

Java SE 6.0

基础及应用案例开发

- 本书采用案例驱动方式编写，基础知识点讲解透彻，示例丰富。
- 代码以图解的方式进行说明，保证初学者从零基础开始学习。
- 项目实践 —— “图书管理系统”。

李刚生 王 焱 焦 玲◎编著



清华大学出版社

Java SE 6.0 基础及应用案例开发

李刚生 王 焱 焦 玲 编 著

清华大学出版社
北 京

内 容 简 介

本书共分为三篇：第一篇为基础篇，共4章，主要讲述Java的产生和发展、Java SE环境的搭建、Java SE基本语法、Java SE流程控制语句、Eclipse集成开发环境安装与配置、项目的创建、调试和运行等。第二篇为任务驱动篇，共5章，主要讲述Java SE类和接口、图形用户界面、Java SE输入输出流、Java SE网络应用以及Java SE数据库编程等。第三篇为综合案例编程篇，共1章，综合前面各章节所讲内容，分析图书管理系统开发的各模块，系统介绍了该系统的开发过程。

本书以实例、项目工程的实现为主线，以应用为目的，循序渐进地讲解Java的具体应用，使初学者易学易用。本书可作为高等院校计算机相关专业及计算机培训学校的教材，也可作为Java程序设计开发人员的参考用书。

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 SE 6.0 基础及应用案例开发/李刚生，王焱，焦玲编著. —北京：清华大学出版社，2009.8
ISBN 978-7-302-20744-3

I. J… II. ①李… ②王… ③焦… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 127742 号

责任编辑：张瑜 张丽娜

装帧设计：杨玉兰

责任校对：周剑云

责任印制：何 芊

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 刷 者：北京市世界知识印刷厂

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：185×260 印 张：32.25 字 数：778 千字

版 次：2009 年 8 月第 1 版 印 次：2009 年 8 月第 1 次印刷

印 数：1~4000

定 价：48.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103 产品编号：031162-01

前 言

当今是一个“信息化”、“网络化”的时代, Internet 将世界各地成千上万的计算机子网连成一个庞大的整体, 用户可以随时随地获取最新的资源。但是世界上的这些子网是由各种不同型号、不同规模、不同操作系统以及不同应用软件平台的计算机构成的, 这样就提出一个问题: 有没有一种语言, 使得程序员用这种语言编写的程序可以在各种不同的计算机上运行, 从而使 Internet 能够更好地发挥作用呢? Java 就是满足这种需求而得到广泛使用的语言, 也就是说, Internet 的发展使 Java 应运而生。

Java 语言由 Sun 公司于 1995 年 5 月 23 日正式推出, Java 作为面向对象的程序设计语言, 集安全性、简单性、易用性、跨平台性于一身, 并且特别适合于网络环境下的编程使用, 目前 Java 正在逐步成为 Internet 应用的主要开发语言, 它彻底改变了应用软件的开发模式, 在计算机领域里掀起了又一次技术革命高潮, 为迅速发展的信息世界增添了新的活力。尤其是 Java 2 平台的推出, 进一步促进和加快了 Java 技术的应用。

本书注重 Java 编程基础, 以图书管理系统案例为驱动, 围绕开发图书管理系统所需的知识点由浅深入地进行讲解, 通过本书的学习, 使读者能够掌握使用 Java 语言开发的一个完整程序。

本书分为三篇, 共十章, 具体内容如下。

第一篇: 基础篇, 共 4 章。

第 1 章 Java SE 概述, 主要讲述 Java 的产生和发展、Java SE 环境的搭建、Java SE 程序开发以及 Java SE API 的使用。

第 2 章 Java SE 6.0 基本语法, 主要讲述 Java 基本数据类型、常量与变量、Java 运算符、数据类型之间的转换以及运算符的优先级等。

第 3 章 Java SE 6.0 流程控制语句, 主要讲述 if 语句、if...else 语句、for 语句、while 语句、do...while 语句、break 语句、continue 语句和 return 语句等。

第 4 章 Eclipse 集成开发环境, 主要讲述 Eclipse 的特点、Eclipse 的安装与配置、Eclipse 的工作台、Java 项目创建、Java 项目调试和运行等。

第二篇: 任务驱动篇, 共 5 章。

第 5 章 Java SE 6.0 的类和接口, 主要讲述 Java 类定义、对象的创建、类的继承、方法的重载和覆盖以及 Java 接口等。

第 6 章 Java SE 6.0 的图形用户界面, 主要讲述简单控件、高级控件、布局管理等。

第 7 章 Java SE 6.0 的输入输出流, 主要讲述输入输出的处理方式、输入流、输出流、输入输出的异常处理以及文件(目录)的创建、移动、复制和删除等。

第 8 章 Java SE 6.0 的网络应用, 主要讲述 Socket 服务器端编程、Socket 客户端编程、Socket 通信的 I/O 流、Socket 异常处理以及多线程的创建、控制和同步等。

第 9 章 Java SE 6.0 的数据库编程, 主要讲述 JDBC 的结构、如何操作数据库以及 JDBC 的异常处理。



第三篇 综合案例编程篇，共 1 章。

第 10 章 图书管理系统，本章综合以上章节所讲内容，分析图书管理系统开发的各模块，系统介绍该系统的开发过程。

本书可以作为高等院校计算机相关专业及计算机培训学校的教材，也可以作为 Java 程序设计开发人员的参考用书。

本书由李刚生、王焱和焦玲编著，其中第 2、3、4、5、9、10 章由李刚生编写，第 1、6 章由焦玲编写，第 7、8 章由王焱编写，在编写的过程中青岛大学的孙更新老师提供了很多建议和意见，并给予大力支持和帮助，在此表示感谢。

限于编者水平有限，编写的过程中难免会有错误之处，希望广大读者不吝赐教。

编 者

目 录

第一篇 基础篇

第 1 章 Java SE 概述	1	2.4 运算符	30
1.1 Java SE 的产生与发展	2	2.4.1 算术运算符	30
1.1.1 Java 技术的产生	2	2.4.2 关系运算符	31
1.1.2 Java SE 的发展	2	2.4.3 赋值运算符	32
1.2 Java SE 的特点	3	2.4.4 逻辑运算符	33
1.3 搭建 Java SE 简易开发环境	5	2.4.5 条件运算符	34
1.3.1 下载 JDK 6.0	5	2.4.6 位运算符	35
1.3.2 安装 JDK 6.0	5	2.4.7 运算符的优先级	36
1.3.3 Windows 环境下 JDK 6.0 的 配置	7	2.4.8 语句和表达式	37
1.3.4 JDK 6.0 的新特性	9	2.5 数据类型间的转换	37
1.4 Java SE 程序开发过程	10	2.5.1 自动类型转换	38
1.4.1 Java SE 6.0 程序的基本结构 ..	10	2.5.2 强制类型转换	39
1.4.2 编写 Java SE 6.0 程序	11	2.6 数组与字符串	40
1.4.3 编译和运行 Java SE 6.0 程序	12	2.6.1 一维数组	40
1.4.4 使用 Java SE 6.0 API 文档	12	2.6.2 多维数组	45
1.5 思考与练习	14	2.6.3 字符串	49
第 2 章 Java SE 6.0 基本语法	15	2.7 基本数据类型与封装类	58
2.1 Java 符号	16	2.7.1 构造方法	58
2.1.1 标识符	16	2.7.2 常用方法	61
2.1.2 关键字	16	2.8 思考与练习	62
2.1.3 运算符和分隔符	17	第 3 章 Java SE 6.0 流程控制语句	63
2.1.4 注释	17	3.1 三种控制结构	64
2.2 数据类型	18	3.2 分支语句	64
2.2.1 基本类型	19	3.2.1 简单 if 语句	64
2.2.2 引用类型	19	3.2.2 if...else 语句	65
2.3 常量与变量	19	3.2.3 多层嵌套 if...else 语句	67
2.3.1 常量的概念与使用	19	3.2.4 判断中的“短路”现象	69
2.3.2 变量的声明与引用	21	3.2.5 使用 if 语句和三目运算符判定 是否是闰年	71
2.3.3 变量的作用域	26	3.2.6 switch 多分支语句	72
		3.3 循环语句	75

3.3.1 while 语句	75	4.2.3 Eclipse 的启动	92
3.3.2 do...while 语句	76	4.2.4 Eclipse 多国语言包的安装	93
3.3.3 for 语句	78	4.3 Eclipse 平台	96
3.3.4 多种循环嵌套	80	4.3.1 Eclipse 平台技术概述	96
3.3.5 使用循环语句计算水仙花数 ...	81	4.3.2 Eclipse 透视图	99
3.4 跳转语句	82	4.3.3 Eclipse 视图	101
3.4.1 break 语句	82	4.3.4 Eclipse 编辑器	101
3.4.2 continue 语句	84	4.4 创建并运行 Java 项目	103
3.4.3 return 语句	87	4.4.1 创建 Java 项目	104
3.4.4 goto 语句	87	4.4.2 创建 Java 类	105
3.5 思考与练习	88	4.4.3 编写 Java 源代码	106
第 4 章 Eclipse 集成开发环境	89	4.4.4 Java 项目的运行	112
4.1 Eclipse 的概念和特点	90	4.4.5 Java 项目的调试	113
4.1.1 Eclipse 的概念	90	4.5 Eclipse 中的项目管理	118
4.1.2 Eclipse 的主要特点	90	4.5.1 导入包	118
4.2 Eclipse 的安装与配置	91	4.5.2 导入 Java 类	118
4.2.1 Eclipse 的下载	91	4.5.3 导入 Java 项目	119
4.2.2 Eclipse 的安装	91	4.5.4 导出 Java 项目	120
		4.6 思考与练习	121

第二篇 任务驱动篇

第 5 章 Java SE 的类和接口	123	5.3 接口	160
5.1 类和对象	124	5.3.1 定义接口	161
5.1.1 Java 类定义	124	5.3.2 实现接口	161
5.1.2 类的成员变量	125	5.3.3 任务: 创建图书的接口	166
5.1.3 类的成员方法	128	5.4 项目实践	166
5.1.4 类的构造方法	135	5.4.1 定义图书类	166
5.1.5 对象的创建和使用	136	5.4.2 定义读者类	169
5.1.6 类的封装	139	5.5 思考与练习	171
5.1.7 包的创建和使用	143	第 6 章 Java SE 6.0 的图形	
5.1.8 任务: 创建用户类 User	146	用户界面	173
5.2 类的继承	147	6.1 简单控件和流式布局	174
5.2.1 类的继承概述	147	6.1.1 JLabel(标签)组件	174
5.2.2 方法的重载和覆盖	151	6.1.2 JTextField(文本框)组件	176
5.2.3 抽象类和最终类	156	6.1.3 JPasswordField	
5.2.4 任务: 创建管理员类		(密码框)组件	177
Admin	160	6.1.4 JButton(按钮)组件	179

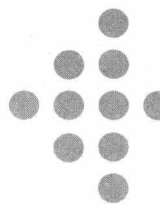


6.1.5 JPanel(面板)组件	180	6.6.1 创建管理用户界面	252
6.1.6 流式布局管理器	183	6.6.2 创建添加、修改、 删除用户界面	255
6.1.7 任务: 创建管理员 登录界面	185	6.7 思考与练习	266
6.2 选择框和网格布局	187	第 7 章 Java SE 6.0 的输入输出流	267
6.2.1 JComboBox(选择框)组件	187	7.1 Java SE 的输入输出类库	268
6.2.2 JList(列表框)组件	188	7.1.1 流概述	268
6.2.3 JRadioButton(单选按钮)和 JCheckBox(复选框)组件	189	7.1.2 Java SE 的输入输出流类	268
6.2.4 JOptionPane(弹出对话框) 组件	190	7.2 Java SE 的输入流	271
6.2.5 网格布局管理器	194	7.2.1 Java SE 的 I/O 处理方式	271
6.2.6 任务: 创建计算器界面	196	7.2.2 标准输入	274
6.3 Java SE 中事件处理模型和 BorderLayout 布局	201	7.2.3 文件的字节输入流	275
6.3.1 JTable(表格)组件	201	7.2.4 文件的字符输入流	281
6.3.2 JScrollPane(滚动面板)组件	202	7.2.5 I/O 异常处理	283
6.3.3 Vector(向量)组件	203	7.2.6 任务: 打开记事本文件	288
6.3.4 Java SE 中事件处理模型	204	7.3 Java SE 的输出流	290
6.3.5 BorderLayout 布局管理器	222	7.3.1 标准输出流	290
6.3.6 任务: 创建图书管理界面	223	7.3.2 文件的字节输出流	291
6.4 高级控件和卡式布局	228	7.3.3 文件的字符输出流	293
6.4.1 JMenu(菜单)组件、JMenuItem (菜单选项)组件和 JMenuBar (菜单栏)组件	228	7.3.4 任务: 保存记事本文件	296
6.4.2 JSplitPane(分割面板)组件	231	7.4 Java SE 的文件类	297
6.4.3 JTree(树状结构)组件	233	7.4.1 文件类	297
6.4.4 卡布局管理器	236	7.4.2 复制和删除文件	300
6.4.5 JFrame 窗体	239	7.4.3 复制和删除文件夹	303
6.4.6 任务: 创建系统主界面	240	7.4.4 任务: 管理记事本文件	305
6.5 文本域、对话框和 BoxLayout 布局	245	7.5 项目实践	310
6.5.1 JTextArea(文本域)组件	245	7.5.1 通过 NIO 操作文件	310
6.5.2 FileDialog(文件对话框) 组件	248	7.5.2 读写二机制文件	311
6.5.3 BoxLayout 布局	248	7.6 思考与练习	312
6.5.4 任务: 创建记事本主界面	250	第 8 章 Java SE 6.0 的网络应用	313
6.6 项目实践	252	8.1 Java SE Socket 服务器端编程	314
		8.1.1 Java SE 的 Socket 编程	314
		8.1.2 创建 Socket 服务器端	315
		8.1.3 Socket 异常处理	316
		8.1.4 任务: 创建图书管理系统 服务器端	317
		8.2 Java SE Socket 客户端编程	322

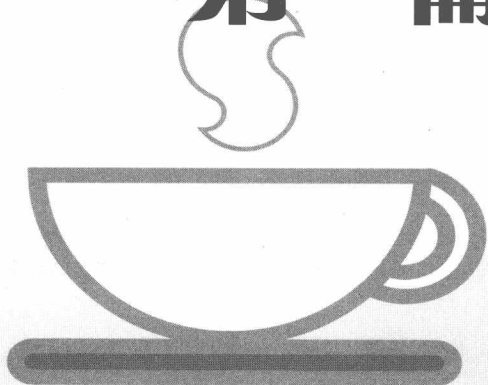
8.2.1 创建 Socket 客户端	322	9.1.1 Java SE 的 JDBC 编程	358
8.2.2 Socket 通信中的 I/O 流	322	9.1.2 建立数据库连接	359
8.2.3 任务: 创建图书管理系统 客户端	324	9.1.3 进行数据库查询	365
8.3 Java SE 的多线程应用	330	9.1.4 任务: 实现管理员登录的 数据库编程	368
8.3.1 Java SE 中的多线程	330	9.2 JDBC 高级编程	372
8.3.2 线程的创建	331	9.2.1 查询图书信息	372
8.3.3 线程的控制	332	9.2.2 添加图书信息	374
8.3.4 线程的同步	334	9.2.3 修改图书信息	375
8.3.5 任务: 实现多线程的 图书管理系统	339	9.2.4 删除图书信息	375
8.4 项目实践——实现 NIO 机制的 服务器	347	9.2.5 JDBC 中的事务处理	376
8.5 思考与练习	355	9.2.6 任务: 实现图书管理的 数据库编程	379
第 9 章 Java SE 6.0 数据库编程	357	9.3 项目实践——管理系统中的 读者信息	394
9.1 JDBC 编程基础	358	9.4 思考与练习	403

第三篇 综合案例编程篇

第 10 章 图书管理系统	405	10.5.2 管理员登录技术实现	414
10.1 系统分析	406	10.6 系统主界面模块	417
10.1.1 需求分析	406	10.6.1 系统主界面	418
10.1.2 可行性分析	406	10.6.2 系统主界面的技术实现	418
10.1.3 开发及运行环境分析	406	10.7 用户管理模块	424
10.2 系统设计	407	10.7.1 用户管理模块的客户端	424
10.2.1 系统流程	407	10.7.2 用户管理模块的服务器端	447
10.2.2 模块功能结构分析	408	10.8 图书管理模块	454
10.3 数据库设计	409	10.8.1 图书管理模块的客户端	454
10.3.1 E-R 图	409	10.8.2 图书管理模块的服务器端	477
10.3.2 数据表结构分析	410	10.9 图书借阅模块	484
10.4 连接数据库模块	411	10.9.1 图书借阅模块的客户端	484
10.5 管理员登录模块	413	10.9.2 图书借阅模块的服务器端	497
10.5.1 管理员登录界面	414	10.10 思考与练习	504



第一篇 基础篇



第 1 章

Java SE 概述

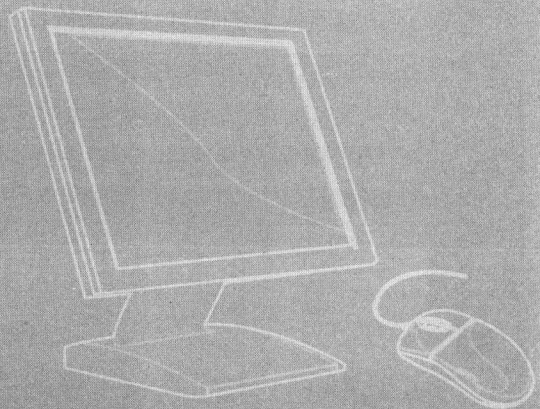
课前导读

Java 语言是由 Sun 公司于 1995 年 5 月 23 日正式推出的面向对象的程序设计语言，它集安全性、简单性、易用性、跨平台性于一身，并且特别适合于网络环境下的编程使用。近年来，随着网络的兴起和分布式计算环境框架技术的成熟，Java 正在逐步成为 Internet 应用的主要开发语言，它彻底改变了应用软件的开发模式，在计算机领域里掀起了又一次技术革命高潮，为迅速发展的信息世界增添了新的活力。尤其是 Java 2 平台的推出，进一步促进和加快了 Java 技术的应用。

本章重点讲解了 JDK(Java 开发工具包)的下载及安装、API 的下载及使用以及如何开发 Java 程序。

重点提示

- 搭建 Java SE 环境
- Java SE 程序开发
- Java SE API 的使用





1.1 Java SE 的产生与发展

1.1.1 Java 技术的产生

Internet 将世界各地成千上万的计算机子网连成一个庞大的整体，用户可以随时随地获取最新的资源。但是世界上的这些子网是由各种不同型号、不同规模、不同操作系统以及不同应用软件平台的计算机构成的，这样就提出一个问题：有没有一种语言，使得程序员用这种语言编写的程序可以在各种不同的计算机上运行，从而使 Internet 能够更好地发挥作用呢？

Java 就是满足这种需求而得到广泛使用的语言，也就是说，Internet 的发展使 Java 应运而生。

Java 产生的历史要追溯到 1991 年，Sun MicroSystem 公司的 Jame Gosling、Bill Joe 等人，为在电视、控制烤面包箱等家用消费类电子产品上进行交互式操作，他们深刻体会到消费类电子产品的特点，产品要求可靠性高、费用低、标准化、使用简单。为了使整个系统脱离具体平台的依赖，他们决定改写 C 编译器，但是他们很快发现，仅仅依靠 C 语言是难以满足要求的，为此，他们于 1994 年开发了一种叫 Oak 的语言，经过多次思考，最后用 Java 作为这种语言的名字。其实，Java 就是印度尼西亚盛产咖啡的爪哇岛的名字，将这种语言命名为 Java，是寄托了设计者“请你喝杯咖啡”的这种情意浓重的寓意，后来，还把这种情意形象化，在许多的 Java 文档中，人们常常可以看到一杯冒着热气的咖啡图标。

提示：Java 是爪哇岛出产的美味咖啡，用其命名表示给每个使用 Java 进行开发的程序员一杯美味的咖啡。

1.1.2 Java SE 的发展

根据不同的开发用途，Java 分为以下三个不同的开发平台版本。

(1) Java SE——Java Standard Edition, Java 的标准版，主要用于进行桌面以及普通网络、数据库等程序的开发。

(2) Java EE——Java Enterprise Edition, Java 的企业版，主要用于企业级开发，提供企业需要的各种技术支持，如 RMI、JNDI、EJB 等。

(3) Java ME——Java Micro Edition, Java 的微型版，主要用于移动、嵌入式开发，提供开发所需要的各种技术支持，如 CDC、CLDC 等。

提示：从 Java 1.2 版本开始，Java 也称为 Java 2，并且分为 J2SE、J2EE、J2ME 三个版本。Java 5.0 的版本原来为 1.5 版本，在 2004 年的 JavaOne 会议之后，版本升级为 5.0，2006 年底 Java 版本升级为 6.0。本书所有的案例都是基于 Java SE 6.0 平台进行讲解的。

Java SE 从诞生到现在主要经历了如下几个重要的过程。

(1) 1995 年, Sun 公司正式对外公布了 Java, 并于 1996 年发布了 Java 1.0, 但是人们很快就发现了其缺陷非常严重。虽然后来的 Java 1.1 的发布改善了其中的缺陷, 为 GUI(Graphical User Interface, 图形用户界面)的编程增加了新的事件处理模型, 但仍然具有极大的局限性。

(2) 1998 年, Java 的头号新闻便是发布了 Java 1.2。该版本更加接近了“一次开发, 随处运行”的承诺。后来的 Java 1.3 版本以及 Java 1.4 版本扩展了标准类库, 提高了系统性能, 同时为 Java 语言增加了很多新的特性。

(3) 2004 年, Java 5.0 的发布为 Java 添加了很多新的特性, 如泛型、安全枚举类型、程序注解等, 大大简化了 Java 程序的开发。

(4) 2006 年底, Java 6.0 的发布不但进一步提升了 Java 的性能, 还加入了对各种脚本语言嵌入式的开发。

1.2 Java SE 的特点

Java 是一种跨平台、适合于分布式计算环境的面向对象的编程语言。具体来说, 它具有简单性、面向对象、平台无关性、多线程、分布式、可靠、安全、可移植、高性能、动态性等特性。下面我们将介绍 Java 语言的几个重要特性。

1. 简单性

Java 语言最初是为了对家用消费类电子产品进行控制而设计的, 因此具有简单明了的特征。IT 业界公认 C 语言是一种灵活高效的语言, 而 Java 语言的开发者拥有丰富的 C 和 C++语言的使用经验, 因此 Java 语言采用了与 C 语言类似的语法结构, 简明、清晰。由于 Java 语言是一种面向对象的语言, 它提供最基本的方法来完成指定的任务, 因此编程人员只需理解一些基本概念, 就可以用它来编写适合于各种情况的应用程序。

2. 面向对象

面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象, 对象之间通过消息相互作用。另外, 现实世界中任何实体都可归属于某类事物, 任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心、以算法为驱动的话, 那么面向对象的编程语言则是以对象为中心、以消息为驱动。

面向对象程序设计是当今世界软件开发中最为常用的技术之一, Java 就是一种新型的面向对象的程序设计语言, 它集成了对象和接口的设计, 并提供了简单的类机制以及动态的接口模型。

所有面向对象的编程语言都支持封装、多态性和继承三个概念, Java 也不例外。

3. 平台无关性

平台无关性是指用 Java 编写的应用程序不用修改就可以在不同的软硬件平台上运行。



计算机系统千差万别,要让应用软件在网络上任何一种计算机系统中都能正常运行,就必须使软件具有平台无关性,也就是说,软件本身不受计算机硬件设备和操作系统的限制,可以在不同的计算机环境下良好地运行。

平台无关有源代码级和目标代码级两种,C 和 C++具有一定程度的源代码级平台无关性,表明用 C 或 C++编写的应用程序不用修改只需重新编译就可以在不同平台上运行。而 Java 主要靠 Java 虚拟机(JVM)在目标代码级实现平台无关性。JVM 是一种抽象机器,它附着在具体的操作系统之上,本身具有一套虚拟机指令,并有自己的栈、寄存器组等。JVM 通常是在软件上而不是在硬件上实现的。

JVM 是 Java 平台无关的基础,在 JVM 中,有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完程序后,通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台计算机只要配备了 Java 解释器,就可以运行这个程序,而不管这种字节代码是在何种平台上生成的。另外,Java 采用的是基于 IEEE 标准的数据类型,通过 JVM 保证数据类型的一致性,也确保了 Java 的平台无关性。

可以说,平台无关性是 Java 语言得以迅速发展并获得成功的主要原因,编程人员无需考虑对用户所使用的不同平台提供专门的特殊支持,真正实现“一次编写,到处运行”的跨平台性目标。

4. 多线程

线程是操作系统的一种新概念,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。进程有自己管理的一组系统资源和一个独立的存储空间,进程所涉及的数据、内存是独立的,因此,多进程系统就会有进程之间的通信机制,为实现进程通信就会让系统付出很多开销。而线程不仅可以共用一个内存区域,而且可以共享同一组系统资源,因此,线程之间通信可以节省许多开销。C 和 C++采用单线程体系结构,而 Java 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。若干个系统线程运行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,Java 语言内置多线程控制,可以大大简化多线程应用程序的开发。Java 提供了一个 Thread 类,由它负责启动运行线程与终止线程,并可检查线程状态。Java 的线程还包括一组同步原语,这些原语负责对线程实行并发控制。利用 Java 的多线程编程接口,开发人员可以方便地编写出支持多线程的应用程序,提高程序的执行效率。

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,对于客户机,Java 提供了一个叫作 URL 的对象,利用这个对象,用户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计算机的文件系统相同。对于服务器,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。

6. 可靠性和安全性

Java 最初的设计目的是应用于电子类消费产品, 因此要求较高的可靠性。Java 虽然源于 C++, 但它消除了许多 C++ 的不可靠因素, 可以防止许多编程错误的出现。首先, Java 是强类型的语言, 要求显式的方法声明, 这保证了编译器可以发现方法调用错误, 保证程序更加可靠; 其次, Java 不支持指针, 这样就杜绝了内存的非法访问; 第三, Java 的自动单元收集功能防止了内存丢失等动态内存分配导致的问题; 第四, Java 解释器运行时实施检查, 可以发现数组和字符串访问的越界; 最后, Java 提供了异常处理机制, 编程人员可以把一组错误代码放在程序中的一个地方, 这样可以简化错误处理任务便于恢复。

由于 Java 主要用于网络应用程序开发, 因此对安全性有较高的要求。如果没有安全保证, 那么用户从网络下载程序执行就非常危险。Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地计算机系统的威胁破坏。当 Java 字节码进入解释器时, 首先必须经过字节码校验器的检查, 然后, Java 解释器将决定程序中类的内存布局, 随后, 类装载器负责把来自网络的类装载到单独的内存区域, 避免应用程序之间相互干扰破坏, 最后, 客户端用户还可以限制从网络上装载的类只能访问某些文件系统。

1.3 搭建 Java SE 简易开发环境

学习 Java 程序设计语言之前, 我们必须首先掌握 Java 运行环境与开发工具的搭建与使用的方法。

JDK 是 Java Development Kit 的简称, 它是由 Sun 公司发展的 Java 开发环境, 是一般学习开发 Java 程序时最初使用的开发环境。其余的开发工具, 如 Borland JBuilder、Visual Cafe、Visual Age、JPadPro、JCreator 等, 均架构在 JDK 环境之上。

1.3.1 下载 JDK 6.0

Java 开发工具包在 Java 1.1 版本以前称作 JDK, Java 1.2 版本后更名为 J2SDK。不过大部分的人仍沿用 JDK 这个称呼。用户可以根据不同的操作系统平台, 到 Sun 公司的网站 (<http://java.sun.com>) 上下载相应的 JDK。在浏览器地址栏中输入 “<http://java.sun.com/javase/downloads/index.asp>”, 按下 Enter 键即可进入下载页面下载工具包。

1.3.2 安装 JDK 6.0

下载完成之后, 即可进行 JDK 6.0 的安装。下面介绍在 Windows 操作系统下安装 JDK 的过程。

(1) 进入存放 Java SE 6.0 工具包安装程序的目录, 双击 JDK-6u10-WINDOWS-I586.EXE 图标, 运行其安装程序。

(2) 安装的过程中会陆续出现如图 1-1~图 1-5 所示的界面, 图 1-1 显示了安装过程中的初始化工作。图 1-2 显示了接受许可协议, 继续下面的安装。图 1-3 显示了选择安装的组



件以及更改安装路径。图 1-4 显示了安装进度。图 1-5 显示了安装完成的界面。

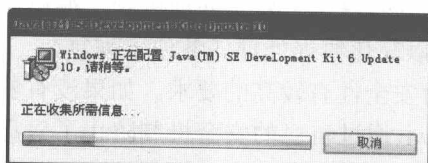


图 1-1 初始化界面

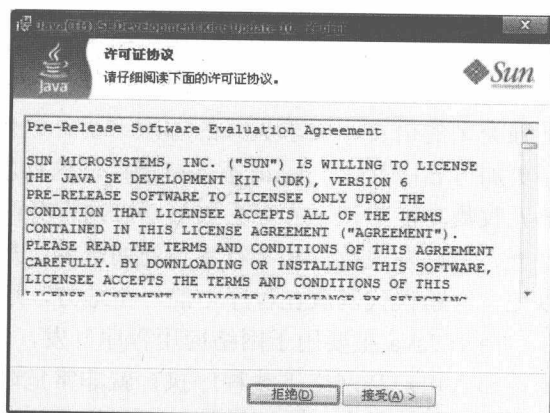


图 1-2 接受许可协议

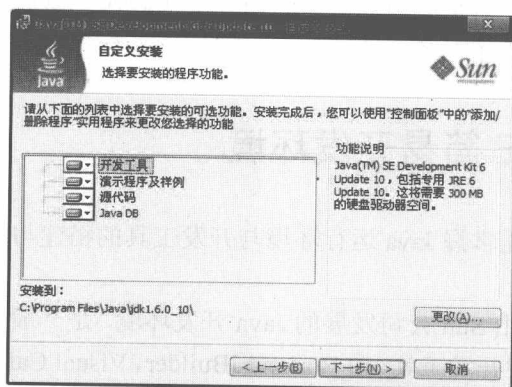


图 1-3 选择安装组件与目录

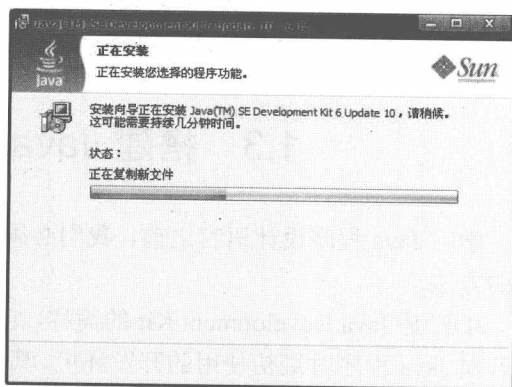


图 1-4 安装进度窗口

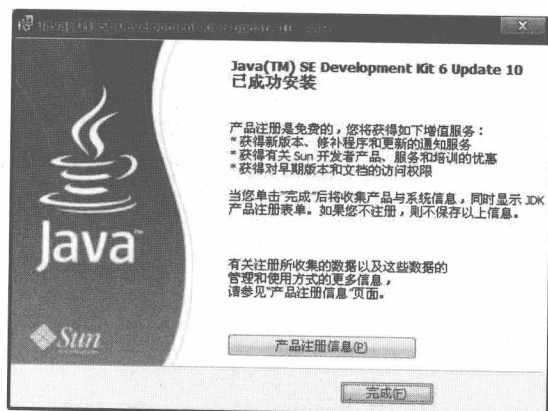


图 1-5 安装完成界面

安装完 JDK 后，在指定的安装位置将出现 jdk1.6.0_10 和 jre6 两个文件夹，如图 1-6 所示。在 jdk1.6.0_10 文件夹中包含了存放不同内容的各个子文件夹及文件，如表 1-1 所示。



图 1-6 安装后的 Java 目录

表 1-1 jdk1.6.0_10 文件夹中存放的主要内容

子文件夹或文件名称	描 述
bin	开发中需要用到的工具，如 javac.exe、java.exe 等
demo	开发工具包自带的演示程序
db	使用嵌入式数据库 Derby 开发所需要的资源以及一些案例
include	关于编译本地方法的 C++ 文件
jre	Java 运行时的环境
lib	开发工具包的类库文件
sample	开发工具包自带的示例程序
src.zip	类库 API 源代码文件

1.3.3 Windows 环境下 JDK 6.0 的配置

如果我们使用的是纯文本编辑器编写的源代码，那么在安装完 JDK 之后，还需要设置操作系统的 path 环境变量和 classpath 环境变量，将开发要使用的命令程序路径加入到 path 变量和 Classpath 变量中后，就可以编译运行 Java 程序了。

1. path 变量

path 变量用于指定操作系统到什么地方去查找 Java 工具。为此可将当前所用计算机的 Java 工具所在的路径(一般在 jdk 文件夹下的 bin 子文件夹中)赋给 path 环境变量。

例如，此处的设置为：path= C:\Program Files\Java\jdk1.6.0_10\bin。

2. classpath 变量

classpath 变量用于指定 Java 工具到什么地方去查找类(.class)文件。

例如，此处的设置为：classpath=.; C:\Program Files\Java\jdk1.6.0_10\lib\tools.jar。

提示： C:\Program Files\Java\jdk1.6.0_10 是 JDK 6.0 的安装目录。

那么如何在操作系统中配置 path 和 classpath 变量呢？下面以 Windows 2000、Windows XP 操作系统为例，介绍配置环境变量的步骤。

(1) 在桌面上右击【我的电脑】图标，从弹出的快捷菜单中选择【属性】命令，打开【系统属性】对话框，切换到【高级】选项卡，如图 1-7 所示，单击【环境变量】按钮，打



开【环境变量】对话框，如图 1-8 所示。在此对话框中，可以设置用户变量和系统变量。

(2) 单击【Administrator 的用户变量】选项组或【系统变量】选项组中的【新建】按钮分别进行 path 变量和 classpath 变量的设置，如图 1-9 所示。

经过上述两个步骤的操作，环境变量就配置完成了。

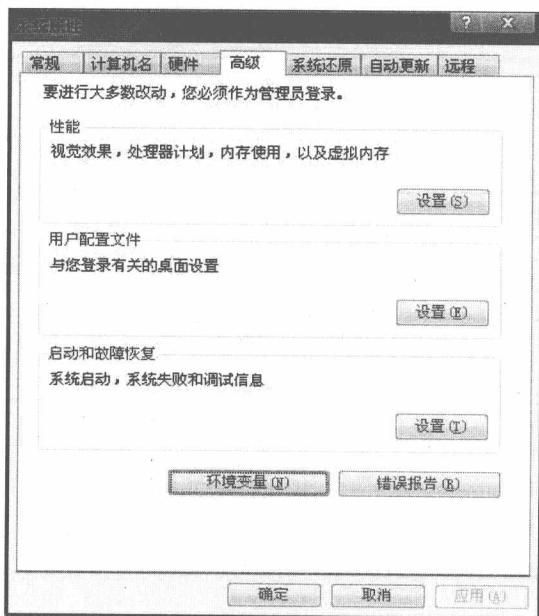


图 1-7 【系统属性】对话框

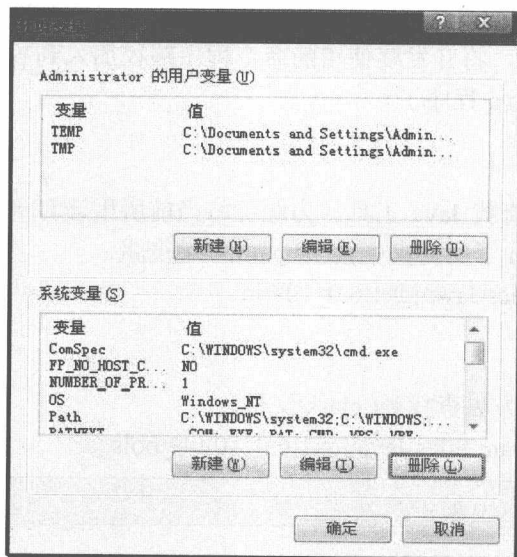


图 1-8 【环境变量】对话框

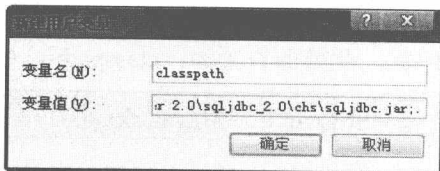
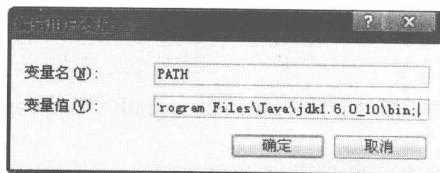


图 1-9 path 变量和 classpath 变量的设置