

**课程设计报告**

**题目**

|  |  |
| --- | --- |
| 学 院 | 计算机与通信工程学院 |
| 专 业 | 计算机类 |
| 班级序号 | 180128 |
| 学 号 | **20188117** |
| 姓 名 | 项溢馨 |
| 指导教师 | 柏禄一 |
| 验收日期 | 2019年 月 日 |

核心知识点清单，由学生确认

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数组 | 链表 | 指针 | 文件读写 | 默认参数 | 函数模板 | 多文件 | 类 | 派生 | 虚函数 | 友元函数 | 重载 | 多继承 |
| √ |  | √ | √ |  |  | √ | √ |  |  | √ |  |  |

以下为教师评分表，学生不可填写

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序质量  （60%） | 课程设计报告（20%） | 答辩效果  （20%） | 总分 | 等级 |
|  |  |  |  |  |

评分标准与说明：

1. 程序质量（60%）包含程序正确性与所用知识点数量（40%），代码可读性（10%）与界面友好性（10%）。
2. 课程设计报告（20%）要求排版规范，模块设计有文字说明，图、表、代码清单要有序号和名字。
3. 现场答辩（20%）要求根据学生制作的PPT、讲述清晰、回答问题等情况综合评分。
4. 收齐所有纸质报告的同时，要求学委收集所有学生的代码工程、报告电子版和答辩PPT以备存档。

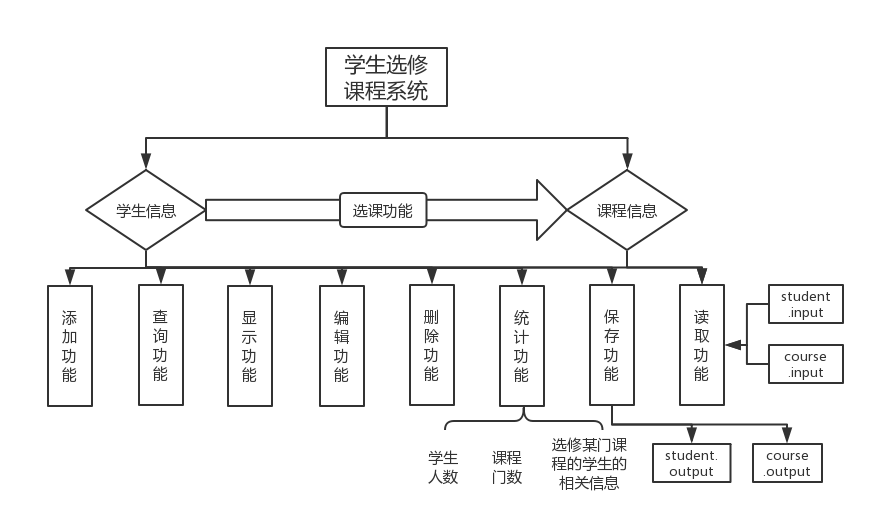
学生选修课程系统设计目录

1. 知识点清单 3
2. 课程设计实现模块结构图、模块的实现流程图、工程组织结构图 4
3. 运行结果截图 5
4. 程序精华 9
5. 创意的技术实现 9
6. 目前存在的问题 10
7. 收获和体会 11
8. 参考文献 12
9. 源代码清单 13

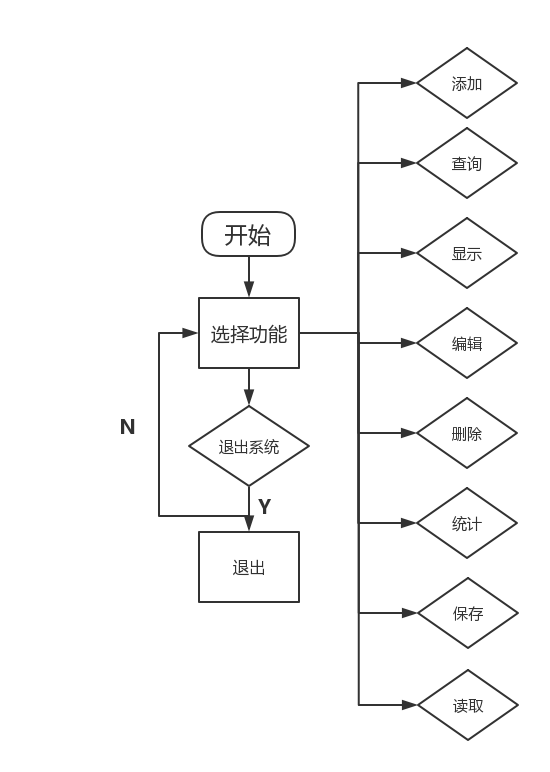
**知识点清单**

1. 数组
2. 指针
3. 类
4. 文件读写
5. 多文件编写
6. 友元机制
7. Vector及STL
8. 构造函数

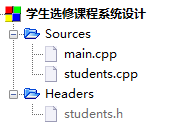
**课程设计实现模块结构图**

****

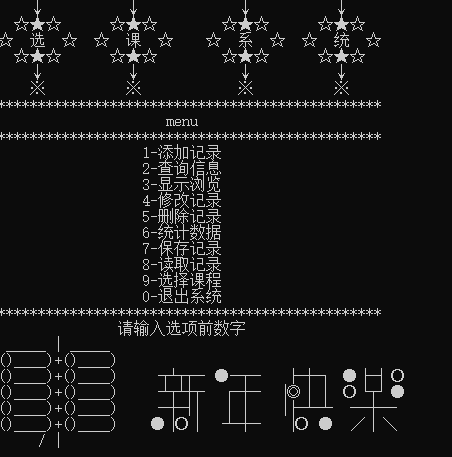
**模块的实现流程图**

****

**工程组织结构图**



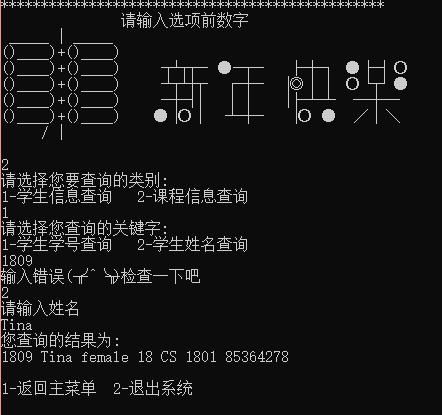
**运行结果截图**



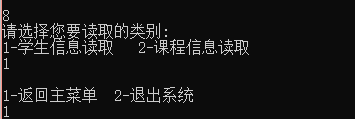
**1.用户界面主菜单**

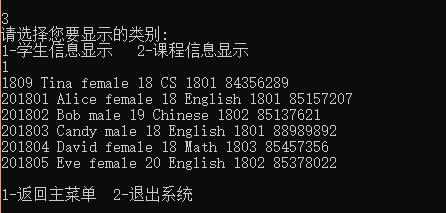


**2.学生信息录入**

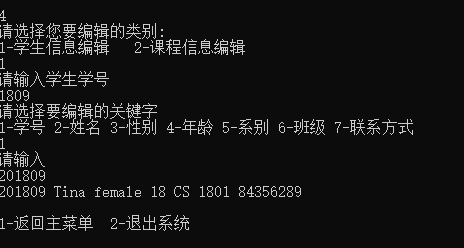


**3.学生信息查询**

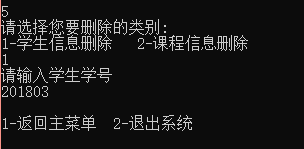


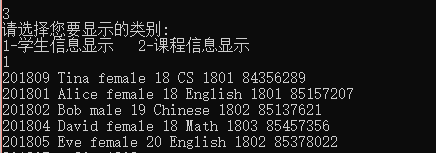


**4.读取记录以及信息显示**

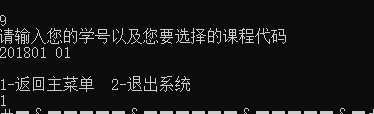


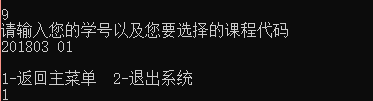
**5.信息修改**

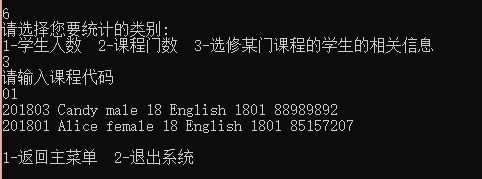




**6.删除记录**







**7.选课系统及信息统计**



**8.退出系统**

**程序精华**

1. **良好的用户界面及操作体验**

以用户数字输入的方式完成功能选择，并及时对每一步操作后是否返回主菜单进行询问，保证了操作的连贯性。而且随时可以退出选课系统。

考虑到选课系统作为学校的门户核心网站之一，对用户界面进行美化，更体现其面向对象编程的特点。

1. **友元机制下的选课系统学生信息与课程信息交互实现**

我认为这个系统的重难点在于如何通过选课功能把学生和课程两大类进行绑定，从而可以在统计功能下实现输入课程代码就可以显示选择这门课的学生的信息。

通过友元类，我解决了这个问题。使课程类成为学生类的友元类，从而做到可以通过课程类访问学生类中的隐私数据。把选课功能里的学生学号记录在课程类中，统计时可根据此学号直接访问学生类，从而输出相关信息。

**创意的技术实现**

1. **Vector在类中的应用**

在选课功能下如何解决学号和课程的匹配记录是关键的一步。学生和课程是多对多的关系，这就注定了无法采取传统的数据类型进行记录，因此可以用自动增长的Vector以及STL库来实现一些基础操作。

在课程类中定义一个int类型的Vector，每次输入学号和课程编号时找到对应课程，把学号存储在类的Vector中，调用STL库里的insert前插数据，大小自动增长。

1. **数据读写**

类的数据读写比传统变量的数据读写要复杂一些，怎样才能尽量简化过程呢？我的办法是先用传统变量的数据读写，通过类的构造函数传入，从而实现类的数据读写。

并通过对fstream和ofstream的文件流运用，可以判断文件是否正常打开关闭。

**目前存在的问题**

1. **只能模拟小规模的数据库功能**

因为采用了数组作为数据存储的载体，无疑会导致存储空间的浪费和固定性。大规模的读入数据恐无法实现。

1. **查找未运用算法优化，大数据轰炸下会导致系统瘫痪**

根据学号搜索对应记录并进行更改的基础算法未通过二分等优化，log（n）的复杂度在数据量极大下可能会产生运行缓慢的问题。

1. **代码利用率不高，重复代码或可编写自定义函数**

因为不太习惯用自定义函数，所以程序中很多代码重复率比较高。通过自定义函数可以节约用户提示的输出代码量，增加代码可读性。

**收获和体会**

写这个系统用时为一天。之前是比较排斥写工程的，更倾向于去研究算法。一天之内从无到有，能学以致用，我认为就是这个工程的意义所在。

拿到这个题目我首先考虑的是程序结构和用户界面。题目已经明显提示了两大类：学生信息和课程信息，思路清晰，各个需要实现的函数也基本给出，框架很容易构建。从VB入门的我深谙用户界面的重要性，这也是面向对象编程区别于面向过程编程的地方。除了对界面进行美化装饰，考虑到用户的体验效果，用户每给出一步操作，系统都会给出相应提示语，引导用户行为。

总体系统的结构基于选择结构。通过用户输入的数字进行操作。主菜单采用switch语句，节省了代码量，增加可读性。各分支总体也是基于三大基本结构，降低了对算法的要求。

调试过程中最先开始出现的问题就是输出中文乱码导致程序异常结束。这个问题我查阅了网上很多的资料，得出比较可靠的结论就是控制台显示的时候缺省的是使用系统默认的字符集，比如windows下用的是GBK，但是默认情况下，编译之后的执行文件时编译成UTF-8的，所以又出现了不统一，乱码由此而生。要解决这个问题要通过对CB编译器的调整，修改编译器对源文件解释编码格式和生成执行文件执行时候采用的编码格式。

第二个思考了一段时间的问题是选课功能怎么能把student类的私有数据共享给course类。因为stu数组的ID等属性都是student私有的，如果在course类中输出会产生问题。解决的方法其实还是挺多的：使用局部变量和全局变量共享数据，通过类的数据成员共享数据，通过类的静态成员实现数据共享，通过友元机制共享数据，指针成员变量与计数。考虑到友元机制实现起来比较简单，因此就把course类作为student类的友元类，从而实现了数据共享。

数据的存取是卡了好久的关键点。之前只实现过普通变量的数据读写，对于类的数据读写有些不知所措。但是如果找寻联系的话，都是可以运用fstream和ofstream流传输的方式，只需要再次通过类的构造函数传入类中就可以了。想明白了这点，输入输出都变得轻而易举了。

写工程最重要的一点就是框架。只要把框架搭起来，填充函数就相对容易了。可以说良好的开端是成功的一半。而且每完成一个功能最好就进行调试，及时发现bug填补上。这样也是为后面的功能扫清障碍。

**参考文献**

1.百度文库《学生选修课程系统设计 源程序清单》

2.CSDN《C++类对象共享数据的5种实现方法》

3.CSDN《Code：Blocks 中文乱码问题原因分析和解决方法》

4.CSDN《fstream读写文件注意问题》

5.博客园《C++中vector使用详细说明》

6.CSDN《在类里定义动态数组》

**源代码清单**

**main.cpp**

#include <bits/stdc++.h>

#include "students.h"

using namespace std;

void goout();

student stu[1000];

course cou[1000];

int len1=0,len2=0;

int main()

{

cout<<"＃〓§〓〓〓〓〓§〓〓〓〓〓〓§〓〓〓〓〓§〓＃"<<endl;

cout<<"　　↓　　　　　↓　　　　　　↓　　　　　↓ "<<endl;

cout<<" ☆★☆　　　☆★☆　　　　☆★☆　　　☆★☆ "<<endl;

cout<<"☆　选　☆　☆　课　☆　　☆　系　☆　☆　统　☆ "<<endl;

cout<<"　☆★☆　　　☆★☆　　　　☆★☆　　　☆★☆ "<<endl;

cout<<"　　↓　　　　　↓　　　　　　↓　　　　　↓ "<<endl;

cout<<"　　※　　　 　 ※　　　　　　※　　　　　※ "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" menu "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 1-添加记录 "<<endl;

cout<<" 2-查询信息 "<<endl;

cout<<" 3-显示浏览 "<<endl;

cout<<" 4-修改记录 "<<endl;

cout<<" 5-删除记录 "<<endl;

cout<<" 6-统计数据 "<<endl;

cout<<" 7-保存记录 "<<endl;

cout<<" 8-读取记录 "<<endl;

cout<<" 9-选择课程 "<<endl;

cout<<" 0-退出系统 "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 请输入选项前数字 "<<endl;

cout<<" \_\_\_\_\_ | \_\_\_\_\_ "<<endl;

cout<<"()\_\_\_\_)+()\_\_\_\_) "<<endl;

cout<<"()\_\_\_\_)+()\_\_\_\_) ┬┴┬┌─　●─┬─　　│─┼─┐　●├─┤ ○"<<endl;

cout<<"()\_\_\_\_)+()\_\_\_\_) ┴┬┴├┬　 ┌─┼─　 │◎│ │　○└┬┘ ●"<<endl;

cout<<"()\_\_\_\_)+()\_\_\_\_) ─┼─││　 │ │　　││─┴─┴　 ──┼──"<<endl;

cout<<"()\_\_\_\_)+()\_\_\_\_) ●│○│　 ┴─┼─　　│○ ● ／│ ＼"<<endl;

cout<<" / | \ "<<endl;

cout<<endl;

int n;

int m,m1,nID,num,q,f;

string name,s;

cin>>n;

while(n>9)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>n;

}

switch(n)

{

case 0:

goout();

break;

case 1:

cout<<"请选择您要添加的类别:"<<endl;

cout<<"1-学生信息录入 2-课程信息录入"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

stu[len1++].Set(stu,len1);

else if(m==2)

cou[len2++].Set(cou,len2);

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 2:

cout<<"请选择您要查询的类别:"<<endl;

cout<<"1-学生信息查询 2-课程信息查询"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

cout<<"请选择您查询的关键字:"<<endl;

cout<<"1-学生学号查询 2-学生姓名查询"<<endl;

cin>>m1;

while(m1>2||m1<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m1;

}

if(m1==1)

{

cout<<"请输入学号"<<endl;

cin>>nID;

stu[0].ID\_Find(stu,len1,nID);

}

else if(m1==2)

{

cout<<"请输入姓名"<<endl;

cin>>name;

stu[0].Name\_Find(stu,len1,name);

}

}

else if(m==2)

{

cout<<"请选择您查询的关键字:"<<endl;

cout<<"1-课程代码查询 2-课程名称查询"<<endl;

cin>>m1;

while(m1>2||m1<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m1;

}

if(m1==1)

{

cout<<"请输入代码"<<endl;

cin>>nID;

cou[0].Num\_Find(cou,len2,nID);

}

else if(m1==2)

{

cout<<"请输入名称"<<endl;

cin>>name;

cou[0].Name\_Find(cou,len2,name);

}

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 3:

cout<<"请选择您要显示的类别:"<<endl;

cout<<"1-学生信息显示 2-课程信息显示"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

stu[0].Show(stu,len1);

else if(m==2)

cou[0].Show(cou,len2);

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 4:

cout<<"请选择您要编辑的类别:"<<endl;

cout<<"1-学生信息编辑 2-课程信息编辑"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

cout<<"请输入学生学号"<<endl;

cin>>nID;

cout<<"请选择要编辑的关键字"<<endl;

cout<<"1-学号 2-姓名 3-性别 4-年龄 5-系别 6-班级 7-联系方式"<<endl;

cin>>m1;

cout<<"请输入"<<endl;

cin>>s;

stu[0].Change(stu,len1,nID,m1,s);

}

else if(m==2)

{

cout<<"请输入课程代码"<<endl;

cin>>nID;

cout<<"请选择要编辑的关键字"<<endl;

cout<<"1-课程代码 2-课程名称 3-课程性质 4-总学时 5-学分 6-开课学期 7-选修人数"<<endl;

cin>>m1;

cout<<"请输入"<<endl;

cin>>s;

cou[0].Change(cou,len2,nID,m1,s);

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 5:

cout<<"请选择您要删除的类别:"<<endl;

cout<<"1-学生信息删除 2-课程信息删除"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

cout<<"请输入学生学号"<<endl;

cin>>nID;

stu[0].Delate(stu,len1,nID);

len1--;

}

else if(m==2)

{

cout<<"请输入课程代码"<<endl;

cin>>nID;

cou[0].Delate(cou,len2,nID);

len2--;

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 6:

cout<<"请选择您要统计的类别:"<<endl;

cout<<"1-学生人数 2-课程门数 3-选修某门课程的学生的相关信息"<<endl;

cin>>m;

while(m>3||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

cout<<"学生的总人数为 "<<len1<<endl;

}

else if(m==2)

{

cout<<"课程的总门数为 "<<len2<<endl;

}

else if(m==3)

{

cout<<"请输入课程代码"<<endl;

cin>>nID;

cou[0].Statistics(cou,len2,nID,stu,len1);

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 7:

cout<<"请选择您要保存的类别:"<<endl;

cout<<"1-学生信息保存 2-课程信息保存"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

ofstream file ("student.output.txt");

if(file.is\_open())

{

int nID;string name;string gender;int age;string major;int nclass;string tel;

for(int i=0;i<len1-1;i++)

{

stu[i].Output(&nID,&name,&gender,&age,&major,&nclass,&tel);

file<<nID<<" "<<name<<" "<<gender<<" "<<age<<" "<<major<<" "<<nclass<<" "<<tel<<endl;

}

file.close();

}

}

else if(m==2)

{

ofstream file ("course.output.txt");

if(file.is\_open())

{

int num,time,term,people,grades;

string name,nature;

for(int i=0;i<len2-1;i++)

{

cou[i].Output(&num,&name,&nature,&time,&grades,&term,&people);

file<<num<<" "<<name<<" "<<nature<<" "<<time<<" "<<grades<<" "<<term<<" "<<people<<endl;

}

file.close();

}

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 8:

cout<<"请选择您要读取的类别:"<<endl;

cout<<"1-学生信息读取 2-课程信息读取"<<endl;

cin>>m;

while(m>2||m<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>m;

}

if(m==1)

{

fstream file("student.input.txt");

if(file.fail())

{

cout << "error open!" << endl;

}

while(!file.eof())

{

int nID;string name;string gender;int age;string major;int nclass;string tel;

file>>nID>>name>>gender>>age>>major>>nclass>>tel;

stu[len1++].Input(nID,name,gender,age,major,nclass,tel);

}

}

else if(m==2)

{

fstream file1("course.input.txt");

if(file1.fail())

{

cout << "error open!" << endl;

}

while(!file1.eof())

{

int num,time,term,people,grades;

string name,nature;

file1>>num>>name>>nature>>time>>grades>>term>>people;

cou[len2++].Input(num,name,nature,time,grades,term,people);

}

}

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

case 9:

cout<<"请输入您的学号以及您要选择的课程代码"<<endl;

cin>>nID>>num;

cou[0].Choose(cou,len2,nID,num);

cout<<endl<<"1-返回主菜单 2-退出系统"<<endl;

cin>>f;

while(f>2||f<1)

{

cout<<"输入错误(╥╯^╰╥)检查一下吧"<<endl;

cin>>f;

}

if(f==1) main();

else goout();

break;

}

return 0;

}

**students.cpp**

#include<bits/stdc++.h>

#include "students.h"

using namespace std;

void student::Set(student \*a,int len)

{

int nID;string name;string gender;int age;string major;int nclass;string tel;

cin>>nID>>name>>gender>>age>>major>>nclass>>tel;

int x=0;

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

{

if(a[i].ID==nID)

{

x=1;

break;

}

}

if(x==1)

cout<<"数据重复添加"<<endl;

else

{

ID=nID;

Name=name;

Gender=gender;

Age=age;

Major=major;

Class=nclass;

Tel=tel;

}

}

void student::Name\_Find(student \*a,int len,string name)

{

int x=0;

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

{

if(a[i].Name==name)

{

cout<<"您查询的结果为:"<<endl;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

x=1;

break;

}

}

if(x==0)

cout<<"未查询到结果 〃'▽'〃 请检查输入"<<endl;

}

void student::ID\_Find(student \*a,int len,int nID)

{

int x=0;

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

{

if(a[i].ID==nID)

{

cout<<"您查询的结果为:"<<endl;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

x=1;

break;

}

}

if(x==0)

cout<<"未查询到结果 〃'▽'〃 请检查输入"<<endl;

}

void student::Show(student \*a,int len)

{

for(int i=0;i<len-1;i++)

{

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

}

}

void student::Change(student \*a,int len,int m,int q,string s)

{

int num,i,x=0;

stringstream ss;

ss<<s;

ss>>num;

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

{

if(a[j].ID==m)

{

i=j;

break;

}

}

switch(q)

{

case 1:

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

{

if(a[j].ID==num)

{

x=1;

break;

}

}

if(x==1)

cout<<"数据重复添加"<<endl;

else

{

a[i].ID=num;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

}

break;

case 2:

a[i].Name=s;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

case 3:

a[i].Gender=s;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

case 4:

a[i].Age=num;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

case 5:

a[i].Major=s;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

case 6:

a[i].Class=num;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

case 7:

a[i].Tel=s;

cout<<a[i].ID<<" "<<a[i].Name<<" "<<a[i].Gender<<" "<<a[i].Age<<" "<<a[i].Major<<" "<<a[i].Class<<" "<<a[i].Tel<<endl;

break;

default:

cout<<"输入有误"<<endl;

}

}

void student::Delate(student \*a,int len,int m)

{

int n=-1;

for(int i=0;i<len-1;i++)

{

if(a[i].ID==m)

{

n=i;

break;

}

}

if(n<0)

cout<<"输入有误"<<endl;

else

{

for(int i=n;i<len;i++)

{

a[i]=a[i+1];

}

}

}

void student::Output(int \*nID,string \*name,string \*gender,int \*age,string \*major,int \*nclass,string \*tel)

{

\*nID=ID;

\*name=Name;

\*gender=Gender;

\*age=Age;

\*major=Major;

\*nclass=Class;

\*tel=Tel;

}

void student::Input(int nID,string name,string gender,int age,string major,int nclass,string tel)

{

ID=nID;

Name=name;

Gender=gender;

Age=age;

Major=major;

Class=nclass;

Tel=tel;

}

void course::Set(course \*a,int len)

{

int num,time,term,people;

double grades;

string name,nature;

cin>>num>>name>>nature>>time>>grades>>term>>people;

int x=0;

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

{

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

{

x=1;

break;

}

}

if(x==1)

cout<<"数据重复添加"<<endl;

else

{

Num=num;

Name=name;

Nature=nature;

Time=time;

Grades=grades;

Term=term;

People=people;

}

}

void course::Name\_Find(course \*a,int len,string name)

{

int x=0;

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

{

if(a[i].Name==name)

{

cout<<"您查询的结果为:"<<endl;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

x=1;

break;

}

}

if(x==0)

cout<<"未查询到结果 〃'▽'〃 请检查输入"<<endl;

}

void course::Num\_Find(course \*a,int len,int nID)

{

int x=0;

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

{

if(a[i].Num==nID)

{

cout<<"您查询的结果为:"<<endl;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

x=1;

break;

}

}

if(x==0)

cout<<"未查询到结果 〃'▽'〃 请检查输入"<<endl;

}

void course::Show(course \*a,int len)

{

for(int i=0;i<len-1;i++)

{

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

}

}

void course::Change(course \*a,int len,int m,int q,string s)

{

int num,i,x=0;

stringstream ss;

ss<<s;

ss>>num;

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

{

if(a[j].Num==m)

{

i=j;

break;

}

}

switch(q)

{

case 1:

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

{

if(a[j].Num==num)

{

x=1;

break;

}

}

if(x==1)

cout<<"数据重复添加"<<endl;

else

{

a[i].Num=num;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

}

break;

case 2:

a[i].Name=s;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

case 3:

a[i].Nature=s;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

case 4:

a[i].Time=num;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

case 5:

a[i].Grades=num;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

case 6:

a[i].Term=num;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

case 7:

a[i].People=num;

cout<<a[i].Num<<" "<<a[i].Name<<" "<<a[i].Nature<<" "<<a[i].Time<<" "<<a[i].Grades<<" "<<a[i].Term<<" "<<a[i].People<<endl;

break;

default:

cout<<"输入有误"<<endl;

}

}

void course::Delate(course \*a,int len,int m)

{

int n=-1;

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

{

if(a[i].Num==m)

{

n=i;

break;

}

}

if(n<0)

cout<<"输入有误"<<endl;

else

{

for(int i=n;i<len-1;i++)

{

a[i]=a[i+1];

}

}

}

void course::Choose(course \*a,int len,int stuID,int num)

{

int n=-1;

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

{

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

{

n=i;

break;

}

}

if(n<0)

cout<<"输入有误"<<endl;

else

{

a[n].ID.insert(ID.begin(),stuID);

}

}

void course::Statistics(course \*a,int len,int nID,student \*b,int len2)

{

int n=-1;

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

{

if(a[i].Num==nID)

{

n=i;

break;

}

}

if(n<0)

cout<<"输入有误"<<endl;

else

{

for(int i=0;i<a[n].ID.size();i++)

{

for(int j=0;j<len2;j++)

{

if(b[j].ID==a[n].ID[i])

{

cout<<b[j].ID<<" "<<b[j].Name<<" "<<b[j].Gender<<" "<<b[j].Age<<" "<<b[j].Major<<" "<<b[j].Class<<" "<<b[j].Tel<<endl;

break;

}

}

}

}

}

void course::Output(int \*num,string \*name,string \*nature,int \*time,int \*grades,int \*term,int \*people)

{

\*num=Num;

\*name=Name;

\*nature=Nature;

\*time=Time;

\*grades=Grades;

\*term=Term;

\*people=People;

}

void course::Input(int num,string name,string nature,int time,int grades,int term,int people)

{

Num=num;

Name=name;

Nature=nature;

Time=time;

Grades=grades;

Term=term;

People=people;

}

void goout()

{

cout<<" quu..\_\_ "<<endl;

cout<<" $$$b `---.\_\_ "<<endl;

cout<<" \"$$b `--. \_\_\_.---uuudP "<<endl;

cout<<" `$$b `.\_\_.------.\_\_ \_\_.---' $$$$\" . "<<endl;

cout<<" \"$b -' `-.-' $$$\" .'| "<<endl;

cout<<" \". d$\" \_.' | "<<endl;

cout<<" `. / ...\" .' | "<<endl;

cout<<" `./ ..::-' \_.' | "<<endl;

cout<<" / .:::-' .-' .' "<<endl;

cout<<" : ::''\ \_.' | "<<endl;

cout<<" .' .-. .-. `. .' | "<<endl;

cout<<" : /'$$| .@\"$\ `. .' \_.-' "<<endl;

cout<<" .'|$u$$| |$$,$$| | < \_.-' "<<endl;

cout<<" | `:$$:' :$$$$$: `. `. .-' "<<endl;

cout<<" : `\"--' | `-. \ "<<endl;

cout<<" :##. == .###. `. `. `\ "<<endl;

cout<<" |##: :###: | > > "<<endl;

cout<<" |#' `..'`..' `###' x: / / "<<endl;

cout<<" \ xXX| / ./ "<<endl;

cout<<" \ xXXX'| / ./ "<<endl;

cout<<" /`-. `. / / "<<endl;

cout<<" : `- ..........., | / .' "<<endl;

cout<<" | ``:::::::' . |< `. "<<endl;

cout<<" | ``` | x| \ `.:``. "<<endl;

cout<<" | .' /' xXX| `:`M`M':. "<<endl;

cout<<" | | ; /:' xXXX'| -'MMMMM:' "<<endl;

cout<<" `. .' : /:' |-'MMMM.-' "<<endl;

cout<<" | | .' /' .'MMM.-' "<<endl;

cout<<" `'`' : ,' |MMM< "<<endl;

cout<<" | `' |tbap\ "<<endl;

cout<<" \ :MM.-' "<<endl;

cout<<" \ | .'' "<<endl;

cout<<" \. `. / "<<endl;

cout<<" / .:::::::.. : / "<<endl;

cout<<" | .:::::::::::`. / "<<endl;

cout<<" | .:::------------\ / "<<endl;

cout<<" / .'' >::' / "<<endl;

cout<<" `',: : .' "<<endl;

cout<<" `:.:'\*/ "<<endl;

cout<<" 您已成功退出系统 "<<endl;

}

**students.h**

#include<bits/stdc++.h>

using namespace std;

class student

{

private:

int ID,Age,Class;

string Name,Gender,Major,Tel;

public:

void Set(student \*a,int len);

void Name\_Find(student \*a,int len,string name);

void ID\_Find(student \*a,int len,int nID);

void Show(student \*a,int len);

void Change(student \*a,int len,int m,int q,string s);

void Delate(student \*a,int len,int m);

void Output(int \*nID,string \*name,string \*gender,int \*age,string \*major,int \*nclass,string \*tel);

void Input(int nID,string name,string gender,int age,string major,int nclass,string tel);

friend class course;

};

class course

{

private:

int Num,Time,Term,People,Grades;

string Name,Nature;

vector<int> ID;

public:

void Set(course \*a,int len);

void Name\_Find(course \*a,int len,string name);

void Num\_Find(course \*a,int len,int num);

void Show(course \*a,int len);

void Change(course \*a,int len,int m,int q,string s);

void Delate(course \*a,int len,int m);

void Choose(course \*a,int len,int stuID,int num);

void Statistics(course \*a,int len,int nID,student \*b,int len2);

void Output(int \*num,string \*name,string \*nature,int \*time,int \*grades,int \*term,int \*people);

void Input(int num,string name,string nature,int time,int grades,int term,int people);

};