课 程 设 计 报 告 1

**课程名称 计算机程序设计基础2**

**班 级 无六五**

**学 号 2016011109**

**姓 名 王春禹**

**指导助教 无**

**2017年 7月 5日**

目 录

[1. 系统需求分析 1](#_Toc171224774)

[2. 总体设计 1](#_Toc171224775)

[3. 详细设计 1](#_Toc171224776)

[4. 系统调试 1](#_Toc171224777)

[5. 结果分析 1](#_Toc171224778)

[6. 总结 1](#_Toc171224779)

附录：源程序清单、文件、使用说明书、 评分表

**1.系统需求分析**

**该系统记录了学生的成绩情况，它包括：学生姓名、学号、五门课的成绩（微积分、计算机、大学物理、英语、体育）、平均绩点、排名等信息。能提供以下功能：**

1. **学生登录界面：**

输入姓名、学号即可查看自己的全部成绩及gpa、排名

1. 教师登录界面：
   1. 录入学生成绩。按照提示，一次录入一个学生的所有成绩
   2. 修改学生成绩。系统列出所有学生的成绩信息，用户选择要修改的序号，并选择要修改的部分，按提示输入修改后的信息
   3. 查询学生成绩。输入学生姓名，即可查看该学生的所有信息
   4. 统计某课程信息。输入课程名，可查看该课程的学分、最高分、最低分、平均分、及格率
   5. 统计所有学生绩点。按gpa排名输出所有学生的绩点
   6. 删除学生信息。选择要删除的信息序号进行删除
   7. 离开系统并自动保存更改到文件

# **2.总体设计**

学生成绩管理系统包含两个功能菜单：学生界面和教师界面，学生界面利用姓名、学号登录后可查看自己的成绩，教师界面功能包括：录入成绩、修改信息、查询学生成绩、统计课程情况、统计学生成绩、删除信息。总之，用户可以方便的在系统提示下进行使用。

学生成绩管理系统功能模块图：

录入学生成绩

统计学生绩点

删除学生信息

统计课程情况

查询学生成绩

修改学生信息

查看个人成绩

姓名、学号登录

教师界面

学生界面

学生成绩管理系统

**3.详细设计**

学生成绩管理系统中四个类的类层次图为：

xitong类

course类

stu类

xinxi类

学生成绩管理系统中各功能模块的实现：

学生界面：

查看

自己的各科绩点、gpa等

输入姓名、学号

教师界面：

输入姓名

输入学号

按百分制输入各科成绩

录入学生成绩

输入学生姓名

选择要修改的信息

输入修改后的信息

修改成功

修改学生成绩

输入学生姓名

输出该学生的各科成绩、gpa、排名

查询学生成绩

统计某课程情况

输入课程名

显示该门课程的学分、最高（低）分、平均分、及格率

输出所有学生的绩点及排名

统计学生绩点

删除学生信息

显示所有学生的信息

选择要删除的序号进行删除

成绩管理系统中四个类的UML图为：

xinxi

private:

-name: string

-id: string

-cla: string

-score: double \*

public:

+xinxi() + operator ==(xinxi &xx1, xinxi &xx2);

+setname(string n): void

+setid(string i): void

+setsco(double \*g) : void

+setsco(double g, int i): void

+getname() : string

+getid() : string

+getsco(): double \*

+getsco(int i) :double \*

+print(): void -num: int

course

private:

-cla: string

-max: double

-min: double

-ave: double

-rate: double

-point: int

public:

+course()

+~course()

+setcla(string c): void

+setpoint(int p) : void

+getcla() : string

+getpoint(): int

+getmax(); int

+getmin(); double

+getave(); double

+getrate(); double

-num; int

stu

Private:

-name, id:string

-gpa; double

-grade: double \*

-rank: int

Public:

+stu() +print(): void

+~stu() -num:int

+setname(string n): void

+setid(string i): void

+setgrade(double \*g); void

+setgrade(double g, int i); void

+getname(): string

+getid(): string

+getgrade(int i) :string

+getgpa():double

+getrank():int

void print()

抽象类

派生类

派生类

xitong

+setname(string n) : virtual void

+setid(string i) : virtual void;

+getname() : virtual string

+getid() : virtual string

+print() : virtual void

+~xitong() {}

**4.系统调试**

在编写程序的过程中，我都是边写边调，同时由于vs15自带纠错功能，调试的过程还算顺利。而且最开始我写的是学生考勤管理系统，写完报告才发现要写成绩管理系统，所以有了一定经验，出现的问题较少。主要发现并解决了以下问题：

（1）在信息类和学生类中，存储分数的私有成员均为double型指针，在set函数里直接传递指针名发现数据无法读入，然后改为一个一个赋值，提示地址错误，后来在xinxi类和stu类构造函数里为该指针新开了大小为kcshu的内存，程序正常运行

（2）原来录入成绩都是以绩点形式录入，后来考虑到老师录入期末成绩是以百分制，统计课程的最高分、最低分、平均分也是一百分制形式，所以录入改为百分制录入，在显示学生成绩的时候才会转化为绩点形式，而不及格则绩点为0

（3）我发现两个int数a,b,语句double(a/b)并不会将其转换成double数，将其中一个改为double数，a/b即可变为double

（4）原来我在每个功能函数执行的最后都会通过调用caidan函数回到菜单，后来一想，这样程序执行时是嵌套关系，没有跳出函数，函数中新开的内存无法释放，所以我改为了在case语句中先调用函数，再回到caidan函数

（5）在从文件录入数据时，原来在考勤管理系统里用的是getline()，遇到空格截止，结果用到这里无论如何也读不掉每行的换行符，读掉了就会导致名字无法正常读入。不明原因，但改成了is>>读入就好了

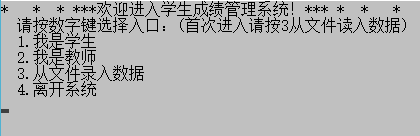
（6）stu类中有一个函数需要调用course类的对象数组，我做了向前引用说明，并紧接着定义了course类的数组，结果编译不通过，因为我还没定义course类，所以我把这个stu类函数的定义放到了course类定义之后，编译通过

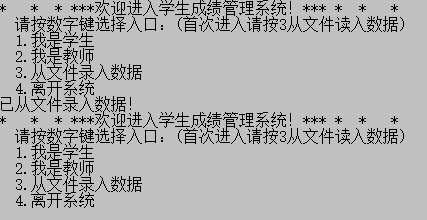
（7）编写过程中还有很多低级失误导致的奇奇怪怪的错误，通常是由于字母太多搞混的原因，这种错误很难找出来，只能设置断点一点点检查、排查

**5．测试结果与分析**

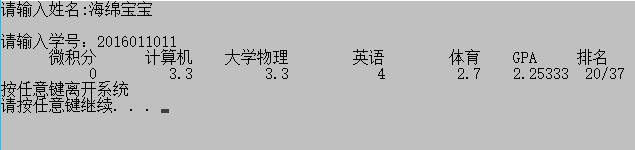
本程序测试文件为“测试数据.txt”，离开系统时读入到该文件中。

测试结果如下：

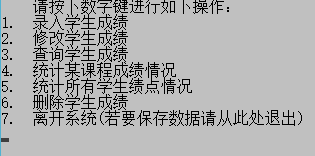




**学生界面：**



**教师界面：**



其他功能还请助教试用，就不一一展示了

这次的程序功能还是有些简单，只能查看预定的五门课成绩，录入时也只能五门成绩一起录，还是不很方便。学生界面功能只有查看所有成绩，而教师界面没有登录验证

程序写得也很初级，首先没有用到链表操作，对象的建立只能靠数组，这样会浪费大量内存，但是我对链表的操作并不得心应手，这里只好图个省事。

其次类的建立并不是很好。录入的信息是以每个学生为单位的，每个stu类对象和一个xinxi类对象对应，两者有较大的重叠性，xinxi类显得多余。其实最初也是为了达到多继承的要求加上的xinxi类，后来发现两者区分开还是有必要的，比如两者的成绩就不一样，一个是百分制，一个是绩点。其实，在原来写的考勤管理系统里我的类关系还比较清晰，xinxi类派生出学生类和课程类。这里课程类就独立出来了，实在没法继承。

总之，功能还算齐全，运行结果还可以，但代码比较low吧。至于创新性，还请助教体验后下结论吧

**6.总结**

头一次写500多行的大作业，对于我来说还是有很好的锻炼作用的。

头一次真正体会到类在面向对象的编程中的作用，对于类的建立有了更清楚的认识。运算符重载只是简单地重载了一个判断xinxi 类对象相等的“==”。但是出于方便，类的继承只有公有继承。

文件录入时对函数的调用不太熟练，最后找到了比较合适的is>>录入，平时对文件的操作运用比较少的原因吧。

还有就是我在开始写之前对于类的建立不是很确定，没有预定好功能，导致写程序时边写边加，不是很好的习惯。

写程序的过程中随时都在debug，真正体会到了断点法的实用性，还好bug不是很多，找起来也很快。

**附录1：源程序清单 （共计547行）**

#include<iostream>

#include<cstring>

#include<iomanip>

#include<string>

#include<cstdlib>

#include<fstream>

#include<conio.h>

using namespace std;

static int xsshu = 0, kcshu = 5, xxshu = 0; //信息数、学生数、课程数作为全局变量

static int r = 0;

class xitong //xitong抽象类

{

public:

virtual void setname(string n) = 0;

virtual void setid(string i) = 0;

virtual string getname() = 0;

virtual string getid() = 0;

virtual void print() = 0;

virtual ~xitong() {};

};

class xinxi //xinxi类

{

public:

xinxi(){ score = new double[kcshu]; }

void setname(string n) { name = n; }

void setid(string i) { id = i; }

void setsco(double \*g) { for (int i = 0; i < kcshu; i++) score[i] = g[i]; }

void setsco(double g, int i) { score[i] = g; }

string getname() { return name; }

string getid() { return id; }

double \*getsco() { return score; }

double getsco(int i) { return score[i]; }

friend bool operator ==(xinxi &xx1, xinxi &xx2);

int num;

void print()

{

cout <<num<<". "<<name <<" "<< id <<" ";

for (int i = 0; i < kcshu; i++)

cout <<" "<< score[i];

cout << endl;

}

private:

string name, id,cla;

double \*score;

};

xinxi xx[200];

bool operator ==(xinxi &xx1, xinxi &xx2)

{

if (xx1.getname() == xx2.getname() && xx1.getid() == xx2.getid() && xx1.getsco() == xx2.getsco())

return 1;

else return 0;

}

class course;

class stu :public xinxi //stu类

{

public:

stu() { grade = new double[kcshu]; }

~stu() { }

void setname(string n) {name = n; }

void setid(string i) { id = i; }

void setgrade(double \*g);

void setgrade(double g, int i);

string getname() { return name; }

string getid() { return id; }

double getgrade(int i) { return grade[i]; }

double getgpa();

int getrank();

void print()

{

cout <<num<<". "<< name <<" "<< id << " ";

for (int i = 0; i < kcshu; i++)

cout << " " << getgrade(i);

cout << setw(10) << getgpa() << setw(4) << getrank() << "/" << xsshu;

cout << endl;

}

int num;

private:

string name, id;

double gpa;

double \*grade;

int rank;

};

stu st[100];

void stu::setgrade(double \*g) //转换绩点函数

{

for (int i = 0; i < kcshu; i++)

{

if (g[i] >= 95) grade[i] = 4.0;

else if (g[i] >= 90) grade[i] = 3.7;

else if (g[i] >= 85) grade[i] = 3.3;

else if (g[i] >= 80) grade[i] = 3.0;

else if (g[i] >= 77) grade[i] = 2.7;

else if (g[i] >= 73) grade[i] = 2.3;

else if (g[i] >= 70) grade[i] = 2.0;

else if (g[i] >= 67) grade[i] = 1.7;

else if (g[i] >= 63) grade[i] = 1.3;

else if (g[i] >= 60) grade[i] = 1.0;

else grade[i] = 0.0;

}

}

void stu::setgrade(double g, int i)

{

if (g >= 95) grade[i] = 4.0;

else if (g >= 90) grade[i] = 3.7;

else if (g >= 85) grade[i] = 3.3;

else if (g >= 80) grade[i] = 3.0;

else if (g >= 77) grade[i] = 2.7;

else if (g >= 73) grade[i] = 2.3;

else if (g >= 70) grade[i] = 2.0;

else if (g >= 67) grade[i] = 1.7;

else if (g >= 63) grade[i] = 1.3;

else if (g >= 60) grade[i] = 1.0;

else grade[i] = 0;

}

int stu::getrank() //排名函数

{

int r = 1;

for (int i = 0; i < xsshu; i++)

if (st[i].getgpa() > this->getgpa()) r++;

return r;

}

class course //course类

{

public:

course() {};

~course() {};

void setcla(string c) { cla = c; }

void setpoint(int p) { point = p; }

string getcla() { return cla; }

int getpoint() { return point; }

double getmax();

double getmin();

double getave();

double getrate();

int num;

private:

string cla;

double max, min, ave,rate;

int point;

};

double course::getmax() //获得最高分函数

{

double m = 0;

if (xxshu)

{

for (int i = 0; i < xxshu; i++)

if (xx[i].getsco(num) > m)

m = xx[i].getsco(num);

return m;

}

else return 0;

}

double course::getmin() //获得最低分函数

{

double m = 100;

if (xxshu)

{

for (int i = 0; i < xxshu; i++)

if (xx[i].getsco(num) < m)

m = xx[i].getsco(num);

return m;

}

else return 0;

}

double course::getave() //获得平均分函数

{

double s = 0;

if (xxshu)

{

for (int i = 0; i < xsshu; i++)

s += xx[i].getsco(num);

return s / xxshu;

}

else return 0;

}

double course::getrate() //获得及格率函数

{

double m=0.0;

for (int i = 0; i < xsshu; i++)

if (st[i].getgrade(num))

m=m+1;

return m / xsshu;

}

course cou[5];

double stu::getgpa() //getgpa函数，计算学生绩点

{

double s = 0.0, t = 0.0;

for (int i = 0; i < kcshu; i++)

{

s += getgrade(i) \* cou[i].getpoint(); t += cou[i].getpoint();

}

if (t)

return s / t;

else return 0.0;

}

void caidan1(); //函数声明

void caidan2();

void luru();

void luru1();

void xiugai();

void chaxun();

void tongji1();

void tongji2();

void shanchu();

void shuchu();

void caidan() //菜单

{

cout << "\* \* \* \*\*\*欢迎进入学生成绩管理系统！\*\*\* \* \* \*" << endl;

cout << " 请按数字键选择入口：(首次进入请按3从文件读入数据)" << endl;

cout << " 1.我是学生" << endl; cout << " 2.我是教师" << endl;

cout << " 3.从文件录入数据" << endl; cout << " 4.离开系统" << endl;

if (r)

switch (\_getch())

{

case '1':caidan1(); break;

case '2':caidan2(); break;

case '3':luru1(); caidan(); break;

case '4':exit(1);

default:system("cls"); caidan();

}

else if (\_getch() == '3') { luru1(); r++; caidan(); }

else { system("cls"); caidan(); }

}

void caidan1() //学生菜单

{

system("cls");

string a, b; int flag = 0; stu st1;

cout << "请输入姓名:";

cin >> a;

for (int i = 0; i < xsshu; i++)

if (st[i].getname() == a) flag = 1;

if (flag)

{

int k = 0;

while (!k)

{

cout << endl << "请输入学号：";

cin >> b;

for (int i = 0; i < xsshu; i++)

if (st[i].getid() == b) { k = 1; st1 = st[i]; }

if (k) {

for (int i = 0; i < kcshu; i++)

cout << setw(12) << cou[i].getcla();

cout << " GPA "<<" 排名" << endl;

for (int i = 0; i < kcshu; i++)

cout << setw(12) << st1.getgrade(i);

cout <<" "<< st1.getgpa() << " " << st1.getrank() << "/" << xsshu << endl;

}

else cout << "学号错误！"<<endl ;

}

}

else cout << "查无此人！"<<endl;

cout << "按任意键离开系统" << endl; system("pause");

}

void caidan2() //教师菜单

{

system("cls");

cout << " 请按下数字键进行如下操作：" << endl;

cout << "1. 录入学生成绩" << endl;

cout << "2. 修改学生成绩" << endl;

cout << "3. 查询学生成绩" << endl;

cout << "4. 统计某课程成绩情况" << endl;

cout << "5. 统计所有学生绩点情况" << endl;

cout << "6. 删除学生成绩" << endl;

cout << "7. 离开系统(若要保存数据请从此处退出)" << endl;

switch (\_getch())

{

case '1':luru(); caidan2(); break;

case '2':xiugai(); caidan2(); break;

case '3':chaxun(); caidan2(); break;

case '4':tongji1(); caidan2(); break;

case '5':tongji2(); caidan2(); break;

case '6': shanchu(); caidan2(); break;

case '7': shuchu(); exit(1); break;

default:system("cls"); caidan2();

}

}

void luru1() //从文件录入函数

{

string str; char a; double g[5];

ifstream is("测试数据.txt", ios\_base::in);

if (is.good())

{

is >> xxshu; xsshu = xxshu;

for (int i = 0; i < xxshu; i++)

{

is >> str; xx[i].setname(str); st[i].setname(str);

is >> str; xx[i].setid(str); st[i].setid(str);

for (int j = 0; j < kcshu; j++)

is >> g[j];

xx[i].setsco(g); st[i].setgrade(g);

xx[i].num=st[i].num=i;

}

cout << "已从文件录入数据！" << endl;

}

else {

cout << "读入错误！"; system("pause");

}

is.close();

}

void luru() //手动录入函数

{

system("cls");

string a, b; double g[5], x; int flag = 0;

cout << "请输入 姓名 学号："<<endl;

cin >> a >> b;

if (xxshu)

{

for (int i = 0; i < xxshu; i++)

if (xx[i].getid() == b)

flag = 1;

}

if (flag)

{

cout << "该学生成绩信息已存在！若要重新录入请选择修改" << endl; system("pause");

}

else

{

xx[xxshu].setname(a); xx[xxshu].setid(b);

st[xsshu].setname(a); st[xsshu].setid(b);

for (int i = 0; i < kcshu; i++)

{

cout << cou[i].getcla() << " 成绩（百分制）：";

cin >> x; while (x < 0 || x>100) { cout << "成绩有误！请重新输入" << endl; cin >> x; }

g[i] = x;

}

xx[xxshu].setsco(g); st[xsshu].num=xx[xxshu].num = xxshu; st[xsshu].setgrade(g);

xxshu++; xsshu++;

cout << "录入成功！是否继续录入？（y/n）";

if (\_getch() == 'y') luru();

}

}

void xiugai() //修改函数

{

system("cls"); string a,b; int i,flag,j;

cout << "输入要修改的学生姓名：";

cin >> a;

int k = 0;

for ( i = 0; i < xxshu; i++)

if (xx[i].getname() == a)

{

k++; break;

}

if (!k)

{

cout << "没有相关信息呀，是否更改名字继续查询？(y/n)" << endl;

if (\_getch() == 'y')

xiugai();

}

else {

cout << "序号 姓名 学生号 ";

for (j = 0; j < kcshu; j++)

cout << setw(8) << cou[j].getcla();

cout << endl;

xx[i].print();

cout << "请按数字键选择要修改的项：1.姓名 2.学号 3.成绩" << endl;

flag = 0;

switch (\_getch())

{

case'1':

do {

cout << "请输入修改后的姓名："; cin >> b;

st[i].setname(b); xx[i].setname(b);

for (int j = 0; j < xxshu; j++)

if (xx[j] == xx[i]&&j!=i) {

cout << "信息已存在！请重新输入" << endl; flag = 1;

}

} while (flag == 1);

cout << "是否继续修改？（y/n）";

if (\_getch() == 'y') xiugai(); break;

case'2':

cout << "请输入修改后的学号："; cin >> b;

for (i = 0; i < xsshu; i++)

if (st[i].getid() == b) flag = 1;

if (flag) { cout << "该学号已存在！"; system("pause"); break; }

else st[i].setid(b); xx[i].setid(b);

cout << "是否继续修改？（y/n）";

if (\_getch() == 'y') xiugai(); break;

case'3':

cout << "请选择要修改的课程：1.微积分 2.计算机 3.大学物理 4.英语 5.体育" << endl;

cin >> i;

cout << "请输入修改后的成绩（百分制）："; double x;

cin >> x; while (x < 0 || x>100) { cout << "成绩有误！请重新输入" << endl; cin >> x; }

st[i].setgrade(x, i - 1); xx[i].setsco(x, i - 1);

cout << "是否继续修改？（y/n）";

if (\_getch() == 'y') xiugai(); break;

}

}

}

void chaxun() //查询函数

{

system("cls");

string a; int i,flag = 0;

cout << "请输入要查询的学生姓名：";

cin >> a;

for ( i = 0; i < xsshu; i++)

if (st[i].getname() == a) { flag = 1; break;}

if(flag)

{

cout << "序号 姓名 学生号 ";

for (int j = 0; j < kcshu; j++)

cout << setw(9) << cou[j].getcla();

cout << " GPA " << "排名"<<endl;

st[i].print();

cout << "是否继续查询（y/n）";

if (\_getch() == 'y') chaxun();

}

else {

cout << "未找到该学生信息！"; system("pause");

}

}

void tongji1() //统计课程情况函数

{

system("cls");

string a; int i,flag=0;

cout << "请输入课程名：";

cin >> a;

for (i = 0; i < 5; i++)

if (a == cou[i].getcla())

{

flag = 1; break;

}

if (flag) {

cout << "课程号：" << cou[i].num << endl;

cout << "学分：" << cou[i].getpoint() << endl;

cout << "最高分：" << cou[i].getmax() << endl;

cout << "最低分：" << cou[i].getmin() << endl;

cout << "平均分：" << cou[i].getave() << endl;

cout << "及格率：" << cou[i].getrate() << endl;

cout << "是否继续统计？(y/n)";

if (\_getch() == 'y') tongji1();

}

else { cout << "无此课程！"; system("pause"); }

}

void tongji2() //统计学生绩点函数

{

system("cls");

int b = 0, k = xsshu, m = k - 1; stu d;

cout << "名次 姓名 学生号 GPA "<<endl;

while (b < m)

{

int j = m - 1; m = 0;

for (int i = b; i <= j; i++)

if ((st[i].getrank()>st[i + 1].getrank())

|| (st[i].getrank() == st[i + 1].getrank()&& st[i].getid()> st[i + 1].getid()))

{

d = st[i]; st[i] = st[i + 1]; st[i + 1] = d; m = i;

}

j = b + 1; b = 0;

for (int i = m; i >= j; i--)

if ((st[i - 1].getrank()>st[i].getrank())

|| (st[i - 1].getrank() == st[i].getrank() && st[i - 1].getid()> st[i].getid()))

{

d = st[i]; st[i] = st[i - 1]; st[i - 1] = d; b = i;

}

}

for (int i = 0; i < xsshu; i++)

cout << st[i].getrank()<<"."<<std::left<<st[i].getname() <<std::right<<setw(14)<< st[i].getid() << " " << st[i].getgpa() << endl;

cout << "按任意键返回主菜单"; system("pause");

}

void shanchu() //删除函数

{

system("cls");

string b; int flag = 0, i;

cout << "序号 姓名 学生号 ";

for (i = 0; i < kcshu; i++)

cout << setw(8) << cou[i].getcla();

cout << endl;

for (i = 0; i < xxshu; i++)

xx[i].print();

cout << "请输入要删除的信息序号:";

int a; cin >> a;

while (a<0 || a>xxshu - 1) { cout << "请重新输入："; cin >> a; }

for (int i = 0; i < xxshu; i++)

if (a ==xx[i].num)

{

if (a < xxshu - 1)

{

for (int j = a; j < xxshu - 1; j++)

{

xx[j] = xx[j + 1]; xx[j].num--;

}

xxshu--;

}

else xxshu--;

}

for (int i = 0; i < xsshu; i++)

if (a == st[i].num)

{

if (a < xsshu - 1)

{

for (int j = a; j < xsshu - 1; j++)

{

st[j] = st[j + 1]; st[j].num--;

}

xsshu--;

}

else xsshu--;

}

cout << "是否继续删除？(y/n)";

if (\_getch() == 'y') shanchu();

}

void shuchu() //数据输出到文件函数

{

ofstream tfile("测试数据.txt");

tfile << xxshu << endl;

for (int i = 0; i < xxshu; i++)

{

tfile << xx[i].getname() << " " << xx[i].getid();

for (int j = 0; j < kcshu; j++)

tfile << " "<<xx[i].getsco(j);

tfile << endl;

}

tfile.close();

}

void main()

{

system("color 70");

cou[0].setcla("微积分"); cou[0].setpoint(5); cou[0].num = 0; //设置五门课程

cou[1].setcla("计算机"); cou[1].setpoint(4); cou[1].num = 1;

cou[2].setcla("大学物理"); cou[2].setpoint(3); cou[2].num = 2;

cou[3].setcla("英语"); cou[3].setpoint(2); cou[3].num = 3;

cou[4].setcla("体育"); cou[4].setpoint(1); cou[4].num = 4;

caidan();

}

**附录2：测试数据文件与写入数据文件**

测试数据文件为“测试数据.txt”，

写入数据文件为“测试数据.txt”，在按7离开系统时会自动将新数据写入其中

放在文件包里了

**附录3：使用说明书**

该系统为大学成绩管理系统，包括五门课程：微积分、计算机、大学物理、英语、体育

运行时以菜单方式工作，分为学生界面和教师界面，

第一次进入时须先按3进行文件数据读入，按其他键无效

录入后，按1可进入学生界面。输入姓名、学号进行登录，登录后系统显示各科成绩、gpa及排名

按2可进入教师界面。共有7个选项，对应7个功能

1. 录入学生成绩。按照提示依次录入学生姓名、学号、各科成绩。注：学号为每个学生唯一的独有标识，若学号与之前录入的学生学号重复，系统会提示
2. 修改学生信息。输入要修改的学生姓名，选择要修改的项目：姓名、学号、成绩。之后按提示输入修改后的内容即可。注：若修改后的信息与其他信息相同系统会提示重复
3. 查询学生成绩。输入要查询的学生姓名，即可查看该生各科绩点、gpa、排名
4. 统计课程信息。输入课程名，系统显示该门课程的学分、最高分、最低分、平均分、及格率
5. 统计所有学生的绩点。系统将按gpa排名输出所有学生的绩点
6. 删除学生信息。系统显示所有学生的成绩，用户选择要删除的信息序号。注：删除意味着该名学生的信息全部删除
7. 按7离开系统，新数据会自动保存至“测试数据.txt”中。（强制关闭不会保存）

**附录4：评分表**

第1题评分标准

|  |  |  |
| --- | --- | --- |
| **项 目** | **评 价** | |
| 设计方案的合理性与创新性 | **3** |  |
| 设计与调试结果 | **4** |  |
| 设计说明书的质量 | **1** |  |
| 程序基本要求涵盖情况 | **4** |  |
| 程序代码编写素养情况 | **2** |  |
| 课程设计周表现情况 | **1** |  |
| 综合成绩 | **15** |  |