

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Java 程序设计教程

Java Programming

张克军 陆迟 孙海伦 庞丽娜 编著

- 详解Java基础，从零起步学习Java程序设计
- 丰富实例应用，通过案例讲解程序开发方法
- 突出实践操作，各章上机指导培养动手能力



高校系列



人民邮电出版社
POSTS & TELECOM PRESS

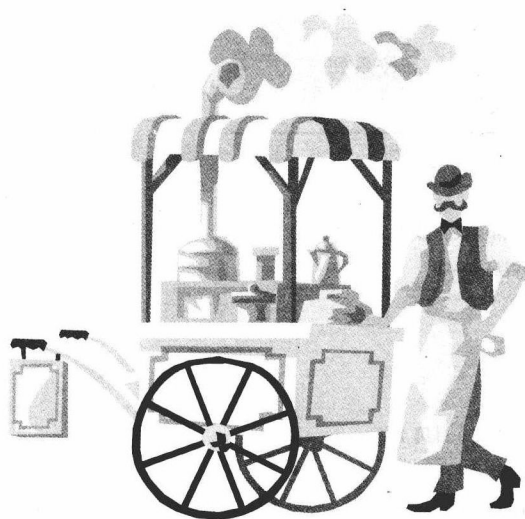
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Java 程序设计教程

Java Programming

张克军 陆迟 孙海伦 庞丽娜 编著



人民邮电出版社

北京

图书在版编目 (C I P) 数据

Java程序设计教程 / 张克军等编著. —北京: 人民邮电出版社, 2009.10
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-20552-0

I. J… II. 张… III. JAVA语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第051164号

内 容 提 要

本书按照实际开发流程并结合实际代码对 Java 程序设计方法进行了深入讲解。全书共 11 章, 循序渐进地介绍了 Java 语言概述、Java 程序设计基础、Java 中的对象与类、集合、数组与字符串、Java Applet、图形用户界面编程、Java 输入输出与数据库操作、网络程序设计和异常处理。

本书可作为普通高等院校 Java 程序设计类课程的教材使用, 也可供 Java 初学人员, 或者是已经具备一定基础的 Java 程序员阅读参考。

21 世纪高等学校计算机规划教材——高校系列

Java 程序设计教程

-
- ◆ 编 著 张克军 陆 迟 孙海伦 庞丽娜
责任编辑 蒋 亮
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21.75
字数: 568 千字
印数: 1—3 000 册
- 2009 年 10 月第 1 版
2009 年 10 月河北第 1 次印刷

ISBN 978-7-115-20552-0/TP

定价: 34.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

出版者的话

现今社会对人才的基本要求之一就是应用计算机的能力。在高等学校,培养学生应用计算机的能力,主要是通过计算机课程的改革,即计算机教学分层、分类规划与实施;密切联系实际,恰当体现与各专业其他课程配合;教学必须以市场需求为导向,目的是培养高素质创新型人才。

人民邮电出版社经过对教学改革新形势充分的调查研究,依据目前比较成熟的教学大纲,组织国内优秀的有丰富教学经验的教师编写了一套体现教学改革最新形势的“21世纪高等学校计算机规划教材——高校系列”。在本套教材的出版过程中,我社多次召开教材研讨会,广泛听取了一线教师的意见,也邀请众多专家对大纲和书稿做了认真的审读与研讨。本套教材具有以下特点。

1. 覆盖面广,突出教改特色

本套教材主要面向普通高等学校(包括计算机专业和非计算机专业),是在经过大量充分调研的基础上开发的计算机系列教材,涉及计算机教育领域中的所有课程(包括专业核心骨干课程与选修课程),适应了目前经济、社会对计算机教育的新要求、新动向,尤其适合于各专业计算机教学改革的特点特色。

2. 注重整体性、系统性

针对各专业的特点,同一门课程规划了组织结构与内容不同的几本教材,以适应不同教学需求,即分别满足不同层次计算机专业与非计算机专业(如工、理、管、文等)的课程安排。同时本套教材注重整体性策划,在教材内容的选择上避免重叠与交叉,内容系统完整。学校可根据教学计划从中选择教材的各种组合,使其适合本校的教学特点。

3. 侧重培养应用能力

目前社会对人才的需要更侧重于其应用能力,包括须具备扎实的计算机基础理论、良好的综合素质和实践能力。本套教材注意通过实践教学与实例教学培养解决实际问题的能力和知识综合运用能力。

4. 教学经验丰富的作者队伍

高等学校在计算机教学和教材改革上已经做了大量的工作,很多教师在计算机教育与科研方面积累了相当多的宝贵经验。本套教材均由有丰富教学经验的教师编写,并将这些宝贵经验渗透到教材中,使教材独具特色。

5. 配套资源完善

所有教材均配有PPT电子教案,部分教材配有实践教程、题库、教师手册、学习指南、习题解答、程序源代码、演示软件和素材等,以方便教与学。

我社致力于优秀教材的出版,恳切希望老师们在使用的过程中,将发现的问题及时反馈给我们,以便再版时修改。

前 言

当前,普通高等院校对于计算机等相关专业毕业生的程序设计能力要求正在变得越来越高。根据用人单位对于人才需求的日益多样化,高校程序设计类课程开始分化为两条主线。一条主线是 C/C++,这是传统的程序设计课程开设思路,目前拥有非常丰富的教学资源。但是 C/C++教学存在两个主要难点:一是如何从面向过程的程序设计思想向面向对象的程序设计思想转换;二是 C++并不是一个纯粹的面向对象的语言,因此很难在 C++平台上透彻地介绍面向对象的程序设计方法。另一条主线是 Java。作为 1995 年由 Sun 公司组织开发的一种编程语言,Java 是一种纯面向对象的语言,具有高度的可移植和安全性,具备简单性、跨平台、面向对象、多线程等诸多特点,已经成为当今业界面向对象程序设计的主流。采用 Java 作为程序设计课程的编程平台,可以有效克服 C/C++所带来的教学难点。因此,越来越多的高校开始将 Java 作为程序设计课程的主线。

本书是编者多年 Java 程序设计课程教学经验的总结。本书以 Sun 公司目前最新版本 JDK1.6,Java SE 平台为基础,通过 11 章的内容,由浅入深地介绍 Java 程序设计所涉及的各项相关知识。其中,第 1、2 章对 Java 的相关基础知识进行了详细介绍;第 3 章详细讲解了 Java 面向对象编程的特性,以及面向对象程序设计的基本思想。第 4 章详细讲解了 Java 中的列表、集合和映射等知识。第 5 章介绍了 Java 中的数组、String 类的使用。第 6 章讲解了 Java Applet 的基础知识与应用,并通过综合实例进一步说明 Java Applet 的使用。第 7 章介绍了如何使用 Java Swing 进行图形界面程序编程,对其中常用的组件、事件处理、布局管理等都作了详细的介绍,并通过丰富的实例说明其使用方法。第 8 章介绍了 Java 中对数据流的输入输出等操作与文件处理,以及 Java 中对数据库的相关操作。第 9 章讲解了 Java 中线程的基础知识,并通过大量实例说明多线程的同步、调度等管理操作。第 10、11 章主要是 Java 高级程序设计内容,介绍了 Java 中的网络编程,以及异常处理等知识。

在多年的教学过程中我们发现,掌握一门语言最好的方式就是实践。本书的着眼点是将基础理论知识讲解和实践应用相结合,使读者快速掌握 Java 编程技术。在本书的大多数章节中,将首先介绍相关的基础知识,然后重点讲解相关的实例,最后再进行小结,并布置了若干具有代表性的习题和上机练习题,使读者们可以通过自己动手,在实践中掌握 Java 的编程方法和技巧。

本书可作为普通高等院校 Java 程序设计类课程的教材使用,建议安排 48~64 课时的课堂教学,并安排不少于 32 课时的上机实践环节。如果课时有限,同时读者具有 C/C++学习经验,则可以略讲前 3 章内容,但是尽可能不缩减上机实践课时。

本书由张克军、陆迟、孙海伦、庞丽娜共同编写完成。此外,参与本书编辑、修改和整理的还有陈晨、陈凌云、符滔滔、郭瑞、李洋、梁倩、刘丹、刘

振清、王娜、王征等，在此，编者对以上人员致以诚挚的谢意！本书中所有例题和相关代码都经过调试通过，此外本书制作了多媒体课件，供老师教学参考使用。

由于时间仓促和编者的水平有限，书中错误和不妥之处在所难免，敬请读者批评指正。

编 者

2009 年 4 月

目 录

第 1 章 Java 语言概述	1	2.3.2 创建封装类对象	21
1.1 Java 的发展历程	1	2.4 Java 操作符	23
1.2 Java 语言的基本特点	1	2.4.1 算术运算符	23
1.3 Java 虚拟机	3	2.4.2 自增自减运算	27
1.4 Java 开发工具箱	3	2.4.3 关系运算	29
1.4.1 下载与安装 Java 开发工具包 (JDK)	4	2.4.4 逻辑运算	30
1.4.2 JDK 目录	6	2.4.5 三元运算符	32
1.4.3 设置环境变量	7	2.4.6 位运算	34
1.5 学会使用 API	8	2.4.7 移位运算	35
1.6 常见 IDE 介绍	10	2.4.8 赋值运算	35
1.6.1 JCreator 软件	10	2.4.9 括号及运算符间的优先级关系	37
1.6.2 JBuilder 软件	11	2.5 Java 中变量的赋值与类型转换	37
1.6.3 Eclipse 软件	11	2.5.1 基本类型转换	38
1.7 第一个 Java 程序	12	2.5.2 强制类型转换	38
1.7.1 开发源代码	12	2.6 Java 中的控制语句	39
1.7.2 编译运行	13	2.6.1 分支语句	39
小结	13	2.6.2 循环语句	42
习题	14	2.6.3 流程跳转语句	45
上机指导	14	2.7 断言	46
实验: 运算并显示某一算式的运算 结果	14	2.7.1 断言的概念	47
		2.7.2 如何启用/关闭断言	48
		2.7.3 防止滥用断言	50
		小结	51
		习题	51
		上机指导	51
		实验一: 不同类型数据间的运算	51
		实验二: 多重 for 循环绘制图形	52
		实验三: 使用 switch 语句实现条件分支	53
第 2 章 Java 程序设计基础	15		
2.1 Java 中的关键字与标识符的命名规则	15		
2.2 Java 的基本数据类型	16		
2.2.1 boolean 类型	18		
2.2.2 byte、short、int 和 long 类型	18		
2.2.3 float 和 double 类型	20		
2.3 封装类型	21		
2.3.1 封装类型概述	21		
		第 3 章 Java 中的对象与类	54
		3.1 面向对象概述	54
		3.2 类的定义与对象的创建	55

3.3 包的使用	55	4.2 集合 (Set)	101
3.3.1 声明创建包	56	4.2.1 Set 接口及含义	101
3.3.2 引入包内的资源	58	4.2.2 HashSet 类的使用	102
3.4 类的成员变量和方法	60	4.2.3 LinkedHashSet 类的使用	103
3.4.1 访问修饰符	61	4.2.4 SortedSet 接口与 TreeSet 类	104
3.4.2 成员变量	61	4.2.5 自定义满足 Sorted 集合的类	106
3.4.3 成员变量的初始值	62	4.2.6 定制 SortedSet 的排序规则	109
3.4.4 方法	66	4.2.7 集合的遍历	110
3.4.5 构造方法	67	4.3 映射 (Map)	113
3.4.6 参数的传递	68	4.3.1 Map 接口及含义	114
3.5 抽象类与接口	69	4.3.2 HashMap 类的使用	114
3.5.1 抽象类与抽象方法	69	4.3.3 Hashtable 类的使用	116
3.5.2 接口	72	4.3.4 LinkedHashMap 类的使用	116
3.6 final 关键字	74	4.3.5 SortedMap 接口与 TreeMap 类	117
3.7 类的继承	75	4.3.6 映射的遍历	119
3.7.1 如何定义子类	76	小结	121
3.7.2 方法的覆盖	78	习题	121
3.7.3 this 与 super 关键字	80	上机指导	122
3.7.4 继承的层次结构	81	实验: 使用迭代器完成遍历	122
3.8 static 关键字	82	第 5 章 数组与字符串	124
小结	86	5.1 数组的声明与创建	124
习题	87	5.1.1 声明数组引用	124
上机指导	88	5.1.2 创建数组对象	125
实验一: 对象引用型成员变量的使用	88	5.2 Java 中数组的实现机制	126
实验二: 含有返回值的方法的使用	89	5.3 数组的初始化	128
实验三: 使用对象模拟“商店客户折 扣卡”功能	90	5.3.1 默认初始化	128
第 4 章 集合	93	5.3.2 利用循环初始化	129
4.1 列表 (List)	94	5.3.3 枚举初始化	130
4.1.1 列表接口	95	5.4 数组的相互赋值	131
4.1.2 列表的数组实现	96	5.4.1 基本类型数组赋值规则	131
4.1.3 向量	98	5.4.2 引用类型数组赋值规则	132
4.1.4 列表的链接实现	99	5.5 数组的基本操作	133
4.1.5 将数组转换为列表	100	5.5.1 数组复制	133
		5.5.2 数组排序	135

5.5.3 搜索指定元素	135	6.4.2 示例——使用 Applet 实现 计算器功能	164
5.5.4 比较数组中的元素	136	6.4.3 示例——使用 Applet 来处理 声音文件	167
5.6 关于 args[]	136	6.4.4 示例——使用 Applet 显示图像	168
5.7 String 类	137	小结	170
5.7.1 对象的创建	138	习题	170
5.7.2 巧用构造器	138	上机指导	171
5.7.3 String 类的重要方法	139	实验一: Applet 简单使用	171
5.7.4 String 的内存机制	141	实验二: 在 Applet 中使用 BorderLayout	172
5.8 StringBuffer 类	145	实验三: Java Applet 中使用字体列表 显示各种字体	173
5.8.1 弥补 String 不足的 StringBuffer 类	145	第 7 章 图形用户界面编程	175
5.8.2 编写方法链以及 StringBuffer 类的重要方法	146	7.1 Swing 和 AWT 的对比	175
5.9 正则表达式	147	7.1.1 Swing 与 AWT 之间的关系	175
5.9.1 正则表达式的基本语法	148	7.1.2 Swing 与 AWT 的控件的混用 建议	176
5.9.2 Pattern 类简介	150	7.2 一个简单的图形用户界面程序	177
5.9.3 Matcher 类简介	151	7.3 容器类	179
5.9.4 Pattern 与 Matcher 类的综合 应用	151	7.3.1 顶级容器概述	179
5.9.5 利用正则表达式进行查找替换	153	7.3.2 JFrame 主窗口	179
小结	154	7.3.3 JDialog 对话框窗口	180
习题	154	7.3.4 JPanel 面板的使用	180
上机指导	155	7.4 布局管理器	180
实验一: 编写数组复制的程序	155	7.4.1 布局管理器的概述	180
实验二: 改变 String 对象引用的指向	156	7.4.2 边框布局 BorderLayout	181
第 6 章 Java Applet	157	7.4.3 流动布局 FlowLayout	182
6.1 Applet 概述	157	7.4.4 卡片布局 CardLayout	182
6.1.1 Applet 的基本工作原理	157	7.4.5 网格布局 GridLayout	182
6.1.2 如何在网页中嵌入 Applet	157	7.4.6 使用绝对定位	183
6.1.3 Applet 的安全性	159	7.5 Java 事件处理机制	184
6.2 Applet 的基本方法	159	7.5.1 事件处理模型	185
6.3 向 Applet 传递参数	161	7.5.2 为组件注册事件	185
6.4 Applet 综合示例	162	7.6 常见的事件处理	186
6.4.1 示例——使用 Applet 显示当前 日期及时间	163		

7.6.1 鼠标事件	186	8.4.1 示例——读取文件并显示	217
7.6.2 鼠标移动事件	188	8.4.2 读取文件的分类及所涉及的类	219
7.6.3 选项事件	189	8.5 写入文件	220
7.6.4 窗口事件	189	8.5.1 示例——将输入的文本保存到文件	220
7.6.5 键盘事件	189	8.5.2 写入文件的分类及所涉及的类	222
7.7 Swing 组件	191	8.5.3 写入文件 Writer 类的基本方法	222
7.7.1 按钮	192	8.6 随机访问文件	222
7.7.2 文本组件	192	8.6.1 RandomAccessFile 类的构造及内部方法	223
7.7.3 单选按钮 JRadioButton	195	8.6.2 示例——使用随机访问文件流来对文件进行读写	224
7.7.4 复选框 JCheckBox	195	8.7 创建压缩文件	225
7.7.5 下拉列表 JComboBox	196	8.7.1 ZipFile 文件压缩类概述	225
7.7.6 滑块 JSlider	197	8.7.2 实现数据文件 Zip 方式的压缩方法	226
7.7.7 进度条 JProgressBar 及 Timer 组件	197	8.7.3 将.class 文件打包成 jar 文件	229
7.7.8 菜单 JMenu 概述	199	8.8 数据库操作	229
7.7.9 工具栏 JToolBar	199	8.8.1 使用 JDBC 连接数据库	229
7.8 一个综合的 Swing 组件示例	200	8.8.2 执行 SQL 语句	235
小结	202	小结	238
习题	203	习题	238
上机指导	203	上机指导	239
实验一：组合使用文本区与滚动窗口	203	实验一：管理目录文件	239
实验二：使用滑块与进度条	205	实验二：学习使用字节流	240
实验三：文件、颜色对话框综合案例	207	实验三：利用 JDBC 连接数据库	241
第 8 章 Java 输入输出与数据库操作	209	第 9 章 多线程——Java 中的并发协作	244
8.1 流概述	209	9.1 线程的基本知识	244
8.1.1 流的定义	209	9.1.1 多线程编程的意义	244
8.1.2 输入输出类结构	209	9.1.2 定义自己的线程	245
8.2 一个读取文件的简单实例	211	9.1.3 创建线程对象	246
8.3 文件类 (File) 的使用	212	9.1.4 启动线程	247
8.3.1 获取指定文件的描述信息	213	9.1.5 同时使用多个线程	249
8.3.2 创建新文件	214	9.2 线程的状态	251
8.3.3 文件改名及文件删除	215		
8.3.4 获取指定文件目录下的所有文件	216		
8.4 读取文件	217		

9.3 线程的调度.....252	10.4 使用 InetAddress 类获取 IP 地址 信息.....283
9.3.1 睡眠.....252	10.5 使用套接字 Socket 类.....285
9.3.2 线程的优先级.....253	10.5.1 Socket 类与 ServerSocket 类.....285
9.3.3 线程的让步.....253	10.5.2 建立与服务器的连接.....286
9.3.4 守护线程.....254	10.6 使用套接字 Datagram 类.....291
9.4 线程的同步.....254	10.6.1 Datagram (数据报) 套接字.....291
9.4.1 同步方法简介.....255	10.6.2 使用 Datagram 实现数据传输.....292
9.4.2 简单使用同步方法.....255	10.7 网络程序经典示例——聊天室程序.....294
9.4.3 线程同步调度的方法.....258	10.7.1 聊天室基本原理.....294
9.4.4 同步的语句块.....258	10.7.2 客户端程序.....294
9.4.5 线程的死锁.....259	10.7.3 服务器端程序.....298
9.5 进程同步经典示例.....261	小结.....305
9.5.1 “生产者—消费者”案例的 框架.....262	习题.....305
9.5.2 “生产者—消费者”案例的 实际运行.....265	上机指导.....306
9.5.3 notify 方法的使用.....266	实验一：使用 URL 标识网络中计算机 上的资源.....306
9.6 volatile 关键字的含义与使用.....268	实验二：打印主机对应 IP 地址的 InetAddress 对象的主机名及 地址.....307
小结.....269	实验三：使用 Socket 编写 C/S 架构 程序.....307
习题.....269	
上机指导.....270	
实验一：Sleep 方法的学习使用.....270	
实验二：利用线程同步实现生产者与 消费者模式.....272	
实验三：线程的优先级.....275	
第 10 章 网络程序设计.....277	第 11 章 异常处理.....310
10.1 网络通信概述.....277	11.1 异常处理的基本知识.....310
10.1.1 什么是网络通信.....277	11.1.1 try 和 catch 捕获异常.....310
10.1.2 网络协议基础.....277	11.1.2 异常的传播过程.....313
10.1.3 Java 中所涉及到的网络应 用类.....278	11.1.3 finally 语句块的使用.....315
10.2 一个简单的网络编程实例.....278	11.2 异常的层次结构.....317
10.3 使用 URL 类.....280	11.2.1 捕获异常.....318
10.3.1 URL 类中常用的方法.....280	11.2.2 未捕获异常.....320
10.3.2 使用 URLConnetion 类.....281	11.3 再次抛出异常.....321
	11.3.1 什么是异常的再抛出.....321
	11.3.2 显性再抛出.....322
	11.3.3 隐性再抛出.....324
	11.4 定义自己的异常.....326

11.4.1 创建自己的异常类.....	326	习题.....	332
11.4.2 使用自定义的异常类.....	327	上机指导.....	334
11.5 异常的匹配.....	329	实验一: finally 语句的使用.....	334
11.5.1 同时捕获多种异常.....	330	实验二: 创建自己的异常类.....	335
11.5.2 多个 catch 语句的先后顺序.....	331	实验三: 匹配异常.....	335
小结.....	332		

第 1 章

Java 语言概述

Java 是一种简单易用、完全面向对象、有平台无关性、安全可靠的开发工具。同时，也是一种使建立分布式应用程序更简单、更易行的技术。分布式应用程序是指由网络上多个计算机共同执行的程序。Java 是第一个允许用户将应用程序通过 Internet 从远端服务器传输到本地计算机上并执行的一种语言。

1.1 Java 的发展历程

Java 的历史要追溯到 1991 年，Sun MicroSystem 公司的 Jame Gosling、Bill Joe 等人为在家用消费类电子产品上进行交互式操作而开发了一个名为 Oak 的软件。当时，并没有引起人们的关注。后来发现 Oak 是一种已有计算机语言的名字，于是将其更名为 Java。Java 这个名字最早是源于爪哇岛出产的美味咖啡，采用 Java 命名该语言表示给每个使用 Java 进行开发的程序员呈上了一杯美味的咖啡。

1996 年初，Sun 公司发布了 Java 1.0，但很快人们就发现其缺陷很严重。虽然后来 Java 1.1 的发布弥补了其中大部分明显的缺陷，为 GUI 编程增加了新的事件处理模型，但仍然具有很大的局限性。

1998 年，Sun 公司发布了 Java 1.2 版本，也称为 Java 2。该版本更加接近了“一次开发，随处运行”的承诺。后来的 1.3 及 1.4 版本扩展了标准类库，提高了系统性能，同时为 Java 语言添加了很多新的特性。在这期间，随着网络的发展，Java 也在各个领域得到了广泛的应用。

1999 年 6 月，Sun 公司正式发布 Java 的 3 个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME）。

2004 年，Java 1.5 发布。在该年度的 JavaOne 会议之后，版本数字升为 5.0。该版本为 Java 添加了很多新的特性，如泛型、安全枚举类型、程序注释等。这些特性大大简化了 Java 的开发。

2006 年底，Java 6.0 发布，不但进一步提升了性能，还加入了对各种脚本语言嵌入式开发的支持等特性，大大提升了 Java 的开发能力和易用性。

此时，Java 的各种版本已经更名，取消其中的数字“2”。J2EE 更名为 Java EE；J2SE 更名为 Java SE；J2ME 更名为 Java ME。

1.2 Java 语言的基本特点

从 Java 诞生以来，就以其在网络应用开发上独特的魅力以及“一次开发，随处运行”的

可移植性引起了人们极大的兴趣。自 1994 年以来, Internet 与万维网迅速发展, 这大大促进了以网络开发见长的 Java 的发展, 使其逐渐成为 Internet 上最受欢迎的编程平台。下面从几个不同的方面介绍 Java 的一些特点, 并且通过与 C++ 的对比, 进一步指出其在开发方面的优势。

1. 简单性

与 C++ 语法相比, Java 语法是一个“升华”版, 其剔除了 C++ 中许多易混淆、难理解的特性。但在设计 Java 的时候, 还是尽可能地接近 C++, 使系统更易于理解。在 Java 中, 略去了头文件、指针运算、操作符重载、多重继承等模糊的概念, 并且通过自动垃圾收集大大简化了程序员的内存管理工作。

同时, Java 也适于在嵌入式系统上运行。其基本解释器以及类库的支持仅为 40KB 左右, 加上多线程的支持也只有 175KB 左右, 这是一个杰出的成就。

2. 面向对象

在相当长时间里, 面向对象已经证明了其自身的价值。Java 的面向对象特性与 C++ 类似, 实现了模块化和信息隐藏。类提供了对象的原型, 子类通过继承机制可以使用父类所提供的数据与方法。Java 语言的设计集中于对象及其接口上。与 C++ 不同的是 Java 取消了多重继承, 取而代之的是接口概念, 使 Java 更加易用。

3. 分布式

Java 在网络方面的强大应用是其他任何语言无法比拟的, 可以说 Java 是面向网络的语言。通过其提供的类库可以方便地处理各种网络协议, 方便地进行传统的套接字网络开发, 如 RMI、CORBA、Web 服务等现在流行的网络开发。

4. 健壮性

Java 在编译和运行时, 都会对程序可能出现的问题进行检查, 并将出错信息报告给程序员。同时, 提供垃圾收集机制来自动管理内存, 避免了程序员很多无心的错误和恶意的攻击。而 C++ 需要程序员自己编写代码管理内存, 这对大多数程序员来说并不是一项愉快的工作。对曾经花费很多时间来检查悬浮指针引起内存冲突的程序员, 一定非常喜欢 Java 的这一特性。

5. 可移植性

与 C++ 不同, Java 规范中没有“依赖具体实现”的地方。与平台无关的特性可以方便地将 Java 部署到任何不同平台的机器上。同时, Java 的类库封装了不同平台上的实现, 为其提供统一的接口, 这使得同样的类库可以在不同的平台上使用。也就意味着用 Java 开发的应用可以“一次开发, 随处运行”。

6. 安全性

在安全性方面, Java 也是首屈一指的, 其是目前极少的本身具有安全性的开发平台的语言。而 JCP (Java Community Process 的缩写, 是制定 Java 标准的一个委员会, 由很多大的厂商和机构参加, 如 Sun、IBM、BEA、Oracle 等都是 JCP 的成员) 也大力支持对 Java 安全性的研究, 制定了使用规范, 实现了虚拟机安全机制和安全库, 添加了数字签名类的概念。从一开始 Java 就被设计为有防范各种病毒、袭击的能力, 举例如下:

- (1) 一切对内存的访问都必须通过对象的实例引用来实现;
- (2) 禁止破坏自己处理空间之外的内存;
- (3) 禁止运行时堆栈溢出;
- (4) 未经授权禁止读写文件。

1.3 Java 虚拟机

Java 虚拟机 (Java Virtual Machine, JVM) 是一个抽象的机器, 在计算机中通过软件模拟来实现。Java 虚拟机有其自己虚拟的硬件, 如处理器、堆栈、寄存器等, 还具有相应的指令系统。

Java 虚拟机是实现 Java 语言的平台无关性的关键。如果是一般的高级语言需要在不同的平台上运行, 那么就至少需要编译成不同的目标代码。而引入 Java 虚拟机后, Java 语言在不同平台上运行时就不需要重新编译。

Java 虚拟机屏蔽了与具体平台相关的信息, 使得 Java 语言编译程序只需生成在 Java 虚拟机上运行的目标代码——字节码, 就可以在多种平台上不加修改地运行。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, 如图 1.1 所示。

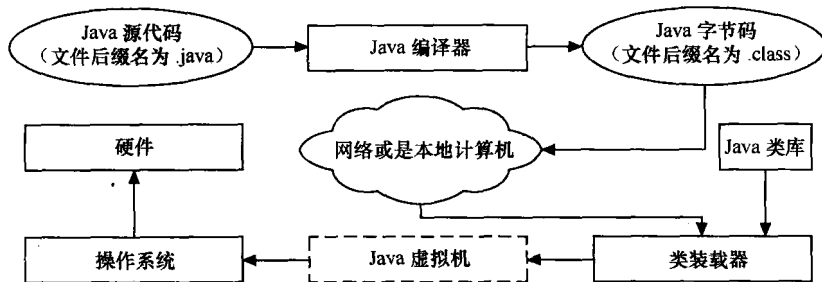


图 1.1 Java 虚拟机的作用

从图 1.1 可以看出, 开发人员编写 Java 代码 (后缀名为 .java 的文件), 然后将之编译成字节码 (后缀名为 .class 的文件)。最后字节码被装入内存, 一旦字节码进入虚拟机, 它就会被虚拟机中的代码解释器解释执行, 或者被虚拟机中的即时编译器有选择地转换成机器码执行。

简单来说, Java 虚拟机在它的生存周期中的主要任务就是运行 Java 程序, 因此当 Java 程序启动的时候, 就产生 Java 虚拟机的一个实例, 当程序运行结束的时候, 该实例随之消失。

在 Java 代码从编写到执行的整个流程中, Java 虚拟机处于核心位置, 这是 Java 程序与底层操作系统和硬件无关的关键。

1.4 Java 开发工具箱

JDK 是 Java Development Kit 的缩写, 即 Java 开发工具包。在某些场合下, 也能看到 SDK 这样的旧术语, 它是 Software Development Kit 的缩写。JDK 是一个编写 Java 的 Applet 和应用程序的程序开发环境。它由一个处于操作系统层之上的运行环境还有开发者编译、调试和运行用 Java 语言编写的 Applet 和应用程序所需的工具组成。

目前 Sun 公司官方网站 (<http://www.sun.com/download/>) 发布的最新版是 Java (TM) SE Development Kits, 即 JDK6。

1.4.1 下载与安装 Java 开发工具包 (JDK)

这里下载的是最新的 JDK6，在下载页面 <http://java.sun.com/javase/downloads/index.jsp> 找到最新的 JDK 6 Update 4，单击进行下载。关于 JDK6 的更加详细的信息可以浏览 Sun 公司官方网页 <http://java.sun.com/javase/6/webnotes/ReleaseNotes.html> 查看，其中详细记录了 JDK6 的新变化及漏洞的修正等问题。下载页面如图 1.2 所示。

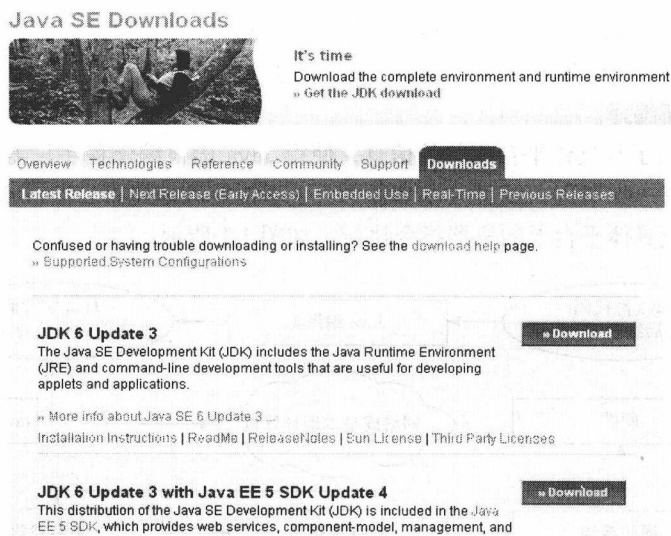


图 1.2 下载页面 1

(1) 单击如图 1.2 所示页面下边的【Download】按钮，进入如图 1.3 所示的下载页面。页面上方提示“Required: You must accept the license agreement to download the product.”，选中该句下方的【Accept License Agreement】单选按钮，接受授权协议后，才能够进行下载。

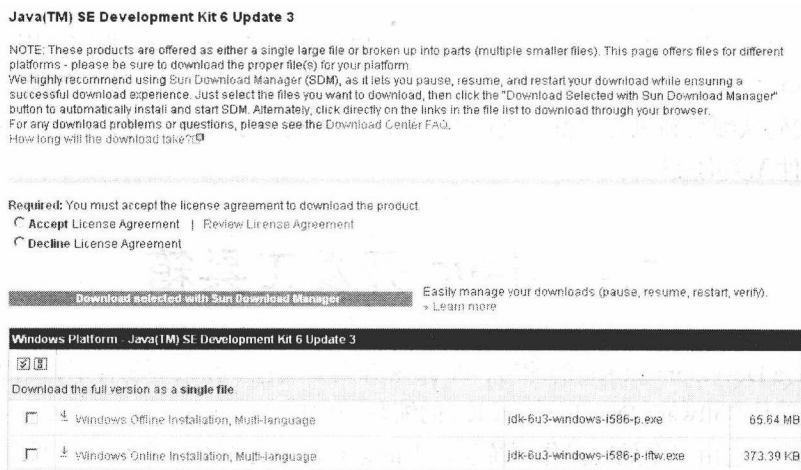


图 1.3 下载页面 2

(2) 下载页面中提供了针对 Windows、Linux 等不同操作系统平台的 JDK 版本，如图 1.3 所

示。针对每个平台又有两个版本，一个是 Windows Offline Installation, Multi-language，另一个是 Windows Online Installation, Multi-language；前一个版本是先下载安装文件，再安装，后一个则是在线安装。本节以先下载后安装为例。下载到文件名为 jdk-6u3-windows-i586-p.exe 的安装文件，双击进入安装过程。

(3) 出现安装向导画面（如图 1.4 所示），单击【下一步】按钮，打开【许可证协议】对话框，如图 1.5 所示。用户阅读许可证协议后，单击【接受】按钮，打开【自定义安装】对话框，如图 1.6 所示。

(4) 在【自定义安装】对话框中读者可以自己选择安装目录。书中示例安装至默认文件夹，选择安装目录后，单击【下一步】按钮，继续安装。安装先经过解压缩文件，再配置环境等，需要耐心等待几分钟，如图 1.7 所示。

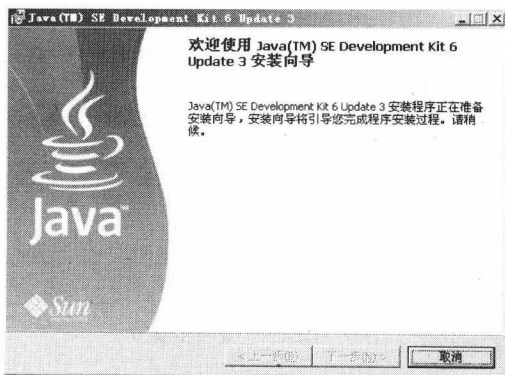


图 1.4 安装步骤 1

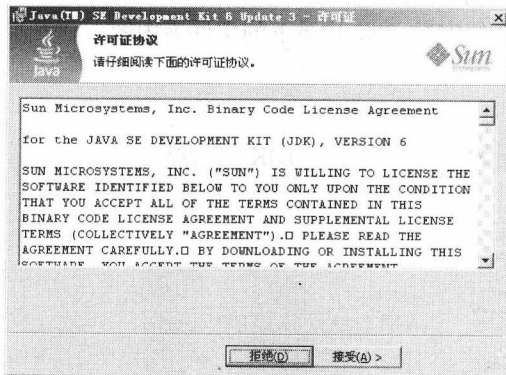


图 1.5 安装步骤 2

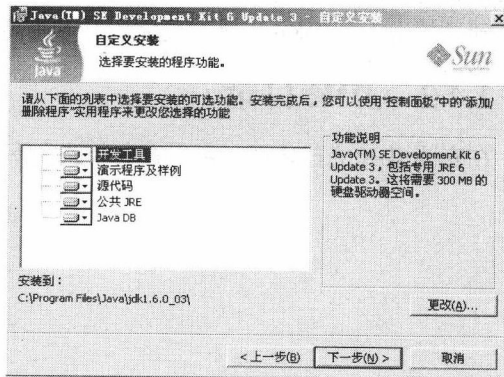


图 1.6 安装步骤 3

(5) JDK 成功安装后，会自动继续默认安装 JRE，如图 1.8 所示。单击【下一步】按钮，继续安装，如图 1.9 所示。

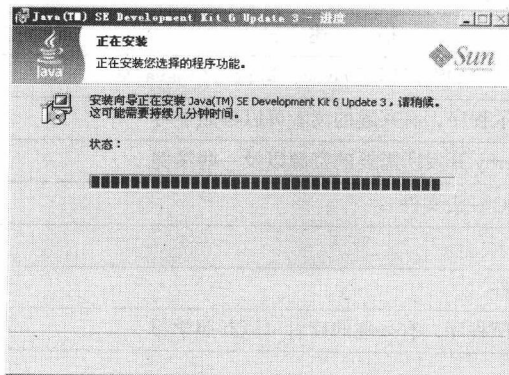


图 1.7 安装步骤 4

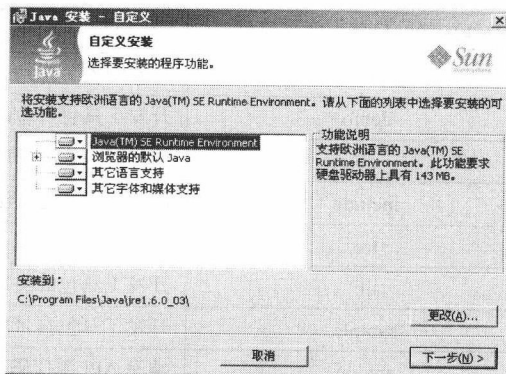


图 1.8 安装步骤 5

(6) 最后安装完成，可以选择显示自述文件，如图 1.10 所示。单击【完成】按钮，完成安装。