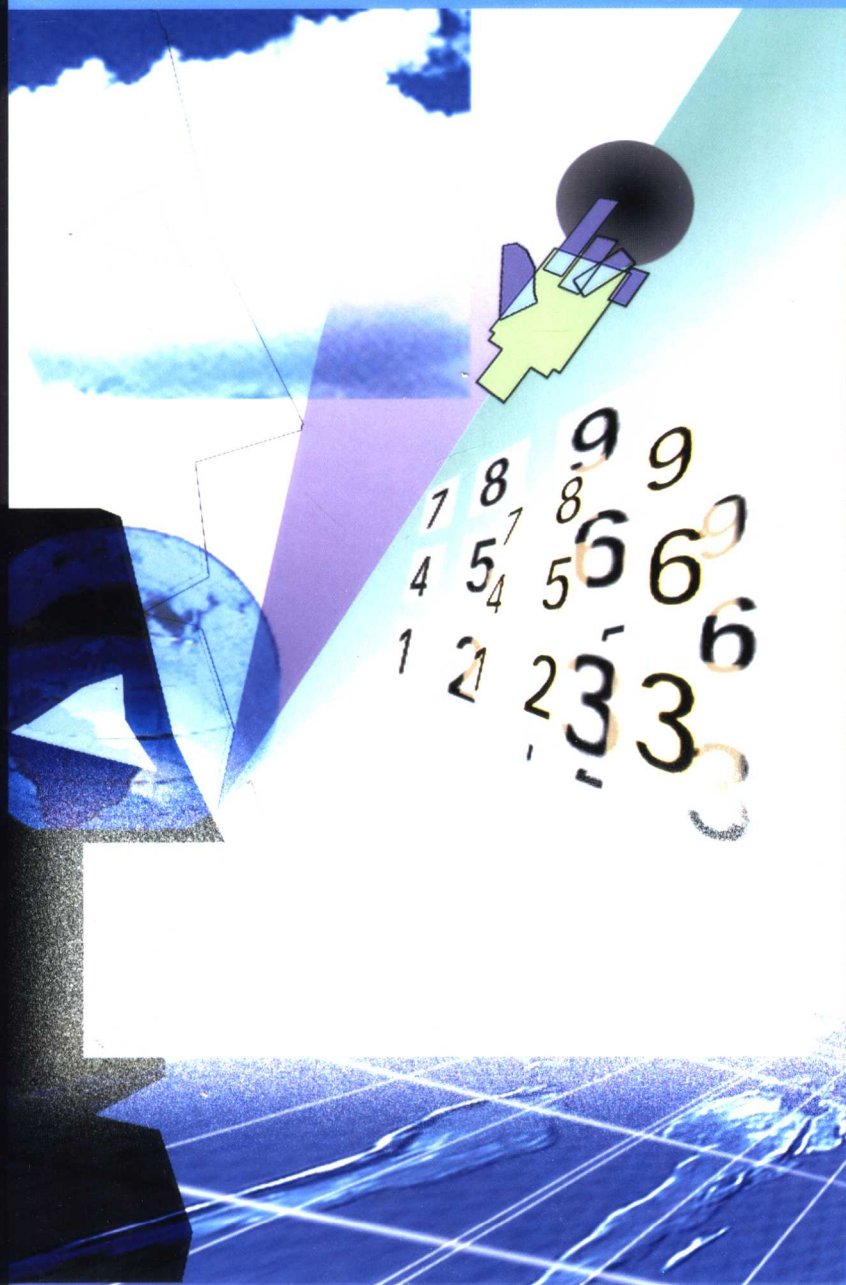


C++ 面向对象程序设计 习题解答与实验指导

陈维兴
陈昕
林小茶
编著



清华大学出版社

C++ 面向对象程序设计

习题解答与实验指导

陈维兴 陈 昕 林小茶 编著

清华大学出版社
北 京

内 容 简 介

本书是《C++ 面向对象程序设计教程(第2版)》(陈维兴、林小茶编著,清华大学出版社2004年出版)的配套用书,也适合单独作为学习C++语言的学习辅导书。书中内容分3部分:第一部分是典型习题与解答,提供了大量作者多年教学中积累、收集与验证的习题,每道题都给出了详细的解答与运行结果;第二部分是上机实验指导,介绍了C++程序设计语言实验环境,并安排了7套精心设计的实验,每个实验都包括详细的实验目的、实验基本要求和实验步骤等;第三部分给出了《C++面向对象程序设计教程(第2版)》各章习题的参考答案。附录A介绍了“上机实验指导”中各个实验的参考源程序,附录B介绍了上机实验时常见的错误信息注释,以供学生上机实验时使用。

本书可作为高校学生学习C++程序设计课程的辅导教材,也可作为C++语言自学者的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C++面向对象程序设计习题解答与实验指导/陈维兴,陈昕,林小茶编著. —北京:清华大学出版社, 2005.11

ISBN 7-302-12049-8

I. C… II. ①陈… ②陈… ③林… 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(2005)第126052号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:柳 萍

文稿编辑:曾 洁

印刷者:北京季蜂印刷有限公司

装订者:三河市新茂装订有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:19.5 字数:458千字

版 次:2005年11月第1版 2005年11月第1次印刷

书 号:ISBN 7-302-12049-8/TP·7798

印 数:1~5000

定 价:26.00元

前 言

C++ 面向对象程序设计习题解答与实验指导

学过程序设计的人,都有一个体会,看别人编写的程序,好像挺明白的,但是一旦要自己编写一个程序,就感觉无从下手。这是因为程序设计是一门对实践环节要求很高的课程,初学者要想真正学会 C++ 语言程序设计,最重要的要抓住两个关键环节:一个是多做习题多编程,另一个就是多上机,写在纸上的程序是否正确,最好的办法就是上机验证一下。为此,我们编写了这本习题解答与实验指导书。本书从比较典型的习题解析着手,对一些基本的程序算法和规则进行了分析,以期帮助学习者尽快掌握 C++ 语言程序设计的基本规则与编程技巧,并能够熟练运用这些规则与技巧,编制出具有良好风格的应用程序,最终能够顺利地通过上机调试。

本书的主要内容分三部分:第一部分是典型习题与解答,按该课程教学的重点内容安排为 7 章,精选了近百道 C++ 的典型习题,并给出了详细解答;第二部分是上机实验指导,介绍了 C++ 程序设计语言实验环境,并安排了 7 套精心设计的实验,每个实验都包括详细的实验目的、实验基本要求和实验步骤,以帮助学习者掌握 C++ 程序设计方法,并进一步加深对课程相关内容的理解与掌握;第三部分是《C++ 面向对象程序设计教程(第 2 版)》(陈维兴、林小茶编著,清华大学出版社 2004 年出版)习题与参考答案,给出了教材各章所有习题的参考答案。附录 A 介绍了“上机实验指导”中各个实验的参考源程序,提供实验参考源程序的主要目的是给学习者一个参考和借鉴,帮助学习者掌握实验内容和理解具体实现步骤,以更好地掌握 C++ 面向对象程序设计的基本方法和概念。另外,考虑到学生在上机操作中,经常遇到一些错误信息,但看不懂,不知如何修改,因此我们把一些常用的错误信息注释放在附录 B 中,以供学生上机实验时使用。

提供习题解答和实验参考源程序的主要目的是给学习者一个参考和借鉴,作者在这里要强调一点,程序设计是一个创作的过程,解决一个实际问题的程序肯定不是惟一的,因此,在阅读本书的参考答案之前,希望读者已经独立思考过教材中的习题及实验题目,这样才有助于程序设计水平的提高,并且不要把本书的参考源程序作为惟一的答案。本书中所有程序都经作者在 Visual C++ 6.0 或 Turbo C++ 3.0 上调试通过。

本书第一部分和附录 B 由陈维兴编写,第二部分 C++ 程序设计语言实验环境介绍部分由林小茶编写,实验部分和附录 A 由陈昕编写,第三部分由陈维兴和林小茶共同编写。全书由陈维兴主编并统稿。

在本书的编写与出版过程中得到了郑玉明教授、陈宝福教授、高明博士以及杨硕和杨光同学的帮助和支持,在此表示最诚挚的感谢。

本书内容是作者多年教学实践的总结,虽然得到了学生的肯定,但由于编者水平有限,错误和不当之处在所难免,在此恳请广大读者批评指正。

作者
2005年9月

目 录

C++ 面向对象程序设计习题解答与实验指导

第一部分 典型习题与解答

第 1 章 面向对象程序设计概述	3
第 2 章 C++ 语言基础	5
第 3 章 类和对象	20
第 4 章 继承与派生	48
第 5 章 多态性与虚函数	71
第 6 章 模板及其使用	99
第 7 章 基于 C++ 流库的输入与输出	117

第二部分 上机实验指导

C++ 程序设计语言实验环境介绍	137
实验 1 C++ 程序设计基础练习	146
实验 2 类与对象	151
实验 3 对象数组与对象指针	154
实验 4 派生类与继承	156
实验 5 虚函数与多态性	158
实验 6 函数模板与类模板	161
实验 7 I/O 流类库	164

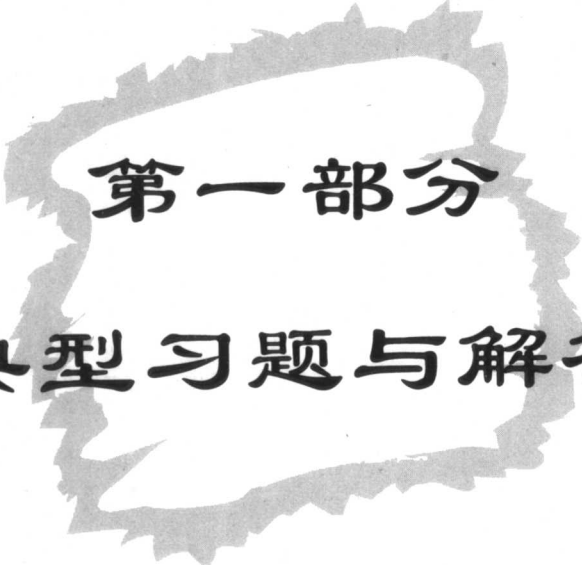
第三部分 《C++ 面向对象程序设计教程(第 2 版)》 习题与参考答案

第 1 章 面向对象程序设计概述	171
第 2 章 C++ 概述	175
第 3 章 类和对象	184

第4章 派生类与继承.....	200
第5章 多态性.....	214
第6章 模板.....	225
第7章 C++的I/O流类库.....	232

附 录

附录A 上机实验题参考源程序	243
附录B C++上机操作常见错误信息	272
参考文献.....	303



第一部分
典型习题与解答



第 1 章 面向对象程序设计概述

【1.1】 在面向对象程序设计中,什么是方法?

【解】 方法是面向对象程序设计中的一个术语。属于一个类的对象具有相同的行为,当某个行为作用在对象时,就称对象执行了一个方法。方法定义了一系列的计算步骤。所以,可以说一个对象请求另一个对象执行一个特定的方法,或者说一个对象发送一个消息给另一个对象引起那个对象方法的执行。从这个意义上看,对象的行为是负责响应消息并进行操作的。

【1.2】 在面向对象程序设计中,对象具有什么特性?

【解】 对象是构成和支撑整个面向对象程序设计(OOP)最重要的细胞与基石,它具有下述特性。

(1) 封装性。对象将私有元素和实现操作的内部细节隐藏起来,并能管理自己的内部状态。外部只能从对象所表示的具体概念、对象提供的服务和对象提供的外部接口来认识对象。通过向对象发送消息来激活对象的自身动作,达到施加特定操作的目的。

(2) 模块独立性。一个对象是一个独立存在的模块,具有独立的计算能力;并且其自身的变化不受外界的干扰,也不会涉及到其他的模块;模块间的依赖性极小或几乎没有;各模块可以独立地成为软件系统的组件,也可以被程序员重用。

(3) 动态连接性。对象的动态连接性是指对象通过消息传递机制将对象动态地联系在一起的特性。对象能接收其他对象发送的消息,也能向其他对象发送消息。通过这种联系,若干对象可协同完成某项任务。

(4) 易维护性。由于对象的功能被“隐蔽”,所以修改和完善的工作都被局限于对象的内部,不会涉及到外部,因此使得对象和整个系统变得非常容易维护。

【1.3】 继承的作用是什么?

【解】 继承的作用有两个:其一是避免公用代码的重复开发,减少代码和数据冗余;其二是通过增强一致性来减少模块间的接口和界面。

如果没有继承机制,每次的软件开发都要从“一无所有”开始,因为类的开发者们在构造类时,各自为政,使类与类之间没有什么联系,分别是一个个独立的实体。继承使程序不再是毫无关系的类的堆砌,而是具有良好的结构。

继承机制为程序员们提供了一种组织、构造和重用类的手段。继承使一个类(基类)的数据结构和操作被另一个类(派生类)重用,在派生类中只需描述其基类中没有的数据

和操作。这样,就避免了公用代码的重复开发,增加了程序的可重用性,减少了代码和数据冗余。同时,在描述派生类时,程序员还可以覆盖基类的一些操作,或修改和重定义基类中的操作,例如子女对所继承的房子进行装修。

继承机制以相关的关系来组织事物,可以减少对相似的事物进行说明和记忆的规模,为程序员提供了一种简化的手段。程序员可以将相关的类收集在一起,生成高一级的、概括了这些类的共性的类。具有适应关系的类处于一个继承层次结构中,高层的类作为低层的类的抽象,使程序员能够忽略那些低层类的不同实现细节,而按照高层类编写通用程序,并且在掌握了高层类的特征以后,很快地掌握低层类的特征,给编程工作带来方便。

【1.4】 从继承内容上划分,继承可分为哪几种?分别予以说明。

【解】 从继承内容上划分,继承可分为取代继承、包含继承、受限继承、特化继承。

取代继承:例如徒弟从师傅那里学到的所有技术,在任何需要师傅的地方都可以由徒弟来替代,这就属于取代继承。

包含继承:例如“柯利狗”继承了“狗”的所有特征,任何一条“柯利狗”都是一条“狗”,这就属于包含继承。

受限继承:例如“鸵鸟”尽管继承了“鸟”的一些特征,但不能继承鸟会飞的特征,这就属于受限继承。

特化继承:例如“运动员”是一类特殊的人,比一般人具有更多体育特长,这就属于特化继承。

【1.5】 面向对象程序设计语言应该具备哪些特征?

【解】 面向对象程序设计语言应该具备下述特征:

- (1) 它支持对象的概念(包括对象所有的特性,如封装等);
- (2) 它要求对象属于类;
- (3) 它提供继承机制。

第 2 章 C++ 语言基础

【2.1】 由用户输入一个整数,要求在屏幕上依次列出 10 个数及其二次方、三次方。

【解】 实现本题功能的程序如下:

```
#include<iostream. h>
main()
{
    int n;
    cout<<"Enter a number: ";
    cin>>n;
    for (int i=n; i<=n+9; i++)
    {
        cout<<i;
        cout<<" "<<long(i) * i<<" "<<long(i) * i * i;
        cout<<endl;
    }
    return 0;
}
```

本程序的运行结果如下:

```
Enter a number: 11
11    121    1331
12    144    1728
13    169    2197
14    196    1744
15    225    3375
16    156    4096
17    289    4913
18    324    5832
19    361    6859
20    400    8000
```

说明:

(1) 为尽可能容纳数值的三次方,程序中使用了 long 型整数,例如:

```
cout<<long(i) * i * i;
```

或

```
cout<<long(i * i * i);
```

(2) long 长整型数容纳的数值范围为-2147483648 到 2147483647

【2.2】 由用户从键盘输入一个数,计算该数的阶乘。

【解】 实现本题功能的程序如下:

```
#include <iostream. h>
main()
{
    int n;
    unsigned long factor=1;
    cout<<"Enter a number: ";
    cin>> n;
    for (int i=n;i>1;i--)
        factor=factor * i;
    cout<<n<<"!="<<factor<<endl;
    return 0;
}
```

本程序的运行结果如下:

```
Enter a number: 6
```

```
6!=720
```

```
Enter a number: 12
```

```
12!=479001600
```

```
Enter a number: 8
```

```
8!=40320
```

注意:

unsigned long 型整数可容纳的数值范围为 0 到 4294967295,故用户最多只能计算到 12!。

【2.3】 由用户从键盘输入一个数 n,计算第 n 个 fibonacci 数。

【解】 实现本题功能的程序如下:

```
#include <iostream. h>
main()
{
    int n;
    unsigned long fib1=0, fib2=1;
    cout<<"Enter a number: ";
    cin>>n;
    // 计算第 n 个 fibonacci 数
```

```

for (int i=2;i<=n;i++)
{
    long x = fib1;
    fib1 = fib2;
    fib2 = x+fib1;
}
cout<<"fibonacci: " <<fib2<<endl;
return 0;
}

```

本程序的运行结果如下：

Enter a number: 8

fibonacci: 21

Enter a number: 14

fibonacci: 377

Enter a number: 16

fibonacci: 987

说明：

fibonacci 数列为：1,1,2,3,5,8,13,32,34,55,...

定义：

$\text{fib}(n) = 1$ (当 $n=1$ 或 $n=2$ 时)

$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$ (当 $n > 2$ 时)

【2.4】 要求用户从键盘输入一段文字，由程序返回该段文字所包含的字符和单词个数。单词定义为用空白符隔开的文本字符串。

【解】 实现本题功能的程序如下：

```

#include <iostream, h>
#include <conio, h>
main()
{
    int char_count=0;
    int word_count=0;
    char ch;
    int num=0;
    cout<<"Please enter a string: \n";
    while ((ch=getche())!='\r')
    {
        char_count++;
        if (ch==' '||ch=='\t')
        {num++; }
        else if (num>0)
        {

```

```
        word_count++;
        num==0;
    }
}
cout<<" \nThis string contains "<<char_count
    <<" characters "<<endl;
cout<<" This string consists of "<<word_count+1
    <<" words "<< endl;
return 0;
}
```

本程序的运行结果如下:

```
Please enter a string:
This is a test.
This string contains 15 characters
This string consists of 4 words
```

说明:

- (1) 键盘上输入的任何字符(包括退格键)均列入字符总数内。
- (2) 按下 Enter 键时 getche()函数返回值为字符'\r',退出循环。

【2.5】 从键盘上输入一个长整型数,判别该数是否为素数。

【解】 实现本题功能的程序如下:

```
#include <iostream. h>
#include <process. h>
void main()
{
    long x;
    cout<<"Please enter a number: \n";
    cin>>x;
    for (long n=2;n<x/2;n++)
        if (x % n==0)
        {
            cout<<"It's not a prime, can be divided by "
                <<n<<endl;
            exit(0);
        }
    cout<<x<<" is a prime !" <<endl;
}
```

本程序的运行结果如下:

```
Please enter a number:
115
It's not a prime,can be divided by 5
```

Please enter a number;

11113

11113 is a prime !

说明:

(1) 素数的定义是:除了1及本身外,不能被其他数整除的数。

(2) 判断某个整数是否为素数,可将该整数自2除起,如被整除表示该数不是素数。

为提高程序的运行效率,除数的最大值应小于该数的1/2。

【2.6】 利用交换式气泡排序方法,由小到大显示出数组元素的值。

【解】 实现本题功能的程序如下:

```
#include <iostream. h>
#include <iomanip. h>
const MAXSIZE=20;
void sort(int list[], int size);
void main()
{
    static int list[MAXSIZE];
    int size=0;
    do
    {
        cout<<"Type number <-999 to exit>: ";
        cin>>list[size];
    }
    while (list[size++]!= -999);
    sort(list, size-1);
    cout<<"After sorting ... \n";
    for (int dex=0; dex<size-1; dex++)
        cout<<setw(5)<< list[dex];
}
void sort(int list[], int size)
{
    int x;
    for (int y=0; y<size-1; y++)
        for (x=y+1; x<size; x++)
            if (list[y]>list[x])
            {
                int temp =list[x];
                list[x]= list[y];
                list[y]=temp;
            }
}
```

本程序的运行结果如下:


```
Type number <-999 to exit>: 134
Type number <-999 to exit>: 120
Type number <-999 to exit>: -334
Type number <-999 to exit>: 55
Type number <-999 to exit>: 77
Type number <-999 to exit>: 567
Type number <-999 to exit>: -999
After sorting ...
-334 55 77 120 134 567
```

说明:

气泡排序(bubble sorting)是一种用来将数据重新排列的最简单、但效率较低的方法。交换式气泡排序使用双层循环,外层循环由 list[0]开始依次和内层循环 list[1]...list[size-1]比较,较小的数存放在 list[0]处。接着外层循环的 list[1]和内层循环 list[2]...list[size-1]比较,较小的数存放在 list[1]中。就这样依次比较,最后,外层循环的 list[size-2]和内层循环的 list[size-1]比较,较小的数存放在 list[size-2]中。

本例设定最多可以容纳 20 个元素的整型数组。由用户自键盘上输入数值,输入 -999 后结束。

【2.7】 试编写一个求两个双精度浮点数中最大值的函数,要求将该函数定义为内置函数。

【解】 实现本题功能的程序如下:

```
#include <iostream. h>
inline double max(double v1, double v2)
{
    double ret;
    ret = (v1 > v2)? v1: v2;
    return(ret);
}
void main()
{
    double v1=1. 23456, v2=2. 34567, v;
    v=max(v1, v2);
    cout<<"max("<<v1<<","<<v2<<)"="<<v<<endl;
}
```

程序的执行结果如下:

```
max(1. 23456, 2. 34567)=2. 34567
```

说明:

- (1) max()被定义为内置函数,其调用语句同普通函数一样。
- (2) 通常内置函数可以提高程序的执行效率,但是当内置函数为较长的复杂函数,且调用频繁时,程序将加长很多,反而会影响执行效率。因此,一般只有较短且简单的函数