

案例汇编系列



21世纪高校计算机应用技术系列规划教材
丛书主编 谭浩强

Java 程序设计案例汇编

焦玲 邢伟 于海波 王燚 等编著



3

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



21 世纪高校计算机应用技术系列规划教材
丛书主编 谭浩强

Java 程序设计案例汇编

焦玲 邢伟 于海波 王燚 等编著

内 容 简 介

本书针对程序设计语言的特点,通过精彩案例全面地介绍了 Java 程序设计的基础知识、编程方法与技巧,通过综合案例将各知识点结合起来,达到学以致用目的。书中的案例既强调实际应用,又突出应用技能,从而提高读者使用面向对象技术和 Java 语言解决实际问题的能力。

全书共分为四篇:第一篇为温故知新,包括 5 个案例;第二篇为基础设计,包括 12 个案例;第三篇为提高设计,包括 19 个案例;第四篇为综合设计,包括 4 个案例。全书通过 40 个案例说明了 Java 的特点、功能和应用。本书所有案例均在 Java SE6.0 平台编译通过。本书主题鲜明,重点突出,结构紧凑,语言简明扼要,便于自学,是一本很好的 Java 语言程序设计入门书。

本书可作为高等院校计算机或非计算机专业的教材用书,也可作为成人高校和在职人员培训班的教材或自学者的参考书。

图书在版编目(CIP)数据

Java 程序设计案例汇编 / 焦玲等编著. —北京: 中国铁道出版社, 2008. 7

(21 世纪高校计算机应用技术系列规划教材. 案例汇编系列)

ISBN 978-7-113-09054-8

I. J… II. 焦… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 105392 号

书 名: Java 程序设计案例汇编

作 者: 焦 玲 邢 伟 于海波 王 焱 等编著

策划编辑: 严晓舟 秦绪好

编辑部电话: (010) 63583215

责任编辑: 王占清

封面制作: 白 雪

责任校对: 侯 颖 孟 欣

责任印制: 李 佳

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 三河市华业印装厂

版 次: 2008 年 8 月第 1 版

2008 年 8 月第 1 次印刷

开 本: 787mm×1092mm

1/16

印张: 12.75

字数: 295 千

印 数: 5 000 册

书 号: ISBN 978-7-113-09054-8/TP·2947

定 价: 19.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

21 世纪高校计算机应用技术系列规划教材

主 任: 谭浩强

副主任: 陈维兴 严晓舟

委 员: (按姓氏音序排列)

安淑芝	安志远	陈志泊	韩 劼	侯冬梅
李 宁	李雁翎	林成春	刘宇君	秦建中
秦绪好	曲建民	尚晓航	邵丽萍	宋 红
宋金珂	王兴玲	魏善沛	熊伟建	薛淑斌
张 玲	赵乃真	訾秀玲		

21 世纪是信息技术高度发展且得到广泛应用的时代, 信息技术从多方面改变着人们的生活、工作和思维方式。每一个人都应当学习信息技术, 应用信息技术。人们平常所说的计算机教育其内涵实际上已经发展为信息技术教育, 内容主要包括计算机和网络的基本知识及应用。

对多数人来说, 学习计算机是为了利用这个现代化工具进行工作或处理面临的各种问题, 使自己能够跟上时代前进的步伐, 同时在学习过程中努力培养自己的信息素养, 使自己具有信息时代所要求的科学素质, 站在信息技术发展和应用的前列, 推动我国信息技术的发展。

学习计算机课程有两种不同的方法: 一是从理论入手; 一是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才, 对他们来说, 不仅需要“知道什么”, 更重要的是“会做什么”。因此, 在学习过程中要以应用为目的, 注重培养应用能力, 大力加强实践环节, 激励创新意识。

根据实际教学的需要, 我们组织编写了这套“21 世纪高校计算机应用技术系列规划教材”。顾名思义, 这套教材的特点是突出应用技术, 面向实际应用。在选材上, 根据实际应用的需要决定内容的取舍, 坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上, 采取“提出问题——解决问题——归纳分析”的三部曲, 这种从实际到理论、从具体到抽象、从个别到一般的方法, 符合人们的认知规律, 且在实践过程中已取得了很好的效果。

本套教材采取模块化的结构, 根据需要确定一批书目, 提供了一个课程菜单供各校选用, 以后可根据信息技术的发展和教学的需要, 不断地补充和调整。我们的指导思想是面向实际、面向应用、面向对象。只有这样, 才能比较灵活地满足不同学校、不同专业的需要。在此, 希望各校的老师把你们的要求反映给我们, 我们将会尽最大努力满足大家的要求。

本套教材可以作为大学计算机应用技术课程的教材以及高职高专、成人高校和面向社会的培训班的教材, 也可作为学习计算机的自学教材。

由于全国各地、各高等院校的情况不同, 因此需要有不同特点的教材以满足不同学校、不同专业教学的需要, 尤其是高职高专教育发展迅速, 不能照搬普通高校的教材和教学方法, 必须要针对它们的特点组织教材和教学。因此, 我们在原有基础上, 对这套教材作了进一步的规划。

本套教材包括以下五个系列:

- 基础教育系列
- 高职高专系列
- 实训教程系列
- 案例汇编系列
- 试题汇编系列

其中基础教育系列是面向应用型高校的教材, 对象是普通高校的应用型专业的本科学

生。高职高专系列是面向两年制或三年制的高职高专院校的学生，突出实用技术和应用技能，不涉及过多的理论和概念，强调实践环节，学以致用。后面三个系列是辅助性的教材和参考书，可供应用型本科和高职学生选用。

本套教材自 2003 年出版以来，已出版了 70 多种，受到了许多高校师生的欢迎，其中有多种教材被国家教育部评为普通高等教育“十一五”国家级规划教材。《计算机应用基础》一书出版三年内发行了 45 万册，这表示读者和社会对本系列教材的充分肯定，对我们是有力的鞭策。

本套教材由浩强创作室与中国铁道出版社共同策划，选择有丰富教学经验的普通高校老师和高职高专院校的老师编写。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版的过程中，得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持，对此谨表衷心的感谢。

本套教材如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本套教材的不断完善和出版，为我国计算机教育事业的发展和人才培养做出更大贡献。

全国高等院校计算机基础教育研究会会长

“21 世纪高校计算机应用技术系列规划教材”丛书主编

谭浩强

Java 语言自 1995 年 5 月 23 日正式发布以来,在短短的 10 年间就已经风靡全球,扩展到各个应用领域,从网上购物、网上银行、远程教学和虚拟课堂,到远程医疗和诊断等,都在纷纷应用 Java 技术。随着网络科技的飞速发展,Java 语言成为较先进、较流行的程序设计语言之一。

Java 语言如此流行,其主要的原因就是 Java 语言的面向对象的特性。面向对象技术已经成为计算机应用开发领域的主流技术,它所倡导的将数据与操作捆绑封装的程序风格,符合现代大规模软件开发的需求。大规模软件开发要求高可靠性、易维护性和代码重用性,Java 语言除了能够满足这些要求之外,还能提供安全性高、移植性好、操作平台无关的特性。所以,Java 语言的面向对象、平台无关、安全机制和高可靠性等特性,使其成为 21 世纪开发应用程序的首选工具。有人预言,在不久的将来,全球 90% 以上的程序代码将是采用 Java 语言开发的。

本书可作为高等院校计算机或非计算机专业的教材用书,也可作为成人高校和在职人员培训的教材或自学者的参考书。本书主题鲜明,重点突出,结构紧凑,语言简明扼要,使读者在短时间内就能学会应用面向对象技术分析和解决问题,应用 Java 语言进行程序开发。本书将复杂的面向对象理论融入到具体的实例当中,要求读者上机多操作实现课本上的程序案例,只有这样才能学好 Java 语言。

本书中 40 个案例的程序代码都做了认真调试,可直接运行,方便初学者上机操作。

全书共分为四篇:第一篇为温故知新,包括 5 个案例,涉及的内容包括 Java 基础、Java 流程控制、方法、数组和字符串;第二篇为基础设计,包括 12 个案例,涉及的内容包括类的封装、多态、继承、接口等;第三篇为提高设计,包括 19 个案例,涉及的内容包括线程、输入输出、图形用户界面、Applet、包以及异常处理;第四篇为综合设计,包括 4 个案例,涉及的内容包括数据库编程和网络编程等。全书通过 40 个案例说明了 Java 的特点、功能和应用。本书所有案例均在 Java SE6.0 平台编译通过。

读者在使用本书时,首先要学习 Java 的语法规则;其次要学习 Java 的类库。优秀的软件开发人员需要经过大量的上机实践的磨炼,才能开发出高质量的程序。希望读者尽可能将书中的案例进行上机练习。

本书温故知新篇、基础设计篇由于海波编写,提高设计篇由焦玲、邢伟、胡晓辉、王秀红编写,综合设计篇由王焱编写。在编写的过程中,参阅了大量的书籍和网站等参考资料。

由于编者水平有限,书中不足之处在所难免,敬请广大读者批评指正。

编 者

2008 年 6 月

目录

CONTENTS

第一篇 温 故 知 新

案例 1 矩阵的加法和乘法	1
案例 2 银行账户管理	4
案例 3 数字时钟	8
案例 4 宝石图案	11
案例 5 会动的变色字体	14

第二篇 基 础 设 计

案例 6 鞍点——二维数组应用	18
案例 7 八皇后——递归法	21
案例 8 八皇后——非递归法	25
案例 9 循环报数	28
案例 10 猜藏物位置	33
案例 11 猜拳游戏	36
案例 12 猜字母代表的数字	39
案例 13 动物排队——继承多态	42
案例 14 测试回文	45
案例 15 求定积分——继承接口	48
案例 16 验证身份证——字符串	52
案例 17 四则运算——栈	55

第三篇 提 高 设 计

案例 18 小小绘图板	63
案例 19 绘制弧线和扇形	71
案例 20 拼图小游戏	75
案例 21 图形组合	81

案例 22 小小计算器	86
案例 23 声音和图像的协调	92
案例 24 动画特例	96
案例 25 图形的动态效果	102
案例 26 跳跃的圆球	106
案例 27 随机访问文件	109
案例 28 文件复制	114
案例 29 文件列表与文件阅读器	117
案例 30 图形反转	125
案例 31 鼠标事件驱动	128
案例 32 布局管理器——BoxLayout	133
案例 33 进度条演示	138
案例 34 多彩的复选框	141
案例 35 JOptionPane——多样的对话框	144
案例 36 树形结构图	154

第四篇 综合设计

案例 37 简单登录系统	161
案例 38 学生信息查询系统	168
案例 39 网上聊天程序	177
案例 40 记事本	187
参考文献	194

第一篇 温故知新

案例 1 矩阵的加法和乘法

知识要点

- ☑ 数组的声明和初始化
- ☑ 数组的引用和输出
- ☑ 循环语句的应用
- ☑ 最终常量的声明和应用

一、案例需求

数学中的矩阵运算是比较常见的运算。在 Java 语言程序设计中,可以用二维数组来表示矩阵。本案例随机生成两个矩阵,并实现两个矩阵的加法和乘法运算。

案例效果如图 1-1 所示。

```
矩阵 A:
9      4      9
6      4      1
0      8      9

矩阵 B:
7      6      6
6      5      4
2      4      0

矩阵AB的和:
16     10     15
12     9      5
2      12     9

矩阵AB的乘积:
105     110     70
68      60     52
66      76     32
```

图 1-1

二、案例设计

矩阵一般要用二维数组来存储。本案例要同时完成两个矩阵的加法和乘法运算，因此在程序中定义了四个行数和列数相等的二维数组，分别为：

int a[] []: 第一个二维数组；

int b[] []: 第二个二维数组；

int c[] []: 存储上述两个数组的和；

int d[] []: 存储上述两个数组的乘积。

在此，需定义一个常量，二维数组的行和列将按照这个常量的大小来定义，二维数组的大小可以很方便地由这个常量来控制。

如果一个类的成员变量用 final 修饰符修饰，则这个成员变量被称为最终成员变量。最终成员变量一般在声明时进行初始化。一旦最终成员变量被赋值，则在整个程序执行过程中将不能被改变。例如：

```
final static int MAX=3;           //定义最终成员变量MAX，并赋初值为3
```

最后，手工输入两个矩阵会比较烦琐，本案例将调用随机函数来解决初始化数组的工作。Java 类库的 Math 类提供了实现常用数学函数运算的标准方法，这些方法都是 static()方法。例如：

```
Math.random()*10;
```

使用类库可以大大提高程序的编程效率、改善程序的编程环境并降低编程的出错率。

三、案例实现

本案例比较简单，将主要的工作都放在主函数中来实现。

程序代码如下所示：

```
/**
 * 实现两个矩阵之间的加法和乘法
 */
public class MatrixApp{
    final static int MAX=3;
    public static void main(String args[]){
        int i,j,k;
        int a[] []=new int[MAX][MAX];
        int b[] []=new int[MAX][MAX];
        int c[] []=new int[MAX][MAX];           //存储 a、b 矩阵的和
        int d[] []=new int[MAX][MAX];           //存储 a、b 矩阵的乘积

        //初始化 a、b 矩阵
        for(i=0;i<MAX;i++){
            for(j=0;j<MAX;j++){
                a[i][j]=(int)(Math.random()*10);
                b[i][j]=(int)(Math.random()*10);
            }
        }

        //求 a、b 矩阵的和
        for(i=0;i<MAX;i++){
            for(j=0;j<MAX;j++){
```

```
        c[i][j]=a[i][j]+b[i][j];
    }
}
//求 a、b 矩阵的乘积
for(i=0;i<MAX;i++){
    for(j=0;j<MAX;j++){
        d[i][j]=0;
        for(k=0;k<MAX;k++){
            d[i][j]+=a[i][k]*b[k][j];
        }
    }
}
//输出各个矩阵
System.out.println("矩阵 A:");
for(i=0;i<MAX;i++){
    for(j=0;j<MAX;j++){
        System.out.print(a[i][j]+"\\t");
        System.out.println();
    }
}
System.out.println("矩阵 B:");
for(i=0;i<MAX;i++){
    for(j=0;j<MAX;j++){
        System.out.print(b[i][j]+"\\t");
        System.out.println();
    }
}
System.out.println("矩阵 AB 的和:");
for(i=0;i<MAX;i++){
    for(j=0;j<MAX;j++){
        System.out.print(c[i][j]+"\\t");
        System.out.println();
    }
}
System.out.println("矩阵 AB 的乘积:");
for(i=0;i<MAX;i++){
    for(j=0;j<MAX;j++){
        System.out.print(d[i][j]+"\\t");
        System.out.println();
    }
}
}
```

四、案例总结与提高

本案例的代码比较简单，只有两个矩阵相乘的部分稍微复杂一点。第一个矩阵行上各个元素和第二个矩阵列上相应元素乘积的和构成目标矩阵的一个元素。

我们可以在本案例的基础上，做如下提高：

(1) 把矩阵的加法和乘法分成两个函数来实现，以增加程序的灵活性，也可以增加矩阵的其他运算。

(2) 可以将本案例用到的常量 MAX 改成两个常量（如 MAXROW 和 MAXCOL），分别来控制数组的行数和列数。也可以把常量改为变量，在程序运行中指定大小。

案例 2 | 银行账户管理

知识要点

- ☒ 一维数组的声明和初始化
- ☒ 一维数组的引用和输出
- ☒ Hashtable 类的应用

一、案例需求

要在银行里存款，就需要为每个储户建立一个新的账户。一个最简单的账户至少应该包括用户名和账户余额两个基本信息。本案例将演示账户的添加、变更以及撤销过程。

案例效果如图 2-1 所示。

```
命令提示符
目前所有账户的信息:
sun:3000.0
zhao:1000.0
qian:2000.0
增加一个账户"Li"之后的信息:
Li:5000.0
sun:3000.0
zhao:1000.0
qian:2000.0
账户"Li"提取2000之后的信息:
Li:3000.0
sun:3000.0
zhao:1000.0
qian:2000.0
账户"Li"存入3000之后的信息:
Li:6000.0
sun:3000.0
zhao:1000.0
qian:2000.0
删除账户"Li"之后的信息:
sun:3000.0
zhao:1000.0
qian:2000.0
```

图 2-1

二、案例设计

账户中的用户名是不可以重复的，是唯一的。本案例将采用 Hashtable 来存储账户信息，用户

名作关键字, 余额作关键值。

Hashtable 的实质是一个数组 + 链表。Hashtable 是一个关键字对应关键值的表, 使用该表可以迅速索引到某个值。Hashtable 是 Java2 集合框架推出之前的一个老的工具类, 在新的 Java2 集合框架下, 已经被 HashMap 取代。平常主要用到 put()、get()、containsKey()和 clear()方法。

put(): 在该散列表中映射指定的键到指定的值。

get(): 返回指定的键在这个散列表中的映射值。

containsKey(): 检测指定的对象是否是该散列表中的一个键。

clear(): 清除该散列表使其不包含键。

三、案例实现

本案例的各个功能分别由相应的函数实现, 主要是对 Hashtable 的操作。

程序代码如下所示:

```
/**
 * 一个简单的账户管理程序
 */
import java.util.*;

public class BankAccount {
    // 建立一个存放账户及其金额的 Hashtable
    private static Hashtable accountList = new Hashtable();
    // 初始化数据
    BankAccount() {
        String[] nameList = {"Zhao", "Qian", "Sun"};
        float[] numList = {1000, 2000, 3000};
        for (int i = 0; i < nameList.length; i++) {
            accountList.put(nameList[i], new Float(numList[i]));
        }
    }
    // 获取账户金额
    public float getNum(String name) {
        if (accountList.containsKey(name))
            return ((Float) accountList.get(name)).floatValue();
        else
            return -1; // 账户里的余额不能为负数, 账户不存在 // 则返回-1
    }
    // 增加一个新的账户
    public void insert(String name, float num) {
        accountList.put(name, new Float(num));
    }
    // 删除一个账户
    public void delete(String name) {
        if (accountList.containsKey(name))
            accountList.remove(name);
    }
    // 提款, 且提款金额不能大于账户中的金额
    public float getMoney(String name, float num) {
        float tempNum = this.getNum(name); // 账户中的金额
        if (tempNum >= 0) {
```

```

        if (num <= tempNum) { //有足够的余额则提取要求的金额
            this.insert(name, tempNum - num);
            return num;
        }
        else { //余额不足则只提取余额数
            this.insert(name, 0);
            return tempNum;
        }
    }
    else {
        return -1;
    }
}

//存款
public float putMoney(String name, float num) {
    float tempNum;
    tempNum = this.getNum(name);
    if (tempNum >= 0) {
        this.insert(name, tempNum + num);
        return num;
    }
    else {
        return -1;
    }
}

//输出所有的账户信息
public void print() {
    Enumeration nameKey = accountList.keys();
    while (nameKey.hasMoreElements()) {
        String name = nameKey.nextElement().toString();
        System.out.println(name + ": " + this.getNum(name));
    }
}

//演示账户变动
public static void main(String[] args) {
    BankAccount myAcc = new BankAccount();
    System.out.println("目前所有账户的信息:");
    myAcc.print();
    myAcc.insert("Li", 5000);
    System.out.println("增加一个账户\"Li\"之后的信息:");
    myAcc.print();
    myAcc.getMoney("Li", 2000);
    System.out.println("账户\"Li\"提取 2000 之后的信息:");
    myAcc.print();
    myAcc.putMoney("Li", 3000);
    System.out.println("账户\"Li\"存入 3000 之后的信息:");
    myAcc.print();
    myAcc.delete("Li");
    System.out.println("删除账户\"Li\"之后的信息:");
    myAcc.print();
}
}

```

四、案例总结与提高

本案例建立了一个账户管理的 BankAccount 类，其中构造函数初始化总的账户信息，即创建一个 Hashtable 的实例来存储账户信息。

账户管理的主要功能有：查询账户余额、添加或撤销账户、存款和取款和显示账户信息。这几部分功能都由相应的函数来实现，程序比较简单，基本都是针对 Hashtable 的操作。

我们可以在本案例的基础上，做如下提高：

(1) 本案例涉及的账户信息比较简单，还可以再加入其他的有关信息，但关键值就不能只是余额了。

(2) 本案例为了简化程序，约定余额不能为负数，这方面可以改进。

案例3 数字时钟

知识要点

- ☑ Applet 程序的运行过程
- ☑ Applet 程序的建立和运行
- ☑ 通过 Runnable 接口实现多线程

一、案例需求

Applet 程序不能独立运行，必须依附在网页上借助浏览器才能运行。通过 Applet 程序，可以在网页上放置一个数字时钟，显示时间每秒更新一次。

案例效果如图 3-1 所示。

二、案例设计

本案例将写成 Java 小应用程序 (Applet)，通过 `init()`、`start()` 和 `stop()` 方法来控制整个程序的运行。

数字时钟要能以一定的格式在网页上显示，这样就需要用到时间类和时间格式类。在程序中，只需要将相应的包引入到程序中即可，例如：

```
import java.util.Date;           // 获得当前时间
import java.text.DateFormat;     // 提供显示时间格式
```

另外，本案例中需要用到线程每隔 1s 更新一次显示。为此，实现了 Runnable 接口。

为了能在浏览器中运行本案例，需要创建一个 HTML 文件 `Clock.html`，内容如下：

```
<applet code="Clock.class" width=150 height=20>
</applet>
```

三、案例实现

程序代码如下所示：

```
/**
 * 数字时钟
 */
```

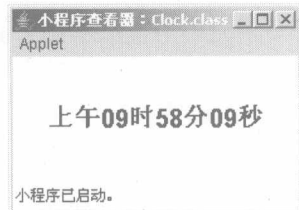


图 3-1