“程序设计基础”

课程设计报告





设计题目 歌手比赛系统

姓 名 袁焕发

学 号 2019217769

专 业 物联网工程

# （一）需求和规格说明

对一次歌手比赛的成绩进行管理

1、输入每个选手的数据包括编总成绩和平均成绩（去掉最高分）

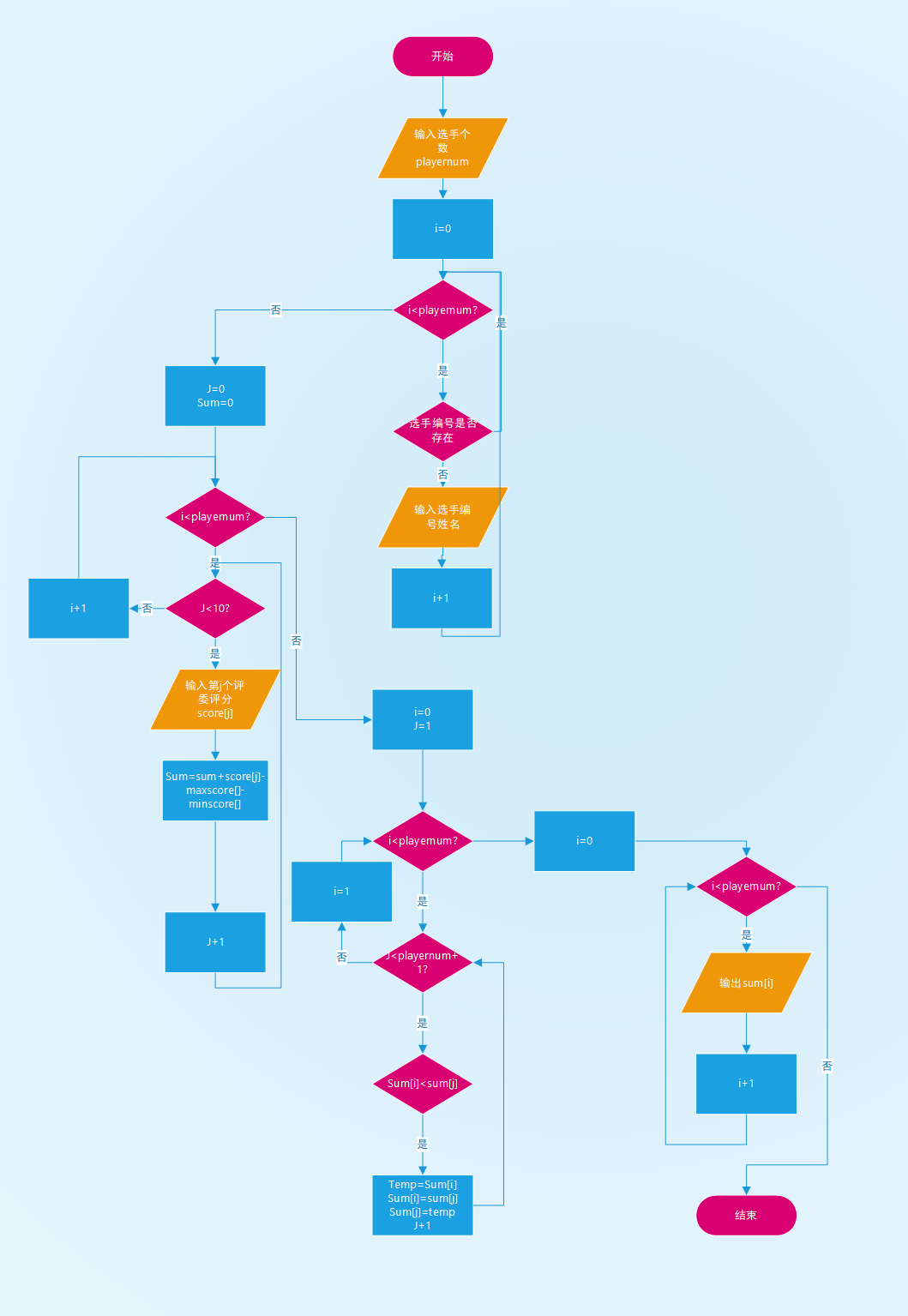
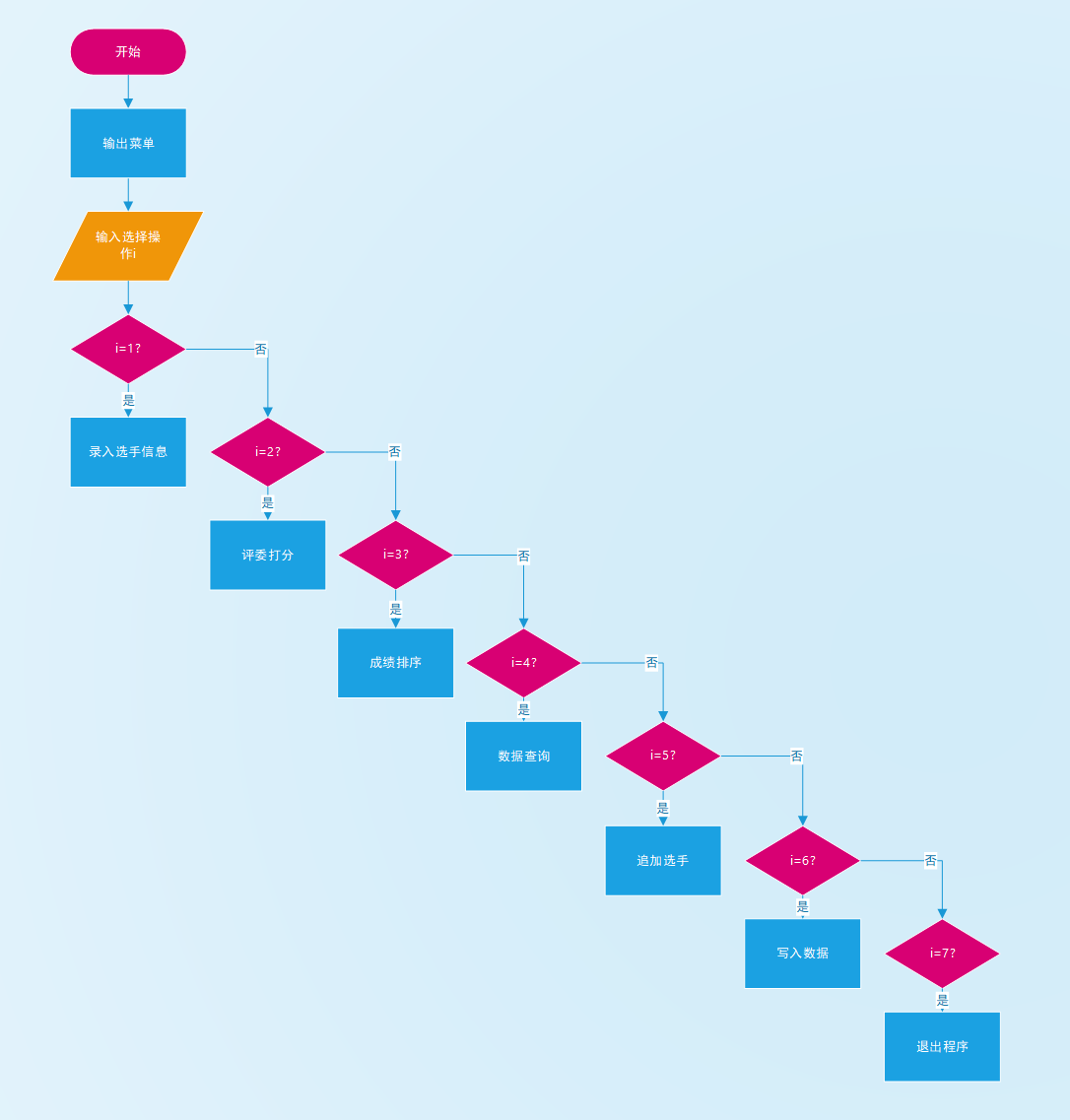
2、显示主菜单如下：1）输入选手数据2）评委打分3）成绩排序（按平均分）4）数据查询5）追加选手数据6）写入数据文件7）退出系统

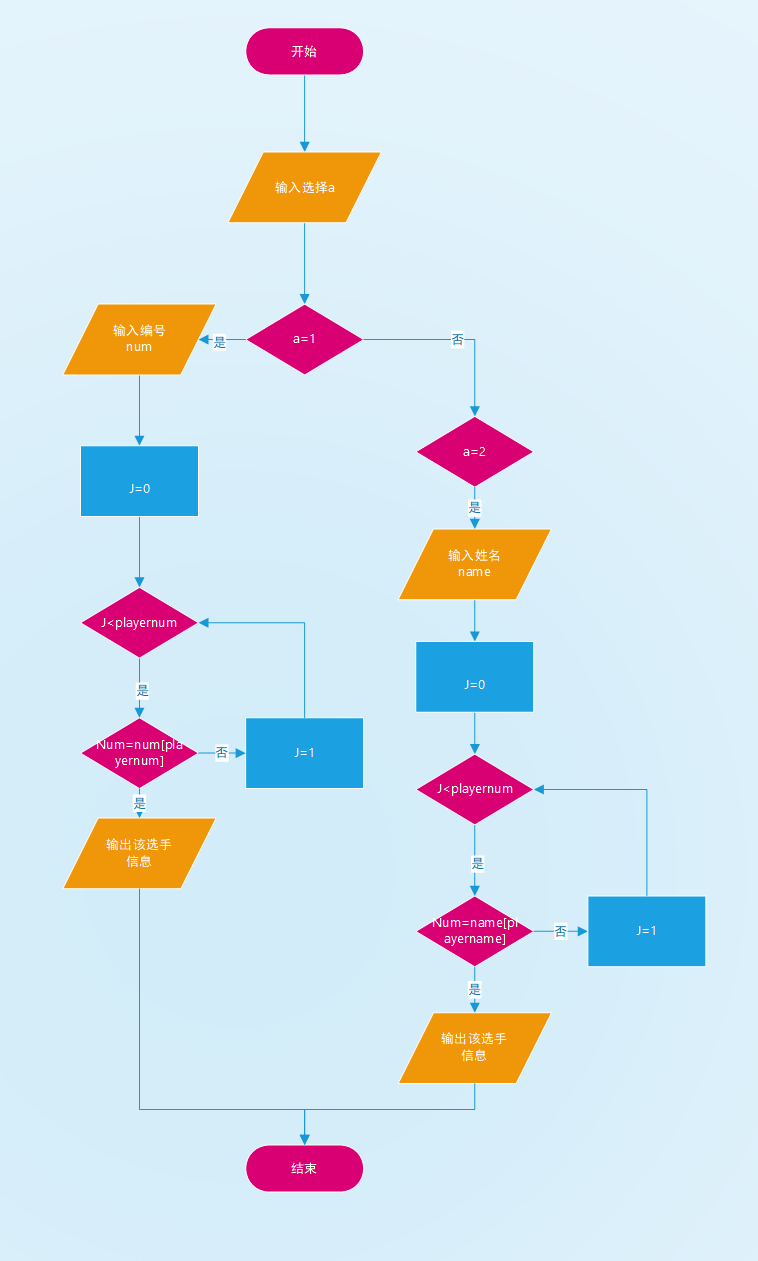
# （二）设计

### (1) 设计思想

构造歌手信息基类包括姓名、编号、分数、总分、平均分等个人信息，构造派生类，在派生类中构造成绩处理函数，对歌手进行录入、打分、排序、查询等操作

### （2）程序框图





|  |  |  |  |
| --- | --- | --- | --- |
| 成员类别 | 类型 | 成员名 | 描述 |
| 属性 | int | PlayerNum | 歌手数目 |
| 属性 | float | Sum | 总分 |
| 属性 | float | Ave | 平均分 |
| 属性 | int | score[] | 分数 |
| 属性 | char | num[] | 歌手编号 |
| 属性 | char | name[] | 歌手姓名 |
| 方法 | int | getMaxScore() | 获得最高分 |
| 方法 | int | getMinScore() | 获得最低分 |
| 方法 | void | Show() | 输出界面 |
| 方法 | void | Choose() | 选择操作 |
| 方法 | void | Input() | 输入选手信息 |
| 方法 | void | GiveScore() | 评委打分 |
| 方法 | void | Sort() | 排序 |
| 方法 | void | Search() | 搜索歌手 |
| 方法 | void | AddPlayer() | 增加歌手 |
| 方法 | void | OutFile() | 输出文件 |
| 方法 |  |  |  |

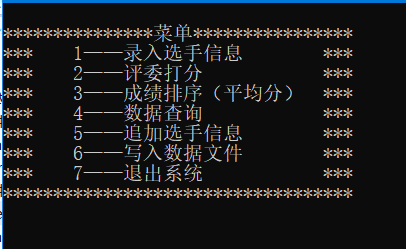
### （三）用户手册

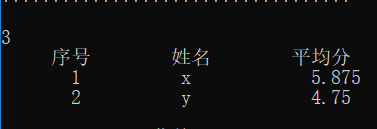
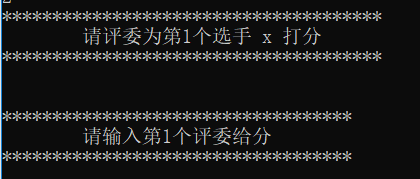
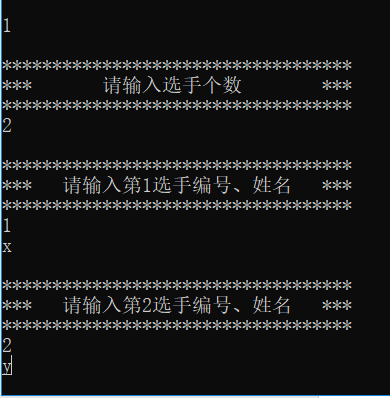
程序运行时现实主菜单，用户按照菜单选择操作，根据提示进一步操作

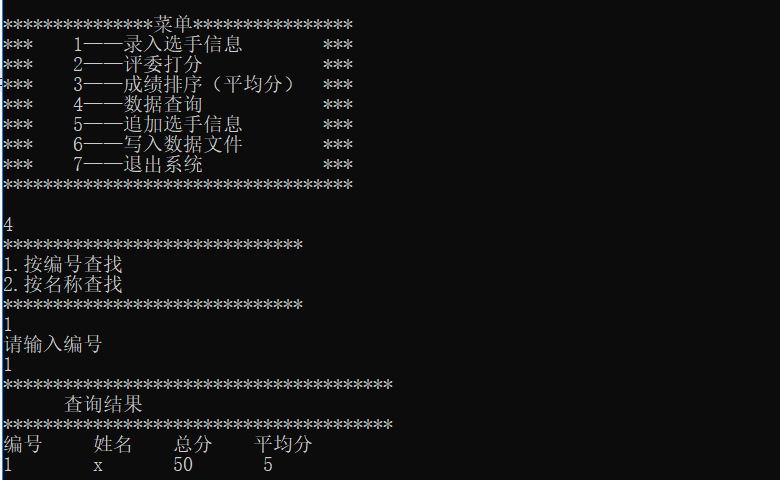
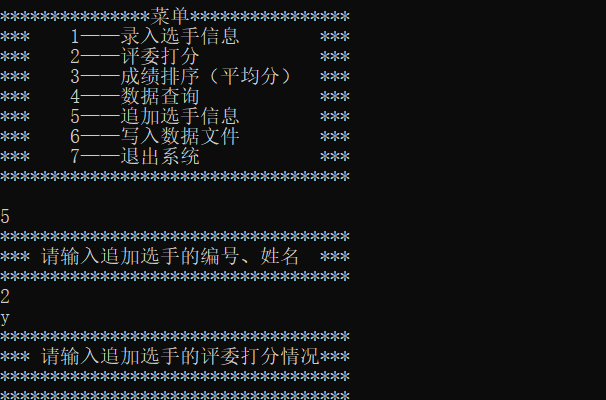
### （四）调试及测试

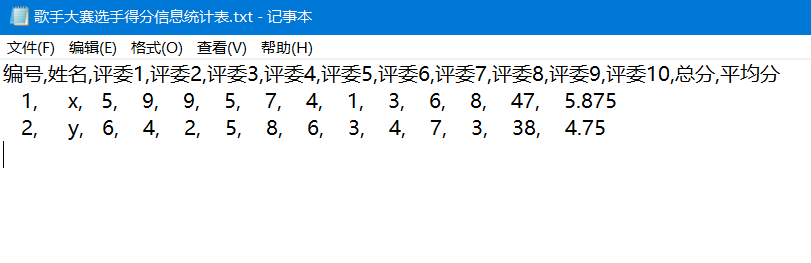
有时程序会出现卡顿和乱码

### （五） 运行实例









### （六）心得体会

进一步熟练基类和派生类的应用，通过派生类调用基类，同时也遇到了一些问题，例如如何查找选手信息，需要用到特定函数，程序虽然不报错但是有时会卡顿乱码，无法准确找到原因.

### （七）对课程设计的建议

锻炼了解决问题的能力，以及对编写代码的熟练度，对于知识点掌握很有帮助

### 附录 源程序

#pragma once

class Player

{

public:

Player();

~Player();

//Player\* P;

int PlayerNum;

char num[20];

char name[20];

int score[10];

int getMaxScore();

int getMinScore();

float sum;

float ave;

};

#include "palyer.h"

#include <iostream>

using namespace *std*;

int Player::getMaxScore()

{

int Maxscore = score[0];

for (int i = 1; i < 10; i++)

{

if (score[i] > Maxscore)

{

Maxscore = score[i];

}

}

return Maxscore;

}

int Player::getMinScore()

{

int Minscore = score[0];

for (int i = 1; i <10; i++)

{

if (score[i] < Minscore)

{

Minscore = score[i];

}

}

return Minscore;

}

Player::Player()

{

for (int i = 0; i < 10; i++)

{

score[i] = 0;

}

sum = 0;

ave = 0;

}

Player::~Player()

{

}

#pragma once

#include "palyer.h"

class Z

{

public:

Z();

~Z();

void Show();

void Choose();

void Input();

void GiveScore();

void Sort();

void Search();

void OutFile();

void AddPlayer();

private:

int PlayerNum;

Player\* P1;

};

#include "Z.h"

#include <iostream>

#include <fstream>

#include <iomanip>

#include <stdlib.h>

#include <string>

using namespace *std*;

int givescore = 0;

void Z::Show()

{

*cout* << *endl*;

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

*cout* << "\*\*\* 1——录入选手信息 \*\*\*" << *endl*;

*cout* << "\*\*\* 2——评委打分 \*\*\*" << *endl*;

*cout* << "\*\*\* 3——成绩排序（平均分） \*\*\*" << *endl*;

*cout* << "\*\*\* 4——数据查询 \*\*\*" << *endl*;

*cout* << "\*\*\* 5——追加选手信息 \*\*\*" << *endl*;

*cout* << "\*\*\* 6——写入数据文件 \*\*\*" << *endl*;

*cout* << "\*\*\* 7——退出系统 \*\*\*" << *endl*;

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

*cout* << *endl*;

}

void Z::Choose()

{

int choice = 1;

while (choice)

{

Show();

*cin* >> choice;

if (choice > 7 || choice < 0) {

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

*cout* << "\*\*\* 选择操作有误，请重新输入 \*\*\*" << *endl*;

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

continue;

}

switch (choice)

{

case 1:

Input();

break;

case 2:

GiveScore();

break;

case 3:

Sort();

break;

case 4:

Search();

break;

case 5:

AddPlayer();

break;

case 6:

OutFile();

break;

}

}

}

void Z::Input()

{

*cout* << *endl*;

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

*cout* << "\*\*\* 请输入选手个数 \*\*\*" << *endl*;

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

*cin* >> PlayerNum;

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

{

*cout* << *endl*;

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

*cout* << "\*\*\* 请输入第" << i + 1 << "选手编号、姓名 \*\*\*" << *endl*;

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

*cin* >> P1[i].num >> P1[i].name;

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

{

if ( i>0&&*strcmp*(P1[i].num, P1[j].num) == 0)

{

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

*cout* << " 已有此编号，请重新输入 " << *endl*;

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

*cin* >> P1[i].num >> P1[i].name;

}

}

}

*cout* << *endl*;

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

*cout* << "\*\*\* 录入完成 \*\*\*" << *endl*;

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

}

void Z::GiveScore()

{

if (PlayerNum == 0||givescore==false)

{

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

*cout* << "\*\*\* 请先录入选手信息 \*\*\*" << *endl*;

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

}

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

{

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

*cout* << " 请评委为第"<<i+1<<"个选手 " << P1[i].name << " 打分 " << *endl*;

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

*cout* << *endl*;

*cout* << *endl*;

for (int j = 0; j < 10; j++)

{

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

*cout* << " 请输入第" << j + 1 << "个评委给分 " << *endl*;

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

*cin* >> P1[i].score[j];

*cout* << *endl*;

P1[i].sum +=P1[i].score[j];

}

P1[i].ave = (P1[i].sum - P1[i].getMaxScore() - P1[i].getMinScore())/ 8;

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

}

givescore = 1;

}

void Z::Sort()

{

//以平均分为关键字进行降序排列

if (PlayerNum == 0)

{

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

*cout* << "\*\*\* 请先录入选手信息! \*\*\*" << *endl*;

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

return ;

}

else if (givescore == 0)

{

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

*cout* << "\*\*\* 请先为选手打分! \*\*\*" << *endl*;

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

}

else

{

Player\* temp = new Player[PlayerNum];

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

{

for (int j = 1; j < PlayerNum + 1; j++)

{

if (P1[i].ave < P1[j].ave)

{

temp[i].ave = P1[i].ave;

P1[i].ave = P1[j].ave;

P1[j].ave = temp[i].ave;

*strcpy\_s*(temp[i].name, 20, P1[i].name);

*strcpy\_s*(P1[i].name, 20, P1[j].name);

*strcpy\_s*(P1[j].name, 20, temp[i].name);

*strcpy\_s*(temp[i].num, 20, P1[i].num);

*strcpy\_s*(P1[i].num, 20, P1[j].num);

*strcpy\_s*(P1[j].num, 20, temp[i].num);

}

}

}

*cout* << " 序号 姓名 平均分" << *endl*;

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

{

*cout* << " " << P1[i].num << " " << P1[i].name << " " << P1[i].ave << *endl*;

}

}

}

void Z::Search()

{

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

*cout* <<"1.按编号查找"<<*endl*

<<"2.按名称查找" << *endl*;

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

int a;

*cin* >> a;

char temp[20];

//Player\* c = new Player[PlayerNum];

if (givescore==0)

{

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

*cout* << "\*\*\* 请先为选手打分! \*\*\*" << *endl*;

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

}

else{

if (a ==1)

{

int flag1 = 0;

*cout* << "请输入编号" << *endl*;

*cin* >>temp;

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

{

if (*strcmp*(temp,P1[i].num) == 0)

{

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

*cout* << " 查询结果 " << *endl*;

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

*cout* << "编号 姓名 总分 平均分" << *endl*;

*cout* << P1[i].num << " " << P1[i].name << " " << P1[i].sum << " " << P1[i].ave << " " << *endl*;

flag1 = 1;

}

}

if (flag1 == 0)

{

*cout* << "查无此人" << *endl*;

}

}

if (a == 2)

{

int flag2 = 0;

*cout* << "请输入姓名" << *endl*;

*cin* >>temp;

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

{

if (*strcmp*(temp, P1[i].name) == 0)

{

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

*cout* << " 查询结果 " << *endl*;

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

*cout* << "编号 姓名 总分 平均分" << *endl*;

*cout* << P1[i].num << " " << P1[i].name << " " << P1[i].sum << " " << P1[i].ave << " " << *endl*;

flag2= 1;

}

}

if (flag2== 0)

{

*cout* << "查无此人" << *endl*;

}

}

}

}

void Z::OutFile()

{

*ofstream* outFile;

if (PlayerNum == 0)

{

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

*cout* << "\*\*\* 数据文件写入失败 \*\*\*" << *endl*;

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

return;

}

outFile.*open*("歌手大赛选手得分信息统计表.txt",*ofstream*::*app*);//创建并打开文件

outFile << "编号" << "," << "姓名" << ",";

for (int i = 0; i <10; i++)

{

outFile << "评委" << i + 1 << ",";

}

outFile << "总分" << "," << "平均分" << *endl*;

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

{

outFile << " " << P1[i].num << ", " << " " << P1[i].name << ", ";

for (int j = 0; j < 10; j++)

{

outFile << P1[i].score[j] << ", ";

}

outFile << P1[i].sum << ", " << P1[i]. ave << *endl*;

}

outFile.*close*();

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

*cout* << "\*\*\* 数据文件写入完成 \*\*\*" << *endl*;

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

}

void Z::AddPlayer()

{

P1->PlayerNum++;//数组扩容

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

*cout* << "\*\*\* 请输入追加选手的编号、姓名 \*\*\*" << *endl*;

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

*cin* >> P1[PlayerNum].num >> P1[PlayerNum].name;

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

{

if (*strcmp*(P1[PlayerNum].num, P1[i].num) == 0)

{

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

*cout* << " 已有此编号，请重新输入 " << *endl*;

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

*cin* >> P1[PlayerNum].num >> P1[PlayerNum].name;

}

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

*cout* << "\*\*\* 请输入追加选手的评委打分情况\*\*\*" << *endl*;

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

}

for (int i = 0; i <10; i++)

{

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

*cout* << " 请输入第" << i + 1 << "个评委给分 " << *endl*;

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

*cin* >> P1[PlayerNum].score[i];

P1[PlayerNum].sum += P1[PlayerNum].score[i];

}

P1[PlayerNum ].sum = P1[PlayerNum ].sum - P1[PlayerNum ].getMaxScore() - P1[PlayerNum ].getMinScore();

P1[PlayerNum ].ave = P1[PlayerNum ].sum /8;

PlayerNum++;

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

*cout* << "\*\*\* 追加成功 \*\*\*" << *endl*;

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

}

Z::Z()

{

PlayerNum =0;

P1 = new Player[PlayerNum];

}

Z::~Z()

{

delete P1;

}

#include "palyer.h"

#include "Z.h"

#include <iostream>

using namespace *std*;

int main()

{

Z singer;

singer.Choose();

return 0;

}