

JAVA

主 编◎邵斐

副主编◎董军 刘晶

程序设计



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

JAVA

主 编◎邵斐

副主编◎董军 刘晶

程序设计

东南大学出版社
SOUTHEAST UNIVERSITY PRESS

内 容 提 要

本书是配合 NIIT 合作项目的 Java 程序设计的中文版教材,是在多年教学基础上编写的。本书可分两部分:第一部分介绍了 Java 的基本概念和 Java 语言的基本特点,并从面向对象的编程方法入手,让读者能全面掌握 Java 的基础知识;第二部分围绕着 Java 的几个主要专题展开,如错误处理机制、多线程设计、图形用户界面开发、输入输出流应用和网络环境编程等。这些内容基本覆盖了 Java 的实用技术,是进一步使用 Java 进行技术开发的基础。

本书内容深入浅出,实例丰富,覆盖面广,图文并茂,独具特色。既有丰富的理论知识,也有大量的实战范例,更提供了精心设计的课后习题。本书适合作为高等学校的教材,也可作为专业人员的参考书。

图书在版编目(CIP)数据

JAVA 程序设计/邵斐主编. --南京:东南大学出版社, 2010. 4

ISBN 978-7-5641-2136-5

I. ①J… II. ①邵… III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 046651 号

JAVA 程序设计

出版发行 东南大学出版社(南京市四牌楼 2 号 邮编 210096)

电 话 (025) 83793191(发行); 57711295(传真)

出版人 江 汉

责任编辑 马子康

经 销 全国新华书店

印 刷 南京玉河印刷厂

版 次 2010 年 4 月第 1 版 2010 年 4 月第 1 次印刷

开 本 787mm×1092mm 1/16

印 张 14.5

字 数 344 千字

印 数 1—2 000 册

书 号 ISBN 978-7-5641-2136-5

定 价 28.00 元

(凡东大版图书因印装质量问题,请直接向读者服务部调换。电话:025-83792328)

前 言

自 20 世纪 90 年代以来,Java 已经逐渐发展成熟。其平台无关性、面向对象、联网功能和图形用户界面(GUI)以及线程支持使得 Java 成为许多应用程序理想的开发工具。本书将帮助您了解这些内容以及有关 Java 的其他重要方面,以便您可以充分地利用其功能。

跨平台、良好的移植性、嵌入式的语言等等这些特点一直伴随着 Java 的发展。随着用户需求的多样性和软件市场的繁荣,Java 与不同平台 API 函数的结合逐渐细分成了三大技术点:面向客户端主要用于桌面应用软件编程的 Java 标准版本(Java2 Standard Edition),简称 J2SE,它包含那些构成 Java 语言核心的类,比如:数据库连接、接口定义、输入/输出、网络编程等,它是学习所有 Java 技术的基础,从 JDK 5.0,不再叫 J2SE 了,改名为 Java SE 了,因为那个 2 已经失去了其应该有的意义;第二个是面向分布式网络程序开发的 Java 企业版本(Java2 Enterprise Edition),简称 J2EE,它是 Java 企业开发的技术规范,不仅仅是比标准版多了一些类,它还包括了许多组件,如 JSP、Servlet、JavaBean、EJB、JDBC、JavaMail 等;第三个是面向嵌入式系统开发的 Java 微缩版本(Java2 Micro Edition),简称 J2ME,它包含了 J2SE 的核心类,又新添加了一些专有类应用场合,如 CLDC 等方面的特性,API 的覆盖范围针对手机和 PDA 等移动的微小的嵌入式设备。本书的重点是 Java SE。

Java 结合了面向对象设计(OOD)和面向对象编程(OOP)的概念和技术,因此,从本书中您能够同时了解 Java 语言和 OOP 技术。OOP 是一种最有影响的现代编程规范;全面了解 OOP 是编程人员的当务之急。Java 为 OOP 提供了良好的支持。本书的第 2 章、第 3 章通过具有面向对象结构的实例详细演示了所包含的主题,目的是以一种简明且实际的方式提供对 Java 和 OOP 全面而详细的介绍。

用 Fortran、C 或 C++ 这类语言编写的程序执行单一控制流程,这些都是单线程程序。Java 允许多个线程同时存在于一个程序中,而且允许独立地执行



这些线程。多线程可以将任务组织起来,在一个程序中并发或并行执行。为了实现功能强大的并行性,需要克服众多与管理与协调同时发生的多项活动相关的难题。本书的第5章有较为详细的说明。

Java还具有对图形用户界面和事件驱动编程的良好支持。添加面向交互作用的窗口—鼠标的图形界面即可轻松地使用应用程序。Java基础类(JFC)可提供构件(窗口对象)、事件处理、图形绘制、图像显示和可动态选择的外观。本书的第6章重点讲解Java图形界面编程。

此外本书的第4章对Java的异常处理机制做了分析,第7章、第8章、第9章分别介绍了Java的输入输出特性、网络编程原理以及JDBC技术。

本书由金陵科技学院信息技术学院的张燕院长牵头,邵斐主编,陈圣国、刘晶、董军参加编著。东大出版社马子康等老师对本书的出版给予了充分重视和周到的安排,使本书能顺利出版。对所有鼓励、支持和帮助过我们的领导、组织、朋友,在此表示真挚的感谢。

由于作者水平有限,本书中难免有疏漏和不妥之处,竭诚欢迎读者多提宝贵意见。

编者

2010年3月



目 录

第 1 章 Java 简介	1
1.1 Java 的历史与语言结构	1
1.1.1 Java 的发展简史	1
1.1.2 Java 语言的特点	2
1.2 Java 的安装	3
1.2.1 Java 语言的开发工具	3
1.2.2 安装 Java 2 SDK	6
1.2.3 更新系统环境变量	7
1.3 Java 应用程序与 Java 小程序	9
实训一 安装与熟悉 Java 开发工具	12
第 2 章 Java 编程入门	14
2.1 Java 程序的结构	14
2.1.1 Java 程序布局	14
2.1.2 Java 源程序的组成	14
2.1.3 注解语句	16
2.1.4 打印语句	16
2.2 标识符	17
2.2.1 Java 字符集	17
2.2.2 标识符构成规则	19
2.2.3 关键字	19
2.3 数据类型	20
2.3.1 整型	20
2.3.2 实型	21
2.3.3 字符型	21
2.3.4 布尔型	22
2.4 运算符与表达式	22
2.4.1 概述	22
2.4.2 算术运算符	24
2.4.3 赋值运算符	25

2.4.4	关系运算符	26
2.4.5	逻辑运算符	26
2.4.6	位运算符	27
2.4.7	条件运算符	28
2.4.8	类型转换	28
2.5	流程控制语句	29
2.5.1	分支语句	29
2.5.2	循环语句	33
2.5.3	break 语句	35
2.5.4	continue 语句	35
实训二	类与对象	38
第3章	面向对象程序设计	40
3.1	面向对象程序设计概述	40
3.1.1	面向对象的概念	40
3.1.2	常用术语的含义	41
3.1.3	程序设计过程	41
3.1.4	面向对象程序设计的特点	45
3.2	Java 语言的面向对象程序设计	47
3.2.1	域	47
3.2.2	方法	48
3.2.3	构造方法	50
3.3	方法的使用和对象数组	50
3.3.1	调用方法	50
3.3.2	访问方法	52
3.3.3	方法重载	53
3.3.4	this	55
3.3.5	对象比较和对象数组	56
3.4	类的继承和多态	60
3.4.1	类继承的概念	60
3.4.2	类继承的实现	61
3.4.3	super	63
3.4.4	类的多态	64
3.5	包和访问控制符	66
3.5.1	包	66
3.5.2	常见的系统包	67
3.5.3	访问控制符	71
3.6	接口	74
3.6.1	抽象类和抽象方法	74



3.6.2 接口的概念	75
3.6.3 接口的实现	76
实训三 类与对象	78
第4章 常见错误和异常处理	80
4.1 常见错误	80
4.1.1 编译错误	80
4.1.2 运行错误	81
4.1.3 逻辑错误	82
4.2 异常处理	84
4.2.1 异常处理的概念	84
4.2.2 Exception 类	86
4.2.3 异常处理的过程	87
4.2.4 多异常的处理	91
4.2.5 自定义异常处理	93
实训四 异常处理	96
第5章 线程及其操作	99
5.1 线程的实现	99
5.1.1 线程的定义	99
5.1.2 创建线程	100
5.2 线程的状态及调度	106
5.2.1 线程的状态	106
5.2.2 线程的调度与优先级	107
5.2.3 控制线程	109
5.3 线程的同步	117
实训五 线程及其操作	123
第6章 图形用户界面的设计与实现	125
6.1 Java 图形用户界面概述	125
6.2 标准组件	125
6.2.1 组件分类	125
6.2.2 容器类组件	126
6.2.3 控制组件	127
6.2.4 Component 类	127
6.3 简单图形用户界面	127
6.3.1 Frame 与 Panel	127
6.3.2 字体设置	130
6.3.3 绘制简单图形	130



6.3.4	设置颜色	131
6.3.5	显示图像	131
6.4	Java 事件处理	132
6.4.1	Java 事件处理机制	132
6.4.2	按钮点击事件	136
6.4.3	捕获窗口事件	137
6.4.4	键盘事件	138
6.4.5	鼠标事件	138
6.4.6	焦点事件	139
6.4.7	事件适配器	139
6.5	布局管理	142
6.5.1	FlowLayout 类	143
6.5.2	BorderLayout 类	144
6.5.3	GridLayout 类	146
6.5.4	CardLayout 类	146
6.5.5	GridBagLayout 类	147
6.6	常用标准组件	148
6.6.1	按钮	148
6.6.2	标签	149
6.6.3	文本组件	149
6.6.4	复选框	149
6.6.5	单选按钮组	150
6.6.6	下拉列表	151
6.6.7	列表	151
6.6.8	滚动条	153
6.6.9	画布	153
6.6.10	对话框	154
6.6.11	文件对话框	154
6.7	Swing 组件	154
6.7.1	JApplet 类	156
6.7.2	JFrame 类	157
6.7.3	JLabel 类	160
6.7.4	JTextField 类	160
6.7.5	JPasswordField 类	161
6.7.6	JButton 类	161
6.7.7	JCheckBox 类	161
6.7.8	JRadioButton 类	162
6.7.9	JComboBox 类	162
实训六	图形用户界面的设计与实现	164



第7章 输入输出	165
7.1 流和文件	165
7.1.1 流	165
7.1.2 文件	166
7.2 基本输入/输出类.....	166
7.2.1 InputStream 类	167
7.2.2 OutputStream 类	169
7.2.3 PrintStream 类	170
7.2.4 其他常用流类	171
7.3 文件的输入/输出.....	171
7.3.1 FileInputStream 类	171
7.3.2 FileOutputStream 类	172
7.3.3 RandomAccessFile 类	172
7.3.4 File 类	175
实训七 流式输入输出.....	177
第8章 Java 网络编程	179
8.1 网络基础	179
8.1.1 TCP/IP	179
8.1.2 通信端口	179
8.1.3 URL	180
8.1.4 客户机/服务器模式	180
8.1.5 Java 网络编程	181
8.2 InetAddress 编程.....	182
8.3 URL 编程.....	184
8.3.1 创建 URL 对象	184
8.3.2 获取 URL 对象的属性.....	185
8.4 Socket 编程	188
8.4.1 Socket 与 ServerSocket 类	188
8.4.2 Socket 通信	189
8.5 数据报通信	193
8.5.1 UDP 套接字	193
8.5.2 实例:利用 DatagramSocket 查询端口占用情况	195
8.5.3 实例:利用数据报通信的 C/S 程序	197
8.5.4 组播套接字	198
实训八 Java 网络编程	200
第9章 JDBC 编程技术	204
9.1 JDBC 概述	204



9.1.1	JDBC 的概念	204
9.1.2	JDBC URL	204
9.1.3	JDBC-ODBC 桥	205
9.2	使用 JDBC 开发数据库的应用	205
9.2.1	应用举例	205
9.2.2	一般步骤	208
9.2.3	JDBC 相关类介绍	209
实训九	数据库应用程序开发	217
参考文献		221

第1章 Java 简介

1.1 Java 的历史与语言结构

1.1.1 Java 的发展简史

Java 的历史要追溯到 1991 年,由 Patrick Naughton 及其伙伴 James Gosling 带领的 Sun 公司的工程师小组想要设计一种小型的计算机语言,主要用于像有线电视转换盒这类的消费设备。由于这些消费设备的处理能力和内存都有限,所以该语言必须非常小且能够生成非常紧凑的代码;另外,由于不同的厂商会选择不同的中央处理器(CPU),因此这种语言的关键是不能与任何特定的体系结构捆绑在一起。这个项目被命名为“Green”。

代码短小、紧凑且与平台无关,这些要求促使开发团队联想起很早以前的一种模型,某些 Pascal 的实现曾经在早期的 PC 上尝试过这种模型,这是以 Pascal 的发明者 Niklaus Wirth 为先驱,率先设计出的一种为假想的机器生成中间代码的可移植语言(假想的机器亦称虚拟机——Java 虚拟机,JVM 的命名由此而来)。这种中间代码可以应用于所有已经正确安装解释器的机器上,因此 Green 项目工程师也使用了虚拟机,从而解决了课题中的主要问题。

不过,Sun 公司的人都有 UNIX 的应用背景,因此,所开发的语言以 C++ 为基础,而不是 Pascal;特别是这种语言是面向对象的,而不是面向过程的。就像 Gosling 在专访中谈到的:“毕竟,语言只是实现目标的工具,而不是目标本身。”Gosling 把这种语言称为“Oak”,Sun 公司的人后来发现 Oak 是一种已有的计算机语言的名字,于是将其改名为 Java,事实证明这是一个极好的选择。

当这一切在 Sun 公司中进行的时候,Internet 的万维网也日渐发展壮大,Web 的关键是把超文本页面转换到屏幕上的浏览器。1994 年大多数人都在使用 Mosaic,这是一个 1993 年出自伊利诺大学超级计算中心的非商业化的 Web 浏览器。它实际是由 Patrick Naughton 和 Jonathan Payne 开发的,并演变为 HotJava 浏览器。为了展现 Java 语言超强的能力,HotJava 浏览器采用 Java 编写;当然设计者也非常清楚 Applet 的威力,因此他们让 HotJava 浏览器具有执行网页中内嵌代码的能力。这一“技术印证”在 1995 年 5 月 23 日的 SunWorld 上得到展示,同时引发了人们延续至今的对 Java 的狂热追逐。

1996 年初,Sun 发布了 Java 的第 1 个版本,但人们很快地意识到 Java 1.0 不能用来进行真正的应用开发。的确,可以使用 Java 1.0 来实现在画布上随机跳动的 nervous 文本,但它却没有提供打印功能。坦率地说,Java 1.0 的确没有为其黄金时期的到来做好准备。后来的 Java 1.1 弥补了其中的大部分明显的缺陷,大大改进了反射能力,并为 GUI 编程增加



了新的事件处理模型,尽管它仍然具有很大的局限性。

1998年JavaOne会议的头号新闻是即将发布Java 1.2版。这个版本取代了早期玩具式的GUI,并且它的图形工具箱更加精细并具有可伸缩性,更加接近“一次编写,随处运行”的承诺。在1998年12月Java 1.2发布三天之后,Sun公司市场部将其名称改为更加吸引人的“Java 2标准版软件开发工具箱1.2版”。

除了“标准版”之外,Sun还推出了两个其他的版本:一个是用于手机等嵌入式设备的“微型版”;另一个是用于服务器端处理的“企业版”。本书主要讲述标准版。

标准版的1.3和1.4版本对最初的Java 2版本做出了某些改进,扩展了标准类库,提高了系统性能,还修正了一些bug。在此期间,Java applet采用低调姿态,并淡化了客户端的应用,使其成为服务器端应用的首选平台。

5.0版是自1.1版以来第一个对Java语言做出重大改进的版本(这一版本原来被命名为1.5版,在2004年的JavaOne会议之后,版本数字升至5.0)。经历多年的研究,这个版本添加了泛型类型(generic type)(类似于C++的模板),其挑战在于添加这一特性并没有对虚拟机做出任何修改。另外,还有几个来源于C#的很有用的语言特性:for each循环、自动打包和元数据。语言的修改总会引起兼容性的问题,然而,这几个如此诱人的新语言特性,使程序设计人员更加易于接受。

版本6(没有后缀.0)于2006年末发布。这个版本没有对语言方面再进行改进,但是,改进了其他性能,并增强了类库。

Java发展到今天已从编程语言成为全球第一大通用开发平台,Java技术已被计算机行业主要公司所采纳。1999年,Sun公司推出了以Java 2平台为核心的J2EE、J2SE和J2ME三大平台。随着三大平台的迅速推进,全球形成了一股巨大的Java应用浪潮。

1) Java 2 Platform, Micro Edition(J2ME)

Java 2平台微型版。Sun公司将J2ME定义为“一种以广泛的消费性产品为目标、高度优化的Java运行环境”。自1999年6月在JavaOne Developer Conference上声明之后,J2ME进入了小型设备开发的行列。通过Java的特性,遵循J2ME规范开发的Java程序可以运行在各种不同的小型设备上。

2) Java 2 Platform, Standard Edition(J2SE)

Java 2平台标准版,适用于桌面系统应用程序的开发。本书例程就是利用J2SE来开发的。

3) Java 2 Platform, Enterprise Edition(J2EE)

J2EE是一种利用Java 2平台来简化企业解决方案的开发、部署和管理等相关复杂问题的体系结构。J2EE技术的核心就是Java平台或Java 2平台的标准版,J2EE不仅巩固了标准版的许多优点,例如“一次编写、随处运行”的特性,方便存取数据库的JDBC API、CORBA技术,能够在Internet应用中保护数据的安全模式等,同时还提供了对EJB(Enterprise Java Beans)、Java Servlets API、JSP(Java Server Pages)以及XML技术的全面支持。

1.1.2 Java语言的特点

Java是一种面向对象、分布式、解释、健壮、安全、可移植、性能优异以及多线程的语言,下面简单介绍其中的几个优点。

1) Write Once, Run Anywhere

“一次编写,随处运行”,这是程序设计师们喜爱 Java 的原因之一,核心就是 JVM(Java 虚拟机)技术。编写好一个 Java 程序,首先要通过一段翻译程序编译成一种叫做字节码的中间代码;然后经 Java 平台的解释器,翻译成机器语言来执行,平台的核心叫做 JVM。Java 的编译过程与其他语言不同。例如,C++ 的编译与机器的硬件平台信息密不可分,编译程序通过查表将所有指令操作数和操作码等,转换成内存的偏移量,即程序运行时的内存分配方式,以保证程序运行。而 Java 却将指令转换成一种扩展名为 class 的文件,这种文件不包含硬件的信息。只要安装了 JVM,创立内存布局后,就可通过查表确定一条指令所在的地址,这就保证了 Java 的可移植性和安全性。

上述 Java 程序的编译和运行流程,如图 1.1 所示。

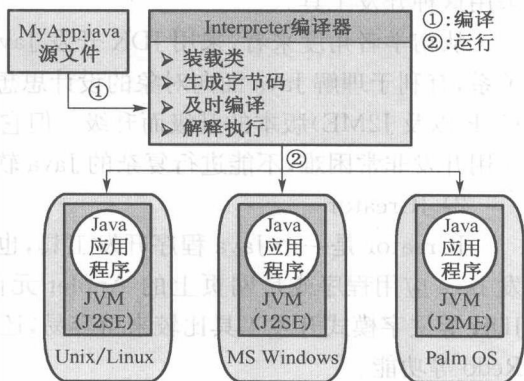


图 1.1 Java 的编译和运行流程

2) 简单

纯粹的面向对象,加上数量巨大的类所提供的方法(函数)库的支持,使得利用 Java 开发各种应用程序可以说是易如反掌。此外,因其面向对象的特性,使得在程序除错、修改、升级和增加新功能等方面的维护变得非常容易。

3) 网络功能

Java 可以说是借助因特网而重获新生的,自然可以编写网络功能的程序。不论是一般因特网/局域网的程序,如 Socket、Email、基于 Web 服务器的 Servlet、JSP 程序,还是分布式网络程序,如 CORBA、RMI 等的支持都是非常丰富的,使用起来很方便。

4) 资源回收处理

Garbage Collection 是由 JVM 对内存实行动态管理的。程序需要多少内存;哪些程序的内存已经不使用了,需要释放归还给系统,这些烦琐且危险的操作全部交由 JVM 去管理,让我们能够更专心地编写程序,而不需要担心内存的问题。内存的统一管理,对于跨平台也有相当大的帮助。

5) 异常处理

为了使 Java 程序更稳定、更安全,Java 引入了异常处理机制,能够在程序中产生异常情况的地方执行相对应的处理,不至于因突发或意外的错误造成执行中断或是死机。通过这种异常处理,不仅使得人们能够清晰地掌握整个程序执行的流程,也使得程序的设计更为严谨。

1.2 Java 的安装

1.2.1 Java 语言的开发工具

1) JDK(Java Development Kit)

SUN 的 Java 不仅提供了一个丰富的语言和运行环境,而且还提供了一个免费的 Java



开发工具集(JDK),开发人员和最终用户可以利用这个工具来开发 Java 程序。

JDK 简单易学,可以通过任何文本编辑器(如 Windows 记事本、UltrEdit、Editplus、FrontPage 以及 DreamWeaver 等)编写 Java 源文件,然后在 DOS 状况下通过 javac 命令将 Java 源程序编译成字节码,通过 Java 命令来执行编译后的 Java 文件。Java 初学者一般都采用这种开发工具。

从初学者角度来看,采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的关系,有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java (J2EE、J2SE 以及 J2ME)版本的升级而升级。但它的缺点也是非常明显的:从事大规模企业级 Java 应用开发非常困难,不能进行复杂的 Java 软件开发,也不利于团体协同开发。

2) JCreator

JCreator 是一个 Java 程序开发工具,也是一个 Java 集成开发环境(IDE)。无论是要开发 Java 应用程序或是网页上的 Applet 元件都难不倒它;在功能上与 Sun 公司所公布的 JDK 等文字模式开发工具比较来得容易,还允许使用者自定义操作窗口界面及无限 Undo/Redo 等功能。

JCreator 为用户提供了相当强大的功能,例如项目管理功能,项目模板功能,个性化设置语法高亮属性、行数、类浏览器、标签文档、多功能编译器功能,向导功能以及完全可自定义的用户界面。通过 JCreator,不用激活主文档就可直接编译或运行 Java 程序。

JCreator 能自动找到包含主函数的文件或包含 Applet 的 Html 文件,然后它会运行适当的工具。在 JCreator 中,可以通过一个批处理同时编译多个项目。JCreator 的设计接近 Windows 界面风格,使用户对它的界面比较熟悉。它最大特点是与我们机器中所装的 JDK 完美结合,这是其他任何一款 IDE 所不能比拟的,是一种初学者很容易上手的 Java 开发工具;其缺点是只能进行简单的程序开发,不能进行企业 J2EE 的开发应用。

3) Borland 的 JBuilder

JBuilder 可满足很多方面的应用,尤其是对于服务器方以及 EJB 开发者们。下面简单介绍一下 JBuilder 的特点:

(1) JBuilder 支持最新的 Java 技术,包括 Applets、JSP/Servlets、JavaBean 以及 EJB (Enterprise JavaBeans)的应用。

(2) JBuilder 可以自动生成基于后端数据库表的 EJB Java 类,同时简化了 EJB 的自动部署功能。此外它还支持 CORBA,相应的向导程序有助于用户全面地管理 IDL (Interface Definition Language, 分布应用程序所必须的接口定义语言)和控制远程对象。

(3) JBuilder 支持各种应用服务器。Jbuilder 与 Inprise Application Server 紧密集成,同时支持 WebLogic Server、EJB 1.1 和 EJB 2.0,可以快速开发 J2EE 的电子商务应用。

(4) JBuilder 能用 Servlet 和 JSP 开发和调试动态 Web 应用。

(5) 利用 JBuilder 可创建(没有专有代码和标记)纯 Java 2 应用。由于 JBuilder 是用纯 Java 语言编写的,其代码不含任何专属代码和标记,因此它支持最新的 Java 标准。

(6) JBuilder 拥有专业化的图形调试界面,支持远程调试和多线程调试;调试器支持各种 JDK 版本,包括 J2ME/J2SE/J2EE。

JBuilder 开发程序方便,是纯 Java 开发环境,适合企业的 J2EE 开发;但一开始往往难于把握整个程序各部分之间的关系,对机器的硬件要求较高,比较消耗内存,使运行速度显

得较慢。

4) Microsoft Visual J++

Visual J++是 Microsoft 公司推出的可视化的 Java 语言集成开发环境(IDE),为 Java 编程人员提供了一个新的开发环境,是一个相当出色的开发工具。无论集成性、编译速度、调试功能,还是易学易用性,都体现了 Microsoft 的一贯风格。Visual J++具有下面的特点:

(1) Visual J++把 Java 虚拟机(JVM)作为独立的操作系统组件放入 Windows,使之从浏览器中独立出来。

(2) Microsoft 的应用基本类库(Application Foundation Class Library, AFC)对 SUN 公司的 JDK 作了扩展,使之更加适合在 Windows 下使用。

(3) Visual J++的调试器支持动态调试,包括单步执行、设置断点、观察变量数值等。

(4) Visual J++提供了一些程序向导(Wizards)和生成器(Builders),它们可以帮助用户便捷快速地生成 Java 程序,在自己的工程中创建和修改文件。

(5) Visual J++界面友好,其代码编辑器具有智能感知、联机编译等功能,使程序编写十分方便。Visual J++中建立了 Java 的 WFC,这一新的应用程序框架能够直接访问 Windows 应用程序接口(API),使人能够用 Java 语言编写完全意义上的 Windows 应用程序。

(6) Visual J++中表单设计器的快速应用开发特性使用 WFC 创建基于表单的应用程序变得轻松、简单。通过 WFC 可以方便地使用 ActiveX 数据对象(ActiveX Data Objects, ADO)来检索数据和执行简单数据的绑定。而在表单设计器中使用 ActiveX 数据对象,可以快速地访问和显示数据。

Visual J++结合了微软的一贯编程风格,能很方便地进行 Java 的应用开发,但它的移植性较差,不是纯的 Java 开发环境。

5) NetBeans

NetBeans 是开放源码的 Java 集成开发环境(IDE),适用于各种客户机和 Web 应用。Sun Java Studio 是 Sun 公司最新发布的商用全功能 Java IDE,支持 Solaris、Linux 和 Windows 平台,适宜创建和部署 2 层 Java Web 应用和 n 层 J2EE 应用的企业开发人员使用。

NetBeans 是业界第一款支持创新型 Java 开发的开放源码 IDE。开发人员可以利用业界强大的开发工具来构建桌面、Web 或移动应用。通过 NetBeans 和开放的 API 模块化结构,第三方能够非常轻松地扩展或集成 NetBeans 平台。

NetBeans 3.5.1 版本与其他开发工具相比,最大区别在于不仅能够开发各种台式机上的应用,而且可以用来开发网络服务方面的应用、开发基于 J2ME 的移动设备上的应用等。

6) Eclipse

Eclipse 是一种可扩展的开放源码的 IDE。2001 年 11 月,IBM 公司捐出价值 4 000 万美元的源代码组建了 Eclipse 联盟,并由该联盟负责这种工具的后续开发。集成开发环境(IDE)经常将其应用范围限定在“开发、构建和调试”的周期之中,为了帮助集成开发环境(IDE)克服目前的局限性,业界厂商合作创建了 Eclipse 平台。Eclipse 允许在同一 IDE 中集成来自不同供应商的工具,并实现工具之间的互操作性,从而显著改变了项目工作流程,使开发者可以专注在实际的嵌入式目标上。

Eclipse 框架的这种灵活性来源于其扩展点——在 XML 中定义的已知接口,并充当插

件的耦合点。扩展点包括从用在常规表述过滤器中的简单字符串到一个 Java 类的描述。任何 Eclipse 插件定义的扩展点都能够被其他插件使用;反之,任何 Eclipse 插件也可以遵从其他插件定义的扩展点。除了解由扩展点定义的接口外,插件不知道它们通过扩展点提供的服务将如何被使用。

利用 Eclipse,我们可以将高级设计(如采用 UML)与低级开发工具(如应用调试器等)结合在一起。如果这些互相补充的独立工具采用 Eclipse 扩展点彼此连接,那么当用调试器逐一检查应用时,UML 对话框可以突出显示正在关注的器件。事实上,由于 Eclipse 并不了解开发语言,所以无论 Java 语言调试器、C/C++ 调试器还是汇编调试器都是有效的,并可以在相同的框架内同时关注不同的进程或节点。

Eclipse 的最大特点是它能接受由 Java 开发者自己编写的开放源代码插件,这类似于微软公司的 Visual Studio 和 Sun 微系统公司的 NetBeans 平台。Eclipse 为工具开发商提供了更好的灵活性,使他们能更好地控制自己的软件技术。

1.2.2 安装 Java 2 SDK

本书采用 Sun 公司发布的较新版本,也是近年来 Java 最重要的一个升级版本——Java 2 Platform Standard Edition 5.0(J2SE 5.0)作为 Java 开发平台。J2SE 5.0 具备哪些新的功能和特性请读者自行查阅相关资料。

J2SE 5.0 的下载地址为 <http://java.sun.com/j2se/1.5.0/download.jsp>。双击下载文件 jdk-1_5_0-windows-i586.exe 就开始了 J2SE 5.0 开发环境的安装过程,如图 1.2 所示。

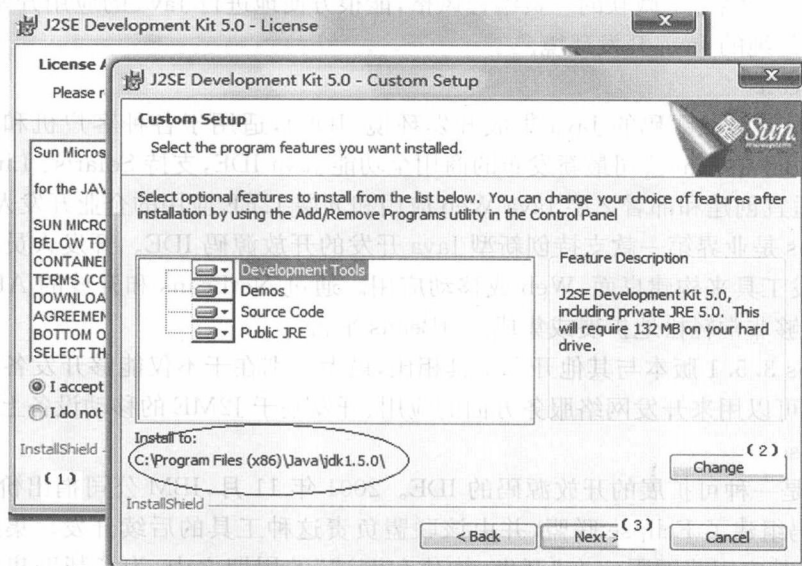


图 1.2 改变默认的安装路径

具体安装过程如下:

- (1) 安装程序经过自解压后,就会出现安装协议的对话框,选择【I accept the terms in