面向对象程序设计

课程设计

设计题目：学生考勤管理系统

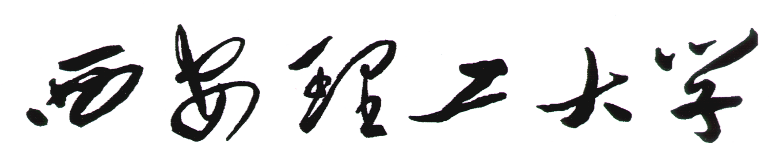
专业 人工智能

班级 人智201

学生 李星宇

学号 3200211013

指导教师 刘光明



2021年秋季学期

# 设计任务

设计一个考勤管理系统，用来描述考勤信息，该信息记录了学生的缺课情况，它包括：缺课日期、第几节课、课程名称、学生姓名、缺课类型（迟到、早退、请假及旷课）。同时，该系统要求实现以下五个功能：（1）录入学生缺课记录；

（2）修改某个学生的缺课记录；

（3）查询某个某个学生的缺课情况；

（4）统计某段时间内，旷课学生姓名及旷课次数，按旷课次数由多到少排序；

（5）统计某段时间内，有学生旷课的课程及旷课人次，按旷课人次由多到少排序；

# 评语(教师填写)

目录

[**第一章 系统需求分析** 1](#_Toc91842940)

[1.1 基本要求 1](#_Toc91842941)

[**第二章 总体设计** 3](#_Toc91842942)

[2.1设计步骤与功能模块图 3](#_Toc91842943)

[**第三章 详细设计** 4](#_Toc91842944)

[3.1类层次图 4](#_Toc91842945)

[3.2 各功能模块的实现 4](#_Toc91842946)

[**第四章 系统实现** 6](#_Toc91842947)

[4.1系统功能展示 6](#_Toc91842948)

[**第五章 系统测试** 9](#_Toc91842949)

[5.1测试用例及运行截图 9](#_Toc91842950)

[**第六章 总结** 10](#_Toc91842951)

[6.1 个人设计小结与心得体会 10](#_Toc91842952)

[**参考文献** 10](#_Toc91842953)

[附录：源程序代码 11](#_Toc91842954)

第一章 系统需求分析

1.1 基本要求

建立考勤管理系统

考勤信息记录了学生的缺课情况，它包括：缺课日期、第几节课、课程名称（课程名称中不会出现空格）、学生姓名、缺课类型（迟到、早退、请假及 旷课）。试设计考勤管理系统，使之能提供以下功能：

1)．录入学生的缺课记录：从键盘输入数据，并将数据存储在文件中以便以后查阅，输入格式为：缺课日期 第几节课 课程名称 学生姓名 缺课类型 每行一条纪录。

例如：

2002.5.5 7 数学 李逍遥 迟到

2002.4.2 6 英语 李四 旷课

2)．修改某个学生的缺课记录：可以对缺课纪录的任意成员进行修改，并对文件中的记录修改。

3)．查询某个学生的缺课情况：将该学生的缺课情况显示到屏幕上。

4)．统计某段时间内，旷课学生姓名及旷课节数，按旷课次数由多到少排序。

输入某段时间的格式例如：2002.4.2 2002.6.3

5)．统计某段时间内，有学生旷课的课程及旷课人次，按旷课人次由多到少排序。

输入某段时间的格式例如：2002.4.2 2002.6.3

6)．系统以菜单方式工作。（所谓菜单指用户可以自由选择所要执行的功能。学生可以通过以上功能录入信息、修改信息、查询信息、整理统计出所要了解的信息。）

7).退出该系统

第二章 总体设计

2.1设计步骤与功能模块图

学生考勤管理系统包含五个大的功能，分别是：录入学生缺课信息、修改学生缺课信息、查询 某个学生的缺课情况、统计某段时间内，旷课学生姓名及旷课节数和统计某段时间内， 有学生旷课的课程及旷课人次。学生的缺课信息主要包含缺课日期，第几节课、课程名称，学生姓名，缺课类型。

在录入学生缺课信息时根据系统提示逐一输入。

在修改学生缺课信息时，用户首先输入要修改的学生姓名，系统会检索如果系统中有该学生的相关信息则系统首先会提示用户修改该学生的那一部分信息，用户可自行选择。如果系统中没有该学生的相关信息，则系统会给相关提示，用户可以重新进行选择。

在查询学生缺课信息时，用户先输入要查询的学生姓名，系统检索判断是否存在该学生的信息然后做出相应的提示，若存在该学生则会在屏幕显示该学生缺课记录。

在统计旷课学生和旷课次数时，用户先输入要统计的起始日期和截止日期，系统判断学生的缺课日期是否在该日期之中，如果在该日期中且缺课类型为旷课，则对应的学生的旷课次数加一。

在统计旷课课程和旷课人次时，用户先输入要统计的起始日期和截止日期，系统判断学生的缺课日期是否在该日期之中，如果在该日期中且缺课类型为旷课，则对应的旷课课程的旷课人次加一。

学生考勤管理系统中功能模块图：

**学生考勤管理系统**

**统计某段时间内，有学生旷课的课程及旷课人数**

**统计某时间内旷课学生姓名及旷课次数**

**查询某学生的缺课信息**

**修改某学生的缺课信息**

**录入学生的缺课信息**

**查询某学生的缺课信息**

**修改某学生的缺课日期、缺课节次、缺课课程名称、缺课类型**

**录入缺课学生的姓名、缺课日期、缺课节次、缺课课程名称、缺课类型**

**按旷课课程的旷课人数由多到少排序**

**按旷课次数由多到少排序**

第三章 详细设计

3.1类层次图

学生考勤管理系统中类的类层次图为：

基类

Absence类

string date //日期

int whatclass //第几节课

string couse //课程

string name //姓名

string type //缺课类型

用于实现sort2()

CLASS类

Absence abs[100]

void Modifyrecord(string Name)

void Search(string Name)

void Sort1()

void Sort2()

SC类

用于实现sort1()

string couse //课程

int num //缺课人数

int flag //标志：1代表前面已出现该课程，0表示前面未出现该课程

SC() //构造函数，用于初始化

派生类

int num //学生缺课的次数

int flag

//标志：1代表前面已出现该学生，0表示前面未出现该学生

Statistics1类

3.2 各功能模块的实现

1、 学生缺课信息录用功能模块:

**学生缺课信息录用功能**

**录入学生缺课类型**

**录入缺课学生姓名**

**录入学生缺课课程名称**

**录入学生缺课节次**

**录入学生缺课日期**

2、 修改某学生缺课信息功能的模块：

**修改某学生缺课信息功能**

**修改某学生的缺课日期**

**修改某学生的缺课节次**

**修改某学生的缺课课程**

**修改某学生的缺课类型**

3、查询学生缺课信息功能模块：

**查询某学生缺课信息功能**

**输入要查询的学生姓名**

**显示要查询的学生缺课信息**

4、 统计某段时间内旷课学生姓名及旷课节数功能模块：

**返回菜单**

**显示某段时间内旷课学生姓名及旷课次数**

**统计某段时间内旷课学生姓名及旷课次数**

**统计某段时间内旷课学生姓名及旷课节数功能**

5、统计某段时间内，有学生旷课的课程及旷课人次功能模块：

**统计某段时间内，有学生旷课的课程及旷课人次功能**

**返回菜单**

**显示某段时间内，有学生旷课的课程及旷课人次**

**统计某段时间内，有学生旷课的课程及旷课人次**

第四章 系统实现

4.1系统功能展示

1. 录入功能：根据系统的提示，依次输入缺课日期，第几节课、课程名称，学生姓名等信息。部分代码如下：

cout << "请分别输入学生的缺课情况，它包括：缺课日期、第几节课、课程名称、学生姓名、缺课类型（迟到、早退、请假及旷课）" << endl;

cin >> date >> whatclass >> couse >> name >> type;

//打开文件并写入学生缺课信息

ofstream out(file,ios::app);

if(out)

{

if (FLAG == 0)

{

out << right<<setw(20) << "缺课日期" << setw(20) << "第几节课" << setw(20) << "课程名称" << setw(20) << "学生姓名" << setw(25) << "缺课类型" << endl;

FLAG = 1;

}

out <<right<< setw(20) << date << setw(20) << whatclass << setw(21) << couse << setw(27) << name << setw(25) << type << endl;

out.close();

}

(2)查询功能：输入要查询的学生姓名，系统会给出该学生的缺课信息。

部分代码如下：

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

{

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

{ cout << "缺课日期：\t" <<abs[i].date<< endl;

cout << "第几节课：\t" << abs[i].whatclass << endl;

cout << "课程名称：\t" << abs[i].couse<< endl;

cout << "学生姓名：\t" << abs[i].name << endl;

cout << "缺课类型：\t" << abs[i].type<< endl;

break;

}

}

（3）修改功能：输入要修改信息的学生姓名以及你选择的修改选项，系统根据你的选项输出相应的内容。

部分关键代码如下：

cin >> j;

switch (j)

{

case 1:

cout << "请输入你要修改成的缺课日期" << endl;

cin >> n;

abs[i].date = n;

break;

case 2:cout << "请输入你要修改成第几节课" << endl;

cin >> k;

abs[i].whatclass = k; break;

case 3:cout << "请输入你要修改成的课程名称" << endl;

cin >> n;

abs[i].couse = n; break;

case 4:cout << "请输入你要修改成的学生姓名" << endl;

cin >> n;

abs[i].name = n; break;

case 5:cout << "请输入你要修改成的缺课类型（迟到、早退、请假及旷课）" << endl;

cin >> n;

abs[i].date = n; break;

case 6:Flag = false;

}

（4）统计排序功能：分为学生旷课统计与旷课课程统计，统计某段时间内，旷课学生姓名及旷课次数，按旷课次数由多到少排序。该功能部分代码如下：

cin >> n >> k;//输入想要查询的日期范围

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

{//筛选出符合日期范围的记录

if (strcmp(abs[i].date.c\_str(), n.c\_str()) >=0 && strcmp(abs[i].date.c\_str(), k.c\_str()) <= 0)

{

sta1[m++].name = abs[i].name;

}

}

//统计重复的学生，并记录该学生缺课多少次

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

{

if (sta1[j].flag == 0)

{

sta[s++].name = sta1[j].name;

sta1[j].flag = 1;

b = s - 1;

for (d = s; d < m; d++)

{

if (sta1[d].flag ==0)

if (sta[b].name == sta1[d].name)

{

sta1[d].flag = 1; sta[b].num++;

}

}

}

}

//冒泡排序，按照学生旷课次数由多到少进行排序

for (i = 0; i < s - 1; i++)

for (j = 0; j < s - i - 1; j++)

{

if (sta[j].num < sta[j + 1].num)

Swap(sta[j], sta[j + 1]);

}

//打印到屏幕上

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

cout << "学生姓名:" <<setw(6)<< sta[i].name <<setw(16)<< "旷课次数： " << sta[i].num << endl;

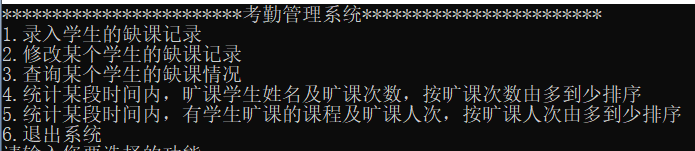
}

旷课课程统计与学生旷课统计代码类似，具体代码请看附录：源程序代码。

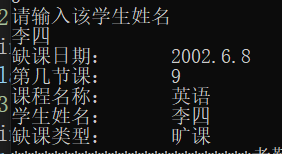
第五章 系统测试

5.1测试用例及运行截图

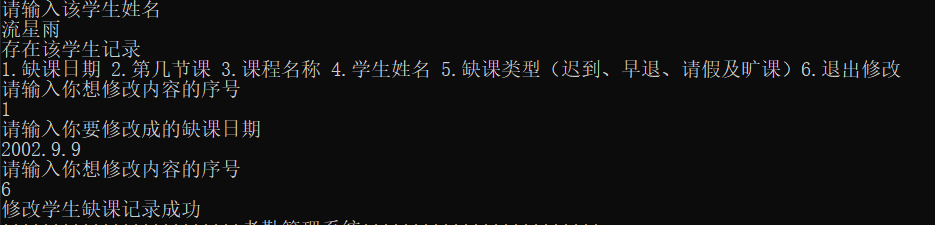
1. 菜单及录入功能运行截图：



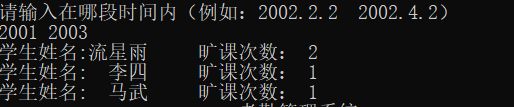
（2）查询功能运行截图：



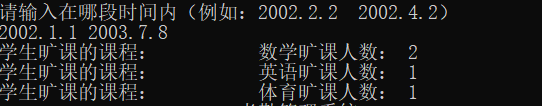
（3）修改功能运行截图：



（4）统计旷课学生功能运行截图：



（5）统计旷课课程功能运行截图：



第六章 总结

6.1 个人设计小结与心得体会

在经历了本次课设之后，我收获良多，我的不足也越发暴露出来。同时通过网上查资料和练习学到了很多以前不懂的东西，这种进步是肉眼可见的，也激发了我编程的决心。

首先便是熟练度的问题，我编代码的时间明显要长于其他人，不止是因为手速太慢，更是因为思维不开阔，由于练习太少所以很多简单的方法我都想不到，而是花费更长的时间想到了更复杂的方法，以至于我的速度大大降低。其次便是知识面的问题，由于我c语言未学习完成，因此对于文件的处理还是很生疏，在转专业班c++课程关于文件教的也很浅，于是造成了严重的阻碍。在文件的读写花费六个小时却bug频出，最终通过学习也只是学会了简单的读和写，但是对于文件内容的修改仍不是很清楚，我也因此发现了我的问题，课下我也会积极学习，查漏补缺。最后就是我不会进行调试，发生错误时只能一行行往下看，看不懂时只能通过百度和询问同学，调试我认为很重要，也是我今后要重点攻克的难题。

本次我的课设题目算是所有任务里面简单的了，但是我写的并不是很尽人意，当然挫折同样是动力，这只是我编程生涯的开始，今后我会更加努力。

**参考文献**

**[1] 王小帆，李薇.面向对象程序设计——C++. 北京：电子工业出版社，2020.7**

附录：

//源程序代码

#include<iostream>

#include<string>

#include<iomanip>

#include<fstream>

using namespace std;

class Absence

{

public:

string date;

int whatclass;//第几节课

string couse;

string name;

string type;//（迟到、早退、请假及旷课）

void Absrecord();

};

static int top=-1;

class CLASS

{public:Absence abs[100];

int Modifyrecord(string Name);

void Search(string Name);

void Sort1();//统计某段时间内，旷课学生姓名及旷课次数，按旷课次数由多到少排序

void Sort2();//统计某段时间内，有学生旷课的课程及旷课人次，按旷课人次由多到少排序

};

class Statistics1:public Absence

{public:

int num;

int flag;//1代表前面已出现该学生，0表示前面未出现该学生

Statistics1()

{

flag = 0;

num = 1;

}

};

string file = "statistics.txt";

//录入学生的缺课记录

static int FLAG = 0;

void Absence::Absrecord()

{

cout << "请分别输入学生的缺课情况，它包括：缺课日期、第几节课、课程名称、学生姓名、缺课类型（迟到、早退、请假及旷课）" << endl;

cin >> date >> whatclass >> couse >> name >> type;

ofstream out(file,ios::app);

if(out)

{

if (FLAG == 0)

{

out << right<<setw(20) << "缺课日期" << setw(20) << "第几节课" << setw(20) << "课程名称" << setw(20) << "学生姓名" << setw(25) << "缺课类型" << endl;

FLAG = 1;

}

out <<right<< setw(20) << date << setw(20) << whatclass << setw(21) << couse << setw(27) << name << setw(25) << type << endl;

out.close();

}

cout << "录入成功" << endl;

};

//修改某个学生的缺课记录

int CLASS::Modifyrecord(string Name)

{

bool Flag=true;

int i,j,k,l;

string n;

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

{

if (abs[i].name == Name) { cout << "存在该学生记录" << endl; break; }

}

if (i > top) { cout << "无该学生记录" << endl; }

else

{

cout << "1.缺课日期 2.第几节课 3.课程名称 4.学生姓名 5.缺课类型（迟到、早退、请假及旷课）6.退出修改" << endl;

while (Flag)

{

cout << "请输入你想修改内容的序号" << endl;

cin >> j;

switch (j)

{

case 1:

cout << "请输入你要修改成的缺课日期" << endl;

cin >> n;

abs[i].date = n;

break;

case 2:cout << "请输入你要修改成第几节课" << endl;

cin >> k;

abs[i].whatclass = k; break;

case 3:cout << "请输入你要修改成的课程名称" << endl;

cin >> n;

abs[i].couse = n; break;

case 4:cout << "请输入你要修改成的学生姓名" << endl;

cin >> n;

abs[i].name = n; break;

case 5:cout << "请输入你要修改成的缺课类型（迟到、早退、请假及旷课）" << endl;

cin >> n;

abs[i].date = n; break;

case 6:Flag = false;

}

}

cout << "修改学生缺课记录成功" << endl; return 0;

}

}

void CLASS::Search(string Name)

{

bool Flag = true;

int i;

string n;

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

{

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

{ cout << "缺课日期：\t" <<abs[i].date<< endl;

cout << "第几节课：\t" << abs[i].whatclass << endl;

cout << "课程名称：\t" << abs[i].couse<< endl;

cout << "学生姓名：\t" << abs[i].name << endl;

cout << "缺课类型：\t" << abs[i].type<< endl;

break;

}

}

if (i > top) cout << "无该学生记录" << endl;

}

void Swap(Statistics1& s, Statistics1& t)

{

Statistics1 a;

a = s;

s = t;

t = a;

}

void CLASS::Sort1()

{

int i,m=0,j,s=0,d=0,b;

string n, k;

Statistics1 sta1[100],sta[50];

cout << "请输入在哪段时间内（例如：2002.2.2 2002.4.2）" << endl;

cin >> n >> k;

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

{

if (strcmp(abs[i].date.c\_str(), n.c\_str()) >=0 && strcmp(abs[i].date.c\_str(), k.c\_str()) <= 0)

{if(strcmp(abs[i].type.c\_str(),"旷课")==0)

sta1[m++].name = abs[i].name;

}

}

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

{

if (sta1[j].flag == 0)

{

sta[s++].name = sta1[j].name;

sta1[j].flag = 1;

b = s - 1;

for (d = s; d < m; d++)

{

if (sta1[d].flag ==0)

if (sta[b].name == sta1[d].name)

{

sta1[d].flag = 1; sta[b].num++;

}

}

}

}

for (i = 0; i < s - 1; i++)

for (j = 0; j < s - i - 1; j++)

{

if (sta[j].num < sta[j + 1].num)

Swap(sta[j], sta[j + 1]);

}

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

cout << "学生姓名:" <<setw(6)<< sta[i].name <<setw(16)<< "旷课次数： " << sta[i].num << endl;

}

class SC

{public:

string couse;

int num;

int flag;

SC()

{

num = 1; flag = 0;

}

};

void Swap(SC& s, SC& t)

{

SC a;

a = s;

s = t;

t = a;

}

void CLASS::Sort2()

{

SC Couse[10],Couse1[50];

int i, m = 0, j, s = 0, d = 0, b;

string n, k;

cout << "请输入在哪段时间内（例如：2002.2.2 2002.4.2）" << endl;

cin >> n >> k;

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

{

if (strcmp(abs[i].date.c\_str(), n.c\_str()) >= 0 && strcmp(abs[i].date.c\_str(), k.c\_str()) <= 0)

{

if (strcmp(abs[i].type.c\_str(), "旷课") == 0)

Couse1[m++].couse = abs[i].couse;

}

}

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

{

if (Couse1[j].flag == 0)

{

Couse[s++].couse = Couse1[j].couse;

Couse1[j].flag = 1;

b = s - 1;

for (d = s; d < m; d++)

{

if (Couse1[d].flag == 0)

if (Couse[b].couse == Couse1[d].couse)

{

Couse1[d].flag = 1; Couse[b].num++;

}

}

}

}

for (i = 0; i < s - 1; i++)

for (j = 0; j < s - i - 1; j++)

{

if (Couse[j].num < Couse[j + 1].num)

Swap(Couse[j], Couse[j + 1]);

}

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

cout << "学生旷课的课程： " <<setw(13)<< Couse[i].couse<< "旷课人数： " << Couse[i].num << endl;

}

void menu()

{ cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*考勤管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "1.录入学生的缺课记录" << endl;

cout << "2.修改某个学生的缺课记录" << endl;

cout << "3.查询某个学生的缺课情况" << endl;

cout << "4.统计某段时间内，旷课学生姓名及旷课次数，按旷课次数由多到少排序" << endl;

cout << "5.统计某段时间内，有学生旷课的课程及旷课人次，按旷课人次由多到少排序" << endl;

cout <<"6.退出系统" << endl;

}

int main()

{

string Name;

CLASS Class;

int n;

bool flag = true;

while (flag)

{

menu();

cout << "请输入您要选择的功能" << endl;

cin >> n;

switch (n)

{

case 1:Class.abs[++top].Absrecord(); break;

case 2:cout << "请输入该学生姓名" << endl;

cin >> Name;

Class.Modifyrecord(Name); break;

case 3:cout << "请输入该学生姓名" << endl;

cin >> Name;

Class.Search(Name); break;

case 4:Class.Sort1();

break;

case 5:Class.Sort2(); break;

case 6:flag = false; break;

default:cout << "您的输入有误，请重新输入" << endl; break;

}

}

return 0;

}