21世纪高等院校计算机科学与技术系列教材

主编　张桂珠　姚晓峰　陈爱国

Java面向对象程序设计

习题解答与实验

Java MIANXIANG DUIXIANG CHENGXU SHEJI
XITI JIEDA YU SHIYAN

北京邮电大学出版社
www.buptpress.com

Java 面向对象程序设计习题解答与实验

张桂珠　姚晓峰　陈爱国　主编

北京邮电大学出版社

内 容 简 介

Java 面向对象程序设计需要做大量的练习题和大量的上机实验题才能逐步掌握其精髓。本书分层次设计了一系列习题和实验，使读者由浅入深地练习和掌握 Java 的编程技巧。

本书是《Java 面向对象程序设计》的配套习题答案和实验，亦可单独作为 Java 的 GUI 程序设计、数据库应用开发、JSP 的 Web 应用开发的习题练习和实验指导书。本书的使用对象是各类编程人员、计算机相关专业的本科生和研究生，也可作为 Java 技术的自学者或短训班人员的自学参考书。

图书在版编目(CIP)数据

Java 面向对象程序设计习题解答与实验/张桂珠，姚晓峰，陈爱国主编. —北京：北京邮电大学出版社，2005

ISBN 7-5635-1080-X

Ⅰ.J… Ⅱ.①张…②姚…③陈… Ⅲ.JAVA 语言—程序设计—高等学校—自学参考资料 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2005)第 096736 号

书　　名：Java 面向对象程序设计习题解答与实验
主　　编：张桂珠　姚晓峰　陈爱国
策 划 人：章　剑
责任编辑：章　剑
出 版 者：北京邮电大学出版社(北京市海淀区西土城路 10 号)　邮编：100876
　　发行部电话：(010)62282185　62283578(传真)
电子信箱：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：北京通州皇家印刷厂印刷
开　　本：787 mm×　1 092　mm　1/16
印　　张：7
字　　数：165 千字
印　　数：1—3 000 册
版　　次：2005 年 8 月第 1 版　2005 年 8 月第 1 次印刷

ISBN 7-5635-1080-X/TP·191　　定价：12.00 元

序

计算机科学技术是科学性与工程性并重的一门学科。它的迅猛发展除了源于微电子学等相关学科的发展外，更主要源于其应用需求的广泛性不断增长，它已渗透到人类社会的各个领域，成为经济发展的倍增器，科学文化与社会进步的催化剂。计算机与通信的融合和全球联网，更显示出它无可限量的发展前景。任何一个领域的发展都离不开计算机已成为无可否认的事实。应用是计算机科学技术发展的动力、源泉和归宿，而计算机科学技术又不断为应用提供先进的方法、设备与环境。

近年来，计算机科学技术的发展不仅极大地促进了整个科学技术的发展，而且明显地推进了经济信息化和社会信息化的进程。计算机科学技术对一个国家在政治、经济、科技、文化、国防等方面的催化作用和强化作用都具有难以估量的意义。计算机知识与能力已成为21世纪人才素质的基本要求之一，因此，计算机科学技术的教育在世界各国都备受重视，我国政府和教育部门对计算机科学技术的教育及人才培养也非常重视。为了适应社会发展对计算机科学技术人才的强烈要求，各高校均在着力培养基础扎实、知识面广、综合素质高、实践能力强、富有创新精神，且具有较强的科学技术运用、推广、转化能力的高层次人才。

由北京邮电大学出版社联合北京邮电大学、武汉大学、华中理工大学及山东、江苏等多所高校的计算机专业教学负责人组成的"21世纪高等院校计算机科学与技术系列教材编委会"按照《中国计算机科学与技术学科教程2002》的要求组织编写的系列教材，体现了近年计算机学科的新理论、新技术。内容涵盖计算机专业学生所应掌握的相关知识，并根据目前计算机科学技术的发展趋势与实际应用相结合，能够满足目前高校计算机专业教学的需要，也可作为计算机专业人员的自学参考材料。

本系列教材作者均为多年从事教学、科研的一线教师，有着丰富的教学和科研实践经验，所编写的这套教材具有结构严谨，内容丰富、理论与实际结合紧密的特点，是他们的教学经验和科研成果的结晶。

计算机科学技术日新月异，所以教材也要不断推陈出新，我希望本系列教材能为我国高校计算机专业教育做出新的贡献。

中国工程院院士　金怡濂

前　言

要提高 Java 面向对象程序设计的能力，需要结合 Java 语言做大量的习题和大量的上机实验题。为了配合《Java 面向对象程序设计》教程的学习，我们编写了《Java 面向对象程序设计习题解答与实验》。

全书共分 3 大部分，第 1 部分是 Java 面向对象程序设计习题答案；第 2 部分是 Java 面向对象程序设计开发环境的介绍；第 3 部分是 Java 面向对象程序设计的实验。对于 Java 的开发环境，我们选择了基于 JDK 的 Java 集成开发环境 JCreator 和支持 JSP 的 Tomcat 服务器环境。本书分层次设计了一系列习题和实验，由浅入深地训练 Java 的编程技巧。

全书由张技珠副教授主编，陈爱国、姚晓峰参编。陈爱国对本书进行了初排版。全书由张桂珠副教授统稿。

本书在编写过程中得到了江南大学教务处、江南大学信息工程学院领导的支持，还得到了毛力、王惠、赵一纯、韩亦强老师的协助和支持。

感谢读者选择使用本书，欢迎您对本书提出批评和修改建议，我们将不胜感激。联系地址及方式如下：

E-mail：zhangguizhu@163.com

通信地址：江苏省无锡市江南大学信息学院 张桂珠 收

邮政编码：214122

作者

2005 年 8 月 7 日

目　录

第 1 部分　习题解答

第 1 章　面向对象程序设计 …… 1
第 2 章　Java 概述和入门程序 …… 1
第 3 章　Java 程序设计基础 …… 2
第 4 章　类和对象 …… 3
第 5 章　类的继承和派生 …… 7
第 6 章　多态性 …… 11
第 7 章　Java 实用包 …… 15
第 8 章　图形和 Java 2D …… 17
第 9 章　GUI 组件和设计 …… 17
第 10 章　异常处理 …… 19
第 11 章　输入和输出流处理 …… 20
第 12 章　线程 …… 21
第 13 章　JDBC 技术和数据库开发应用 …… 22
第 14 章　JSP 技术和开发实例 …… 23

第 2 部分　Java 的编程环境

第 1 章　Java 应用开发环境 …… 28
1.1　JDK 下载、安装与使用 …… 28
1.2　JCreator 集成开发环境的安装 …… 30
1.3　使用 Java 开发环境的例子 …… 31
第 2 章　JSP 运行环境的安装 …… 35
2.1　Tomcat 的安装和配置 …… 35
2.2　在 Tomcat 上部署 Web 应用程序 …… 36

第 3 部分　实验

实验 1　对 Java 应用程序进行编辑、编译、运行 …… 37
实验 2　对 Java 小应用程序进行编辑、编译、运行 …… 38
实验 3　数据类型及表达式 …… 39

实验 4　流程控制语句 …… 41
实验 5　数组 …… 45
实验 6　类和对象 …… 48
实验 7　类的继承和派生 …… 48
实验 8　多态性 …… 49
实验 9　Java 实用包 …… 49
实验 10　图形与 Java 2D …… 50
实验 11　一个简单 GUI 的创建 …… 54
实验 12　布局管理器的使用 …… 56
实验 13　GUI 的综合应用开发 …… 57
实验 14　Java 异常处理的程序设计 …… 57
实验 15　Java 自定义异常的程序设计 …… 59
实验 16　标准输入输出流的程序设计 …… 59
实验 17　文件读写的程序设计 …… 60
实验 18　Java 多线程程序设计 …… 61
实验 19　使用 JDBC 简单查询数据库信息 …… 63
实验 20　使用 JDBC 对数据库信息进行复杂查询 …… 71
实验 21　应用 JSP 动作的程序设计 …… 75
实验 22　使用 JSP 通过 JDBC 访问 SQL Server 数据库 …… 76
实验 23　使用表单请求 JSP 访问数据库 …… 79

附录　部分实验参考答案

实验 6　类和对象 …… 83
实验 7　类的继承和派生 …… 89
实验 8　多态性 …… 91
实验 9　Java 实用包 …… 94
实验 15　Java 自定义异常的程序设计 …… 97
实验 16　标准输入输出流的程序设计 …… 98

第1部分　习题解答

第1章　面向对象程序设计

1. 将一个要解决的问题分解成若干个子问题，每个子问题又被划分成若干个子子问题。这种自顶向下的功能分解一直持续下去，直到子问题足够简单，可以在相应的子过程中解决。

2. 首先，它将数据及对数据的操作行为放在一起，作为一个相互依存、不可分割的整体——对象。对相同类型的对象抽象出其共性，形成类。类中的大多数数据，只能用本类的方法进行处理。对象通过一个简单的外部接口与外界发生联系，对象与对象之间通过消息进行通信。

3. 类是相同对象的集合描述。描述了一类对象的共同属性和操作。对象是类的实例，对应着现实世界的一个实体。

4. 引用变量的类型是类，引用变量指向类的一个对象。

5. 封装性：把对象的属性和操作（或服务）结合为一个独立的整体（系统单位），并尽可能隐藏对象的内部实现细节。

抽象性：类作为一个抽象的数据类型，允许用户从底层实现细节中抽象出来，提供给用户的是在公共接口上的上层操作。这是抽象性的含义。

继承性：在已有类（父类或超类）的基础上派生出新的类（子类），新的类能够吸收已有类的属性和行为，并扩展新的能力。

多态性：是指在超类中定义的属性或行为，被子类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在超类及其各个子类中具有不同的语义。

第2章　Java概述和入门程序

1. 面向对象、平台无关性、可靠性、安全性、多线程、分布式。

2. 应用程序（application）、小应用程序（applet）、servlet和bean。

3. 提示："\n"为换行回车字符。

4. 提示：在JCreator继承环境工具中，编写一个Java的application程序，将3条System.out.println语句放在main方法中。程序编译、运行，观察结果。

5. 提示：主要使用JOptionPane类的输入和输出方法。关键代码如下：

```
String name = JOptionPane.showInputDialog("输入用户的姓名");
String age = JOptionPane.showInputDialog("输入用户的性别");
String concat = name + " " + age;
JOptionPane.showMessageDialog(null,"结果是" + concat,"结果",
        JOptionPane.PLAIN_MESSAGE);      //完整代码参考实验 1
```

第 3 章　Java 程序设计基础

1. 标识符的 4 项规定：

(1) 标识符可以由字母、数字和两个特殊字符下划线(_____)以及美元符号($)组合而成。

(2) 标识符必须以字母、下划线或美元符号开头。

(3) 大小写敏感。

(4) 应该使标识符在一定程度上反映它所表示的变量、常量、对象或类的意义。所以，(1)、(2)、(5)对，(3)、(4)错。

2. (1) 该算术表达式的值为：2.5。

(2) 该算术表达式的值为：3.5。

3. (1) $a=24$　(2) $a=0$　(3) $a=60$　(4) $a=0$　(5) $a=0$　(6) $a=-120$

4. (1) true　(2) true　(3) true　(4) false

5. (1) 解答：关键代码如下

```
grade = Integer.parseInt(in.readLine());
    if (grade >= 90)
        System.out.println("A");
    else if (grade >= 80)
       System.out.println("B");
       else if (grade >= 70)
          System.out.println("C");
          else if (grade >= 60)
             System.out.println("D");
             else
                System.out.println("E");
                ...
```

(2) 解答：关键代码如下

```
...
for (i = 0;i < myArray.length;i++){
   max = myArray[i];
      k = i;
      for (j = i + 1;j < myArray.length;j++)
```

```
            if(myArray[j]>max){
              max=myArray[j];
              k=j;
            }
        if(k!=i){
          myArray[k]=myArray[i];
          myArray[i]=max;
        }
      }
    …
```

(3) 略。

第4章　类和对象

1. public,protected,private,private protected。

2. 创建新对象时执行构造函数,由系统自动调用。

3. 属性、方法。

4. 静态属性是类的属性,不专属于某个方法,类对象可以访问或修改静态属性。

5. 方法的重载。不同类的同名方法用类名区分,同类的同名方法用形式参数数目、顺序和类型区分。

6. 包是相关类的松散集合。使用 package 语句可以创建包。使用包可以方便相关的类共同工作,也方便其他包中的类引用它们。包物理地对应文件夹,其中应保存包中类的字节码文件。

7. package MyPackage;应该在程序第一句。

8. import MyPackage.*;
import MyPackage.MyClass1;

9. 程序的输出为:

```
the original data is:-1
now the data is:10
```

10. 设计提示:

(1) 秒表类 Stopwatch 的属性

m_iMinute:整型量,代表分针指示的时间数。

m_iSecond:双精度型量,代表秒针指示的时间数。

(2) 秒表类 Stopwatch 的方法

startCount:无形式参数无返回值的方法,实现开始计时的功能。

stopCount:无形式参数无返回值的方法,实现停止计时的功能。

reset:无形式参数无返回值的方法,实现分针秒针归零的功能。

getMinute:无形式参数的方法,返回整型量,实现获得分针示数的功能。

getSecond:无形式参数的方法,返回双精度型量,实现获得秒针示数的功能。

getTotalTime:无形式参数的方法,返回整型量,实现获得总时间的毫秒数的功能。

toString:无形式参数的方法,把当前秒表指示的时间用"分数:秒数:毫秒数"的格式组合成字符串返回。

编程实现这个类,各方法内容可以先空缺,有能力的读者也可以尝试实现这些方法。(提示:java.util.Date类的对象可以表示当前时间,其方法getTime()可以返回当前时间距某一个固定时间点的毫秒数,可以用来在程序中辅助计时。)

11. 设计提示:

(1) 复数类ComplexNumber的属性

m_dRealPart:实部,代表复数的实数部分。

m_dImaginPart:虚部,代表复数的虚数部分。

(2) 复数类ComplexNumber的方法

ComplexNumber(double r,double i):构造函数,创建复数对象的同时完成复数的实部、虚部的初始化,r为实部的初值,i为虚部的初值。

getRealPart():获得复数对象的实部。

getImaginPart():获得复数对象的虚部。

setRealPart(double d):把当前复数对象的实部设置为给定的形式参数的数字。

setImaginaryPart(double d):把当前复数对象的虚部设置为给定的形式参数的数字。

complexAdd(ComplexNumber c):当前复数对象与形式参数复数对象相加,所得的结果也是复数值,返回给此方法的调用者。

complexMinus(ComplexNumber c):当前复数对象与形式参数复数对象相减,所得的结果也是复数值,返回给此方法的调用者。

complexMulti(ComplexNumber c):当前复数对象与形式参数复数对象相乘,所得的结果也是复数值,返回给此方法的调用者。

toString():把当前复数对象的实部、虚部组合成a+bi的字符串形式,其中a和b分别为实部和虚部的数据。

源代码如下:

```
//ComplexNumberTes.java
class ComplexNumber{
    private double m_dRealPart;
    private double m_dImaginPart;
    public ComplexNumber(double r,double i){
        m_dRealPart=r;
        m_dImaginPart=i;
    }
    public double getRealPart(){
        return m_dRealPart;
    }
```

```
    public double getImaginPart(){
        return m_dImaginPart;
    }
    public void setRealPart(double d){
        m_dRealPart = d;
    }
    public void setImaginPart(double d){
        m_dImaginPart = d;
    }
    public ComplexNumber complexAdd(ComplexNumber c){
        this.m_dRealPart += c.m_dRealPart;
      this.m_dImaginPart += c.m_dImaginPart;
        return this;
    }
    public ComplexNumber complexMinus(ComplexNumber c){
        this.m_dRealPart -= c.m_dRealPart;
      this.m_dImaginPart -= c.m_dImaginPart;
        return this;
    }
    public ComplexNumber complexMulti(ComplexNumber c){
        this.m_dRealPart *= c.m_dRealPart;
      this.m_dImaginPart *= c.m_dImaginPart;
        return this;
    }
    public String toString(){
        return m_dRealPart + "+" + m_dImaginPart + "i";
    }
}
public class ComplexNumberTest{
    public static void main(String args[]){
        ComplexNumber c1,c2;
        c1 = new ComplexNumber(3,4);
        c2 = new ComplexNumber(1,2);
        System.out.println("c1 + c2 = " + c1.complexAdd(c2));
        c1 = new ComplexNumber(3,4);
        c2 = new ComplexNumber(1,2);
        System.out.println("c1 - c2 = " + c1.complexMinus(c2));
        c1 = new ComplexNumber(3,4);
        c2 = new ComplexNumber(1,2);
```


```
            System.out.println("c1 * c2 = " + c1.complexMulti(c2));
        }
    }
```

12. 源代码如下：

```
//BoxTest.java
import java.text.DecimalFormat;
import javax.swing.JOptionPane;
class Box{
    private double length;
    private double width;
    private double high;
    public Box(){
        length = 1;
        width = 1;
        high = 1;
    }
    public Box(double a,double b,double c){
        if(a<=0||b<=0||c<=0){
            System.out.println("invalid input");
            System.exit(0);
        }
        else{
            length = a;
            width = b;
            high = c;
        }
    }
    public double getVolume(){
        return(length * width * high);
    }
    public double getLength(){
        return length;
    }
    public double getWidth(){
        return width;
    }
    public double getHigh(){
        return high;
    }
```

```
}
public class BoxTest{
    public static void main(String args[]){
        Box box=new Box(1.5,2.4,3);
        DecimalFormat twoDigits=new DecimalFormat("0.00");
        String output="The length of box is:"+twoDigits.format(box.getLength())
                +"\nThe width of box
                is:"+twoDigits.format(box.getWidth())
                +"\nThe high of box is:"+twoDigits.format(box.getHigh())
                +"\nThe volume of box
        JOptionPane.showMessageDialog(null,output,"Box
                Test",JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}
```

13. 设计思路:将班级总人数定义为静态变量,获得班级总人数定义为静态方法。可在构造函数中进行班级总人数的增加,在 finalize 方法中进行班级总人数的减少。

14. 设计思路:利用类的组合,将课程类数组作为学生类的一个属性,从而可以表示一个学生可以选择多门课程的情况。

第 5 章　类的继承和派生

1. 父类是所有子类的公共属性的集合,而每一个子类则是父类的特殊化,是在公共属性的基础上的功能、内涵的扩展和延伸。

2. 只有一个单一父类称为单重继承。一个类可以有一个以上的父类称为多重继承。

3.
```
import java.awt.*;
class MyFrame extends Frame
```

4. 这个属性是从其父类那里继承来的。

5. 子类重新定义父类的同名属性可以实现同名属性共存,若此父类的同名属性又被其子类所隐藏,可以有更多的同名属性共存。

6. 使用子类定义的方法将操纵子类定义的属性;或在子类定义的方法中用 super 关键字操纵父类定义的同名属性;使用父类定义的方法将操纵父类定义的属性。

7. 方法的覆盖。父类方法不能再使用,它的内存位置已经被替代。

8. super.method1()。

9. 使用 this。使用 super。

10. 阅读下面的程序并写出类 MyClass3 的所有成员,并写明它们的定义来自哪里。

pro12(MyClass1),pro21(MyClass3),pro31(MyClass3)。

11. 源代码如下:

```
//MySquareTest.java
class MyRectangle{
    private double length;
    private double width;
    public MyRectangle(){
        this.length=1;
        this.width=1;
    }
    publiC MyRectangle(double length,double width){
        this.length=length;
        this.width=width;
    }
    public double getLength(){
        return length;
    }
    public double getWidth(){
        return width;
    }
    public void setLength(double length){
        this.length=length;
    }
    public void setWidth(double width){
        this.width=width;
    }
    public double getArea(){
        return length*width;
    }
    public String toString(){
        return "我是一个矩形,我的长为"+this.getLength()+"\t宽为"
            +this.getWidth()+"\t我的面积是"+this.getArea();
    }
}
class MySquare extends MyRectangle{
    private double sideLength;
    public MySquare(double sideLength){
        this.sideLength=sideLength;
    }
    public double getSideLength(){
        return sideLength;
```

```
    }
    public void setSideLength(double sideLength){
        this.sideLength = sideLength;
    }
    public double getArea(){
        return sideLength * sideLength;
    }
    public String toString(){
        return "我是一个正方形,我的边长为" + this.getSideLength() + "\t 我的
            面积是" + this.getArea();
    }
}
public class MySquareTest{
    public static void main(String args[]){
        MyRectangle mr1 = new MyRectangle(2,3);
        MySquare ms1 = new MySquare(4);
        System.out.println(mr1);
        System.out.println(ms1);
    }
}
```

12. 源代码如下：

```
//Letter.java
class Address{
    private String country;
    private String province;
    private String city;
    private String street;
    private String number;
    private String department;
    private String postCode;
    public Address(String ct,String pc,String cty,String strt,String num,
            String dep,String pst){
        country = ct;
        province = pc;
        city = cty;
        street = strt;
        number = num;
        department = dep;
        postCode = pst;
```