

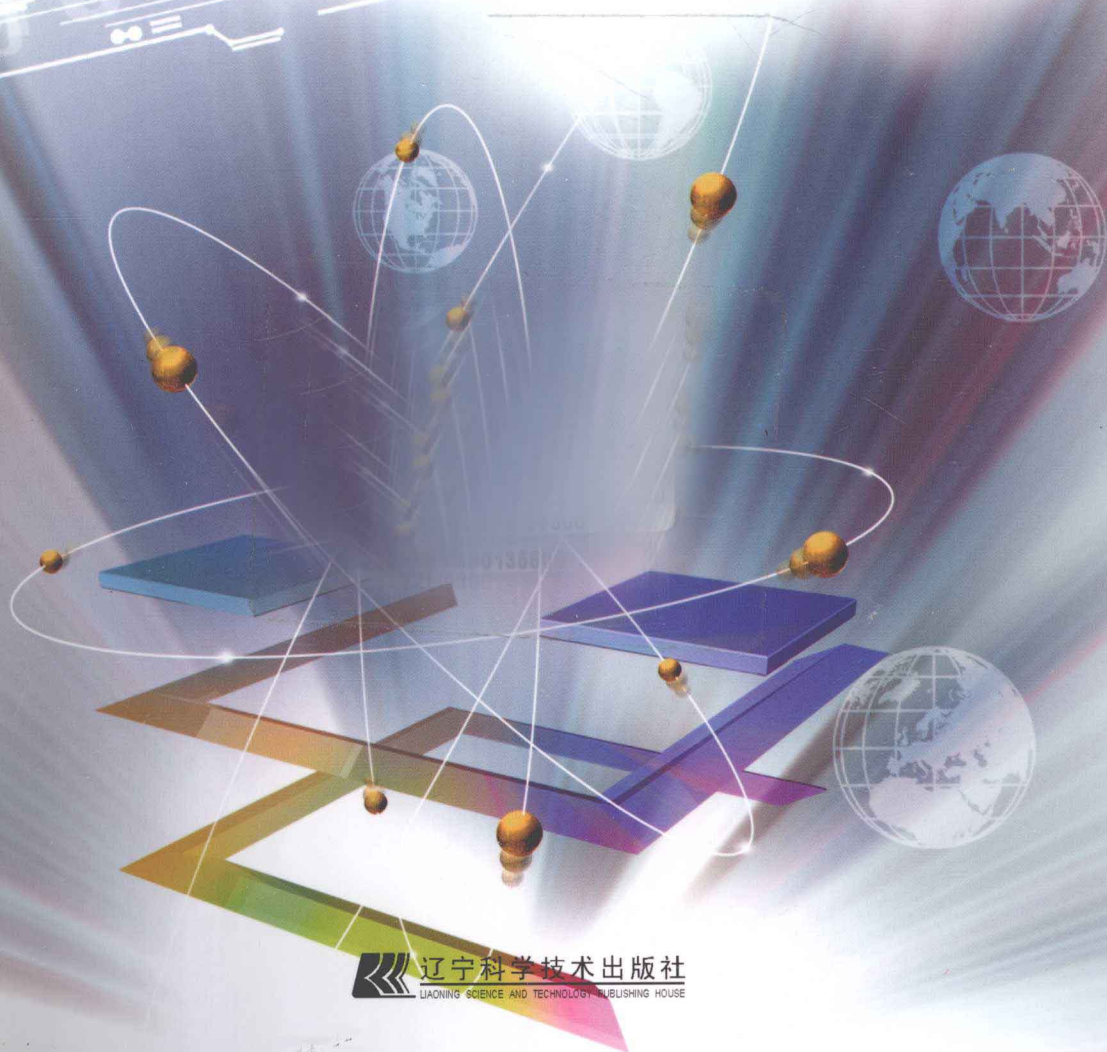
高等院校“十二五”规划教材·数字媒体技术
示范性软件学院系列教材



Visual C++ 开发技术及面向 对象软件工程 案例分析

丛书主编 肖刚强
副主编 高强 秦放

本书主编 邹丽
主审 田宏



辽宁科学技术出版社
LIAONING SCIENCE AND TECHNOLOGY PUBLISHING HOUSE

高等院校“十二五”规划教材·数字媒体技术
示范性软件学院系列教材

Visual C++开发技术 及面向对象软件工程 案例分析

丛书主编	肖刚强	
本书主编	邹丽	
副主编	高田	秦放
主审	田宏	

辽宁科学技术出版社
沈阳

丛书编委会

编委会主任: 孙 辉

顾 问: 徐心和 陈利平

副 主 任: 李 文

丛 书 主 编: 肖刚强

编委会成员: (按姓氏笔画为序)

于林林	王立娟	王艳娟	王德广	冯庆胜	史 原	宁 涛	田 宏	申广忠
任洪海	刘 芳	刘月凡	刘丽娟	刘瑞杰	孙淑娟	何丹丹	宋丽芳	张家敏
张振琳	张晓艳	李 红	李 瑞	邹 丽	陈 晨	周丽梅	郑 巍	侯洪凤
赵 波	秦 放	郭 杨	郭发军	郭永伟	高 强	威海英	雷 丹	翟 悦
魏 琦								

图书在版编目(CIP)数据

Visual C++开发技术及面向对象软件工程案例分析 / 邹丽

主编. —沈阳: 辽宁科学技术出版社, 2012.2

高等院校“十二五”规划教材·数字媒体技术/肖刚强主编

ISBN 978-7-5381-7232-4

I. ①V… II. ①邹… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第239227号

出版发行: 辽宁科学技术出版社

(地址: 沈阳市和平区十一纬路29号 邮编: 110003)

印 刷 者: 辽宁美术印刷厂

经 销 者: 各地新华书店

幅面尺寸: 185mm × 260mm

印 张: 14.75

字 数: 350千字

印 数: 1 ~ 3000

出版时间: 2012年2月第1版

印刷时间: 2012年2月第1次印刷

责任编辑: 于天文

封面设计: 赵苗苗

版式设计: 于 浪

责任校对: 刘 庶

书 号: ISBN 978-7-5381-7232-4

定 价: 30.00元

投稿热线: 024-23284740

邮购热线: 024-23284502

E-mail: lnkj@126.com

http://www.lnkj.com.cn

本书网址: www.lnkj.cn/uri.sh/7232

序 言

当前,我国高等教育正面临着重大的改革。教育部提出的“以就业为导向”的指导思想,为研究人才培养的新模式提供了明确的目标和方向,强调以信息技术为手段,深化教学改革和人才培养模式改革,根据社会的实际需求,培养具有特色显著的人才,是我们面临的重大问题。认真领会和落实教育部指导思想后提出新的办学理念 and 培养目标。新的变化必然带来办学宗旨、教学内容、课程体系、教学方法等一系列的改革。为此,组织学校有多年教学经验的专业教师,多次进行探讨和论证,编写出这套“数字媒体技术”专业的系列教材。

这套系列教材贯彻了“理念创新,方法创新,特色创新,内容创新”四大原则,在教材的编写上进行了大胆地改革。教材主要针对软件学院数字媒体技术等相关专业的学生,包括了多媒体技术领域的多个专业方向,如图像处理、二维动画、多媒体技术、面向对象计算机语言等。教材层次分明,实践性强,采用案例教学,重点突出能力培养,使学生从中获得更接近社会需求的技能。

本套系列教材在原有学校使用教材的基础上,参考国内相关院校应用多年的教材内容,结合当前学校教学的实际情况,有取舍地改编和扩充了原教材的内容,使教材更符合本校学生的特点,具有更好的实用性和扩展性。

本套教材可作为高等院校数字媒体技术等相关专业学生使用,也是广大技术人员自学不可缺少的参考书之一。

我们恳切地希望,大家在使用教材的过程中,及时给我们提出批评和改进意见,以利于今后教材的修改工作。相信经过大家的共同努力,这套教材一定能成为特色鲜明、学生喜爱的优秀教材。

肖刚强

前 言

Visual C++是Microsoft公司出品的一个功能非常强大的可视化编程工具，它以C++语言为基础，利用MFC类库的强大优势，使其成为Windows平台最为优秀的开发工具之一，也是目前功能最为强大的程序开发平台之一。MFC是采用Visual C++开发环境编写Windows应用程序的最佳选择，它是一组C++类，采用它可以为用户定制特定的Windows应用程序。

本书的目的是帮助广大Visual C++的学习者快速熟悉和掌握Visual C++工具并了解面向对象程序开发的基本技术和开发过程，使具有不同编程背景的人更加精通使用Visual C++进行Windows编程的技术。

本书内容共分为两大部分，总共11章。其中第1~8章属于第1部分，主要介绍Visual C++开发技术，内容涉及C++基础知识，在应用程序中使用控件、添加对话框、常用控件、菜单、工具栏和状态栏、图形、文本和字体、单文档和多文档应用程序、文本和字体、数据库编程等。第9~11章属于第2部分，概括介绍了面向对象软件工程的基本知识并给出了两个典型案例。

本书在编写过程中力求符号统一，图表准确，语言通俗，结构清晰。主要具有以下特点：

(1) 内容全面、系统、实用。介绍了最基本的概念和操作，难度适中，使读者在阅读过程中很顺畅，自然地了解并掌握MFC应用程序的基本知识。

(2) 实例丰富、实用。本书的实例步骤清晰、代码明确，案例章节涉及内容广泛，具有很强的实用价值。

(3) 语言通俗易懂、简洁明了。全书没有晦涩的字句，不但适合课堂教学，也适合读者自学使用。

本书既可以作为大专院校计算机科学与技术专业、软件工程专业学生的教材，也是广大工程技术人员自学不可缺少的参考书之一。

本书由邹丽、高强、秦放编写。由于编者水平有限，书中难免有疏漏和不足之处，恳请专家和广大读者指正。如果您在阅读时遇到问题或困难，可以直接与作者联系，E-mail: stu_zl@126.com。

如需本书课件和习题答案，请来信索取，地址：mozi4888@126.com

邹丽

目 录

第1章 C++基础	1
1.1 简单C++程序	1
1.2 类和对象	3
1.2.1 类的定义	3
1.2.2 对象的定义	5
1.3 初始化和清除	6
1.3.1 构造函数	6
1.3.2 析构函数	7
1.3.3 拷贝构造函数	8
1.4 继承和派生	9
1.4.1 单继承	10
1.4.2 多继承	12
1.4.3 派生类的构造函数和析构函数	12
1.5 多态和虚函数	14
1.5.1 虚函数	15
1.5.2 纯虚函数和抽象基类	16
1.6 模板	18
1.6.1 函数模板	18
1.6.2 类模板	19
1.7 本章小结	21
1.8 习题	21
第2章 在应用程序中使用控件	23
2.1 创建第一个项目	23
2.2 Windows的基本控件概述	27
2.3 在窗口中添加控件	27
2.3.1 创建应用程序	28
2.3.2 指定控件的Tab顺序	29
2.4 为控件添加变量	29
2.5 为控件添加功能	30
2.5.1 显示用户消息	32
2.5.2 清除用户消息	32
2.5.3 关闭应用程序	33
2.6 本章小结	33
2.7 习题	33
第3章 在应用程序中添加对话框	34
3.1 系统对话框	34
3.1.1 消息对话框	34

3.1.2	消息对话框编码	35
3.1.3	通用对话框	37
3.2	创建自己的对话框	40
3.3	在应用程序中使用对话框	42
3.4	本章小结	44
3.5	习题	44
第4章	常用控件	45
4.1	静态控件	45
4.2	编辑框控件	46
4.2.1	编辑框属性	46
4.2.2	编辑框基本操作	46
4.2.3	编辑框实例	47
4.3	按钮控件	52
4.3.1	命令按钮	52
4.3.2	单选按钮	53
4.3.3	复选按钮	53
4.3.4	按钮使用实例	53
4.4	列表框控件	57
4.4.1	列表框消息	57
4.4.2	列表框基本操作	57
4.4.3	列表框实例	59
4.5	组合框控件	62
4.5.1	组合框消息	62
4.5.2	组合框基本操作	63
4.5.3	组合框实例	64
4.6	滚动条控件	66
4.6.1	滚动条消息	66
4.6.2	滚动条基本操作	67
4.6.3	滚动条实例	67
4.7	日期时间控件	71
4.8	本章小结	74
4.9	习题	74
第5章	菜单、工具栏和状态栏	76
5.1	菜单	76
5.1.1	菜单风格简介	76
5.1.2	创建菜单	76
5.1.3	创建加速键	79
5.2	工具栏和状态栏	79
5.2.1	工具栏和状态栏简介	79
5.2.2	创建工具栏	79
5.2.3	创建状态栏	84
5.3	本章小结	85
5.4	习题	85

第6章 图形、文本和字体	86
6.1 图形设备接口	86
6.1.1 设备场景类	86
6.1.2 创建图形应用程序	87
6.2 文本和字体	92
6.2.1 显示可用字体	93
6.2.2 字体显示示例	95
6.3 本章小结	99
6.4 习题	99
第7章 单文档和多文档应用程序	100
7.1 文档/视图体系结构	100
7.2 单文档应用程序的创建	101
7.3 多文档应用程序的创建	107
7.4 本章小结	116
7.5 习题	116
第8章 数据库编程	117
8.1 使用ODBC连接数据库	117
8.1.1 ODBC概述	117
8.1.2 CDatabase类、CRecordset类和CRecordView类	118
8.1.3 创建ODBC数据库应用程序	119
8.2 使用DAO连接数据库	129
8.3 ADO技术	129
8.4 本章小结	134
8.5 习题	134
第9章 面向对象软件工程概述	135
9.1 生命周期模型	135
9.1.1 瀑布模型	136
9.1.2 快速原型模型	137
9.1.3 进化树模型	138
9.1.4 极限编程	138
9.1.5 螺旋模型	139
9.2 软件过程	141
9.2.1 五个核心工作流	141
9.2.2 二维生命周期模型图	143
9.3 面向对象的分析	144
9.3.1 面向对象分析的基本过程	144
9.3.2 建立对象模型	145
9.4 面向对象的设计	147
9.4.1 面向对象设计基本准则	148
9.4.2 设计的辅助工具	148
9.4.3 设计的度量	148
9.5 面向对象的实现	149
9.5.1 编程语言的选择	149
9.5.2 代码重用	152

9.6	维护	152
9.6.1	维护的定义	152
9.6.2	维护的类型	153
9.7	本章小结	153
9.8	习题	153
第10章	案例分析1: 图书管理系统	154
10.1	图书管理系统概述	154
10.2	可行性分析	155
10.3	系统分析	156
10.4	系统设计	157
10.5	系统数据库设计	158
10.6	系统物理配置方案设计	162
10.7	系统实现	163
10.8	系统测试及不足	212
10.9	本章小结	212
第11章	案例分析2: 绘制迷宫	213
11.1	迷宫问题概述	213
11.2	迷宫算法	214
11.3	绘制迷宫的程序实现	214
11.4	本章小结	225

第1章 C++基础

C++是美国贝尔实验室于1980年开发出来的一种过程性与面向对象性相结合的程序设计语言。最初被称为“含类的C”，到1983年才取名为C++。C++保持与C的兼容性而且用C++编写的程序可读性更好，代码结构更为合理，生成代码的质量高，其运行效率仅比汇编语言代码段慢10%~20%。从开发时间、费用到形成的软件的可重用性、可扩充性、可维护性和可靠性等方面有了很大的提高，使得大中型的程序开发项目变得容易得多。而且支持面向对象的机制，可方便地构造出模拟现实问题的实体和操作。

本章概括介绍C++面向对象程序设计语言的主要内容，重点讲解：

- 类和对象的创建
- 初始化和清除
- 继承和派生
- 多态和虚函数
- 模板

学习本章后，要求了解简单的C++程序的开发过程，理解并掌握类和对象定义及使用的基本方法，理解并掌握三种类型的继承和派生，理解多态的基本概念，了解模板的概念并学会定义简单的模板，为后面章节的学习奠定坚实的基础。

1.1 简单C++程序

和C语言一样，一个C++程序也是由预处理命令、函数及注释等几个基本部分组成。一个C++程序的开发过程如图1-1所示，主要经过编辑、编译、链接和执行4个主要步骤。

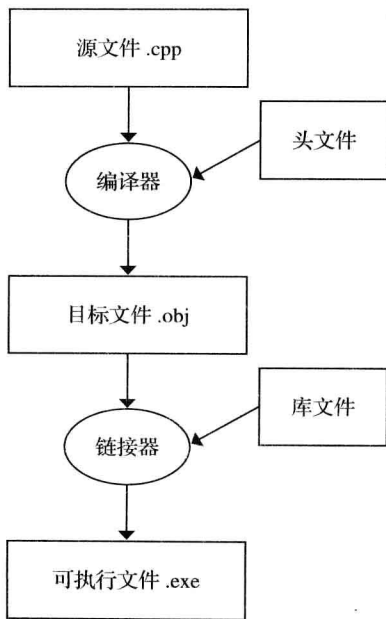


图1-1 C++程序的开发过程

首先来看一个简单例子。

【例1.1】一个简单的C++程序。

```
#include <iostream>           // 载入头文件
using namespace std;         // 使用命名空间std
void main ( )                // 程序入口
{
    cout << "Hello World!"<<endl;
}
```

关于程序的几点说明：

在C++程序中，使用“//”进行单行注释。也可以由“/*”开始，由“*/”结束来实现多行注释。

C++语言的预处理命令都是以“#”号开头。iostream是一个头文件，系统头文件不带.h后缀。

main是主函数名，一个C++语言程序有且仅有一个main函数。C++程序执行时就是从main函数开始。

程序语句间用分号进行间隔即每条语句都以分号结尾。

cout为标准输出流对象，它与显示器相连。<< 是插入操作符，endl为换行符号。整个语句的功能是将“Hello World !”字符串与endl依次插入到cout中。

程序运行结果为：Hello World!

例1.1的主要功能是在屏幕上显示输出“Hello World!”字符串。从例中可以看出C++程序与C语言程序非常相似但也有一定区别。

下面再来看一个例子。

【例1.2】实现两个数互换的C++程序。

```
#include<iostream>
using namespace std;
void swap ( int &a , int& b )      // 函数定义
{
    int t;
    t=a, a=b, b=t;
}
void main ( )
{
    int x=7, y=11;
    cout<<"x="<<x<<" y="<<y<<endl;
    swap ( x, y );                // 函数调用
    cout<<"after swap:";
    cout<<"x="<<x<<" y="<<y<<endl;
}
```

关于程序的几点说明:

本例中包含两个函数: swap和main。swap函数采用引用数据类型做函数的参数, 其主要功能是实现两个数交换。main函数中首先定义并从键盘接受了两个变量x和y的输入, 然后调用swap函数, 实现对两个数的互换, 最后对交换后的结果进行显示输出。

注意:

函数的调用在C语言中我们就学习过, 包括传值调用和传址调用。普通变量做函数参数的时候实现的是传值调用, 而指针和引用做函数参数的时候实现的都是传址调用。

该程序经过编译、连接和执行之后, 在屏幕上显示:

```
x=7 y=11
after swap: x=11 y=7
```

从上面两个例子中我们可以发现, 不涉及“类”的概念的时候, C++程序跟C语言程序的结构是基本一样的。C++与C的区别主要体现在前者具有面向对象性, 而类和对象是面向对象技术的核心。下面我们就来学习一下类和对象。

1.2 类和对象

在面向过程的结构化程序设计中, 程序模块是由函数构成, 函数将对数据处理的语句放在函数体内, 完成特定的功能, 数据则通过函数参数传递进入函数体。

在面向对象的程序设计中, 程序模块是由类构成的。类是对逻辑上相关的函数与数据的封装, 它是对问题的抽象描述。

面向对象的程序设计方法就是运用面向对象的观点来对现实世界中的各种问题进行抽象, 然后用计算机程序来描述并解决该问题, 这种描述和处理是通过类和对象实现的。

类和对象是C++程序设计中最重要概念。

1.2.1 类的定义

类是一种用户自定义的数据类型, 简单来讲, 类是一个包含函数的结构体。因此, 类的定义与结构类型的定义相似, 其格式如下:

```
class 类名
{
    public:
        公有数据成员及公有函数成员的定义;
    protected:
        保护数据成员及保护函数成员的定义;
    private:
        私有数据成员及私有函数成员的定义;
};
```

其中，关键字class表明定义的是一个类；类名是类的名称，应是一个合法的标识符，一般以大写的C开头；public、protected、private为存取控制属性（访问权限），用来控制对类的成员的存取。如果没有标明访问权限，默认访问权限为private。三种存取控制属性的说明详见表1-1。

表1-1 存取控制属性说明

存取属性	意义	可存取对象
public	公开（公有）级	该类成员及所有对象
protected	保护级	该类成员及其子类成员
private	私有级	该类成员

类的成员有数据成员与函数成员两类，类的数据成员和函数成员统称为类的成员，类的数据成员一般用来描述该类对象的属性，称为属性；函数成员是描述类行为，称做方法。函数成员由函数构成，这些作为类成员的函数因此也叫成员函数。

类的成员函数的函数体可以在类内给出，也可以在类外给出。如果函数体在类外给出，需要注意，在函数的返回值类型和函数名之间要增加类名及作用域符号“::”。

来看一个简单类定义的例子。

【例1.3】简单类定义。

```
#include <iostream>
using namespace std;
class CBox{
    int l, w, h;
public:
    void set ( int ll, int ww, int hh ) {l=ll;w=ww;h=hh;}
    int volume ( ) ;
    int area ( ) ;
};
int CBox::volume ( )
{   int v=l*w*h;
    return v;
}
int CBox::area ( )
{
    int s=2* ( w*h+h*l+w*l ) ;
    return s;
}
```

关于程序的几点说明：

本例中定义类是CBox（盒子）类，类的数据成员有三个：l，w和h，分别表示盒子的长、宽和高，它们都是整型的变量。类的成员函数有三个：set，volume和area。其

中set函数的主要功能是对盒子的长、宽和高赋值进行初始化的；volume函数的主要功能是求盒子的体积；area函数的主要功能是求盒子的表面积。在本例中，成员函数set是在类内定义的，而volume和area是在类外定义的，需要注意两者的区别。另外，在定义类的时候还需要注意，关键字public、protected和private可以在类中出现多次，且前后顺序没有关系。

数据成员的类型可以是任意的，包含整型、浮点型、字符型、数组和指针等，也可以是另一个类的对象，但是不允许对所定义的数据成员进行初始化，即在类内不允许对数据成员赋初值。

尽量将类单独存放在一个文件中，或者将类的声明放在头文件（后缀名是.h）中，而将类的成员函数的实现放在与头文件同名的源文件（后缀名是.cpp）中。以后将会看到，Visual C++6.0为用户创建的应用程序框架中都是将各个类以同名的头文件和源文件来组织的。

1.2.2 对象的定义

类可以看做是所有对象的概括和抽象，而对象可以看做是类的实例。类相当于一种包含函数的自定义数据类型，它不占内存，是一个抽象的“虚”体，使用已定义的类来建立对象就像用数据类型定义变量一样。对象建立后，对象占据内存，变成了一个“实”体。类与对象的关系就像数据类型与变量的关系一样。其实，一个变量就是一个简单的不含成员函数的数据对象。对象定义的基本语法为：

类名、对象名；

其中，类名是用户已经定义过的类的名称，对象名可以有一个或多个，多个的时候要用逗号分隔。被定义的对象既可以是一个普通对象，也可以是一个数组对象或指针对象。例如：CBox one, *two, three[10]; one就是类CBox的一个普通对象，two是一个指针对象，three是一个数组对象。在建立对象后，就可以通过对象存取对象中的数据成员，调用成员函数。存取语法如下：

对象名.属性

对象名.成员函数名（实参1，实参2，…）

注意能通过对象名调用的属性或者成员函数，都必须是public类型的。

通过指针对象名来存取对象中的数据成员，调用成员函数的存取语法如下：

对象名—>属性

对象名—>成员函数名（实参1，实参2，…）

注意：

“.”和“—>”都是成员运算符，都用来取得对象的成员，不同的是“.”用来访问普通对象的成员，而“—>”用来访问指针对象的成员。

假定类的定义已经给出，如例1.3所示，而且类的定义存放在CBox.h文件中，我们再看一个对象定义的例子。

【例1.4】定义类对象。

```
#include "CBox.h"
#include <iostream>
```



```
using namespace std;
void main ( )
{
    int s, v;
    CBox mybox;
    mybox.set ( 6, 6, 8 );
    s=mybox.area ( );
    v=mybox.volume ( );
    cout<<"盒子的表面积是: "<<s<<endl;
    cout<<"盒子的体积是: "<<v<<endl;
}
```

关于程序的说明:

本例子中我们定义的对象是mybox, 我们设置mybox的长、宽和高分别为6, 6和8。s和v是定义的两个整型变量, 用来存放盒子的表面积和体积。可以看到通过对象名来调用成员函数的时候使用的是“.”运算符。

在本节中我们主要学习了类和对象的基本概念、定义方法及使用方法。下一节主要学习类中两个重要成员函数、构造函数及析构函数。

1.3 初始化和清除

在定义一个对象的同时, 希望能给它的数据成员赋初值-对象的初始化。在特定对象使用结束时, 还经常需要进行一些清理工作。C++程序中的初始化和清理工作分别由两个特殊的成员函数来完成, 它们就是构造函数和析构函数。

1.3.1 构造函数

构造函数是与类名相同的, 在建立对象时自动调用的函数。如果在定义类时, 没有为类定义构造函数, 编译系统就生成一个默认形式的隐含的构造函数, 这个构造函数的函数体是空的, 因此默认构造函数不具备任何功能。

如果用户至少为类定义了一个构造函数, C++就不会生成任何默认的构造函数, 而是根据对象的参数类型和个数从用户定义的构造函数中选择最合适的构造函数完成对该对象的初始化。

作为类的成员函数, 构造函数可以直接访问类的所有数据成员, 可以是内联函数, 可以不带任何参数, 可以带有参数表以及默认形参值, 还可以重载, 用户可以根据不同问题的具体需要, 有针对性地设计合适的构造函数将对象初始化为特定的状态。

构造函数是类的一个成员函数, 除了具有一般成员函数的特征之外, 还归纳出如下特殊的性质:

- (1) 构造函数的函数名必须与定义它的类同名。
- (2) 构造函数没有返回值。如果在构造函数前加void是错误的。
- (3) 构造函数被声明定义为公有函数。
- (4) 构造函数在建立对象时由系统自动调用, 完成对对象的初始化。

下面来看一个示例。

【例1.5】构造函数定义。

```

#include <iostream>
using namespace std;
class CBox{
    int l, w, h;
public:
    CBox ( ) {l=2;w=3;h=5;}           // 不带参数的构造函数
    CBox ( int ll, int ww, int hh ) {l=ll;w=ww;h=hh;} // 带三个参数的构造函数
    .....
};
void main ( )
{
    CBox mybox;                       // 定义对象，对象名是mybox
    CBox yourbox ( 7, 8, 9 );         // 定义对象，对象名是yourbox
    .....
}

```

关于程序的说明：

在本例中，CBox类包含两个构造函数，CBox ()和CBox (int ll, int ww, int hh)。两个函数的函数名都跟类名相同，两个函数都没有返回值类型，连void也没有，两个函数都定义成事类的公有成员。定义对象mybox的时候，由于没给任何参数，所以系统会自动调用不带参数的构造函数对mybox对象进行初始化，即mybox对象的长、宽和高的值被初始化成2、3和5。定义对象yourbox的时候，系统会自动调用带三个参数的构造函数对yourbox对象进行初始化，即yourbox对象的长、宽和高的值被初始化成7、8和9。

注意：

由于构造函数可以重载，可以定义多个构造函数，在建立对象时根据参数来调用相应的构造函数。如果相应的构造函数没有定义，则出错。

1.3.2 析构函数

与构造函数相对应析构函数在对象生命周期结束时由系统自动调用，以释放分配给对象的内存。析构函数的函数名应为类名前加“~”，析构函数中没有参数，也不能为它指定返回值类型。例如：

```

class CBox{
    int l, w, h;
public:
    .....
}

```