实验目的：

* **进一步加深对类和对象的理解**
* **掌握几种对象传递的使用方法**
* **掌握静态成员的概念和使用**

实验内容1：

* **了解三种不同的对象传递方式**

代码1：# 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;

}

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);

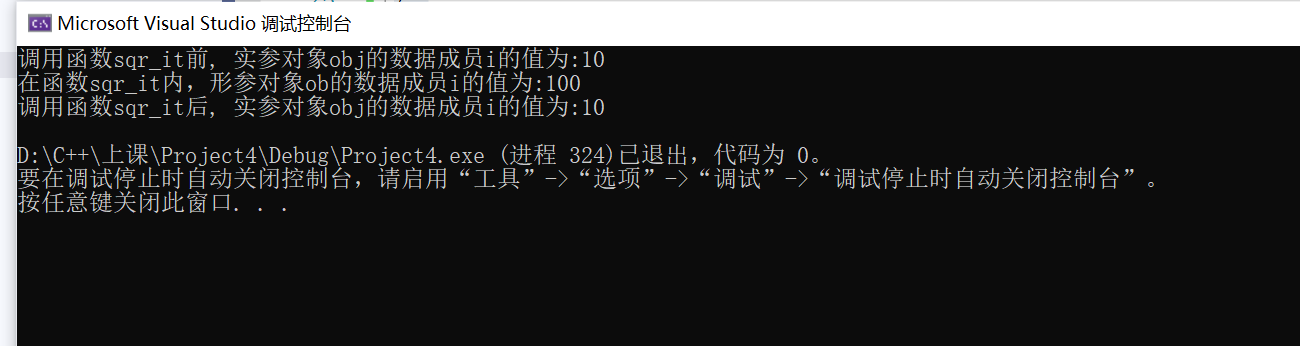
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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

return 0;

}

运行结果：



实验内容2：

* **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。**
* **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。**

代码：

#include <iostream>

#include<string>

using namespace std;

class TStudent {

private:

static float m\_ClassMoney;

string Name;

public:

void InitStudent(const char name[]) {

Name = name;

}

void ExpendMoney(float money) {

m\_ClassMoney = m\_ClassMoney - money;

}

void ShowMoney() {

cout<<Name<< "：班费还剩余" << m\_ClassMoney << endl;

}

};

float TStudent::m\_ClassMoney = 1000;

int main()

{

TStudent A, B, C;

A.InitStudent("A");

A.ExpendMoney(50);

A.ShowMoney();

B.InitStudent("B");

B.ExpendMoney(98.5);

B.ShowMoney();

C.InitStudent("C");

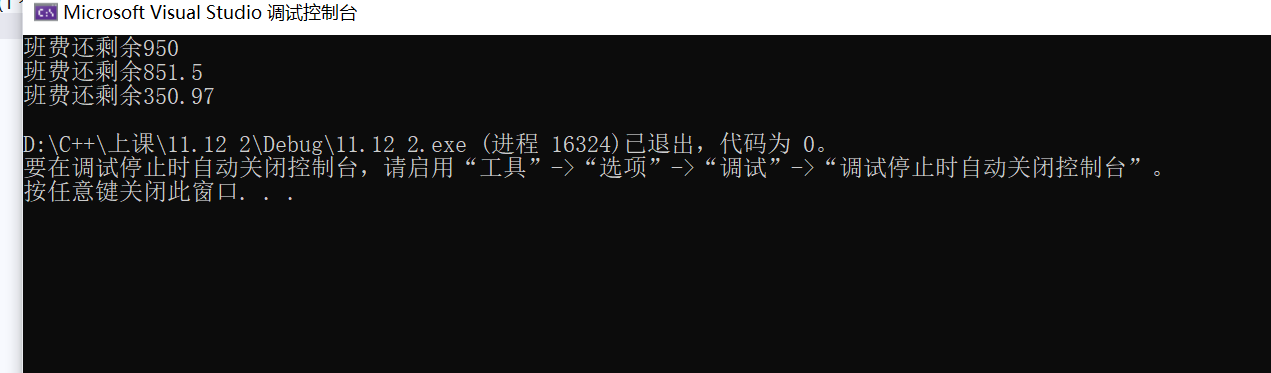
C.ExpendMoney(500.53);

C.ShowMoney();

return 0;

}

运行结果：



感想：

目前新版本的visual中没有iostream.h 文件，所以课件中的代码在我的编译器中是错误了，应该使用#include <iostream>#include<string>这两个头文件。其次，姓名传参中使用const,便于进行类型检查，使编译器对处理内容有更多了解，消除了一些隐患。

Copyright ©2021-2099 Chen Yilu. All rights reserved