第四次上机实验报告

**学号：**202030310186  **姓名：申一帆**  **班级：**自动2002

1. **实验目的**

1、进一步加深对类和对象的理解

2、掌握几种对象传递的使用方法

3、掌握静态成员的概念和使用

1. **实验内容**
2. **实验一**

了解三种不同的对象传递方式

1. **实验二**

掌握静态成员的概念和使用

由关键字static 修饰说明的成员，称为静态成员（static class member）。

静态成员为所有对象共享，只有一份存于公用内存中。

静态成员包括静态数据成员和静态函数成员。

1. **实验三**

**创建一个TStudent类，完成以下功能：**

该类包括三个函数：InitStudent、ExpendMoney和ShowMoney；

采用float m\_ClassMoney变量作为静态变量，用于存储班级的班费，初始值设为1000；

InitStudent (char name[])主要负责完成学生姓名的初始化；

ExpendMoney(float money)主要完成班费的花销计算；

ShowMoney()主要完成班费余额显示；

主程序中分别定义A、B、C三个学生，每个学生为一个对象，每个学生分别消费班费50,98.5,500.53，最后显示班费的余额。

1. **实验程序结果**
2. **实验一**

#include<iostream>

using namespace std;

class Tr {

public:

Tr(int n)

{

i = n;

}

void set\_i(int n)

{

i = n;

}

int get\_i()

{

return i;

}

private:

int i;

};

void sqr\_it(Tr ob) // 对象ob作为函数sqr\_it的形参

{

ob.set\_i(ob.get\_i() \* ob.get\_i());

cout << "在函数sqr\_it内，形参对象ob的数据成员i的值为:" << ob.get\_i();

cout << endl;

}

//void sqr\_it(Tr \* ob) // 对象指针ob作为函数sqr\_it的形参

//{

// ob->set\_i(ob->get\_i()\*ob->get\_i());

// cout <<"在函数sqr\_it内，形参对象ob的数据成员i的值为:"<< ob->get\_i();

// cout << endl;

//}

//void sqr\_it(Tr&ob)

//{

// ob.set\_i(ob.get\_i()\*ob.get\_i());

// cout <<"在函数sqr\_it内，形参对象ob的数据成员i的值为:"<< ob.get\_i();

// cout << endl;

//}

int main()

{

Tr obj(10);

cout << "调用函数sqr\_it前, 实参对象obj的数据成员i的值为:";

cout << obj.get\_i() << endl;

sqr\_it(obj);

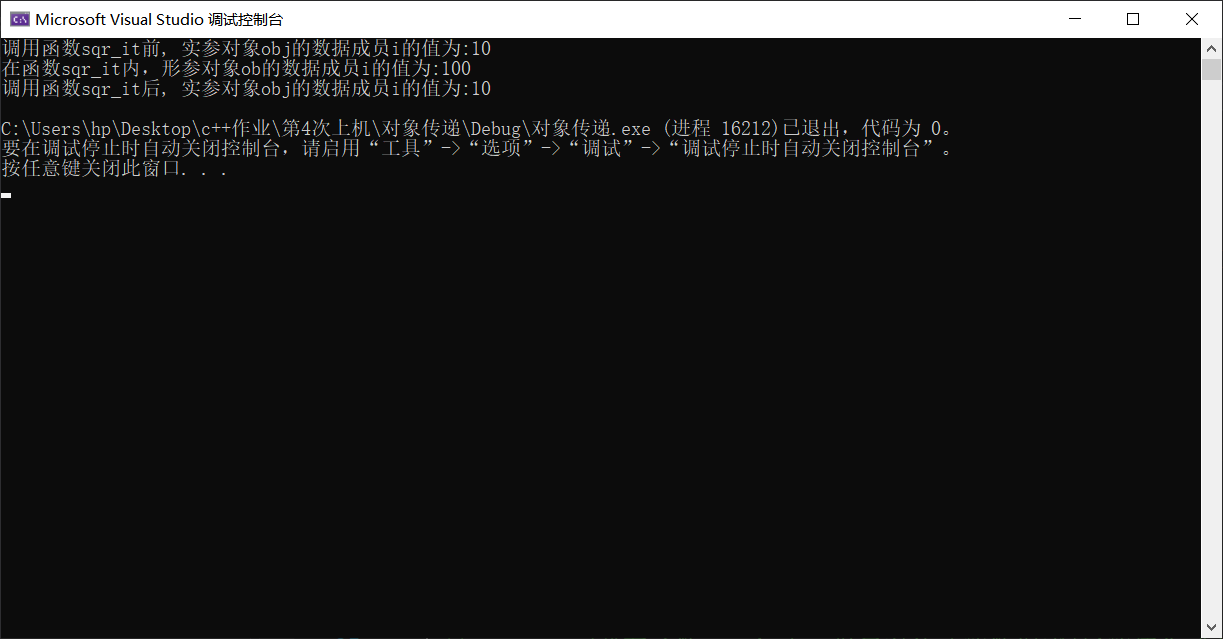
//sqr\_it(&obj);// 对象指针ob作为函数sqr\_it的形参

cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

cout << obj.get\_i() << endl;

return 0;

}



1. **实验二**

# include<iostream>

using namespace std;

class Ctest {

static int count;

public:

Ctest() {

++count; cout << "对象数量=" << count << '\n';

}

};

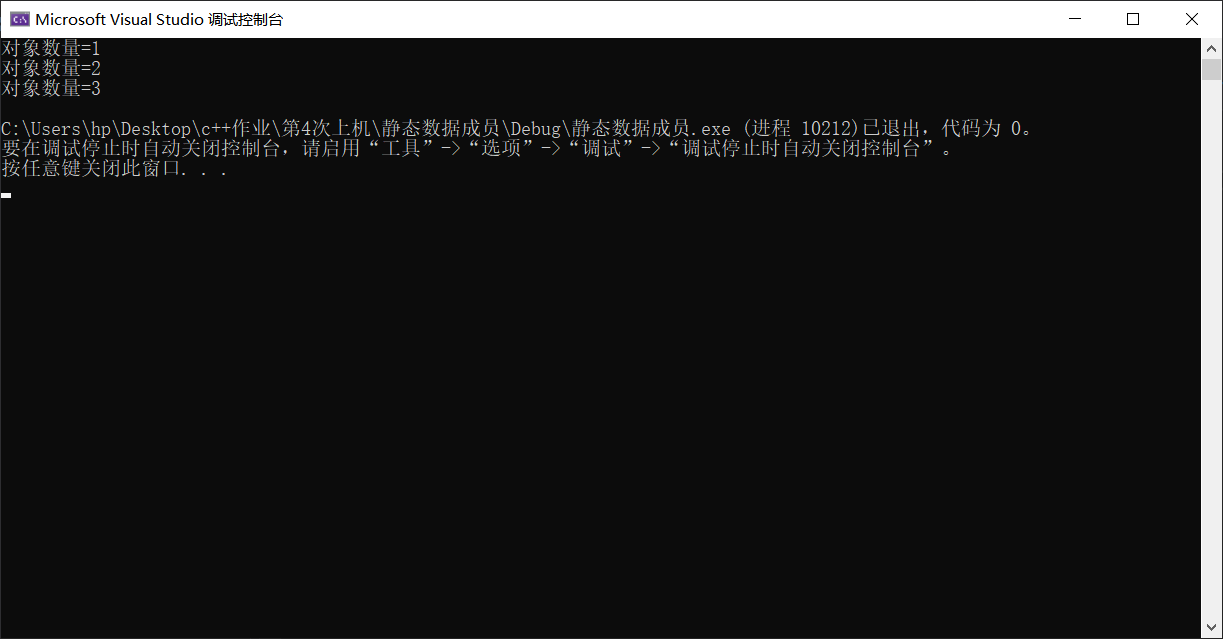
int Ctest::count = 0;

int main(void) {

Ctest a[3];

return 0;

}



1. **实验三**

#include <iostream>

using namespace std;

class TSstudent

{

public:

float money;

static double m\_ClassMoney;

void InitStudent()

{

char name[100];

}

void ExpendMoney(double a)

{

money = a;

}

double ShowMoney()

{

m\_ClassMoney -= money;

return m\_ClassMoney;

}

};

double TSstudent::m\_ClassMoney = 1000;

int main()

{

TSstudent A;

TSstudent B;

TSstudent C;

A.ExpendMoney(50);

cout << "班费还剩余" << A.ShowMoney() << endl;

B.ExpendMoney(98.5);

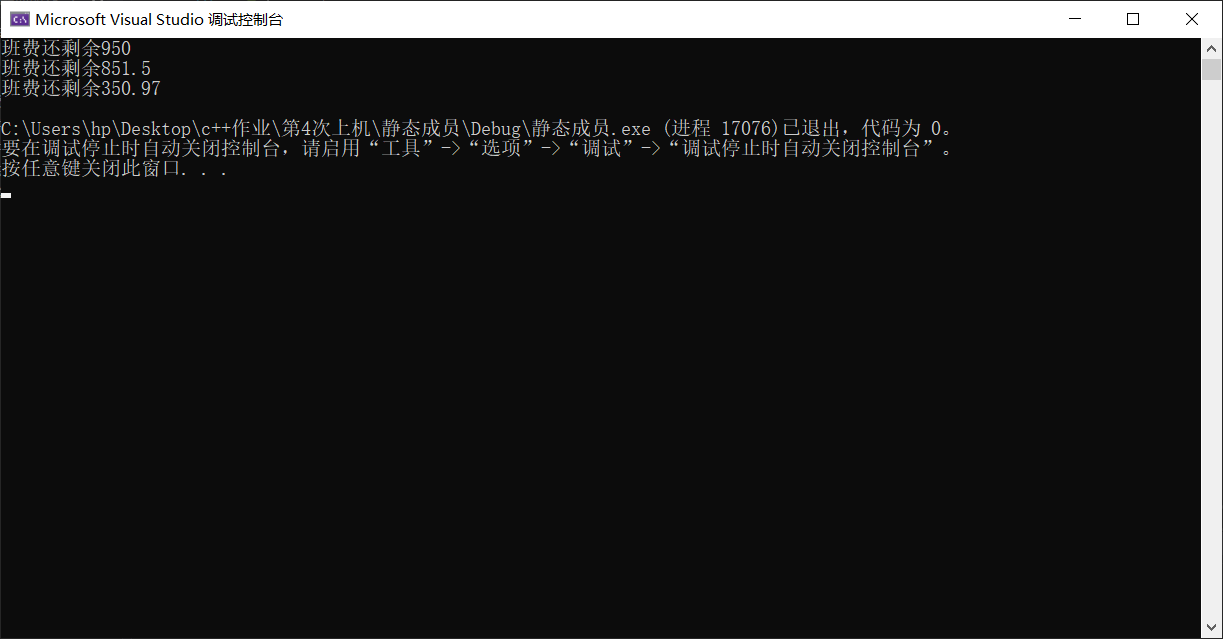
cout << "班费还剩余" << B.ShowMoney() << endl;

C.ExpendMoney(500.53);

cout << "班费还剩余" << C.ShowMoney() << endl;

return 0;

}



1. **实验心得**
2. **类和对象**

类是抽象的，不占用内存。  
对象是具体的，占用内存。  
类是对象的抽象，而对象是类的具体事项。

1. **静态成员**

静态成员可以封装这些算法，调用简单，实现了一个类的多个对象间的数据共享。

Copyright ©2021-2099 YiFanShen. All rights reserved