

全国计算机自学考试全程过关必备丛书
◆计算机及其应用专业◆

面向对象程序设计

习题与真题解析

(本科)

徐 春 编 著

deep



中国水利水电出版社
www.waterpub.com.cn

全国计算机自学考试全程过关必备丛书

面向对象程序设计习题与真题解析

徐春 编著

中国水利水电出版社

内 容 提 要

本书是配合计算机及应用专业(本科)全国高等教育自学考试指定教材《面向对象程序设计》一书及大纲而编写的参考书。内容包括三部分:第一篇为配套教材习题解答,包括了经济科学出版社出版的《面向对象程序设计》(刘振安主编)一书各章节全部习题的详细解答;第二篇为往年试题及经典题目解析,并附详细解答;第三篇为最新自考试卷解析,对最新自考试题作了详细分析和解答。

本书内容丰富、概念清晰、实用性强,是学习 C++ 语言的一本实用参考书,是自学考试学生不可缺少的参考资料。同时亦可作为其他 C++ 语言教材的参考书。本书也适合高等学校或计算机培训班使用。

图书在版编目(CIP)数据

面向对象程序设计习题与真题解析/徐春编著. —北京:中国水利水电出版社, 2003

(全国计算机自学考试全程过关必备丛书)

ISBN 7-5084-1809-3

I. 面… II. 徐… III. 面向对象语言—程序设计—高等教育—自学考试—解题 IV. TP312-44

中国版本图书馆 CIP 数据核字(2003)第 108743 号

书 名	面向对象程序设计习题与真题解析
作 者	徐春 编著
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 11 印张 245 千字
版 次	2004 年 1 月第一版 2004 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	15.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

面向对象的程序设计与传统的面向过程的程序设计技术完全不同，面向对象的程序设计技术，把现实世界的问题抽象为“类”，而要解决的问题是对类所生成的对象的一系列操作。

C++是一种常见的面向对象的程序设计语言，近几年被国家自学考试委员会列为计算机应用技术专业考试课程之一，对广大自考生来说是一门很关键的考试课程，为此，作者根据刘振安教授的《面向对象程序设计》（经济科学出版社2000版）编写了这本辅导教材，希望各位考生能够通过对这本书的使用，顺利通过考试。

本书中的所有例题和习题中的程序均在 Visual C++6.0 系统下运行成功。

本书共分三部分：第一部分为九章，内容紧扣教学大纲，在章节安排上与《面向对象程序设计》（刘振安主编）同步。每一章都先介绍基础知识和重点、难点，接着给出相关习题及其答案；第二部分为九章，本部分主要以往年真题为例，包括全国自学考试和各地区自学考试的真题，进行详细的分析，使广大考生能够有的放矢的去复习；第三部分包含两份全国自学考试近两年的试题，并附相关的解析，能够帮助考生对全国自学考试的形式有全面的了解。

由于时间仓促、水平有限，书中难免有不足之处，希望获得广大考生和老师的指导，以趋完善。

编者

2003 年 10 月于南京农业大学

目 录

第一部分 配套习题分析与解答	1
第一章 面向对象及 C++ 基础知识	2
第二章 类和对象	5
第三章 构造函数与析构函数	18
第四章 继承和派生类	26
第五章 多态性与虚函数	38
第六章 进一步使用成员函数	46
第七章 运算符重载及流类库	54
第八章 模板	66
第九章 进一步掌握面向对象程序设计	72
第二部分 历年试卷真题及经典题目解析	77
第十章 面向对象及 C++ 基础知识	78
第十一章 类和对象	81
第十二章 构造函数与析构函数	90
第十三章 继承和派生类	97
第十四章 多态性与虚函数	106
第十五章 进一步使用成员函数	115
第十六章 运算符重载及流类库	125
第十七章 模板	130
第十八章 进一步掌握面向对象程序设计	135
第三部分 最新自考试卷与解析	139
全国 2000 年 10 月高等教育自学考试 面向对象程序设计试题	140
参考答案	148
全国 2001 年 10 月高等教育自学考试 面向对象程序设计试题	150
参考答案	158
全国 2002 年 10 月高等教育自学考试 面向对象程序设计试题	160
参考答案	169

第一部分 配套习题分析与解答

第一章 面向对象及 C++ 基础知识

复习要点:

面向对象程序设计方法要求语言必须具备抽象、封装、继承和多态性这几个关键要素。

所谓封装,就是将一组数据和与这组数据有关的操作集合组装在一起,形成一个能动的实体,即对象。这种情况下,用户不能直接操作数据,而必须通过数据操作访问数据。数据封装就是给数据提供了与外界联系的标准接口,只有通过这些接口,使用规范的方式,才能访问这些数据。

C++语言的输入、输出,是通过流来实现的;

C++的注释方式有两种,用“//”和“/*.....*/”都可以;

常量可用 const 来修饰。经 const 修饰的变量在程序中的值不可改变。

C++中用 new 和 delete 两个运算符。处理指针的动态内存分配。

引用是 C++对 C 的一项重大改进。引用运算符是“&”。引用不是变量,所以不能说明引用的引用,也不能说明数据类型为引用数组或是指向引用的指针,但是可以说明对指针的引用。引用的主要用途之一是建立函数参数的引用传递方式。有了引用,不用指针也能改变实参的值。

出题形式一般为填空或选择。

习题解答:

一、编程题

1. 写出下列计算公式的 C++赋值语句

(1) 球的体积: $V=4/3 \pi r^3$

(2) 圆面积: $S=\pi r^2$

(3) 并联电阻的阻抗: $r=1/(\frac{1}{r_1}+\frac{1}{r_2})$

【解答】

(1) $V=4/3*3.1415926*r*r*r;$

(2) $S=3.1415926*r*r;$

(3) $r=1/(1/r_1+1/r_2);$

2. 定义数组输入一个班 50 个学生的一门课程的成绩,然后再按成绩从高分到低分排序输出学生的成绩。

【解答】

```
#include "iostream.h"
#include "stdio.h"
void main()
{
```

```
const int M=50;
int i,j,k=0,t=0;
int stu[M][2]; //二维数组, 一列放学号, 一列放成绩
for (i=0; i< M; i++)
{
    stu[i][0]=i;
    cout << "请输入学号为"<< i+1<<"的学生的成绩: " ;
    cin >> stu[i][1];
}
cout << endl;
for (i=0; i< M-1; i++)
for (j=i+1; j<M; j++)
{ //排序
    if(stu[i][1]< stu[j][1])
    { //排成绩
        k=stu[i][1]; stu[i][1]=stu[j][1]; stu[j][1]=k;
        //排学号
        t=stu[i][0]; stu[i][0]=stu[j][0]; stu[j][0]=t;
    }
}
//输出
cout << "排序后的成绩表如下: " << endl;
cout << "学号" << " " << "成绩" << endl;
for (i=0; i<M; i++)
    cout << stu[i][0]+1<< " " << stu[i][1] << endl;
}
```

3. 用 new 操作作为一个包含 20 个整数的数组分配内存, 输入若干个值到数组中, 分别统计其中正数和负数的个数后再用 delete 操作释放内存。

【解答】

```
#include <iostream.h>
void main()
{
    int *p;
    p=new int[20];
    int i,k=0,q=0;
    cout << "请输入整数,输入 0 结束" << endl;
    for (i=0; i< 20; i++)
    {
        cin >> p[i] ;
        if(p[i]==0) break;
        if(p[i]> 0) k+=1; if(p[i]< 0) q+=1;
    }
    cout <<"正数的个数为: " << k << endl;
    cout <<"负数的个数为: " << q << endl;
    delete p;
}
```

4. 编写一个函数, 计算 $n!$ 。**【解答】**

```
#include <iostream.h>
long fun(int n);
void main()
{   int n;
    cout << "请输入一个整数";
    cin >> n;
    cout << n << "!=" << fun(n);
}
long fun(int n)
{   int i;
    long nn=1;
    for (i=1; i<= n; i++)
        nn*=i;
    return nn;
}
```

二、分析程序题

1. 程序如下

```
#include <iostream.h>
void main()
{
    int num =500;
    int &ref=num;
    cout << ref;
    ref=ref+100;
    cout << " " << num;
    num=num+50;
    cout << " " << ref;
}
```

【解答】 500 600 650

2. 程序如下

```
#include <iostream.h>
void main()
{
    const int num=500;
    int &ref=num;
    cout<< ref<<endl;
    cout <<num<< endl;
}
```

【解答】 500

600

500

第二章 类和对象

复习要点:

C++规定任何数据类型都必须先定义后使用,类也不例外。

类就是一种用户自己定义的数据类型。与其他数据类型不同的是,组成这种类型的不仅可以有数据,而且可以有对数据进行操作的函数,它们分别叫做类的数据成员和类的函数成员。

定义类的一般形式:三个关键字 `private` (私有)、`public` (公有)、`protected` (保护),要记清其所定义类成员的访问权限分别是什么。如定义类时没使用这几个关键字,则所有成员被缺省定义为什么权限。

不能在类说明中对数据成员使用表达式进行初始化。并且,在类中说明的任何成员不能使用 `extern`、`auto` 和 `register` 关键字进行修饰。注意:是在类中不能用。

以类这种类型说明的变量就是对象,对象成员选择运算符“.”和成员访问运算符“->”的使用。

C++中,数据封装是通过类实现的。封装可由 `struct`、`union` 和 `class` 等关键字提供。也就是说,用 `struct` 把数据和函数组合后也是一个类,不一定非得用 `class`。

内联成员函数:在类定义时给出成员函数的实现或者在定义成员函数时在前面使用 `inline` 关键字,都能使成员函数成为内联成员函数。使用内联函数可提高程序运行效率,简单的成员函数一般采用内联函数实现。

重载函数就是名字相同但其参数和(返回)类型不同的函数,也可以带有缺省参数。

在C++中,当一个成员函数被调用时,系统自动向它传递一个隐含的参数是 `this` 指针。通过它,成员函数知道该对哪个对象进行操作。

`Struct` (结构)也可以定义类,像使用 `class` 一样。但更适合建立无相关函数的数据结构。

`Union` (联合)既不能作任何类的基类,也不能从任何类中派生出联合,也不能说明虚函数。

类作用域:就是说明类时使用的一对花括号中间的域。类中的成员名可使用类名和作用域运算符(也叫作用域限定符::)来显示限定,称为成员名限定。

类中各数据成员的生存期由对象的生存期决定。成员函数具有外部连接属性,不随对象的消失而消失。

类对象的性质:① 对象之间可赋值;② 对象可作数组元(素);③ 指向对象可有指针(地址);④ 对象可以作参数(值传递);⑤ 对象可作对象的成员。

本章的出题形式主要是选择、填空及简答。

习题解答:

一、单选题

1. 在 C++ 中, 封装是借助于 () 达到的。

- A. 结构 B. 类 C. 数组 D. 函数

【解答】B。

2. 所有在函数中定义的变量, 连同形式参数, 都属于 ()。

- A. 全局变量 B. 局部变量 C. 静态变量 D. 函数

【解答】B。

3. 以下不属于类的存取权限的是 ()。

- A. public B. static C. protected D. private

【解答】B。

二、多选题

1. 以下 C++ 中结构的特性对类也同样适用的是 ()。

- A. 对象之间可以相互赋值
B. 对象可以用作函数参数
C. 对象可以用作数组的元素
D. 对象可以用作另一对象的成员

【解答】A、B、C、D。

2. C++ 中, 封装可以由下列哪些关键字提供 ()。

- A. struct B. union C. class D. type

【解答】A、B、C。

3. 面向对象的程序中, 对象的特点有 ()。

- A. 多态性 B. 抽象性 C. 封装性

【解答】A、B、C。

三、问答题

1. 类和结构有何区别?

【解答】C++ 语言的类是一种用户自定义的数据类型, 组成这种类型的不仅可以有数据, 而且可以有对数据进行操作的函数。

类中定义的数据和函数称为这个类的成员, 类成员均具有一个属性, 叫做访问权限 (private、public、protected), 如果没有使用关键字, 则以 class 说明的类所有的成员缺省定义为 private (私有成员)。

结构也是类, 是类的一种特例, 其中的成员在缺省的情况下是公有的。其余特征相同但结构本身更适合建立无相关函数的数据结构。

2. 公有函数如何保护私有数据?

【解答】在类中, 一般我们将数据成员说明为私有的, 以便隐藏数据; 而将部分成员函数说明为公有的, 用于提供外界和这个类的对象相互作用的接口。

通过定义一个类, 就不能再从对象外部直接存取对象属性, 而作用对象行为才可以操作数据, 外界要使用对象属性, 只有通过公用成员函数来支持。对象内部实现的细节对外

部来说是不可见的,这就是数据封装。

公有成员函数通过数据封装来保护私有数据成员的。

四、编程题

1. 定义一个类表示公路上的车辆,它存放一辆车有几个轮子和一辆车能载几名乘客的信息。

【解答】

```
class bus{
private:
    int wheels,passengers;
public:
    void init(int w,int p);
    int getw();
    int getp();
};
```

2. 设计一个类,体现其封装性。

【解答】

```
class mankind{
private:
    //定义私有数据成员性别、肤色、年龄
    char sex,color; int age;
public:
    //定义公有成员函数
    void init(char s,char c,int a)
    char getsex();
    char getcolor();
    int getage();
};
```

3. 使用内联函数设计一个类,用来表示直角坐标系中的坐标。

【解答】

```
class coordinate{
private:
    //定义私有成员 X,Y
    int X,Y;
public:
    void init(int initx,int inity)
    //在定义类时给出成员函数的实现,这时成员函数就是内联函数
    {
        X=initx;
        Y=inity;
    }
    int getX(){return X;}
    int getY(){return Y;}
};
```

4. 设计一个类,使它具有一个计算两个数之和的成员函数,并使用一个测试程序验

证程序。

【解答】

```
//sum.h
class sum{
    private:
        int a,b;
    public:
        void init(int initA,int initB);
        int getA();
        int getB();
        int sumAB();
};

//sum.cpp    类 sum 的实现文件
#include "sum.h"
void sum::init(int init A,int initB)
{
    a=initA;
    b=initB;
}
int sum::getA(){return a;}
int sum::getB(){return b;}
int sum::sumAB(){return a+b;}
//使用类 sum 的测试程序
#include <iostream.h>
void main()
{
    sum sum1,sum2;    //定义两个对象
    sum1.init(3,5);    //初始化
    cout<<sum1.getA()<<"+"<<sum1.getB()<<"="<<sum1.sumAB()<<endl;
    sum2.init(12,33);
    cout<<sum2.getA()<<"+"<<sum2.getB()<<"="<<sum2.sumAB()<<endl;
}
```

5. 下面是以传统形式的 struct 来定义一存放字符的栈类型,并定义了有关栈操作的各种函数。编写一主函数,定义并输入一个字符串,用此串来验证栈的各种操作,包括把串压入栈后再弹出,以相反的次序将它们显示出来。

```
const max_len=1000;
enum boolean{false,true};
enum{EMPTY=-1,FULL=max_len-1};
struct stack{
    char s[max_len];
    int top;
}

void reset(stack *st)
{
    st->top=EMPTY;}
```

```
void push(char c, stack *st)
{
    st->top++;
    st->s[st->top]=c;
}

char pop(stack *st)
{ return(st->s[st->top--]);}

char top(stack *st)
{ return(st->s[st->top]);}

boolean empty(const stack *st)
{ return (boolean)(st->top ==EMPTY);}
```

```
boolean full(const stack *st)
{ return(boolean)(st->top==FULL);}
```

【解答】主程序如下:

```
#include <iostream.h>
const max_len=1000;
enum boolean{ffalse, ttrue};
enum {EMPTY=-1, FULL=max_len-1};
//////////定义类//////////
struct stack{
    char s[max_len];
    int top;
};
//////////定义类的实现//////////
void reset(stack *st)
{ st->top=EMPTY;}

void push(char c, stack *st)
{
    st->top++;
    st->s[st->top]=c;
}

char pop(stack *st)
{ return(st->s[st->top--]);}

char top(stack *st)
{ return (st->s[st->top]);}

boolean empty(const stack *st)
{ return (boolean)(st->top == EMPTY);}
```

```

boolean full(const stack *st)
{ return (boolean) (st->top==FULL); }

//////////主函数//////////
void main( )
{
    stack test,*pt; char c[max_len];
    pt=&test;
    cout << "请输入一个字符串: ";
    cin >> c ;
    for(int i=0;c[i]!='\0';i++)
    {
        cout << c[i];
    }
    reset(pt);          //////////置空栈
    for ( i=0;c[i]!='\0'&&(full(pt)==ffalse); i++)
        push(c[i],pt);  //////////进栈
    if (empty(pt)==ffalse)
        cout << endl << "现在的栈顶元素是: " << top(pt) << endl;
    cout << endl;
    cout << "弹出后的字符串序列: " << endl;
    for (i=0; (empty(pt)==ffalse);i++)
        cout << pop(pt) ;  //////////退栈并显示
    cout << endl;
}

```

6. 用类来定义第 5 题的字符栈, 并把所有的栈操作函数定义成类成员函数, 其中入栈操作 push 函数和出栈操作 pop 函数用内联关键字 inline 在栈类定义之外来定义。

【解答】

```

const max_len=1000;
enum boolean{ffalse,ttrue};
enum {EMPTY=-1, FULL=max_len-1};
//////////定义类//////////
class stack{
    char s[max_len];
    int top;
    //////////成员函数//////////
    void reset(stack *st)
    {
        st->top=EMPTY;
    }
    void push(char ,stack *);
    char pop(stack*);
    char gettop(stack *st)
    { return (st->s[st->top]); }
}

```

```

boolean empty(const stack *st)
{
    return (boolean) (st->top == EMPTY);
}
boolean full(const stack *st)
{
    return (boolean) (st->top==FULL);
};
};
//////////外部定义的内联成员函数//////////
inline void stack :: push(char c,stack *st)
{
    st->top++;
    st->s[st->top]=c;
}
inline char stack :: pop(stack *st)
{
    return(st->s[st->top--]);
}

```

7. 在第6题程序的基础上, 定义两个重载函数 `push(char c,int n)` 和 `pop(int n)`, 表示连续压入或弹出 `n` 个字符。

【解答】两个重载函数如下:

```

const max_len=1000;
enum boolean{ffalse, ttrue};
enum {EMPTY=-1, FULL=max_len-1};
//////////定义类//////////
class stack{
    char s[max_len];
    int top;
    void reset(stack *st)
    {
        st->top=EMPTY;
    }
    void push(char ,stack *);
    ///////////新增定义的两个重载函数//////////
    void push(char , int , stack *);
    char pop(int , stack*);
    ///////////
    char pop(stack*);
    char gettop(stack *st)
    { return (st->s[st->top]);}

    boolean empty(const stack *st)
    {
        return (boolean) (st->top == EMPTY);
    }
}

```

```

    boolean full(const stack *st)
    {
        return(boolean)(st->top==FULL);
    };
    void push(char c ,int n);
    char pop( int n);
};
//////////外部定义的内联成员函数//////////
inline void stack :: push(char c,stack *st)
{
    st->top++;
    st->s[st->top]=c;
}
inline char stack :: pop(stack *st)
{
    return(st->s[st->top--]);
}
//////////新增的两个重载函数//////////

inline void stack :: push(char c, int n, stack *st)
{
    for (int i=0; i<n&&full(st)==ffalse; i++)
        push(c,st);
}
inline char stack :: pop(int n, stack *st)
{
    for (int i= 0; i<n&&empty(st)==ffalse; i++)
        pop(st);
}
//////////

```

8. 在第 7 题的基础上,把所有的成员函数定义为公有的成员,而其他的成员都定义为私有成员,并编写主函数调用测试各种栈操作函数的程序。

【解答】

```

#include <iostream.h>
const max_len=1000;
enum boolean{ffalse,ttrue};
enum {EMPTY=-1, FULL=max_len-1};
//////////定义类//////////
class stack{
private:
    char s[max_len];
    int top;
public:
    void reset(stack *st)
    {
        st->top=EMPTY;
    }
}

```