



高等学校计算机科学与技术教材

- ④ 原理与技术的完美结合
- ④ 教学与科研的最新成果
- ④ 语言精炼，实例丰富
- ④ 可操作性强，实用性突出

Java 程序设计与网络编程

□ 梁普选 主编



清华大学出版社

● 北京交通大学出版社

TP312/2578

2007

高等学校计算机科学与技术教材

Java 程序设计与网络编程

梁普选 主编

清华大学出版社

北京交通大学出版社

·北京·

内 容 简 介

本书全面系统地介绍了 Java 语言基础、Java 面向对象的编程模式、界面设计、多线程编程、JDBC 数据库访问、Applet 的应用,在此基础上重点讲述了基于 TCP/IP 的 Socket 网络接口编程、基于 HTTP 的 Web 服务器编程、基于服务器端的 Servlet 及 JSP 网站设计,最后讲述了基于远程对象的访问技术。本书语言流畅,实例丰富,内容易懂,可作为计算机类、软件技术类及网络工程等专业教材,也可供计算机网络开发研究人员参考学习。

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 13501256678 13801310933

图书在版编目(CIP)数据

Java 程序设计与网络编程/梁普选主编. —北京:清华大学出版社;北京交通大学出版社, 2007.11

(高等学校计算机科学与技术教材)

ISBN 978-7-81123-135-9

I. J… II. 梁… 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 151808 号

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京瑞达方舟印务有限公司

经 销:全国新华书店

开 本:185×260 印张:15.5 字数:400 千字

版 次:2007 年 10 月第 1 版 2007 年 10 月第 1 次印刷

书 号:ISBN 978-7-81123-135-9/TP·384

印 数:1~5 000 册 定价:26.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

本书编委会

主 编 梁普选

副主编 齐 林 苏连青 周长杰 李国刚
王彦新

主 审 齐 林

前 言

Java 是一种基于网络及 Internet 的面向对象的程序设计语言,它具有跨平台、高性能、安全性好的特点,是计算机网络及通信应用软件开发的首选工具,也是网络编程的标准开发工具,在 IT 界得到了广泛的应用。

高等院校计算机、网络工程及相关专业的 Java 程序设计课程既是一门概念复杂、知识面广的理论课,又是一门面向实践、需要动手的实践课。几乎所有的计算机初学者都想有朝一日在计算机上一展身手,让程序代码在自己的手下源源不断地流淌,成为驾驭计算机的高手。然而,学生在学完程序设计课程开始编写程序时,却往往感到无从下手,尽管自己高分通过考试,但并不会将所学知识用于编程中。一方面原因是学生没有掌握程序设计的一般规律,仅停留在课程内容的枝节上,所学到的知识支离破碎,不知道如何将所学知识应用到编程中;另一方面是教材内容组织不当,教学理念陈旧,与软件开发实际脱节,不能学以致用。

本教材共分 7 章,从教材内容的组织上分成了 3 个模块。第 1 个模块由前 2 章组成,主要介绍了 Java 语言基础及面向对象的编程模式。第 2 个模块是由第 3 章和第 4 章组成,主要讲述了 Java AWT 界面设计、多线程、数据库编程及网络编程基础知识。第 3 个模块由第 5、6、7 章组成,重点描述了基于 HTTP 协议的网络编程和基于服务器端的 JSP 编程,最后简要介绍了 Java 的远程对象访问技术。该教材内容基本涵盖了 Java 网络编程的各个应用领域。

参编本书的教师具有丰富的计算机教学及软件编程经验。第 1~4 章及第 6 章由梁普选编写,第 5 章由齐林编写,第 7 章由苏连青、周长杰、李国刚编写。在本书的编写过程中本着学以致用的基本原则,遵循程序设计一般性规律来组织教材内容。本书首先由 Java 语言基础知识入手,在介绍了面向对象的设计模式之后,着重讲述了 Java 的界面设计,数据库编程,网络编程及远程对象访问等实用知识。内容深入浅出,丰富的编程实例融会贯通在整个教材中,即使读者对计算机编程一无所知,也能通过本书获取 Java 编程的基本知识,了解并掌握程序设计的一般性规律。如果读者对计算机编程略有经验,已经在其他程序设计语言中掌握了基本技能,也会在本书中发现有用的编程知识。由于作者学识有限,书中难免存在不妥之处,敬请读者指正。

编 者

2007 年 7 月

目 录

第 1 章 Java 语言基础	1
1.1 Java 发展简史及特点	1
1.1.1 发展简史	1
1.1.2 Java 特点	1
1.2 Java 运行环境	4
1.2.1 Java 平台	4
1.2.2 安装 JDK	4
1.2.3 设置 Java 开发环境	6
1.2.4 Java 集成开发环境	8
1.3 常量与变量	12
1.3.1 标识符	12
1.3.2 关键字	12
1.3.3 常量	13
1.3.4 变量	13
1.4 数据类型	13
1.4.1 数据输入	14
1.4.2 byte 类型	15
1.4.3 short 类型	15
1.4.4 int / long 类型	16
1.4.5 char 类型	16
1.4.6 float / double 类型	17
1.4.7 boolean 类型	17
1.5 运算符与表达式	18
1.5.1 算术运算符	18
1.5.2 位运算符	19
1.5.3 关系运算符	19
1.5.4 运算符的优先级	20
1.6 数组和 String 类	20
1.6.1 定义数组	20
1.6.2 数组的使用	20
1.6.3 二维数组定义及初始化	21
1.6.4 String 类	23
1.7 Java 源程序结构	26

1.7.1 公有类·····	26
1.7.2 Java 包·····	28
习题·····	31
第2章 面向对象程序设计 ·····	32
2.1 面向对象·····	32
2.1.1 基本概念·····	32
2.1.2 数据抽象及数据封装·····	32
2.1.3 继承性与派生类·····	33
2.1.4 多态性·····	33
2.2 类与对象·····	34
2.2.1 类的定义·····	34
2.2.2 方法·····	35
2.2.3 创建对象实例·····	37
2.2.4 对象初始化·····	38
2.2.5 静态成员·····	41
2.2.6 对象数组及对象参数·····	43
2.2.7 包装类·····	45
2.3 接口·····	45
2.3.1 接口的作用·····	45
2.3.2 接口与内部类·····	47
2.3.3 内部类·····	49
2.4 继承与多态·····	50
2.4.1 单继承·····	51
2.4.2 用接口实现多继承·····	56
2.4.3 多态性·····	57
2.5 对象容器·····	61
2.5.1 迭代器·····	62
2.5.2 列表容器·····	64
2.5.3 集合容器·····	66
2.5.4 映射容器·····	67
2.6 异常处理·····	69
2.6.1 异常处理机制·····	69
2.6.2 异常的捕获与处理·····	69
2.6.3 异常的收尾·····	71
习题·····	72
第3章 Java 应用 ·····	74
3.1 界面设计·····	74
3.1.1 图形组件设计·····	74
3.1.2 常用组件·····	75

3.1.3	布局	80
3.1.4	事件处理	86
3.1.5	Swing 组件	94
3.2	I/O 流	98
3.2.1	基本的输入与输出	99
3.2.2	文件的输入与输出	100
3.2.3	字节流和字符流	102
3.2.4	文件的随机读写	102
3.2.5	对象流	103
3.3	多线程编程	108
3.3.1	创建与启动线程	108
3.3.2	线程优先级	111
3.4	Applet 及应用	118
3.4.1	Applet	118
3.4.2	Applet 的安全性	119
3.4.3	AWT 绘制	121
3.5	JDBC 数据库编程	124
3.5.1	装载驱动程序	124
3.5.2	建立连接	125
3.5.3	常用数据库 JDBC 连接	127
3.5.4	访问结果集数据	128
3.5.5	存取 BLOB 对象	129
	习题	134
第 4 章	Socket 编程	136
4.1	相关术语	136
4.2	URL 类	137
4.3	解析 Internet 地址	139
4.4	客户-服务器编程	139
4.4.1	客户-服务器通信	139
4.4.2	TCP 套接字编程	140
4.4.3	UDP 套接字编程	142
	习题	150
第 5 章	Web 应用	151
5.1	HTTP 协议	151
5.1.1	持久连接和非持久连接	151
5.1.2	HTTP 报文格式	152
5.1.3	常见的状态码	153
5.1.4	条件 GET 方法	155
5.2	Web 服务器	155

5.2.1	构造 Web 服务器	155
5.2.2	代理服务器	158
5.3	FTP 协议及工作原理	160
5.3.1	FTP 协议	160
5.3.2	FTP 服务器和客户端	160
5.3.3	FTP 地址格式	161
5.3.4	FTP 的传输模式	161
5.3.5	FTP 的工作方式	162
5.3.6	FTP 命令及编程思路	162
5.3.7	FTP 命令的处理	163
	习题	173
第 6 章	Servlet 与 JSP 编程	174
6.1	Servlet	174
6.1.1	Servlet 与 CGI	174
6.1.2	Servlet 结构	174
6.1.3	HTTP 与 Servlet	175
6.1.4	编写 HTTPServlet 程序	175
6.1.5	配置 Tomcat Web 服务器	180
6.2	JSP 编程	187
6.2.1	JSP 工作原理	187
6.2.2	JavaBean	188
6.2.3	JavaBean 与 JSP 调试	189
6.2.4	JSP 指令和脚本元素	190
	习题	211
第 7 章	RMI 编程	212
7.1	客户和服务端角色	212
7.2	远程方法调用	212
7.3	定位服务器对象	215
7.4	启动服务器	217
7.5	远程方法的参数传递	220
7.6	传递远程对象	229
7.7	激活服务器对象	231
	习题	238
	参考文献	240

第 1 章 Java 语言基础

1.1 Java 发展简史及特点

近年来,随着 Internet 网络以及信息技术的发展,人们对网络功能的要求越来越高,同时对网络应用程序的要求也越来越高,特别是需要一些能在不同的操作系统平台上运行的应用程序。Java 语言的出现恰恰适应了这种跨平台应用程序的需求,从而得到了广泛的应用。Java 语言不仅仅是一种程序设计语言,更是一个网络操作系统。通过对 Java 语言的使用,原本静止的 Web 页面可以变成生动诱人的动画。不仅如此,Java 语言的出现,也使信息技术的应用和影响扩大到空前的范围,它从根本上改变了网络应用程序的开发和使用方式。本章主要介绍 Java 语言的特点和编程方法。

1.1.1 发展简史

20 世纪 90 年代初,Sun 公司出于消费类家电项目的需求推出了一种面向对象的“可移植”语言。1995 年用 Java 语言开发出了一个浏览器 HotJava,并获得了极大的成功。由此引发了世界范围内的 Java 热,至今未衰。1998 年 Java 1.2 版本的出现,使它成为现代开发工具中的利器。Java 现在的体系结构可以分为三个模块:J2SE(Java2 Platform Standard Edition)、J2EE(Java2 Platform Enterprise Edition)和 J2ME(Java2 Platform MicroEdition)。

1.1.2 Java 特点

(1) 简单数据类型和对象

设计一种面向对象语言所面临的主要问题是平衡对象和简单数据类型。从纯理论的观点来看,每种数据类型都是一种对象,并都应该从一个共同的父类派生而来。这就使得所有的数据类型以相同的模式运作,并共享一个公共的基类属性集合。将简单数据类型作为对象处理会导致总体性能的下降。由于简单数据类型常用于控制语句中,所以这些额外开销带来广泛的负面影响。诀窍就是如何在“一切都是对象”的理想和“性能平衡”的现实之间找到平衡点。

Java 非常巧妙地解决了对象与简单数据类型之间的问题。首先,Java 定义了 8 种简单数据类型:byte,short,int,long,char,float,double 和 boolean。这些类型可直接转换为二进制代码。因此 CPU 可直接处理简单数据类型的变量,而无须任何额外的开销。在 Java 中,处理简单数据类型和其他语言一样高效。除了这些简单数据类型外,Java 其他数据类型都是从一个共同超类 Object 派生而来。因此,所有这些数据类型都拥有从父类继承而来的方法和属性集。例如,所有对象都有 toString()方法,因为 toString()是父类 Object 中定义的方法。

由于简单数据类型不是对象,因此 Java 以略有不同的方式处理对象和非对象。在 Java

中,所有对象都通过引用来访问。只有简单数据类型才可直接访问。因此,Java 程序绝对不会直接操作一个对象。这种策略可带来很多的好处,最直接的就是能够高效地实现垃圾回收。因为所有对象都通过引用访问,当一个对象没有被直接引用时,它将被回收。另一个好处就是每个 Object 类型的引用都可引用系统的任何对象。

当然,通过引用访问对象将产生额外的开销。因为一个引用实际上就是一个地址(指针)。于是对每个对象的访问不是直接进行的,而是通过地址间接完成的。尽管现代 CPU 可高效处理间接访问,但是间接访问总是不如直接处理数据本身快。尽管简单数据类型的处理是高效的,但是有时需要使用与某个简单类型等价的对象。例如,在运行时创建了一个整型链表,当不再使用时将其回收。为了处理此类问题,Java 为简单类型定义了包装器(Wrapper)。包装器使得简单类型在必要时可参与到对象层次的操作中来。Java 解决了对象和简单数据类型之间的平衡问题,可编写出高效程序,同时又完美地解决了允许实现对象模型,而不用担心简单数据类型的性能会产生负面影响。

(2) 通过垃圾回收实现内存管理

垃圾回收作为一种内存管理技术已存在了很长时间,而 Java 使它焕发出崭新的活力。在 C++ 语言中,内存必须自己管理,程序员必须显式地释放不再使用的对象。这是问题产生的根源,因为忘记释放不再使用的资源,或者释放了正在使用的资源是很常见的事。Java 则代替程序员完成这些工作,防止了此类问题的发生。在 Java 中,所有的对象都是通过引用访问的。这样,当垃圾回收器发现一个没有被引用的对象时,就知道该对象不再被使用了,可以进行回收。如果 Java 允许对象的直接访问,则这种有效的垃圾回收方法将无法实现。Java 垃圾回收策略反映了 Java 的理念。

(3) 完美的简单多线程模型

Java 的设计者所提供的编程特性,包括对多线程多任务支持。多任务具有两种类型:基于进程的多任务和基于线程的多任务。在基于进程的多任务中,最小的可调度单元是进程。进程实际上就是正在执行的一个程序。因此,基于进程的多任务允许计算机同时运行两个或多个程序。在基于线程的多任务中,最小的调度单元是线程。线程定义了一个程序内的某条执行路径。因此一个进程可包含两个或多个执行线程,而多线程程序可有两个或多个可并发执行的部分。

尽管基于进程的多任务通常是操作系统提供的功能,但基于线程的多任务却能极大地受益于程序设计语言级别的支持,例如,C++ 没有对多线程编程提供内置支持,于是就必须依赖于操作系统来处理多线程任务。这就意味着,创建、启动、同步和结束线程必须通过对操作系统的多次调用来实现,因此 C++ 中的多线程代码是不可移植的。这使得 C++ 中的多线程没有得以广泛的应用。

由于 Java 内置了对多线程的支持,在 C++ 语言中必须由手工完成的工作,现在都可由 Java 自动处理。Java 多线程模型中最有特色的部分之一就是其实现同步的方式,同步建立在两种创新的特性之上。首先,Java 中的所有对象都有互斥锁的内置侦听器,在给定的时刻,任何侦听器只能由一个线程拥有。通过使用 synchronized 关键字对方法进行修饰,可启用锁定特性。在调用同步方法时,该对象被锁定,而试图访问该对象的其他线程只能等待。其次,Java 对同步的支持也可从所有类的共同超类 Object 中体现。Object 声明了三个同步方法:wait(),notify()和 notifyAll(),这些方法支持线程之间通信。因此,所有的对象都对线程间通

信有内置的支持。通过与同步方法结合使用,这些方法可为线程的互操作提供高级的控制。

(4) 完全集成的异常机制

异常在 Java 的最初设计中就已经引入,异常机制在 Java 语言中是完全集成的,它是 Java 的基本特征之一。Java 异常机制的关键在于,异常是必须使用的,而不是可选的。通过异常处理是 Java 语言的规则,这与 C++ 是有区别的。例如,C++ 同样支持异常机制,但是这种机制并未集成到整个编程环境中。考虑打开并读取一个文件的操作,在 Java 中,如果某个操作发生错误,将抛出一个异常。而在 C++ 中,打开并读取文件的方法会返回一个专用的错误代码,报告操作中发生的错误。因为 C++ 库仍然依赖于错误返回代码,而不是异常,所以从本质上来说,C++ 不支持异常,程序必须不断地进行人工检查,以避免可能出现的错误。而在 Java 中,程序员只须使用一个 try...catch 代码块,就可自动捕捉到任何错误。

(5) 对多态支持的改进

多态性是面向对象编程的基本特征之一,它允许多个方法使用同一个接口。Java 从多个方面支持多态性,其中有两个方面最为突出。第 1 个是每个方法(标志为 final 的方法除外)都可被子类重写;第 2 个是设立了 interface 关键字。由于超类中的方法可在子类中重写,因此所创建类的层次结构非常清晰。超类的引用可引用它的任何一个子类,而且通过超类的引用调用某个子类对象的一个方法时,会自动调用由该子类重写后的版本,因此,可用超类来定义对象的形式并提供对象的默认实现,而子类根据这种默认实现修改,以更好地适应具体情况的要求。因此在超类中定义的一个接口可作为多个不同实现的基础。

当然,Java 进一步采取了“一个接口,多个方法”的概念。它定义了 interface 关键字,可实现将类的方法和类的实现代码完全分离。尽管接口是抽象的,但是仍然可声明接口的引用。这个概念非常重要,因为它可改进多态性的应用。只要某个类实现了一个接口,并且该接口提供了某种支持,那么任何需要这种功能的代码都可使用这个类的对象。

(6) 通过字节码保证可移植性和安全性

字节码是 Java 语言的一个重要的组成部分,它对于程序员而言是透明的。Java 编译器输出的不是可被 CPU 直接执行的机器指令,而是一种经过高度优化的可移植的指令集合,这种指令集合称为字节码,它只能由 Java 虚拟机(Java Virtual Machine, JVM)执行。最初,Java 虚拟机只是一个简单的字节码解释器,现在 Java 虚拟机也将字节码的编译技术应用到可执行代码中。无论字节码的执行采用何种方式,它的优势对于 Java 的成功都是至关重要的。

字节码的第一个优势是一致性。无论计算机使用何种类型的 CPU(或操作系统),只要具有 Java 虚拟机,由 Java 程序编译的字节码就可在其中运行。换言之,只要为某个特定的环境实现了 Java 虚拟机,则每个 Java 程序都可在该环境下运行,没有必要为每个不同的环境创建单独的执行代码,因为同一种字节码可在所有环境中运行。因此通过使用字节码,Java 为程序员提供了“一次编写,到处运行”的能力。

字节码的第二个优势是安全性。由于字节码在 JVM 的控制下执行,因此 JVM 限制了恶意操作的 Java 程序。保证主机安全的能力对于 Java 的成功是至关重要的,因为它允许创建 Applet,由于 Applet 是可通过 Internet 动态下载的小程序,因此避免 Applet 破坏主机的机制是非常必要的。字节码和 Java 虚拟机的结合,保证了 Applet 的安全下载,保证了 Java 在 Web 编程中的地位。

(7) Java 类库

从概念上讲,计算机语言由两个部分组成:一是语言本身,由关键字和语法所定义;二是标准库,包含一系列面向程序的类、接口和方法。人们在最初创建 Java 时,它的库包括一组核心程序包,例如:java.lang,java.io 和 java.net。随着 Java 不断发布新的版本,新的类和程序包被不断添加进来。从 Java 创建之初,Java 函数库中就提供了对网络的支持。在开发 Java 程序的时候,它通过提供相应的类,使得连接和使用 Internet 的处理非常方便,有力地推动了 Internet 的发展。

Java 核心库中另外一个关键的程序包是 Java.awt,它支持抽象窗口工具集。程序员用 AWT 创建可移植的、基于 GUI 的代码。在程序中使用各种标准的 GUI 元素。例如:滚动条、复选框和单选框等。由于 AWT 的存在,程序员创建的 GUI 应用程序可运行在任何支持 Java 虚拟机的环境中。而这种层次上的 GUI 移植性,在 Java 之前是没有的。在 Java 中加入 AWT 彻底改变了程序员思考应用程序环境的方式。在 Java 之前,基于 GUI 的程序必须明确指出其运行环境。这意味着任何 Windows 程序需要重新编译才能在一台 Applet 机上运行。Java 通过一个可移植的 GUI 提供统一的编程环境。后来 Swing 被添加入 Java 中,Swing 为程序员提供了一组丰富的 GUI 组件。与 AWT 相比,它的移植性更高。如今,Java 库添加了许多新的库,新的功能不断添加到已有的程序包中。

1.2 Java 运行环境

1.2.1 Java 平台

Java 平台是由 Java 虚拟机和 Java 应用程序接口(API)组成的,如图 1-1 所示。

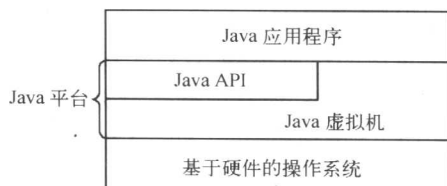


图 1-1 Java 平台示意图

1.2.2 安装 JDK

Java 是一个开放的开发环境,Java 开发工具包(JDK)和 JDK 技术文档可在 Sun 公司的网站中直接下载(<http://www.sun.com/Downloads/>),下载得到的 JDK 软件包与 JDK 技术文档如图 1-2 所示。

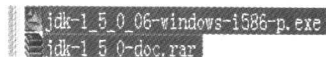


图 1-2 JDK 软件包与 JDK 技术文档

安装方法如下。

从 <http://www.sun.com/Downloads/> 下载 Java 开发包及技术文档包。

双击 Java 开发包安装程序,出现安装界面,如图 1-3 所示。

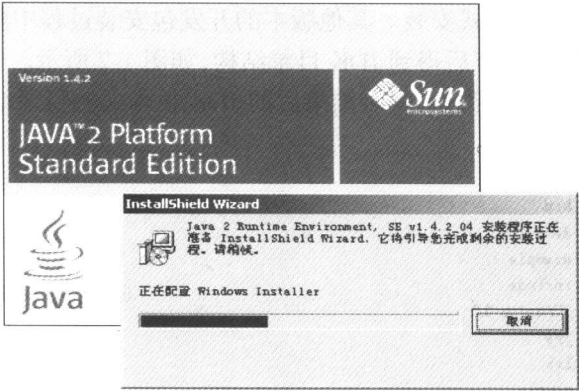


图 1-3 安装界面

安装程序弹出许可证条款界面,如图 1-4 所示。
选择接受协议条款,然后单击“下一步”按钮,出现安装类型选择界面,如图 1-5 所示。

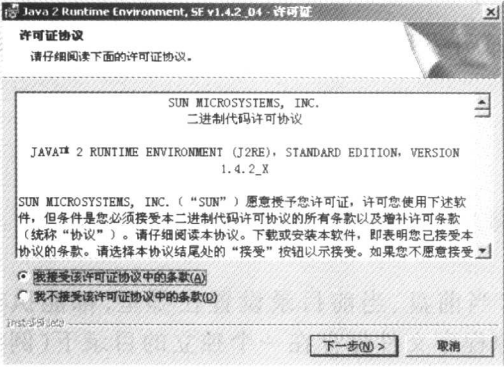


图 1-4 许可证条款界面

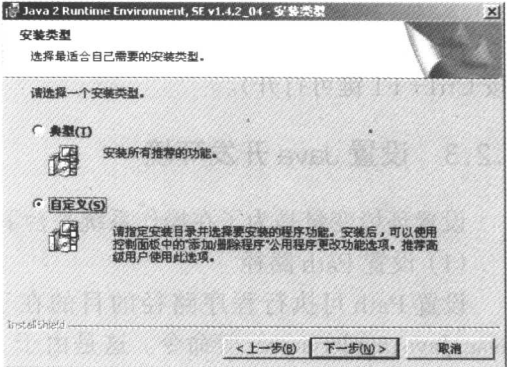


图 1-5 安装类型选择界面

选择“自定义”,然后单击“下一步”按钮,出现图 1-6 所示的“自定义安装”选择界面。

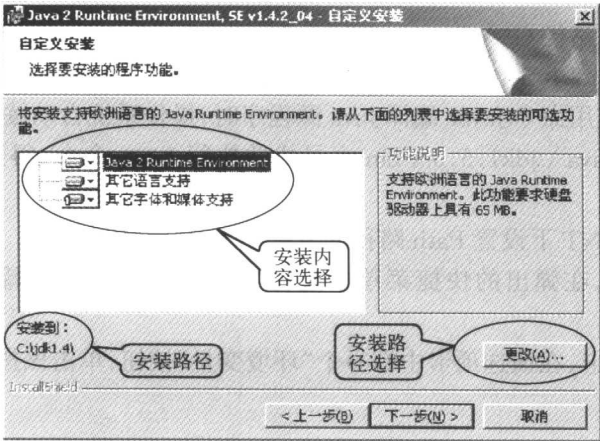


图 1-6 自定义安装界面

单击“下一步”按钮进入正式安装。其他版本的开发包安装过程中除了显示画面略有不同外,其他方面都类似。安装结束后得到 JDK 目录结构,如图 1-7 所示。

JDK 技术文档的安装过程类似,安装结束后的 JDK 技术文档目录结构如图 1-8 所示。

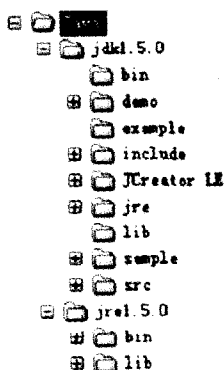


图 1-7 JDK 目录结构

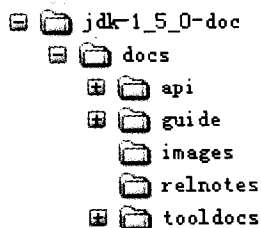


图 1-8 JDK 技术文档目录结构

注:安装 JDK 技术文档目的是为了在 Java 集成开发环境(JCreator)中获取联机文档帮助(按 Ctrl + F1 键可打开)。

1.2.3 设置 Java 开发环境

设置环境变量是为了在操作系统平台下更有效地对 Java 源程序进行编辑、编译并运行。

(1) 设置 Path 路径

设置 Path 可执行程序路径的目的在于不管当前盘、当前目录设置在哪里,都能执行 Javac, Java, appletviewer 等命令。这是由于 Java 源程序文件保存在一个独立的目录下(例如 F: \ JavaExample),而 JDK 安装完毕后,就把 Java 编译器 Javac.exe 和运行字节码程序 Java.exe 安装在 JDK1.5.0 \ bin 目录下。如果把 F: \ JavaExample 设置为当前目录,又没有设置 Path 路径,在 DOS 环境下用 Javac 命令直接编译 Java 源程序,会得到找不到 Javac 命令的错误结果。

假设 JDK 安装在路径 G: \ ProgramFiles \ Java \ jdk1.5.0 下,则使用 Path 命令将 G: \ ProgramFiles \ Java \ jdk1.5.0 \ bin 设置为可执行程序的搜索路径,当在 Java 源程序目录 F: \ JavaExample 中使用命令 Javac 编译 Java 源程序时,操作系统自动按照 Path 环境设置路径 G: \ ProgramFiles \ Java \ jdk1.5.0 \ bin 去寻找并执行 Javac 命令,得到 Java 源程序的编译结果。

在 Windows XP/NT 下设置 Path 路径的方法如下。

右击“我的电脑”,在弹出的快捷菜单中选择“属性”,出现“系统特性”对话框,如图 1-9 所示。

选择“高级”选项卡,在该选项卡中有一个“环境变量”按钮,单击该按钮进入“环境变量”窗口,如图 1-10 所示。

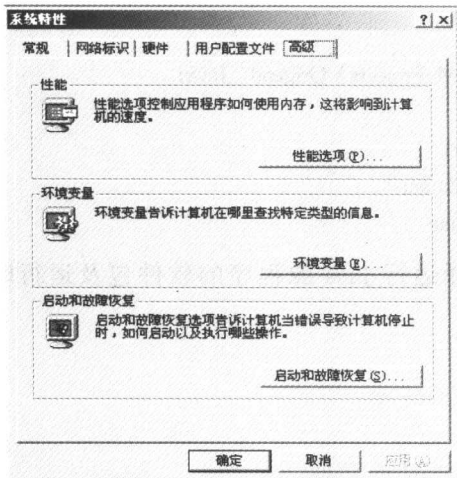


图 1-9 “系统特性”对话框

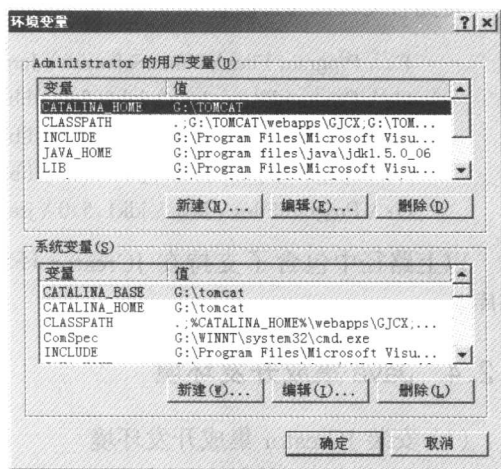


图 1-10 “环境变量”窗口

(2) 设置类路径 Classpath

在 Java 源程序中经常见到下列语句：

```
import java.awt.*;  
import java.net.*;
```

这是为了将 Java 中的系统包或类库导入到当前源文件中联编,这时就需要设置类路径,Java 编译器会根据 Classpath 的设置,自动搜索 Java 安装目录下的 awt 包和 net 包,把该包中的类导入到当前源文件中,以支持 Java 程序的正常运行。在操作系统环境下设置类路径的命令格式如下:

```
SET CLASS PATH=<类路径 1>;<类路径 2>;...;<类路径 n>
```

但在安装 JCreator 过程中会根据 JDK 目录自动搜索类路径,打开 Project Properties(工程属性)对话框,如图 1-11 所示。

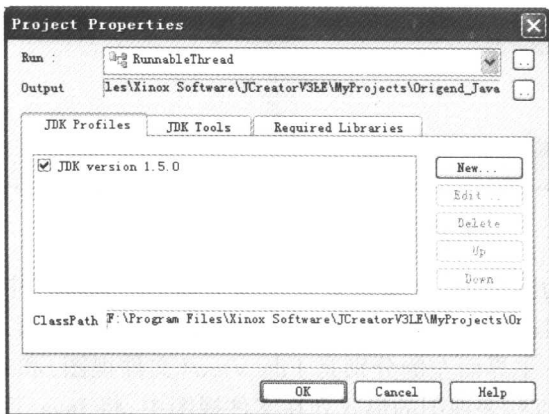


图 1-11 工程属性对话框

类路径 ClassPath 设置如下：

```
E: \ Program Files \ XinoxSoftware \ JcreatorV4 \ MyProjects \ Origend _ Java;  
G: \ ProgramFiles \ Java \ jdk1.5.0 \ lib \ dt.jar;  
G: \ ProgramFiles \ Java \ jdk1.5.0 \ lib \ tools.jar;  
G: \ ProgramFiles \ Java \ jdk1.5.0 \ lib \ classes12.jar;  
G: \ ProgramFiles \ Java \ jdk1.5.0 \ jre \ lib \ rt.jar
```

以上路径中包含了支持在 JCreator 环境下编译运行 Java 源程序的软件包及运行时动态库。

1.2.4 Java 集成开发环境

(1) 安装 JCreator 集成开发环境

JCreator400.zip 软件包可在 <http://www.jcreator.com/> 中下载。JCreator 的主界面如图 1-12 所示。

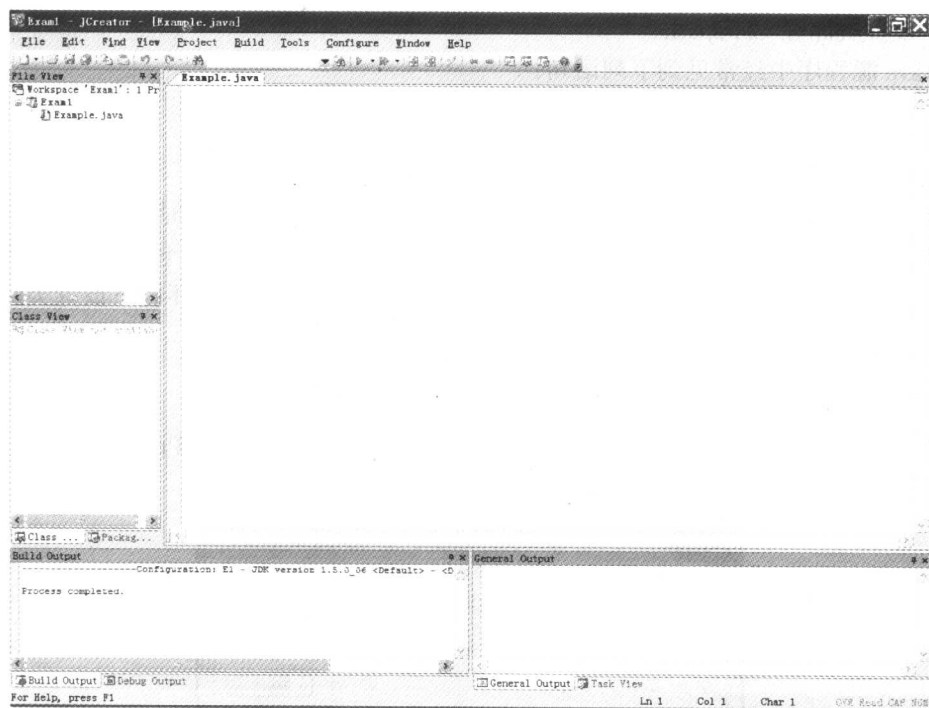


图 1-12 JCreator 主界面(Windows XP)

JCreator 主界面分为五部分,上面是菜单栏和工具栏。菜单栏中包括 File,Edit,Search,View,Project,Build,Tools,Configure,Window,Help 10 个主菜单项;工具栏中包含系统中常用的工具栏按钮。JCreator 主界面左侧分别是 File View(文件视图)和 Class View(类视图),将 Java 工程所包含的文件以列表视图的形式列在文件视图中;将 Java 工程中包含的类以列表视图的形式列于类视图中。

JCreator 主界面的右侧是 Java 源程序代码的编辑窗口,用于输入、编辑、编译 Java 源程序