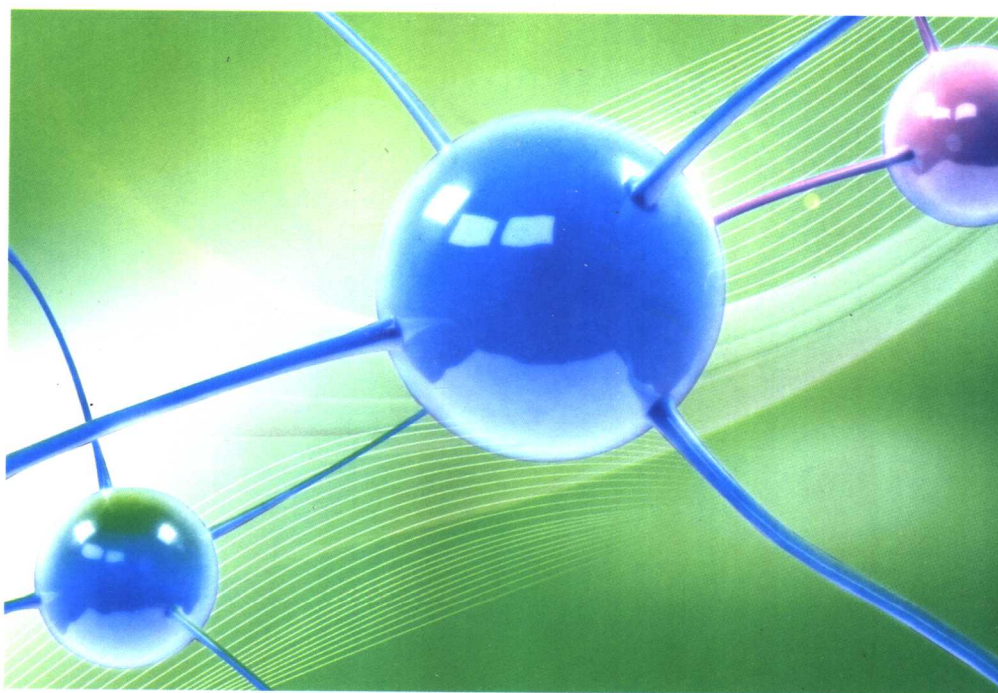


21世纪高等院校应用型规划教材

Java 程序设计 习题解答及上机实践



刘慧宁 王东 等编著



机械工业出版社
CHINA MACHINE PRESS

TP312
1954

21 世纪高等院校应用型规划教材

Java 程序设计习题解答 及上机实践

刘慧宁 王东 等编著



机械工业出版社

本书是与《Java 程序设计》配套的习题解答及上机实践辅导教材。

本书分为习题解答和上机实践两部分。其中,习题解答部分给出了《Java 程序设计》各章习题的参考答案;上机实践部分给出了 16 个精心设计的实验,每一个实验又分为几个子实验,每个子实验针对一个专门的题目,分为“实验题目”、“实验目的”、“实验步骤”和“参考答案”四部分。

本书适合高等院校和培训机构用作 Java 语言程序设计课程的辅导教材,也适合初、中级读者自学。

图书在版编目(CIP)数据

Java 程序设计习题解答及上机实践/刘慧宁等编著.

—北京:机械工业出版社,2006.4

(21 世纪高等院校应用型规划教材)

ISBN 7-111-18852-7

I. J... II. 刘... III. Java 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 030784 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:戴 琳

责任印制:李 妍

北京中兴印刷有限公司印刷

2006 年 5 月第 1 版·第 1 次印刷

184mm×260mm 1/16·12.5 印张·304 千字

0001—5000 册

定价:18.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010)68326294

编辑热线(010)88379739

封面无防伪标均为盗版

出版说明

进入信息时代,我国高等教育面临的情况发生了巨大变化。信息技术日新月异,使得与其相关的课程知识结构更新迅速。由于社会对应用型人才的需求日趋强烈,高校也越来越注重对学生实践能力的培养。大多数高校的上机环境、教师的业务水平和工作条件都得到了明显改善,为教学模式、方法与手段的改革提供了必备的条件。多媒体教室的建设、学生上机时数的增加,实验室建设这一系列措施对教材的建设提出了新的要求。

为了切实体现教育思想和教育观念的转变,依据高等院校教学内容、教学方法和教学手段的现状,机械工业出版社推出了这套“21世纪高等院校应用型规划教材”。

本教材系列以建设“一体化设计、多种媒体有机结合的立体化教材”为宗旨,其目标是:建设一批符合应用型人才培养目标的、适合应用型人才培养模式的系列精品教材。本系列教材的编写者均为相关课程的一线主讲教师,教材内容注重理论与实际应用相结合,其中大力补充新知识、新技术、新工艺、新成果,非常适合各类高等院校、高等职业学校的教学。

为方便老师授课,本套教材为主干课程配备了电子教案、实验指导、习题解答等相关辅助内容。

机械工业出版社

前 言

实践出真知。计算机编程是一门实践性很强的学科,纸上谈兵永远无法真正学会它。因此,实践是读者掌握计算机编程的最重要环节。

本书是与《Java 程序设计》配套的习题解答及上机实践辅导教材。书中内容分为习题解答和上机实践两部分。

习题解答部分给出了《Java 程序设计》各章习题的参考答案。由于编程是一项创造性的活动,编程题的解答从来不是惟一的,因此,笔者不希望读者被书中给出的答案所束缚,应该独立思考,举一反三,尽量自己完成习题之后再参考答案。当然,读者也可以将习题解答当作例题阅读,因为,读程序同样是程序员需要掌握的重要技能。

上机实践部分给出了 16 个精心设计的实验。实验的编排与《Java 程序设计》各章相吻合,内容注重训练基本的编程能力,使读者在实践中达到对书中概念的深入理解和熟练掌握。每一个实验又分为几个子实验,每个子实验针对一个专门的题目,分为“实验题目”、“实验目的”、“实验步骤”和“参考答案”四部分。笔者同样希望读者能独立完成每一个实验。如果时间不允许,也可以选做其中的部分实验,而把其他实验当作例题阅读。不过,实验 1 介绍了 JDK 开发工具的安装和使用,它是所有 Java 编程的基础,读者应该完成它。

本书所有的程序都在 JDK 1.5 环境下调试运行通过。JDK 1.5 针对以前版本作了重大的修改,特别是引入了泛型的概念。因此,部分程序在旧版本中将无法调试通过。

本书上机实践部分中凡提到习题,均用两个数字表示,如“习题 1.6”表示第 1 章的第 6 题,请读者注意。

参加本书编写工作的还有刘晓杰、刘蕾、李清华、郝琦、宋蕊。

本书的错误和不足之处,希望读者及专业人士不吝赐教,我们将非常感激并及时更正。

作 者

目 录

出版说明

前言

第一部分 习题解答

第1章	Java 语言概述	1
第2章	Java 语言编程基础	4
第3章	方法	10
第4章	类和对象	22
第5章	继承与多态	29
第6章	基础类	43
第7章	数组	47
第8章	集合与泛型	57
第9章	异常处理	66
第10章	IO 系统	70
第11章	图形用户界面	75
第12章	Applet 及绘图	93
第13章	多线程	105
第14章	多媒体	110
第15章	网络编程	120

第二部分 上机实践

实验1	安装和使用JDK开发工具	136
实验2	基本数据类型和表达式	140
实验3	控制语句	143
实验4	方法的定义及使用	146
实验5	类、对象和包	150
实验6	继承和多态的编程实现	154
实验7	基础类的使用	157
实验8	数组操作	159
实验9	泛型和集合的使用	163
实验10	异常定义及处理	166
实验11	输入输出	170
实验12	图形用户界面编程	175

实验 13	小应用程序及绘图	179
实验 14	多线程编程	183
实验 15	图像、动画及声音	188
实验 16	网络编程实现	191

第一部分 习题解答

第1章 Java 语言概述

1. Java 语言是由哪个公司推出的?

解答: Sun 公司。

2. 简述 Java 语言的特点。

解答: Java 是一种面向对象的、分布式的、解释型的编程语言,具有简单、健壮、安全、体系结构中立、可移植、高效、多线程、动态等特点。

3. 面向对象编程与面向过程编程的最大区别是什么?

解答: 采用面向过程编程时,程序中数据和处理数据的过程分别存储于不同的地方,数据和过程之间没有逻辑或组织上的联系。而采用面向对象编程时,数据和处理数据的过程被抽象成一个具有特定身份和某些属性的自包含实体——对象。

4. 面向对象系统的最突出特点是什么?

解答: 面向对象系统的最突出特点是封装性、继承性和多态性。

5. 使用解释程序 java 可以运行 Java applet 吗? 如果不能,如何运行它?

解答: Java applet 的源代码经编译生成的字节码文件不能使用解释程序 java 运行。要运行 Java applet 必须首先将 Java applet 字节码文件嵌入网页,然后在内置了 Java 解释程序的浏览器中运行,或者使用 Java applet 查看程序(appletviewer.exe)运行。

6. 编写一个 Java 应用程序,在屏幕上输出“Welcome to Java”。

解答:

```
//WelcomeToJava.java
public class WelcomeToJava
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java");
    }
}
```

习题 6 的运行结果如图 1.1-1 所示。



图 1.1-1 习题 6 的运行结果

7. 编写一个 Java applet, 在屏幕上输出“Welcome to Java”。

解答:

```
//WelcomeApplet.java
import java.awt.Graphics;
import javax.swing.JApplet;

public class WelcomeApplet extends JApplet
{
    public void paint(Graphics g)
    {
        g.drawString("Welcome to Java", 20, 40);
    }
}
```

8. 编写一个 HTML 文件, 将习题 7 中的小应用程序嵌入网页。

解答:

```
//AppletInclude.html
<HTML>
<APPLET CODE="WelcomeApplet.class" WIDTH= 300 HEIGHT= 60>
</APPLET>
</HTML>
```

习题 7 的运行结果如图 1.1-2 所示。

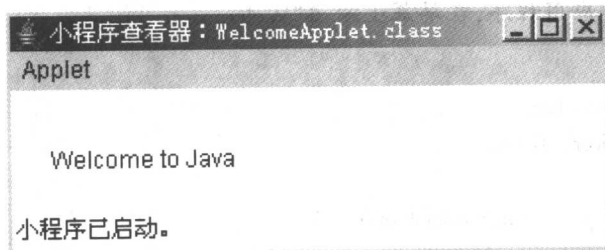


图 1.1-2 习题 7 的运行结果

9. 注释的作用是什么? Java 语言有几种注释? 这几种注释之间有什么区别?

解答: 编写程序时, 在程序中加上必要和清晰的注释, 可提高程序的可读性, 便于程序的日后维护。

Java 语言中, 有 3 种格式的注释信息:

(1) 两道斜线“//”。从“//”开始, 本行中所有字符都是注释信息。

(2) “/*”与“*/”结合。“/*”表示注释的开始, “*/”表示注释的结束。在“/*”、“*/”之间的字符均被视作注释信息。

(3) “/** */”与“*/”结合。“/** */”表示注释的开始, “*/”表示注释的结束。在“/** */”、“*/”之间的字符均被视作注释信息。这种注释信息是 Java 语言所特有的, 它可以被 Java 开发工具中的 javadoc 程序提取加入到程序的文档中。

10. 编写一个程序,从键盘读入一个整数,加 1 后输出结果。

解答:

```
//MyAdd.java
import java.io.*;

public class MyAdd
{
    public static void main(String[] args)
    {
        int i=0,j;
        System.out.print("请输入一个整数:");
        try
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            i = Integer.parseInt(br.readLine());
        } catch (IOException ex) {}
        j = i + 1;
        System.out.println(i + " + " + "1 = " + j);
    }
}
```

习题 10 的运行结果如图 1.1-3 所示。



图 1.1-3 习题 10 的运行结果

第2章 Java 语言编程基础

1. Java 语言中,下列标识符哪些是合法的?

Total、X、A、a、else、a1、-a、7star、_a、my\$、my_friend

解答: Total、X、A、a、a1、_a、my\$、my_friend 是合法的标识符,其余的不是。

2. Total、total 表示的是同一个标识符吗?

解答: 不是。

3. 什么是关键字?

解答: 关键字是 Java 语言的保留字,它们各自有其特殊含义,不能用作标识符。

4. boolean 数据的取值范围是什么? 为什么说 char 属于整数类型?

解答: boolean(布尔型)数据的值只能是 false 或 true。char 归为整数类型是因为 char 型数据可用于算术运算,参与运算时,其值为 Unicode 码值。

5. 声明 double 型变量 x、y,并初始化 y 的值为 4.6。

解答:

```
double x, y=4.6;
```

6. Java 语言中,有哪几种常用的文字常量?

解答: Java 语言中,常用的文字常量有:整型常量、浮点型常量、字符常量、字符串常量和布尔常量。

7. 整型常量的默认数据类型是什么?

解答: 整型常量的默认数据类型是 int 型。

8. 声明一个 float 型变量 x,并初始化 x 的值为 4.6。

解答:

```
float x=4.6f;
```

9. 什么是表达式? 2 是一个表达式吗?

解答: 表达式由运算符和操作数组成,是变量、常量、方法调用和运算符等的组合,用于计算求值等。2 是一个表达式。

10. 计算下列表达式的值。

(1) 21/5

(2) 21%5

(3) 21/5.0

(4) 21%5.2

解答:

(1) 4

(2) 1

(3) 4.2

(4) 0.2

11. 假设下列语句中 a、b、c 为 int 型变量,在按顺序执行完下述语句后它们的值分别是多少?

```
a=1;
b=++a;
c=b++;
```

解答: a=2, b=3, c=2

12. 写出下述程序的输出结果。

```
class A
{
    public static void main(String[] args)
    {
        int i=1;
        int j=2;
        int k=3;
        boolean b=--i>1&& j++<2&& k--<3;
        b=++i>1||j--<2&&--k>3;
        b=++i>1|j--<2&--k>3;
        System.out.println("i="+i);
        System.out.println("j="+j);
        System.out.println("k="+k);
    }
}
```

解答: 输出结果是:

```
i=2
j=0
k=2
```

13. 利用移位运算改写下列表达式。

(1) 20/4

(2) 5*8

解答:

(1) 20>>2

(2) 5<<3

14. 已知两个变量 x、y,利用条件运算符写 1 条语句,求出其中的大者,并将大者的值赋给变量 z。

解答: z=x>y? x:y;

15. 表达式 i=3 与 i==3 的意义相同吗? 如果不同,请阐述两者的差别。

解答: 不同。表达式 i=3 是一个赋值表达式,它的意义是把 3 赋给 i,赋值后表达式和 i 的值均为 3;而表达式 i==3 是一个关系表达式,它首先判断 i 的值是否等于 3,若相等,则表达

式的值为 true, 否则为 false, 但 i 自身的值保持不变。

16. 下列变量声明有何错误?

```
byte b = 150;
short s = 40000;
char c = 400000;
```

解答: 对 int 型常量, 如果其值超出范围, 就不能直接赋给 byte、short 或 char 类型的变量。上述用于赋值的常量都超出了相应变量的取值范围。

17. Java 语言提供了哪几种语句来控制程序的执行流程?

解答: 提供 3 种语句: 条件语句、循环语句、转移语句。

18. 利用 switch 语句与 if 语句的嵌套编写一个程序, 首先提问用户“您的考分是多少?(0~100)”, 然后根据用户的输入值判断其考分等级, 并显示出来。规则如下:

等级 =	{	优	$90 \leq \text{分数} \leq 100$
		良	$80 \leq \text{分数} < 90$
		中	$70 \leq \text{分数} < 80$
		及格	$60 \leq \text{分数} < 70$
		不及格	$0 \leq \text{分数} < 60$

解答:

```
//MyScore.java
import java.io.*;

public class MyScore
{
    public static void main(String[] args)
    {
        int grade = 0;
        System.out.print("您的考分是多少(0~100)?");
        try
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            grade = Integer.parseInt(br.readLine());
        } catch (IOException ex) {}
        if (grade > 100 || grade < 0)
            System.out.println("输入错误! 考分必须在 0~100 之间");
        else
        {
            switch (grade / 10)
            {
                case 10:
                case 9:
                    System.out.println("你的成绩是优!");
                    break;
            }
        }
    }
}
```

```

case 8:
    System.out.println("你的成绩是良!");
    break;
case 7:
    System.out.println("你的成绩是中!");
    break;
case 6:
    System.out.println("你的成绩是及格!");
    break;
default:
    System.out.println("你的成绩是不及格!");
}

```

习题 18 的运行结果如图 1.2-1 所示。



图 1.2-1 习题 18 的运行结果

19. 分别利用 while 循环、do-while 循环、for 循环编程,打印出 1~100 之间能被 9 整除的所有整数。

解答:

(1) while 循环

```

//MyWhile.java
public class MyWhile
{
    public static void main(String[] args)
    {
        int a=0;
        while( ++a<= 100)
            if((a%9)==0)
                System.out.print(a+" \t");
        System.out.println();
    }
}

```

(2) do-while 循环

```
//MyDoWhile.java
public class MyDoWhile
{
    public static void main(String[] args)
    {
        int a = 1;
        do
        {
            if((a%9) == 0)
                System.out.print(a + " \t");
            while( ++a <= 100);
            System.out.println();
        }
    }
}
```

(3) for 循环

```
//MyFor.java
public class MyFor
{
    public static void main(String[] args)
    {
        int a;
        for(a = 1; a <= 100; a++)
            if((a%9) == 0)
                System.out.print(a + " \t");
            System.out.println();
    }
}
```

20. 编程求 $\sum_{i=1}^{10} i!$ (即求 $1! + 2! + 3! + \cdots + 10!$ 的值)。

解答:

```
//MySum.java
public class MySum
{
    public static void main(String[] args)
    {
        int i = 1, j;
        int sum = 0, total;
        while(i <= 10)
        {

```

```

total = 1;
for(j = 1; j <= i; j++)
    total *= j;
sum += total;
i++;

System.out.println("the sum is " + sum);

```

习题 20 的运行结果如图 1.2-2 所示。

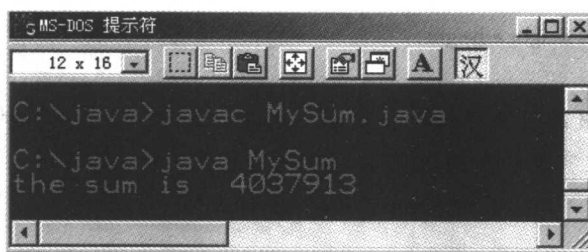


图 1.2-2 习题 20 的运行结果

21. break 语句与 continue 语句在使用上有何不同?

解答:

break 语句用在 switch 语句中,其作用是使程序的执行流程从它所在的 switch 语句中跳出;其用在循环语句中,有两种使用形式:① 不带标号的 break 语句的作用是中断当前循环体的执行,开始执行该循环后面的语句,对于多重循环,执行不带标号的 break 语句只能使程序从其所在的那重循环中跳出;② break 语句与标号搭配使用可以中断多重循环,使程序流程转跳到标号标识的循环体之外。

continue 语句只能用在循环语句的循环体内,它也有两种使用形式:① 不带标号的 continue 语句的作用是中断本次循环,直接回到当前循环的起始处,开始下一轮循环;② continue 语句与标号搭配使用可以使程序流程转跳到标号标识处,直接开始该标号之后循环的下一轮。

第3章 方 法

1. 定义方法计算一个立方体的体积,方法名为 cube,返回值为 double 型,3 个 double 型参数分别为立方体的长、宽、高。

解答:

```
double cube(double length,double width,double height)
{
    return length * width * height;
}
```

2. 编程求两个整数的最大公约数和最小公倍数。其中,定义一个方法求两个整数的最大公约数,在 main 方法中实现两个待求整数的输入及结果的输出。

解答:

```
//DivissorMultiple.java
import java.io.*;

public class DivissorMultiple
{
    static int highestCommonDivissor(int a,int b)    //求最大公约数
    {
        int temp;
        if(a<b)
        {
            temp=a;
            a=b;
            b=temp;
        }
        while(b!=0)
        {
            temp=a%b;
            a=b;
            b=temp;
        }
        return a;
    }

    public static void main(String args[])
    {
        int a=0,b=0,c;
        try
```