新编高等院校计算机科学与技术规划教材

Java（第3版）

面向对象程序设计(JDK 1.6)

张桂珠　张　平　陈爱国　主编

MIANXIANG DUIXIANG CHENGXU SHEJI(JDK 1.6)

内附光盘

北京邮电大学出版社
www.buptpress.com

新编高等院校计算机科学与技术规划教材

Java 面向对象程序设计(JDK 1.6)

(第 3 版)

张桂珠　张　平　陈爱国　主　编

北京邮电大学出版社
·北京·

内 容 简 介

本书是Java面向对象程序设计的一本经典教材，它将Java语言与面向对象程序设计的原理和方法相结合，使用Java最新类库，以大量实例详细介绍如何使用Java进行面向对象的程序设计、GUI的程序设计、网络通信应用的程序设计、数据库应用的程序设计和Web应用的程序设计。

本书可作为Java面向对象程序设计课程的教材，也可作为Java的GUI程序设计、网络应用程序设计、数据库应用开发、JSP的Web应用开发的入门和提高学习教材。本书的读者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学习教材。为方便人员学习，本书还配有电子课件、实例代码、习题答案与实验。

图书在版编目(CIP)数据

Java面向对象程序设计:JDK 1.6/张桂珠，张平，陈爱国主编. --3版. --北京:北京邮电大学出版社，2010.8
ISBN 978-7-5635-2366-5

Ⅰ. ①J… Ⅱ. ①张…②张…③陈… Ⅲ. ①JAVA语言—程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2010)第156622号

书　　名：Java面向对象程序设计(JDK 1.6)(第3版)
主　　编：张桂珠　张　平　陈爱国
责任编辑：张珊珊
出版发行：北京邮电大学出版社
社　　址：北京市海淀区西土城路10号(邮编:100876)
发 行 部：电话:010-62282185　传真:010-62283578
E-mail：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：北京忠信诚胶印厂
开　　本：787 mm×1 092 mm　1/16
印　　张：20.25
字　　数：532千字
印　　数：1—4 000册
版　　次：2010年8月第1版　2010年8月第1次印刷

ISBN 978-7-5635-2366-5　　定　价：36.50元

前　言

Java语言自从1995年Sun Microsystem推出以来，因其面向对象，跨平台，稳定，多线程，支持用户图形界面、数据库网络应用和Internet/Web功能，使得Java编程系统以史无前例的速度在世界范围内赢得了人们的认可。现在，在各个领域有很多软件都是采用Java语言开发的，Java语言已经成为软件开发人员的理想工具。

Java的核心是创新的Java编程语言和SDK软件开发工具包。通用的Java语言完全面向对象，通过构建软件对象来编程。面向对象编程(Object-Oriented Programming，OOP)使得构建、维护、修改和重用软件更加简单。SDK包含一套丰富的代码库和工具。同时，该语言和SDK使得编程更加方便和高效。

本书是Java面向对象程序设计的一本经典教材，它将面向对象程序设计的理论、技术、方法和Java语言相结合，使用Java最新类库，详细介绍如何使用Java进行面向对象的程序设计、GUI的程序设计、网络通信应用的程序设计、数据库应用的程序设计和Web应用的程序设计。全书内容注重理论、技术和实际应用一体化，内容覆盖面大，实用性强。本书可作为Java面向对象程序设计课程的教材，也可作为Java的GUI程序设计、数据库应用开发、JSP的Web应用开发的入门和提高学习教材。本书的读者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技术的自学者或相关培训人员的学习教材。为方便相关人员学习，本书还配有电子教学软件、实例代码、习题答案与实验。

全书共有15章。第1章介绍面向对象编程和设计的基本概念、结构和方法；第2章介绍Java语言概况，并引入几个Java入门程序，说明Java程序类型和Java开发环境的使用；第3章讲述Java语言基础知识，包括标识符、运算符、数据类型、表达式、语句、基本算法控制、方法和数组；第4章讲述类的定义、对象的创建和使用、包的创建和使用；第5章讲述类的继承机制及其程序设计；第6章讲述多态性技术，包括抽象类、接口、内部类的定义和一般性的程序设计；第7章介绍在编程中常用的Java实用包；第8章介绍图形和Java 2D；第9章讲述GUI组件和用户界面程序设计；第10章讲述异常的概念、声明、抛出和处理以及异常的程序设计；第11章讲述线程技术和多线程的程序设计；第12章讲述输入/输出流处理，应用输入/输出流对顺序文件、随机文件进行读写；第13章讲述Java网络技术和网络应用程序设计；第14章讲述JDBC技术和数据库的开发应用；第15章讲述JSP技术和Web应用程序设计。

本书内容丰富，在使用本书学习或教学时，可分成4个部分循序渐进地进行。第一部分(1～3章)是Java语言的基础知识，学习Java语言的语法和使用；第二部分(4～7章)是Java的面向对象编程技术和应用，训练OOP的程序设计的能力；第三部分(8～9章)是Java绘制图形、GUI组件技术和用户界面设计应用，训练开发系统中用户界面的设计能力；第四部分(10～15章)是综合训练Java系统的开发能力，包括网络通信应用开发能力、数据库的应用开

发能力和 Web 应用开发能力。通过本书的学习，对 Java 的认识和应用能力会提高到一个新的水平。

全书由张桂珠主编。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，还得到了丁宏伟、刘丽、韩亦强、陈爱国、姚晓峰、张平、王惠、马晓梅老师的协助和支持，在此一并致谢。

感谢读者选择使用本书，欢迎各位读者对本书提出批评和修改建议。我们将不胜感激，并在再版时予以考虑。作者联系方式：zhangguizhu@163.com。

编　者

目　录

第1章　面向对象程序设计概述

本章首先通过比较，介绍传统的面向过程与面向对象程序设计(Object-Oriented Programming，OOP)这两种方法的特点，接着介绍了面向对象编程和设计的基本概念、结构和方法，它们都是Java面向对象程序设计中最具有技术性和挑战性的内容。随着本书的深入，OOP的概念和技术将落实在今后的面向对象编程的应用实例中。本章提供了面向对象编程和设计的核心内容的初步阐述。

1.1　面向对象与面向过程程序设计

1. 面向过程程序设计方法

一个程序往往由多个模块构成，这些模块可以被单独设计、编码和测试，然后被组装成一个完整的程序。在面向过程的语言中，如C、Fortran或Pascal，这些模块就是过程或函数(function)，过程或函数组成了面向过程语言编制的程序单位。一个过程或一个函数就是一个语句序列，例如求两个整数的最大值，用C语言的函数表达如下：

```
int max(int num1,int num2) {
if (num1>num2)
    return num1;
else
  return num2;
}
```

面向过程的程序是由赋值语句、条件语句、循环语句和过程调用语句构成的语句序列。

面向过程的程序设计方法　采用自顶向下的功能分解法(top-down，functional decomposition)，即一个要解决的问题被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。而程序结构也按功能划分为若干个模块，这些模块形成一个树型结构。每个模块用一过程或函数实现，如C语言编制的程序是由main函数加若干个子函数组成，在main中调用子函数。图1-1说明了自顶向下的功能分解法与面向过程的程序结构的关系。

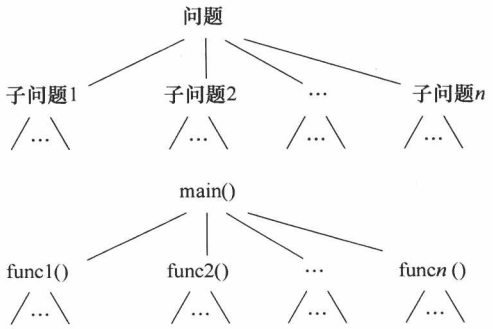

图1-1　自顶向下的功能分解与程序结构

自顶向下设计方法存在的问题：自顶向下设计方法把数据和处理数据的过程分离为相对

独立的实体，当数据结构改变时，所有相关的处理过程都要进行相应的修改，每一种相对于老问题的新方法都要重新设计程序，程序的可复用性差。因此，面向过程的程序设计方法的开发和维护都很困难。

2. 面向对象程序设计方法

面向对象程序设计是代替传统的面向过程程序设计的另一种程序设计方法，是为了解决过程型程序设计和自顶向下设计中的主要问题。程序的模块单位变成了类，类是比函数更大的单位，它将操作数据的函数和数据封装在一起而形成的。在 Java 语言中，像 max 这样的函数被叫做方法，方法变成了类的成员。

面向对象的语言有很多，如 Smalltalk、Ada、Object Pascal、C++、Java，其中 C++和 Java 是目前使用最多的语言。而 Java 与 C++语言相比，它去掉了 C++语言的复杂性和二义性的成分，增加了安全性和可移植性的成分。Java 比 C++也简单得多，而且文件尺寸小。Java 去除了头文件、预处理程序、指针运算、多重继承、运算符重载、结构体、共用体和模板。另外，Java 还可以自动执行无用内存单元收集，从而不需要显式进行内存管理。

面向对象的程序设计方法 首先，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。再对相同类型的对象进行分类、抽象后，得出共同的特性而形成了类。类中的大多数数据只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生联系。面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看做是对象。对象之间通过消息相互作用。另外，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。

面向对象的程序模块间的关系简单，程序的独立性高、数据安全。面向对象方法的显著特性有：封装性、抽象性、继承性和多态性，使软件具有可重用性，开发和维护的代价小。

1.2 类与对象

面向对象的设计，将客观事物（或实体）看做具有属性和行为（或称服务）的对象（object），通过抽象找出同一类型对象的共同属性（静态特征）和行为（动态特征），进而形成类（class）的概念。类是相同类型对象的集合的描述。例如，类 Human 就是现实世界中人（对象）的集合，我、你、他都是 Human 的对象。分析类 Human 的所有对象——人，得到这些对象共同的数据属性和行为：

- 数据属性：编号、姓名、年龄等；
- 行为：吃饭、走路、跳舞等。

定义 Human 类，如下所示：

```
class Human {
  int no;
  String name;
  int age; …
  void eat() {…}
  …
}
```

其中，Human 被称为类名。no、name、age 被称为变量或域（fields），eat()被称为方法（methods），它们都是类的成员。在面向对象的编程语言中，类是一个数据类型，对象是类的实例

(instance)。在 Java 中,具有类类型的变量被称为对象引用(object reference)。例如(如图 1-2 所示):

```
Human p1 = new Human(1,"张三",20,…);
//p1 被称为对象引用变量,new Human(1,"张三",20,…) 被称为 Human 的对象
```

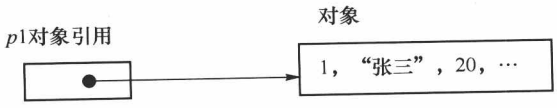

图 1-2　一个对象引用和一个对象

面向对象系统中的对象,是用来描述现实世界中实际存在的事物的实体,它是用来构成系统的一个单位。对象由一组属性和一组行为构成。具体地,对象是具有唯一对象名和固定对外接口的一组属性和操作的集合,用来模拟组成或影响现实世界问题的一个或一组因素。其中对象名是区别于其他对象的标志;对外接口是对象在约定好的运行框架和消息传递机制中与外界通信的通道;对象的属性表示了它所处于的状态;而对象的操作则是用来改变对象状态达到特定的功能。

1.3　封装与信息隐藏

封装(Encapsulation)是面向对象方法的重要原则,就是把对象的属性和操作(或服务)结合为一个独立的整体,并尽可能隐藏对象的内部实现细节。这里有两个含义:其一,把对象的全部属性和全部服务结合在一起,形成一个不可分割的独立单位;其二,"信息隐蔽",尽可能隐蔽对象的内部细节,对外界形成一个边界,只保留有限的对外接口,使之与外部发生联系。

数据封装的作用为:①对象的数据封装特性彻底消除了传统结构方法中数据与操作分离所带来的种种问题,提高了程序的可复用性和可维护性,降低了程序员保持数据与操作相容的负担;②对象的数据封装特性还可以把对象的私有数据和公共数据分离开,保护了私有数据,减少了可能的模块间干扰,达到降低程序复杂性、提高可控性的目的。

Java 语言中,定义类时通过大括号{ }封装了类的成员——域(变量)和方法,并使用 private 和 public 等关键字来控制对类的成员的访问,其中 private 修饰的成员是隐藏的,而 public 修饰的成员则定义了类对外的公共接口。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。

1.4　继　承

继承(Inheritance)支持着软件代码的复用,是提高软件开发效率的重要因素之一。继承是在已有类(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。图 1-3 说明了 Java 中的继承的一个例子:给定一个 Window 类,通过继承扩展它而得到一个子类 MenuWin。继承机制中,往往从一组类中抽象出公共属性放在父类。例如,给定类 Car、Motocycle 和 Truck,

```
class Window {
    …// Window 的成员定义
}
class MenuWin extends Window {
    …//MenuWin 新增加成员的定义
```

图 1-3　Java 继承的基本语法

把它们的公共属性放在一个称为 Vehicle 的公共父类中。

继承分为单继承和多继承。单继承是指一个子类最多只能有一个父类。多继承是一个子类可有两个以上的父类。由于多继承会带来二义性，在实际应用中应尽量使用单继承。Java 语言中的类只支持单继承，而接口支持多继承。

如何设计继承并完成继承层次是面向对象设计和编程的核心问题。继承是多态性的前提条件。

1.5 多态性

多态性(Polymorphism)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。

例如，定义一个几何图形类 Shape，它具有“画图”行为，用 draw()表示。但具体画什么图并不确定；定义 Shape 类的一些子类，如 Circle(圆)和 Rectangle(矩形)。在子类中“画图”的具体行为可重新定义为：圆类中 draw()画圆，矩形类中 draw()画矩形。定义 Shape s，s 作为引用变量可指向 Circle 圆类的对象，也可指向 Rectangle 矩形类的对象。

通过执行下面的代码：

```
s.draw();
```

s 调用 draw()方法，s 指向对象不同会画出不同的图形(圆或矩形)。

多态性也是泛指在程序中同一个符号在不同的情况下具有不同解释的现象。

面向对象的设计特性为抽象性、封装性、多态性和继承性，必须在实际的面向对象系统设计中体现。将面向对象设计方法应用于程序的开发工具和开发过程中，不仅可以加快开发的速度，还可极大地增强程序的可维护性和可扩展性，提高代码重用率。

1.6 面向对象的建模和 UML

统一建模语言(Unified Modeling Language，UML)是一种流行的建模语言，由对象管理组(Object Management Organization，OMO)发布。UML 是一种图形化语言，允许系统构造人员(软件设计师、系统工程师、程序员等)用一种通用表示法描述系统的需求以及面向对象的分析和设计结果。作为一个建模语言，UML 由一个用于表达模型的词汇表和一个定义怎样组合词汇的语法规则构成，即

UML＝UML 词汇表＋UML 建模的语法规则

UML 词汇表＝UML 事物＋关系＋模型图

UML 事物(Thing)就是被模拟的实体或对象。事物可能是包、类、接口等。事物之间的语义上的联系用关系表示，UML 中共有 4 种关系：依赖关系、关联关系、泛化关系和实现关系。从软件的体系结构出发，UML 把软件模型分成了 4 个视图：用例视图、逻辑视图、实现视图和分布视图。在本书的有关章节中，案例分析和设计的结果用 UML 图形化表示，使读者对 UML 有一个感性化认识。有关 UML 更完整的文档请参阅 Web 站点(www.omg.org)。