

2012年全国电子信息类优秀教材 浙江省“十一五”重点教材建设项目

21世纪高等教育网络工程规划教材

21st Century University Planned Textbooks of Network Engineering

Java面向对象程序设计 实践教程 (第3版)

Java Object-Oriented Programming
Practice Tutorial (3rd Edition)

杨晓燕 李选平◎主编



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

2012年全国电子信息类优秀教材 浙江省“十一五”

21世纪高等教育网络工程规划教材

21st Century University Planned Textbooks of Network Engineering

Java面向对象程序设计 实践教程 (第3版)

主要内容

Java Object-Oriented Programming
Practice Tutorial (3rd Edition)

杨晓燕 李选平◎主编



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Java面向对象程序设计实践教程 / 杨晓燕, 李选平
主编. — 3版. — 北京: 人民邮电出版社, 2015. 8
21世纪高等教育网络工程规划教材
ISBN 978-7-115-39421-7

I. ①J… II. ①杨… ②李… III. ①JAVA语言—程序
设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字 (2015) 第156438号

内 容 提 要

本书是《Java 面向对象程序设计 (第3版)》(杨晓燕 李选平主编, 人民邮电出版社同期出版) 的配套实验教材。全书依照基础实验、综合实践和综合设计的思路组织编写内容, 实验有程序填空、程序测试分析、程序设计等形式。全书实验内容主要包括: Java 开发工具及程序设计初步, Java 语言基础, Java 输入/输出, 程序流程控制、算法和方法设计, Java 数组, 类的结构和设计, UML 类图及面向对象设计的基本原则和模式, Java 包, GUI 和事件驱动, Java 图形及多线程, JDBC 编程, 综合设计等。

书中程序在 JDK5.0/JDK6.0 中经过验证, 并都给出了运行结果。

本书既可作为“Java 程序设计”和“面向对象程序设计”的实验及课程设计指导用书, 也可作为 Java 自学者的入门用书。

◆ 主 编 杨晓燕 李选平

责任编辑 邹文波

责任印制 沈 蓉 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市海波印务有限公司印刷

◆ 开本: 787×1092 1/16

印张: 12.75

字数: 346 千字

2015 年 8 月第 3 版

2015 年 8 月河北第 1 次印刷

定价: 29.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

第 3 版前言

诞生于 1995 年的 Java 语言，是目前最为流行的面向对象程序设计语言，它简单、高效、与平台无关、安全、支持多线程，是计算机世界的“国际语言”。面向对象技术具有模拟现实世界的思维方式，数据与操作相捆绑的程序风格符合现代大规模软件开发的要求，成为计算机应用开发领域的主流趋势。不仅如此，Java 的跨平台造就它在 Internet 上无可比拟的应用前景，使得 Java 成为当今 Internet 上最流行、最受欢迎的一种程序开发语言。原 Sun 公司总裁兼首席运营官 Jonathan Schwartz 说：“Java 技术正在成为全球网络应用的事实标准，它将大大加快和简化提供移动、消费和企业市场的服务。”现在 Java 平台仍继续为 Java 经济注入活力，驱动全球企业在移动应用和服务端领域的技术创新。

2005 年在 Java 发布 10 周年之际，我们编写了第 1 版 Java 面向对象程序设计实践教程，受到读者好评，连续加印多次。2012 年出版了第 2 版，依然获得很多赞誉；今年重新编写了第 3 版。本次修订在 Java 内容体系和结构上做了微调，突出“学中用”和“用中学”。

本书是《Java 面向对象程序设计（第 3 版）》（杨晓燕 李选平主编）的配套实验教材，主要供 Java 课程学生实验课和课程设计使用，书中提供了大量有趣、实用的案例，每个实验分为实验指导、程序设计和思考题，思考题主要针对程序促使学生思考代码结构和关键知识点，使用 UML 图使学生理解面向对象模块化的程序结构。实验题目涉及基础知识点训练、程序填空、程序测试分析、综合实践及综合设计等。本书重视知识的循序渐进和深入浅出，让学生在“练中学、学中思”，掌握 Java 面向对象的原则、方法和程序编写。所有程序填空和基础题在本书附录中提供参考答案。

在本书出版之际，要感谢我的师长姜遇姬教授的指导，感谢我的同事邓芳、刘臻、张梁斌，及我的学生的帮助与支持。

由于编者水平所限，书中难免还存在一些缺点和错误，希望读者批评指正。联系方式 yangxy3225@163.com。

编 者
2015 年 5 月

目 录

第 1 章 Java 开发工具及程序设计初步	1
1.1 J2SDK 开发工具入门	1
1.1.1 JDK 的下载、安装	1
1.1.2 环境变量和配置	4
1.1.3 JDK 开发工具简介	6
1.2 Java 程序开发步骤	6
1.3 Java 程序基本结构	8
1.4 良好的编程习惯	9
1.5 实验目的	9
1.6 实验内容	9
实验 1 第一个 Java 应用程序	9
实验 2 第一个 Java Applet 小程序	12
实验 3 读程序, 答问题	14
1.7 TextPad 工具的使用	15
第 2 章 Java 语言基础	17
2.1 知识点	17
2.2 实验目的	17
2.3 实验内容	17
实验 1 程序填空与测试分析	17
实验 2 编程测试 Java 数值类型的最大值和最小值	18
实验 3 韩信点兵问题	19
实验 4 实现简易移位加密	19
实验 5 基本数据类型应用: 自我介绍	20
第 3 章 Java 输入/输出	21
3.1 知识点	21
3.2 实验目的	21
3.3 实验内容	22
实验 1 标准输入/输出方法	22
实验 2 键盘输入——Scanner 类	23
实验 3 综合实践	25
第 4 章 程序流程控制、算法和方法设计	27
4.1 知识点	27
4.2 实验目的	28

4.3 实验内容	28
实验1 选择结构	28
实验2 循环结构	30
实验3 循环嵌套	33
实验4 迭代和穷举算法	34
实验5 综合实践	38
第5章 Java 数组	42
5.1 知识点	42
5.2 实验目的	43
5.3 实验内容	43
实验1 一维数组实验	43
实验2 二维数组实验	46
实验3 Arrays 类	48
实验4 综合实践	50
第6章 类的结构和设计	56
6.1 知识点	56
6.2 实验目的	57
6.3 实验内容	57
实验1 类的定义及对象的创建、使用	57
实验2 对象比较和字符串的比较	61
实验3 引用型参数传递	62
实验4 静态变量和静态方法应用	67
实验5 类的继承: this 和 super	69
实验6 抽象类和接口	72
实验7 方法重载和方法重构	75
实验8 成员变量的隐藏	76
实验9 泛型应用	76
实验10 综合实践	78
第7章 UML 类图及面向对象设计的基本原则和模式	93
7.1 知识点	93
7.2 实验目的	95
7.3 实验内容	95
实验1 面向抽象编程	95
实验2 多用组合少用继承编程	98
实验3 策略模式设计	99
实验4 中介者模式	101
实验5 模板方法模式	105
第8章 Java 包	107
8.1 知识点	107
8.2 实验目的	108
8.3 实验内容	108

实验 1 jar 包的创建	108
实验 2 包的定义和互连	111
第 9 章 GUI 和事件驱动	114
9.1 知识点	114
9.2 实验目的	116
9.3 实验内容	116
实验 1 组件应用入门	116
实验 2 文本框的应用	118
实验 3 菜单的应用	121
实验 4 窗口及对话框的应用	123
实验 5 表格的应用	126
实验 6 MVC 结构	128
实验 7 音乐播放器	130
实验 8 综合实践	132
第 10 章 Java 图形及多线程	139
10.1 知识点	139
10.2 实验目的	141
10.3 实验内容	141
实验 1 绘制图形	141
实验 2 用 Thread 类创建线程	144
实验 3 实现 Runnable 接口创建线程	145
实验 4 线程间的数据共享: 模拟航空售票	147
实验 5 多线程的同步控制: 模拟银行取款	148
实验 6 综合实践	149
第 11 章 JDBC 编程	152
11.1 知识点	152
11.2 实验目的	155
11.3 实验内容	155
实验 1 Access 数据库的创建与 ODBC 数据源	155
实验 2 运用 JDBC 操作数据库	158
第 12 章 综合设计	160
实验 1 UML 分析和模块化实现猜数字游戏	160
实验 2 UML 设计	163
实验 3 网络通信	167
实验 4 四则运算和日期计算	171
附录 部分实验参考答案	187

第1章

Java 开发工具及程序设计初步

1.1 J2SDK 开发工具入门

1.1.1 JDK 的下载、安装

1. 下载 JDK

J2SDK 是 Java 2 Software Development Kit 的简称,人们往往习惯简称为 JDK (Java Development Kit),即 Java 开发工具包。目前应用较多的版本是 JDK5.0 或 JDK6.0,读者根据运行平台的不同,下载相应的 JDK 版本。JDK 软件包提供了 Java 编译器、Java 解释器和 AppletViewer 浏览器等可执行文件,但没有提供 Java 编辑器,初学者推荐使用 Windows 的“记事本”。



提示

一个阶段学习之后,读者对 Java 编译、运行等命令已经熟悉了,可以在网上下载使用 TextPad,图标为 ,直接默认安装,前提是 JDK 已安装并配置好 path 环境变量。在 TextPad 中,可以打开已经编写好的程序,或者直接编辑程序,关键字会自动突出显示。编辑好之后,可以直接通过菜单编译、运行 Application 程序或 Applet 程序,非常方便。

学习 Java 语言初期,最好直接选用 Java SE 提供的 JDK。各种集成开发环境不仅系统界面复杂,还需要很多配置,而且会屏蔽掉一些知识点。在掌握了 Java 语言之后,再去熟悉、掌握一个流行的 Java 集成开发环境为好。

J2SDK 或 JDK 是原 Sun 公司免费提供的, Sun 公司目前被 Oracle 公司收购,最新 JDK 下载地址为 <http://www.oracle.com/technetwork/java/javase/downloads>。读者也可以在 www.google.com 或百度中搜索下载。

2. 安装 JDK5.0

由于目前大多数用户使用的是 Windows 操作系统,所以在此以在 Windows 操作系统上安装 jdk-1_5_0_06-windows-i586-p.exe 为例,说明安装 JDK5.0 的过程。其中包含了 Java 运行环境: Java Runtime Environment。

安装工作实际上分为两个步骤。安装程序首先会收集一些信息,用于安装的选择,然后才开始复制文件、设置 Windows 注册表等具体的安装工作。

双击 jdk-1_5_0_06-windows-i586-p.exe,安装初始界面,如图 1.1 所示。



图 1.1 安装初始界面

紧接着弹出准备安装界面，如图 1.2 所示。

随后出现 JDK5.0 的许可协议，如图 1.3 所示。

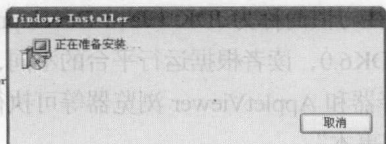


图 1.2 安装欢迎界面

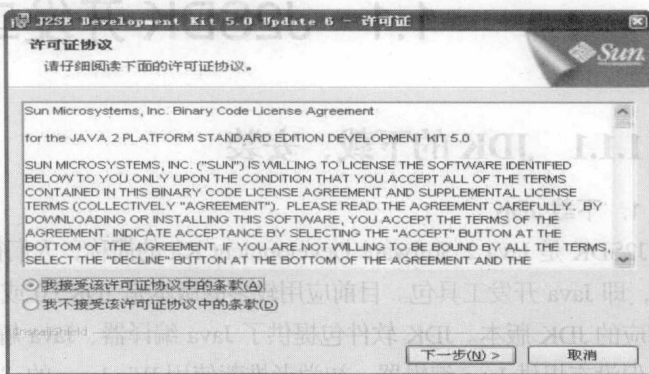


图 1.3 安装协议

选中接受协议条款，单击“下一步”按钮，安装程序会出现让用户选择安装目标路径的对话框，如图 1.4 所示。

在对话框中，选择系统默认路径 C:\Program Files\Java\jdk1.5.0_06\。单击“下一步”按钮，JDK 的所有程序就会被安装到 C:\Program Files\Java\jdk1.5.0_06\目录下。用户也可以在本对话框中单击“更改”按钮，选择 JDK 程序的其他安装路径。

紧接着出现安装进度提示界面，如图 1.5 所示。

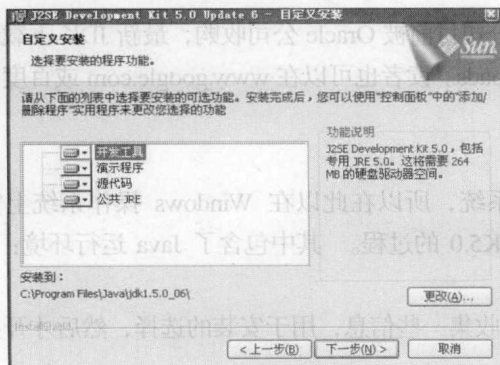


图 1.4 JDK 安装目标路径的选择

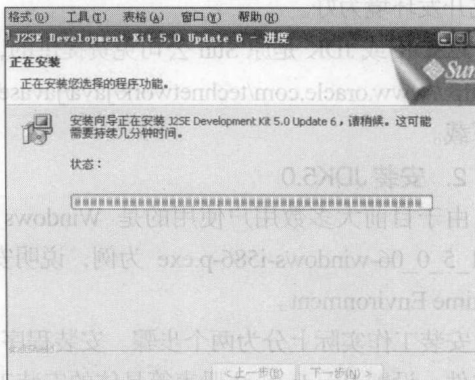


图 1.5 安装进度选择

接着,是自定义安装 J2SE Runtime Environment 5.0,使用默认路径,单击“下一步”按钮,如图 1.6 所示。

接着是浏览器注册,默认选择是 IE 浏览器,选择“下一步”按钮,如图 1.7 所示。

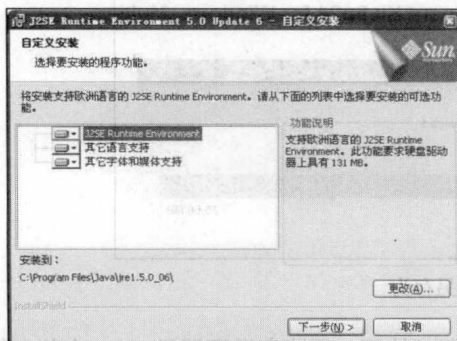


图 1.6 自定义安装选择

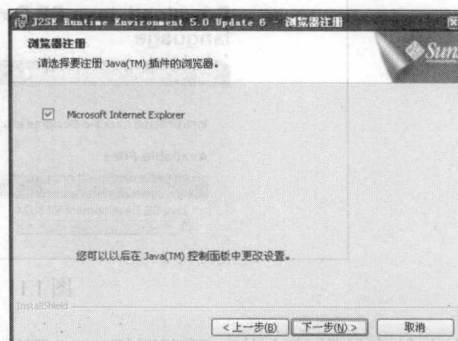


图 1.7 浏览器注册

出现 J2SE Runtime Environment 安装进度界面,如图 1.8 所示。

JDK 安装完成的提示界面如图 1.9 所示,单击“完成”按钮,结束安装。

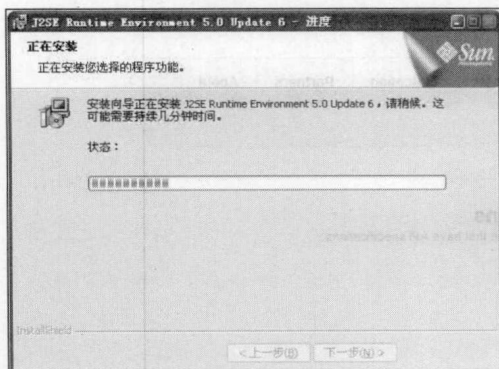


图 1.8 JRE 安装

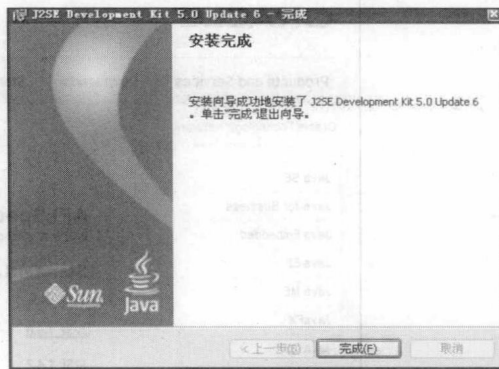


图 1.9 JDK 安装结束

3. 安装 JDK6.0

JDK6.0 (JDK1.6) 与安装 JDK5.0 类似,这里做简单介绍。

JDK6.0 官方下载页面如图 1.10 所示,在 Platform 下拉列表中,根据自己的计算机操作系统平台进行选择,这里选择“Windows”,同时选中其下的复选框。

单击 Continue 按钮,进入 JDK6.0 下载地址页面,如图 1.11 所示。右键单击下载链接,在弹出的快捷菜单中选择“另存为”命令,即可将大小为 76.58MB 的 JDK6.0 下载到自己的计算机中。



图 1.10 JDK6.0 下载界面

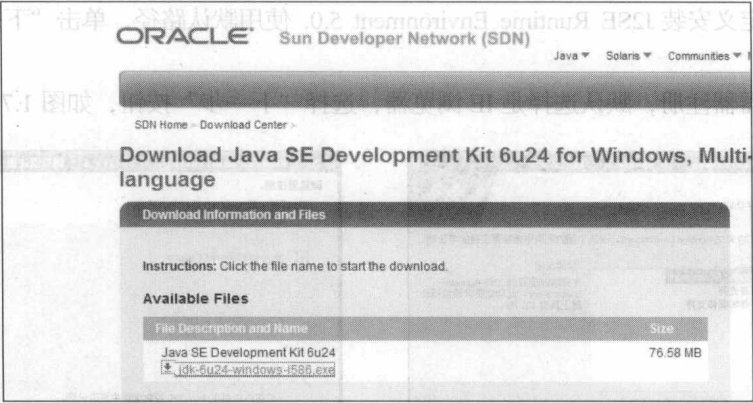


图 1.11 目标另存为

双击文件直接默认安装，再配置好 path 环境变量即可使用，环境变量配置见 1.1.2 小节。各个版本 API 参阅官方地址：<http://www.oracle.com/technetwork/java/api-141528.html>，如图 1.12 所示。

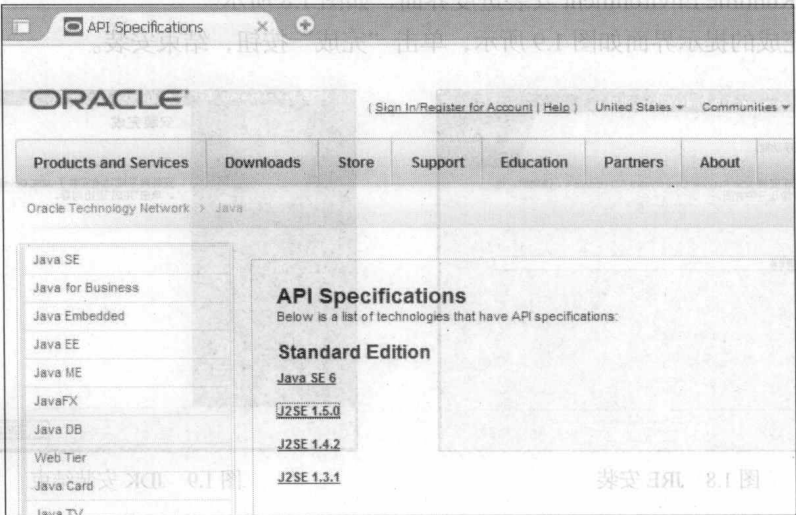


图 1.12 API 下载

1.1.2 环境变量和配置

在 Windows 平台下，经常设置的环境变量是 path 和 classpath，它们分别指定了 JDK 命令搜索路径和 Java 类路径。在这里假设 JDK 安装在 C:\Program Files\Java\jdk1.5.0_06 目录下，JDK 的所有命令都放在 C:\Program Files\Java\jdk1.5.0_06\bin 目录中。若使用 %path% 表示 path 环境变量的已有的当前的字符串取值，JDK 的路径放在 %path% 前边，以确保 JDK 提供的命令的使用，因为计算机中 system32 文件下也有一个 java.exe 程序。

设置环境变量 path 的作用是使 DOS 操作系统可以找到 JDK 命令。设置环境变量 classpath 的作用是告诉 Java 类装载器到哪里去寻找第三方提供的类和用户定义的共享类。在 classpath 环境变量中添加的 (.) 代表 Java 虚拟机运行时的当前工作目录。



安装 JDK 一般不需要设置环境变量 `classpath` 的值。如果读者的计算机安装过一些商业化的 Java 开发产品或带有 Java 技术的一些产品，这些产品所带的旧版本的类库，可能导致程序无法运行的情况。出现这种情况，编辑 `classpath` 的值，增加 JDK 文件中 `jre` 文件夹的 `lib` 文件夹中的 `rt.jar` 文件。

`path` 环境变量的作用是设置供操作系统去寻找和执行的应用程序的路径，也就是说，如果操作系统在当前目录下没有找到我们想要的命令工具时，它就会按照 `path` 环境变量指定的目录依次去查找，以最先找到的为准。`path` 环境变量可以存放多个路径，Windows 下路径和路径之间用英文分号 (;) 隔开。

平台为 Windows 2000/XP 时，右键单击桌面上的“我的电脑”，单击菜单中的“属性”命令，在出现的“系统属性”面板中选择“高级”标签，如图 1.13 所示；然后单击“环境变量”按钮，打开“环境变量”面板。在这里可以看到上下两个窗口：上面窗口为“某用户的环境变量”，下面窗口为“系统变量”，如图 1.14 所示。Win7 环境变量设置方法类似为：计算机→属性→高级系统设置→环境变量。

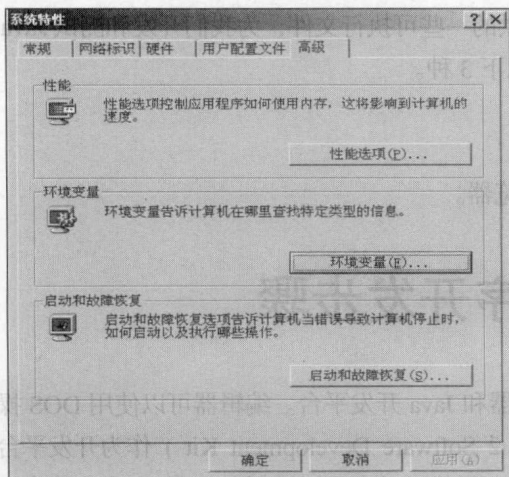


图 1.13 “系统属性”面板

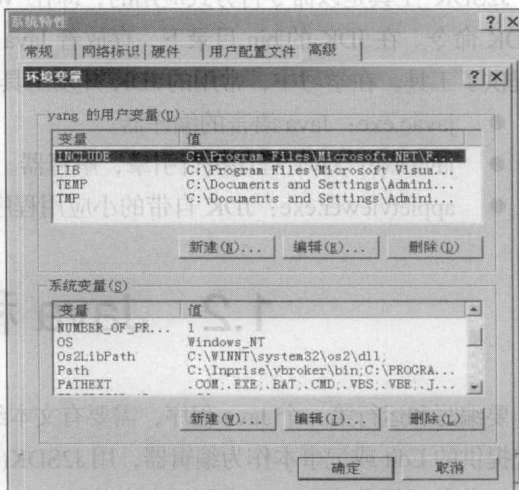


图 1.14 “环境变量”面板

可以在任意一个窗口进行设置，区别在于上面的窗口设置用于个人环境变量，只有以该用户身份登录时才有效，而下面窗口中的设置则对所有用户都有效。以设置系统变量为例，单击变量名为“Path”的变量（如果系统没有 `path` 环境变量选项，则在“用户变量”或“系统变量”中单击“新建”按钮来添加）后，单击“编辑”按钮。然后在打开的“编辑系统变量”对话框中的“变量值”输入框中加入 JDK 开发工具 `bin` 文件夹所在的路径（这里是 `C:\Program Files\Java\jdk1.5.0_06\bin`），这个目录包含 Java 编译器和解释器，路径末尾一定以英文分号 (;) 结尾，然后单击“确定”，如图 1.15 所示。注意：为了确保 JDK 的 `bin` 的正确搜寻，请把这个路径放在 `path` 变量值的最前边，别忘以英文分号结尾。

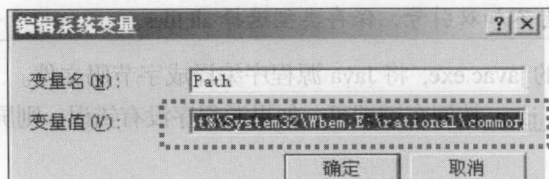


图 1.15 “编辑系统变量”对话框

path 环境变量设置好之后,需要重新打开 DOS 命令提示符界面,path 环境变量才起作用。接着,在 DOS 下用 cd 命令进入源程序所在的工作目录,程序就可以编译、运行了。如果偶尔有问题,可以进一步在 DOS 下,使用设置语句: set classpath=c:\myjava (myjava 为存放自己源代码的文件夹),再运行自己的源文件就可以了。



如果用户在安装 jdk1.5 时,选择了另外的 JDK 安装路径,则环境变量 path 和 classpath 要作相应的调整。

环境变量设置完成后,在 DOS 窗口下,输入 javac 并按回车键后,如果出现 javac 的用法参数提示信息,则安装正确。

1.1.3 JDK 开发工具简介

J2SDK 工具是以命令行方式应用的,即在 Windows 操作系统的 DOS 命令行提示符窗口中执行 J2SDK 命令。在 JDK 的 bin 目录下,存放着 Java 提供的一些可执行文件,为我们开发和测试 Java 程序提供了工具。在学习中,常用的 JDK 开发工具有以下 3 种。

- javac.exe: Java 语言的编译器。
- java.exe: Java 程序执行引擎,解释器。
- appletviewer.exe: JDK 自带的小应用程序浏览器。

1.2 Java 程序开发步骤

要编写和运行第一个 Java 程序,需要有文本编辑器和 Java 开发平台。编辑器可以使用 DOS 操作系统提供的 Edit 或记事本作为编辑器,用 J2SDK(Java 2 Software Development Kit)作为开发平台,J2SDK 往往习惯称为 JDK。也可以在安装好 JDK 之后,下载 TextPad 作为编辑和运行平台。通常,初学者使用 Windows 环境中的记事本作为创建源文件的文件编辑器。

要创建一个 Java 需要 3 个基本步骤。

- (1) 创建带有文件扩展名.java 的源文件。
- (2) 利用 Java 编译器生成文件扩展名为.class 的字节码文件。
- (3) Application 程序利用 Java 解释器运行该字节码文件,Applet 利用 Java 自带查看器或浏览器运行嵌有字节码文件的 HTML 文件。



保存文件时一定要使用 public 的类名作为文件名,用.java 作为后缀。记事本默认的扩展名是.txt,所以必须修改文件扩展名为.java,可在文件名的开始和扩展名的结尾处加上一对双引号后保存;或者不加双引号,保存类型选择 all files。

Java 编译器是 JDK 中的 javac.exe,将 Java 源程序编译成字节码文件。

使用语法: javac 类名.java 按回车键即可。如果源程序没有错误,则屏幕上没有输出,否则将显示出错误信息。

Java 解释器是 JDK 中的 java.exe,解释和执行 Java 应用程序。

使用语法：java 类名 按回车键即可。Java 的平台无关性就是因为每一种计算机上都安装了一个合适的解释器，将不同计算机上的系统差别隐藏起来，使字节码面对一个相同的运行环境，实现了“编写一次，到处运行”的目标。

对于 Applet 程序来说，需要 HTML 文件的配合。

使用语法：appletviewer HTML 文件名.html 按回车键即可。字节码文件嵌入 HTML 文件中，appletviewer 为 Applet 查看器（JDK 中的 appletviewer.exe），含有内置 Java 解释器。appletviewer 又称小浏览器，它仅显示相关 Applet 的属性，初学者使用很方便。

Java 源程序的开发步骤如图 1.16 所示，经过编辑、编译和运行的过程，JVM 执行的是 Java 字节码，操作系统可以是不同的操作系统，也就是所谓跨平台的特点。

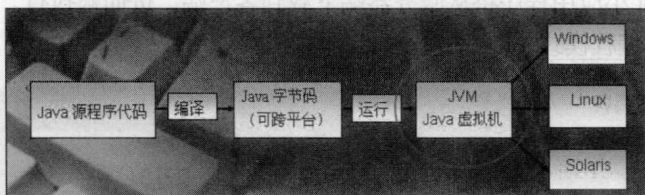


图 1.16 Java 程序开发过程

开发一个 Java 程序有 3 个步骤。

(1) 编辑。利用 Windows 的“记事本”，或者用使用其他文本编辑器编辑 Java 源程序文件，Java 源程序文件的扩展名一定为.java。

(2) 编译。打开 MS-DOS 窗口，首先通过 cd 进入源文件所在的盘符，这里 D 盘的“特色教材-source”文件夹为例，如图 1.17 所示。

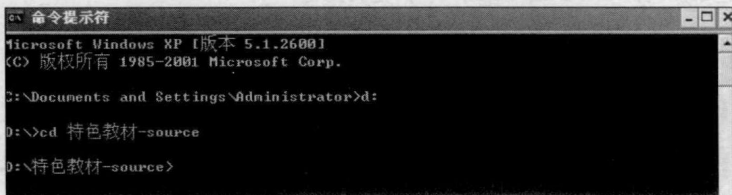


图 1.17 进入源文件所在目录

然后键入 javac 编译命令和 Java 源文件，中间用空格隔开，源文件的后缀保证是.java，文件名的命名一般第一个字母大写，不能以数字打头，其他命名规则后续章节会讲到。这里以 Welcome.java 为例，键入

```
javac Welcome.java
```

然后按回车键执行。编译之后生成的文件是以.class 为后缀的字节码文件。

(3) 运行。Java 基本程序有两种类型——Application 和 Applet。

①对于 Application，打开 MS-DOS 窗口，进入 Java 字节码文件所在目录，这里仍以 Welcome.java 为例，在命令行提示符下执行运行程序，格式为：`java Welcome`。



这里不能带 class 文件后缀，因为编译器默认的后缀已经是 class 了。class 字节码文件编译生成之后自动和源文件保存在同一级目录。

②对于 Applet，需要将 Applet 字节码文件加入到一个 HTML 文件中，然后在命令行状态下通过 JDK

自带的 AppletViewer 显示网页，执行之后就可以看到 Applet 的运行结果。或者用 IE 浏览器打开该网页，Applet 的运行结果同样会在浏览器窗口中显示出来。

1.3 Java 程序基本结构

用 Java 书写的程序有两种类型：Java 应用程序（Java Application）和 Java 小应用程序（Java Applet）。Java 应用程序必须得到 Java 虚拟机的支持才能够运行。Java 小应用程序则需要客户端浏览器的支持。Java 小应用程序运行之前必须先将其嵌入 HTML 文件的 <applet> 和 </applet> 标记中。当用户浏览该 HTML 页面时，Java 小应用程序将从服务器端下载到客户端，进而被执行。

Application 的基本编程模式：

```
class 用户自定义的类名 // 定义类
{
    public static void main(String args[ ]) // 定义 main() 方法
    {
        方法体
    }
}
```

Applet 的基本编程模式：

```
import java.awt.Graphics; // 引入 java.awt 系统包中的 Graphics 类
import java.applet.Applet; // 引入 java.applet 系统包中的 Applet 类
class 用户自定义的类名 extends Applet // 定义类
{
    public void paint(Graphics g) // 调用 Applet 类的 paint() 方法
    {
        方法体
    }
}
```

Applet 需要的 HTML 文件的最小集的格式为：

```
<HTML>
<applet code=类名.class width=宽度 height=高度>
</THML>
```



HTML 标记包含在尖括号内，并且总是成对出现，前面加斜杠表明标记结束。<HTML> 和 </THML> 来标记 HTML 文件的开始和结束，用 <applet> 和 </applet> 标记 applet 的开始和结束。必须把以 .class 结尾的字节码文件名嵌入到 HTML 文件中。HTML 文件应和字节码文件放在同一目录下。另外，HTML 对字符大小写是不敏感的，参数值可加引号也可不加。

综上所述，Applet 和 Application 是 Java 程序的两种基本类型，从源代码的角度来看，Applet 和 Application 有两个基本的不同点。

- （1）一个 Applet 类必须定义一个从 Applet 类派生的类，Application 则没有这个必要；
- （2）一个 Application 必须定义一个包含 main 的方法，以控制它的执行，即程序的入口；而 Applet

不会用到 main 方法，它的执行是由 Applet 类中的几个系统方法来控制的。两者共同之处是：编程语法是完全一样的。

1.4 良好的编程习惯

- (1) 所有的 Java 语句必须以英文分号“;”结束。
- (2) Java 区分大小写，拼写时要注意关键字和标识符构成字母的大小写。
- (3) 花括号成对出现。在写左花括号时，立即再写一个右花括号，这样有助于防止漏写右花括号。类名称后面的花括号标示着类定义的开始和结束。
- (4) 习惯上，类名应以首字母大写开头，变量以小写字母开头，变量名有多个单词的第一个单词后边的每个单词首字母应大写。当读一个 Java 程序时，寻找以大写字母开头的标识符，这些通常代表 Java 类。
- (5) 程序段中适当增加空白行会增加程序的可读性。在定义方法内容的花括号中，将整个内容部分缩进一层，使程序结构清晰，程序易读。编译器会忽略这些空白行和空格字符。
- (6) 在程序中，一行最好只写一条语句。Java 允许一个长句分割写在几行中，但是不允许从标识符或字符串的中间分割。
- (7) 文件名与 public 类名在拼写及大小写上必须保持一致。
- (8) 如果一个 .java 文件含有多于一个 public 类，则是一个错误。
- (9) 不以 .java 为扩展名的文件名是一个错误。
- (10) 运行 appletviewer 时，文件扩展名不是 .htm 或 .html 是一个错误，这将导致无法使 appletviewer 装载 Applet。

1.5 实验目的

- (1) 熟悉 JDK 开发工具的下载、安装和环境变量的设置。
- (2) 通过实验，详细了解 Application 程序和 Applet 程序的结构、编译和运行。
- (3) 培养良好的 Java 编程习惯。

1.6 实验内容

实验 1 第一个 Java 应用程序

实验题目：实现第一个简单的应用程序：打印一行自己喜欢的字符串。

【实验指导】

首先，用户需要下载和安装 J2SDK (JDK)。以 JDK 1_5_0_06 版本为例，暂且把程序源文件放置在 JDK 的 bin 目录之下自己创建的 code 文件夹中。

其次, 确定文本编辑器。在本例中, 使用记事本。以 Windows 2000/XP 为例, 从“开始”菜单项中选择“程序”→“附件”→“记事本”。当然, 用户也可以选择其他文本编辑器。



提示

在编译器 javac 运行正常, 解释器 java 不能正常运行时, 且提示的异常为: Exception in thread "main" java.lang.NoClassDefFoundError: Welcome, 其中 Welcome 是程序的主类名称, 此时, 请一定在“我的电脑”中打开 classpath, 把英文实心点和分号“;”添加到其变量值中。“;”表示加载应用程序当前目录及其在子目录中的类。



注意

在你的计算机上编辑、保存和运行一个 Application 和 Applet 程序, 注意保存时的文件名必须是程序中的类名, 源文件中有多类, 只能有一个 public 类。如果有一个 public 类, 则源文件的名字必须和这个 public 类完全相同; 如果源文件中没有 public 类, 那么源文件的名字只要和某个类相同, 后缀必须是 .java, 文件名和类名大小写都必须一致。

【实验步骤】

(1) 在“记事本”中编写如下源程序:

```
// 文件名: Welcome.java
public class Welcome {
    public static void main( String args[] )
    {
        System.out.println( "Welcome to Java Programming!" );
    } //结束main方法的定义
} //结束类Welcome的定义
```

(2) 语法说明。

程序中的“//”为单行注释符, 只对当前行有效, 表示该行是注释行。程序人员在程序中加入注释, 用于提高程序的可读性, 使程序便于阅读和理解。程序执行时注释行会被 Java 编译器忽略。多行注释用“/*”开始, 以“*/”结束。

Java 程序是由类或类的定义组成的, 类构成了 Java 程序的基本单元。创建一个类是 Java 程序的首要工作。Java 用关键字 class 标志一个类定义的开始, class 前面的 public 关键字代表该类的访问属性是公共的, 表示这个类在所有场合中可使用。一个程序文件中可以声明多个类, 但仅允许有一个公共的类, 程序文件名要与公共类的名称相同, 包括字母的大小写。class 后面是该类的类名, 在本例中是 Welcome。

Application 中有一个显著标记就是必须定义一个 main() 主方法, 而且应该按照源程序中所示来定义其修饰符和命令行参数, 用关键字说明它是 public, 静态的 static, 无返回值的 void, 主方法的参数是字符串类型 String 的数组 args[]。一个类中可以声明多个方法, Java 应用程序自动从 main 主方法开始运行, 通过主方法再调用其他的方法。Java 语言的每条语句都必须用分号结束。

System.out 是标准输出对象, 它用于在 Java 应用程序执行的过程中向命令窗口显示字符串和其他类型的信息。方法 System.out.println 在命令窗口中显示一行文字后, 会自动将光标位置移到下一行 (与在文本编辑器中按 Enter 键类似)。

(3) 编译运行程序。

源程序编写并保存好之后, 接下来准备执行该程序。为此, 打开一个命令提示符窗口, 用 cd 退到根目录, 如图 1.18 所示。