



高职高专计算机专业“十二五”规划教材

Java程序 案例教程

主 编 田 智 杨艳慈
副主编 王 欣 李占平 旭 日
曹 慧 杨 飞



西安电子科技大学出版社
<http://www.xduph.com>

高职高专计算机专业“十二五”规划教材

Java 程序案例教程

主 编 田 智 杨艳慈

副主编 王 欣 李占平 旭 日

 曹 慧 杨 飞

参 编 任 靖 荣艳冬

西安电子科技大学出版社

内 容 简 介

Java 是当今最受欢迎的网络编程语言之一，它是一种安全、简单、完全面向对象、解释执行、动态下载、具有多线程能力、健壮且安全的新一代语言。

本书共 13 章，内容包括 Java 语言特色和 Java 开发环境介绍，Java 的数据类型、常量、运算符、表达式、控制结构等编程基础知识，Java 中类、对象、继承、多态、成员变量、成员方法等面向对象程序设计的概念、特点和用法，数组和字符串的声明、创建和用法，异常的概念及用法，数据输入与输出方法，Java 的多线程机制及其用法，Java 网络程序设计以及图形用户界面设计。

本书结构合理，语言简练，通俗易懂，实例丰富，并配有完善的资料包(包括书中的全部实例)，非常适合作为高职高专院校的教材，也可供 Java 爱好者参考。

图书在版编目(CIP)数据

Java 程序案例教程/田智，杨艳慈主编. —西安：西安电子科技大学出版社，2013.9

高职高专计算机专业“十二五”规划教材

ISBN 978-7-5606-3187-5

I. ① J… II. ① 田… ② 杨… III. ① JAVA 语言—程序设计—高等职业教育—教材
IV. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 209203 号

策 划 胡华霖

责任编辑 许青青

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2013 年 9 月第 1 版 2013 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 21

字 数 496 千字

印 数 1~3000 册

定 价 33.00 元

ISBN 978-7-5606-3187-5/TP

XDUP 3479001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜，谨防盗版。

专家委员会名单

编委会主任：田 智 杨艳慈 李占平

编委会成员：（排名不分先后）

卜耀华 冯建平 荣艳冬 高 丽

武书琴 海 川 邢苗苗 包志林

王 欣 曹 慧 杨 飞 任 靖

前 言

Java 是当今最受欢迎的网络编程语言之一，它是一种安全、简单、完全面向对象、解释执行、动态下载、具有多线程能力、健壮且安全的新一代语言。

在最受欢迎的程序设计语言排行榜上，Java 语言已经连续数年位列榜首。“Write once, Run anywhere(一次编写，到处运行)”是 Java 语言的基本特征，这是一种很有效率的编程方式。跨平台、完全面向对象、既适于单机编程也适于 Internet 编程等特点，给 Java 语言注入了强大的生命力，因此 Java 语言取得了举世瞩目、全球公认的地位。

当前，Java 技术应用范围非常广泛，电子设备嵌入 Java 技术，PC 中广泛应用 Java 技术，电信运营商支持 Java 技术，Java 智能卡海量发行，Java 开发人员数量也日增月长。另外，Java 开发者锐意创新，将 Java 技术应用到各种领域，使得 Java 技术已经成为世界上卓越的企业应用和移动应用开发平台之一。

本书主要阐述了 Java 语言的关键字、语句、语法规则以及类库等的使用方法，具体内容包包括 Java 语言开发工具的介绍、面向对象编程的概念、Java 基础语句、Java 类、类的继承与多态等特性、接口、包、异常、Java 基础类库的应用，并通过应用型案例帮助读者掌握 Java 语言的输入和输出操作、图形用户界面操作、多线程处理、GUI 开发、网络编程等相关类库的使用。

本书的特点是：脉络清晰，易学好用；分篇的组织结构，给读者一个全局认识的框架；案例丰富，剖析透彻，用来阅读的程序和需要设计的程序，均给出到位的分析。从起点上说，本书兼顾了零起点和有语言基础的读者；从目标上说，本书适合具有 Java 程序设计基础目标和进阶目标的读者。

田智、杨艳慈担任本书主编，王欣、李占平、旭日、曹慧、杨飞担任副主编，任靖和荣艳冬也参与了编写。

本书在编写的过程中得到了北京神州优胜教育科技有限公司经理杨雄的关心和指导，在此表示衷心的感谢！

由于编写时间仓促，加之编者水平和经验有限，书中不足之处在所难免，敬请广大读者批评指正。

编 者
2013 年 6 月

目 录

第 1 章 认识 Java.....	1	3.1.3 基本数据类型	26
1.1 Java 的历史.....	1	3.1.4 数据类型的转换	35
1.2 Java 的现状.....	2	案例 1 数据类型转换.....	37
1.3 Java 的特点	2	3.2 运算符、表达式与语句	38
1.3.1 Java 语言的优点	2	3.2.1 表达式与运算符	38
1.3.2 Java 语言的关键特性	3	3.2.2 运算符的优先级	46
1.4 Java 虚拟机(JVM)	4	3.2.3 表达式	47
1.5 Java 的开发工具与开发环境	5	3.2.4 表达式的类型转换	49
1.5.1 JDK 的下载、安装及环境 变量配置	5	案例 2 根据成绩计算绩点	50
1.5.2 JDK 的主要开发工具 Eclipse	8	3.3 循环与选择性语句	51
1.6 编写第一个 Java 程序	11	3.3.1 程序的结构设计	51
实训 Java 程序的运行	12	3.3.2 选择结构	52
习题	13	3.3.3 多重选择语句——switch 语句.....	56
第 2 章 简单的 Java 程序	14	3.3.4 while 循环	58
2.1 一个简单的例子	14	3.3.5 do...while 循环	60
2.2 简单的 Java 程序解析	16	3.3.6 for 循环	61
2.2.1 类(class).....	16	3.3.7 循环嵌套	62
2.2.2 大括号、段及主体	16	3.3.8 循环的跳离	64
2.2.3 程序运行的起始点 main() method ...	17	3.3.9 局部变量	66
2.2.4 Java 程序的注释	17	案例 3 利用循环输出图形	67
2.2.5 Java 中的标识符	18	实训 输出某年某个月的天数	68
2.2.6 Java 的关键字	18	习题	70
2.2.7 变量	19	第 4 章 数组与方法	72
2.2.8 变量的设置	19	4.1 一维数组	72
2.2.9 println().....	20	4.1.1 一维数组的声明与内存的分配	72
2.3 程序的检测	20	4.1.2 数组中元素的表示方法	74
2.4 提高程序的可读性	22	4.1.3 数组初值的赋值	75
实训 Java 程序的运行	23	4.1.4 与数组操作有关的 API 方法.....	75
习题	24	案例 1 找出数组元素中的最大值与 最小值	77
第 3 章 Java 基本程序设计	25	4.2 二维数组	78
3.1 变量与数据类型	25	4.2.1 二维数组的声明与分配内存	78
3.1.1 变量与常量	25	4.2.2 二维数组元素的引用及访问	80
3.1.2 Java 的变量类型	26	4.3 多维数组	80

4.4 方法	81
4.4.1 方法操作的简单范例	82
4.4.2 方法的参数与返回值	83
4.4.3 方法的重载	84
4.4.4 将数组传递到方法中	85
案例 2 遍历数组	88
实训 逆序输出一个数组	89
习题	90
第 5 章 类的基本形式	91
5.1 面向对象程序设计的基本概念	91
5.1.1 对象	92
5.1.2 类	92
5.1.3 封装性	92
5.1.4 继承性	93
5.1.5 类的多态性	93
5.2 类与对象	93
5.2.1 类的声明	94
5.2.2 创建新的对象	95
5.2.3 创建多个新对象	97
案例 1 对象组合案例	98
5.3 类的封装性	99
5.4 在类内部调用方法	105
5.5 引用数据类型的传递	106
5.6 匿名对象	110
5.7 构造方法	110
5.8 对象的比较	115
5.9 this 关键字的使用	118
5.10 static 关键字的使用	123
5.10.1 静态变量	123
5.10.2 静态方法	127
5.10.3 main()方法	130
5.10.4 静态代码块	131
案例 2 static 成员变量和实例变量的区别	132
5.11 构造方法的私有	133
5.12 对象数组的使用	135
实训 计算面积	136
习题	138

第 6 章 类的继承	140
6.1 继承的基本概念	140
6.1.1 子类对象的实例化过程	142
6.1.2 super 关键字的使用	146
6.1.3 限制子类的访问	148
6.1.4 复写	150
案例 1 继承案例	154
6.2 抽象类	156
6.3 Object 类	159
6.4 final 关键字	161
6.5 接口(interface)	163
案例 2 接口案例	166
6.6 对象的多态性	167
6.6.1 instanceof 关键字的使用	170
6.6.2 复写 Object 类中的 equals 方法	172
6.6.3 接口对象的实例化	174
6.7 匿名内部类	177
实训 上转型对象调用子类重写	179
习题	180
第 7 章 异常处理	184
7.1 异常的基本概念	184
7.1.1 异常处理的目的	184
7.1.2 简单的异常范例	184
7.1.3 发生异常时的处理	185
7.1.4 异常处理的方法	188
7.2 异常类的继承架构	188
7.3 抛出异常	189
7.3.1 在程序中抛出异常	189
7.3.2 指定方法抛出异常	190
7.4 编写自己的异常类	192
习题	193
第 8 章 包及访问权限	197
8.1 包的概念及使用	197
8.1.1 包(package)的基本概念	197
8.1.2 import 语句的使用	198
8.1.3 JDK 中常见的包	200
8.2 类成员的访问控制权限	200
8.3 Java 的命名习惯	202
习题	202

第 9 章 多线程	205	10.4 字符编码.....	260
9.1 进程与线程.....	205	10.5 对象序列化.....	263
9.2 认识线程.....	206	实训 使用数据流读、写文件.....	265
9.2.1 通过继承 Thread 类实现多线程.....	207	习题.....	266
9.2.2 通过实现 Runnable 接口实现 多线程.....	209	第 11 章 Java 常用类库	268
9.2.3 两种多线程实现机制的比较.....	211	11.1 API 概念.....	268
9.3 线程的状态.....	216	11.2 String 类和 StringBuffer 类.....	268
9.4 线程操作的一些方法.....	217	11.3 基本数据类型的包装类.....	269
9.4.1 取得和设置线程的名称.....	218	11.4 System 类与 Runtime 类.....	270
9.4.2 线程是否启动的判断.....	221	11.4.1 System 类.....	270
9.4.3 后台线程与 setDaemon()方法.....	222	11.4.2 Runtime 类.....	272
9.4.4 线程的强制运行.....	223	11.5 Date 与 Calendar、DateFormat 类.....	272
9.4.5 线程的休眠.....	225	11.6 Math 与 Random 类.....	274
习题.....	227	实训 使用 Date 类或 Calendar 类处理 日期、时间.....	275
第 10 章 文件 IO 操作	229	习题.....	276
10.1 File 类.....	229	第 12 章 Java 网络程序设计	277
10.2 RandomAccessFile 类.....	231	12.1 Socket 介绍.....	277
案例 1 随机流.....	234	12.2 Socket 程序.....	278
10.3 流类.....	234	12.3 DatagramSocket 程序.....	286
10.3.1 字节流.....	235	实训 使用套接字进行简单的网络通信.....	289
10.3.2 字符流.....	239	习题.....	291
10.3.3 管道流.....	243	第 13 章 图形用户界面设计	292
10.3.4 ByteArrayInputStream 与 ByteArrayOutputStream.....	245	13.1 GUI 概述.....	292
10.3.5 System.in 和 System.out.....	247	13.1.1 AWT 简介.....	292
10.3.6 打印流.....	247	13.1.2 Swing 简介.....	293
10.3.7 DataInputStream 与 DataOutputStream.....	249	13.2 组件的创建与使用.....	293
10.3.8 合并流.....	253	13.3 布局管理器.....	298
10.3.9 字节流与字符流的转换.....	255	13.4 事件处理.....	302
10.3.10 IO 包中的类层次关系图.....	257	实训 制作简单的计算器.....	313
案例 2 缓冲流案例.....	258	习题.....	317
		附录 Java 定义的格式	318
		参考文献	326

第 1 章

认识 Java

1.1 Java 的历史

Java 是 Sun 公司于 20 世纪 90 年代初开发的。Java 最初并不是为了用于 Internet, 而是作为一种小家用电器的编程语言, 用来解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题, 命名为 Oak(橡树)。由于这些家用电器的市场需求当时没有预期的高, Sun 公司放弃了该项计划。就在 Oak 几近夭折之时, Internet 异常火爆起来。Sun 公司看到了 Oak 在计算机网络上的广阔应用前景, 于是改造了 Oak, 将 Oak 技术应用于 Web 上, 并且开发出了 HotJava 的第一个版本, 于 1995 年 5 月发表。HotJava 在产业界引起了巨大的轰动, Java 的地位也随之得到肯定。经过一年的试用和改进, Java1.0(JDK1.0)在 1996 年初正式发布。由于最初的 1.0 版和 1.1 版存在着不少缺点, Sun 公司在 1.2 版上花费了很大的力气进行全面的修正, 并加入了许多新设计。1.2 版较之过去的版本有着很大的差别。Sun 公司将 1.2 版及其以后的版本命名为“Java2”。

Java 的出现引发了遍及全球、至今未衰的 Java 热潮。由于 Java 提供了强大的图形、图像、动画、音频、多线程及网络交互能力, 因此它在设计交互式、多媒体网页和网络应用程序方面大显身手, 成为当今推广速度最快的一门计算机程序语言。

随着 Java2 的诞生, Java 形成了三个技术分支, 相应地也就产生了三个版本的 Java 运行平台。

(1) J2SE(Java2 Platform Standard Edition): 标准版, 主要用于开发桌面应用程序、低端的服务器应用程序和 Java Applet 程序。

(2) J2EE(Java2 Platform Enterprise Edition): 企业版, 主要用于构建企业级服务器应用。

(3) J2ME(Java2 Platform Micro Edition): 微型机版, 主要用于嵌入式消费产品, 如移动电话、掌上电脑以及其他无线设备。

Java 具有“一次编写, 到处运行”的独特之处, 用 Java 语言开发的软件可以不用修改或重新编译而直接应用于任何计算机上, 极大地提高了软件开发的效率。这一特性使 Java 具有广泛的应用潜力。目前 Java 主要应用于以下几个方面:

- (1) 计算过程可视化、可操作化的软件的开发;
- (2) 动态画面的设计, 包括图形、图像的调用;
- (3) 所有面向对象的应用开发, 包括面向对象的事件描述、处理、综合等;
- (4) 交互操作的设计(选择交互、定向交互、控制流程等);
- (5) Internet 的系统管理功能模块的设计, 包括 Web 页面的动态设计、管理和交互操作设计等;
- (6) Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件);
- (7) 方便与各种数据库利用 SQL 语句连接。
- (8) 其他应用类型的程序。

1.2 Java 的现状

Java 是 Sun 公司推出的新一代面向对象程序设计语言, 特别适于 Internet 应用程序开发, 它的平台无关性直接威胁到微软的垄断地位, 主要表现在以下几个方面:

- (1) 计算机产业的许多大公司购买了 Java 许可证, 包括 IBM、Apple、DEC、Adobe、SiliconGraphics、HP、Oracle、TOSHIBA 以及 Microsoft。这说明 Java 已得到了业界的认可。
- (2) 众多软件开发商支持 Java 软件产品。例如, Inprise 公司、Sysbase 公司和 Oracle 公司均已支持 Java。
- (3) Intranet 正在成为企业信息系统的最佳解决方案, 而其中 Java 将发挥不可替代的作用。Intranet 的目的是将 Internet 用于企业内部, 它的优点是便宜, 易于使用和管理, 用户不管使用何种类型的机器和操作系统, 界面是统一的 Internet 浏览器, 而数据库、Web 页面、Applet、Servlet、JSP 则存储在 Web 服务器上, 无论是开发人员还是管理人员, 或是用户都可以受益于该解决方案。

1.3 Java 的特点

1.3.1 Java 语言的优点

Java 语言是一种优秀的编程语言。它的最大优点就是与平台无关, 在 Windows XP、Windows NT、Solaris、Linux、MacOS 以及其他平台上, 都可以使用相同的代码。“一次编写, 到处运行”的特点, 使其在互联网上被广泛采用。由于 Java 语言的设计者十分熟悉 C++ 语言, 所以在设计时很好地借鉴了 C++ 语言。可以说, Java 语言是一种比 C++ 语言还“面向对象”的编程语言。Java 语言的语法结构与 C++ 语言的语法结构十分相似, 这使得 C++ 程序员学习 Java 语言更加容易。

当然, 如果仅仅是对 C++ 改头换面, 那么就不会有今天的 Java 热了。Java 语言提供一些有用的新特性, 使得使用 Java 语言比 C++ 语言更容易写出“无错代码”。这些新特性包括:

- (1) 提供了对内存的自动管理, 程序员无需在程序中分配、释放内存, 那些“可怕”

的内存分配错误不会再打扰设计者。

(2) 去除了 C++ 语言中令人费解、容易出错的“指针”，用其他方法来进行弥补。

(3) 避免了赋值语句(如 `a = 3`)与逻辑运算语句(如 `a == 3`)的混淆。

(4) 取消了多重继承这一复杂的概念。

Java 语言的规范是公开的，可以在 <http://www.sun.com> 上找到。阅读 Java 语言的规范是提高技术水平的好方法。

1.3.2 Java 语言的关键特性

Java 语言的关键特性有以下几个：

(1) 简洁有效。Java 语言是一种相当简洁的“面向对象”程序设计语言。Java 语言省略了 C++ 语言中所有难以理解、容易混淆的特性，如头文件、指针、结构、单元、运算符重载、虚拟基础类等，它更加严谨、简洁。

(2) 可移植性。对于一个程序员而言，写出来的程序如果不需修改就能够同时在 Windows、MacOS、UNIX 等平台上运行，简直就是美梦成真的好事，而 Java 语言让这个原本遥不可及的事情变得越来越近了。使用 Java 语言编写的程序，只要做较少修改，甚至有时根本不需修改就可以在不同平台上运行。

(3) 面向对象。可以这么说，“面向对象”是软件工程学的一次革命，大大提升了人类的软件开发能力，是一个伟大的进步，是软件发展的一个重大的里程碑。在过去的 30 年间，“面向对象”有了长足的发展，充分体现了其自身的价值，到现在已经形成了一个包含“面向对象的系统分析”、“面向对象的系统设计”、“面向对象的程序设计”的完整体系。所以 Java 语言作为一种现代编程语言，是不能够偏离这一方向的。

(4) 解释型。Java 语言是一种解释型语言，相对于 C/C++ 语言来说，用 Java 语言写出来的程序效率低，执行速度慢。但它正是通过在不同平台上运行 Java 解释器对 Java 代码进行解释来实现“一次编写，到处运行”的宏伟目标的。为了达到目标，牺牲效率是值得的，况且现在的计算机技术日新月异，运算速度也越来越快，用户并不会感到太慢。

(5) 适合分布式计算。Java 语言具有强大的、易于使用的联网能力，非常适合开发分布式计算的程序。Java 应用程序可以像访问本地文件系统那样通过 URL 访问远程对象。使用 Java 语言编写 Socket 通信程序十分简单，使用它比使用任何其他语言都简单。Java 语言还适用于公共网关接口(CGI)脚本的开发。另外，还可以使用 Java 小应用程序(Applet)、Java 服务器页面(Java Server Page, JSP)、Servlet 等来构建更丰富的网页。

(6) 拥有较好的性能。正如前面所述，由于 Java 是一种解释型语言，所以它的执行效率相对就会低一些，但由于 Java 语言采用了两种手段，因此其性能还是不错的。

① Java 语言源程序编写完成后，先使用 Java 伪编译器进行伪编译，将其转换为中间码(也称为字节码)，之后再解释。

② Java 语言提供了一种“准实时”(Just-in-Time, JIT)编译器，当需要更快的速度时，可以使用 JIT 编译器将字节码转换成机器码，然后将其缓冲下来，这样速度就会更快。

(7) 健壮，防患于未然。Java 语言在伪编译时，做了许多早期潜在问题的检查，并且在运行时又做了一些相应的检查，可以说是一种最严格的“编译器”。它的这种“防患于未然”的手段将许多程序中的错误消灭在萌芽状态，经常有许多在其他语言中必须通过运行

才会暴露出来的错误，在 Java 程序编译阶段就被发现了。另外，Java 语言还具备许多保证程序稳定、健壮的特性，有效地减少了错误，这样使得 Java 应用程序更加健壮。

(8) 具有多线程处理能力。线程是一种轻量级进程，是现代程序设计中必不可少的一种特性。多线程处理能力使得程序能够具有更好的交互性、实时性。Java 在多线程处理方面性能超群，具有让设计者惊喜的强大功能，而且在 Java 语言中进行多线程处理很简单。

(9) 具有较高的安全性。Java 语言在设计时，在安全性方面考虑得很仔细，做了许多探究，这使得 Java 语言成为目前最安全的一种程序设计语言。尽管 Sun 公司曾经许诺：“通过 Java 可以轻松构建出防病毒、防黑客的系统”，但“世界上没有绝对的安全”这一真理是不会因为某人的许诺而失灵的。就在 JDK (Java Development Kit)1.0 发布不久后，美国 Princeton(普林斯顿)大学的一组安全专家发现了 Java 1.0 安全特性中的第一例错误。从此，Java 安全方面的问题开始被关注。不过至今所发现的安全隐患都微不足道，而且 Java 开发组成员还宣称，他们对系统安全方面的 Bug 非常重视，会对这些被发现的 Bug 立即进行修复。另外，由于 Sun 公司开放了 Java 解释器的细节，所以有助于通过各界力量，共同发现、防范、制止这些安全隐患。

(10) 是一种动态的语言。这表现在以下两个方面：

- ① 在 Java 语言中，可以简单、直观地查询运行时的信息。
- ② 可以将新代码加入到一个正在运行的程序中。

(11) 是一种中性结构。Java 编译器生成的是一种中性的对象文件格式。也就是说，Java 编译器通过伪编译后，将生成一个与任何计算机体系统无关的“中性”的字节码。Java 的这种字节码经过了许多精心的设计，使得其能够很好地兼容于当今大多数流行的计算机系统，在任何机器上都易于解释，易于动态翻译成为机器代码。

1.4 Java 虚拟机(JVM)

Java 虚拟机(JVM)是可运行 Java 代码的假想计算机。只要根据 JVM 规范描述将解释器移植到特定的计算机上，就可保证经过编译的任何 Java 代码能够在该系统上运行。Java 虚拟机如图 1-1 所示。

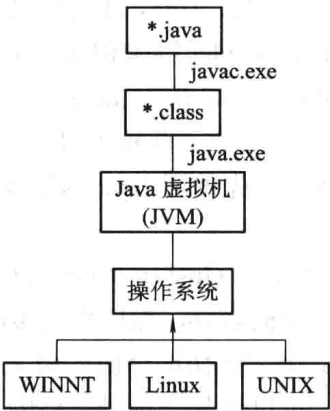


图 1-1 Java 虚拟机

从图 1-1 中不难明白 Java 可以具有可移植性的原因。只要在操作系统(Windows NT、Linux、UNIX)上植入 JVM(Java 虚拟机), Java 程序就具有可移植性,这也符合 Sun 公司提出的口号“Write Once, Run Anywhere”(一次编写,到处可行)。

1.5 Java 的开发工具与开发环境

1.5.1 JDK 的下载、安装及环境变量配置

Sun 公司于 2007 年年初发布了 JDK6.0 版本。下面介绍 JDK 6 Update 12 版本的下载、安装及配置方法。

1. 下载

JDK6.0 可以从 Sun 公司的官方网站 <http://java.sun.com/> 中下载。在页面右边 Popular Downloads 栏中可看到最新版的 JDK 软件包,如图 1-2 所示。打开“Java SE”超链接,单击“JDK 6 Update 12”超链接,开始下载 jdk-6-windows-i586(1).exe,如图 1-3 所示。

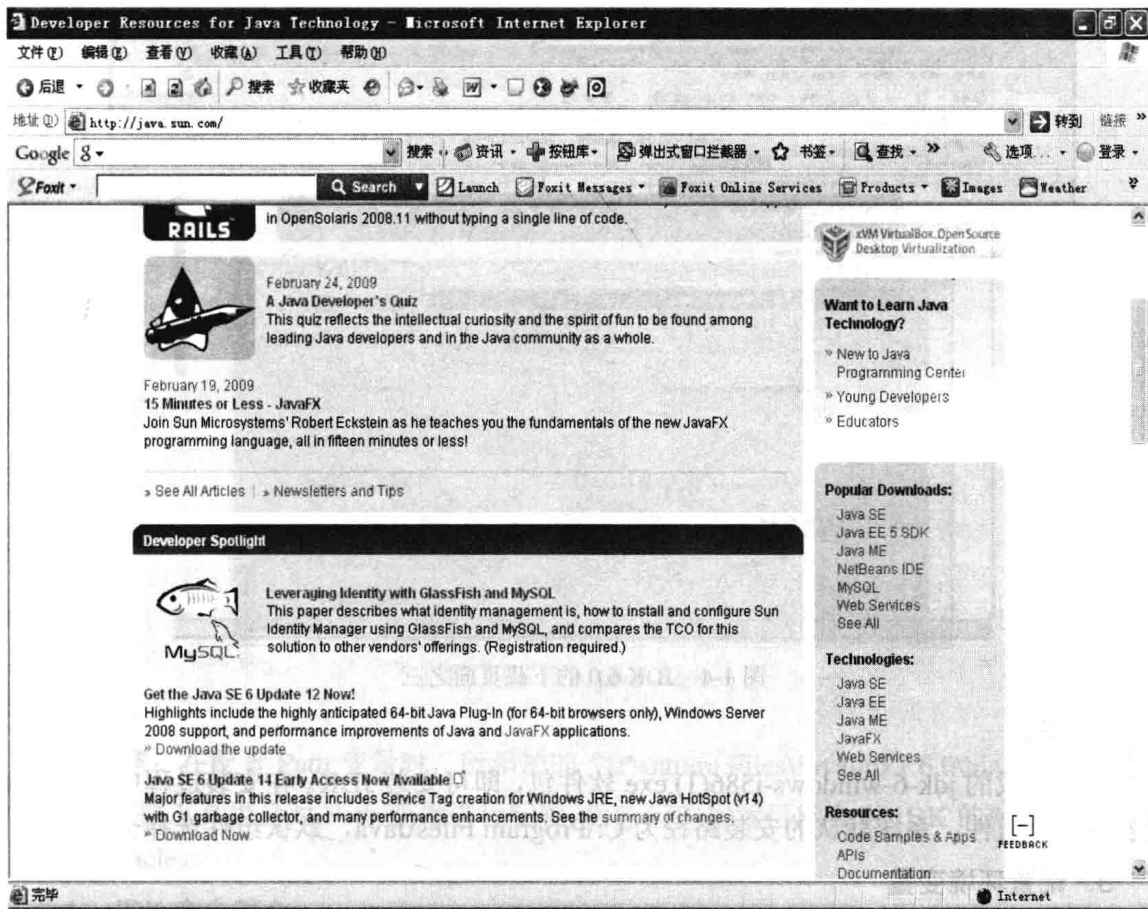


图 1-2 JDK6.0 的下载页面之一



图 1-3 JDK 6.0 的下载页面之二

在进行下载之前，要选择“Platform”为“Windows”，勾选同意下载协议，如图 1-4 所示。

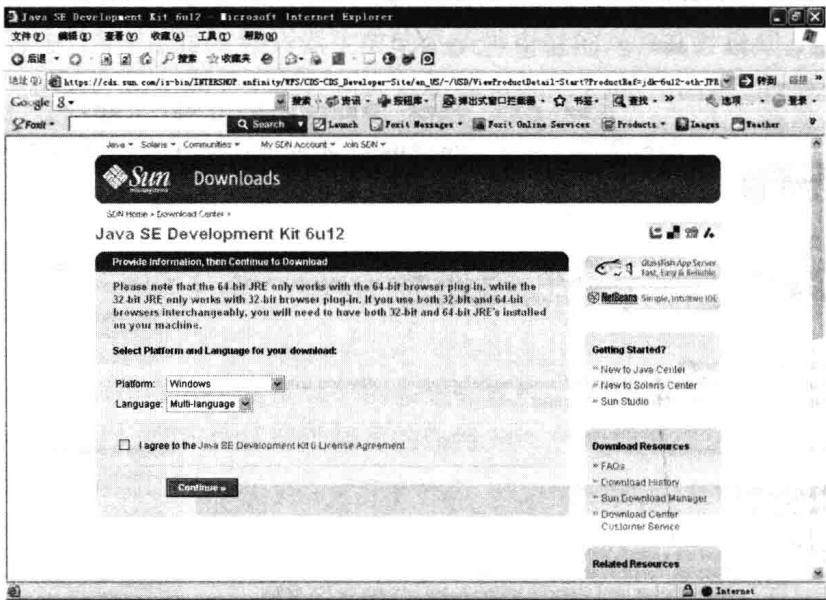


图 1-4 JDK 6.0 的下载页面之三

2. 安装

运行下载的 jdk-6-windows-i586(1).exe 软件包，即可安装 JDK。在安装过程中可以设置安装路径及组件，系统默认的安装路径为 C:\Program Files\Java，默认组件选择全部安装。

3. 配置环境变量

在安装好 JDK 后，还需要对系统的环境变量进行一些必要的配置才能使用。

用鼠标右键单击“我的电脑”，选择“属性”，在系统属性对话框中单击“高级”按钮，单击“环境变量”按钮，打开“环境变量”对话框，如图 1-5 所示。找到变量名“Path”，

将其选中后,单击“编辑”按钮,增加路径“\Program Files\Java\jdk1.6.0\bin”,如图 1-6 所示。新建变量名“CLASSPATH”,其值为“.”,如图 1-7 所示。保存后,打开一个新的 DOS 窗口,输入 `java -version` 命令,发现已经生效了,如图 1-8 所示。

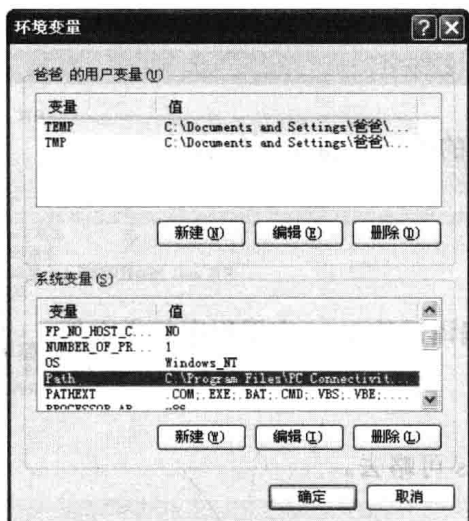


图 1-5 找到环境变量(Path)

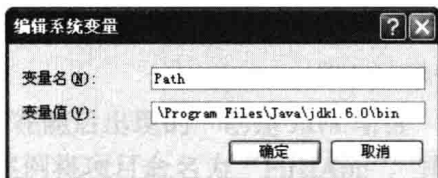


图 1-6 增加路径

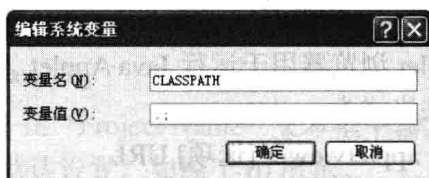
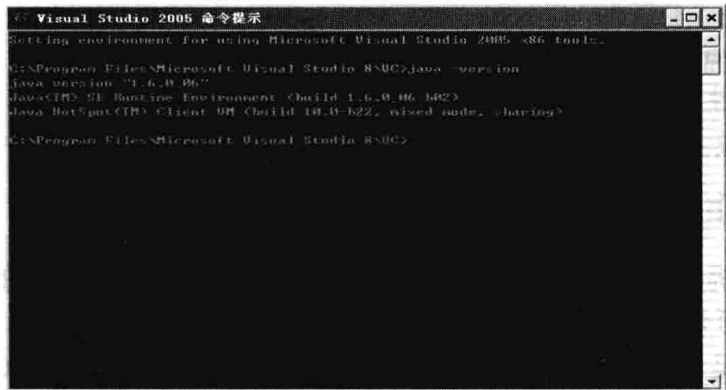


图 1-7 新建变量名

图 1-8 输入 `java -version`

注意: 在配置 Path 变量时,所添加的“\Program Files\Java\jdk1.6.0\bin”要放在最前面。因为其他工具也会对 Path 变量进行配置,而它们的配置可能会对 JDK 的配置产生影响,尤其是 Oracle。

4. JDK 命令简介

Java JDK 包括了所有编译、运行 Java 程序所需要的工具: Java 基本组件、库、Java 编译器、Java 解释器、小应用程序浏览器等。这些工具都在 Java JDK 安装目录下的 bin 子目

录中。下面介绍一些基本的工具。

1) Java 编译器 javac.exe

Java 编译器将以“.java”为扩展名的 Java 源程序文件编译成类文件(扩展名为.class)。

语法格式为

javac [选项] 源程序文件名



注意：源程序文件名的扩展名.java 不能省略。

例如：

javac Hello.java

2) Java 解释器 java.exe

Java 解释器用于执行编译过的 Java 应用程序的类文件。

语法格式为

java [选项] 类名

类文件名的扩展名.class 可略去。

例如：

java Hello

3) Applet 浏览器 appletviewer.exe

Applet 浏览器用于运行 Java Applet 程序。

语法格式为

appletviewer [选项] URL

URL 表示 URL 描述的 HTML 文档，文件的扩展名必须是.html。

例如：

appletviewer HelloApplet.html

4) Java 文档生成工具 javadoc.exe

Java 文档生成工具用于在 Java 源文件中生成基于标志的 HTML 文件。

1.5.2 JDK 的主要开发工具 Eclipse

虽然 JDK 中提供了一些编译、运行和调试程序的工具，但其命令行的工作方式使用户感到不方便。下面介绍一款优秀的 IDE 集成开发环境 Eclipse。

Eclipse 最初是 IBM 的一个软件产品，前期投入了 4000 万美金。2001 年 11 月，IBM 宣布将其捐给开放源码组织 Eclipse.org。目前 Eclipse 的市场占有率已经超过了 Borland 公司的 JBuilder，成为 Java 开发平台中的主流。Eclipse 的设计思想是：一切皆为插件。Eclipse 自身的核心是非常小的，其他所有的功能都以插件的形式附加到该核心上。

Eclipse 中最吸引人的地方包括三个：一是它创新性的图形 API，即 SWT/JFace；二是它的插件机制；三是利用它的插件机制开发的众多功能强大的插件。

Eclipse 平台是一个具有一组强大服务的框架，这些服务支持插件，比如 Java 开发环境插件(JDT)、插件开发环境(PDE)和其他第三方插件(New tool)。Eclipse 平台由几个主要部分构成：平台运行库、工作区、工作台、团队支持和帮助。

本书案例基于绿色中文版 Eclipse 3.3 版本。下面介绍 Eclipse 的操作步骤。

- (1) 双击启动 Eclipse，首次运行时会出现“Workspace Launcher”对话框，用户可以在其中设置工作空间。本案例采用默认工作空间，单击“OK”，将进入 Eclipse 主界面。
- (2) 在菜单栏中选择“文件”→“新建”→“项目”选项，创建一个项目，在弹出的“新建项目”窗口中选择“Java 项目”，如图 1-9 所示，然后单击“下一步”按钮。

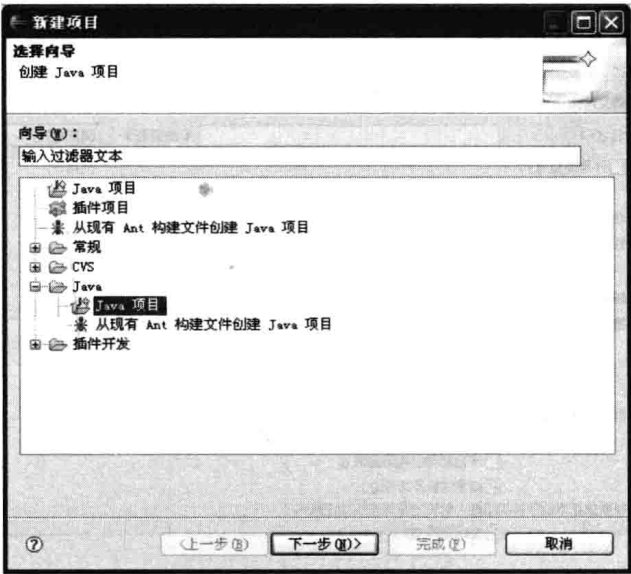


图 1-9 选择创建“Java 项目”

- (3) 在随后出现的“创建 Java 项目”窗口中，在“Project Name”文本框中输入项目名称。本案例将项目命名为“FirstApp”，其他采用默认设置，如图 1-10 所示。

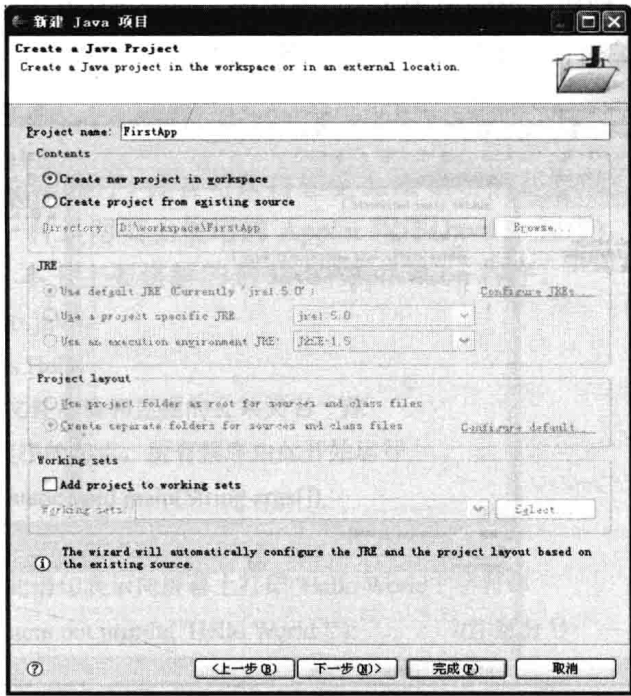


图 1-10 创建 FirstApp 项目

- (4) 单击“完成”按钮，之后就可以在项目中创建 Java 应用程序了。在菜单栏中依次