWriter)	183
13.5 对象流	183
13.5.1 对象的序列化和反序列化	183
13.5.2 ObjectInputStream 和 ObjectOutputStream	184
13.5.3 对象序列化的控制	187

13.6	其他常用流	190
13.7	练习	192
第 14 章	字符串解析、日期格式化	195
14.1	字符串	195
14.1.1	String	195
14.1.2	StringBuffer	201
14.1.3	StringTokenizer	203
14.1.4	Scanner 扫描输入	205
14.2	日期、日期的格式化以及字符串的解析	206
14.2.1	Date	206
14.2.2	Calendar	207
14.2.3	日期的格式化和解析	210
14.3	练习	214
第 15 章	线程和多线程	216
15.1	线程简介	216
15.1.1	程序、进程、线程	216
15.1.2	线程的生命周期	217
15.2	Java 中创建多线程	218
15.2.1	定义任务	218
15.2.2	线程中常用的方法	222
15.2.3	线程的优先级	226
15.3	多线程的同步和死锁	227
15.3.1	多线程同步	228
15.3.2	多线程的死锁问题	233
15.3.3	Daemon 线程	235
15.4	练习	235
第 16 章	网络编程	237
16.1	网络编程基础	237
16.1.1	TCP/IP 基本概念	237
16.1.2	URL 及应用	238
16.2	基于套接字的 Java 网络编程	238
16.2.1	Socket 通信	238
16.2.2	Socket 通信的过程	239
16.2.3	Socket 基于 TCP 协议的网络编程	241
16.2.4	Socket 基于 UDP 协议的网络编程	246
16.3	练习	248

第 17 章 图形界面编程	249
17.1 Java GUI 编程简介	249
17.2 AWT 图形界面编程	250
17.2.1 AWT 中的容器组件	250
17.2.2 布局管理器	253
17.2.3 AWT 中的事件处理	258
17.2.4 AWT 中的其他组件	261
17.3 Swing 组件	266
17.4 Java 中的 AWT 绘图	269
17.5 练习	273

扩展篇

第 18 章 反射	277
18.1 Java 反射机制的基本原理	277
18.2 Java 反射机制的应用	280
第 19 章 javac、java 命令的使用	287
19.1 JDK 环境配置中的环境变量	287
19.2 javac 命令	287
19.3 java 命令	288
第 20 章 JDBC 数据库连接技术	293
20.1 JDBC 的概念和类型	293
20.2 JDBC 中主要的类和接口	295
20.3 用 JDBC 进行数据库操作	298
20.4 数据库连接池技术	303
附录 A JavaBeans 命名规则	308
附录 B Eclipse 的安装和使用	310

基 础 篇

第 1 章 Java 概述

本章学习目标:

- (1) 了解 Java 语言的发展历史。
- (2) 熟悉 Java 语言的设计原则。
- (3) 熟悉 Java 的语言特性。
- (4) 熟悉 Java 语言的应用范围。
- (5) 掌握 Java 环境的搭建。
- (6) 掌握 Java 程序编译运行原理。

1.1 Java 的历史

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE、Java EE 和 Java ME)的总称。

在 1991 年,美国 Sun Microsystems 公司以 James Gosling(詹姆斯·高斯林)为首的一群技术人员创建了一个名为 Oak 的项目。此项目旨在寻找一种能在消费类电子产品上开发应用程序的语言。由于消费类电子产品种类繁多,比如手机、机顶盒、PDA 等,即便是同一类消费电子产品所采用的处理芯片和操作系统也不尽相同,所以 Oak 项目最大的难点就是跨平台问题的解决。当时最流行的编程语言是 C 和 C++ 语言,但是 C++ 语言相对庞大和复杂,对于消费类电子产品来说并不适用,安全性也不令人满意。所以 Oak 项目小组最终采用了许多 C 语言的语法,采用了 C++ 的面向对象的思想,并在安全性做了进一步提升,最终设计开发出了一种语言,并以项目的名字来命名此语言,由此诞生了 Oak 语言。但是 Oak 语言在商业上并没有获得成功。1995 年,互联网在世界上蓬勃发展,Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全等特点非常符合互联网的需求,于是改进了 Oak 语言的设计,并达到了以下几个目标:

- (1) 创建一种面向对象的设计语言,而不是面向过程的。
- (2) 提供一个解释执行的程序运行环境,使程序代码独立于平台。
- (3) 综合 C 和 C++ 的优点,使程序员容易掌握。
- (4) 去掉 C 和 C++ 中影响程序健壮性的部分,使程序更安全(例如指针、内存申请和释放)。
- (5) 实现多线程,使得程序能够同时执行多个任务。

最终,Sun 公司给该语言取名为 Java,在 1995 年 Sun 公司正式向 IT 业界推出了 Java 语言。

Java 语言一面世就引起了程序员和软件公司的极大关注,它的安全性、跨平台、面向对象、简单等特点非常适用于当时 Web 为主要形式的互联网的发展。程序员们纷纷尝试用 Java 编写网络应用程序,并利用网络把程序发布到世界各地进行运行,来验证 Java 的跨平

台。IBM、Oracle、微软、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订了合同,被授权使用 Java 平台技术。曾经微软的总裁比尔·盖茨先生在经过研究后认为“Java 语言是长时间以来最卓越的程序设计语言”,目前,Java 已经成为最流行的网络编程语言之一。在国内,近几年对 Java 技术人员的需求量也是与日俱增,许多高校纷纷开设了 Java 课程,Java 已逐步成为世界上程序员使用最多的编程语言。

1.2 Java 语言的特点

Java 具有卓越的通用性、高效性、跨平台性和安全性,广泛应用于 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

Java 语言的特点很多,在此归纳以下几个方面:

1. 简单性

Java 语言借鉴了 C++ 的成功,其中大部分语法结构都与 C++ 类似,因此一般熟悉 C++ 的编程人员上手 Java 都非常容易,同时 Java 去掉了 C++ 中容易混淆和较少适用的特性(比如指针、运算符重载、多重继承等),特别增加了内存空间自动回收功能,大大简化了 Java 程序的设计和开发。

2. 跨平台性

Java 程序运行在虚拟机之上,Java 虚拟机主要用于实现 Java 字节码(Byte Code)的解释和执行等功能,为不同的系统平台提供统一的接口。因此 Java 开发的程序可以运行在不同的系统平台。Java 字节码是一种近似于机器码的中间码,不受计算机硬件设备和操作平台种类的限制,只要计算机中有 Java 运行的环境,Java 字节码就可以在其上运行,这也正是 Java 最为突出的特征:一次编译,到处运行,即跨平台性(有人亦把它称为平台无关性)。配合 Java 本身语法的严格的数据类型和类结构的标准,Java 编写的程序就具有非常良好的可移植性。

3. 安全性

Java 的设计目的是提供一个用于网络分布式的计算环境,故 Java 本身特别强调安全性。从内存管理的角度看,Java 不允许用户利用程序强制对内存进行存取操作,即对程序员来说内存分配是透明的,这种机制使内存出错的几率大大减小;此外,Java 的编译器也没有对内存分配的决定权,而是通过系统所在的软硬件平台来决定的,Java 虚拟机在解释 Java 字节码的同时会对每段代码进行安全检测,对所有不合法的数据和无权限的存取操作加以阻止,不合法的字节码是无法被解释和执行的。

4. 面向对象性

Java 是纯面向对象的程序设计语言,它吸收了 C++ 面向对象的概念,将数据封装于类中,利用类的优点实现了程序的简洁和便于维护性。用 Java 语言编写程序时,通常以面向对象的思想来思考,程序设计人员只需要把精力用在类和接口的设计以及应用上即可,而不是专注于程序流程。

Java 本身提供了许多一般用途的类,程序中可以通过继承或者直接调用使用这些类的方法,与 C++ 中不同的是 Java 中类的继承是单重继承,即一个类只能有一个父类。

5. 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上,操作分布是指把一个计算分散在不同主机上处理。

Java 支持 WWW 客户机/服务器计算模式,因此它支持这两种分布性。对于前者,Java 提供了一个叫作 URL 的对象,利用这个对象,可以打开并访问具有相同 URL 地址上的对象,访问方式与访问本地文件系统相同。对于后者,Java 的 applet 小程序可以从服务器下载到客户端,即部分计算在客户端进行,提高系统执行效率。

Java 提供了一整套网络类库,开发人员可以利用类库进行网络程序设计,方便地实现 Java 的分布式特性。

6. 多线程

线程是操作系统中的一个概念,它又被称作轻量进程,是比传统进程更小的可并发执行的单位。

C 和 C++ 程序中均采用单线程体系结构,而 Java 却提供了多线程支持。

Java 在两方面支持多线程。一方面,Java 环境本身就是多线程的。若干个系统级线程运行负责无用内存的回收、系统维护等操作;另一方面,Java 语言内置多线程控制,可以大大简化多线程应用程序开发。利用 Java 的多线程编程接口,开发人员可以方便地写出支持多线程的应用程序,提高程序执行效率。但 Java 对多线程的支持在一定程度上受系统平台的限制,若操作系统本身不支持多线程,Java 的多线程特性就表现不出来。

7. 动态性

Java 的设计使其适应不断发展的环境,在类库中可以自由地加入各种新的类和方法以适应新的环境要求,并且不会影响用户程序的执行。

1.3 Java 平台和主要应用方向

Java 是程序设计语言和平台的统称,也就是说从某种意义上 Java 还是一个开发平台。目前 Java 的平台划分成 Java EE、Java SE 和 Java ME,这 3 个平台主要针对不同的市场目标和设备进行定位。

(1) Java SE(Java Standard Edition)标准版。它的主要目的是为台式机和 workstation 提供一个开发和运行的平台。包括标准的 Java SDK、工具、运行时环境和 API,是桌面开发和低端商务应用的解决方案。

(2) Java EE(Java Enterprise Edition)企业版。它的主要目的是为企业计算机提供一个应用服务器的运行和开发平台。Java EE 本身是一个开放的标准,任何软件厂商都可以推出自己的符合 Java EE 标准的产品,使用户可以有多种选择。是以企业为环境而开发应用程序的解决方案。IBM、Oracle、BEA、HP 等 29 家企业已经推出了自己的产品,其中以 BEA 公司的 WebLogic 产品和 IBM 的 WebSphere 最为著名。

(3) Java ME(Java Micro Edition)小型版。它主要面向消费类电子产品,为消费电子产品提供一个 Java 的运行平台,使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行包含高度优化的 Java 运行时环境,致力于电子消费产品和嵌入式设备的解决方案。

Java 语言目前在服务器端确立了强大的战略优势,同时由于其独有的一些特性,在嵌

入式系统方面的应用前景非常被看好,目前比较主流的智能手机操作系统 Android 就是以 Java 为基础开发语言的,未来 Java 的发展方向更是与互联网和移动互联网发展需求紧密地联系在一起。

1.4 Java 开发环境的搭建

1.4.1 JVM

Java 很重要的一个特点是跨平台性,这一特性的关键是通过 Java 虚拟机实现的。

Java 虚拟机(Java Virtual Machine,JVM)是一个想象中的机器,是在物理机上通过软件模拟来实现的,Java 虚拟机有自己想象中的硬件,比如处理器、堆栈、寄存器等,还具有相应的指令系统。

一般的高级语言想在不同的操作系统上运行,至少需要编译成不同的操作系统的目标代码,而 Java 中引入了 Java 虚拟机后,Java 的编译器只需要生成能够在 Java 虚拟机上运行的目标代码(字节码),Java 虚拟机在执行字节码时把字节码解释成具体平台上的机器指令执行。即 Java 程序在不同的平台上运行时不需要重新编译,Java 虚拟机屏蔽了具体平台相关的信息。

Java 虚拟机是 Java 语言底层实现的基础,对 Java 感兴趣的人都应该对 Java 虚拟机有个大概的了解,这有利于理解 Java 语言的一些行为,也有利于更好地使用 Java 语言。

1.4.2 JRE 与 JDK

通过第 1.4.1 节已经了解了 Java 程序要运行必须运行在 JVM 之上,但是只有 JVM 还不能完成字节码的程序的运行,因为 JVM 在解释字节码的时候需要用到一些类库,我们把 JVM+JVM 解释所需要的类库统称为 Java 运行时环境(Java Runtime Environment,JRE)。

也就是说只有有了 Java 运行时环境(JRE)Java 程序才能正常运行。

但对于开发人员来说只能运行 Java 程序的环境是远远不够的,还需要一个开发的调试环境,Sun 公司为全世界使用 Java 开发的人员提供了一套免费的开发工具集,取名为 JDK (Java Developers Kits),它不仅包含 Java 的运行时环境(JRE),还包括 Java 的开发环境。

JDK 是整个 Java 的核心,除了运行时环境和开发环境外,还包括一些实用 Java 工具和 Java 基础的类库(rt.jar)。不论什么 Java 应用服务器,实质都是内置了某个版本的 JDK。因此掌握 JDK 是学好 Java 的第一步。

最主流的 JDK 是 Sun 公司发布的 JDK,除了 Sun 之外,还有很多公司和组织都开发了自己的 JDK,例如 IBM 公司开发的 JDK,BEA 公司的 Jrocket,还有 GNU 组织开发的 JDK,等等。其中 IBM 的 JDK 包含的 JVM(Java Virtual Machine)运行效率要比 Sun JDK 包含的 JVM 高出许多。而专门运行在 x86 平台的 Jrocket 在服务端的运行效率也要比 Sun JDK 好很多。但不管怎么说,还是需要先把 Sun JDK 掌握好。

要掌握 JDK 就是要弄清楚 Java 程序的编译和运行的整个过程:对于程序员编写的一个完整的 Java 程序(.java 文件),正常的运行需要经过两个过程,首先需要经过编译的过