

高 职 高 专 计 算 机 系 列 教 材



Java程序设计

主 编 焦卫峰 赵家玲



WUHAN UNIVERSITY PRESS

武汉大学出版社

Java程序设计

第2版



图在就编(CIP)编就

高职高专计算机系列教材

目书机算机算高(系)高
ISBN 978-7-307-08223-8

Java程序设计

主 编 焦卫峰 赵家玲

苗 支:为端 为 版:叔对丑賣 陈 林:舞融丑賣

(山建新 昌高 37005A) 社版出学大武汉:行发总出
(www.wuhanupress.com.cn) 社版出学大武汉:行发总出

同公法中政金北德:物印
半字:87E:德字 8E:德印 1:16 1:16 1:16 1:16
同印为1:德字 1:16 1:16 1:16 1:16
示 00 1:16 1:16 1:16 1:16

知17明高学作图编总子图:图印



WUHAN UNIVERSITY PRESS

武汉大学出版社

图印高学作图编总子图:图印 图印高学作图编总子图:图印

图书在版编目(CIP)数据

Java 程序设计/焦卫峰,赵家玲主编. —武汉:武汉大学出版社,2008.7
(高职高专计算机系列教材)
ISBN 978-7-307-06223-8

I. J… II. ①焦… ②赵… III. JAVA 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 063605 号

责任编辑:林 莉 责任校对:刘 欣 版式:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北金海印务公司

开本:787×1092 1/16 印张:16 字数:376 千字

版次:2008 年 7 月第 1 版 2008 年 7 月第 1 次印刷

ISBN 978-7-307-06223-8/TP·291 定价:27.00 元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

高职高专计算机系列教材

编 委 会

主 任 王化文

编 委 (排名不分先后)

章启俊 陈 晴 李守明 李晓燕 李群芳 张传学 万世明

戴远泉 杨宏亮 明志新 龙 翔 郝 梅 谭琼香

执行编委 黄金文, 武汉大学出版社计算机图书事业部主任, 副编审



内 容 简 介

本书由浅入深，每章利用一个典型案例贯穿始终，详细讲解了 Java 的语法和 Java 面向对象的思想。全书共分 12 章，内容包括 Java 语言基础、控制语句、面向对象编程、数组、异常处理、Java 的常用类、图形用户界面、多线程、Applet 等。

知识点的讲解采用调试通过的完整范例，内容翔实，通俗易懂，读者能够在短时间内掌握 Java 编程的基础知识，提高编程的能力。

本书作为高职高专 Java 编程的入门教材，也可供 Java 初学者使用。



前

言

Java 是 SUN 公司开发的一种面向对象的新一代网络编程语言,它可以在各种不同的机器、操作系统的网络环境中进行开发,具有解释型语言和编译型语言的特性。Java 是目前最流行的编程语言之一,是计算机专业必修的一门专业基础课。

本教材根据高职高专应用型人才的培养目标而编写,具有以下特色:

(1) 以案例组织教材,每一章的内容为一相对完整案例的开发过程。

(2) 对于每一章节,都包括学习内容及目标、同步练习、阅读材料、单元自测、实训指导等内容,便于读者自学。其中的阅读材料是为了开拓知识面而编写,实训指导是实践学的指导书,要求读者在上机前阅读,编写程序代码,完成要求的作业。

(3) 本书的重点是学会编程的基本概念和面向对象的程序设计方法,对 Applet 等内容作了相应删减。目的在于引导学生走入程序设计的大门,培养学生的编程兴趣。在介绍 Java 程序设计基础的同时,还介绍了 Java 常用类库、GUI 等,注重培养学生学习面向对象处理问题的能力,学会运用 Java 语言和面向对象技术解决实际问题,培养与提高学生的自学能力。

(4) 本书通俗易懂,提供了大量的完整范例代码,都已在 Jdk1.5 环境中调试通过。

教材共分 12 章,内容包括 Java 语言基础、控制语句、面向对象编程、数组、异常处理、Java 的常用类、图形用户界面、多线程、Applet 等。

本书在编写的过程中,参考了大量国内外 Java 编程的相关资料,其中包括孙鑫老师的 Java 教学视频、Java 网络资源,以及全国计算机等级考试(二级 Java)的试题库,在此一并表示感谢!

由于作者水平有限,再加上时间仓促,书中难免有不当或疏漏之处,请读者批评指正。

作者

2008 年 4 月



目 录

第 1 章 认识 Java——Hello World 案例	1
1.1 Java 的诞生及特点	1
1.2 Java 开发环境概述	2
1.3 JDK 的安装、环境配置及使用	3
1.4 JDK 下 Java 编程 - Hello World 程序	4
第 2 章 Java 语言基础——Area 程序	12
2.1 Area 程序	12
2.2 标识符、注释和分隔符	13
2.3 变量和常量	14
2.4 基本类型	16
2.5 运算符及表达式	20
2.6 运算符的优先级	25
2.7 数据类型转换	27
2.8 简单语句和复合语句	28
第 3 章 控制语句——四则运算案例	34
3.1 分支语句——简单的四则运算	34
3.2 循环语句——改进的四则运算	40
3.3 break 和 continue 语句	49
第 4 章 Java 面向对象初步——银行账户案例	53
4.1 面向对象的基本概念	53
4.2 类的定义和使用	54
4.3 变量的分类	58
4.4 方法	60
4.5 变量的作用域	64
4.6 封装	65
4.7 构造方法	66
4.8 面向对象的思考方法	69
4.9 static 关键字	70
4.10 this 关键字	72

4.11 方法参数的传值	73
4.12 生成帮助文档	75

第 5 章 面向对象的高级特性——猜花名案例

5.1 类的继承	81
5.2 方法的覆盖	85
5.3 多态性	87
5.4 包	89
5.5 类、方法与变量的说明符	91
5.6 接口	94
5.7 内部类	96
5.8 Java 编程规范	97

第 6 章 数组——成绩管理案例

6.1 成绩管理案例	102
6.2 一维数组	105
6.3 多维数组	110
6.4 与数组相关的操作	112

第 7 章 异常处理——Lucky52 案例

7.1 何为异常	116
7.2 异常类的层次结构	119
7.3 异常的处理机制	120
7.4 throws 关键字	124
7.5 自定义异常	125
7.6 正确地使用异常	126

第 8 章 Java 的常用类

8.1 Java 可重用类的结构	130
8.2 Java 语言包 (java.lang)	131
8.3 java.util 包	138
8.4 输入输出包 Java.io	147

第 9 章 图形用户界面 (GUI) ——通讯录案例

9.1 一个充满个性化的通讯录	164
9.2 抽象窗口工具箱——AWT 组件	169
9.3 Swing 组件	178
9.4 组件的布局	180
9.5 事件处理	185



第10章 多线程——运动的图片	197
10.1 理解线程	197
10.2 线程的实现	199
10.3 线程的四种状态	203
10.4 线程的优先级	205
第11章 Java 小程序 Applet	219
11.1 Applet 程序开发步骤	219
11.2 Applet 的安全	220
11.3 Applet 类	221
11.4 Applet 的 4 种常用方法	221
第12章 Eclipse 的使用——贪吃蛇游戏	225
12.1 认识 Eclipse	225
12.2 开发过程	227
参考文献	241



第1章 认识 Java——Hello World 案例

学习内容及目标

1. Java 的诞生及特点 (了解)。
2. Java 的开发环境 (了解)。
3. JDK 的安装和使用 (了解)。
4. 简单的 Java 程序 (掌握)。
5. 上机步骤 (掌握)。

1.1 Java 的诞生及特点

1.1.1 Java 的诞生

Java 的发展历史, 要从 1990 年开始追溯。当时 Sun Microsystem 公司为了发展消费性电子产品而进行了一个名为 Green 的项目计划。这个计划的负责人是 James Gosling。起初他以 C++ 来写一种内嵌式软件, 可以放在烤面包机或 PAD 等小型电子消费设备里, 使得机器更聪明, 更具有人工智能。但后来他发现 C++ 并不适合这类的任务, 因为 C++ 常会有使系统失效的程序错误, 尤其是内存管理, C++ 是采用直接参考的方式, 需要程序设计师记录并管理内存资源。这造成设计师们极大的负担, 并可能产生许多 bugs。为了解决所遇到的问题, Gosling 最后决定要开发一种新的语言, 来解决 C++ 的潜在性危险问题, 这个语言名叫 Oak。它保留了大部分与 C++ 相似的语法, 但却把些较具危险性的功能加以改进, 像内存资源管理, 便由语言本身来管理, 以减少程序设计师的负担及错误的产生。Oak 是一种可移植性语言, 也就是一种平台独立的语言, 能够在各种芯片上运行。这样各家厂商就可降低研发成本, 直接把应用程序应用在自家的产品上。后来, 因为 Oak 这个商标已被注册, 工程师们便想到以手中常享用的咖啡(Java)来重新命名, 并于 Sun World 95 中被发表出来。从此以后, Java 就随着网络的快速发展, 而逐渐流行起来。

1.1.2 Java 的特点

Java 语言有如下的特点: 简单性、面向对象性、分布性、健壮性、安全性、平台无关性、可移植性、解释性、高效性、多线程、动态性。

简单性: Java 语言的简单性主要体现在三个方面。

(1) Java 的风格类似于 C++, 因而 C++ 程序员初次接触 Java 语言, 就会感到很熟悉。从某种意义上讲, Java 语言是从 C 及 C++ 语言发展而来的, 因此, C++ 程序员可以很快地掌握 Java 编程技术。



(2) Java 摒弃了 C++ 中容易引发程序错误的一些特性,如指针、结构、枚举以及内存管理等。

(3) Java 提供了丰富的类库,可以帮助我们很方便地开发 Java 程序。

面向对象性:面向对象可以说是 Java 最重要的特性,它支持继承、重载、多态等面向对象的特性。Java 语言的设计是完全面向对象的。

健壮性:Java 也是一种强类型的语言,其类型检查比 C++ 还要严格,类型检查帮助我们检查出许多开发早期出现的错误。Java 自己负责内存管理,提供了垃圾内存回收机制,有效地避免了 C++ 中最头疼的内存泄露问题。

安全性:Java 的安全性可从两个方面得到保证。一方面,在 Java 语言里,删除了指针和释放内存等 C++ 功能,避免了非法内存操作;另一方面,通过 Java 的安全体系架构来确保 Java 代码的安全性。

解释性:Java 代码是解释执行的,使用 Java 编译器将 Java 代码先编译成字节码,这是一种中间代码,然后由 Java 解释器解释执行。而 C++ 程序是编译执行的,C++ 程序代码被编译为本地机器指令,然后执行。

与平台无关:Java 作为一种网络语言,其源代码被编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。Java 源程序被编译成一种与机器无关的字节码格式,在 Java 虚拟机上运行。

多线程:Java 语言的一个重要特性就是在语言级支持多线程的程序设计。如果一个人完成某项任务就看成是单线程,那么现在有多个人同时完成该任务,就可以认为是多线程了,这两种方式效率的高低,相信大家都能区分出来。

动态性:Java 的动态特性允许程序动态地装入运行过程中所需要的类,这是采用 C++ 语言进行面向对象程序设计所无法实现的。在 C++ 程序设计过程中,每当在类中增加一个实例变量或一个成员函数后,引用该类的所有子类都必须重新编译,否则将导致程序崩溃。

同步练习

1. Java 是由_____公司开发的。
2. Java 语言有哪些特点?
3. 什么是平台无关性?

1.2 Java 开发环境概述

要开发设计一个 Java 程序必须用到一个工具包 JDK(JDK(Java Development Kit)指的是 Java 开发工具包,你还会听到 JRE, JRE (Java Runtime Environment)指的是 Java 运行环境, JDK 包含 JRE。运行一个 Java 程序必须有 JRE 的支持,但是对 Java 程序的开发需要安装 JDK。

最新的 JDK 可以在 Java.sun.com 免费下载,其中包括了 Java API(Application Programming Interface)的全套文档资料。所谓 API,指的是 Java 类库中现存类的接口,程序员可使用它们构建自己的程序,也就是说关于 Java 的学习实际上包括两部分内容:编写自己的类和方法,学习使用类库。目前有很多公司已经成功地开发了集成化的 Java 开发环境。对于编程初学者,使用 JDK 学习 Java 的编程,有利于理解 Java 程序的开发过程。

Java 开发环境分为三个版本：j2se (Java 2 Standard Edition)、j2ee (Java 2 Enterprise Edition) 和 j2me (Java 2 Micro Edition)。本书讲述的是 j2se。

Java 目前的版本为 1.6 版，从 1.2 版本开始，Java 称之为 Java2。

Java 程序既可以是单独运行的 Java 应用程序 (Application)，也有应用于网页的 Java 小程序 (Applet)。现在，已经有很多新的技术支持各种网页动态效果，Java Applet 的优势不再明显。本书着重讲述的是 Java 应用程序。

同步练习

1. Java 程序可以分为_____和_____两种。
2. Java 开发环境的三个版本是_____、_____和_____。
3. 什么是 API?

1.3 JDK 的安装、环境配置及使用

只要有一台安装好操作系统 (例如 Windows) 的普通 PC 机，再安装 JDK 后即可开发 Java 程序了。除了 Windows 系列的操作系统外，也可下载到适合其他常用的操作系统的 JDK。

(1) 先从 Java 的官方站点 <http://java.sun.com> 下载 JDK，对于 Windows 操作系统，它是一个单独的 .exe 文件，双击该文件进行安装，在安装过程中可以自行设定安装的目录，例如把它安装到 c:\jdk 目录下。

(2) 接下来要配置环境变量，以便 Java 程序能找到已安装的 JDK 和其他配置信息：右键点击“我的电脑”，选择“属性”，在弹出的对话框中选择“高级”、“环境变量”，就可以看到环境变量对话框，如图 1-1 所示。

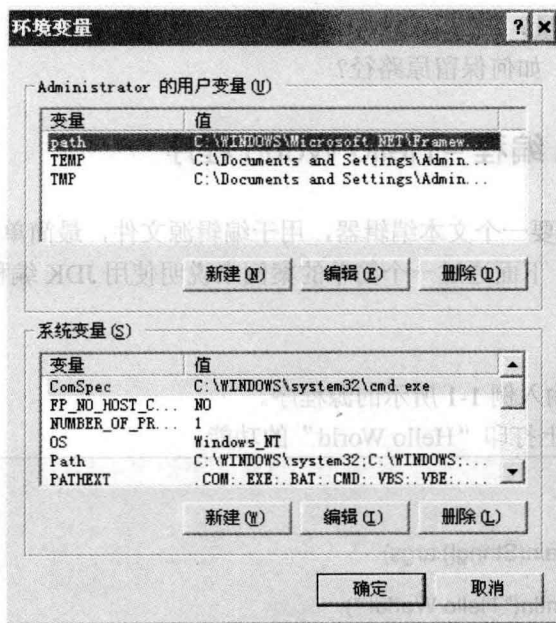


图 1-1 环境变量的配置



窗口上面是用户变量，只对当前用户有效，窗口下面是系统变量，对所有用户都有效。

下一步，找到系统变量 path，点击“编辑”，在最后添加 JDK 的可执行文件的所在目录，如果 JDK 安装到 c:\jdk，这个目录就应该是 c:\jdk\bin。注意和前面的目录要以分号“;”隔开。如图 1-2 所示。

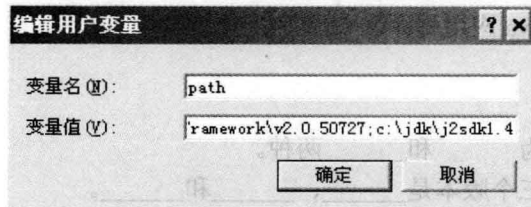


图 1-2 编辑用户变量

系统变量 CLASSPATH 用于设置.class 文件的查找路径，对于初学者，可以不必设定，在解释执行时，在当前目录下查找字节码文件，也可以设置为“.”，“.”代表当前目录。

可以在 Windows 的自动批处理文件 autoexec. bat 中加入：

```
set path=c:\jdk\bin; %path%
```

然后重新启动计算机。

set path 是设置可执行文件的搜索路径，当运行一个可执行文件时，首先在当前目录下进行查找，找到后运行，如果在当前目录下没有找到可执行文件，就在 path 设置的路径中依次进行搜索，找到后运行，所有路径搜索完毕后，仍没有找到，就出现“Bad command or filename”。

set path 设置时，会用新设置的路径替换原路径，为了保留原路径，使用 %path%。

同步练习

1. 如何设置环境变量？
2. 使用 set path 时，如何保留原路径？

1.4 JDK 下 Java 编程 – Hello World 程序

利用 JDK 编程，需要一个文本编辑器，用于编辑源文件，最简单的是 Windows 系统自带的记事本（Notepad）。下面通过一个简单的案例来说明使用 JDK 编程的过程。

1.4.1 编写源文件

打开文本编辑器，输入例 1-1 所示的源程序。

例 1-1 实现在屏幕上打印“Hello World”的功能。

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

文件存为 Hello.java。这里假设保存在 C:\dmt 子目录下。

说明:

(1) Java 程序的基本单位是 class (类), 一个源文件可以有一个或多个 class。

(2) public class Hello

给类起了一个名字“Hello”, “public class”是 Java 中的关键字不能改变, 但 public 可省略。“Hello”是程序员自定义的一个名字。

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

这是固定用法, 程序在执行时总是先找到这一句, 由此开始向下执行。“args”是按照自己习惯命名的变量。

(4) System.out.println(“”);

这是 Java 中的打印输出语句, 用于在屏幕显示一串信息, 其中引号内的内容, 可以换成想显示的任何内容。

[注意]

1' Java 区分大小写, 大小写的含义是不同的。

2' 注意文件格式必须是文本类型, 后缀必须是 .java, 有时候记事本会自动在文件名后加上 .txt 后缀, 在 dos 下可以发现文件名为 *.java.txt 形式, 这时可选择资源管理器的“文件夹选项”菜单, 单击“查看”选项卡, 将“隐藏已知文件类型的扩展名”选项设为不选中, 再在资源管理器中修改文件扩展名。也可保存时在输入的文件名两端加上双引号。

3' 一个连续字符串不能分开在两行上, 可以分开用加号连接, 如:

System.out.println(“my first java” + “ program”);

4' 语句的最后必须用分号(;)结束, 该分号是英文的(;) , 不要误写成中文的(;), 源代码中的标点全部是英文标点。使用中文标点, 程序会出现错误。

1.4.2 将源文件编译成字节码文件

利用 JDK 中的 Java 编译器-javac, 可以将源文件编译成 Java 虚拟机 (JVM) 能够解析的字节码文件。进入 MS-DOS 方式, 找到 Hello.java 文件所在的目录, 在命令行中输入:

C:\dmt>javac Hello.java

如果命令提示符重新出现, 表明没有任何错误, 文件已经成功通过编译, 现在当前目录下应该存在编译产生的 Hello.class 文件。

[注意]

1' 显示“Bad command or filename”: 找不到 javac 命令。一般是路径设置不正确。

2' 如果程序本身有语法错误, 编译将失败, 并会指出错误位置。

3' 尤其注意程序中 String 和 System 的开头字母应均为大写, 其余则为小写。

1.4.3 运行程序

利用 JDK 中的 Java 解析器, 将字节码文件转化为系统能够理解的指令, 加以执行。在同



一个目录下的命令行中输入:

C: \dmt>java Hello

运行结果如下所示:

Hello World!

[注意]

1' 常见的错误是 Exception in thread "main" java.lang.NoClassDefFoundError:Hello, 出现这种错误后可检查当前目录下是否有出错提示的文件 Hello.class; 并检查大小写是否完全一致。

2' Java 命令后面加的是类名, 应该是 Hello, 不是 Hello.class, 类名区分大小写。

同步练习

简述 java 的上机步骤。



阅读材料:

JDK 的环境工具及其使用

1. javac

javac 是 Java 语言编译器。其格式如下:

javac [选项] 文件名.java

javac 选项表

-classpath<path> 被引用类的路径名

-d <directory> 类文件存放路径

-g 为调试器生成附加信息

-ng 不生成附加信息

-nowarn 不显示警告性错误

-o 优化类文件, 使生成的类文件不包含行号

-verbose 显示编译过程的详细信息

javac 命令把 Java 语言源码编译成字节码。javac 编译器对每一个 Java 源文件中定义的类, 生成一个保存编译结果字节码的文件, 其后缀为.class, 存放在与 Java 源文件相同的目录中(除非使用-d 选项, 重新指定存放路径)。

2. java

java 是 Java 语言解释器。其格式如下:

java [选项] 类名 <参数表>

-classpath<path> 设置类搜索路径

-cs 或 -checksource	检查类文件和源程序之间的一致性
-Dpropertyname=value	设置属性值
-debug	允许调用 Java 调试器 jdb
-ms initmem[k m]	设置初始内存空间
-mx maxmem[k m]	设置最大内存空间
-noasyncgc	无用空间搜集程序的自动搜集动作
-noverify	不检验类文件
-oss stacksize[k m]	设置每线程的 Java 代码栈大小
-ss stacksize[k m]	设置每个线程的原始码栈大小
-v, -verbose	显示类装载信息
-verbosegc	显示无用空间搜集程序的运作信息
-verify	检验所装载的全部类文件
-verifremote	检验类装载器所装载的类文件

java 命令用来执行由 javac 命令生成的字节代码。类名参数表示要执行的类名，必须是带包名的全称名称。

3. jdb

jdb 是 Java 语言调试工具。“jdb”命令用于调试 Java 语言编写的程序，其格式如下：

jdb [选项]class

类调试器装载指定的类，启动内嵌的一个 Java 解释器，然后暂停，等待用户发出相应的“jdb”命令。

jdb 命令表

!! 重复上一命令

? 求助命令，列出所有的帮助内容

Catch [exception-class] 出现异常情况时暂停。如果不带参数运行该命令，jdb 将列出当前已经捕捉到的所有异常情况

Classes 列举已经装载的所有类

Clear[class:line] 清除设置在指定类指定行的断点

Cont 恢复执行、直至下一断点

down[n] 对当前线程的调用栈，栈指针下移 n 次，n 默认为 1

Dump id(s) 显示指定对象中所有域的当前值

Exit 退出 jdb

Gc 强制运行无用空间搜集程序

Help 求助命令，与“?”命令相同

ignore exception class 忽略遇到的异常情况，关闭 catch 命令的作用效果

list[line_number] 显示指定的源代码行

Load classname 装载指定的类

locals 显示当前线程的局部变量表

memory 显示内存使用情况