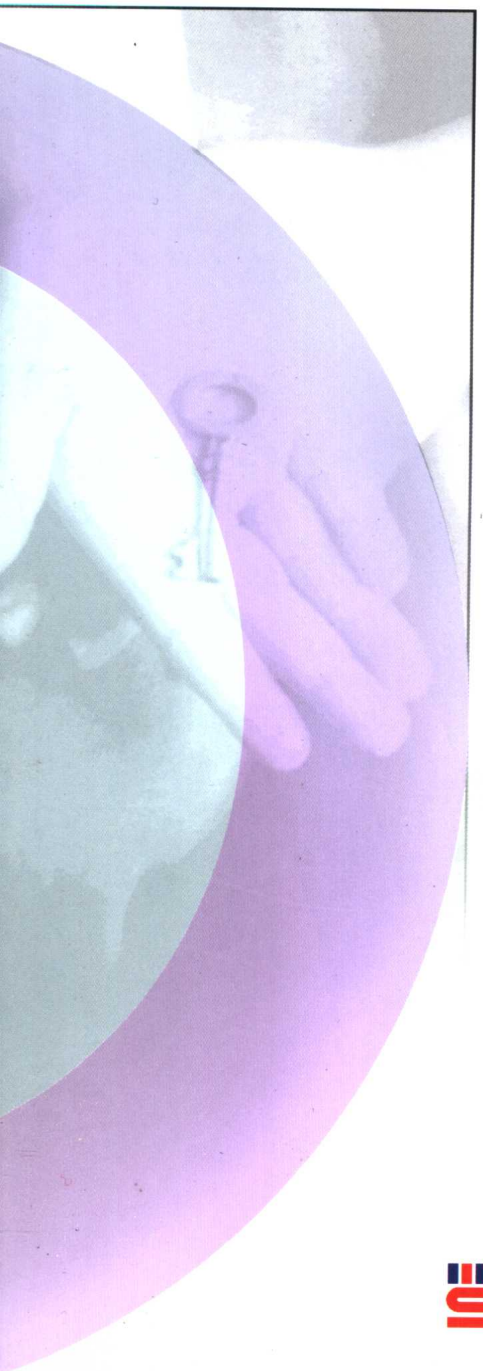
高等教育自学考试　计算机类

学习指导与题典



面向对象程序设计

王 兵　王睿伯　编著

科学出版社
www.sciencep.com

高等教育自学考试 计算机类

学习指导与题典

面向对象程序设计

王兵 王睿伯 编著

科学出版社

北京

内 容 简 介

本书是根据全国高等教育自学考试委员会指定教材《面向对象程序设计》(独立本科段)编写的同步辅导教材。本书围绕教材内容，紧扣自考大纲而编写。每一章分为大纲要求、内容提要、经典例题解析、教材习题答案和自测题5个部分，最后在附录中提供了若干模拟题和近年全国自考试题，供学生进行最后的自我测试用。

本书是参加自学考试学生的辅导教材，也可作为高等学校本科及专科学生的学习参考书。

图书在版编目（CIP）数据

学习指导与题典：面向对象程序设计/王兵，王睿伯编著. —北京：科学出版社，2003

（高等教育自学考试　计算机类）

ISBN　7-03-012374-3

Ⅰ.学…　Ⅱ.①王…　②王…　Ⅲ.高级语言—程序设计—高等教育—自学考试—自学参考资料　Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第094164号

策划编辑：李　娜 / 责任编辑：陈砺川

责任印制：吕春珉 / 封面设计：东方人华平面设计部

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行　各地新华书店经销

*

2003年11月第 一 版　开本：787×1092　1/16

2003年11月第一次印刷　印张：17

印数：1—4 000　字数：387 000

定价：24.00元

（如有印装质量问题，我社负责调换〈路通〉）

前　言

在我国，高等教育自学考试正方兴未艾，据不完全统计，全国每年参加自学考试的考生以百万计。尤其是计算机专业的考生更是占相当部分。相对于全日制高等学校的学生来说，自考学生的学习受到多方面因素的制约：第一，他们一般不会像全日制学生那样系统地参加学习，大多是通过自学的形式完成学业；第二，在参加自考的学生中有相当一部分是已经参加工作的在职人员，因此，他们又不可能像全日制学生那样在学习时间上有充分的保证；第三，自学考生大多不在学校里学习，少了一种氛围，有问题往往不能及时得到解答，因此学习效果也大打折扣。

因此，如何帮助自学考生解决上述困难就是一件十分重要的事情。而一本好的辅导书对他们来说就显得至关重要。这也是我们写本套丛书的出发点。本书作者具有丰富的计算机类自学考试辅导教学经验，他们能准确地把握考生心理，编写出这套能够帮助自学考生在学习上达到事半功倍效果的丛书。

本辅导书是以全国高等教育自学考试委员会指定教材为基础，以考试大纲为依据编写而成的。书中每一章都按照大纲要求、重点难点提要、经典例题及解答技巧、教材练习题同步辅导和自测题五个部分进行安排，这五个环节的设定是从考生的要求出发考虑的。“大纲要求”部分使考生能够根据大纲提出的对知识把握的四个不同层次对教材中的内容有重点地加以学习；“重点难点提要”部分对教材中的重点难点做了解析，考生在学习教材的同时不妨加以参照，必定会从中受益匪浅；在“经典例题解题技巧”部分，我们从历年的考题中选择了一些经典例题进行解答指导，使考生掌握解题的技巧和基本方法；考虑到自学考生在解答指定教材中的课后习题时苦于没有标准答案可以参考的实际问题，我们在“教材练习题同步辅导”部分对指定教材中的课后习题都给出了标准答案，这样考生可以根据答案来检验自己对每一章知识的掌握程度；“自测题”部分中有些习题是近些年全国和各省市的自考考题，同时我们在书后附了自测题的标准答案，考生可以参照。所有这五个部分是紧密相扣的，也是考生在自学的时候对知识的掌握逐步深入的过程，所以如果考生能按照这样的顺序一步一步扎扎实实地学习，一定能取得很好的效果。

通过对近几年的考题研究分析我们了解到，本门课程的命题是严格按照大纲的要求来进行的，即“识记”为20%，“领会”为30%，“简单应用”为30%，“综合应用”为20%。四个层次的难易程度分别为易、较易、较难、难。所以考生在学习的时候应该注意对基本知识的掌握，要认真研读大纲，对大纲中重点要求的地方要多花些精力，而对于大纲中要求层次比较低的地方可以一带而过甚至可以忽略其中的一部分内容。本辅导书的编写正是在对考题和大纲的分析基础上形成的，所以说是专注于重点难点，而且五个环节的安排也是由浅入深的，希望考生严格按照辅导书的次序进行学习。

对于“面向对像程序设计”这门课，近两年全国的命题题型都是单项选择题、填空题、程序改错题、完成程序题和程序分析题五个部分，它们所占的分数比例分别为20%、20%、10%、20%、30%。书中每一章后面所附的自测题也正是按照这个要求来做的，希望考生要认真练习。另外，程序设计需要大量的实际编程训练才能达到效果，俗话说熟能生巧，那些

眼高手低的考生是绝对达不到理想效果的，所以平时多进行一些实际的编程训练是十分必要的。

本辅导书后面还附了几套模拟题和近两年的自考真题，以便考生在考试之前能有实战练习。每套模拟题和真题后面都附有标准答案，考生可以做完题后根据答案对自己的学习效果进行评测。

本书由乔川龙负责组稿，王兵、王睿伯编写。同时感谢科学出版社的同志们为本书的出版付出了大量的辛勤劳动。由于编者水平有限，错误之处在所难免，恳请各位考生以及同仁不吝赐教，以便再版时进行修正。本书也可以作为高等院校本、专科相关专业学生的学习参考用书。

编　者

2003 年 9 月于国防科技大学计算机学院

目　录

第 1 章　面向对象及 C++基础知识

1.1　大纲要求

课程内容

- 面向对象程序设计基础知识。
- 基本程序结构。
- 类型修饰符。
- 函数原型。
- 内联函数。
- 引用。
- 动态内存分配。
- 编译指令。
- 改变习惯重新思考。

自学要求

本章主要介绍面向对象设计的基础知识和 C++语言对 C 语言的基本改进部分。这些改进部分包括基本程序结构、类型修饰符、函数原型、内联函数、引用和编译指令等，并且讨论了如何改变习惯重新思考，利用已有知识学习 C++的问题。

本章的要求是理解面向对象程序设计的思想及 C++语言中的新思想，目的是为学习 C++的类打下基础。

考核知识点及考核要求

- 面向对象程序设计基础知识，要求达到“识记”层次。
  面向对象程序设计的思想。
  C++语言与 C 语言的关系。
- 基本程序结构，要求达到“综合应用”层次。
  序的基本结构和作用，正确书写序。
  通过对比 C 语言，认识 C++的注释、换行、标准输入及输出语句。
  头文件及常量的定义。
  不使用宏来定义常量的意义。

主函数的基本作用。

- 类型修饰符，要求达到“领会”层次。
  const 修饰符的使用方法。
  const 修饰符的作用。
- 函数原型，要求达到“综合应用”层次。
  函数原型的作用及意义。
  函数原型的使用方法。
- 动态内存分配，要求达到“简单应用”层次。
  new 的作用。
  delete 的作用。
- 内联函数，要求达到“综合应用”层次。
  内联函数的作用。
  内联函数的使用方法。
- 引用，要求达到“综合应用”层次。
  引用的作用。
  引用的使用方法。
- 编译指令，要求达到“领会”层次。
  编译指令的几种形式及其作用。
  在程序中应用编译指令。
- 改变习惯重新思考，要求达到“识记”层次。
  应从哪几个方面初步转变思想。
  C++的基本知识。
  使用已有知识学习 C++。

1.2 重点难点提要

1.2.1 面向对象程序设计基础知识

1. 概念

面向对象程序设计 是一种通过为数据和代码建立分块的内存区域，以便提供对程序进行模块化的程序设计方法，这种模块可以被用作样板，在需要的时候再建立其副本。

对象 是内存中的一块区域，通过将内存分块，每个模块在功能上保持相互间相对独立性。

面向过程 程序员不必了解计算机的内部逻辑结构，而是把精力主要集中在解决算法的逻辑和过程的描述上，通过程序把解决问题的执行步骤告诉计算机。

抽象 面向对象程序设计中的抽象概念要求程序员集中于事物的本质特征，而不是具体的细节，通过类来构成程序。

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。

多态性 不同的对象可以调用相同名称的函数，并且可以导致完全不同的行为的现象。

继承 面向对象程序设计中的继承概念是一个对象可以获得另一个对象的特性的机制。

2. 重点回顾

- 计算机语言的发展过程：从机器语言、汇编语言到高级语言的发展，从面向过程思想向面向对象设计思想的发展。
- C++与C语言的关系：C语言比C++语言的编译效率更高，但是，前者是后者的扩充。一方面，它将C语言作为它的子集，使得它与C语言兼容；另一方面，C++语言支持面向对象的程序设计，这是它对C的最重要的改进。总之，从开发时间、费用到形成的软件的可靠性和可重用性、可扩充性和可维护性上，C++都比C语言表现出了明显的优越性。

3. 问题

C++语言是纯的面向对象语言吗?

答：不是，至少它还含有main()这个全局函数。

1.2.2 C++基本程序结构

1. 概念

内联函数 使用关键字“inline”标志的函数，主要针对频繁使用的函数，提高编译效率，在编译时而不是运行时把该函数的目标代码插入每个调用这个函数的地方。

引用 给变量起别名，使新变量和原变量共用一个地址，见教材P10。

编译指令 包括嵌入指令、宏定义和条件编译指令，是用来指示编译器对源代码进行翻译之前的预处理操作。

函数原型 函数说明的一般形式：

类型 函数名 （参数类型说明列表）

基本程序结构 新的输入和输出风格；灵活的注释方式；告别宏定义；使用函数原型和缺省参数；新的动态内存分配函数；新的换行符。

2. 重点回顾

- 熟悉C++语言新的注释、换行、标准输入和输出语句的新变化。
- 告别宏定义：因为const修饰符和内联函数的引入，使得宏在C++语言中失去了存在的意义。
- 函数原型：这是C++语言对C语言的扩充，而且是C++编译器所必需的，但要十分注意省缺参数在参数序列中的位置及其对应规则。
- 引用：这是C++语言对C语言的扩充，它是通过引用运算符&来建立的。其主要应用有3个：用于变量别名、用于函数参数的引用传递方式和用于返回引用的函数。

3. 注意

- 为引用提供的初始值必须是一个也具有初始值的变量。

- 不能建立引用的数组。
- 不能对引用进行引用。
- 编译指令不是 C++语言的一部分。

4. 问题

- 为引用提供的初始值如果是一个常量，会发生什么情况?

请参见教材第 P11 页。

- 函数可以出现在复制运算符的左边吗?

答：可以，只要它的返回值可以作为左值。

- new 类型（数字）和 new 类型[数字]的意义相同吗?

答：不相同。

1.2.3 改变习惯重新思考

重点回顾

请通过以下几个方面来初步开始从 C 到 C++的转变：

- 通过重新编译已有的 C 语言程序来了解 C++语言；
- 通过重新设计 C 语言程序的核心部分掌握 C++的新特性；
- 将所有的外部变量的说明放到头文件中；
- 减少使用预处理；
- 重视函数类型。

1.3 经典例题及解题技巧

1. 下面是一个用户口令检查程序，请填空

```
#include<iostream.h>
#include<string-h>
const char*PASS="wolleh": (或coast char  PASS[]="wolleh":)//定义由PASS指针所指向的口令wolleh（说明：const可有可无）
void main()
{
  char user[10]://用于存放输入的口令
  cout  <<"plcase input your password:"<<endl: cin.getline(user.9):(或cin>>user:)          //输入口令
   if((strcmp(user, PASS))= =0)
cout<<"your password is correct"<<endl:
else
cout<<"your password is error" <<endl:
}
```

★根据提示，第一个空中应该定义一个字符串常量 PASS，而且因为它是一个验证程序，不改变 PASS 内容，所以最好把它定义成常量。

根据提示，要求用户输入一个串，根据上下文，填入“cin>>user”:，

2. 程序分析：给出运行结果

```
#include<iostream.h>
void main()
{
int * a:
int * &p=a:
int b = 10:
p = &b:
cout<<*a:
}
```

输出结果：10

★容易看出这是考查对指向指针的引用的掌握。由第二行 int * &p=a:知，a 是指针，p 是指向 a 的引用，那么所有对 p 的内容的改动都会完整地反映到 a 上。所以，对 p 进行赋值后，a 的内容（内含的地址）就变成 b 的地址了，所以输出为 b 的值 10。

3. 程序挑错

指出下面程序中的错误，在错误处说明出错原因。

```
#include <iostream.h>
const float pi = 3.1416;
const float r = 3.2;
void main() {
float s1,s2,c1,c2,r1;
c1 = pi*r*r;
s1 = 2*pi*r;
r = 2.8;   //r是常量，不能改变
c2 = pi*r*r;
s2 = 2*pi*r;
cout<<c1<<s2<<c2<<s2<<endl; }
```

★此处无它，弄清楚什么叫常量即可。

1.4 教材练习题同步辅导

1. 编程题

（1）赋值语句

```
cout<<"probleml"<<endl;
      cout<<"please input the radium"<<endl;

      V=PI*4*r*r*r/3;
      cout<<"the volume of the global is :"<<v<<endl;
      S=PI*r*r;
```

```
        cout<<"the area of the circle is:"<<S<<endl;
        cout<<"please input the resistance of r1 and r2"<<endl;
        cin>>r1>>r2;
        resistance=1/(1/r1+1/r2);
        cout<<"the resistance of the parallel resistance is "
           <<resistance<<endl;

        //that is the end of the problem1
1   V=PI*4*r*r*r/3;
2   S=PI*r*r;
3   resistance=1/(1/r1+1/r2);
```

(2) 成绩输入

```
//that is the end of the problem1
        cout<<"problem2"<<endl;
        initial(marks,50);//give initial value to the marks
        for(i=0;i<50;i++)
        {
               for (j=i+1;j<50;j++)
               {
                      if (marks[i]<marks[j])
                             { temp=marks[i];
                               marks[i]=marks[j];
                               marks[j]=temp;
                             };
               }
               cout<<marks[i]<<endl;
        }
        //this is the end of problem2
```

(3) new &delete

```
//problem3
        cout<<"problem3"<<endl;
        initial (number,20);
        count1=count2=0;
        for (i=0;i<20;i++)
        {
               if (number[i]>15) count1++;
               if (number[i]<15) count2++;
        }
        delete  number;
        cout<< "there are "<<count1<<"number bigger than 0"<<endl;
        cout<< "there are "<<count2<<"number smaller than 0"<<endl;
        //end of prohlem3
```

(4) n!计算

```
int  multi(int n)
{
```

```
int i;
int count=1;
for (i=1;i<=n;i++)
{
   count*=i;
};
return  count;

}
```

（5）编程题源程序

```
//--------------------------------------------------------------------------
#include<iostream.h>
#include <stdlib.h>
#include <time.h>
const float PI=3.1415926;

//--------------------------------------------------------------------------
void initial(float  *  value,int size);
void initial(int    *  number,int size);
int  multi(int n);

int main(int argc, char* argv[])
{      // for problem 1
       float r,r1,r2;
       float V,S,resistance;
       // volume, S ,resistance;
       // for problem 2
       int i ,j;
       float marks[50],temp;
       //for problem 3
       int * number=new int [20];
       int count1,count2;
       //for problem 4
       int n;
       //end of data definition
       cout<<"problem1"<<endl;
       cout<<"please input the radium"<<endl;

       V=PI*4*r*r*r/3;
       cout<<"the volume of the global is :"<<V<<endl;
       S=PI*r*r;
       cout<<"the area of the circle is:"<<S<<endl;
       cout<<"please input the resistance of r1 and r2"<<endl;
       cin>>r1>>r2;
       resistance=1/(1/r1+1/r2);
       cout<<"the resistance of the parallel resistance is "
          <<resistance<<endl;
```

```
        //that is the end of the problem1
        cout<<"problem2"<<endl;
        initial(marks,50);//give initial value to the marks
        for(i=0;i<50;i++)
        {
               for (j=i+1;j<50;j++)
               {
                      if (marks[i]<marks[j])
                             { temp=marks[i];
                               marks[i]=marks[j];
                               marks[j]=temp;
                             };
               }
               cout<<marks[i]<<endl;
        }
        //this is the end of problem2
        //problem3
        cout<<"problem3"<<endl;
        initial (number,20);
        count1=count2=0;
        for (i=0;i<20;i++)
        {
               if (number[i]>15) count1++;
               if (number[i]<15) count2++;
        }
        delete  number;
        cout<< "there are "<<count1<<"number bigger than 0"<<endl;
        cout<< "there are "<<count2<<"number smaller than 0"<<endl;
        //end of prohlem3
        cout<<"problem4"<<endl;
        cout<<"please input ur  n for n!";
        cin>>n;
        cout<<"the n! is "<<multi(n);
        //endof problem4
        cin>>r;
        return 0;
}
void initial(float  *  value,int size)
{
int i;
    randomize();
    for (i=0;i<size;i++)
    {
    value[i]=random(size);
    //cout<<"Random number in the 0-99 range:"<<value[i];
    }
}
```

```
void initial(int    *  number,int size)
{
    int i;
    randomize();
    for (i=0;i<size;i++)
    {

    number[i]=random(size);
    //cout<<"Random number in the 0-99 range:"<<number[i];
    }

}

int  multi(int n)
{
 int i;
 int count=1;
 for (i=1;i<=n;i++)
 {
    count*=i;
 };
 return   count;

}
//    ------------------------------------------------------------------
```

2. 分析程序题

(1) prog1

输出结果:

500 600 650

(2) prog2

输出结果:

500

600

500

1.5 自测题

1. 什么是面向对象程序设计?
2. 如何理解对象的封装性?
3. 面向对象程序设计语言有哪些特征?
4. 对象具有哪些特性?