第三次上机实验报告

1. **实验目的**

1、理解类和对象的概念，掌握声明类和定义对象的方法

2、掌握构造函数和析构函数的实现方法

3、初步掌握使用类和对象编制C++程序

1. **实验内容**
2. **实验一**

**创建一个类，完成以下功能：**

连续输入一组二维坐标值；

二维坐标值的数目可以由用户自定义（默认为2个，最多为100组）；

显示用户输入的坐标值；

显示用户输入坐标值的均值；

**观察例程中的构造函数和析构函数的运行顺序；**

**在main()函数中加入如下代码，观察运行结果：**

Coordinate y(5);

y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

1. **实验二**

**创建一个Score类，完成以下功能：**

连续输入多位学生的成绩（成绩=科目A成绩+科目B成绩+科目C成绩）；

学生数目可以由用户自定义（默认为2个，最多为100个）；

显示每位同学的每科成绩和平均分；

显示每门科目的平均成绩；

对每门成绩进行排序并由高到底显示；

对整个文件进行打包。

**三、程序代码及运行结果**

1. **实验一**

#include<iostream>

using namespace std;

class Coordinate { // 定义Coordinate类

public:

Coordinate()

{

times = 2;

cout << "Coordinate construction1 called!" << endl;

}

Coordinate(int times1)

{

times = times1;

cout << "Coordinate construction2 called!" << endl;

}

~Coordinate()

{

cout << "Coordinate destruction called!" << endl;

}

void InputCoord()

{

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

{

cout << "Please Input x:" << endl;

cin >> Coord[i][1];

cout << "Please Input y:" << endl;

cin >> Coord[i][2];

}

}

void ShowCoord()

{

cout << "The coord is:" << endl;

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

{

cout << "(" << Coord[i][1] << "," << Coord[i][2] << ")" << endl;

}

}

void ShowAvgCoord()

{

float avgx = 0;

float avgy = 0;

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

{

avgx = avgx + Coord[i][1];

avgy = avgy + Coord[i][2];

}

avgx = avgx / times;

avgy = avgy / times;

cout << "The AVG coord is:" << endl;

cout << "(" << avgx << "," << avgy << ")" << endl;

}

private:

float Coord[100][100]; // 存放输入坐标的数组

int times; // 存放输入坐标数目

};

int main()

{

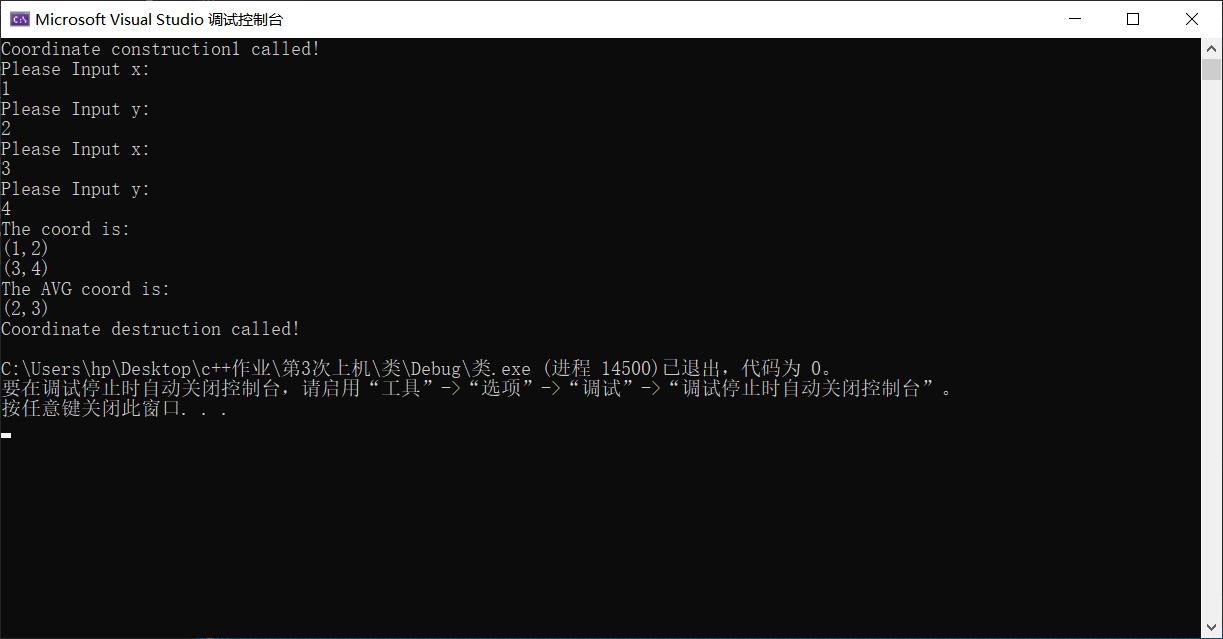
Coordinate y(5);

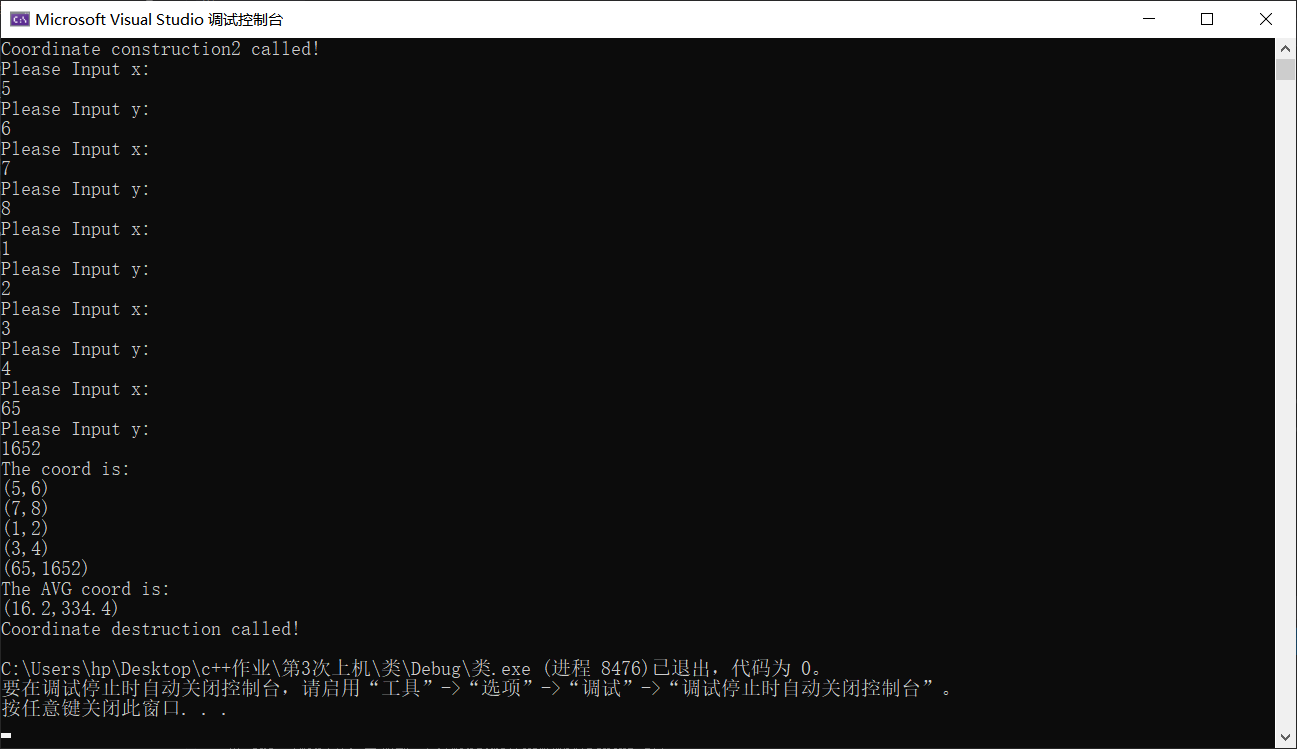
y.InputCoord();

y.ShowCoord();

y.ShowAvgCoord();

}





1. **实验二**

#include<iostream>

#include<string>

using namespace std;

class Score {

public:

Score()

{

times = 2;

}

Score(int times1)

{

times = times1;

}

void InputNameAndScore()

{

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

{

cout << "请输入学生姓名:" << endl;

cin >> Name[i];

cout << "请输入科目A成绩:" << endl;

cin >> SScore[i][1];

cout << "请输入科目B成绩:" << endl;

cin >> SScore[i][2];

cout << "请输入科目C成绩:" << endl;

cin >> SScore[i][3];

}

}

void ShowNameAndScore()

{

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

{

cout << "姓名: " << Name[i] << " 科目A成绩: " << SScore[i][1] << " 科目B成绩 " << SScore[i][2] << " 科目C成绩: " << SScore[i][3] << endl;

}

}

void ShowStdentAvgScore(int Sid)

{

float avg = 0;

avg = (SScore[Sid][1] + SScore[Sid][2] + SScore[Sid][3]) / 3;

cout << "姓名: " << Name[Sid] << " 平均成绩: " << avg << endl;

}

void ShowClassAvgScore(string ClassName)

{

int Cid;

float avg = 0;

if (ClassName == "A") Cid = 1;

if (ClassName == "B") Cid = 2;

if (ClassName == "C") Cid = 3;

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

{

avg = avg + SScore[i][Cid];

}

avg = avg / times;

cout << "课程名称: " << ClassName << "平均成绩: " << avg << endl;

}

void OrderScore(string ClassName)

{

int Cid;

if (ClassName == "A") Cid = 1;

if (ClassName == "B") Cid = 2;

if (ClassName == "C") Cid = 3;

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

{

SScore1[i] = SScore[i][Cid];

}

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

{

Name1[i] = Name[i];

}

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

{

if (SScore1[i] > SScore1[i - 1])

{

float temp = SScore1[i - 1];

SScore1[i - 1] = SScore1[i];

SScore1[i] = temp;

string temp1;

temp1 = Name1[i - 1];

Name1[i - 1] = Name1[i];

Name1[i] = temp1;

}

}

cout << "课程名称: " << ClassName << endl;

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

{

cout << "姓名: " << Name1[i] << " 成绩: " << SScore1[i] << endl;

}

}

private:

float SScore[100][3], SScore1[100];

string Name[100], Name1[100];

int times;

};

int main()

{

Score x;

x.InputNameAndScore();

x.ShowNameAndScore();

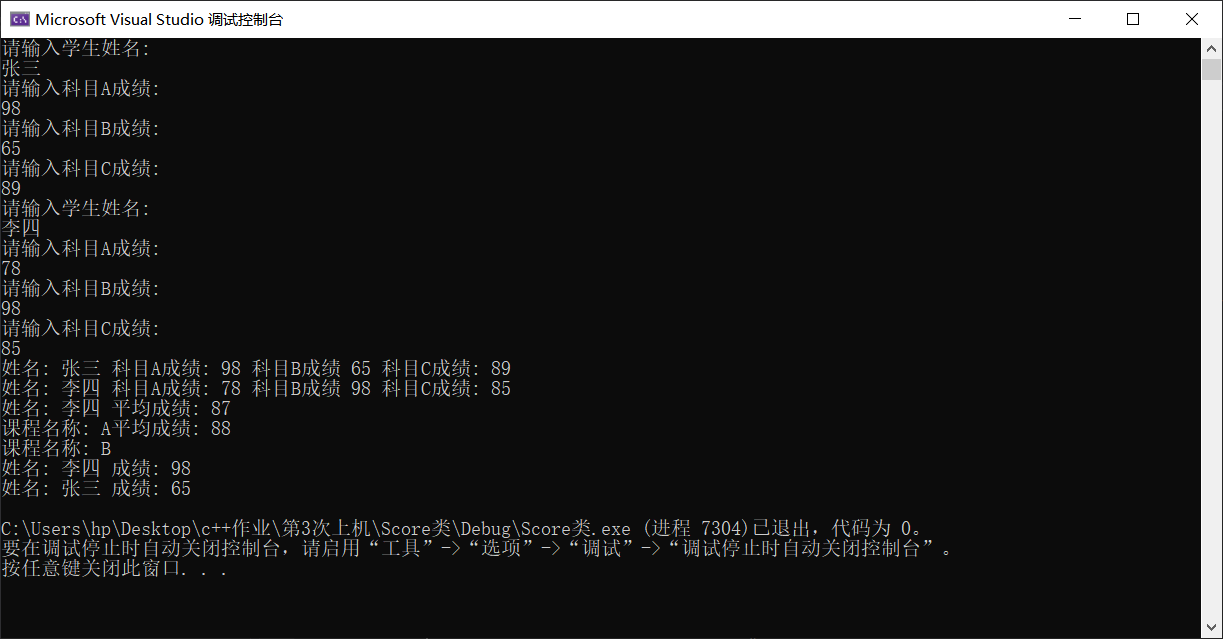
x.ShowStdentAvgScore(1);

x.ShowClassAvgScore("A");

x.OrderScore("B");

return 0;

}



**四、感想心得**

从此次上机实验中我真正开始体会到了面向对象程序设计与面向过程程序设计的区别，C++可以更加系统、结构更加清晰地去设计一个程序，每一部分都通过一个类来完成，这样一来程序更加直观，降低了出错的几率。与此同时，通过这次实验，我对于“类”逐渐有了更为直观的了解。

Copyright ©2021-2099 ChenkaiZhang. All rights reserved