

普通高等学校计算机教育规划教材

面向对象程序设计 ——Java语言

杜春涛 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校计算机教育规划教材

面向对象程序设计——Java 语言

杜春涛 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书参照全国计算机等级考试二级 Java 考试大纲进行编写。

全书共分 11 章,前 3 章主要涵盖了 Java 在结构化程序设计中的基本概念和应用,介绍了 Java 语言概述;Java 语言基础以及流程控制语句。第 4~10 章介绍了类与对象;消息、继承、封装与多态;常用系统类;异常处理;输入输出及文件操作;多线程;图形用户界面的设计与实现。通过对这部分的学习,读者能够真正理解什么是面向对象的程序设计以及怎样利用面向对象的方法在 Java 中进行程序设计,并能使读者学会利用 Java 语言本身提供的强大的系统类库去编写程序。第 11 章介绍了 Java 语言在网络编程和数据库编程方面的应用,为读者从学习程序设计走向真正的软件开发奠定基础。

本书适合作为高等院校的 Java 语言或面向对象程序设计的教材,也适合于对 Java 语言程序设计感兴趣的读者以及想参加全国计算机等级二级 Java 考试的读者使用。

图书在版编目(CIP)数据

面向对象程序设计:Java 语言 / 杜春涛编著. —北京:
中国铁道出版社, 2007. 6

普通高等学校计算机教育规划教材

ISBN 978-7-113-08024-2

I. 面… II. 杜… 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 (2007) 第 099360 号

书 名: 面向对象程序设计——Java 语言

作 者: 杜春涛

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 秦绪好

责任编辑: 杨 勇 鲍 闻

封面设计: 路 瑶

封面制作: 白 雪

印 刷: 三河市华晨印务有限公司

开 本: 787×1092 1/16 印张: 20 字数: 468 千

版 本: 2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

书 号: ISBN 978-7-113-08024-2/TP·2395

定 价: 28.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

面向对象技术被认为是程序设计方法学的一场革命,它已经逐步替代面向过程的程序设计技术,成为计算机应用开发领域的主流趋势。Java 语言作为网络时代最为流行的一种语言,它采用了纯面向对象的程序设计思想,是当今时代网络编程技术与面向对象程序设计技术最完美结合的一种计算机语言。2004 年全国计算机等级考试开设了二级 Java 的考试科目,使得这种语言在我国更加盛行,大多数高校也都相继开设了这门课程。本书参照全国计算机等级考试二级 Java 考试大纲,详细介绍了面向对象程序设计的思想及其在 Java 语言编程中的实现。

一、本书目标

目前,大多数大、中专院校都开设了 Java 语言这门课程,师生们都非常希望学完这门课程后能直接参加全国计算机等级考试二级 Java 的考试,目前市面上销售的针对计算机等级考试二级 Java 的辅导教材很多,但是选择一本合适的教材比较困难,不仅因为其教学对象都是针对已经学过 Java 语言的学生,难度较大,而且其内容编排也不符合正常教学要求;同时市场上销售的其他 Java 书籍又很少考虑到等级考试的问题,这些教材的内容与等级考试大纲要求相差较大。因此,一本既能接近计算机等级考试,又能符合正常的教学要求,做到教学、考试以及最后的毕业设计都能满足要求的实用 Java 书籍是许多读者非常需要的。

本书内容力求做到符合全国计算机等级考试二级 Java 考试大纲要求,在内容的编排上符合同学们正常学习的要求,同时又兼顾 Java 语言的最新发展以及学生以后做毕业设计的需要编写而成。力争做到简单易学,讲解细致,便于自学,通过现实生活中的例子以及程序示例将面向对象程序设计的思想及方法讲解清楚,使抽象问题具体化,复杂问题简单化。

二、本书特色

1. 本教材属于“北京市属市管高校人才强教计划资助项目”。

2. 在内容取舍方面,既靠近全国计算机等级考试二级 Java 考试大纲,又结合 Java 语言最新发展情况,同时还考虑到学生在学习过程中的实际情况,做到内容新颖,简单易学,符合学生正常学习及参加全国计算机等级考试的要求。

3. 在内容编排方面,遵循由浅入深、循序渐进的学习规律,尽量做到用到的内容是前面已经讲过的内容。在写作风格方面,立足于理论与实际结合,将复杂的面向对象的理论融入众多的实例之中,使读者学会能够用面向对象的思维方法分析问题并编写程序,从而解决实际问题。

4. 注意细节,使学生编写程序时少犯错误。例如,书中给出了许多注意的问题,便于学生自学过程中及时解决编程中遇到的问题。同时,本书一开始就把编译时经常出错的方面都列举出来,并教会学生如何调试程序,这对于刚刚开始学习编程的读者来说具有重要指导意义。

5. 在运行环境的方面,选择了 JCreator Pro 作为 Java 语言的运行环境,该运行环境和 VC++6.0 运行环境相近,用过 VC++6.0 运行环境的读者很容易接受;另外,该软件占据空间

很小，一般配置的计算机都能安装使用；该环境还具有代码提示等功能，极大地提高了编程效率，非常适合 Windows 程序设计要求，而且安装非常方便，配置也非常简单。

6. 附录中加入 Java 语言程序设计的规范，使学生一开始学习 Java 编程时就养成严格按照规范编写程序的习惯，这对程序设计的规范化具有重要意义。

7. 本书例题标题采用【例 X.Y】的形式，“X”代表第几章，“Y”代表第几个例题。由于 Java 语言要求文件名和主类名必须保持一致，因此对【例 X.Y】例题来说，它对应的文件名为“CX_Y.java”，该文件中的主类名为“CX_Y”。本书的习题中如果有源程序，源程序中主类名称采用“XX_Y_Z”的形式，第 1 个“X”表示“习题”的汉语拼音字头，第 2 个“X”表示习题几，“Y”表示第几道题，“Z”表示第几小题，依此类推。这种例题和习题名称设计的好处是：只要给出源程序，就可以知道该源程序出自哪个例题或哪个习题，使用非常方便。

8. 本书所有程序都经测试通过，以确保程序源代码的正确性。

三、主要内容

全书共分 11 章，前 3 章主要涵盖了 Java 在结构化程序设计中的基本概念和应用，讲述了 Java 语言概述、Java 语言基础以及流程控制语句等内容。该部分内容确保那些没有任何计算机语言基础的初学者能够学会程序设计语言所共有的基本概念和知识，同时也能帮助那些有一定计算机语言基础的读者加深对结构化程序设计的理解和认识，并学会 Java 语言运行环境的搭建。

第 4 章讲述了类与对象，第 5 章讲述了消息、继承、封装与多态。这两章全面讨论了面向对象程序设计的思想方法及其在 Java 中的实现。通过对这部分的学习，使读者能够真正明白什么是面向对象的程序设计以及怎样利用面向对象的方法在 Java 中进行程序设计。

第 6 章讲述了常用系统类，包括语言基础类、Applet 类、数组和字符串等内容。通过对这部分内容的学习，不仅使读者能够加深对前 5 章知识的理解和认识，并能使读者学会如何利用 Java 语言本身提供的强大的系统类库去编写程序，这对 Java 编程具有重要意义。

第 7 章讲述了异常处理，第 8 章讲述了输入输出及文件操作，第 9 章讲述了多线程。这 3 章是对 Java 语言程序设计特性的介绍。掌握 Java 语言所具有的这些特性能为读者设计出完整、实用的程序提供保证。

第 10 章讲述了图形用户界面的设计与实现，系统介绍了利用 AWT 和 Swing 组件进行图形用户界面的设计方法。通过对这部分内容的学习，读者能够设计出专业化的图形用户界面，并使该界面具有强大的事件处理功能，真正做到外形与功能的完美结合。

第 11 章 Java 的应用中讲述了 Java 语言在网络编程和数据库编程方面的应用。这既是 Java 的综合应用，也是 Java 实用性的体现。为读者从学习程序设计走向真正软件开发奠定了基础。

书中最后提供了 2 个附录，附录 A 提供了 Java 编程规范，使读者在一开始学习编程时就养成严格按照规范进行编程的习惯；附录 B 提供了全国计算机等级考试二级 Java 考试大纲，为那些准备参加等级考试的读者提供参考。

四、适用对象

本书定位于高等院校的学生以及相关培训学校的学员，也适合对 Java 语言程序设计感兴趣的读者以及想参加全国计算机等级考试的读者使用。

五、致谢

本书全稿由杜春涛编写。王恩波老师、古梅老师、王若宾老师、张师林老师、汲淑丽老师提出了许多宝贵的意见和建议。中国铁道出版社对本书的出版给予了很大的支持与帮助，提出了许多很好的修改意见和建议，在此对他们的大力支持与热情帮助表示诚挚的谢意。

尽管编者在 Java 语言及其他计算机语言课程教学方面具有多年的教学经验，但由于时间仓促和水平有限，书中难免存在不妥之处，欢迎读者多提宝贵意见。

为了方便教师教学，我们特向教师提供电子教案等相关教学素材，有需要的教师可与编者联系。编者电子邮箱：duct@ncut.edu.cn。

编 者

2007 年 6 月

第 1 章 Java 语言概述	1
1.1 Java 语言的产生及特点	1
1.1.1 Java 语言的产生	1
1.1.2 Java 语言的特点	1
1.2 Java 语言的实现机制	3
1.2.1 JVM (Java 虚拟机)	3
1.2.2 垃圾回收机制	3
1.2.3 代码的安全检验	4
1.3 Java 语言的体系结构	4
1.3.1 JDK 目录结构	4
1.3.2 Java 的 API 结构	4
1.3.3 Java 语言的程序结构	5
1.4 Java 的开发运行环境	5
1.5 Java 程序的运行步骤	5
1.5.1 Java Application 程序的建立及运行	5
1.5.2 Java Applet 程序的建立及运行	9
1.6 Java 程序的调试	12
习题 1	13
第 2 章 Java 语言基础	15
2.1 Java 符号集	15
2.1.1 标识符和关键字	15
2.1.2 运算符和分隔符	16
2.1.3 注释	16
2.2 简单数据类型	17
2.2.1 数据类型	17
2.2.2 常量	18
2.2.3 变量	19
2.2.4 数据类型之间的转换	24
2.2.5 基本数据类型类	25
2.3 运算符和表达式	26
2.3.1 算术运算符和表达式	26
2.3.2 关系运算符和表达式	28
2.3.3 逻辑运算符和表达式	29
2.3.4 位运算符和表达式	30
2.3.5 赋值运算符和表达式	31

2.3.6 条件运算符和表达式.....	31
2.3.7 运算符的优先级.....	32
2.4 常用数学方法.....	33
2.5 Java 标准输入、标准输出和标准错误.....	35
2.5.1 标准输入	35
2.5.2 标准输出	38
2.5.3 标准错误	38
习题 2	39
第 3 章 流程控制语句	42
3.1 顺序结构.....	42
3.2 分支语句.....	43
3.2.1 if 语句	43
3.2.2 switch 语句	48
3.3 循环语句.....	51
3.3.1 while 语句.....	51
3.3.2 do...while 语句.....	52
3.3.3 for 语句.....	53
3.3.4 循环语句的比较及其控制要点.....	54
3.3.5 循环嵌套	56
3.4 跳转语句.....	57
3.4.1 break 语句.....	57
3.4.2 break lab 语句.....	58
3.4.3 continue 语句.....	59
3.4.4 continue lab 语句.....	60
习题 3	60
第 4 章 类与对象	65
4.1 类	65
4.1.1 类的概念	65
4.1.2 类的定义	65
4.2 对象	67
4.2.1 创建对象	67
4.2.2 使用对象	67
4.2.3 对象的初始化和构造方法.....	68
4.3 数据成员.....	70
4.3.1 数据成员的声明.....	70
4.3.2 static 修饰的静态数据成员	70
4.3.3 final 修饰的最终数据成员	71
4.4 成员方法.....	72

4.4.1	成员方法的分类.....	72
4.4.2	声明成员方法	73
4.4.3	方法体中的局部变量.....	73
4.4.4	成员方法的返回值.....	75
4.4.5	形式参数与实际参数.....	75
4.4.6	成员方法的引用.....	76
4.4.7	static 修饰的静态方法	77
4.4.8	final 修饰的最终方法	78
4.4.9	native 修饰的本地方法.....	79
4.5	内部类.....	79
4.6	Java 系统类库.....	80
	习题 4	82
第 5 章	消息、继承、封装与多态	85
5.1	消息	85
5.1.1	消息的概念	85
5.1.2	公有消息和私有消息.....	85
5.2	继承	86
5.2.1	继承的概念	86
5.2.2	继承的特征	87
5.2.3	继承的实现	88
5.2.4	this 与 super	90
5.3	抽象类、接口和包.....	92
5.3.1	抽象类	93
5.3.2	接口	96
5.3.3	包	100
5.4	封装	103
5.4.1	公共访问控制符 public.....	104
5.4.2	缺省访问控制符 friendly.....	105
5.4.3	保护访问控制符 protected.....	105
5.4.4	私有访问控制符 private.....	106
5.4.5	封装与继承的关系.....	107
5.5	多态	107
5.5.1	多态的概念	107
5.5.2	方法重载	107
5.5.3	覆盖	108
5.5.4	构造方法的重载与继承.....	109
	习题 5	111

第 6 章 常用系统类	115
6.1 语言基础类	115
6.1.1 Object 类	115
6.1.2 System 类	115
6.2 Applet 类与 Applet 小程序	117
6.2.1 Applet 的基本工作原理	117
6.2.2 Applet 类	117
6.2.3 HTML 文件参数传递	121
6.3 数组	122
6.3.1 一维数组	122
6.3.2 多维数组	124
6.3.3 数组元素的复制	126
6.3.4 数组之间的赋值	126
6.3.5 数组作为方法参数	127
6.3.6 数组类 Arrays	128
6.4 字符串	129
6.4.1 String 类	130
6.4.2 StringBuffer 类	133
习题 6	134
第 7 章 异常处理	139
7.1 异常概述	139
7.2 异常类型	140
7.2.1 系统定义的运行异常	140
7.2.2 用户自定义的异常	141
7.3 抛出异常	142
7.3.1 系统自动抛出异常	142
7.3.2 利用 throw 语句抛出异常	143
7.3.3 利用 throws 语句抛出异常	144
7.4 捕捉并处理异常	144
7.4.1 利用 try...catch 语句捕捉并处理异常	145
7.4.2 利用 try...catch...finally 语句捕捉并处理异常	147
7.4.3 捕捉并处理 throw 语句抛出的异常	149
7.4.4 捕捉并处理 throws 语句抛出的异常	150
习题 7	151
第 8 章 输入/输出及文件操作	154
8.1 流类概述	154
8.1.1 流与计算机数据的 I/O 方向	154
8.1.2 Java 中包含的输入/输出流类	155

8.2	字节 I/O 流	160
8.2.1	字节输入流	160
8.2.2	字节输出流	162
8.2.3	内存的读写	164
8.3	字符 I/O 流	165
8.3.1	字符输入流	165
8.3.2	字符输出流	166
8.4	标准 I/O 流	167
8.4.1	标准输入	168
8.4.2	标准输出	168
8.4.3	标准错误	169
8.5	文件	169
8.5.1	File 类	170
8.5.2	文件 I/O 流	172
习题 8		179
第 9 章	多线程	183
9.1	Java 的多线程机制	183
9.1.1	线程的生命周期	183
9.1.2	多线程的实现方法	184
9.2	通过继承 Thread 类和实现 Runnable 接口实现多线程	185
9.2.1	Thread 类	185
9.2.2	通过继承 Thread 类实现多线程	186
9.2.3	通过实现 Runnable 接口实现多线程	187
9.3	线程等待	188
9.4	线程同步	190
9.5	线程的优先级	193
习题 9		194
第 10 章	图形用户界面的设计与实现	196
10.1	用 AWT 编写图形用户界面	196
10.1.1	java.awt 包	196
10.1.2	构件、容器和布局管理器	197
10.1.3	常用容器	198
10.1.4	布局管理器 (LayoutManager)	200
10.2	AWT 事件处理模型	206
10.2.1	事件类	208
10.2.2	事件监听器	209
10.2.3	AWT 事件及其相应的监听器接口	209
10.2.4	事件适配器	212

10.2.5 AWT 事件处理方式	214
10.3 AWT 构件库	218
10.3.1 标准构件的应用	218
10.3.2 基本构件与监听器的对应关系	251
10.4 AWT 用户自定义成分	251
10.4.1 与用户自定义成分有关的基础知识	252
10.4.2 Color 类	252
10.4.3 绘制文字	254
10.4.4 绘制图形	256
10.4.5 显示图像	260
10.4.6 实现动画	260
10.5 用 Swing 编写图形用户界面	262
10.5.1 Swing 的特性	262
10.5.2 Swing 构件及其事件处理	264
10.5.3 JOptionPane 类对话框	266
习题 10	270
第 11 章 Java 的应用	274
11.1 网络编程	274
11.1.1 URL 通信	274
11.1.2 Socket 通信	278
11.1.3 UDP 通信	285
11.2 数据库编程	288
11.2.1 数据库简介	288
11.2.2 JDBC 简介	289
11.2.3 常用 JDBC 类与接口	291
11.2.4 建立 ODBC 数据源	294
11.2.5 JDBC 编程	295
习题 11	301
附录 A Java 编程规范	302
附录 B 全国计算机等级考试二级 Java 考试大纲	306
参考文献	308

Java 语言是美国加州 Sun Microsystem 公司于 1995 年正式推出的纯面向对象 (Object-Oriented) 的程序设计语言。由于它很好地解决了网络编程语言中的诸多问题, 一经推出, 便受到计算机界的普遍欢迎和接受, 得到广泛应用和发展, 成为目前网络时代最为流行的程序设计语言。

1.1 Java 语言的产生及特点

1.1.1 Java 语言的产生

1991 年初, 美国加州的 Sun Microsystem 公司 (以下简称 Sun 公司) 成立了一个以 James Gosling 为首的名为 Green 的项目研发小组, 其目标是开发一个面向家用电器市场的软件产品, 用软件实现一个对家用电器进行集成控制的小型控制装置。他们首先注意到这个产品必须具有平台独立性, 即让该软件在任何类型的 CPU 上都能运行。为达此目的, Gosling 首先从改写 C++ 语言的编译器着手。但是, 他们很快便意识到这个产品还必须具有高度的简洁性和安全性, 而 C++ 在这方面显然无法胜任。因此, Gosling 决定自行开发一种新的语言, 并将该语言命名为 Oak (橡树)。

Oak 是 Green 项目小组开发的一个名为 “*7” (StarSeven) 产品中的一个组成部分。StarSeven 是一个集成了 Oak、GreenOS (一种操作系统)、用户接口模块和硬件模块 4 个部分的类似于 PDA (Personal Digital Assistant, 个人数字助理) 的设备。StarSeven 的第一个原型于 1992 年 8 月问世。尽管这个原型非常成功, 但在竞争激烈的家用电器市场上却败给了竞争对手。

“有心栽花花不开, 无意插柳柳成荫。” 有趣的是, 在这段时间里, WWW 的发展却如日中天。1993 年 7 月, 伊利诺斯大学的 NCSA 推出了一个在 Internet 上广为流行的 WWW 浏览器 Mosaic1.0 版。然而, 这时的 WWW 页面虽然内容丰富, 可以实现声、图、文并茂, 但它却是静态的, 若想增强 WWW 的动感, 需要通过一种机制来使它具有动态性。其解决方案显然是嵌入一种既安全可靠, 又非常简练的语言。Oak 完全满足这一要求。但是, 要将其推向市场, 为人们所广泛接受, 还必须采用一种合适的策略。1994 年, Sun 公司的创始人之一 Bill Joy 的介入, 使 Oak 成为 Java 而得以走红。

由于 Oak 这个名称与其他产品的名称雷同, 开发小组后来为这个新语言取一个新名称——Java (爪哇), 据说取这个名称的灵感来自于这样一个故事: 研发小组的成员经常在公司附近的一家咖啡厅喝咖啡, 而咖啡的原产地是爪哇 (即 Java)。

1.1.2 Java 语言的特点

Java 是一种新的基于面向对象技术的网络编程语言。Java 语言利用现有的许多软件新技术, 实现了许多原有编程语言所不能完成的功能, 弥补了一些编程语言的缺陷。

总体上说, Java 语言有以下特点。

1. 简单易学

Java 语言的基础是 Oak。简单性首先体现在系统本身的小巧, 解释程序、基本类库、基本标准类库和多线程支持一共只占有 215KB 存储空间, 同时对硬件要求不是很高, 以尽量精简的系统实现尽量强大的功能。其次, 在程序语言的语法方面, Java 浅显易懂, 不需要很好的编程基础, 只需要有一定的基本知识和概念就可以。

2. 面向对象技术

面向对象技术是一种新的编程思想, 它将人们描述客观世界的方法应用于编程语言中, 即将客观世界看成是由不同对象组成的, 具有相同或者相似特点的对象实体可以抽象成类 (class), 每个类中封装了自己特有的属性和方法。这种方法比原有的面向过程的语言有更好的可维护性、可重用性和可扩展性。Java 作为一种很成熟的面向对象编程语言, 把所有的 Java 应用和 Applet 程序都作为对象, 并且封装成不同的类, 同时 Java 语言对其简单类型数据也提供了封装对象。封装性实现了模块间的独立性和信息隐藏, 继承性实现了代码的可重用。根据这些基本概念, 用户可以根据自己的需要定义数据类型、定义属性、建立方法和建立自己的类。

面向对象技术的几个基本概念如下。

(1) 对象: Java 编程的最基本单元是对象。

(2) 类: 在 Java 语言中类是对象的抽象。

(3) 继承: 实现代码重用, 使用继承创建一个新类, 这个新类继承已有类的有关功能。这个新类被称为子类; 而被继承的类被称作超类或者父类。

(4) 封装: 对象把实例变量和相关方法封装进单个可表示的单元。

(5) 包: Java 描述相关类的集合。

3. 分布式计算

分布式计算是当前计算机网络研究领域中的一个很热门的方向。Java 的类库支持分布式计算所需要的基本协议 TCP/IP。应用程序可以很容易地通过 URL 地址访问网络上任何其他地方的对象。

4. 健壮性

健壮性也称为鲁棒性。Java 语言在编译和运行时进行大量的类型检查, 防止不匹配数据类型的出现, 对于非法的数据类型, Java 语言能在编译和解释时指出。Java 语言还禁止使用指针, 以及指针数组访问内存。更有特色的是, Java 语言有自动回收垃圾的功能, 这一点很好地解决了正确计算内存地址的问题, 同时也省去了在编程时管理内存分配的额外工作量。

5. 安全性

在提供高效的分布式网络计算的同时, Java 语言还提供了可靠的安全性保障, 是目前安全性最佳的语言。在 Java 虚拟机中运行 Java 程序时, 对程序的访问权限有严格的检查; 执行字节码之前进行必要检查, 不允许使用指针, 防止对内存的非法入侵。

6. 跨平台和可移植性

在 Web 应用中, 跨平台性是衡量一个软件的重要指标。Java 语言在这方面做得很成功, 因为 Java 解释器采用生成与系统结构无关的字节代码指令技术。也就是说, 在任何不同的操作系统下, 只要正确安装了 Java 运行系统, 就有了编写调试 Java 程序的平台。在分布式应

用中,Java 的这个特点使同一个 Java 程序能在不同的系统上运行,从而提高了软件生产效率。可移植性是跨平台特性的一个延伸,即具有了跨平台性,则保证了可移植性。Java 程序、Java 类库、Java 编译器及 Java 系统都具有可移植性。

7. 解释执行

不同于 C++ 语言的编译执行,Java 程序采用解释执行的方法。在系统编译运行 Java 程序时,Java 编译器将 Java 程序转化为字节码,字节本身带有编译时产生的信息,减轻了连接时系统的开销;在运行时,解释器将编译得到的字节码进行解释执行。

8. 高效性

虽然解释执行的效率会比较低,并且在系统中额外加载一个 JVM (Java 虚拟机),增加了系统的负荷,但是由于字节码很容易直接被转化为一些特定的 CPU 可识别的机器代码,因此执行效率非常高。另外,在编译时还可以通过优化字节码来生成更高效的代码,同时在 Java 语言中提供的多线程技术也为提高程序执行效率提供了强大的支持。

9. 多线程

在处理许多大规模并发、同步和共享事务的时候,多线程技术是一个很好的解决方案。Java 语言的多线程技术,为实现网络大量并发事件的处理提供了良好的编程模式。

10. 动态性

Web 应用对系统的可扩充性、可伸缩性要求很高,Java 语言为这种要求提供了实现方法。在 Java 语言中,类本身是可扩展和可伸缩的,可以增加或者删除新的方法或者属性等。同时 Java 语言还提供了很多编程接口,用来支持多重继承,提高了类继承的灵活性和扩展性。

11. Applet 的特点

嵌入式程序设计也是 Java 语言的一大特点。Applet 程序是一种特殊的 Java 应用程序,它通过嵌入网页发布到 Web 上。在 HTML 页面中嵌入 Applet 程序,使互联网上的信息系统具有更好的动态性和交互性。Applet 类作为 Java 语言的一个基本的类库,提供了很好的程序设计模式。编写此类应用程序时,通过继承 Applet 类,用户就只需要考虑相应的方法实现,而不需要过多地注意窗口如何创建、事件如何被处理等基本问题,从而大大降低了程序设计的复杂度。

1.2 Java 语言的实现机制

1.2.1 JVM (Java 虚拟机)

Java 语言是一种解释执行的语言,Java 编译系统先将源文件 (.java) 编译为字节码文件 (.class),然后再由 Java 虚拟机解释执行字节码文件。字节码不能直接在操作系统上运行,而只能通过虚拟机解释执行,因此虚拟机将字节码程序与操作系统及硬件分开,使得 Java 程序能在异构平台上执行。虚拟机执行过程有 3 个特点:多线程、动态链接和异常处理。

1.2.2 垃圾回收机制

Java 语言提供了一个系统级的线程,实现对内存使用的监控,在程序不需要时回收闲置不用的内存,从而减轻了程序员对内存管理的负担。

1.2.3 代码的安全检验

除了对一般的编译和解释方面的语法、规则和格式进行检查之外, Java 语言还提供了访问权限和类型转换合法性的检验, 操作数堆栈溢出检验及参数类型合法检验。另外, Java 语言还提供了资源域概念, 通过分配方案, 设定不同的资源域, 无需区分本地与远程。

1.3 Java 语言的体系结构

1.3.1 JDK 目录结构

现在常用的 JDK 是 j2sdk1.4.2 版, 以该版为例, 其目录结构描述如下。

JDK\bin\	包括编译器、解释器和一些基本工具。
JDK\demo\	包括一些简单的示例。
JDK\include\	是 Win32 子目录, 包括本地方法文件。
JDK\jre\	是 Java 程序运行环境的根目录。
JDK\jre\bin	包括平台运行时的工具和类库的可执行文件和动态链接库文件。
JDK\jre\lib	包括 Java 运行环境的代码库、默认安装文件等。
JDK\lib	包括类库文件。
JDK\src.zip	源代码压缩文件。

1.3.2 Java 的 API 结构

1. Java 核心包

(1) java.lang 包。封装了所有编写 Java 程序所需要的基本类, 包括 Object、Class 和 System 等。其中, Object 类是所有类的根类, 任何类都是继承该类而建立的; Class 类是由编译器动态生成对象的一个伴随每一个类的特殊类, 它的对象包含所对应的类的所有信息; System 类是一个 final 类, 主要提供标准 I/O 和对系统环境信息的访问等, 该类不能被实例化。

(2) java.awt 包。窗口工具包, 提供操作图形用户界面的基本功能。

(3) java.applet 包。封装 Applet 类的包, 提供 Applet 执行时需要的类。

(4) java.io 包。封装 I/O 操作的包。

(5) java.util 包。提供实用程序类及集合类等常用的工具类。

(6) 其他。java.net、java.math、java.rmi、java.sql、java.text、java.bean、java.security 等。

2. Java 扩展包 (javax)

主要包括 javax.naming、javax.swing、javax.rmi、javax.transaction、javax.sound 和 javax.accessibility 包等。

3. Java 语言文档

API 文档是很好的编程工具, 它提供了类库的详细说明。文档中类的描述一般由以下几个部分组成: 包名、类名、继承结构、类的实现接口、类的定义和说明、类的成员变量列表、构造方法列表、方法返回类型、方法列表、方法详细列表及其描述。

1.3.3 Java 语言的程序结构

Java 程序代码的结构主要包括如下几个部分。

首先在源文件最开始处是一个 `package` 语句，用于指定源文件所在的包，如果没有该语句则默认将源文件保存在当前文件夹下；紧接着在所有类定义之前的是 `import` 语句，用于引入标准的类，一般可有 0 个或者多个；接下来是 `public classDefinition` 语句，用于指定源文件的类名，也是源文件名，可以有 0 个或者 1 个；接着是若干 `classDefinition` 语句，用于一般类的定义，可以没有；最后是若干 `interfaceDefinition` 语句，用于定义接口，可以没有。

注意：一个 Java 源文件中最多只能有一个 `public` 类，并且源文件名必须和 `public` 类名相同。

1.4 Java 的开发运行环境

Java 开发运行环境是指 Java 的开发工具和相应的软/硬件环境。目前有许多用于编写 Java 程序的开发工具，如 Sun 公司的 Java 2 SDK 和 Java Workshop，Borland 公司的 Jbuilder，IBM 公司的 Visual Age for Java，Microsoft 公司的 VJ++，Symantec 公司的 Visual Cafe 以及 Xinox Software 公司的 JCreator Pro 等。本书以 Sun 公司的 Java 2 SDK 为后台工具，以 JCreator Pro 为前台工具作为 Java 的运行环境。

Sun 公司的 Java 2 SDK 有 Solaris SPARC/x86、Linux x86、Microsoft Windows 等版本，它们的建立过程略有不同。我们以 Microsoft Windows 版本的 Java 2 SDK 为蓝本，说明其建立过程。

安装 Java 2 SDK 的过程如下。

(1) 从 Sun 公司的网站 (<http://www.sun.com>) 上下载 Java 2 SDK 工具包。

(2) 从 Java 2 SDK 工具包中找到运行文件 `j2sdk-1_4_0_03-windows-i586.exe` 文件，完成 Java 2 SDK 的安装。

安装 JCreator Pro 的过程如下。

(1) 从 Xinox Software 公司的网站 (<http://www.jcreator.com>) 下载 JCreator Pro 工具包。

(2) 从 JCreator Pro 工具包中找到运行文件 `setup.exe` 文件，完成 JCreator Pro 的安装，然后将 Crack 文件夹中的 3 个文件复制到安装目录下。

1.5 Java 程序的运行步骤

Java 是半编译半解释型的语言。Java 的用户程序分为两类：Java Application 和 Java Applet。这两类程序在组成结构和执行机制上都有一定的差异。Java Application 是完整的程序，需要独立的 Java 解释器来解释运行。Java Applet 程序不能单独运行，必须嵌入到 HTML 语言编写的 Web 页面中，通过与 Java 兼容的浏览器来控制执行。

1.5.1 Java Application 程序的建立及运行

Java Application 程序建立过程如下。

(1) 打开 JCreator 软件，并选择菜单“File | New | Project”，如图 1-1 所示。