

图书基本信息

书名：<<C/C++程序设计教程.面向对象分册>>

13位ISBN编号：9787121154898

10位ISBN编号：7121154897

出版时间：2012-1

出版时间：电子工业出版社

作者：郑秋生 主编

页数：287

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

前言 本书的主要作者都是有着丰富教学经验的一线教师，从事C/C++程序设计课程教学多年，深知学生在学习C++程序设计这门课程后，对程序设计方法、算法设计、调试程序、习题解答的茫然和问题，因此本书在介绍理论知识、相关概念和语言语法时，始终强调其在程序设计中的作用，使语言语法与程序设计相结合。

同类书籍大部分偏重于对语言语法和概念的介绍，虽然在书中有针对语法和知识点的程序实例，但学生对每章内容在实际程序设计中的作用缺乏了解，而本书每章后都附有针对性较强的应用实例分析，尽可能使初学者在学习每章的内容后，即能够独立设计程序、解决实际问题，而不至于无从下手。

本书有以下五个鲜明特点： 第一，改变了传统的教学模式。

先讲C语言程序设计，再讲C++对C语言的扩展、面向对象的程序设计。

本教材将C/C++语言的学习很好地融在一起，让读者把面向过程和面向对象的程序设计方法有机地结合在一起。

面向过程和面向对象两分册都统一使用Visual C++ 6.0编译器。

第二，改变了传统教材以语言、语法学习为重点的缺陷，本教材从基本的语言、语法学习上升到程序的“设计、算法、编程、调试”层次。

为了让学生更好地掌握程序开发思想、方法和算法，书中提供了大量简短精辟的代码，有助于初学者学习解决问题的精髓。

在每章后都有一节关于程序综合设计的内容，有一个或多个较大的程序，以帮助学生更好地掌握程序设计方法和解决实际问题的能力。

第三，教材强调程序的设计方法，大量例题配有流程图、N-S图和UML图，即突出程序的算法和设计，而不仅是语法和编程，培养学生程序设计能力和程序调试技能，养成好的编程习惯，为专业程序员的培养打下良好的基础。

第四，培养学生面向对象程序设计的能力，引导学生建立程序设计的大局观，帮助学生掌握从客观事物中抽象出C++类的方法。

通过系统的学习，使学生的编程能力上一个台阶，具备解决复杂问题的程序设计能力。

第五，根据当前实际大型软件项目开发的需要，加大了异常处理、模板等内容，新增STL标准模板库，并通过流行的UML工具设计C++类。

本教材编写充分考虑了目前应用型本科C/C++程序设计课程教学的实际情况和存在的问题。

第一，学生在大一阶段的基础课程较多，不可能投入过多的精力来学习本门课程；第二，学生对这门课学习的期望值很高，但对学习时可能遇到的困难估计不足；第三，学生现有的上机实践条件大大改善，特别有利于贯彻先进的精讲多练的教学思想；第四，学生学会了语言的语法，仍不具备解决实际问题的能力，学生的程序设计、算法设计、编程、调试的能力相对较差。

本教材作者正是考虑了学生的这些实际问题，从而精心编写了这一套面向应用型本科的C/C++程序设计教程，特别适合于分两个学期系统讲授C/C++程序设计。

第1学期讲授面向过程分册，第2学期讲授面向对象分册。

本面向对象分册共分8章，第1章到第3章主要阐述面向对象程序设计的重要概念，包括类和对象、继承与派生、多态性；第4章介绍输入/输出流技术；第5章主要介绍异常的概念、异常的产生及异常的处理机制；第6章和第7章介绍模板和STL标准模板库；第8章主要讲述面向对象的分析与设计方法，以实例的形式详细介绍如何用C++进行程序设计。

为了方便使用本教材的教师备课，我们还提供了配套的电子教案，公开放在网站上，供任课教师自由下载使用。

相信我们多年的教学经验会对广大师生的教和学有所帮助。

建议本分册的教学学时为60学时，其中理论教学为44学时，课内上机实践为16学时，课外上机不少于32学时。

本教材的编写得到了河南省计算机学会的大力支持，组织了河南多所高校编写了高等教育计算机学科“应用型”系列教材。

参编本教材的高校有中原工学院、郑州大学、河南科技大学、郑州轻工业学院。

本书由郑秋生任主编，第1章和附录由王海龙和夏敏捷编写，第2章由罗菁和潘惠勇编写，第3章由李晓宇编写，第4章由冀治航编写，第5章由郑秋生编写，第6章和第7章由王文奇编写，第8章由刘凤华编写。

全书最终由郑秋生修改并统稿。

郑州大学王黎明老师为本书提出改进意见，在此谨向他们表示衷心的感谢。

由于编者水平有限，时间仓促，书中难免有错，敬请广大读者批评指正，在此表示感谢。

作者E-mail：zqs@zzti.edu.cn。

编者 2011年9月

内容概要

《C/C++程序设计教程》系列教材分为面向过程和面向对象两个分册。

面向对象分册系统阐述了C++语言中面向对象程序设计的语法和思想。主要内容包括类与对象、继承与派生、虚函数与多态性、异常处理、模板和STL标准模板库等内容。书中通过流行的UML工具描述C++类，内容讲解清晰、实例丰富，力避代码复杂冗长，注重程序设计思想。简短的实例和UML图特别有助于初学者更好地理解、把握解决问题的精髓，帮助读者快速掌握面向对象程序设计的基本方法。

书籍目录

第1章 类和对象

1.1 面向对象程序设计概述

1.2 面向对象程序设计的基本概念

1.2.1 类

1.2.2 对象

1.2.3 封装与数据隐藏

1.2.4 继承

1.2.5 多态性

1.2.6 消息

1.3 类和对象的定义

1.3.1 类的定义

1.3.2 成员函数的定义

1.3.3 类对象的定义

1.3.4 对象成员的访问

1.3.5 类对象的内存分配

1.3.6 this指针

1.4 构造函数和析构函数

1.4.1 构造函数的定义

1.4.2 构造函数的重载

1.4.3 带默认参数的构造函数

1.4.4 析构函数

1.4.5 拷贝构造函数和默认拷贝构造函数

1.5 类和对象的进一步应用

1.5.1 堆对象

1.5.2 对象数组

1.5.3 类对象作为成员

1.5.4 面向对象程序中的常量

1.6 静态成员

1.6.1 静态数据成员

1.6.2 静态成员函数

1.7 友元函数和友元类

1.7.1 友元函数

1.7.2 友元类

1.8 string类

1.8.1 char型字符串

1.8.2 string型字符串定义

1.8.3 string类构造函数

1.8.4 string类成员函数

1.9 综合应用实例

习题一

第2章 继承与派生

2.1 继承与派生的基础知识

2.1.1 继承与派生的基本概念

2.1.2 派生类的定义

- 2.1.3 派生类的生成
- 2.2 类的继承方式
 - 2.2.1 公有继承
 - 2.2.2 私有继承
 - 2.2.3 保护继承
 - 2.2.4 继承方式的总结和比较
- 2.3 派生类的构造函数与析构函数
 - 2.3.1 简单派生类的构造函数
 - 2.3.2 析构函数
 - 2.3.3 复杂派生类的构造函数和析构函数
 - 2.3.4 派生友元类
- 2.4 基类与派生类的转换
- 2.5 多重继承
 - 2.5.1 多重继承的定义
 - 2.5.2 多重继承中的二义性问题
- 2.6 虚继承和虚基类
 - 2.6.1 虚继承和虚基类的定义
 - 2.6.2 虚基类及其派生类构造函数执行顺序
- 2.7 综合应用实例
- 习题二
- 第3章 多态性
 - 3.1 多态性的概念
 - 3.2 运算符重载
 - 3.2.1 运算符重载概述
 - 3.2.2 双目运算符重载
 - 3.2.3 赋值运算符重载
 - 3.2.4 单目运算符重载
 - 3.2.5 下标运算符重载
 - 3.2.6 类型转换运算符重载
 - 3.3 联编和虚函数
 - 3.3.1 静态联编和动态联编
 - 3.3.2 虚函数的引入
 - 3.3.3 虚函数的定义和多态性
 - 3.3.4 使用引用变量的多态性
 - 3.3.5 动态联编的要素——指针（引用）变量
 - 3.3.6 动态联编的工作机制
 - 3.3.7 虚析构函数
 - 3.4 纯虚函数和抽象类
 - 3.4.1 纯虚函数
 - 3.4.2 抽象类
 - 3.5 综合应用实例
- 习题三
- 第4章 输入/输出流
 - 4.1 输入/输出流的基本概念

4.2 输入/输出流类体系

4.2.1 流类库

4.2.2 标准流对象

4.3 输入/输出流的操作

4.3.1 输入/输出流的格式化

4.3.2 用流成员函数实现输入/输出

4.4 文件流和文件的输入/输出

4.4.1 文件流类与文件流对象

4.4.2 定义文件流对象

4.4.3 文件的打开和关闭

4.4.4 文本文件的输入/输出（读/写）

4.4.5 二进制文件的输入/输出（读/写）

4.4.6 文件的随机访问

4.5 字符串流

4.6 重载插入和提取运算符

4.7 综合应用实例

习题四

第5章 异常处理及命名空间

5.1 异常处理

5.1.1 异常的概念

5.1.2 异常处理机制

5.1.3 异常函数

5.1.4 标准C++库中的异常类

5.2 命名空间

5.2.1 命名空间的定义

5.2.2 命名空间的使用

5.2.3 标准命名空间std

5.2.4 无名空间

5.3 综合应用实例

习题五

第6章 模板

6.1 函数模板

6.1.1 函数模板语法

6.1.2 函数模板实例化

6.1.3 使用函数模板实例

6.2 类模板

6.2.1 类模板的语法

6.2.2 类模板实例化

6.2.3 派生类和类模板

6.3 综合应用实例

习题六

第7章 标准模板库STL介绍及应用

7.1 标准模板库STL的概念

7.1.1 什么是STL

7.1.2 STL组成部分

7.1.3 STL对C++的影响

7.2 容器（Container）

- 7.2.1 容器简介
- 7.2.2 容器的结构
- 7.2.3 容器的使用
- 7.3 迭代器 (Iterator)
- 7.3.1 输入迭代器
- 7.3.2 输出迭代器
- 7.3.3 前向迭代器
- 7.3.4 双向迭代器
- 7.3.5 随机存取迭代器
- 7.3.6 迭代器的使用
- 7.4 算法 (Algorithm)
- 7.4.1 算法和函数对象
- 7.4.2 算法分类介绍
- 7.5 综合应用实例
- 习题七
- 第8章 面向对象程序设计实例
- 8.1 图书管理系统需求分析
- 8.1.1 需求分析的任务
- 8.1.2 图书管理系统需求描述
- 8.1.3 图书管理系统需求
- 8.2 图书管理系统需求模型
- 8.2.1 图书管理系统用例图
- 8.2.2 图书管理系统用例规约
- 8.3 图书管理系统设计
- 8.3.1 分析类
- 8.3.2 顺序图
- 8.3.3 设计类图
- 8.3.4 系统结构设计
- 8.4 图书管理系统实现
- 8.4.1 类的定义
- 8.4.2 类的实现
- 8.4.3 用户界面设计
- 8.4.4 系统主函数
- 8.4.5 系统管理员功能模块
- 8.4.6 普通管理员功能
- 习题八
- 附录A 常用容器与算法介绍
- 附录B 统一建模语言 (UML)
- 参考文献

章节摘录

版权页：插图：面向对象方法中的对象，是描述系统中某一客观事物的一个实体，它是构成系统的一个基本单位。

对象由一组属性和一组行为构成。

属性是用来描述对象静态特征的数据项，而行为是用来描述对象动态特征的操作序列。

类和对象的关系就像模具与产品之间的关系，一个属于某类的对象称为该类的一个实例，如张三就是人这个类的一个实例，或是这个类的具体表现。

1.2.3 封装与数据隐藏封装是指将数据和代码捆绑在一起，从而避免外界的干扰和不确定性。

在C++中，封装是通过类来实现的。

类是描述具有相同属性和方法的对象的集合，定义了该集合中每个对象所共有的属性和方法。

封装也是面向对象方法中的一个重要原则，它把对象的属性和行为结合成一个独立的系统单位，并且尽可能地隐藏对象的内部细节。

这里有两层含义：第一是把对象的全部属性和全部行为结合在一起，形成一个不可分割的独立单元；第二是信息隐蔽，也就是尽可能隐蔽对象的内部细节，对外部世界形成一个边界或屏障，只保留有限的公用的对外接口，使之与外部世界发生联系。

1.2.4 继承继承（inheritance）是面向对象程序设计能够提高软件开发效率的重要原因之一，也是软件规模化的一个重要手段。

特殊类的对象拥有其一般类的全部属性和行为，称为特殊类对一般类的继承。

继承具有重要的现实意义，它简化了人们对于现实世界客观事物的认识和描述。

例如，人们认识了汽车的特征之后，再考虑小轿车时，因为知道小轿车也是汽车，于是认为小轿车具有汽车的全部一般特征，从而可以把精力用于发现和描述小轿车不同于一般汽车的独有的那些特征。

软件的规模化生产是影响软件产业发展的重要因素，它强调软件的复用性，也就是程序不加修改或进行少许修改，就可以用在不同的地方。

继承对于软件的复用具有重要意义，特殊类继承一般类，本身就是软件复用。

不仅如此，如果将开发好的类作为构件放到构件库中，在开发新系统时可以直接使用或继承使用。

编辑推荐

《C/C++程序设计教程:面向对象分册(第2版)》是高等教育计算机学科“应用型”规划教材。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>