

高职高专计算机规划教材 · 任务教程系列

Java 程序设计

韦鹏程 石 熙 肖 丽 主编

任务驱动



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机规划教材·任务教程系列

Java 程序设计

主 编 韦鹏程 石 熙 肖 丽
副主编 张 伟 吴海霞 杨华千
姚富光 邹晓兵

内 容 简 介

本书采用任务驱动教学模式,通过任务的实施,使读者在读程序、学知识、写程序的过程中,逐渐掌握面向对象的 Java 程序设计思想和技能。本书共分 12 个单元,主要包括 Java 程序设计过程、基本语法规则、面向对象技术、数组与字符串、异常处理、GUI 编程、输入/输出处理、多线程编程以及基础网络编程等内容。

本书适合作为高等职业院校计算机相关专业“Java 程序设计”或者“面向对象程序设计”课程的教材,也可作为相关技术人员学习 Java 的参考用书。

图书在版编目(CIP)数据

Java 程序设计 / 韦鹏程, 石熙, 肖丽主编. —北京:
中国铁道出版社, 2011. 2
高职高专计算机规划教材·任务教程系列
ISBN 978-7-113-12482-3

I. ①J… II. ①韦… ②石… ③肖… III. ①
JAVA 语言—程序设计—高等学校: 技术学校—教材 IV.
①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 009206 号

书 名: Java 程序设计

作 者: 韦鹏程 石 熙 肖 丽 主编

策划编辑: 翟玉峰 赵 鑫

责任编辑: 翟玉峰

读者热线电话: 400-668-0820

封面制作: 白 雪

版式设计: 于 洋

责任印制: 李 佳

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 河北新华第二印刷有限责任公司

版 次: 2011 年 2 月第 1 版

2011 年 2 月第 1 次印刷

开 本: 787mm × 1092mm

1/16

印张: 14 5

字数: 342 千

书 号: ISBN 978-7-113-12482-3

定 价: 24.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有印制质量问题, 请与本社计算机图书批销部联系调换。

Java 语言是一种可以用来编写跨平台应用程序的面向对象的程序设计语言。Java 诞生于 20 世纪 90 年代初期,伴随着计算机平台的多样化以及互联网的迅猛发展,逐渐成为主流的网络编程语言。

Java 是一种完全的面向对象的程序设计语言。它将“对象”作为程序的基本单元,将程序和数据封装其中,从而提高软件的重用性、灵活性和扩展性。面向对象程序设计可以被视为一种在程序中包含各种独立而又可互相调用的单位和对象的思想,这与传统的编程思想刚好相反——传统的程序设计主张将程序看做一系列函数的集合,或者直接就是一系列对计算机下达的指令。

本书主要侧重于培养读者使用 Java 语言进行面向对象程序设计的思想和技能,而并非对 Java 技术进行百科全书式的介绍。将 Java 语言作为面向对象程序设计的教学语言,可以最简单最快速地培养读者面向对象编程的思想。而这种现代的编程思想,无论对于读者真正进行实际编程工作或者学习其他程序设计语言都至关重要。

本书的主要特色如下:

(1) “任务驱动”教学模式

本书彻底地采用先进的“任务驱动”教学模式,每个章节由若干编程任务组成,通过任务的实施过程,掌握程序设计思想和技能。每个任务的实施,就是一个读程序、学知识、写程序的过程。

(2) 内容选取适度够用,任务程序经典

本书尽可能避免对 Java 技术的贪多求全,在内容选取上做到了适度、够用,任务程序也选用经典的,重视思想与方法的传授,而不是知识的罗列。同时,还紧密联系 Java 语言的发展状况,对于 Applet 等已经逐渐退出历史舞台的技术给予淡化。

(3) 强调“面向对象”的编程思想

Java 是最适合的面向对象技术的教学语言。本书的绝大部分内容,无论是简单的类与对象的介绍,还是后期的 GUI 图形界面编程,都紧扣面向对象这个主题,在程序设计的各个环节都体现出这种思想。本书能够使读者逐渐体会并掌握这种编程思想,为进一步深入学习打下基础。

本书共分 12 单元。单元一讲解了 Java 语言的开发过程,并编写了一些最简单的 Java 程序。单元二则简要地介绍了 Java 的基本语法规则,这对于有程序设计基础的读者非常容易。单元三到单元五,由浅入深地分析了面向对象技术的方方面面,使读者有了一个最基本的面向对象编程思想。单元六讲解 Java 中数组与字符串的使用。单元七介绍了 Java 的异常处理机制。单元八简要介绍了已经不再流行的 Java Applet 小程序。单元九为 Java GUI 图形界面编程,着重强调 Java 的事件处理机制,以及查询 API 的编程方法。单元十介绍了 Java 独特的输入输出方式。单元十一介绍了多线程编程。单元十二介绍了最基础的 Java 网络编程。

作为高等职业院校相关专业的教材,要求读者应该具备基本的程序设计知识,如一定的 C 语言编程基础。而在本课程基础之上,还可以进一步学习 JSP 网络编程、JBuilder 数据库编程或者移动应用开发技术等。

本书所有程序均在 JDK6 环境下调试通过。由于时间紧迫,加上编者水平有限,如有不足之处,请广大读者批评指正。

编 者

2010 年 11 月

单元一 最简单的 Java 程序 1

任务一 编写第一个 Java 程序 1



任务实施

1. 搭建 Java 开发环境 1
2. 第一个 Java 程序 5



相关知识

1. Java 语言简介 5
2. Java Application 的开发 6

任务二 带命令行输入参数的 Java 程序.. 7



任务实施

1. 编写源程序 8
2. 编译和运行程序 8



相关知识

1. 命令行参数 8

任务三 简单的输入/输出处理 9



任务实施

1. 编写源程序 9
2. 编译和运行程序 10



相关知识

1. import 语句 11
2. 输入/输出语句 11

拓展任务 12

单元二 Java 基本语法规则 13

任务一 数据类型转换 13



任务实施

1. 编写源程序 13
2. 编译、运行及调试 14



相关知识

1. 标识符和关键字 15
2. 数据类型 16

任务二 输出小于 N 的所有素数 18



任务实施

1. 编写源程序 19
2. 编译和运行程序 19
3. 改进程序 19



相关知识

1. 运算符与表达式 20
2. 程序控制语句 21
3. 数学函数 21

任务三 猜数字小游戏 22



任务实施

1. 编写源程序 23
2. 编译和运行程序 24



相关知识

1. 字符串转换类型 24
2. Java 程序的基本结构 25

拓展任务 25

单元三 认识类与对象 27

任务一 用程序描述学生特性 27



任务实施

1. 声明学生类 27
2. 创建学生类的对象 29
3. 使用对象测试程序 29
4. 使用构造方法 30



相关知识

1. 面向对象编程 32

- 2. 类与对象 32
- 3. 对象的内存存储 33

任务二 计算两点之间的距离 35



任务实施

- 1. 建立点类 36
- 2. 改进构造方法 36
- 3. 编译和运行程序 37
- 4. 统计创建点的数量 38



相关知识

- 1. 方法重载 40
- 2. this 关键字 41
- 3. static 关键字 41

任务三 复数的加法运算 42



任务实施

- 1. 设计复数类 42
- 2. 编译和运行程序 43
- 3. 使用访问控制修饰符 43



相关知识

- 1. 封装 46
- 2. 访问控制 46

拓展任务 47

单元四 类的继承 48

任务一 用程序描述员工与经理 48



任务实施

- 1. 设计员工类以及经理类 48
- 2. 使用继承机制 50
- 3. 重写方法 52



相关知识

- 1. 继承与“is-a”关系 53
- 2. 重写 54

任务二 计算空间中两点之间的距离 55



任务实施

- 1. 继承点类 55

- 2. 测试空间点类 57
- 3. 计算 XY 平面上投影的距离 58



相关知识

- 1. 区别重载和重写 59
- 2. 父类引用子类对象 60

任务三 用程序描述不同的学生 61



任务实施

- 1. 建立大学生与研究生类 61
- 2. 测试学生类 63



相关知识

- 1. Java 的单继承机制 63
- 2. final 关键字 64

任务四 计算各种形状的面积 64



任务实施

- 1. 建立形状类 64
- 2. 测试形状类 66
- 3. 抽象的形状类 66
- 4. 体现多态性 67



相关知识

- abstract 关键字 69

拓展任务 70

单元五 接口与包 71

任务一 计算各种形状的面积与周长 71



任务实施

- 1. 改写抽象类 71
- 2. 增加计算周长的接口 72
- 3. 更灵活地使用接口 74



相关知识

- 1. 接口的使用 75
- 2. 接口的继承 77

任务二 用包管理程序 77



任务实施

- 1. 使用简单的包结构 78

- 2. 使用层次的包结构 79
- 3. 导入包扩展程序 81



相关知识

- 1. 包的定义与使用 83
- 2. 访问权限 84
- 3. Java API 包 84

拓展任务 85

单元六 数组与字符串 86

任务一 数组排序 86



任务实施

- 1. 编写源程序 86
- 2. 编译和运行程序 87



相关知识

- 1. 一维数组的创建 87
- 2. 一维数组元素的引用 88

任务二 矩阵运算 89



任务实施

- 1. 编写矩阵类 89
- 2. 测试矩阵类 90



相关知识

- 1. 二维数组的声明 90
- 2. 二维数组内存分配 91
- 3. 二维数组的初始化 93
- 4. 二维数组元素的引用 93

任务三 诗词录入 93



任务实施

- 1. 编写源程序 94
- 2. 编译和运行程序 95



相关知识

- ArrayList 类 95

任务四 统计单词数目 97



任务实施

- 1. 使用 StringBuffer 类统计单词数 97

2. 使用 StringTokenizer 类改进程序 98



相关知识

- 1. String 类 99
- 2. StringBuffer 类 101
- 3. StringTokenizer 类 103

拓展任务 104

单元七 异常处理 106

任务一 捕获异常 106



任务实施

- 1. 未进行异常处理的程序 106
- 2. 捕获异常 108



相关知识

- 1. 异常的基本概念 109
- 2. 异常捕获处理 109

任务二 声明异常 110



任务实施

- 1. 使用 throws 子句声明异常 111
- 2. 编译和运行程序 111



相关知识

- 1. throws 子句 112
- 2. 异常的声明和捕获处理 112

任务三 抛出异常和自定义异常 113



任务实施

- 1. throw 语句抛出异常 113
- 2. 自定义员工年龄异常类 114



相关知识

- 1. 抛出异常对象 115
- 2. 自定义异常类 116

拓展任务 118

单元八 Java Applet 程序 119

任务一 编写 Hello Applet 的网页 119



任务实施

1. 编写 Applet..... 119
2. 创建 HTML 网页文件..... 120
3. 运行 Applet..... 120



相关知识

1. Applet 程序的创建..... 122
2. Applet 程序的生命周期..... 123
3. HTML Applet 标记..... 126

任务二 在 Applet 中绘制图形..... 127



任务实施

1. 绘制简单图形的 Applet..... 127
2. 给 Applet 添加背景音乐..... 129
3. 给 Applet 添加动画..... 129



相关知识

1. Applet 与图形绘制有关的方法..... 131
2. Applet 的图形绘制..... 132
3. Applet 的图像..... 133

拓展任务..... 134

单元九 图形用户界面编程..... 135

任务一 统计按钮单击次数..... 135



任务实施

1. 编程实现界面..... 135
2. 编写事件代码..... 137



相关知识

1. GUI 编程基础..... 139
2. 事件处理模型..... 139
3. 常用组件与容器..... 140

任务二 统计计算器..... 142



任务实施

1. 编程实现界面..... 142
2. 使用 ActionListener 监听器..... 146
3. 使用 ItemListener 监听器..... 148



相关知识

1. 布局管理..... 149
2. 常用组件与容器..... 152

任务三 绘制常见形状..... 157



任务实施

1. 使用鼠标事件编程..... 157
2. 使用键盘事件编程..... 159



相关知识

1. 事件类型..... 162
2. 事件适配器..... 163

拓展任务..... 165

单元十 文件与输入/输出..... 166

任务一 复制文本文件..... 166



任务实施

1. 实现字节流文件输入..... 166
2. 实现字节流文件输出..... 168
3. 实现字符流文件输入/输出..... 169



相关知识

1. File 类..... 169
2. 字节流与字符流..... 170

任务二 简单文本编辑器..... 174



任务实施

1. 编程实现界面..... 174
2. 编程实现功能..... 177
3. 编译运行程序..... 179



相关知识

- JFileChooser..... 181

拓展任务..... 182

单元十一 线程..... 183

任务一 简易时钟..... 183

	任务实施
1. 通过实现 Runnable 接口创建线程 ...	184
2. 通过继承 Thread 类创建线程	185

	相关知识
1. Thread 类和 Runnable 接口	187
2. 主线程	188
3. 线程的生命周期	189

任务二 工作与休息	190
------------------------	-----

	任务实施
1. 编写源程序	191
2. 运行程序	191

	相关知识
1. isAlive()和 join()方法	192
2. interrupt()方法	193
3. 控制线程状态的其他方法	194

任务三 ATM 机的存款与提款	195
------------------------------	-----

	任务实施
1. 不采用线程同步的 ATM 机	195
2. 采用线程同步的 ATM 机	198

	相关知识
1. 使用同步方法	199
2. 使用同步语句	200

拓展任务	201
-------------------	-----

单元十二 网络编程	203
------------------------	-----

任务一 编写简单聊天系统	203
---------------------------	-----

	任务实施
1. 实现服务器端	203
2. 实现客户端	207
3. 运行程序	209
4. 改进服务器端	211

	相关知识
1. 网络基础	211
2. Socket 编程步骤	212
3. ServerSocket 和 Socket	212
4. URL 类	213

拓展任务	214
-------------------	-----

附录 A 常用字符 Unicode 表	215
----------------------------------	-----

附录 B Java 语言关键字表	216
-------------------------------	-----

附录 C KeyEvent 类中定义的 键码表	217
--	-----

参考文献	219
-------------------	-----

单元 一

最简单的 Java 程序

Java 程序的开发最简单的方式是安装 JDK 并配置环境变量，搭建开发环境。Java 程序代码编写的规范也与 C 语言等有很大的区别，如严格区分大小写。对于大多数学生来说，Java 很可能是第一次接触的面向对象的程序设计语言。因此，本项目将首先介绍 Java 开发环境的搭建，再通过几个最简单的程序，介绍 Java 程序的编写、编译和运行的方法，尤其是简单的数据输入/输出的方法。

学习目标

- 掌握 Java 开发环境的搭建
- 了解 Java 程序代码编写的基本规范
- 掌握 Java 程序的编译和运行方法
- 熟悉数据输入/输出的基本方法

任务一 编写第一个 Java 程序

任务描述

Java 程序分成 Applet 和 Application 两大类。本任务将编写一个最简单的 Java Application 程序，即 Hello World 程序。在此任务中，我们首先搭建 Java 的开发环境，然后将详细介绍 Java Application 程序编码、编译和运行的完整过程。

任务实施

1. 搭建 Java 开发环境

了解 Java 的开发运行环境是使用 Java 语言编程的第一步，也是学好 Java 语言的第一步。目前已经有很多 Java 开发环境，但是对于初学者而言，SUN 公司提供的免费的 Java 开发和运行环境——JDK 无疑最为简单易用。JDK 的全称为 Java Development's Kit，即 Java 开发工具包，主要包括编译器、JVM（Java 虚拟机）和 Java API（Java 应用编程接口）等。

（1）JDK 的下载

SUN 公司已于 2009 年被 ORACLE（甲骨文公司）收购，但仍然不断地更新 JDK 的版本，当前最新的版本为 JDK 1.6，或者直接称为 JDK6。可以登录以下网址下载 JDK：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

我们需要根据自己的操作系统，下载最适合的 JDK。以最常见的 Windows 操作系统为例，本书稿写作时，下载的 JDK 安装文件名为 `jdk-6u17-windows-i586.exe`，以后均以此版本为例进行讲解。其中，6u17 代表了当前下载的版本为 JDK6 的第 17 次更新。

但是要注意的是，JDK 的版本会不断进行更新。读者可以通过各种搜索引擎如 Google 查找最新的 JDK 版本，并进行下载。

(2) JDK 的安装

① 直接双击下载的 JDK 安装文件 `jdk-6u17-windows-i586.exe`，即可进入安装界面，如图 1-1 所示。

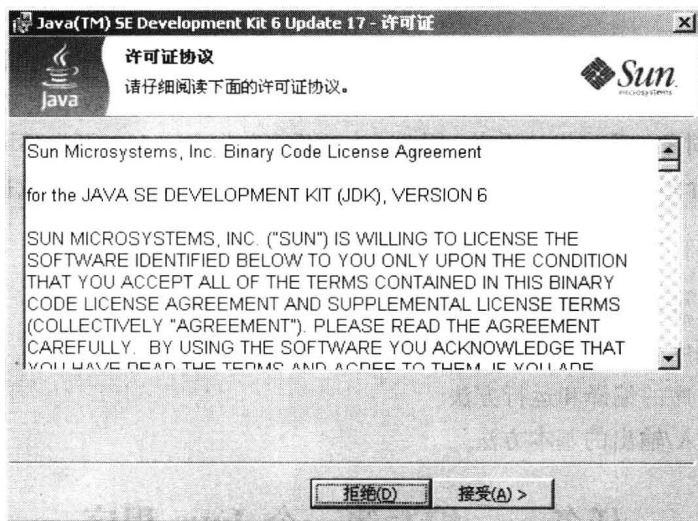


图 1-1 JDK 安装界面

② 单击【接受】按钮接受许可证协议，即可进入 JDK 的安装。

③ 单击【更改】按钮，选择安装目录，在此选择 `D:\jdk1.6\`，如图 1-2 所示。

④ 自定义安装程序的功能，除“源代码”外，建议安装其他全部功能，如图 1-2 所示。

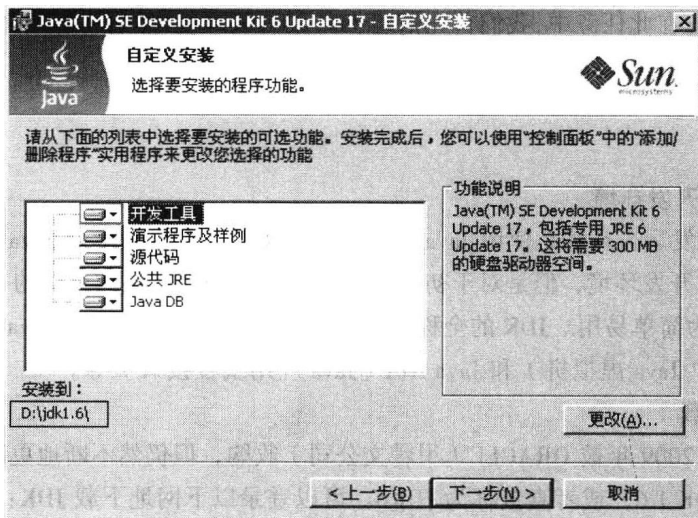


图 1-2 选择 JDK 安装目录

⑤ 单击【下一步】按钮进行安装，在安装过程中，会弹出 JRE（Java 运行环境）的安装提示，因为 JDK 已经包含 JRE，所以可以取消 JRE 的安装。

⑥ 弹出如图 1-3 所示的对话框，单击【完成】按钮，即可完成 JDK 的安装。

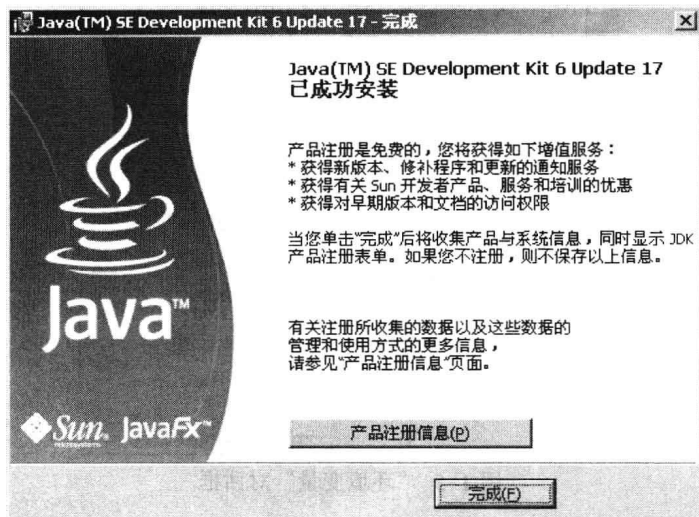


图 1-3 JDK 成功安装

（3）配置环境变量

JDK 安装完成后，还需要对环境变量进行设置。

右击“我的电脑”图标，在弹出的快捷菜单中选择“属性”命令，打开“系统属性”对话框，选择“高级”选项卡，如图 1-4 所示。单击【环境变量】按钮，打开如图 1-5 所示的“环境变量”对话框。

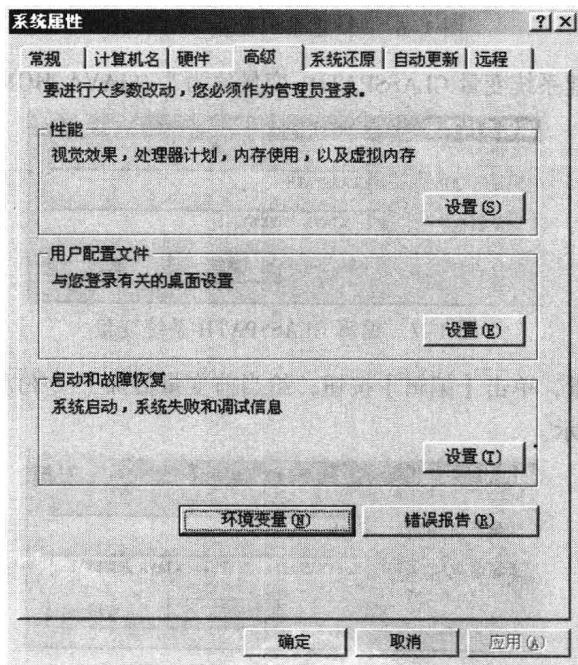


图 1-4 “系统属性”对话框

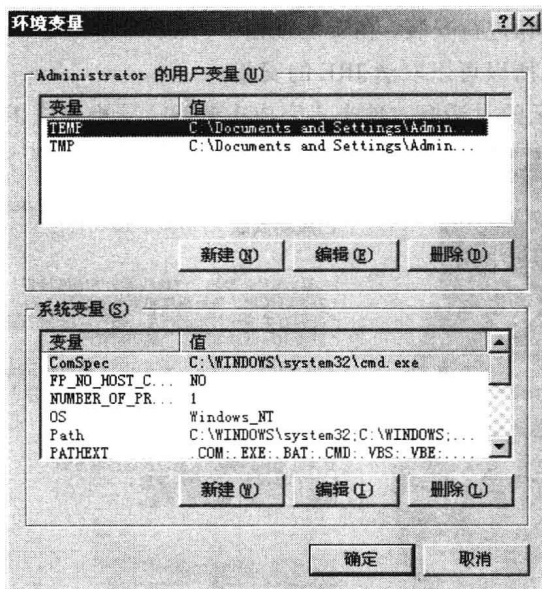


图 1-5 “环境变量”对话框

在“环境变量”对话框中单击系统变量中的【新建】按钮，在弹出的“新建系统变量”对话框中，输入变量名“JAVA_HOME”，变量值“D:\jdk1.6”，即 JDK 的安装目录，如图 1-6 所示。

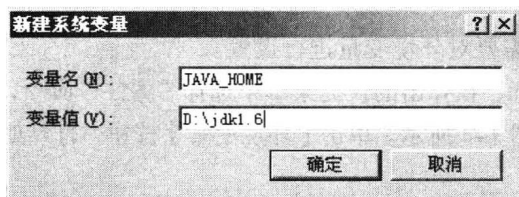


图 1-6 “新建系统变量”对话框

按照同样的方法新建系统变量 CLASSPATH，变量值为“.;%JAVA_HOME%\lib”，如图 1-7 所示。

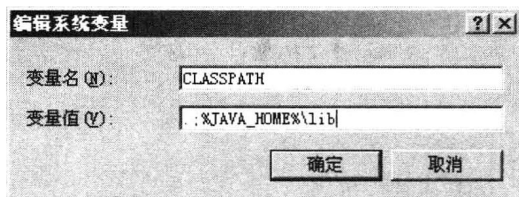


图 1-7 编辑 CLASSPATH 系统变量

选择 Path 系统变量，单击【编辑】按钮，给当前变量值加上“%JAVA_HOME%\bin”，并用分号隔开，如图 1-8 所示。

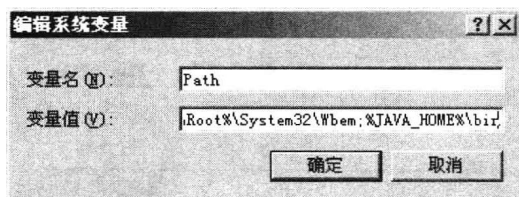


图 1-8 编辑 Path 系统变量

最后单击【确定】按钮，完成环境变量的配置。

2. 第一个 Java 程序

(1) 编写源代码

打开记事本，输入以下程序代码：

程序 1.1 HelloWorld:

```
public class HelloWorld                                //定义一个类，名为 HelloWorld
{
    public static void main(String args[])            //main 是类的主方法
    {
        System.out.println("Hello World ");          //在命令行下输出 Hello World
    }
}
```

一个 Java Application 由若干个类组成，程序 1.1 只有一个类，类名为 HelloWorld。最外层的大括号{}以及括号内的内容就是类体。main()是类 HelloWorld 的一个方法，并且是主方法。一个 Java Application 必须有一个类，也只能有一个类包含 main()方法，这个类就是程序的主类。如程序 1.1，类 HelloWorld 就是主类。程序的执行是从 main()方法开始的，其格式是确定不变的：

```
public static void main(String args[])
```

Java 源程序命名有严格的限制，源文件的扩展名为“.java”。源文件中只能有一个 public 修饰的类，源文件的名称必须和这个 public 的类的名称一致。因此，程序 1.1 在编辑完成之后，应该保存成“HelloWorld.java”。

在此，我们建议，读者应创建一个专用目录来保存源程序，例如 D:\java，然后把源程序文件保存到这个文件夹中。

(2) 编译源程序

Java Application 需要先将源文件编译成字节码文件，才能被 Java 解释器解释运行。

选择“开始”→“运行”命令，在打开的“运行”对话框中，输入“cmd”，单击“确定”按钮，进入命令提示符窗口。将命令提示符的当前路径切换到程序源文件所在目录，如 D:\java，方法如下：输入“D:”，按【Enter】键；再输入“cd java”，按【Enter】键即可。

输入以下命令对“HelloWorld.java”进行编译：

```
javac HelloWorld.java
```

可执行程序 javac.exe 是 Java 编译器，用于对 Java 源文件进行编译。编译成功后，在 D:\java 目录下将会出现一个名为 HelloWorld.class 的文件，这就是编译后得到的字节码文件。

(3) 运行程序

现在就可以使用 JDK 的解释器 java.exe 对编译后得到的字节码文件进行解释运行：

```
Java HelloWorld
```

运行结果如图 1-9 所示。

相关知识

1. Java 语言简介

Java 是一种可以跨平台的面向对象的程序设计语言，由 SUN 公司的 James Gosling 等人于 20 世纪 90 年代初开发。最初的 Java 语言被命名为 Oak，用来解决一些家用电器如电视机、电话等

的控制和通信问题。Oak 并没有获得家电市场的认可，但随着互联网的发展，Oak 在计算机网络上得到了一定的应用。于是，SUN 公司改造了 Oak，并于 1995 年 5 月正式发布了 Java 语言。

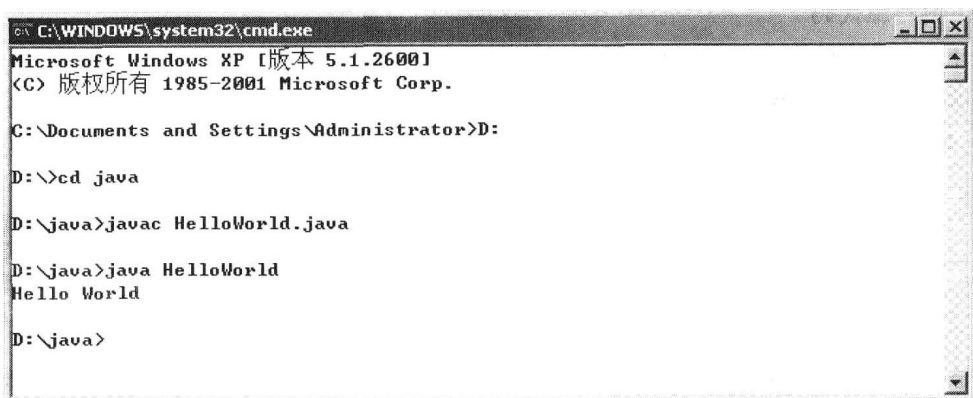


图 1-9 程序运行结果

Java 编程语言是一种简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言，这些优点将在本书后面的单元中逐渐体现。Java 伴随着互联网的迅猛发展而发展，逐渐成为重要的网络编程语言。

1999 年 6 月，SUN 公司发布 Java 的三个版本：标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。2005 年 6 月，这三个版本被更名为 Java SE、Java EE 和 Java ME。本书所有的关于 Java 语言的讲解都是基于 Java SE 的。

2. Java Application 的开发

(1) 开发流程

Java Application 程序的开发过程如图 1-10 所示。首先编写 Java 源程序，即保存为“`.java`”的文件；源程序经过 Java 编译器编译后产生字节码文件，即“`.class`”文件；最后通过 Java 解释器解释运行字节代码文件。

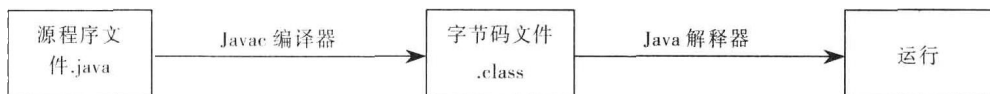


图 1-10 Java 应用程序开发流程

编译器 `javac.exe` 和解释器 `java.exe` 都安装在 JDK 的安装目录中的 `bin` 文件夹中，这个子目录可以说是 Java JDK 的核心。

跨平台性可以说是 Java 语言最优秀的特性之一，这个特性经常被称为“一次编译，到处运行”。Java 源程序经过编译，形成了一个半成品——字节码。凡是安装了 Java 运行环境(JRE)的平台，无论是 Windows、Linux 或者 Mac OSX，都能执行这个字节码。JRE 内部有一个 Java 虚拟机(JVM)以及一些标准的类库(class library)。

(2) Java 编译器 `javac` 的使用

在任务一中，编译源程序文件采用的方法是首先进入源文件所在的目录，再直接使用 `javac` 命令进行编译。例如，程序 1.1 源文件保存在 `D:\java` 中，则先进入此目录，再直接输入：

javac 源文件名
进行编译。

事实上，也可以打开命令提示符，用以下方法进行编译：

javac 带完整路径的源文件名

以程序 1.1 为例，打开命令提示符，直接输入：

javac D:\java\HelloWorld.java

也可以进行编译。

(3) Java 程序编写的基本规范

对于初学者而言，在编写 Java 源程序时，有一些基本规则必须记牢：

- ① Java 语言严格区分大小写，如 System 和 system 是不同的。
- ② 每一条语句必须以分号结束。
- ③ 大括号{}用于构成一个语句块，总是成对出现。

(4) Java 程序的注释

注释是程序中的说明文字，用于帮助阅读程序。注释不是语句，不会影响程序的运行和效率。

Java 语言的注释有三种形式：

单行注释：以“//”开始到本行结束的内容都是注释。例如：

//这是一行单行注释

块注释：在“/*”和“*/”之间的所有内容都是注释。例如：

/*这是一个块注释

它可以分布在多行之中*/

文档注释：在“/**”和“*/”之间的内容都是文档注释。可以通过 JDK 提供的 javadoc 命令，提取程序的文档注释，生成程序的 HTML 文档。

任务小结

本任务首先介绍了 Java 开发环境 JDK 的安装过程，这是编写 Java 程序的必要条件。然后，本任务通过一个最简单的 HelloWorld 程序实例介绍了 Java 程序的基本编写规范，讲述了 Java Application 的编写、编译和运行的方法。本任务是学习 Java 语言的基础，很多概念也许并不是特别清楚，但首先应学会的就是用 Java 语言编写代码和进行编译的方式。在今后的学习中要注意多动手编程，逐渐理解 Java 语言的各种特点。

任务二 带命令行输入参数的 Java 程序

任务描述

在使用解释器 java.exe 对字节码文件进行解释运行的时候，可以带上命令行参数（command-line arguments）实现对 main() 方法的参数传递，这也是最简单的输入参数的方式。所谓的命令行参数就是程序执行时在命令行中紧跟在程序名后的信息，这个参数本质上是一个字符串数组。在该任务中，我们将编写一个简单的程序，实现将两个字符串参数交换次序后再输出的功能。

任务实施

1. 编写源程序

打开记事本，输入以下代码，将程序以 `Exchange.java` 为文件名保存到 `D:\java` 下。

程序 1.2 Exchange:

```
public class Exchange
{
    public static void main(String args[])
    {
        System.out.println(args[1]+ "!" +args[0]); /*将命令行第二个参数和第一个参数连接后输出，中间用“!”间隔*/
    }
}
```

2. 编译和运行程序

在命令提示符窗口下将当前路径切换到 `D:\java`，先进行编译：

```
javac Exchange.java
```

使用以下命令传递参数：

```
java Exchange Hello World
```

`Hello` 是第一个参数，将传递给 `args[0]`；`World` 为第二个参数，将传递给 `args[1]`；中间用空格隔开。

运行结果如图 1-11 所示。先输出第二个参数，再输出第一个参数，中间有“!”间隔。

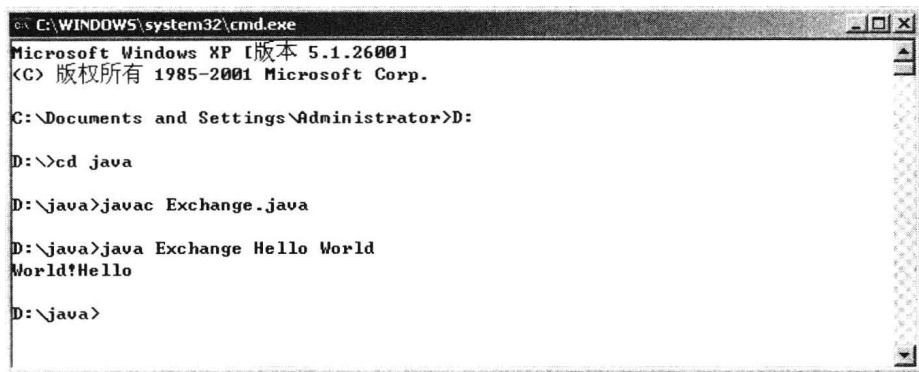


图 1-11 程序运行结果

相关知识

命令行参数

使用带命令行参数的运行方式时，参数依次传递给 `main()` 方法的形式参数 `args`。`args` 为一个字符串数组，和其他程序设计语言一样，对数组元素的引用通过下标来实现，下标从 0 开始。关于数组，我们将在后续内容中详细介绍。

值得注意的是，所有的命令行参数都是以字符串的形式传递的。以程序 1.3 为例，这个程序将输出所有的命令行参数。