

21 世纪高等学校计算机基础实用规划教材

Java 程序设计教程

温秀梅 祁爱华 编著

清华大学出版社

21 世纪高等学校计算机**基础**实用规划教材

Java程序设计教程

温秀梅 祁爱华 编著

清华大学出版社

内 容 简 介

本书通过具有代表性的例子、详尽的讲解和丰富的练习全面介绍了 Java 2 的相关知识。本书在体系结构的编排上由浅入深,重点、难点突出,对于典型例题进行了分析解释,还在附录中整合了习题、实验、课程设计、模拟题等,使全书结构严谨、通俗易懂,兼有普及与提高的双重功能。

本书由 3 个部分组成:第一部分(第 1~6 章、第 8 章)对 Java 的基本特性、面向对象程序设计知识以及语法进行了较为系统的介绍;第二部分(第 7 章、第 9~12 章)是关于 Java Applet 编程、图形用户界面、网络编程及数据库技术等内容的介绍;第三部分是本书的附录部分,包括实验、课程设计及模拟题等。

本书本着少而精的原则,全书版面清晰、结构紧凑,知识信息含量高,特别适合作为计算机专业、非计算机专业的本科生教材或计算机应用培训班的教材,同时,还可以作为自学或函授学习的参考书。

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 程序设计教程/温秀梅,祁爱华编著. --北京:清华大学出版社,2014

21 世纪高等学校计算机基础实用规划教材

ISBN 978-7-302-36753-6

I. ①J… II. ①温… ②祁… 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(2014)第 124153 号

责任编辑:魏江江 王冰飞

封面设计:常雪影

责任校对:焦丽丽

责任印制:王静怡

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 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:23.5 字 数:571 千字

版 次:2014 年 9 月第 1 版 印 次:2014 年 9 月第 1 次印刷

印 数:1~2000

定 价:44.50 元

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

Java 是一种纯面向对象程序设计语言。Java 具有跨平台使用、安全性好、支持多线程和分布式计算等特点,被广泛应用于基于 Internet 的网络应用开发。

在编写本书之前,作者已从事了多年的“Java 语言程序设计”、“C/C++ 程序设计”等课程的教学及科研工作,对于该语言的基本概念、功能及应用有着较深入的理解和丰富的实践经验。本书在内容安排、教学深度、习题、实验及课程设计等方面做了细致的考量,以便更加贴近“Java 程序设计”课程的教学需求。

本书以现代教育理念为指导,在讲授方式上注意结合应用开发实例,注重培养学生理解面向对象程序设计思想,以提高分析问题和解决实际问题的能力。本书采用由浅入深、理论与实践相结合的教学思路,通过大量的实例阐述 Java 程序设计的基本理念及编程技巧。

本书本着少而精的原则,全书版面清晰、结构紧凑,知识信息含量高,特别适合作为计算机专业、非计算机专业的本科生教材或计算机应用培训班的教材,同时,还可以作为自学或函授学习的参考书。

如果教学课时不足,第 9 章到第 12 章内容可以有选择地讲解,为了提高教学质量,建议增加更多的课外学时。此外,读者在进行编程实践时应当养成查阅“帮助文档”的习惯。

本书由河北建筑工程学院温秀梅、祁爱华编著,孟凡兴、赵巍、孙皓月任副主编,参加编写的还有王利霞、岳杰、赵建光、李风云、庞炜、赵翊君、王剑雄,全书由温秀梅教授进行审校并统稿。

在编写本书的过程中,编者参考了大量的相关文献,在此对文献的作者表示深深的感谢。

本书在编写和出版的过程中得到了清华大学出版社的大力支持和帮助,在此表示诚挚的感谢。

由于编者水平有限,书中难免有疏漏、错误和不妥之处,恳请广大专家和读者批评指正,我们将不胜感激。

编者

2014 年 3 月

目 录

第 1 章 Java 概述	1
1.1 Java 的诞生与发展	1
1.2 Java 的特点	2
1.3 Java 开发运行环境的安装与配置	4
1.3.1 安装 JDK	4
1.3.2 安装 Java 帮助文档	5
1.3.3 配置环境变量	7
1.4 Java 的工作原理与运行系统	8
1.4.1 Java 的工作原理	8
1.4.2 Java 的运行系统	9
1.5 简单的 Java 程序	10
1.5.1 Java 应用程序(Application)	10
1.5.2 Java 小应用程序	16
习题	19
第 2 章 Java 语言基础	20
2.1 标识符和关键字	20
2.1.1 标识符	20
2.1.2 关键字	21
2.2 基本数据类型	21
2.3 常量、变量	22
2.3.1 常量	23
2.3.2 变量	26
2.4 运算符与表达式	29
2.4.1 赋值运算符和赋值表达式	30
2.4.2 算术运算符和算术表达式	32
2.4.3 关系运算符和关系表达式	33
2.4.4 逻辑运算符和逻辑表达式	34
2.4.5 条件运算符和条件表达式	34
2.4.6 位运算符和位表达式	35

2.5 流程控制语句.....	36
2.5.1 顺序结构	36
2.5.2 选择结构	37
2.5.3 循环结构	46
2.6 Java 开发类库的组成	60
习题	62
第3章 类和对象	63
3.1 概述.....	63
3.1.1 对象	63
3.1.2 类	64
3.1.3 消息	64
3.1.4 面向对象系统的特性	65
3.2 类.....	66
3.2.1 类的定义	66
3.2.2 方法重载	70
3.2.3 构造方法	70
3.3 对象的定义和使用.....	71
3.3.1 创建对象	71
3.3.2 对象的使用	71
3.3.3 对象的清除	72
3.3.4 方法的参数传递	73
3.4 实例变量、实例方法和类变量、类方法.....	75
3.4.1 变量与方法	75
3.4.2 变量的作用域	77
3.4.3 变量的初始值	78
3.5 this 关键字	78
3.6 包.....	79
3.6.1 包的定义	80
3.6.2 设置类路径	81
3.6.3 包的使用	82
3.7 内部类和匿名类.....	84
3.7.1 内部类	84
3.7.2 匿名类	87
3.8 访问控制和类的封装性.....	88
3.8.1 访问控制	88
3.8.2 类的封装性	88
习题	89

第 4 章 数组、字符串和向量	90
4.1 数组	90
4.1.1 一维数组	90
4.1.2 二维数组	102
4.2 字符串	107
4.2.1 String 类	108
4.2.2 StringBuffer 类	115
4.2.3 StringTokenizer 类	118
4.3 向量	120
习题	124
第 5 章 继承性和多态性	125
5.1 继承性	125
5.1.1 派生类的定义	125
5.1.2 super 关键字	126
5.1.3 protected 和 final 关键字	131
5.2 多态性	134
5.3 抽象类和接口	137
5.3.1 抽象类	137
5.3.2 接口	140
5.4 初始化块	149
习题	151
第 6 章 异常处理	152
6.1 异常处理概述	152
6.1.1 什么是 Java 异常	152
6.1.2 异常处理的目的	152
6.2 异常类的分类与层次	154
6.2.1 异常类的层次结构	154
6.2.2 Exception 类	155
6.2.3 自定义异常类	156
6.3 Java 的异常处理机制	156
6.3.1 捕获异常	156
6.3.2 声明抛出异常	159
习题	162
第 7 章 图形用户界面	163
7.1 Java GUI 概述	163

7.1.1	AWT 与 Swing	163
7.1.2	Swing 组件	165
7.1.3	Swing 容器	166
7.1.4	基于 Swing 的 Java GUI 设计思路	171
7.2	图形与绘图	172
7.2.1	绘图表面	172
7.2.2	图形环境和图形对象	172
7.2.3	颜色	174
7.2.4	字体	175
7.2.5	在面板上绘图	176
7.3	事件处理与 Swing 常用组件	180
7.3.1	Java 事件处理机制	180
7.3.2	精简事件处理程序的技巧	183
7.3.3	JLabel 组件与鼠标事件处理	185
7.3.4	按钮、文本字段与动作事件	189
7.3.5	单选按钮、复选框、组合框与项目事件	194
7.3.6	列表与 ListSelectionEvent 事件	200
7.3.7	进度条与时间组件	202
7.3.8	菜单与工具栏	204
7.3.9	表格组件与树	208
7.3.10	桌面面板与内部窗口	212
7.4	布局管理器	215
7.4.1	FlowLayout 布局管理器	216
7.4.2	BorderLayout 布局管理器	218
7.4.3	GridLayout 布局管理器	219
7.4.4	CardLayout 布局管理器	222
7.4.5	BoxLayout 布局管理器与 Box 类	224
习题		227
第 8 章	文件、流和输入/输出技术	228
8.1	I/O 流概述	228
8.2	I/O 流类	229
8.2.1	字符流	230
8.2.2	字节流	231
8.2.3	标准输入/输出	232
8.3	过滤流与流的串接	233
8.4	文件流类与文件操作	236
8.4.1	File 类	236
8.4.2	FileInputStream 类、FileOutputStream 类与读/写二进制文件	239

8.4.3	RandomAccessFile 类	243
8.4.4	FileReader 类、FileWriter 类与读/写文本文件	245
8.5	对象序列化	248
8.5.1	对象流	248
8.5.2	序列化举例	249
习题		251
第 9 章	Java 的 Applet 编程	252
9.1	Java Applet 概述	252
9.1.1	Applet 的生命周期	253
9.1.2	Applet 的绘图周期	254
9.2	实现一个简单的 Applet 程序	255
9.2.1	Applet 程序的编写与编译	255
9.2.2	将 Applet 嵌入 Web 网页	255
9.2.3	Applet 的执行	257
9.3	Applet 生命周期的状态切换举例	259
9.4	Applet 的常用方法	260
9.4.1	获取 Applet 的有关信息	261
9.4.2	获取多媒体文件	262
9.4.3	同页 Applet 间的通信	263
9.5	Applet 的安全限制	265
习题		266
第 10 章	多媒体与多线程	267
10.1	使用 Graphics2D 类绘图	267
10.1.1	Java 2D 图形	267
10.1.2	绘制 Java 2D 图形	267
10.2	图像的操作	269
10.2.1	图像的创建、加载和显示	269
10.2.2	图像映射	271
10.3	加载和播放音频剪辑	273
10.4	多线程	275
10.4.1	多线程概述	275
10.4.2	创建线程的方式	276
10.4.3	线程的生命周期及控制	279
10.4.4	多线程的互斥与同步	282
10.4.5	守护线程	290
10.5	动画	291
10.5.1	使用 Timer 类显示动画	291

10.5.2 使一系列图像成为动画	294
习题	298
第 11 章 利用 JDBC 访问数据库	299
11.1 数据库与 JDBC 概述	299
11.1.1 数据库概述	299
11.1.2 SQL	300
11.1.3 什么是 JDBC	302
11.1.4 JDBC 与 ODBC	303
11.1.5 ODBC 数据源设置	303
11.2 通过 JDBC 访问数据库	305
11.2.1 JDBC 数据库驱动程序	305
11.2.2 JDBC API	307
11.2.3 编写 JDBC 程序的一般步骤	310
11.3 ATM 应用程序	312
习题	318
第 12 章 Java 网络程序设计	319
12.1 Java 与网络通信	319
12.2 URL 通信	321
12.2.1 InetAddress 类	322
12.2.2 URL 类	323
12.2.3 URLConnection 类	325
12.3 TCP Socket 通信	326
12.4 数据报通信	334
习题	338
附录 A Java 支持的编码集	339
附录 B 应用程序的打包与发布	340
附录 C 实验	343
附录 D Java 课程设计	349
附录 E 模拟试题	353
附录 F 模拟试题参考答案	357
附录 G 参考课时安排	361
参考文献	362

Java 是 Sun 公司(2009 年 4 月 20 日被美国 Oracle 公司收购)推出的新一代面向对象程序设计语言,具有安全、体系结构中立、面向对象、简单、稳定、解释执行、多线程等显著特点,是目前使用最为广泛的网络编程语言之一。

1.1 Java 的诞生与发展

Java 来自于 Sun 公司的一个名为 Green 的项目,早期的成员是 Patrick Naughton、James Gosling 和 Mike Sheridan,现在大家较为熟悉的成员是 James Gosling,其最初的开发目的是为家用消费电子产品开发一个分布式代码系统,通过把 E-mail 发给电冰箱、电视机等家用电器来实现对它们的控制。在开始的时候准备采用 C++,但 C++ 太复杂,且安全性差,最后基于 C++ 开发了一种新的语言 Oak(Java 的前身)。Oak 是一种用于网络的精巧、安全的语言,Sun 公司曾以该语言投标一个交互式电视项目,但结果被 SGI 打败,恰巧这时 Mark Ardreesen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员,他们用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,触发了 Java“进军”Internet。Java 的名称来源于印度尼西亚的一个岛屿“爪哇”,此岛盛产咖啡。Java 应用程序标题栏默认图标是一杯热气腾腾的咖啡,这就是 Java 程序的标志,其寓意是为世人端上一杯热气腾腾的咖啡。

Java 的快速发展得益于 Internet 和 Web 的出现,Internet 上有各种不同的计算机,它们可能使用完全不同的操作系统和 CPU 芯片,但仍希望运行相同的程序,Java 的出现标志着真正的分布式系统的到来。

自从 1995 年 Sun 公司正式发布 Java 1.0 版本以来,在全球范围内引发了经久不衰的 Java 热潮,Java 的版本依次更新为 v1.1、v1.2、v1.3、v1.4、v1.5(现更名为 5.0 版本)、v1.6(现更名为 6.0 版本)、v1.7(现更名为 7.0 版本),其内容也有了巨大的改进和扩充。

1999 年,Sun 公司重新组织 Java 平台的集成方法,加强了 Java 企业级应用平台的功能。目前,Java 程序支持智能消费型电子产品的开发、各种应用程序的开发,尤其是网络程序的开发,Java 语言拥有“互联网上的世界语”的美称。

Java 不仅是一种编程语言,还是一个开发运行平台。目前,Java 平台划分为 Java SE、Java EE 和 Java ME 共 3 个平台,针对不同的市场目标和设备进行定位。

(1) Java SE(Java Platform, Standard Edition): Java 标准版。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序,也可以开发 Java Applet 小应用程序。用户在学习 Java 的过程中,主要用 J2SE 进行程序的开发。

(2) Java EE(Java Platform, Enterprise Edition): Java 企业版。Java EE 定义了基于组件的多层企业级应用的开发标准,面向企业级和高端服务器的 Internet 应用开发。Java EE 是基于 Java SE 的,主要包括 Enterprise JavaBeans(EJB)、Java Servlets API 以及 Java Server Pages(JSP)等技术,并为企业级应用的开发提供了各种服务和工具。

(3) Java ME(Java Platform, Micro Edition): Java 微型版。该平台是用于嵌入式应用的 Java 平台,它是为无线电子市场设计的,被应用在各种各样的电子消费产品上,例如智能卡、手机、PDA 和电视机顶盒等产品。

1.2 Java 的特点

Java 的特点主要表现在简单、面向对象、垃圾自动回收机制、分布式计算、稳定、安全、解释执行、体系结构中立、可移植性、多线程以及异常处理等方面。

1. 简单性

Java 的语法规则和 C/C++ 类似,所以熟悉 C/C++ 的编程人员较容易掌握 Java 的编程技术。Java 摒弃了 C++ 中容易引发程序错误的概念,例如指针、运算符重载、多重继承等。通过 Java 提供的丰富类库大大简化了 Java 程序的开发。Java 对环境要求非常低,它的基本解释器及类的支持只有几十千字节,加上标准类库和线程的支持也只有两百多千字节。

2. 面向对象

Java 是一种纯面向对象的程序设计语言。面向对象的程序设计思想是对面向过程程序设计思想的一次革命,是一种以数据(对象)及其接口为中心的程序设计技术,面向对象程序设计可以说是定义程序模块如何“即插即用”的机制。与以前的结构化程序设计相比,它能够更好地模拟现实世界,有利于开发人员对应用程序的构思。Java 的面向对象机制实际上可以看作是 C++ 面向对象机制的延伸。Java 提供了简单的类机制和动态的构架模型,对象中封装了它的数据成员(状态变量)和成员方法(函数、过程),实现了模块化和信息隐藏;而类是创建一组相似对象的一种模板,通过继承、重载和覆盖机制,子类可以使用或者重新定义父类(也称为超类)所提供的成员,从而实现代码的复用。

3. 垃圾自动回收机制

在程序的执行过程中,部分内存使用后处于闲置状态,如果不及时回收这些垃圾内存就会造成内存泄露,甚至导致系统崩溃。在 C++ 中,由程序员负责内存的回收,内存管理增加了程序员的工作量。而在 Java 运行环境中,始终存在着一个系统级的线程,专门跟踪内存的使用情况,定期检测不再使用的内存并进行自动回收。Java 的垃圾自动回收机制在简化程序开发的同时提高了程序的稳定性和可靠性。

4. 分布式计算

Java 具有支持分布式计算的特征。在分布式计算中,“分布”具有两层含义,一是数据分布,即应用系统所操作的数据可以分散地存储在不同的网络节点上;二是操作分布,即应用系统的计算可由不同的网络节点完成。相应的,Java 实现两种层次上的分布。

(1) 数据分布支持:通过 Java 的 URL 类,Java 程序可以访问网络上的各类信息资源,其访问方式完全类似于本地文件系统。

(2) 操作分布支持:Java 通过嵌入到 WWW 页面中的小应用程序(Applet)将计算从服

务器分布至客户机。Applet 由 WWW 浏览器在客户端执行,从而避免了网络拥挤,提高了系统效率。

5. 健壮性

Java 非常重视及早地检查错误,Java 编译器可以查出许多其他语言在运行时才能发现的错误。Java 丢弃了其他语言中容易引起错误的某些数据类型,例如它不支持指针,从而避免了因重写内存而造成数据破坏的可能性。Java 的垃圾自动回收机制、异常处理机制和简单性等特点提高了程序的健壮性(又称鲁棒性)。

6. 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,因此 Java 强调安全性,例如确保无病毒、小应用程序运行的安全控制等。Java 的验证技术是以公钥(public-key)加密算法为基础,而且从环境变量、类加载器、文件系统、网络资源和名字空间等方面实施安全策略。Java 程序本身不操纵内存,这减少了内存出错的可能性,而让操作系统管理内存,这有效地避免了恶意代码的破坏和病毒的入侵,极大地提高了在网络环境下运行应用程序的安全性,同时也大大减轻了程序员的负担。

7. 解释执行

Java 解释器(interpreter)直接对 Java 字节码进行解释执行,字节码本身携带了许多编译时的信息,使得执行过程更加简单。

8. 体系结构中立(平台无关性)

网络上充满了各种不同类型的计算机和操作系统,为了使 Java 程序能够在网络的任何地方运行,Java 编译器生成了与体系结构无关(即体系结构中立(architecture neutral))的字节码文件(又称类文件),可以在所有提供 Java 虚拟机(Java Virtual Machine, JVM)的多种不同主机、不同处理器上运行。

9. 可移植性

Java 解释器生成与体系结构无关的字节码文件(*.class),只要是安装了 Java 虚拟机的计算机都可以执行这种字节码文件,Java 虚拟机把这种字节码文件翻译成所在系统的机器代码并执行。因此,Java 具有高度的可移植性,实现了其他程序设计语言所不具备的“一次编写,到处运行”(write once, run everywhere)的平台无关性。使用 Java 开发的系统的移植工作几乎为零,一般情况下只需要对配置文件、批处理文件进行相应修改即可实现平滑移植。

10. 多线程

线程是比进程更小的可并发执行的单位。在 C++ 语言中没有内置的多线程(multithread)机制,因此必须调用操作系统的多线程功能来进行多线程程序设计。而 Java 语言却提供了多线程支持。Java 语言在两个方面支持多线程,一方面是 Java 环境本身就是多线程的,若干个运行的系统线程负责必要的无用单元回收、系统维护等系统级操作;另一方面是 Java 语言内置的多线程机制可以大大简化多线程应用程序的开发。同时,Java 语言的线程还包括一组同步原语,这些原语负责对线程进行并发控制。利用 Java 语言的多线程 API,开发人员可以方便地写出支持多线程的应用程序,从而提高程序的执行效率。

11. 高性能

虽然 Java 是解释执行的,但是它仍然具有非常高的性能,在一些特定的 CPU 上,Java 字节码可以快速地转换成机器码来执行。而且 Java 字节码的格式设计就是针对机器码的

转换,在实际转换时相当简便,自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和即时编译技术的出现使得 Java 的执行速度有了更大的提高。Java 解释器在一秒钟内可调用 30 万个过程,编译目标代码的速度与 C/C++ 的性能没什么区别。

12. 动态性

Java 的设计使得它适用于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量且不会影响用户程序的执行,并且 Java 通过接口支持多重继承,使之比严格的类继承具有更灵活的方式和可扩展性。另外,类又是在运行时动态装载的,这就使得 Java 可以在分布环境中动态地维护程序及类库。

1.3 Java 开发运行环境的安装与配置

如果要实现“一次编写,到处运行”的目标,就必须提供相应的 Java 平台。学习 Java 必须从 Java SE 开始,因此,本书基于 Java SE 来学习 Java 程序设计。Java SE 包括两个部分,即 Java 开发工具集(Java SE Development Kit,JDK)和 Java 运行时环境(Java SE Runtime Environment,JRE)。JDK 是整个 Java 的核心,包括 Java 运行时环境、Java 工具和 Java 基础类库。用户登录到 Sun 公司的网站 <http://java.sun.com>(自动跳转到 <http://www.oracle.com/technetwork/java/index.html>)可以免费下载 Java SE Development Kit(jdk-7u11-windows-i586.exe),本书使用 jdk1.7.0_11 版本,大家可以根据自己所用计算机的操作系统免费下载不同的 JDK。

1.3.1 安装 JDK

在此以 Windows 平台上的 JDK 安装为例进行介绍,下载的安装程序是一个自解压的 jdk-7u11-windows-i586.exe 文件,执行该程序就可以在 Windows 上开始安装。其安装过程比较简单,当出现如图 1.1 所示的对话框时单击“更改”按钮可以更改安装目录。用户还可以根据实际需要选择 JDK 的安装部件,在默认情况下,可在“向导”指引下安装所有的 JDK 组件。

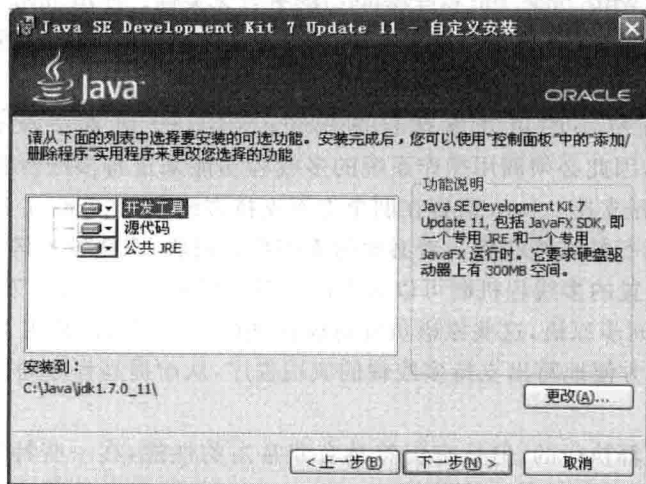


图 1.1 JDK 安装组件和目录选择对话框

下面是对安装组件的说明。

(1) 开发工具: Java 的开发工具包,用来编译、运行和打包 Java 程序。

(2) 源代码: 此组件提供 Java 类库的源码,展示了 Java 的一些核心技术。

(3) 公共 JRE: 公共的 Java 运行时环境组件,选择此组件,将在浏览器中安装最新的 Java 插件,使浏览器支持最新的 Java 类库。

安装成功后,在 JDK 文件夹下会出现 bin、db、include、jre、lib 等子文件夹(如图 1.2 所示),下面是对这些子文件夹的说明。

- bin: 包含编译器 javac、解释器 java、小应用程序浏览器 appletviewer 等可执行文件。
- db: 包含开源的 Apache Derby Java 数据库相关文件。如果没有选择安装 Java DB,则该文件夹为空。
- include: 存放本地文件(在 Windows 系统下是一些 C 语言头文件)。
- jre: 存放 Java 运行时环境(JRE)相关的文件。
- lib: 保存 Java 的类库文件。

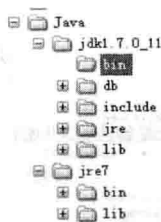


图 1.2 JDK 目录结构

说明: jre 是为开发好的 Java 程序提供执行平台。在图 1.2 中,jre 安装在 Java\jre7 文件夹下。jdk1.7.0_11 文件夹下的 jre 是 JDK 本身所附带的,主要是在开发 Java 程序时做测试之用。这二者的区别在于 JDK 本身所附带的 jre 比公用的 jre7 多了一个 Server 的 VM(Virtual Machine)执行选项。

1.3.2 安装 Java 帮助文档

JDK 帮助文档是重要的编程手册,可以帮助读者学习 JDK 的使用方法和完整地掌握 JDK 所有方面的内容。在 JDK 的安装程序中并不包含帮助文档,因此必须从网站上下载进行安装,JDK 帮助文档在 JDK 的下载页面也有下载链接。下载帮助文档的压缩文件 jdk-7u11-apidocs.zip,解压缩后通常安装在 JDK 所在目录的 docs 文件夹下,用浏览器打开 docs 文件夹下的 index.html 文件就可以阅读所有的帮助文档,如图 1.3 所示。

在 Java SE 中包含了 Java 语言在内的所有组成部分,如图 1.4 所示。JDK 是 JRE 的超集,包含了 JRE 的所有内容,此外还包含了开发 Java 小应用程序(Applets)和 Java 应用程序(Applications)的工具,例如编译器 javac 和解释器 java 等。另外,JRE 提供了运行 Java Applets 和 Applications 的类库以及 Java 虚拟机。

在 Java SE 中包含了两种虚拟机,分别是 Java HotSpot Client VM 和 Java HotSpot Server VM。

(1) Java HotSpot Client VM: 它是 Java 运行环境默认的 Java 虚拟机,最适合在客户端环境中运行 Java 程序,能够使程序的运行得到最佳的性能。

(2) Java HotSpot Server VM: 它是为在服务器端运行 Java 程序而设计的,可以使这种环境下的应用系统得到最快的运行速度。一般情况下,通过在命令行中使用-server 参数