

21世纪高职高专教学做一体化规划教材

Java

程序设计案例教程

教 学 做 一体化

主 编 郭振民 生桂勇
副主编 严春风 徐 芹 李文杰

- ★ 问题导入
- ★ 案例驱动
- ★ 理论够用
- ★ 实践丰富



中国水利水电出版社
www.waterpub.com.cn

21 世纪高职高专教学做一体化规划教材

Java 程序设计案例教程

主 编 郭振民 生桂勇

副主编 严春风 徐 芹 李文杰



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

随着 Internet 和无线网络通信的日益扩大与普及, Java 语言的使用更加广泛。本书依据高等职业教育培养高素质技能型人才的培养目标, 遵循“教学做一体化”的教学模式, 讲解程序员和软件工程师必须掌握的重要语言——Java。

本书共分 3 篇 11 个单元。第一篇“基础篇”包括单元一到单元四, 主要讲解 Java 语言语法基础和面向对象编程基本知识; 第二篇“提高篇”包括单元五到单元九, 主要讲解异常处理、Java 输入输出、GUI 程序应用和线程等; 第三篇“实践篇”包括单元十到单元十一, 主要讲解 Java 项目设计开发和实训。

本书贯穿着“教学做合一”的教学理念和“案例教学法”的主导思想, 具有易学乐学、理实一体、风格新颖的特点。

本书既适合作为高职院校计算机及相关专业的 Java 语言基础教材, 也可作为软件开发人员和爱好者的自学资料。

图书在版编目 (C I P) 数据

Java 程序设计案例教程 / 郭振民, 生桂勇主编. --
北京: 中国水利水电出版社, 2009. 9

21 世纪高职高专教学做一体化规划教材
ISBN 978-7-5084-6828-0

I. ①J… II. ①郭… ②生… 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 (2009) 第 170705 号

策划编辑: 石永峰 责任编辑: 李 炎 加工编辑: 庞永江 封面设计: 李 佳

书 名	21 世纪高职高专教学做一体化规划教材 Java 程序设计案例教程
作 者	主 编 郭振民 生桂勇 副主编 严春风 徐 芹 李文杰
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 12.75 印张 312 千字
版 次	2009 年 9 月第 1 版 2009 年 9 月第 1 次印刷
印 数	0001—4000 册
定 价	22.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

前 言

互联网是“20 世纪最伟大的发明之一”，它对人类社会的发展产生了深远的影响；21 世纪更是“信息时代”，互联网的应用将会更加广泛而深入。Java 语言为支持网络应用程序而设计，具有移植性、平台无关性和安全性等特点。随着 Internet 和无线网络通信的日益扩大与普及，Java 语言的使用也将更加广泛。许多计算机、通信工程和信息技术公司在进行涉及网络、手机或游戏软件开发时，已离不开 Java 语言了。因此，Java 语言是程序员和软件工程师必须掌握的重要语言。

高等职业教育主要是以培养具有职业理想、职业道德，掌握职业技能，知晓职业规范，面向生产、建设、管理、服务第一线需要的高素质技能型人才为培养目标。

从这个目标出发，高职教育应采用“教学做一体化”的教学模式。这正是本书编写的基本原则。

本书秉承陶行知先生“教学做合一”的教学思想，坚持学生的主体地位，在做中教，做中学，以用促学，先会后懂，让学生在实践中掌握理论知识、提高编程能力；采用“案例教学法”，从分析案例入手，提出问题，在解决问题的过程中，学习程序设计的基础知识和原理，进而启迪学生的思维，从而激发学生的学习兴趣。

本书共分 3 篇 11 个单元。第一篇“基础篇”包括单元一到单元四，主要讲解 Java 语言语法基础和面向对象编程基本知识；第二篇“提高篇”包括单元五到单元九，主要讲解异常处理、Java 输入输出、GUI 程序应用和线程等；第三篇“实践篇”包括单元十到单元十一，主要讲解 Java 项目设计开发和实训。

本书贯穿着“教学做合一”的教学理念和“案例教学法”的主导思想，具有如下几个特点：

- **易学乐学：**每个单元针对知识点精心设计了案例。每个案例都采用【案例任务】、【知识必备】、【案例透析】和【现场演练】的结构形式，能充分调动学生的学习兴趣。
- **理实一体：**每个案例的【现场演练】栏目都安排与该案例紧密相关的训练，能及时巩固所学知识；基础篇和提高篇的每个单元最后都结合 Java 二级考试安排了一定的习题，其目的是使本书不仅适用于计算机专业教学，也便于非计算机专业的自学者使用。实践篇的 Java 项目设计开发单元进一步提高读者解决实际问题的工程应用能力与创新意识。
- **风格新颖：**教材通过一个个案例，讲解了 Java 编程的基础知识和主要内容。此外，每个单元的【知识拓展】栏目，对本单元内容进一步扩充和深化，以满足基础好的读者的要求。各个学校可根据需要有选择地教学。

本书既可用作高职院校计算机及相关专业的 Java 语言的基础教材，同时也可作为软件开发人员和爱好者的自学资料。

本书由郭振民、生桂勇任主编，严春风、徐芹、李文杰任副主编。郭振民教授精心策划，

生桂勇老师拟定大纲并组织实施，严春风和李文杰老师参与编写。徐芹老师参与了教材的校对和排版工作，张毅、高峰和杨阳同学参与了部分单元的校对工作。

由于作者水平和经验有限，书中难免会有一些错误，希望广大读者批评指正。

作者 E_mail: gysheng@sina.com

编 者

2009年6月

目 录

前言

基础篇 Java 语言面向对象编程基础

单元一 Java 语言基础.....	2	案例 2-6 静态变量表示赛车数量.....	46
案例 1-1 输出“欢迎学习 Java!”.....	2	知识拓展.....	48
案例 1-2 求圆柱的底面积和体积.....	5	单元训练.....	53
案例 1-3 不同类型数据之间的混合运算.....	7	单元三 继承、抽象类和接口.....	55
案例 1-4 运算符的使用.....	12	案例 3-1 Person 类与 Student 类.....	55
案例 1-5 评定学生考试成绩的等级.....	18	案例 3-2 super 关键字的使用.....	57
案例 1-6 循环语句求和.....	21	案例 3-3 继承抽象类 Shape.....	59
案例 1-7 求素数.....	24	案例 3-4 实现接口.....	62
知识拓展.....	26	知识拓展.....	64
单元训练.....	32	单元训练.....	66
单元二 类与对象.....	35	单元四 数组与字符串.....	70
案例 2-1 Student 类.....	35	案例 4-1 求数组最大值和次大值.....	70
案例 2-2 Student 类的对象“一牛”.....	38	案例 4-2 杨辉三角形.....	72
案例 2-3 利用方法重载求和.....	40	案例 4-3 判断“回文”字符串.....	75
案例 2-4 初始化“通讯录”.....	41	知识拓展.....	78
案例 2-5 this 关键字的使用.....	44	单元训练.....	80

提高篇 Java 语言高级编程

单元五 Java 异常处理.....	84	案例 7-3 学生信息管理系统界面.....	121
案例 5-1 处理典型异常.....	84	案例 7-4 简单“绘图板”.....	126
案例 5-2 抛出异常.....	87	知识拓展.....	135
知识拓展.....	89	单元训练.....	144
单元训练.....	91	单元八 线程.....	145
单元六 Java 输入输出处理.....	92	案例 8-1 利用多线程分解任务.....	145
案例 6-1 输出新建文件的相关信息.....	92	案例 8-2 简单的“电子时钟”.....	149
案例 6-2 基于字节流的输入/输出.....	94	知识拓展.....	153
案例 6-3 基于字符流的输入/输出.....	98	单元训练.....	160
知识拓展.....	102	单元九 Java Applet.....	162
单元训练.....	107	案例 9-1 Applet 生命周期.....	162
单元七 Java GUI 应用程序开发.....	109	案例 9-2 简单的 Applet “影集”.....	167
案例 7-1 登录窗口.....	109	知识拓展.....	169
案例 7-2 计算器界面.....	117	单元训练.....	172

实践篇 Java 简单项目设计开发

单元十 Java 项目设计开发	174	实训四 抽象类、接口和包	189
项目 10-1 计算器	174	实训五 数组与字符串	190
项目 10-2 赛车游戏	178	实训六 异常处理	191
单元训练	185	实训七 Java 输入输出	191
单元十一 实训	186	实训八 图形用户界面	192
实训一 熟悉 Java 开发环境	186	实训九 线程	193
实训二 Java 语言基础	187	实训十 Java Applet	194
实训三 面向对象程序设计	188	参考文献	196

基 础 篇

Java 语言面向对象编程基础

- 单元一 Java 语言基础
- 单元二 类与对象
- 单元三 继承、抽象类与接口
- 单元四 数组与字符串

单元一 Java 语言基础



内容要点

1. Java 程序的类型和组成结构。
2. 标识符、保留字和注释。
3. 数据类型、运算符和表达式。
4. 条件选择结构、循环结构和跳转结构。

案例 1-1 输出“欢迎学习 Java!”



案例任务

编写一个 Java 应用程序运行后显示“欢迎学习 Java!”, 运行结果如图 1-1 所示。

```
C:\j2sdk1.4.2_10\bin>javac HelloJava.java  
C:\j2sdk1.4.2_10\bin>java HelloJava  
欢迎学习Java!
```

图 1-1 Java 应用程序



知识必备

Java 程序分为 Java 应用程序 (Java Application) 和 Applet 小程序 (Java Applet) 两类。Java 应用程序是一种能在支持 Java 的平台上, 通过解释器 (Java.exe) 独立运行的程序; Applet 小程序则是嵌入在 HTML 编写的 Web 页面中, 由 Web 浏览器内含的 Java 解释器解释运行的非独立程序。

每一种语言都有自己特定的书写格式, Java 也不例外。我们先学习 Java 语言的特点。

1. Java 程序的组成

Java 源程序主要由 5 部分组成: package 和 import 语句、类、方法和语句。

1) package 语句位于程序的最前面, 用来定义该程序所属的包, 每个程序只允许使用一条 package 语句。package 语句在程序中可以省略。若省略该语句, 则程序属于默认包。关于包的概念和作用将在单元三中介绍。

2) import 语句用来引入其他包中的类。该语句位于类定义之前, 并且可以多次使用, 引

入多个类。不过，因为 System 是个常用类，该类可以直接使用，也就是说程序一般省略 import java.io.*;。

3) 类 (class) 是整个源程序的核心部分，也是编写程序的地方。一个源程序文件至少有一个类。每个类的内容是用一对大括号括起来的，有不同的名字，但是程序的文件名必须和程序主类的名字相同。主类就是 main() 方法所在的类。class 是定义类的关键字，其后就是类的名称，public 表示此类是公用的，其他程序也可以调用。公用类的定义格式为：

```
public class 类名{  
    语句体;  
}
```

4) 方法：每个 Java 应用程序都要有且只有一个 main() 方法，它是程序运行的开始点。

main() 方法的格式永远都是 public static void main(String args[])，不能更改。关于方法的定义将在单元二中做详细介绍。

5) 语句：类或者方法中的语句体是由一条条以分号结尾的语句组成的。语句是 Java 程序的基本单位之一，是程序具体操作的内容。每条语句占一行，以分号结尾。语句有赋值语句、调用方法语句和对对象定义语句等多种形式，以后的单元中将一一讲解。

2. Java 虚拟机 (JVM)

JVM 是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚拟的机器指令，并有自己的栈、寄存器组等，它是 Java 平台无关的基础。但是，JVM 通常是在软件上面而不是在硬件上实现。在 JVM 上，由 Java 解释器来解释 Java 编译器编译后的程序，即 Java 编程人员在编写软件之后，通过 Java 编译器将 Java 源程序编译为 Java 字节码。

任何一台机器，只要配备了 Java 解释器就可以运行 Java 程序，从而使 Java 程序可以方便地被移植到网络上的不同机器上。图 1-2 为一个 Java 程序的执行过程。

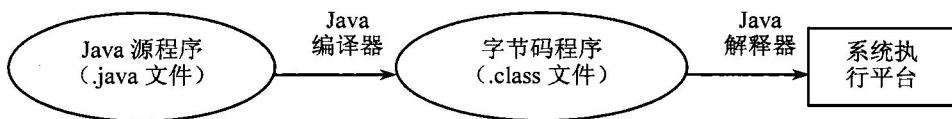


图 1-2 Java 程序的执行过程



案例透析

```
1. public class HelloJava{                //定义类  
2.     public static void main(String args[]){    //main 方法  
3.         System.out.println("欢迎学习 Java! ");    //输出数据  
4.     }  
5. }
```

第 1 行是类的定义。使用关键字 class 定义一个 HelloJava 类，如果 welcome 是一个类，我们可以这样定义：class welcome。class 前面的 public 关键字表示类 HelloJava 的使用范围为公用。HelloJava 类的具体定义始于第 1 行的“{”，结束于第 5 行的“}”。

注意：一个程序可以定义多个类，但只允许有一个公用 (public) 类，程序的文件名必须

要与公用类名相同（包括大小写）。所以，HelloJava 程序只能保存为 HelloJava.java。

第 2 行比较复杂，现在想完全理解它还比较困难，留待以后的学习中逐步理解。我们只需知道这是个方法叫 main。每个 Java 应用程序必须含有一个 main 方法。它从第 2 行的“{”开始，到第 4 行的“}”结束。

第 3 行命令是 main 方法唯一的一个语句，其作用是在 DOS 窗口中输出“欢迎学习 Java!”。System 类是系统包 java.lang 中的一个类。out 是负责输出的对象，也是 System 类的一个静态成员变量，主要用于输出用户信息。这不太好理解，但是请您不要担心，随着学习的深入，您会慢慢理解。“System.out.println()”是一条执行 out 对象的 println()方法的语句。“println”是 print line 的缩写，就是输出一行的意思，括号内双引号包含的内容就是要输出的内容。与此类似的方法是 print，与 println 主要的区别是输出内容后不换行。

关于 Java 应用程序如果您确实有不明白的地方，那就先跳过去吧！在后面的学习中，您一定会理解的！

使用 Applet 小程序显示“欢迎学习 Java!”，运行结果如图 1-3 所示。

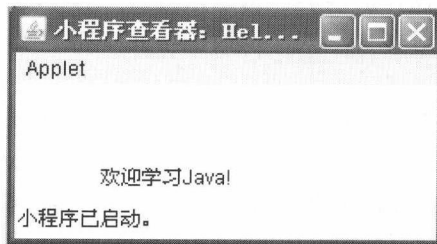


图 1-3 Applet 小程序

1. import java.awt.*;	//导入 java.awt 包中的类
2. import java.applet.*;	//导入 java.applet 包中的类
3. public class HelloApplet extends Applet{	//继承 Applet
4. public void paint(Graphics g){	//重写 paint 方法
5. g.drawString("欢迎学习 Java!",50,60)	//在(50,60)位置输出字符串
6. }	
7. }	

嵌入字节码文件的 HTML 文件：

```
<HTML>
<BODY>
<APPLET code="HelloApplet.class" width="400" height="400">
</APPLET>
</BODY>
</HTML>
```

Applet 程序必须有且仅有一个类是 Applet 的子类。extends 关键字说明 HelloApplet 是 Applet 的子类。该类用 public 修饰，是主类。

Applet 程序中不需要有 main()方法。paint()方法是系统类 Applet 中已经定义好的成员方法。它在显示的 Web 页面需要重画时（例如浏览器窗口在屏幕上移动或放大、缩小等）被浏览器自动调用并执行。使用 Graphics 对象 g 的一个成员方法 drawString()方法，可在指定的位置显

示字符串。



现场演练

1. 简述 Java 程序的执行过程。
2. 编写一个 Java 应用程序和小应用程序，在屏幕上输出如下文字：
我是一名大二的学生。我对 Java 语言很感兴趣！

案例 1-2 求圆柱的底面积和体积



案例任务

编写一个应用程序，定义圆柱的底半径和高，并分别赋值。计算并输出圆柱的底面积和体积。运行结果如图 1-4 所示。

```
C:\Program Files\Java\jdk1.6.0_11\bin>javac Cylinder.java  
  
C:\Program Files\Java\jdk1.6.0_11\bin>java Cylinder  
圆柱底面积是: 314.0  
圆柱体积是: 6280.0
```

图 1-4 案例 1-2 运行结果



知识必备

1. 关键字

关键字又称为保留字，是 Java 语言中具有特殊含义的标识符，由系统使用。因此 Java 语言不允许用户对关键字再赋予其他的含义。Java 语言中的关键字如表 1-1 所示。

表 1-1 Java 语言关键字

abstract	boolean	break	byte	case	catch
char	class	continue	default	do	double
else	extends	false	final	finally	float
for	if	implements	import	instanceof	int
interface	long	native	new	null	package
private	protected	public	return	short	static
super	switch	synchronized	this	throw	throws
transient	true	try	void	volatile	while

2. 标识符

生活中任何事物都有自己的名字。Java 语言中任何一个变量、常量、方法、对象和类都需要有名字，这些名字就是标识符。标识符由编程人员自由指定，但要满足如下规则：

- 1) 标识符可以由字母、数字和下划线（_）、美元符号（\$）组合而成。
- 2) 标识符必须以字母、下划线（_）或美元符号（\$）开头，不能以数字开头。

注意：关键字（保留字）不能作为标识符。Java 标识符区分大小写。

此外，使用 Java 语言编程经常遵循的编码习惯（不是强制性的）是：类名首字母大写，变量、方法及对象的首字母小写。

3. 注释

Java 语言有三种形式的注释：

- 1) 单行注释：// 单行注释
- 2) 多行注释：/* 单行或多行注释 */
- 3) 文件注释（Java 语言特有的 doc 注释）：/** 文件注释 */



案例透析

```

1. /**
2.  *案例 1-2
3.  *作者:一牛
4.  */
5. /*创建一个 Cylinder 类, class 为关键字*/
6. class Cylinder{
7.     public static void main(String args[]){
8.         //public,static,void,main 和 String 都是关键字
9.         double radius=10.0;
10.        //定义 double 型变量 radius 表示底半径, 并赋初值为 10.0
11.        int height=20;
12.        //定义 int 型变量 height 表示高, 并赋初值为 20
13.        double area=0;
14.        //定义 double 型变量 area 表示底面积, 并赋初值为 0
15.        double volume=0;
16.        //定义 double 型变量 volume 表示体积, 并赋初值为 0
17.        area=3.14*radius*radius;           //计算底面积并赋值给变量 area
18.        volume=3.14*radius*radius*height;   //计算体积并赋值给变量 volume
19.        System.out.println("圆柱底面积是: "+area); //输出底面积
20.        System.out.println("圆柱体积是: "+volume); //输出体积
21.    }
22. }
```

程序中的/**.....*/、/*.....*/和//都是注释。注释不会被编译，仅仅起到解释说明的作用。程序中添加适当的注释，可以帮助读者更好地理解代码的含义，增加程序的可读性。第 9、11、13 和 15 行定义了四个变量，分别表示圆柱底半径、高、底面积和体积。第 17、18 行利用公

式计算底面积和体积，并赋值给相应的变量。第 19、20 行分别输出底面积和体积。
System.out.println()输出语句一般的用法有如下几种：

- System.out.println(); //表示换行
- System.out.print(); //表示不换行
- System.out.println("....."); //输出" "中的内容
- System.out.println(变量名); //输出变量值
- System.out.println("....."+变量名);
System.out.println(变量名+".....");
System.out.println("....."+变量名+"....."); }
//输出" "中的内容和变量值。注意变量名和" "之间用"+"相连。

例如：

```
int i=10,j=20;  
System.out.println("输出变量 i 和 j 的值");  
System.out.println("i="+i+"j="+j);
```

输出：输出变量 i 和 j 的值

i=10, j=20



现场演练

1. 下面哪些字符是关键字？

CLASS、sizeof、abstract、LONG、goto、NULL

2. 下面哪些是 Java 标识符？

&Day7、\$ _5678College、You%Me INT、\$ \$ _you89、new、class

3. 参照案例，编写一个程序，求圆的周长和面积并输出（添加适当的注释）。

案例 1-3 不同类型数据之间的混合运算



案例任务

定义不同类型的变量并赋初值，变量之间进行各种运算，并输出运算后的结果。运行结果如图 1-5 所示。

```
C:\Program Files\Java\jdk1.6.0_11\bin>javac DataType.java  
C:\Program Files\Java\jdk1.6.0_11\bin>java DataType  
int 型变量 i= 188  
int 型变量 j= 98  
double 型变量 d= 101.00000015258789  
String 型变量 str= 01210  
short 型变量 s= 1210
```

图 1-5 案例 1-3 运行结果



知识必备

1. 数据类型

在程序设计当中,数据是程序必要的组成部分,也是程序处理的对象。不同的数据有不同的数据类型,不同的数据类型有不同的数据结构和不同的存储方式,并且参与的运算也不同。数据类型定义了数据的性质、取值范围、存储方式以及对数据所能进行的运算和操作。

Java 语言中的数据类型分为基本数据类型、空类型和复合数据类型。复合数据类型在以后的单元中介绍,基本数据类型如表 1-2 所示。

表 1-2 Java 语言的基本数据类型

数据类型	关键字	字节数	取值范围	默认数值
布尔型	boolean	1	true, false	false
字节型	byte	1	-128~127	0
短整型	short	2	-32768~32767	0
整型	int	4	$-2^{31} \sim -2^{31}-1$	0
长整型	long	8	$-2^{63} \sim -2^{63}-1$	0L
单精度浮点型	float	4	$1.4 \times 10^{-45} \sim -3.4 \times 10^{38}$	0.0F
双精度浮点型	double	8	$4.9 \times 10^{-324} \sim -1.8 \times 10^{308}$	0.0D
字符型	char	2	'\u0000'~'\uffff'	'\u0000'

(1) 整型 (byte/short/int/long)

整数有正整数、0、负整数,其含义与数学中的含义相同。Java 语言的整数有 3 种进制的表示形式。

- 十进制:用多个 0~9 之间的数字表示,如 123 和 -100,其首位不能为 0。
- 八进制:以 0 开头,后跟多个 0~7 之间的数字,如 0123。
- 十六进制:以 0x 或 0X 开头,后跟多个 0~9 之间的数字或 a~f 之间的小写字母或 A~F 之间的大写字母。a~f 或 A~F 均分别表示十进制 10~15,如 0x12E。

Java 语言定义了 4 种表示整数的类型,byte (字节型)、short (短整型)、int (整型)、long (长整型)。每种整型数据都是带符号的,每种数据类型的都对应一个默认值(如表 1-2 所示)。一个整数隐含为整型(int 型)。当要将一个整数强制表示为长整型时,需在后面加 L 或 l。

(2) 浮点型 (float/double)

Java 语言用浮点型表示数学中的实数,也就是说既有整数部分又有小数部分的数。浮点数有两种表示形式。

- 标准计数法:由整数部分、小数点和小数部分组成,如 5.0, 2.3612 等。
- 科学计数法:由十进制数、小数点和指数部分组成,指数部分由字母 E 或 e 跟上正负号的整数表示,如 1234.56 可表示为 1.23456E+3。

一个浮点数默认为 double 型。如果数后加 f 或 F 为 float,加 d 或 D 为 double。

(3) 布尔型 (boolean)

布尔型只有 true 和 false 两个取值。其中，true 代表“真”，false 代表“假”，true 和 false 不能转换成数字表示形式。

所有关系运算（如 i>j）的返回值都是逻辑型的值。逻辑型也用于控制语句中的条件表达式，如 if、while、for 等。

(4) 字符型 (char)

Java 语言中的字符型 (char) 用来存储字符。Java 语言是用 Unicode 编码表达的字符，在内存中占 2 个字节。Java 语言中所有可见的 ASCII 字符都可以用单引号括起来成为字符，如 's'、'G'、'*'等。有些很难用一般方式来表达的特殊字符，如回车符、换行符等。为了表达清楚这些特殊字符，Java 中用反斜杠 (\) 开头的字符，将其后面的字符转变为另外的含义，称为转义字符。如表 1-3 所示。

表 1-3 常用转义字符

转义字符	含义
\f	换页，走纸到下一页
\b	退格，后退一格
\n	换行，将光标移到下一行的开始
\r	Enter，将光标移到当前行的行首，但不移到下一行
\t	横向跳格，将光标移到下一个制表符位置
\\	反斜杠字符，输出一个反斜杠
\'	单引号字符，输出一个单引号
\"	双引号字符，输出一个双引号

为了使用上的方便，Java 语言提供了数值型数据的最大值与最小值的标识符及常量值。如表 1-4 所示。

表 1-4 数值型常量的特殊值代码

数据类型	所在的类	最小值代码	最大值代码
byte	Java.lang.Byte	Byte.MIN_VALUE	Byte.MAX_VALUE
short	Java.lang.Short	Short.MIN_VALUE	Short.MAX_VALUE
int	Java.lang.Integer	Integer.MIN_VALUE	Integer.MAX_VALUE
long	Java.lang.Long	Long.MIN_VALUE	Long.MAX_VALUE
float	Java.lang.Float	Float.MIN_VALUE	Float.MAX_VALUE
double	Java.lang.Double	Double.MIN_VALUE	Double.MAX_VALUE

2. 变量

程序通过内存变量来操纵内存中的数据，程序在使用任何变量之前应该首先在该变量和内存之间建立联系。Java 语言的每个变量都具有一个类型，变量的类型决定了变量的数据性质和范围、变量存储在内存中所占空间的大小以及对变量可以进行的合法操作等。

(1) 变量声明

变量声明的一般格式如下:

数据类型 变量名[=初值][, 变量名[=初值]……];

例如: `int i;` 表示声明了标识符 `i` 是 `int` 类型的变量。声明后系统将给变量分配内存空间。

当有多个变量同属一个类型时, 各变量之间用逗号隔开。

如: `int a,b,c;` 表示声明了 3 个 `int` 类型的变量 `a`, `b`, `c`。

(2) 变量初始化

在声明的同时也可以对变量初始化, 即赋初值。

如: `int i=10;`

(3) 变量的赋值

当声明一个变量时没有赋初值或需要重新赋值时, 就需要使用赋值语句, 格式为:

变量名=值;

例如:

```
boolean b=true;      //声明 boolean 型变量并赋值
int i,j=6;           //声明 int 型变量 i 和 j, 并给变量 j 赋值
float f=3.88f        //声明 float 型变量并赋值
double d=3.1415;     //声明 double 型变量并赋值
char c;              //声明 char 型变量
c='\u0088';          //为 char 型变量赋值
x=12;                //为 int 型变量 x 赋值
```

3. 常量

生活中存在着很多常量, 如光在真空中的速度恒定不变, 某人的姓名一般不会轻易改变等。常量是程序中不能修改的固定值, 即常量是在程序运行的整个过程中保持其值不变的量。

常量的声明形式与变量基本一样, 只需在数据类型前加上关键字 `final`。格式如下:

`final 数据类型 常量名=常量;`

例如:

```
final float PI=3.1415f;
final int MIN=100;
```

Java 语言约定常量标识符全部用大写字母表示。程序中使用常量有两点好处: 一是增加可读性, 从常量名可知常量的含义; 二是增强可维护性, 程序中多处使用常量时, 当要对它们进行修改时, 只需修改一处即可。

4. 数据类型转换

程序设计时经常遇到涉及不同类型的数据之间的相互操作。比如: 一个浮点数和一个整数相加。由于浮点数和整数占用的内存位数不同, 运算结果是浮点型还是整型呢? 这就需要进行类型转换处理, 也就是数据类型转换, 简称类型转换。

(1) 数值型数据的转换

● 自动转换

凡是把占用内存比特数少的数据类型向占用内存比特数多的数据类型转换, 编译系统自动完成, 不需要特殊说明, 也就是自动转换。转换规则为: