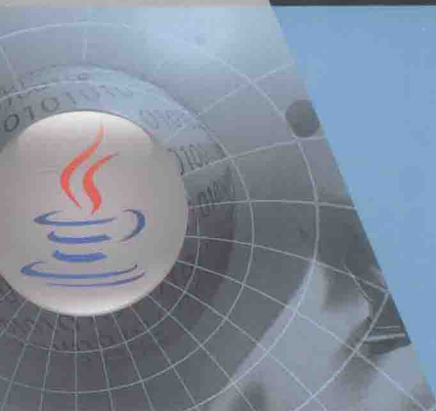


计算机类主干课程系列教材

面向对象程序设计 Java

Java

©周 斌 王新颖 王继鹏 主编



科学出版社

计算机类主干课程系列教材

面向对象程序设计 Java

周 斌 王新颖 王继鹏 主编



科学出版社

北京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书是高校培养应用型人才的配套教材,注重将面向对象的思想与 Java 程序设计技术相结合,培养读者使用面向对象的思维方式思考问题,并使用 Java 语言解决问题。本书共分为 11 章。其中,前两章主要介绍 Java 语言的基础知识;第 3、4 章是本书的核心,介绍 Java 中面向对象的概念及具体实现,重点讲述继承和多态;第 5、6 章主要介绍 Java 中各种类型的数组、Java 中的一些常用类、数据结构以及一些常用的算法;第 7 章主要介绍 Java 中的各种流及相关的应用;第 8 章介绍图形化用户界面的设计;第 9 章介绍异常的处理及多线程技术;第 10 章主要介绍 Java 中常用的几种网络通信模式;第 11 章介绍数据库的基础知识以及利用 JDBC 实现 Java 数据库编程;第 12 章以一个实例来介绍 Java 的实际应用。

本书主要作为高等学校计算机及相关专业 Java 课程的教材,也可供 Java 入门的参考书供面向对象编程爱好者和自学 Java 编程的读者使用。

图书在版编目(CIP)数据

面向对象程序设计 Java/周斌,王新颖,王继鹏主编.—北京:科学出版社,2015.1
计算机类主干课程系列教材
ISBN 978-7-03-042637-6

I.①面… II.①周… ②王… ③王… 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(2014)第 277371 号

责任编辑:张颖兵 社 权/责任校对:肖 婷
责任印制:高 嵘/封面设计:苏 波

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉市科利德印务有限公司印刷
科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2015 年 1 月第 一 版 印张:20 1/2

2015 年 1 月第一次印刷 字数:470 000

定价:49.50 元

(如有印装质量问题,我社负责调换)

前 言

面向对象编程技术早已成为当前及今后相当长的一段时间内的主流编程技术,它正在逐步代替传统的面向过程的程序设计技术。面向对象是当前计算机界关心的重点,它是 90 年代软件开发方法的主流。面向对象的概念和应用已超越了程序设计和软件开发,扩展到很宽的范围。如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

Java 语言则是面向对象技术应用最为成功的典范之一,它由 Sun Microsystems 公司于 1995 年推出,从诞生之日起便引起广泛关注,在短短几年的时间内便风靡全球。它以纯面向对象、多线程、平台无关性、高安全性、出色的可移植性和可扩展性受到了计算机界的广泛欢迎,并得到了广泛的应用。随着 IT 业的迅猛发展,Java 在软件开发领域显得越来越重要。目前国内外对 Java 开发人员需求量极大,特别是在最新的软件研发领域。

本书旨在培养读者使用面向对象的思维来分析和解决问题,掌握 Java 程序的编写方式和技巧,以适应现在社会对人才的需求。本书在内容取舍上做了精心选择,强调以简单、易懂的示例来讲述知识要点,使读者能够快速理解和上手,同时确保有一定的深度与广度。本书主要作为高等学校的教材使用,也可作为对 Java 语言感兴趣的读者自学使用。

下面简要介绍本书的主要内容。

第 1 章 Java 语言概述:主要介绍了 Java 语言的发展历史、Java 开发环境的安装与配置,讨论了 Java 程序的分类,通过最简单的 Java 程序示例,介绍 Java 程序开发的基本步骤。

第 2 章 Java 基础语法:主要介绍了 Java 语言的基础语法,例如常量、变量、数据类型和表达式等,为后续章节的程序设计打下基础。

第 3 章 类与对象:主要介绍了 Java 语言的面向对象技术,包括面向对象的基本概念、面向对象的程序设计方法、Java 中的类与对象、方法和域、类的继承以及访问控制属性。

第 4 章 多态、包与接口:主要介绍了面向对象程序设计中多态的概念及具体应用,以及包、接口和枚举类型。

第 5 章 数组与字符串类:首先介绍了 Java 编程中经常用到的数组,包括一维数组、二维数组和对象数组,通过示例程序进一步来讨论它们的使用方式与技巧。然后介绍了一些 Java 中常用的类。

第 6 章 数据结构与常用算法:首先介绍了向量、哈希表的概念与使用方式,以及数据结构中的接口,然后讨论了数据结构中常见的堆栈、队列、链表和二叉树结构以及排序与查找的算法。

第 7 章 流与文件:介绍了流的概念、输入与输出流、缓冲区输入与输出流、格式字节数据流以及文件的处理。

第 8 章 GUI 设计:介绍了 GUI 的概念、文字与图形的 GUI 设计、常用的 GUI 设计组件以及布局设计。

第 9 章 异常处理与多线程:介绍了 Java 中异常处理的机制、处理语句、多线程的实现与

管理。

第 10 章 Java 网络编程:介绍了 Java 网络通信中常见的 Socket 通信模式、UDP 通信模式以及 URL 通信模式。

第 11 章 Java 数据库接口:介绍了数据库的基础知识、SQL 语言的基本语法以及使用 JDBC 实现 Java 访问数据库。

第 12 章 Java 应用实例:以一个实例来介绍 Java 的实际应用。

任何程序设计课程都是一门需要大量实践的课程。读者在学习本书内容的同时应注意辅以大量的实践练习,这样才能较好的理解和掌握书中所介绍的各知识点。同时注意举一反三,增强自己编程时的灵活应变能力,为将来从事软件开发打下良好的基础。

在本书出版过程中,得到科学出版社的大力支持,在此表示感谢。

由于时间仓促,作者水平有限,书中难免存在疏漏之处,欢迎广大读者批评指正。

作者
2014 年 1 月

本书主要介绍 Java 语言的基础知识、SQL 语言的基本语法以及使用 JDBC 实现 Java 访问数据库。本书共分 12 章。第 1 章介绍 Java 语言的发展历史、特点、应用等。第 2 章介绍 Java 语言的基本语法。第 3 章介绍 Java 语言的基本数据类型。第 4 章介绍 Java 语言的控制语句。第 5 章介绍 Java 语言的数组。第 6 章介绍 Java 语言的字符串。第 7 章介绍 Java 语言的异常处理。第 8 章介绍 Java 语言的集合类。第 9 章介绍 Java 语言的线程。第 10 章介绍 Java 语言的网络编程。第 11 章介绍 Java 语言的数据库接口。第 12 章介绍 Java 语言的应用实例。

本书可作为高等院校计算机专业及相关专业的教材,也可供从事 Java 开发的工程技术人员参考。本书由科学出版社出版,定价 49.00 元。

本书由科学出版社出版,定价 49.00 元。

本书由科学出版社出版,定价 49.00 元。

本书由科学出版社出版,定价 49.00 元。

本书由科学出版社出版,定价 49.00 元。

目 录

第 1 章 Java 语言概述	1
1.1 Java 语言的起源与发展	1
1.2 Java 程序的运行机制	2
1.3 Java 运行环境的安装与配置	3
1.4 Java 程序的分类	5
1.5 简单的 Java 程序示例	5
习题 1	9
第 2 章 Java 语言基础	10
2.1 Java 程序的基本结构	10
2.2 Java 程序中的标识符、关键字及分隔符	11
2.3 数据类型	13
2.4 常量与变量	16
2.5 表达式	19
2.6 流程控制语句	25
2.7 Java 基础编程综合示例	38
习题 2	44
第 3 章 类与对象	45
3.1 面向对象的基本概念	45
3.2 面向对象的基本特征	46
3.3 Java 中的类与对象	47
3.4 类的继承	53
3.5 抽象类与最终类	61
3.6 域	65
3.7 方法	70
3.8 访问控制符	74
习题 3	76
第 4 章 Java 面向对象的高级特性	78
4.1 多态机制	78
4.2 内部类	83
4.3 包与接口	88
4.4 枚举类型	93
4.5 程序举例	100
习题 4	105

第 5 章 数组与 Java 常用类	108
5.1 数组	108
5.2 Java 常用类	122
习题 5	138
第 6 章 数据结构与常用算法	140
6.1 向量	140
6.2 哈希表	146
6.3 数据结构中的接口	148
6.4 链表	153
6.5 堆栈	155
6.6 队列	156
6.7 二叉树	158
6.8 排序算法	161
6.9 查找算法	165
习题 6	168
第 7 章 流与文件	169
7.1 流的概念	169
7.2 基本的输入输出流	170
7.3 常用的输入输出流	173
7.4 标准输入输出	179
7.5 文件处理	181
习题 7	187
第 8 章 GUI 设计	188
8.1 GUI 设计概述	188
8.2 Applet 小程序	189
8.3 文字、图形 GUI 设计	193
8.4 常用组件 GUI 设计	200
8.5 容器	222
8.6 布局设计	230
8.7 菜单设计	236
8.8 对话框	240
8.9 Swing GUI 组件	243
习题 8	250
第 9 章 异常处理与多线程	251
9.1 异常	251
9.2 异常处理语句	253
9.3 多线程技术	261

习题 9	275
第 10 章 Java 网络编程	276
10.1 Socket 通信模式	276
10.2 UDP 通信模式	283
10.3 URL 通信模式	287
习题 10	292
第 11 章 Java 数据库应用	293
11.1 数据库基础	293
11.2 利用 JDBC 访问数据库	296
11.3 利用 JDBC 实现 Java 数据库应用实例	302
习题 11	306
第 12 章 Java 应用实例	308
习题 12	317
参考文献	318

第 1 章 Java 语言概述

本章首先介绍 Java 语言的发展历史,并对 Java 开发环境的安装与配置进行详细介绍,接着讨论 Java 程序的分类。本章将通过两个最简单的 Java 程序示例,详细讲述 Java 程序开发的基本步骤。

1.1 Java 语言的起源与发展

Java 语言是由 Sun Microsystems(以下简称 Sun)公司于 1995 年推出。1991 年, Sun 公司预料嵌入式系统将在家电行业发展辉煌,于是试图为下一代智能家电编写一个通用的控制系统,其主要目的是为家用电子产品开发一个分布式编程代码,以便与电冰箱、电视机等家用电器进行信息交流,如把 E-mail 发给它们,对它们进行控制。因此便成立了一个由 James Gosling 为领导的“Green”计划。

开发小组开始准备采用 C++,但由于太复杂和安全性差,最后开发了一种基于 C++ 的新语言 Oak(一种橡树的名字),它是 Java 的前身。Oak 能够用于网络且安全,但当时的 Oak 并没有引起人们的注意,直到 1994 年,互联网和浏览器的出现给 Oak 语言带来了生机,随着互联网和 3W 的飞速发展,Gosling 将 Oak 语言进行了小规模的修改,设计出了一个网页浏览器 WebRunner(该浏览器后来被更名为 HotJava),得到了 Sun 公司首席执行官 Scott McNealy 的支持,才得以研发和发展。在准备注册该商标时,发现 Oak 已经被别人使用,于是更名为 Java。

Sun 公司在 1995 年 5 月正式推出 Java 语言,该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点,当时以 Web 为主要形式的互联网正在迅猛发展,Java 语言的出现迅速引起所有程序员和软件公司的极大关注,程序员们纷纷尝试用 Java 语言编写网络应用程序,并利用网络把程序发布到世界各地进行运行。包括 IBM、Oracle、微软、Netscape、Apple、SGI 等大公司纷纷与 Sun 公司签订合同,授权使用 java 平台技术。

Java 语言与 C 和 C++ 语言存在直接联系。它的语法源自 C,对象模型源自 C++。因此,熟悉 C 和 C++ 的编程人员可以比较容易地学习 Java。但 Java 与 C++ 既不向上兼容,也不向下兼容。Java 并不能完全取代 C++,两者将并存很长一段时间。

目前,Java 语言应用非常广泛,在应用领域占有较大优势,具体体现在以下几个方面:

- (1) 开发桌面应用程序,如银行软件、商场结算软件等。
- (2) 开发面向 Internet 的 Web 应用程序,如门户网站、网上商城、电子商务网站等。
- (3) 提供各行业的数据移动、数据安全等方面的解决方案,如金融、电信、电力等。

Java 语言目前已发展成为最优秀的应用软件开发语言,它有着众多的开源工具。另外 Java 具有面向对象、跨平台、多线程、分布式处理等优点,其使用者越来越多,Java 在应用程序开发领域所占的份额也越来越大。

1.2 Java 程序的运行机制

高级语言的运行机制分为两种：编译型和解释型。

编译型程序语言是指使用相关的编译器，针对特定平台将某种高级语言的源程序文件编译成该平台的硬件所能运行的机器码文件，并生成该平台操作系统所能识别和运行的文件（例如.exe 文件）。以后在该平台运行时无需重新编译，直接使用编译的结果（.exe 文件）即可，如图 1-1 所示。因为编译只执行了一次，所以编译型语言的程序执行效率较高。但是，使用编译型程序语言书写的源程序在编译后生成的目标码文件只能在本平台上运行，无法移植到其他平台运行。典型的编译型语言有 C、C++、Pascal 等。

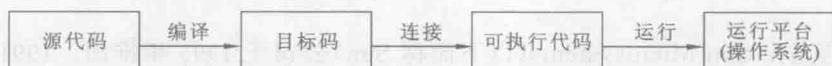


图 1-1 编译型语言的运行机制

解释型程序语言是指使用专门的解释器，将某种高级语言的源程序逐条解释成特定平台的机器码并立即执行，即解释一句执行一句，而不是进行整体编译后再执行，如图 1-2 所示。所以解释型的语言相当于把编译型语言的编译和执行这两步合二为一，并且每次执行都要重复进行“编译”，所以使用这类语言书写的程序执行效率相对较低。但是，只要针对不同平台提供相对应的解释器，就能实现将书写好的程序移植到其他平台。

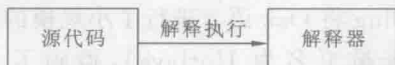


图 1-2 解释型语言的运行机制

Java 是一种将编译型和解释型相结合的语言。Java 程序的执行需要经过编译和解释两个步骤。Java 编译器首先将 Java 源程序编译成与平台无关的字节码文件（.class 文件），然后在 Java 虚拟机（Java Virtual Machine, JVM）上用解释的方式执行该文件，Java 虚拟机中的解释器负责将字节码文件转换成特定操作系统的机器码来执行。所以，无论那一台计算机，无论安装了什么类型的操作系统，只要配置了相应的 Java 虚拟机，即可执行 Java 字节码文件，如图 1-3 所示。因此，Java 同时具备了编译型语言执行效率高和解释型语言跨平台性能好这两个优点，真正实现了“一次编译，随处运行”。

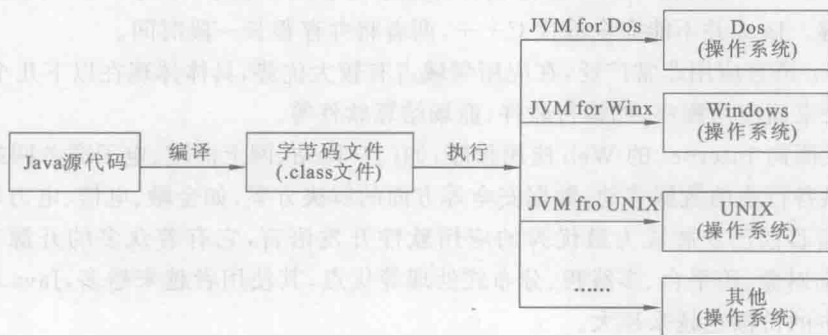


图 1-3 Java 程序的运行机制

1.3 Java 运行环境的安装与配置

Java 开发环境大体上分为两种方式：一种是使用 JDK，这是一种命令行的使用方式；另一种是使用集成开发环境，如 Borland 公司的 JBuilder、IBM 公司的 Eclipse、Sun 公司的 NetBeans 等。其中 JDK 是初学者首选的开发环境，也是构建其他开发工具的基础。

1.3.1 JDK 运行环境安装

JDK(Java Development Kit)是 Sun 公司为 Java 编程人员提供的一套免费的 Java 开发和运行环境。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。JDK 是整个 Java 的核心，包括了 Java 运行环境，Java 工具和 Java 基础的类库。1996 年 1 月，Sun 公司发布了 JDK1.0。1997 年 2 月，Sun 公司发布了 JDK1.1，增加了即时(JIT)编译器，它可以将指令保存在内存中，下次调用时无需重新编译。1998 年 12 月，Sun 公司发布了 JDK1.2，即 Java2 平台。该平台分为标准版 J2SE(Standard Edition)、企业版 J2EE(Enterprise Edition)和微型版 J2ME(Micro Edition)。2005 年 6 月，Sun 公司发布 Java SE6 并将 J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

从 Sun 公司的网站 <http://www.sun.com> 上可以免费下载适合于不同计算机操作系统的 JDK。本书使用的 JDK 是 Windows 操作系统下的开发工具“jdk-6u22-windows-i586.exe”，下面详细介绍 JDK 的安装过程。

获得安装文件“jdk-6u22-windows-i586.exe”后直接双击启动安装，在安装过程中可以自定义安装目录等信息，安装设置界面如图 1-4 所示。

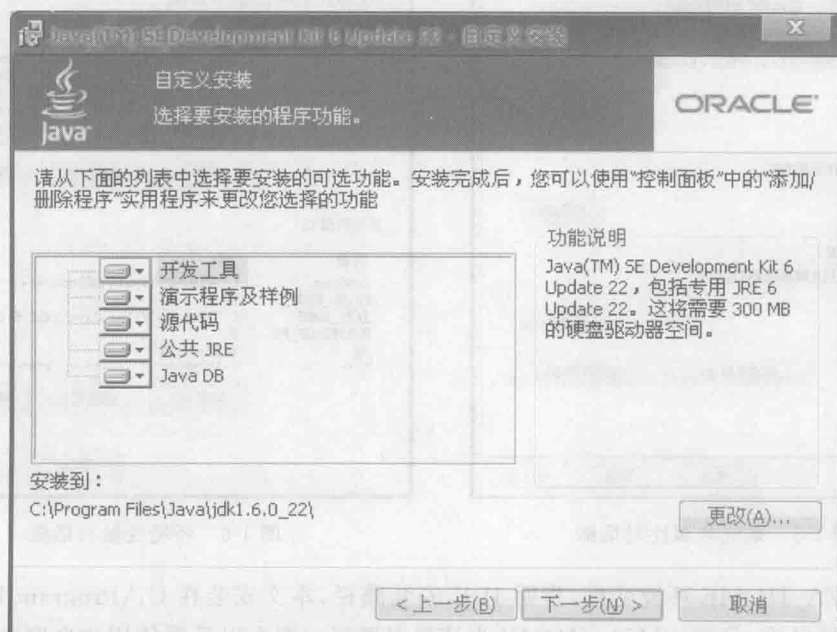


图 1-4 安装设置界面

在安装设置界面,可以通过“更改”按钮修改程序的安装路径,这里采用默认路径为 C:\Program Files\Java\jdk1.6.0_22。单击“下一步”按钮继续安装。

安装 JDK 后产生的主要目录结构如下:

- \bin 目录 存放 jdk 开发工具的可执行文件,包括 Java 编译器、解释器等;
- \demo 目录 存放着含有源代码的不同程序示例;
- \include 目录 包含 C 语言头文件,支持 Java 本地接口与 Java 虚拟机调试程序接口的本地编程技术;
- \jre 目录 Java 运行时环境的根目录,包含 Java 虚拟机,运行时的类包和 Java 应用启动器,但不包含开发环境中的开发工具;
- \lib 目录 Java 类库或库文件,是开发工具使用的归档包文件。

1.3.2 Java 运行环境配置

安装完 JDK 后,必须设置环境变量,JDK 才能正常工作。其中两个变量 PATH 和 CLASSPATH 是必须的。本文设置三个环境变量:JAVA_HOME、CLASSPATH 和 PATH。

下面详细介绍如何配置环境变量。

选中“我的电脑”点击右键,选择“属性”,弹出系统属性对话框,选择“高级”选项,如图 1-5 所示。

在图 1-5 所示系统属性对话框中,选择“环境变量”按钮,进入环境变量对话框(图 1-6)。分别设置如下变量。

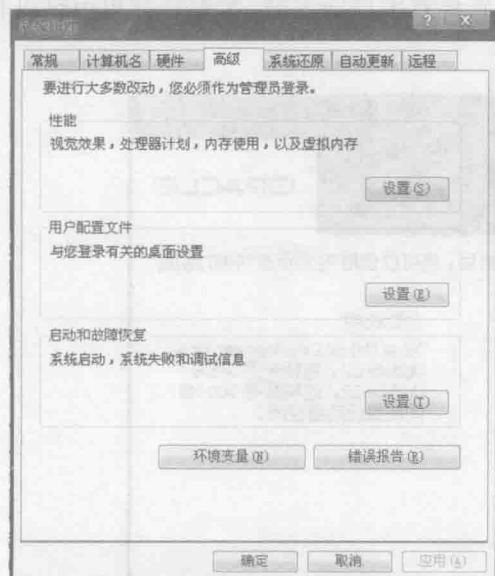


图 1-5 系统属性对话框

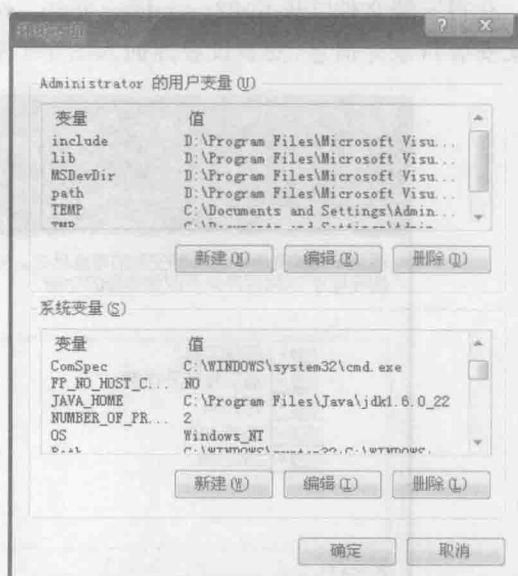


图 1-6 环境变量对话框

(1) JAVA_HOME 环境变量,指明 JDK 安装路径,本文安装在 C:\Program Files\Java\jdk1.6.0_22 目录下,设置 JAVA_HOME 为该目录路径,那么以后要使用这个路径的时候,只需输入 %JAVA_HOME% 即可,避免每次引用都输入很长的路径串。

在图 1-6 所示对话框中,点击系统变量部分的“新建”按钮,设置 JAVA_HOME 变量。

变量名: JAVA_HOME
变量值: C:\Program Files\Java\jdk1.6.0_22

(2) CLASSPATH 环境变量: 指定类的路径, Sun 公司提供了丰富的类包, 一个是 dt.jar, 一个是 tools.jar, 这两个 jar 包都位于 C:\Program Files\Java\jdk1.6.0_22\lib 目录下, 通常都把这两个 jar 包加到 CLASSPATH 环境变量中。

在图 1-6 所示对话框中, 点击系统变量部分的“新建”按钮, 设置 CLASSPATH 环境变量。
变量名: CLASSPATH

变量值: .; %JAVA_HOME%\lib\dt.jar; %JAVA_HOME%\lib\tools.jar

注意: CLASSPATH 变量值最前面有个“.”, 表示当前目录, 这样当运行 Java 程序的时候, 系统就会先在当前目录寻找类文件。

(3) PATH 环境变量: 指定 Java 工具的路径, 使得系统可以在任何路径下识别 Java 命令。

在图 1-6 所示对话框中, 在系统变量部分找到 PATH 变量, 点击“编辑”按钮。

变量名: PATH

变量值: %JAVA_HOME%\bin(PATH 变量里面已经有很多的变量值, 在变量值的最前面加上 %JAVA_HOME%\bin;)

注意: 如果读者的安装路径不一样, 可根据具体的安装路径进行调整。

1.4 Java 程序的分类

Java 程序按其实现环境通常可分为三类: Java Application、Java Applet、Java Servlet。这三种类型程序的区别很少, 并且每种类型的程序都要运行在 Java 虚拟机上。

(1) Java Application: 独立的 Java 应用程序, 只需要 Java 虚拟机就能够运行, 可在命令行单独执行。

(2) Java Applet: 小应用程序, 不能单独运行, 必须依附于一个用 HTML 语言编写的网页并嵌入其中, 通过与 Java 兼容的浏览器来控制执行。当浏览器装入一个含有 Applet 的 Web 页面时, Applet 被下载到该浏览器中, 并开始执行。

(3) Java Servlet: 运行于 Web 服务器端的 Java 程序, 用来生成动态网页。它担当客户请求(Web 浏览器或其他 HTTP 客户程序)与服务器响应(HTTP 服务器上的数据库或应用程序)的中间层。Servlet 是位于 Web 服务器内部的服务器端的 Java 应用程序, 与传统的从命令行启动的 Java 应用程序不同, Servlet 由 Web 服务器进行加载, 该 Web 服务器必须包含支持 Servlet 的 Java 虚拟机。

其他还有 JSP 程序和 Java Bean 等。JSP 程序是 HTML 标记和 Java 程序的混合, 用于产生动态网页, 在支持 JSP 的 Web 服务器上运行。严格地说它并不是独立的 Java 程序。Java Bean 则是可重用的、独立于平台的 Java 程序组件, 通常作为其他 Java 应用程序的一部分, 不单独运行。

1.5 简单的 Java 程序示例

Java 源程序可以用任何文本编辑器来编写代码, 如 Windows 中的记事本等, 然后用命令

行工具进行编译和运行,但是这样使用 JDK 比较麻烦。还可以使用各种功能强大的 Java 集成开发环境,如免费的 Eclipse;或是其他商业产品,如 JBuilder。下面,我们以 Windows 中的记事本为例来说明 Java 程序是如何编写、编译和运行的。

1.5.1 第一个 Java Application 程序

打开 Windows 的记事本,编写例 1-1 所示的 Java 程序代码,保存源程序,修改文件名为 FirstApplication.java。后缀“.java”表示文件格式是 java 类型的。

例 1-1 FirstApplication.java 源代码。

```
public class FirstApplication
{
    public static void main(String args[])
    {
        System.out.println("Welcome to Java World!");
    }
}
```

本程序的运行结果是输出一行信息:

Welcome to Java World!

下面通过 Java 的编译器编译源文件 FirstApplication.java。在 MS-DOS 命令行中输入如下命令进行编译:

```
javac FirstApplication.java
```

上面命令要正确执行需要确保两个条件:

- (1) JDK 已安装成功并配置环境变量。
- (2) 当前处于 FirstApplication.java 文件所在路径,假设该程序放在 D:\java 目录下。

上面命令将 Java 文件编译生成一个以类名字命名,以.class 为后缀的字节码文件,源代码中定义了几个类,编译结果就生成几个字节码文件。例 1-1 中源代码文件中只定义了一个类 FirstApplication,所以编译结果将生成一个名为 FirstApplication.class 的字节码文件,该文件和 FirstApplication.java 在相同的目录下。

调用 Java 解释器解释执行字节码文件 FirstApplication.class,输出程序运行结果。执行命令如下:

```
java FirstApplication
```

图 1-7 显示了程序的运行结果。

上面程序中有如下几个 Java 语法知识点:

(1) 用关键词 class 声明一个新类 FirstApplication。类定义由 {} 括起来,在类中定义了类的变量和类的方法。任何一个 Java 程序都由若干个类定义组成,这些类只能有一个是 public 类(公共类),程序文件名应与公共类同名。

(2) 在该类中定义了一个 main 方法,它是 Java Application 程序执行的入口点。main 方法所在的类称为主类,一个 Java Application 程序只能有一个主类。任何一个 Java Application 类型的程序必须有且只能有一个 main 方法,而且这个 main 方法的方法头必须按照下面的格式书写:

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

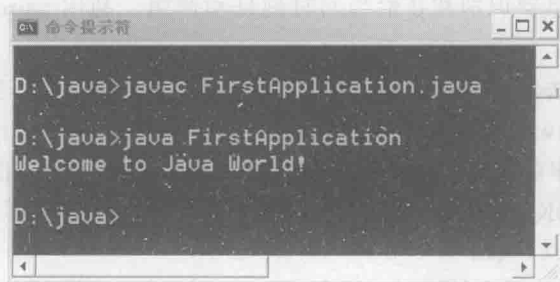


图 1-7 程序的运行结果

当执行 Java Application 时,整个程序将从这个 main 方法的方法体的第一个语句开始执行。

(3) 在 main 方法中,只有一条语句:

```
System.out.print("Welcome to Java World!");
```

它用来将字符串输出到系统屏幕上。

(4) Java 是严格区分大小写的语言,因此文件名必须与主类名称完全一致,包括大小写。

1.5.2 第一个 Java Applet 程序

Applet 是嵌入到 HTML 页面并能够在浏览器中运行的 Java 小应用程序。它的源程序编辑与字节码生成过程与 Java Application 相同。

例 1-2 FirstApplet.java 源代码。

```
import java.awt.Graphics;           //导入类 Graphics
import java.applet.Applet;         //导入类 Applet
public class FirstApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Welcome to Java World!", 20, 20);
    }
}
```

程序说明如下:

(1) 在程序里使用了行注释,在 Java 程序中,两道斜线(//)代表行注释的开始,跟在它后面的所有内容都将被编译器和解释器忽略而作为提高程序可读性的注释部分。

(2) 用关键词 import 分别导入 java.awt 包中的 Graphics 类和 java.applet 包中的 Applet 类。Graphics 类使 Java Applet 能绘制各种图形(如直线、矩形、椭圆等)和字符串。类 Applet 规定了 Applet 程序如何与执行它的解释器——WWW 浏览器配合工作。

(3) 用 class 关键词声明了一个名为 FirstApplet 的类,关键词 extends 表示继承,即 FirstApplet 是系统类 Applet 的子类,它拥有 Applet 类的所有属性和方法。

Java Applet 中不需要有 main 方法,但要求程序中有且必须有一个类是系统类 Applet 的子类。

(4) paint 方法是系统类 Applet 中已经定义好的成员方法,与其他一些 Applet 中的方法

一样,能够被 WWW 浏览器识别和在恰当的时刻自动调用。利用 paint 方法可以绘制出用户想要的各种图形。

(5) drawString 方法是 paint 方法的形式参数 g 的一个成员方法,其功能是在屏幕上的特定位置输出字符串。drawString 方法有 3 个参数,第 1 个参数是要输出的字符串,第 2 个和第 3 个参数是要输出字符串的左下角坐标(20,20),它们是以像素为单位的。

使用如下命令在 JDK 中将源程序编译为 class 文件:

```
javac FirstApplet.java
```

编译后生成 Java 字节码文件 FirstApplet.class。Applet 中没有 main() 方法作为 Java 解释器的入口,为了执行该代码,需要把它嵌入到 HTML 页面中,代码如下所示:

```
<HTML>
<BODY>
<APPLET CODE="FirstApplet.class" WIDTH=300 HEIGHT=150>
</APPLET>
</BODY>
</HTML>
```

HTML 是一种简单的排版描述语言,称为“超文本标记语言”,它通过各种各样的标记来编排超文本信息。例如,<HTML>和</HTML>这一对标记标志着 HTML 文件的开始和结束。在 HTML 文件中嵌入 Java Applet 同样需要使用一组约定好的特殊标记<APPLET>和</APPLET>,其中<APPLET>标记必须包含三个参数:

- CODE 指明嵌入 HTML 文件中的 Applet 字节码文件的文件名。
- WIDTH 指明 Applet 程序在 Web 页面中占用区域的宽度。
- HEIGH 指明 Applet 程序在 Web 页面中占用区域的高度。

将该 HTML 代码保存到 FirstApplet.html 文件中,然后运行该程序输出结果,运行该程序有两种方式。

(1) 通过支持 Java 的浏览器,打开 FirstApplet.html 文件,可以看到程序运行结果,如图 1-8 所示。



图 1-8 程序的运行结果

(2) 通过 JDK 提供的 AppletViewer 运行程序,例如运行例 1-2 中的 Java Applet,可用如下命令:

```
appletviewerFirstApplet.html
```

运行结果如图 1-9 所示。

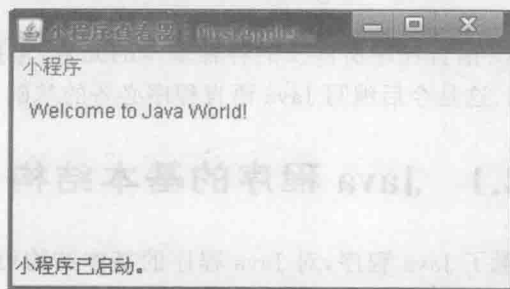


图 1-9 程序的运行结果

习 题 1

- 1-1 下载安装 JDK 软件包,并配置环境变量。
- 1-2 Java 程序有哪几种,各有什么特点?
- 1-3 编写一个 Java application 程序,在屏幕上输出“Hello! Java world!”。
- 1-4 编写一个 Java Applet 程序,在窗口中显示“Hello! Java world!”,并编写相应的 HTML 文件。
- 1-5 编写程序输出如下信息:

```
***
*****
*****
```