高等工科院校电子、信息类教材

C++与数据结构

C++ and Data Structures

高 飞 聂 青 李蕙芳 薛艳明 编著

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

高等工科院校电子、信息类教材

C++与数据结构

高 飞 聂 青 李蕙芳 薛艳明 编著

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 提 要

本书系统地介绍了面向对象的封装性、继承性和多态性以及C++程序设计基础。在此基础上，采用面向对象的思想和抽象数据类型的概念，用C++语言有效地组织和描述了线性表、数组、串、栈、队列、树和图等各种常用的数据结构的相关类及其实现，并介绍了每一种数据结构的不同存储方法、典型操作及其应用。

本书分两大部分，共15章。第一部分介绍面向对象的概念及C++程序设计基础；第二部分介绍各种常用数据结构的C++抽象类及其实现。

本书可作为高等院校电子信息类专业以及其他相关专业本科生的教科书，也可供从事程序设计的工程人员参考使用。

图书在版编目（CIP）数据

C++与数据结构/高飞等编著. —北京：北京理工大学出版社，2006.9
高等工科院校电子、信息类教材
ISBN 7-5640-0853-9

Ⅰ.C… Ⅱ.高… Ⅲ.①C语言-程序设计-高等学校-教材②数据结构-高等学校-教材 Ⅳ.①TP312②TP311.12

中国版本图书馆CIP数据核字（2006）第096707号

出版发行/ 北京理工大学出版社
社　　址/ 北京市海淀区中关村南大街5号
邮　　编/ 100081
电　　话/ （010）68914775（办公室） 68944990（批销中心） 68911084（读者服务部）
经　　销/ 全国各地新华书店
印　　刷/ 北京国马印刷厂
开　　本/ 787毫米×1092毫米　1/16
印　　张/ 27
字　　数/ 640千字
版　　次/ 2006年9月第1版　2006年9月第1次印刷
印　　数/ 1～5000册　　责任校对/ 郑兴玉
定　　价/ 36.00元　　责任印制/ 刘京凤

前　言

面向对象技术使基于 C++程序设计语言的软件开发技术得到了迅速发展。C++是一种混合型的面向对象程序设计语言，它既具有独特的面向对象特征，可以为面向对象技术提供全面支持，又具有对 C 语言的向后兼容性，使很多已有的程序稍加修改就可以重用，许多有效的算法也可以继续利用。就软件产品而言，最重要的就是建立合理的软件体系结构、程序结构和设计有效的数据结构。因此，数据结构已经成为计算机程序设计的重要理论基础。本书力求以算法为中介，使学习程序语言和学习数据结构共进。因为语言只有满足算法的需要，才能被认识和掌握，数据结构只有依赖语言的发展才能拓展自己的应用领域。

对于信息类非计算机专业的本科生，有必要在学习《C 语言程序设计》的基础上，进一步学习、掌握面向对象的 C++程序设计和有效组织各种数据在计算机中的存储、传递和转换的方法，以提高程序设计和软件开发能力。本书正是针对这种需要，既介绍了面向对象的 C++程序设计方法，又将系统的数据结构融合到面向对象的方法中。

本书系统地介绍面向对象的 C++程序设计方法和数据结构的 C++描述，并通过具体的实例分析，使读者能够真正掌握数据结构应用于面向对象设计的方法，从而更好地从事软件开发和工程应用。

全书分为两部分，共 15 章。

第一部分从第 1 章到第 6 章，是面向对象的程序设计基础。第 1 章主要介绍面向对象的思想、概念和基本特征以及 C++的初步知识；第 2 章主要介绍了 C++类及其对象的封装性，重点介绍了类、类成员函数和对象的定义方式，以及构造函数与析构函数的作用与定义方式；第 3 章主要介绍 C++中的几种常用语法和一些特殊用途，包括函数和运算符的重载、友员和引用的定义和使用；第 4 章主要介绍继承与派生类的定义和使用，包括单继承与多继承、继承方式及基类数据在派生类中访问权限的变化情况，并引入了虚基类的概念及其用途；第 5 章主要介绍多态性与虚函数在 C++中的体现，重点讲述运行时的多态性，包括虚函数的定义和使用，并引入了纯虚函数和抽象类的概念；第 6 章主要介绍模板的概念、模板的定义以及使用模板函数和模板类的方法及模板函数重载和模板类派生类的方式。

第二部分从第 7 章到第 15 章，是用面向对象方法与 C++描述的数据结构。第 7 章主要介绍数据结构的定义及抽象数据类型的概念，并讨论了数据结构所研究的逻辑结构、存储结构和算法的内容，简要介绍了数据结构的分类及其抽象的层次；第 8 章至第 11 章依次介绍了线性表、数组、串、栈和队列的概念、抽象类的描述及其不同存储表示的各种类的具体定义、实现和应用；第 12 章至第 13 章分别介绍了树和图的基本概念和多种存储表示，并给出了相应抽象类的定义、实现和应用；第 14 章主要介绍了查找与散列表的概念、散列函数的设计方法及解决冲突的多种策略；第 15 章主要介绍了排序的概念、各种典型排序的算法和应用。

本书可作为高等院校电子信息类专业以及其他相关专业本科生的教科书，也可供从事程

序设计的工程人员参考使用。

本书第1章至第3章由李慧芳编写，第4章至第7章由薛艳明编写，第8章至第11章由聂青编写，第12章至第15章由高飞编写，并由高飞、薛艳明统稿。

本书在编写过程中得到了苏广川教授的大力支持和悉心指导，在此谨致谢意。在本书的出版过程中，得到了北京理工大学出版社的大力支持和热心帮助，在此表示衷心的感谢。

由于编者水平有限，不足之处在所难免，诚恳地希望读者批评指正。

编 者

目　录

第一部分　面向对象的C++程序设计基础

第二部分 数据结构——用面向对象方法与C++描述

第一部分
面向对象的C++程序设计基础

第1章　面向对象的设计方法

面向对象的设计方法是当前最有效、最实用和最流行的软件开发技术，是一种利用类、对象、继承性、封装性、消息传递和多态性等概念来构造系统的软件开发方法。本章首先引入了面向对象的思想；其次介绍了面向对象的基本概念（对象、消息和类）与面向对象的基本特征（封装性、继承性和多态性）；最后简要介绍了有关C++语言的初步知识。

1.1　面向对象的思想

面向对象方法是为计算机软件的创建提出的一种模型化世界的抽象方法，其基本思想是，尽可能地运用人类的自然思维方式来建立问题空间的模型，构造尽可能直观、自然地表达问题求解方法的软件系统。现实世界中的问题是由客观实体和实体之间的联系构成的，对象就是客观实体的抽象。面向对象方法将数据和操作放在一起，作为一个相互依存、不可分割的整体来处理。

为了理解面向对象的含义，请看一个现实世界对象的例子——学生。“本科生”是更大对象类“学生”的一个成员（称为“子类”）。所有的学生都有一些共同的属性，如姓名、性别、年龄、学习成绩、学分等，这些属性对于类“学生”的成员，例如研究生、本科生、专科生等总是可以使用的。因为本科生是类“学生”的成员，“本科生”继承了类“学生”所定义的一切属性和操作。

一旦某个类被定义，当该类的一个新子类被创建时，属性可以被复用。例如定义一个新的称为“研究生”的对象，它也是类“学生”的成员，“研究生”也继承了“学生”的所有属性，如姓名、性别、年龄、学习成绩等。

在类“学生”的每个对象上可以定义一系列操作，例如入学注册、选课等。这些操作将修改对象的一个或多个属性，例如选课操作将修改学习成绩和学分这两个属性的值。软件工程专家用如下等式简明地描述“面向对象”：

面向对象 = 对象+分类+继承+消息通信

也就是说，面向对象就是既使用对象又使用类和继承等机制，而且对象之间仅能通过消息的

传递实现通信。如果一个软件系统使用了这四个概念来设计和实现，这个软件系统就是面向对象的。

1.1.1 面向对象的程序设计

在面向对象的程序设计之前，占主流的是结构化的程序设计，也就是面向过程的程序设计。例如：Fortran、Pascal 以及最初的 C 语言都是面向过程的程序设计。结构化程序设计的主要思想是自顶向下、逐步求精。程序结构是按功能划分为若干个模块，各模块之间的关联尽可能地少，各个模块的功能相对独立，这些模块联合形成一个树状结构。虽然结构化程序设计方法有很多优点，但它仍然是一种面向过程的程序设计方法，将数据和处理数据的操作相分离。当数据结构发生变化时，所有处理数据的相关操作都要进行相应的修改，不适合编制大型的软件系统。

1.1.2 面向对象的语言

面向对象的程序设计语言必须支持抽象的数据类型和继承性。面向对象的程序设计语言经历了一个漫长的发展过程。例如：LISP 语言、Simula67 语言和 SmallTalk 语言，它们都或多或少地引入了面向对象的一些概念，如数据抽象、类结构与继承机制。目前，应用最广泛的面向对象程序设计语言是在 C 语言基础上扩充出来的 C++语言。C++对 C 的向后兼容，使得很多基于 C 的程序稍加修改就可以重用，许多有效的算法也可以重用。C++是一种混合型的面向对象程序设计语言，它的出现使得面向对象的各种语言越来越多地得到重视和广泛应用。

1.2 面向对象的基本概念

对象是面向对象程序设计的核心，正确地认识和定义对象，对进一步掌握面向对象的理论大有帮助。

1.2.1 对象

1. 对象的定义

对象是现实世界中存在的一个事物，对象可以是具体的有形物体，如学生、房屋、汽车；也可以是无形的事物或概念，如国家、生产计划。对象是构成世界的一个独立单位，它具有自己的静态特征（也称属性，描述内部状态的数据）和动态特征（具有的功能或行为）。从系统建模和实现的角度，对象描述了客观事物的一个实体，是构成系统的一个独立的基本单元，它由一组属性（数据）和一组服务（操作）构成。系统的服务是通过新对象的建立和对象之间的消息通信来完成的。对象中的服务（操作）是对象动态特征的体现，它常是一个可执行的语句或过程，对属性进行操作，实现某种服务。从系统实现的角度，可以把对象看作是由一组数据以及有权对这些数据施加操作的一组服务封装在一起构成的统一体，即：

对象 = 数据+动作（方法、操作）

在面向对象程序设计中的对象包含两个方面的含义：对象的属性和它的行为。属性是指对象的自然属性或自身状态，例如人的年龄、身高、肤色和体重等。行为是指对象的功能，

例如人的特长或技能等。对象是其自身所具有的特征以及可以对这些状态施加的操作结合在一起构成的一个独立实体，它具有以下特征：

（1）有一个名字以区别于其他对象；

（2）有一个状态用来描述它的一些属性；

（3）有一组操作，每一个操作决定对象的一种功能或行为。

对象的操作可以分为两类：一类是自身所承受的操作，另一类是施加于其他对象的操作。

例如有一个人叫李玉，身高 1.65 m，体重 60 kg，可以讲授高等数学，会程序设计。下面是这个对象的特征描述：

对象名：李玉

对象的状态：

身高 1.65 m

体重 60 kg

对象的功能：

回答身高
回答体重 } 自身所承受的操作

讲授高等数学课
会程序设计 } 施加于其他对象的操作

2. 对象的特性

对象具有以下三个特性：

（1）模块的独立性。利用封装技术将对象的特性隐藏在模块内部，使用者只需了解它的功能，不必知道功能实现的细节。因此，外界的变化不会影响模块内部的状态，各模块可独立为系统所组合选用，也可被程序员重用。

（2）动态的连接性。可以通过消息激活机制，将对象之间的动态联系连接在一起，使整个机体运转起来。

（3）易维护性。由于对象的数据和操作都被封装在模块内部，所以对它们的修改及完善都限制在模块内部，不会涉及对象外部，从而使整个对象和整个系统变得更易于维护。

1.2.2　消息

消息（Message）是对象之间在交互中所传送的通信信息。在 C++中，消息的具体表现为函数，它是对象之间相互请求和协同工作的手段，是请求某个对象执行其中某个功能操作的规格说明，对象之间的联系只能通过传递消息来进行。只有当对象接收到消息时，才会激活有关的对象代码，“知道”如何去操作它的私有数据以完成所要求的服务操作。

消息具有三个性质：

（1）同一对象可接收不同形式的多个消息，并产生不同的响应；

（2）相同形式的消息可以传递给不同类型的对象，所产生的响应可能截然不同；

（3）消息的发送可以不考虑具体的接收者，对象可以响应消息，也可以不响应消息。

在面向对象系统中，为了完成某个事件，有时需要向同一个对象发送多个消息或者向不同的对象发送多个消息，把这多个消息称为消息序列。对一个对象而言，由外界对象直接向它发送的消息称为公有消息；由它自己向本身发送的消息称为私有消息。

某个特定对象的消息，根据功能的不同可分为以下三种类型：

（1）可以返回对象内部状态的消息；

（2）可以改变对象内部状态的消息；

（3）可以完成一些特定操作，并改变系统状态的消息。

1.2.3 类

类（Class）是一种具有相同属性和相同操作的对象的集合。一个类就是对一组相似对象的共同描述，它整体地代表一组对象。类封装了对描述某些现实世界对象的内容和行为所需要的数据和服务（操作）的抽象，它给出了属于该类的全部对象的抽象定义，包括类的属性、服务（操作）。对象只是符合某个类定义的一个实体，属于某个类定义的一个具体对象称为该类的一个实例（Instance）。可以把类看成是某些对象的模板（Template），它抽象地描述了属于该类的全部对象的共有属性和操作。类与对象的关系是抽象与具体的关系，类是多个对象（实例）的抽象，对象（实例）是类的个体实物。例如：王洋是一个教师。教师是一个类，属于抽象的概念；而王洋是教师类的一个具体对象，即教师类的实例。

面向对象的系统设计主要归结为类的建立和使用，类的确定采用归纳法来完成。在系统设计中，通过对相同性质的物质对象进行类比分析，归纳出它们的共性，包括数据特性和行为特性，构建一个类。C++的类是对所研究问题的若干对象的抽象描述，它对逻辑上相关的数据和函数进行封装。

1.3 面向对象的基本特性

在面向对象的程序设计中，三个核心的概念是：封装性、继承性和多态性。

1.3.1 封装性

封装（Encapsulation）是将一段程序代码“包装”起来，应用时只需要知道这段代码所完成的功能，而不必知道该功能的实现细节。类和对象是实现封装的重要机制。类是对具有相同属性的客观对象的抽象描述，并将抽象出来的数据和操作行为封装在类中，构成一个不可分割的、独立的整体。在类中将一部分代码作为对外部的接口，而将数据和其他行为尽可能隐蔽。外界只能通过外部接口才能访问封装在类中的数据，从而实现了对数据访问权限的有效控制。

封装的目的是将对象的设计者和对象的使用者分开。使用者只需要知道对象所表现的外部行为，利用设计者提供的消息来访问该对象，而不必了解对象行为的内部实现细节。封装作为面向对象方法的一种信息隐蔽技术，应具有以下几个条件：

（1）有一个清楚的边界。对象的所有私有数据和服务（操作）都被限定在该边界内，外界是不可直接访问的。

（2）至少有一个接口。这个接口描述了该对象与其他对象之间的相互请求和响应的消息格式和功能。

（3）对象行为的内部实现细节是隐蔽的，即其他对象不能直接修改该对象所拥有的相关数据和程序代码。