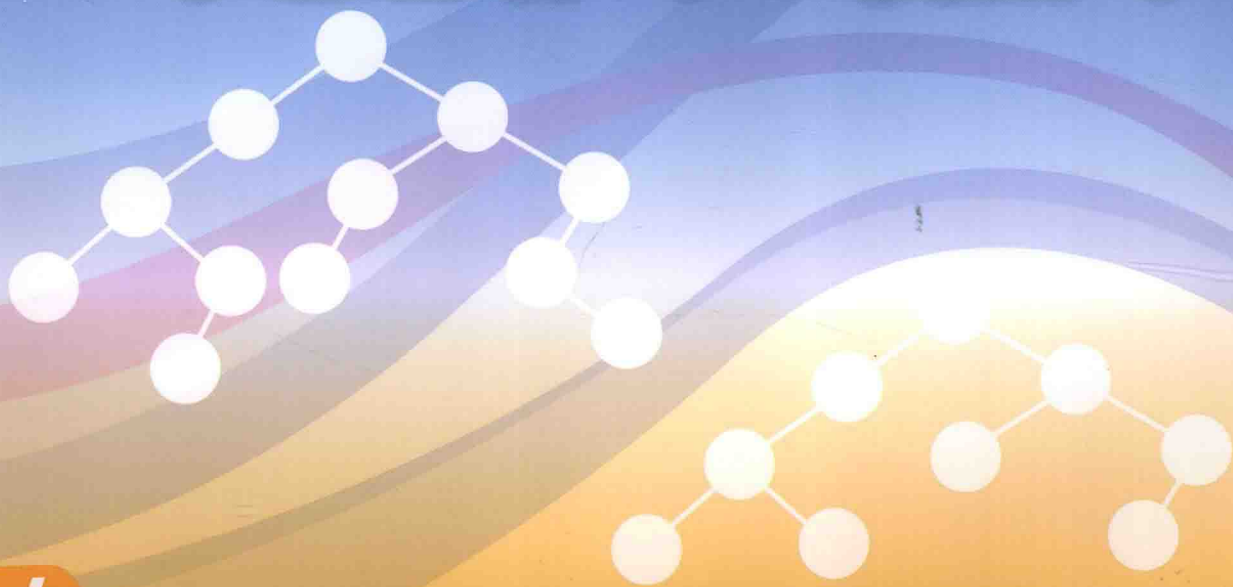


高等学校计算机专业规划教材

C++面向对象程序设计 习题解答与上机指导



邵兰洁 母俐丽 马睿 邢雁平 编著

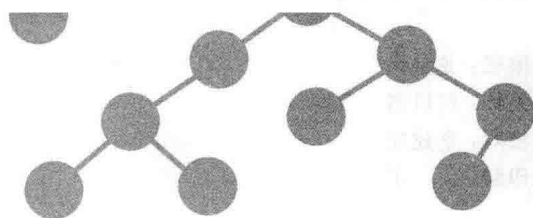
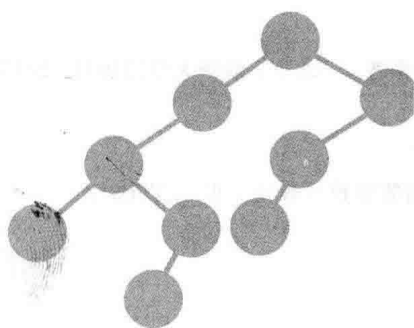
清华大学出版社



高等学校计算机专业规划教材

C++面向对象程序设计 习题解答与上机指导

邵兰洁 母俐丽 马睿 邢雁平 编著



清华大学出版社
北京

内 容 简 介

“程序设计”是一门实践性很强的课程,任何一门程序设计课程,不上机实践是不可能学好的。本书是邵兰洁主编的《C++ 面向对象程序设计》(清华大学出版社出版)的配套教学用书。全书共分3部分,第1部分是主教材习题解答,给出了主教材所有习题的解答。第2部分是上机指导,设计了10个实验,实验1介绍C++编程环境,给出了基于Visual C++ 6.0的编程和程序运行方法,以帮助读者快速掌握编程环境。除实验1外,其余9个实验都是每一个实验对应主教材的一章,针对每一章的知识点,精心设计每一个实验的内容。通过实验,不仅可以使读者进一步巩固所学知识,更重要的是让读者通过实验编程逐步掌握C++面向对象编程技巧。第3部分是全国计算机等级考试二级C++语言程序设计真题汇编,以帮助完成C++语言程序设计学习后想报考全国计算机二级C++考试的读者了解相关考试内容。

本书可作为高等院校计算机及相关专业本科生学习C++面向对象程序设计的实践环节用书,也可作为广大C++面向对象编程爱好者的编程训练指导参考书。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++ 面向对象程序设计习题解答与上机指导/邵兰洁等编著. —北京:清华大学出版社,2015
高等学校计算机专业规划教材

ISBN 978-7-302-40771-3

I. ①C… II. ①邵… 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(2015)第161808号



责任编辑:龙启铭

封面设计:何凤霞

责任校对:李建庄

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.25 字 数:454千字

版 次:2015年10月第1版 印 次:2015年10月第1次印刷

印 数:1~2000

定 价:35.00元

产品编号:065131-01



C++ 面向对象程序设计是一门实践性很强的课程,要想把自己所学的知识变成一种编程能力,上机实践是必不可少的。作者在教学过程中发现“眼高手低”的现象在学生中普遍存在,学生在上课或看书时,对教师所讲的内容或书中所阐述的内容都能够理解,但到自己编程时却感觉无从下手。为此,特编写此配套教学用书,希望能让读者在巩固所学 C++ 面向对象程序设计知识的同时,掌握 C++ 面向对象程序设计的核心技术,提高 C++ 编程实践能力。

本书具有如下特点:

(1) 内容丰富,选题典型,实用性强,力求在让读者巩固所学 C++ 知识点的同时,提高其运用 C++ 解决实际问题的能力。

(2) 重视读者实际编程能力的培养。上机实验中对编程题的解答,不仅给出程序参考代码,更进一步对代码进行了解析。

(3) 强调程序的可读性和标准化。书中程序全部遵循良好的程序设计风格:类名、函数名和变量名的定义做到“见名知义”;采用缩排格式组织程序代码;适当添加注释。所有程序均按照 C++ 标准编写,力求培养读者从一开始就写标准 C++ 程序的习惯。

本书为邵兰洁主编的《C++ 面向对象程序设计》的配套教学用书,共分 3 部分,第一部分是主教材习题解答,给出了主教材所有习题的解答。第二部分是上机指导,设计了 10 个实验,实验 1 介绍 C++ 编程环境,给出了基于 Visual C++ 6.0 编程和程序运行的方法,以帮助读者快速掌握编程环境。除第 1 个实验外,其余 9 个实验都是每个实验对应教材的一章,针对每章的知识点,精心设计每个实验的内容。通过实验,不仅可以使读者进一步巩固所学知识,更重要的是能让读者通过实验编程逐步掌握 C++ 面向对象编程技巧。第三部分是全国计算机等级考试二级 C++ 语言程序设计真题汇编,以帮助完成 C++ 语言程序设计学习后想报考全国计算机二级 C++ 考试的读者了解相关考试内容。

本书第一部分的主教材习题解答由邵兰洁、徐海云、马睿、张玉英编写,第二部分的上机指导由邵兰洁、邢雁平编写,第三部分的全国计算机等级考试二级 C++ 语言程序设计真题汇编由邵兰洁、母俐丽编写。全书由邵兰洁统稿并审定。



本书中给出的 C++ 程序参考代码不一定是最优的,它们仅仅代表了编者的思路 and 想法。欢迎读者提出自己的见解,帮助我们编写出更高质量的程序。同时欢迎读者对本书的内容提出批评和修改建议,我们将不胜感激。编者邮箱:shaolanjie789@163.com。

编者

2015年7月



第1部分 习题解答

- 第1章 面向对象程序设计概述 /3
- 第2章 面向过程程序设计概述 /18
- 第3章 类和对象 /28
- 第4章 继承与派生 /38
- 第5章 多态性与虚函数 /53
- 第6章 友元与静态成员 /59
- 第7章 运算符重载 /65
- 第8章 泛型编程 /77
- 第9章 输入输出 /86
- 第10章 异常处理 /95

第2部分 上机指导

- 实验1 Visual C++ 6.0 上机操作 /103
- 实验2 面向过程程序设计概述 /110
- 实验3 类和对象 /121
- 实验4 继承与派生 /144
- 实验5 多态性与虚函数 /166
- 实验6 友元与静态成员 /177
- 实验7 运算符重载 /180
- 实验8 泛型编程 /193
- 实验9 输入输出 /206



实验 10 异常处理 /213

第 3 部分 全国计算机等级考试二级 C++ 语言 程序设计真题汇编

选择题真题库 /221

操作题真题库 /256

附录 全国计算机等级考试二级 C++ 语言程序设计

考试大纲(2013 年版) /281

附录 1 全国计算机等级考试二级 C++ 语言程序设计真题汇编

1.1 选择题真题库 /221

1.2 操作题真题库 /256

1.3 附录 1 考试大纲(2013 年版) /281

1.4 附录 2 考试大纲(2013 年版) /281

1.5 附录 3 考试大纲(2013 年版) /281

1.6 附录 4 考试大纲(2013 年版) /281

1.7 附录 5 考试大纲(2013 年版) /281

1.8 附录 6 考试大纲(2013 年版) /281

1.9 附录 7 考试大纲(2013 年版) /281

第 1 部分

习 题 解 答

第 1 章

面向对象程序设计概述

一、简答题

1. 简述面向过程程序设计和面向对象程序设计的编程思想。

【答案要点】

面向过程程序设计的编程思想：

功能分解、逐步求精、模块化、结构化。当要设计一个目标系统时,首先从整体上概括出整个系统需要实现的功能,然后对系统的每项功能进行逐层分解,直到每项子功能都足够简单,不需要再分解为止。具体实现系统时,每项子功能对应一个模块,模块间尽量相对独立,通过模块间的调用关系或全局变量而有机地联系起来。

面向对象程序设计的编程思想：

(1) 客观世界中的事物都是对象(object),对象之间存在一定的关系。

(2) 用对象的属性(attribute)描述事物的静态特征,用对象的操作(operation)描述事物的行为(动态特征)。

(3) 对象的属性和操作结合为一体,形成一个相对独立、不可分的实体。对象对外屏蔽其内部细节,只留下少量接口,以便与外界联系。

(4) 通过抽象对对象进行分类,把具有相同属性和相同操作的对象归为一类,类是这些对象的抽象描述,每个对象是其所属类的一个实例。

(5) 复杂的对象可以用简单的对象作为其构成部分。

(6) 通过在不同程度上运用抽象的原则,可以得到一般类和特殊类。特殊类继承一般类的属性与操作,从而简化系统的构造过程。

(7) 通过关联表达类之间的静态关系。

(8) 对象之间通过传递消息进行通信,以实现对象之间的动态联系。

2. 面向对象程序设计有哪些优点?

【答案要点】

与传统的面向过程程序设计相比,面向对象程序设计的优点如下:

(1) 从认识论的角度看,面向对象程序设计改变了软件开发的方式。软件开发人员能够利用人类认识事物所采用的一般思维方式来进行软件开发。

(2) 面向对象程序中的数据的安全性高。外界只能通过对象提供的对外接口操作对象中的数据,这可以有效保护数据的安全。

(3) 面向对象程序设计有助于软件的维护与复用。某类对象数据结构的改变只会引

起该类对象操作代码的改变,只要其对外提供的接口不发生变化,程序的其余部分就不需要做任何改动。面向对象程序设计中类的继承机制有效解决了代码复用的问题。人们可以像使用集成电路(IC)构造计算机硬件那样,比较方便地重用对象类来构造软件系统。

二、编程题

运用以前曾学习过的 C 语言知识,编写一个实现自己所在班级全体同学本学期成绩管理的完整 C 程序。

【程序参考代码】

```

/* 学生信息管理系统 C 语言源代码 student.c */
#include<stdio.h>           /* 包含输入/输出头文件 */
#include<string.h>         /* 包含字符串处理头文件 */
#include<stdlib.h>
#define MAXSIZE 100       /* 能够处理的学生总人数,可以随意修改 */
typedef struct {           /* 用于存放生日信息的结构体 */
    int year;
    int month;
    int day;
}Date;
typedef struct Stud{      /* 用于存放学生信息的结构体 */
    char num[12];         /* 学号为 11 位 */
    char name[11];       /* 姓名,最多 5 个汉字 */
    char sex[2];         /* 性别,男记为 m,女记为 f */
    Date birthday;       /* 出生日期 */
    float English, DataStructure, Cplusplus; /* 三门课成绩 */
    float sum, average; /* 总成绩、平均成绩 */
}Student;

char curFile[40];        /* 存放当前正在操作的磁盘文件的文件名 */
int isOpen=0;           /* 当前是否有磁盘文件被打开标志 */
int found=0;            /* 在查找学生信息时是否找到标志 */
Student stud[MAXSIZE]; /* 用于存放读入内存中的所有学生信息的全局数组 */
int index=0;            /* 存放实际学生人数的全局变量 */

/* 各自定义函数原型声明 */
void create();          /* 新建学生信息文件 */
void open();            /* 打开学生信息文件,并读取学生信息到全局数组 stud 中 */
void display();        /* 显示学生信息 */
void search();         /* 查询学生信息 */
int searchNum(char * num); /* 按学号查询学生信息 */
int searchName(char * name); /* 按姓名查询学生信息 */
void append();         /* 添加学生信息 */
void modify();         /* 修改学生信息 */

```

```

void delete();          /* 删除学生信息 */
void total();          /* 统计所有学生某一科目总成绩 */
void sort();          /* 学生信息排序 */
void backup();        /* 备份学生信息 */
void menu()           /* 系统功能菜单 */
{   int choice;       /* 用于保存用户对功能菜单的选择结果 */
    for(;;)
    {   /* 显示系统功能菜单 */
        printf("\n*****\n");
        printf("***** 学生信息管理系统 *****\n");
        printf("***** \n");
        printf("***** 1. 新建学生信息文件 *****\n");
        printf("***** 2. 打开学生信息文件 *****\n");
        printf("***** 3. 显示学生信息 *****\n");
        printf("***** 4. 查询学生信息 *****\n");
        printf("***** 5. 添加学生信息 *****\n");
        printf("***** 6. 修改学生信息 *****\n");
        printf("***** 7. 删除学生信息 *****\n");
        printf("***** 8. 统计学生信息 *****\n");
        printf("***** 9. 学生信息排序 *****\n");
        printf("***** 10. 备份学生信息 *****\n");
        printf("***** 0. 退出系统 *****\n");
        printf("***** \n");
        printf("          请选择要执行的操作(0~10):_");
        scanf("%d", &choice);
        switch(choice){
            case 1: create(); break;
            case 2: open(); break;
            case 3: display(); break;
            case 4: search(); break;
            case 5: append(); break;
            case 6: modify(); break;
            case 7: delete(); break;
            case 8: total(); break;
            case 9: sort(); break;
            case 10: backup(); break;
            case 0: return;
            default: printf("选择错误!请重新选择.\n");
        } /* switch 结束 */
    } /* for 结束 */
}

void ReOrEx()
    /* 在用户执行完一项系统功能后,可以选择:是继续运行系统,还是退出系统 */
{   int n;

```

```

printf("\n***** \n");
printf("*****      1. 返回上级菜单      ***** \n");
printf("*****      0. 退出系统      ***** \n");
printf("***** \n");
printf("                请选择 (1/0)? _");
scanf("%d", &n);
if(n==0)
{ printf("\n***** \n");
  printf("*****      谢谢使用本系统!      ***** \n");
  printf("***** \n");
  exit(1);
}
}

void main()
{ printf("\n***** \n");
  printf("*****      欢迎使用学生信息管理系统!      ***** \n");
  printf("***** \n");
  system("pause");
  menu(); /* 系统功能以菜单的形式提供给用户 */
  printf("\n***** \n");
  printf("*****      谢谢使用本系统!***** \n");
  printf("***** \n");
} /* main 函数结束 */

/* 各自定义函数实现代码 */
int new(char * fileName) /* 创建磁盘文件 */
{ FILE * fp;
  if((fp=fopen(fileName, "w"))==NULL)
  { return 0; }
  else
  { fclose(fp); index=0; return 1; }
}

void create() /* 新建学生信息文件 */
{ char fileName[40];
  printf("请输入新建文件的名称: ");
  scanf("%s", &fileName);
  if(strcmp(fileName, "studentbackup"))
  { strcat(fileName, ".dat");
    if(!new(fileName))
      printf("%s 文件创建失败!\n", fileName);
    else
    { strcpy(curFile, fileName);
      printf("%s 文件创建成功!\n", fileName);
    }
  }
}

```

```
    }
    else
    { printf("%s 是备份文件,禁止创建与此文件同名的文件!\n", fileName); }
    ReOrEx();
}

void open() /* 打开学生信息文件 */
{ char fileName[40];
  printf("请输入要打开的数据文件的名称:");
  scanf("%s", &fileName);
  if(strcmp(fileName, "studentbackup"))
  { strcat(fileName, ".dat");
    if(isOpen==0)
    { FILE * fp;
      if((fp=fopen(fileName, "rb"))==NULL)
      { printf("%s 文件打开失败!\n", fileName); }
      else
      { isOpen=1;
        index=0;
        while(!feof(fp))
        { fread(&stud[index], sizeof(struct Stud), 1, fp);
          index++;
        }
        index--;
        printf("学生总人数为: %d\n", index);
        fclose(fp);
        printf("%s 文件打开成功!\n", fileName);
        strcpy(curFile, fileName);
      }
    }
    else
      printf("%s 文件已经打开!\n", fileName);
  }
  else
    printf("%s 是备份文件,禁止打开此文件!\n", fileName);
  ReOrEx();
}

void display() /* 显示全部学生信息 */
{ int i;
  if(!strcmp(curFile, ""))
  { printf("当前并未打开或新建文件,无法显示!\n"); }
  else
  { printf("\n          显示所有学生成绩信息\n\n");
    printf("%--12s%--11s%--5s%--14s%--12s%--15s%--12s%--12s%--12s\n",
      "n", "Num", "Name", "Sex", "Birthday", "English", "DataStructure",
```

```

    "CPlusPlus", "Sum", "Average");
    for(i=0; i<index; i++)
    {   printf("%-12s%-11s%-5s%4d/%2d/%2d%10.2f%15.2f%12.2f%12.2f
            %12.2f\n\n", stud[i].num, stud[i].name, stud[i].sex,
            stud[i].birthday.year, stud[i].birthday.month,
            stud[i].birthday.day, stud[i].English,
            stud[i].DataStructure, stud[i].CPlusPlus,
            stud[i].sum, stud[i].average);
    }
}
ReOrEx();
}
int searchNum(char * num)                /* 按学号查询学生信息 */
{   int i;
    for(i=0; i<index; i++)
    {   if(!strcmp(stud[i].num, num))
        {   printf("对应此学号的学生信息存在!\n");
            found=1;
            return i;
        }
    }
    printf("没有此学生的信息!\n");
    found=0;
    return 0;
}
int searchName(char * name)              /* 按姓名查询学生信息 */
{   int i;
    for(i=0; i<index; i++)
    {   if(!strcmp(stud[i].name, name))
        {   printf("对应此学号的学生信息存在!\n");
            found=1;
            return i;
        }
    }
    printf("没有此学生的信息!\n");
    found=0;
    return 0;
}
void search()                            /* 查询学生信息 */
{   int n;
    int i;
    char num[12];
    char name[10];
    if(!strcmp(curFile, ""))

```

```
{ printf("当前并未打开或新建文件,无法查询!\n"); }
else
{ printf("\n                查询某一学生信息\n");
  printf("*****\n");
  printf("*****      1. 按学号查询      *****\n");
  printf("*****      2. 按姓名查询      *****\n");
  printf("*****\n");
  printf("                请选择(1/2)? _");
  scanf("%d", &n);
  if(n==1)
  { printf("请输入学生学号: \n");
    scanf("%s", num);
    i=searchNum(num);
  }
  else if(n==2)
  { printf("请输入学生姓名: \n");
    scanf("%s", &name);
    i=searchName(name);
  }
  printf("该学生的具体信息为: \n\n");
  printf("%--12s--11s--5s--14s--12s--15s--12s--12s--12s\n",
        "Num", "Name", "Sex", "Birthday", "English", "DataStructure",
        "Cplusplus", "Sum", "Average");
  printf("%--12s--11s--5s%4d/%2d/%2d%10.2f%15.2f%12.2f%12.2f
        %12.2f\n\n", stud[i].num, stud[i].name, stud[i].sex,
        stud[i].birthday.year, stud[i].birthday.month,
        stud[i].birthday.day, stud[i].English, stud[i].DataStructure,
        stud[i].Cplusplus, stud[i].sum, stud[i].average);
  }
  ReOrEx();
}

void addData() /* 添加一条学生信息 */
{ char num[12];
  char name[10];
  char sex[2];
  int year, month, day;
  float English, DataStructure, CPP;
  int location;
  if(index>=MAXSIZE)
  { printf("错误!学生信息已满,不能添加!\n"); }
  else
  { printf("执行添加学生信息操作!\n");

    printf("\n 请输入学生学号: ");
```



```
scanf("%s", num);
location=searchNum(num);
if (!found)
{ printf("可以进行添加操作!\n");
  printf("\n 请输入学生姓名:");
  scanf("%s", name);
  printf("\n 请输入学生性别:");
  scanf("%s", &sex);
  printf("\n 请输入学生出生年份:");
  scanf("%d", &year);
  printf("\n 请输入学生出生月份:");
  scanf("%d", &month);
  printf("\n 请输入学生出生日:");
  scanf("%d", &day);
  printf("\n 请输入学生英语成绩:");
  scanf("%f", &English);
  printf("\n 请输入学生数据结构成绩:");
  scanf("%f", &DataStructure);
  printf("\n 请输入学生 C++程序设计成绩:");
  scanf("%f", &CPP);
  printf("\n");
  strcpy(stud[index].num, num);
  strcpy(stud[index].name, name);
  strcpy(stud[index].sex, sex);
  stud[index].birthday.year=year;
  stud[index].birthday.month=month;
  stud[index].birthday.day=day;
  stud[index].English=English;
  stud[index].DataStructure=DataStructure;
  stud[index].CPlusPlus=CPP;
  stud[index].sum=English+DataStructure+CPP;
  stud[index].average=stud[index].sum/3;
  index++;
  printf("插入一条学生信息操作成功!\n");
}
else printf("不能进行添加学生信息操作!\n");
}
}

void save(char * fileName) /* 学生信息存盘 */
{ FILE * fp;
  int i;
  if((fp=fopen(fileName, "wb"))==NULL)
  { printf("文件打开失败!"); return; }
  for(i=0; i<index; i++)
```