

高等应用型人才培养规划教材

Java 程序设计实验指导

(双语版)

高永平 主编 何月顺 官芬芬 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

高等应用型人才培养规划教材

Java 程序设计实验指导 (双语版)

主 编 高永平

副主编 何月顺 官芬芬

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是一部双语实验教材,在教材中提供了相应内容的英文和中文;每一章开始首先对本章主要的知识点给出了概括和精辟的讲解与分析,同时给出 1~2 个综合性的例题;在实验部分围绕本章的重点、难点知识合理选择实验题目,同时给出了实验的参考答案和运行结果。本教材实验内容涵盖广泛,重点培养学生的程序设计能力,列举的例题和实验题目紧密联系实际(银行账户问题、银行支票问题、信用卡问题、事务处理、多线程同步以及死锁问题等),实用性很强。

本教材适合开设该课程的研究生、本科以及高职等各层次人才作为学习 Java 程序设计的实验教材,也可供参加自学考试的学生以及 Java 语言的自学者学习使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计实验指导: 双语版: 汉英对照 / 高永平主编. —北京: 电子工业出版社, 2017.2

ISBN 978-7-121-30676-1

I. ①J… II. ①高… III. ①JAVA 语言—程序设计—高等学校—教学参考资料—汉、英 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2016) 第 311319 号

策划编辑: 程超群 魏建波

责任编辑: 郝黎明

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 10.75 字数: 330 千字

版 次: 2017 年 2 月第 1 版

印 次: 2017 年 2 月第 1 次印刷

定 价: 34.50 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254577, ccq@phei.com.cn。

前 言

Java 是一门面向对象编程语言, Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。

Java 程序设计是实践性很强的课程, 学习 Java 的一个有效方法是多上机实践。本书作者在编写过程中力求概念清晰、通俗易懂, 使复杂问题简单化, 取材新颖, 本书从实际教学出发, 加强了对 Java 语言的重点和难点的知识讲解, 在实践过程中, 充分选用紧密联系实际的问题来深化学生对理论知识的认识, 使学生掌握 Java 语言的程序设计的基本方法, 让学生基本具备使用 Java 开发实际系统的能力, 并培养学生解决实际问题的能力。

本书主要涵盖了 Decisions (选择结构)、Loops (循环结构)、Methods (函数)、Arrays and ArrayLists (数组与数组列表)、Input Output and Exception Handling (输入输出与异常处理)、Objects and Classes (类与对象)、Inheritance and Interfaces (继承与接口)、Graphical User Interfaces (图形用户接口)、the Java Collections Framework (Java 集合框架)、Streams and Binary Input/Output (流与二进制的输入/输出)、Multithreading (多线程)、Programming with JDBC (JDBC 编程)、Internet Networking (Internet 网络) 等 13 个章节的内容, 每章都从 Key points of this chapter (本章要点)、Example (例题)、Experimental contents (实验内容)、Experimental steps (实验步骤)、Experimental result (实验结果) 等四个部分进行介绍。书中所有程序均在 JDK8.0 开发包下编译通过并正确运行; 同时在书后面还附加了两套全英文的试卷和参考答案。

为了使学生在上机实验时目标明确以及方便各院校开展双语教学, 本实验指导书针对课程内容编写了 13 个实验的中英内容。学生可以在实验课时先仔细查看指导书中给出的本章要点、例题, 在此基础上再编写实验程序。为了方便不同背景和实验学时的学生使用, 每个实验都是独立性的实验, 在教学过程中, 教师可以根据实际情况进行适当的裁剪。同时为了方便双语教学的学习和开展, 在本书的最后给出了 2 套英文的考试试卷, 方便学生对专业知识和英语能力的测试。

在长期的 Java 语言教学过程中, 我们发现学生在理论课堂学习的知识总是不能有效地应用于实际编程中, 对于遇到的许多问题无从下手, 影响了学习效果。本书详细地给出了相应的实验步骤, 可引导读者在课后一步一步、循序渐进地完成操作, 同时指出了相应的知识要点, 难度适中, 可以激发读者的学习兴趣, 并为以后更深入地学习 Java 程序设计打

下扎实的基础。

最后本书在编写的过程中，PRABAHARAN KRISHNAMOORTHY、SIVAPRAKASAM MANIKANDAN、李祥、徐洪珍、张军、王强、汪雪元、吴光明、贾惠珍、章伟、吴建东、姜林、汪宇玲、蔡友林等同行专家为本书提供了大量的素材，并为本书的编写提供了宝贵的意见和建议，丁琪琪、吴深深、欧阳浩、朱芮、于浩瀚、欧阳浩、吴文彬、张特等多位同学参与了本书的资料整理工作，在此一并感谢。

由于作者水平有限，书中难免有疏漏和不妥之处，竭诚希望广大读者和同行专家批评指正。

编者

目 录

第 1 章 Decisions (选择结构)	1
1.1 Key points of this chapter (本章要点)	1
1.2 Example (例题)	1
1.3 Experimental contents (实验内容)	3
1.4 Experimental steps (实验步骤)	4
1.5 Experimental result (实验结果)	5
第 2 章 Loops (循环结构)	6
2.1 Key points of this chapter (本章要点)	6
2.2 Example (例题)	7
2.3 Experimental contents (实验内容)	7
2.4 Experimental steps (实验步骤)	8
2.5 Experimental result (实验结果)	10
第 3 章 Methods (函数)	12
3.1 Key points of this chapter (本章要点)	12
3.2 Example (例题)	13
3.3 Experimental contents (实验内容)	16
3.4 Experimental steps (实验步骤)	18
3.5 Experimental result (实验结果)	21
第 4 章 Arrays and ArrayLists (数组与数组列表)	22
4.1 Key points of this chapter (本章要点)	22
4.2 Example (例题)	24
4.3 Experimental contents (实验内容)	27
4.4 Experimental steps (实验步骤)	28
4.5 Experimental result (实验结果)	31
第 5 章 Input Output and Exception Handling (输入输出与异常处理)	32
5.1 Key points of this chapter (本章要点)	32
5.2 Example (例题)	36
5.3 Experimental contents (实验内容)	39
5.4 Experimental steps (实验步骤)	40

5.5	Experimental result (实验结果)	43
第 6 章	Objects and Classes (类与对象)	45
6.1	Key points of this chapter (本章要点)	45
6.2	Example (例题)	47
6.3	Experimental contents (实验内容)	49
6.4	Experimental steps (实验步骤)	50
6.5	Experimental result (实验结果)	53
第 7 章	Inheritance and Interfaces (继承与接口)	54
7.1	Key points of this chapter (本章要点)	54
7.2	Example (例题)	59
7.3	Experimental contents (实验内容)	62
7.4	Experimental steps (实验步骤)	64
7.5	Experimental result (实验结果)	69
第 8 章	Graphical User Interfaces (图形用户接口)	70
8.1	Key points of this chapter (本章要点)	70
8.2	Example (例题)	75
8.3	Experimental contents (实验内容)	82
8.4	Experimental steps (实验步骤)	84
8.5	Experimental result (实验结果)	87
第 9 章	the Java Collections Framework (Java 集合框架)	88
9.1	Key points of this chapter (本章要点)	88
9.2	Example (例题)	94
9.3	Experimental contents (实验内容)	96
9.4	Experimental steps (实验步骤)	97
9.5	Experimental result (实验结果)	99
第 10 章	Streams and Binary Input/Output (流与二进制的输入/输出)	100
10.1	Key points of this chapter (本章要点)	100
10.2	Example (例题)	102
10.3	Experimental contents (实验内容)	106
10.4	Experimental steps (实验步骤)	107
10.5	Experimental result (实验结果)	109
第 11 章	Multithreading (多线程)	110
11.1	Key points of this chapter (本章要点)	110

11.2	Example (例题)	113
11.3	Experimental contents (实验内容)	117
11.4	Experimental steps (实验步骤)	119
11.5	Experimental result (实验结果)	124
第 12 章	Programming with JDBC (JDBC 编程)	125
12.1	Key points of this chapter (本章要点)	125
12.2	Example (例题)	129
12.3	Experimental contents (实验内容)	130
12.4	Experimental steps (实验步骤)	131
12.5	Experimental result (实验结果)	133
第 13 章	Internet Networking (Internet 网络)	135
13.1	Key points of this chapter (本章要点)	135
13.2	Example (例题)	139
13.3	Experimental contents (实验内容)	142
13.4	Experimental steps (实验步骤)	144
13.5	Experimental result (实验结果)	147
附录		148
参考文献		162

第 1 章 Decisions (选择结构)

1.1 Key points of this chapter (本章要点)

1. The if Statement: The if statement is used to implement a decision. When a condition is fulfilled, one set of statements is executed. Otherwise, another set of statements is executed.

if 语句：使用 if 语句来实现一个决定。当条件满足时，执行一组语句。否则，执行另一组语句。

```
if (condition)
{
    statements
}
if (condition) { statements1 }
else { statements2 }
```

2. Nested Branches: It is often necessary to include an if statement inside another. Such an arrangement is called a nested set of statements.

嵌套分支：将一个 if 语句包含在另一个 if 语句内，这通常是必要的。这样的结构被称为嵌套的语句。

1.2 Example (例题)

Here is a typical example: In the United States, different tax rates are used depending on the taxpayer's marital status. There are different tax schedules for single and for married taxpayers. Married taxpayers add their income together and pay taxes on the total. Table 1 gives the tax rate computations. A different tax rate applies to each "bracket". In this schedule, the income in the first bracket is taxed at 10 percent, and the income in the second bracket is taxed at 25 percent. The income limits for each bracket depend on the marital status.

一个典型的例子：在美国，不同的税率是根据纳税人的婚姻状况而决定的。单身和已婚的纳税人有不同的税收计划。已婚的纳税人将他们的收入加在一起，并支付总的税。表 1-1 给出了税率计算。不同税率适用于每一个“分支”。在表中，第一个分支的收入为 10%，第二分支的收入为百分之 25%。每个分支的收入范围取决于婚姻状况。

税率表见表 1-1。

Table1-1 Tax Rate Schedule

If your status is Single and if the taxable income is	the tax is	of the amount over
at most \$32,000	10%	\$0
over \$32,000	\$3,200 + 25%	\$32,000
If your status is Married and if the taxable income is	the tax is	of the amount over
at most \$64,000	10%	\$0
over \$64,000	\$6,400 + 25%	\$64,000

```
import java.util.Scanner;
/**
 * This program computes income taxes, using a simplified tax schedule.
 */
public class TaxCalculator
{
    public static void main(String[] args)
    {
        final double RATE1 = 0.10;
        final double RATE2 = 0.25;
        final double RATE1_SINGLE_LIMIT = 32000;
        final double RATE1_MARRIED_LIMIT = 64000;
        double tax1 = 0;
        double tax2 = 0;
        // Read income and marital status
        Scanner in = new Scanner(System.in);
        System.out.print("Please enter your income: ");
        double income = in.nextDouble();
        System.out.print("Please enter s for single, m for married: ");
        String maritalStatus = in.next();
        // Compute taxes due
        if (maritalStatus.equals("s"))
        {
            if (income <= RATE1_SINGLE_LIMIT)
            {
                tax1 = RATE1 * income;
            }
            else
            {
                tax1 = RATE1 * RATE1_SINGLE_LIMIT;
                tax2 = RATE2 * (income - RATE1_SINGLE_LIMIT);
            }
        }
        else
        {
            if (income <= RATE1_MARRIED_LIMIT)
            {
                tax1 = RATE1 * income;
            }
            else
            {
                tax1 = RATE1 * RATE1_MARRIED_LIMIT;
                tax2 = RATE2 * (income - RATE1_MARRIED_LIMIT);
            }
        }
    }
}
```

```

    }
    else
    {
        tax1 = RATE1 * RATE1_MARRIED_LIMIT;
        tax2 = RATE2 * (income - RATE1_MARRIED_LIMIT);
    }
}
double totalTax = tax1 + tax2;
System.out.println("The tax is $" + totalTax);
}
}

```

1.3 Experimental contents (实验内容)

Write a program that reads a number between 1,000 and 999,999 from the user, where the user enters a comma in the input. Then print the number without a comma. Here is a sample dialog; the user input is in color:

Please enter an integer between 1,000 and 999,999: 23,456

23456

Hint: Read the input as a string. Measure the length of the string. Suppose it contains n characters. Then extract substrings consisting of the first $n - 4$ characters and the last three characters.

1.

编写一个程序，为用户读取一个介于 1,000 和 999,999 之间的数字，用户在输入数字中输入了一个逗号。然后打印需要没有逗号的数字。下面是一个示例对话框，用户输入的数值为颜色数值：

请输入一个介于 1,000 和 999,999 之间的整数：23,456

23456

提示：读取输入为字符串。计算字符串的长度。假设它包含 n 个字符。然后提取子字符串组成的第 $N - 4$ 个字符和最后三个字符。

A supermarket awards coupons depending on how much a customer spends on groceries. For example, if you spend \$50, you will get a coupon worth eight percent of that amount. The following table shows the percent used to calculate the coupon awarded for different amounts spent. Write a program that calculates and prints the value of the coupon a person can receive based on groceries purchased.

2.

Here is a sample run:

Please enter the cost of your groceries: 14

You win a discount coupon of \$ 1.12. (8% of your purchase)

一个超市奖励券取决于顾客在商品上花费的多少。例如，如果你花 50 美元，你会得到一张价值 8% 的优惠券。表 1-2 的折扣表显示用于计算所花费的不同金额的优惠

券的百分比。编写一个程序，在一个人购买商品的基础上计算和打印优惠券的值。

示例运行结果：

请输入你的商品成本：14

你赢了 1.12 美元的折扣。（你购买的 8%）

Table 1-2 Discount Schedule

Money Spent (花费)	Coupon Percentage (折扣率)
Less than \$10	No coupon
From \$10 to \$60	8%
More than \$60 to \$150	10%
More than \$150 to \$210	12%
More than \$210	14%

1.4 Experimental steps (实验步骤)

1.

```
import java.util.Scanner;
public class CharactersTranverse {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("Please enter an integer between 1,000 and
            999,999:");
        Scanner in=new Scanner(System.in);
        String num=in.next();
        in.close();
        String result="";
        while(num.length()>3){
            result=num.substring(num.length()-3)+result;
            num=num.substring(0,num.length()-4);
        }
        result=num+result;
        System.out.print("The result is "+result);
    }
}
```

2.

```
import java.util.Scanner;
public class Coupon {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("Please enter the cost of your groceries:");
        Scanner in= new Scanner(System.in);
        double cost=in.nextDouble();
        if(cost<=10){
            System.out.print("Sorry,there is no coupon!");
        }
    }
}
```

```

    }
    else if(cost>10&&cost<=60){
        System.out.print("You win a discount coupon of $" + cost*0.08+
            "(8% of your purchase)!");
    }
    else if(cost>60&&cost<=150){
        System.out.print("You win a discount coupon of $" + cost*0.1+
            "(10% of your purchase)!");
    }
    else if(cost>150&&cost<=210){
        System.out.print("You win a discount coupon of $" + cost*0.12+
            "(12% of your purchase)!");
    }
    else{
        System.out.print("You win a discount coupon of $" + cost*0.14+
            "(14% of your purchase)!");
    }
}
}
}

```

1.5 Experimental result (实验结果)

实验结果如图 1-1 和图 1-2 所示。

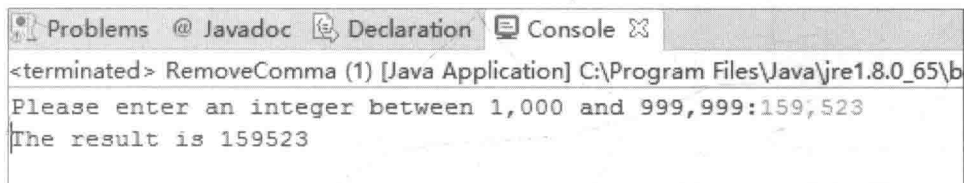


Fig1-1. CharactersTraverse

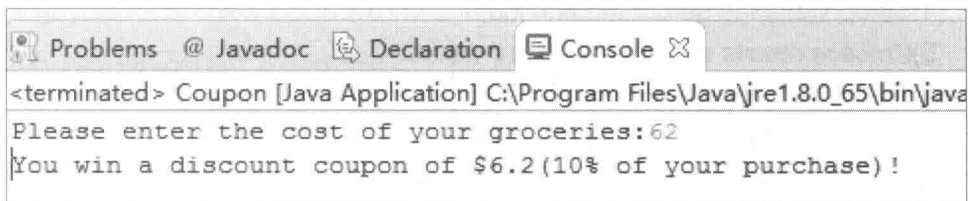


Fig1-2. Coupon

第 2 章 Loops (循环结构)

2.1 Key points of this chapter (本章要点)

1. The while Loop: In Java, the while statement implements such a repetition. It has the form.

while 循环: 在 Java 里面, while 语句实现这样一个重复。它的形式如下:

```
while (condition)
{
    statements
}
```

2. The for Loop: there is a special form for it, called the for loop.

for 循环: 这里有一个特殊的形式叫 For 循环。

```
for (initialization; condition; update)
{
    statements
}
```

Steps to Writing a Loop

- (1)Decide what work to do inside the loop;
- (2)Specify the loop condition;
- (3)Determine loop type;
- (4)Setup variables before the first loop;
- (5)Process results when the loop is finished;
- (6)Trace the loop with typical examples Coding;
- (7)Implement the loop in Java.

3. 循环的步骤

- (1) 决定在循环里面要做什么;
- (2) 明确循环条件;
- (3) 决定采用哪种循环方式;
- (4) 在第一个循环开始之前设置变量;
- (5) 在循环结束后对结果进行加工;
- (6) 用典型的例子代码来跟踪这个循环;
- (7) 在 Java 里面实现这个循环。

2.2 Example (例题)

You put \$10,000 into a bank account that earns 5 percent interest per year. How many years does it take for the account balance to be double the original investment?

你把 10,000 美元放在银行里，每年的利息是 5%，多少年之后，你的余额会变成最初的 2 倍？

```
/**
 * This program computes the time required to double an investment.
 * 这个程序计算要花多少年投资会变成原来的 2 倍。
 */
public class DoubleInvestment
{
    public static void main(String[] args)
    {
        final double RATE = 5;
        final double INITIAL_BALANCE = 10000;
        final double TARGET = 2 * INITIAL_BALANCE;
        double balance = INITIAL_BALANCE;
        int year = 0;
        // Count the years required for the investment to double
        while (balance < TARGET)
        {
            year++;
            double interest = balance * RATE / 100;
            balance = balance + interest;
        }
        System.out.println("The investment doubled after "
            + year + " years.");
    }
}
```

2.3 Experimental contents (实验内容)

Write programs that read a sequence of integer inputs and print:

- The smallest and largest of the inputs;
- The number of even and odd inputs;
- Cumulative totals. For example, if the input is 1 7 2 9, the program should print 1 8 10 19;
- All adjacent duplicates. For example, if the input is 1 3 3 4 5 5 6 6 6 2, the program should print 3 5 6.

编写一个程序，读输入的一串整型值并按要求输出：
输入的最大值和最小值；

奇数和偶数的个数;

依次累计输入总值的和。例如, 如果输入 1 7 2 9, 那么程序应该输出 1 8 10 19

输出输入数值串中相邻重复的值。例如, 如果输入是 1 3 3 4 5 5 6 6 6 2, 那么程序应该输出 3 5 6

2.4 Experimental steps (实验步骤)

1.

```
import java.util.Scanner;
/**
 * The smallest and largest of the inputs:找出输入值中的最大值和最小值
 * @author *****
 */
public class MaxAndMin {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("请输入一些数值 (输入非数值值结束): ");
        Scanner in=new Scanner(System.in);
        boolean run=false;    //监测是否输入值
        int temp;
        int min=Integer.MAX_VALUE;    //使用极大值初始化最小值
        int max=Integer.MIN_VALUE;    //使用极小值初始化最大值
        while(in.hasNextInt()){
            run=true;
            temp=in.nextInt();
            if(temp>max){
                max=temp;
            }
            if(temp<min){
                min=temp;
            }
        }
        in.close();
        if(run){
            System.out.println("最大值为: "+max+", 最小值为: "+min);
        }
        else{
            System.out.println("没有输入任何值!");
        }
    }
}
```

2.

```
import java.util.Scanner;
/**
```



```

* The number of even and odd inputs:合计输入整数中奇偶数的个数
* @author *****
*
*/
public class EvenAndOdd {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("请输入一些整数(输入非数值值结束): ");
        Scanner in=new Scanner(System.in);
        int temp;
        int NumOfEven=0;           //偶数记数
        int NumOfOdd=0;           //奇数记数
        while(in.hasNextInt()){
            temp=in.nextInt();
            if(temp%2==0){
                NumOfEven++;
            }
            else{
                NumOfOdd++;
            }
        }
        in.close();
        System.out.println("偶数个数为"+NumOfEven+", 奇数个数为"+NumOfOdd);
    }
}

```

3.

```

import java.util.Scanner;
/**
 * Cumulative totals:依次累计输入数值的总和
 * @author *****
 *
 */
public class Total {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("请输入一些整数(输入非数值值结束): ");
        final int MAXLENGTH=20;
        Scanner in=new Scanner(System.in);
        int temp[]=new int[MAXLENGTH];
        int length=0;
        while(in.hasNextInt()){
            temp[length]=in.nextInt();
            length++;
        }
        in.close();
        int total=0;
        for(int i=0;i<length;i++){
            total+=temp[i];
        }
    }
}

```