

十五规划计算机系列教材



Visual C++程序设计 与实验指导

主 编：孔令德

副主编：张岳新 马 锐

兵器工业出版社

Visual C++ 程序设计与实验指导

主 编 孔令德

副主编 张岳新 马 锐

兵器工业出版社

内 容 简 介

C++通过类和对象等概念提供了对面向对象特征的全部支持,是学习可视化程序设计语言的基础。C++不仅功能强大,而且结构层次清晰,为许多初学者所钟爱。要想在较短的时间内提高C++语言程序设计水平,最快捷的方式莫过于简单了解基本概念后尽快上机实践。

本书共分两部分。第1部分学习指导包括第1章至第16章是基础理论部分,每章针对教学重点给出内容概括、典型例题分析和误点分析,为第2部分的上机实验提供了知识储备。第2部分实验指导包括17个精选实验是上机实践部分,每个实验针对教学难点指导学生一步步完成上机,在实践中领悟C++的深邃内涵。

本书可作为计算机专业学生的教材,也是报考计算机专业硕士研究生的参考书。

图书在版编目(CIP)数据

Visual C++程序设计实验指导 / 孔令德主编. —北京:
兵器工业出版社, 2004. 1
ISBN 7-80172-172-1

I. C... II. 孔... III. C语言—程序设计—高等学
校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 000779 号

出版发行: 兵器工业出版社
责任编辑: 张小洁
责任技编: 魏丽华
邮编社址: 100089 北京市海淀区车道沟 10 号
经 销: 各地新华书店
印 刷: 军事医学科学院印刷厂
版 次: 2006 年 2 月第 1 版第 3 次印刷
印 数: 3501—6500

封面设计: 李 晖
责任校对: 朴 喆
责任印制: 莫丽珠
开 本: 787×1092 1/16
印 张: 17.75
字 数: 450 千字
定 价: 28.00 元

计算机系列教材编辑工作委员会

主 任：闫达远

副主任：胡星光 李晓梅

委 员：（按姓氏笔画为序）

马星国 孔令德 王复兴 王 琰 李凤霞

李 梁 张 华 张岳新 陈立潮 苏春辉

梁建民 梁国栋 崔广才 薛 虹

前 言

C++是面向对象的程序设计语言。近年来，随着大型应用系统开发需求的增长，可视化面向对象技术的发展如火如荼。Visual C++、Visual Basic、Visual Foxpro、Power Builder、Java 等语言已成为最流行的前端开发工具。作为学习可视化面向对象技术的入门基础，C++语言已开始代替传统的 C 语言成为计算机教学语言。C++以类、对象、继承、封装、消息等概念提供了对面向对象特征的全部支持，又向下兼容了传统的 C 语言的结构化程序设计特征。直接从 C++开始学习计算机基本编程语言，可以较快地为掌握可视化的面向对象语言奠定良好的基础。

学习 C++语言的最快捷的方式就是在于简单掌握了基本概念后，尽快上机操作。本书在这方面提供了具体的指导。本书共分两部分：学习指导和实验指导。学习指导进行了知识概括、典型例题分析和错误提示，每章后面都有思考题和知识测试，学生可以根据学习进度自己检测知识掌握情况。实验指导则通过手把手指导学生上机完成具体实验题目，在实践中把握 C++的精髓。

编者根据多年讲授 C++的经验选定了本书的结构和内容。本书精心设计的大量例题都在 Microsoft Visual C++ 6.0 系统上调试通过。

本书由华北工学院分院计算机工程系孔令德博士担任主编，南京理工大学张岳新老师和北京理工大学马锐老师担任副主编，华北工学院分院王建国老师参加了本书的编写。

本书可作为大专院校计算机专业学生的教材。也可作为教师的参考书。

本书的编写得到华北工学院分院电子工程系田生喜老师的大力支持，华北工学院分院计算机工程系的赵美虹老师审阅了全书，编者在此一并表示感谢。

由于编者水平有限，书中难免存在不足之处，敬请读者指正。

作 者
2003.11

目 录

第1部分 学习指导

第1章 C++概述	(2)
1.1 面向对象方法概述	(2)
1.1.1 面向对象的主要概念	(2)
1.1.2 面向对象设计方法	(3)
1.2 C++的基本语法规则	(3)
1.2.1 编译预处理命令	(4)
1.2.2 标识符和关键字	(4)
1.2.3 变量命名规范	(4)
1.2.4 语句书写格式	(4)
1.2.5 分隔符	(4)
1.2.6 注释	(4)
1.2.7 函数	(5)
1.3 C++程序的编辑、编译和运行	(5)
1.4 典型例题分析	(5)
1.5 思考题	(7)
1.6 知识测试	(7)
第2章 数据类型、运算符与表达式	(9)
2.1 基本数据类型	(9)
2.2 常量和变量	(9)
2.2.1 常量类型	(9)
2.2.2 常量定义	(10)
2.2.3 变量定义	(10)
2.3 运算符	(10)
2.3.1 算术运算符	(10)



2.3.2	自增、自减运算符	(11)
2.3.3	赋值运算符	(11)
2.3.4	关系运算符	(11)
2.3.5	逻辑运算符	(11)
2.3.6	位运算符	(11)
2.3.7	逗号运算符	(11)
2.3.8	条件运算符	(12)
2.4	表达式	(12)
2.4.1	算术表达式	(13)
2.4.2	逻辑表达式	(13)
2.4.3	关系表达式	(13)
2.4.4	赋值表达式	(13)
2.4.5	条件表达式	(13)
2.4.6	逗号表达式	(13)
2.5	类型转换	(13)
2.5.1	隐式转换	(13)
2.5.2	显式转换	(14)
2.6	类型定义	(14)
2.7	典型例题分析	(14)
2.8	思考题	(17)
2.9	知识测试	(17)
第 3 章	简单的输入输出	(18)
3.1	cin 和 cout	(18)
3.1.1	cin 与提取运算符 >>	(18)
3.1.2	cout 与插入运算符 <<	(18)
3.1.3	I/O 流格式控制符	(18)
3.2	printf 和 scanf	(19)
3.2.1	printf 函数	(19)
3.2.2	scanf 函数	(21)
3.3	典型例题分析	(21)
3.4	思考题	(24)
3.5	知识测试	(24)
第 4 章	C++ 的流控制语句	(28)
4.1	顺序结构	(28)
4.1.1	表达式语句	(28)
4.1.2	空语句	(28)



4.1.3 复合语句	(28)
4.2 选择结构	(28)
4.2.1 if-else 选择结构	(28)
4.2.2 switch 选择结构	(29)
4.3 循环结构	(29)
4.3.1 while 语句	(29)
4.3.2 do-while 语句	(29)
4.3.3 for 语句	(30)
4.4 转向语句	(30)
4.4.1 goto 语句	(30)
4.4.2 break 语句	(30)
4.4.3 continue 语句	(30)
4.5 典型例题分析	(31)
4.6 思考题	(34)
4.7 知识测试	(34)
第 5 章 函数	(36)
5.1 函数的分类	(36)
5.2 函数的定义	(36)
5.3 函数的声明	(36)
5.3.1 函数原型	(37)
5.3.2 函数声明	(37)
5.4 函数的调用	(37)
5.4.1 形参和实参	(37)
5.4.2 函数的返回值	(37)
5.4.3 传值调用和引用调用	(38)
5.4.4 函数调用方式	(38)
5.5 嵌套调用	(38)
5.6 递归调用	(38)
5.7 内联函数和重载函数	(39)
5.7.1 内联函数	(39)
5.7.2 重载函数	(39)
5.8 存储类	(39)
5.8.1 变量存储类	(39)
5.8.2 函数存储类	(40)
5.9 编译预处理	(40)
5.9.1 宏定义	(40)
5.9.2 文件包含	(40)





5.9.3 条件编译	(40)
5.10 典型例题分析	(41)
5.11 思考题	(46)
5.12 知识测试	(46)
第 6 章 数组	(49)
6.1 一维数组	(49)
6.1.1 一维数组的声明	(49)
6.1.2 一维数组初始化	(49)
6.2 二维数组	(49)
6.2.1 二维数组的声明	(49)
6.2.2 二维数组初始化	(50)
6.3 多维数组	(50)
6.4 字符数组	(50)
6.5 数组作为函数参数	(51)
6.6 典型例题分析	(51)
6.7 思考题	(56)
6.8 知识测试	(56)
第 7 章 结构体	(58)
7.1 结构体	(58)
7.1.1 结构体类型和结构体变量的定义	(58)
7.1.2 访问结构体成员	(59)
7.1.3 结构体变量的初始化和赋值	(59)
7.1.4 位域的概念	(59)
7.2 共用体	(59)
7.2.1 共用体类型和变量的定义	(59)
7.2.2 共用体变量的赋值	(60)
7.3 枚举	(60)
7.3.1 枚举类型和变量的定义	(60)
7.3.2 枚举变量的赋值	(60)
7.4 典型例题分析	(60)
7.5 思考题	(65)
7.6 知识测试	(65)
第 8 章 指针和引用	(68)
8.1 指针的概念	(68)
8.1.1 什么是指针	(68)



8.1.2	指针的定义	(68)
8.1.3	指针的赋值	(68)
8.1.4	const 指针	(69)
8.1.5	指针的运算	(69)
8.1.6	指针的动态内存管理	(69)
8.2	数组指针和指针数组	(69)
8.2.1	数组元素的指针	(69)
8.2.2	指向一维数组的指针	(69)
8.2.3	指向二维数组的指针	(70)
8.2.4	指针数组	(70)
8.3	函数的指针	(70)
8.3.1	指针作为函数参数	(70)
8.3.2	函数指针	(70)
8.3.3	指针函数	(70)
8.4	引用的概念	(71)
8.4.1	引用的声明	(71)
8.4.2	引用做函数参数	(71)
8.4.3	引用做函数的返回值	(71)
8.4.4	用 const 限定引用	(71)
8.5	典型例题分析	(71)
8.6	思考题	(75)
8.7	知识测试	(75)
第 9 章	类和对象	(78)
9.1	类的概念	(78)
9.1.1	类的定义	(78)
9.1.2	定义类时注意的问题	(79)
9.2	对象的定义和成员表示	(79)
9.2.1	对象的定义格式	(79)
9.2.2	对象成员的表示方法	(79)
9.3	典型例题分析	(79)
9.4	思考题	(83)
9.5	知识测试	(84)
第 10 章	构造函数和析构函数	(87)
10.1	构造函数	(87)
10.2	拷贝构造函数	(87)
10.3	析构函数	(88)





10.4	典型例题分析	(88)
10.5	思考题	(94)
10.6	知识测试	(94)
第 11 章	继承和派生类	(97)
11.1	继承与派生	(97)
11.1.1	基本概念	(97)
11.1.2	派生类的定义与构成	(97)
11.2	访问控制	(98)
11.3	派生类的构造函数和析构函数	(98)
11.3.1	派生类的构造函数	(98)
11.3.2	派生类的析构函数	(100)
11.4	二义性问题	(100)
11.5	虚基类	(101)
11.6	子类型关系	(101)
11.7	典型例题分析	(102)
11.8	思考题	(105)
11.9	知识测试	(105)
第 12 章	类的其他特性	(109)
12.1	友元	(109)
12.1.1	友元函数	(109)
12.1.2	友元成员	(109)
12.1.3	友元类	(109)
12.2	虚函数	(110)
12.2.1	虚函数	(110)
12.2.2	纯虚函数	(111)
12.3	静态成员	(111)
12.3.1	静态数据成员	(111)
12.3.2	静态成员函数	(112)
12.4	常成员	(112)
12.4.1	常对象	(112)
12.4.2	常成员函数	(112)
12.4.3	常数据成员	(113)
12.5	典型例题分析	(113)
12.6	思考题	(117)
12.7	知识测试	(117)



第 13 章 运算符重载	(121)
13.1 基本概念	(121)
13.2 运算符函数重载的两种形式	(121)
13.2.1 成员函数运算符	(122)
13.2.2 友元函数运算符	(122)
13.2.3 两种重载形式的比较	(122)
13.3 典型例题分析	(123)
13.4 思考题	(127)
13.5 知识测试	(127)
第 14 章 输入/输出流类库	(130)
14.1 基本概念	(130)
14.1.1 流	(130)
14.1.2 流类库	(130)
14.1.3 预定义的流对象	(131)
14.2 标准设备的输入/输出流	(131)
14.2.1 输入流	(131)
14.2.2 输出流	(132)
14.2.3 插入符和提取符的重载	(132)
14.2.4 输入/输出的格式控制	(133)
14.3 文件流	(133)
14.3.1 文件的打开	(133)
14.3.2 文件的关闭	(133)
14.3.3 文件的随机访问	(134)
14.4 典型例题分析	(134)
14.5 思考题	(139)
14.6 知识测试	(139)
第 15 章 模板与异常处理	(142)
15.1 模板	(142)
15.1.1 基本概念	(142)
15.1.2 函数模板	(142)
15.1.3 类模板	(143)
15.2 异常处理	(144)
15.2.1 异常处理的基本思想	(144)
15.2.2 C++异常处理基础	(144)
15.2.3 异常处理的执行过程	(145)





15.2.4 异常处理中的构造和析构.....	(145)
15.3 典型例题分析.....	(146)
15.4 思考题.....	(151)
15.5 知识测试.....	(151)
第 16 章 MFC 程序设计.....	(156)
16.1 使用 MFC Appwizard 进行程序设计.....	(156)
16.2 MFC 类.....	(159)
16.3 典型例题分析.....	(161)

第 2 部分 实验指导

实验 1 熟悉 VC++ 的集成环境.....	(166)
实验 2 项目文件的建立.....	(172)
实验 3 循环程序设计.....	(178)
实验 4 函数的应用.....	(182)
实验 5 编译预处理和多文件的组织.....	(187)
实验 6 数组的应用.....	(193)
实验 7 结构体和枚举的应用.....	(196)
实验 8 指针的应用之一.....	(200)
实验 9 指针的应用之二.....	(206)
实验 10 类与对象的概念.....	(212)
实验 11 构造函数与析构函数的用法.....	(220)
实验 12 继承和派生类的应用.....	(229)
实验 13 友元函数及虚函数的应用.....	(234)
实验 14 运算符重载的应用.....	(246)
实验 15 标准设备与文件流的输入输出.....	(248)
实验 16 模板与异常处理.....	(251)
实验 17 MFC 菜单设计.....	(255)
附录 1 实验报告的格式和要求.....	(260)
附录 2 考试模拟题.....	(261)
参考文献.....	(270)

第1部分 学习指导

第1章 C++概述

C++称为“带类的C”，就是在C的基础上引入面向对象的机制而形成的一门程序设计语言，而C是面向过程的程序设计语言。C++几乎继承了C的所有特点，同时添加了面向对象的特征。C++既支持面向过程的程序设计，又支持面向对象的程序设计。面向过程的程序设计语言是基于功能分析的，以算法为中心的程序设计方法；面向对象的程序设计语言是基于结构分析的，以数据为中心的程序设计方法。面向对象的程序设计方法具有三大特征：封装性、继承性和多态性，其基本思想是尽可能模拟人类的自然思维方式来构造软件系统，不仅可以提高对用户需求的适应性，而且支持软件复用。可视化的面向对象语言（如 Visual C++、Visual Basic 等）是当前软件开发工具发展的主流方向，而学好 C++是基础。C++包括三个要素：类、对象和继承。

1.1 面向对象方法概述

软件工程学家 Coad/Yourdon 认为：面向对象=对象+类+继承+消息。如果一个计算机软件系统采用这些概念来建立模型并予以实现，就是面向对象的。

1.1.1 面向对象的主要概念

1. 对象

对象是一个实体，可以是现实世界中具体的物理实体或概念化的抽象实体。一个学校是对象，桌、椅是对象，规章制度也是对象。对象是一个封装数据（属性，静态特征）和操作（服务，动态特征）的实体，是构成系统的基本单元。

2. 类

类是具有相同属性和相同操作的对象的集合，是抽象数据类型的实现。类是创建对象的模板，给出了属于该类的全部对象的抽象定义。对象的抽象是类，类的实例是对象。在客观世界存在的是类的实例，即对象。

3. 封装

封装是指把对象属性和操作结合在一起，构成独立的单元，其内部信息对外界是隐蔽的，不允许外界直接存取对象的属性，只能通过有限的接口与对象发生联系。类是数据封装的工具，对象是封装的实现。类的访问控制机制体现在类的成员中，可以有公有成员、私有成员和保护成员。对于外界而言，只需要知道对象所表现的外部行为，而不必了解内部实现细节。封装体现了面向对象方法的“信息隐蔽和局部化原则”。

4. 继承

继承指子类（派生类）可以自动拥有父类（基类）的全部属性和服务。父类和子类是一般与特殊的关系。在定义一个子类时，可以把父类所定义的内容作为自己的内容，并加入若干新的内容。继承是面向对象语言的重要特性，提高了软件的可重用性。继承分为单重继承和多重继承。单重继承时，一个子类只有一个父类；多重继承时，一个子类可以有多个父类。单重继承构成的类之间的关系是树状结构，多重继承构成的类之间的关系是网状结构。

5. 消息

消息指对象之间在交互通讯中所传送的信息。消息由三部分构成：消息名、接收消息的对象标识和参数。一个对象向另一个对象发送消息请求某项服务，接收消息的对象响应该消息，进行所要求的服务，并把操作的结果返回给请求服务的对象。

6. 多态性

多态性是指在基类中定义的属性和服务被子类继承后，可以具有不同的数据类型和表现出不同的行为。当一个对象接收到一个请求进行某项服务的消息时，将根据对象所属的类，动态地选用该类中定义的操作。不同的类对消息按不同的方式解释。多态性机制不但为软件设计提供了灵活性，减少信息冗余，而且显著提高了软件的可复用性和可扩充性。重载是实现多态性的方法之一。C++语言可以定义虚函数，支持动态联编。动态联编是多态性的一个重要表现。

1.1.2 面向对象设计方法

面向对象方法由面向对象分析（OOA）和面向对象设计（OOD）构成。

1. 面向对象分析

对应5个层次：（1）识别对象；（2）识别类的结构；（3）确定主题；（4）定义属性；（5）定义服务。

2. 面向对象的设计

在面向对象分析的5个层次上确定4个组元：（1）设计问题域组元；（2）设计人机界面组元；（3）设计任务管理组元；（4）设计数据管理组元。

尽管面向对象分析和面向对象设计的定义有明显区别，但在软件开发过程中其界线是模糊的。许多分析结果可以直接映射成设计结果，而在设计中又往往会加深对系统的理解，进一步完善分析的结果。因此分析活动和设计活动是一个多次迭代的过程。面向对象方法无论在概念上还是表示方法上是一致的，保证了开发活动的平滑过渡。

1.2 C++的基本语法规则

C++程序的组成：注释部分（两种风格）、编译预处理部分（宏定义，文件包含和条件编译）、程序正文部分（类型定义、常变量定义、函数定义）。最终，程序源代码由ASCII码组成类似单词或词组的单元（词法单元），可以用任意的文本编辑器编辑，源代码中的空白（空格、Tab、回车换行）用来表示词法单元的开始和结束，除这一功能外其余空白将被忽略，但如果是字符串内部的空白（不含回车换行，或者说字符串内不能直接回车换行，需要使用转





义符)将作为字符串的一部分输出,不会忽略。

其中,由开发平台 Visual C++定义的词法单元被称为关键字,而编程人员定义的词法单元被称为标识符(含宏名、自定义类名、常量名、变量名和函数名)。关键字表有:ANSI 国际标准 C++关键字,VC++扩展关键字,X86 寄存器伪变量三部分,其中国际标准部分可以分为:C 兼容部分和 C++专有部分。

1.2.1 编译预处理命令

C++的编译预处理命令以“#”开头。C++提供了三类编译预处理命令:宏定义、文件包含和条件编译。

1.2.2 标识符和关键字

C++的标识符由字母、下划线和数字组成。标识符的第一个字符不能为数字,长度为 32 个字符,文件名只能识别前 8 个。标识符不能取为关键字,并且大小写敏感。同一个单词的不同大小写被编译器看做是不同的标识符。在实际使用时应尽量采用有意义的单词做标识符,做到见名知意。关键字又称保留字,是系统定义的一些特殊标识符,不允许程序员将他们作为一般标识符使用。在用 Visual C++的编译器编辑代码时,关键字能够自动显示为蓝色。

1.2.3 变量命名规范

大型程序中使用许多变量,如果不遵守规范的命名方式,每个变量的含义和作用不容易记住,可能会造成理解性的降低。Microsoft 公司提供了一套“匈牙利命名法”,为每个变量提供前缀,使得每个变量有姓有名,易于区别。如: `i_sum` 是用于结果累计的整型变量,“姓”: `i`,“名”: `sum`。推荐大型程序的开发者使用匈牙利命名法,以便于交流。

1.2.4 语句书写格式

语句以分号结束。短语句一行写一个语句;长语句可以分行,分行原则是不能将一个单词分开,也不要将用双引号括起的一个字符串分开。程序代码的布局对于程序的可读性有很大的影响,一般要求用缩排书写格式显示程序的逻辑结构。Visual C++中的代码布局要求很多,读者应当遵守:

- (1) 尽量使用 Tab 键对齐代码。
- (2) 使用花括号的时候,要注意位置匹配。
- (3) 同一行内不要写多个语句。

1.2.5 分隔符

分隔符被称为程序中的标点符号,是用来分隔单词或程序正文。常用的分隔符包括空格、逗号、分号、冒号和花括号。

1.2.6 注释

注释是编程人员和阅读者的“人-人”通讯方式。程序的注释一般分为序言性注释和功