**实验内容**

* 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；

**实验代码**

# 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()

{

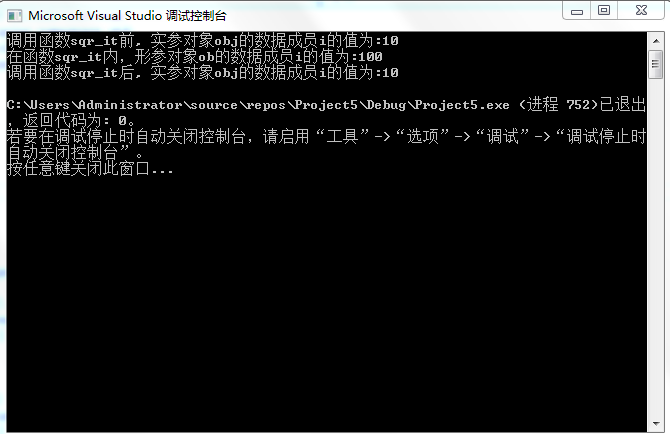
Tr obj(10);

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;

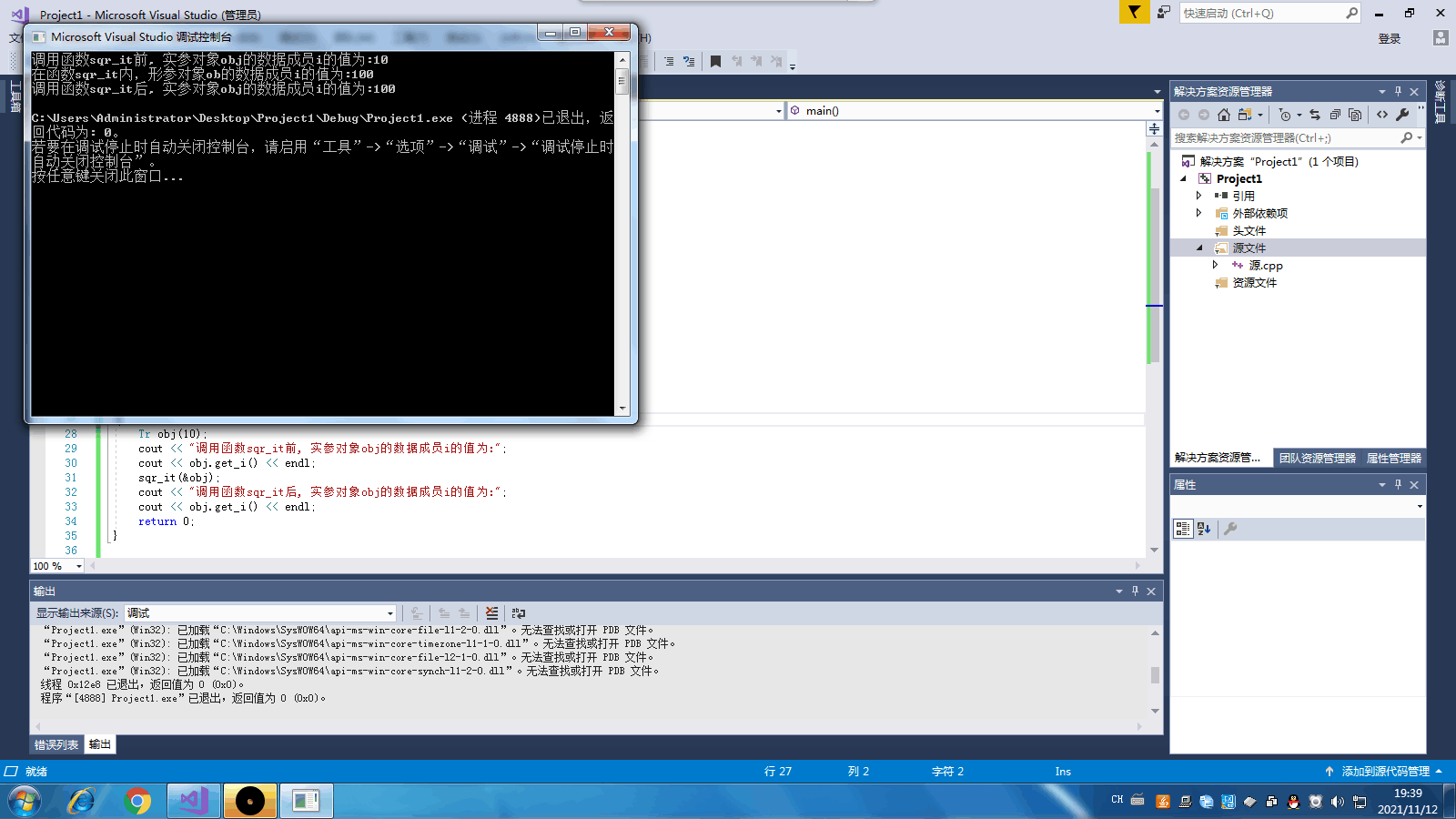
void sqr\_it(Tr\* ob) // 对象指针ob作为函数sqr\_it的形参

{ 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;

}



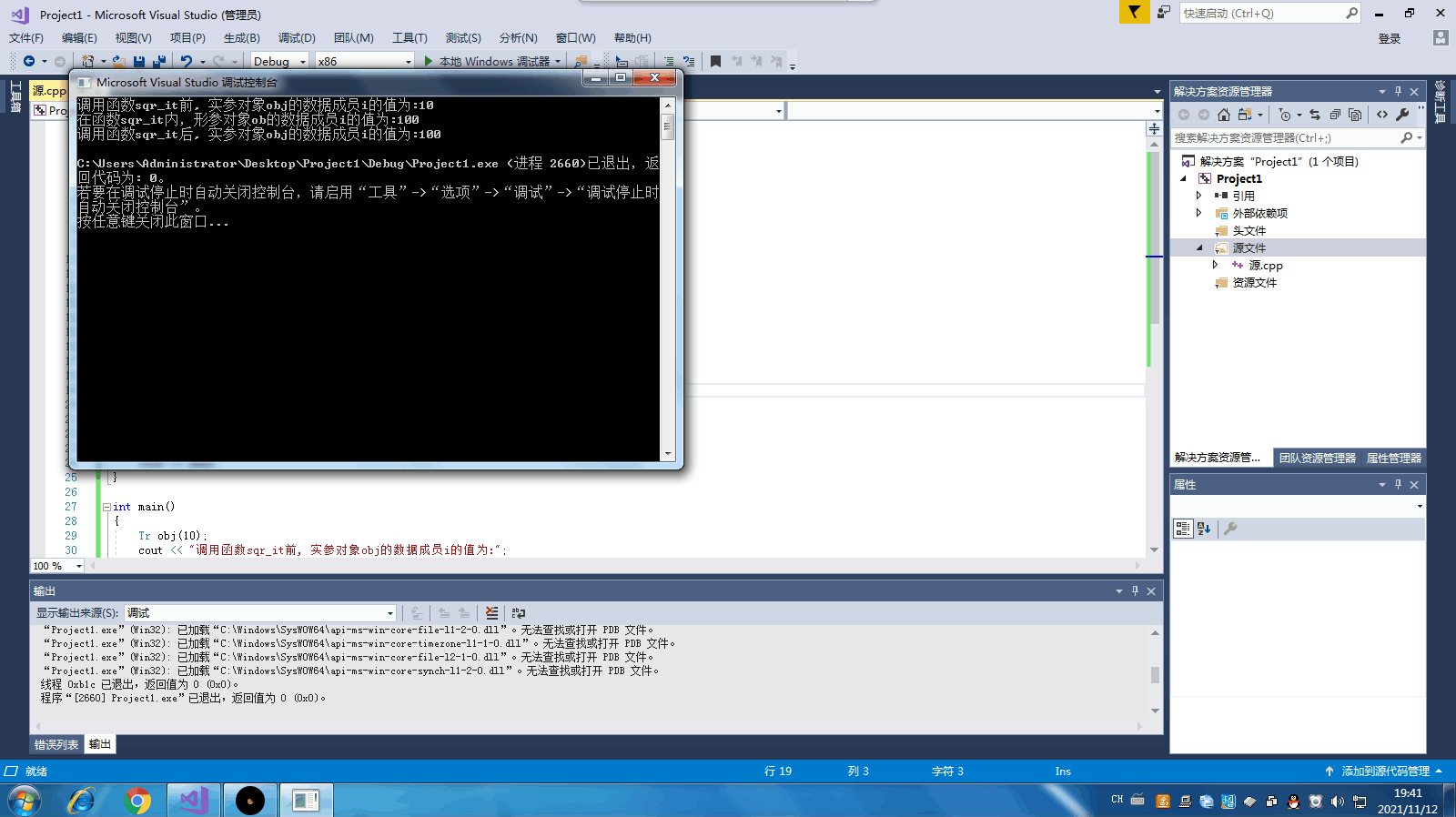
void sqr\_it(Tr& ob)

{ 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;

}



**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。

**程序代码**

#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");

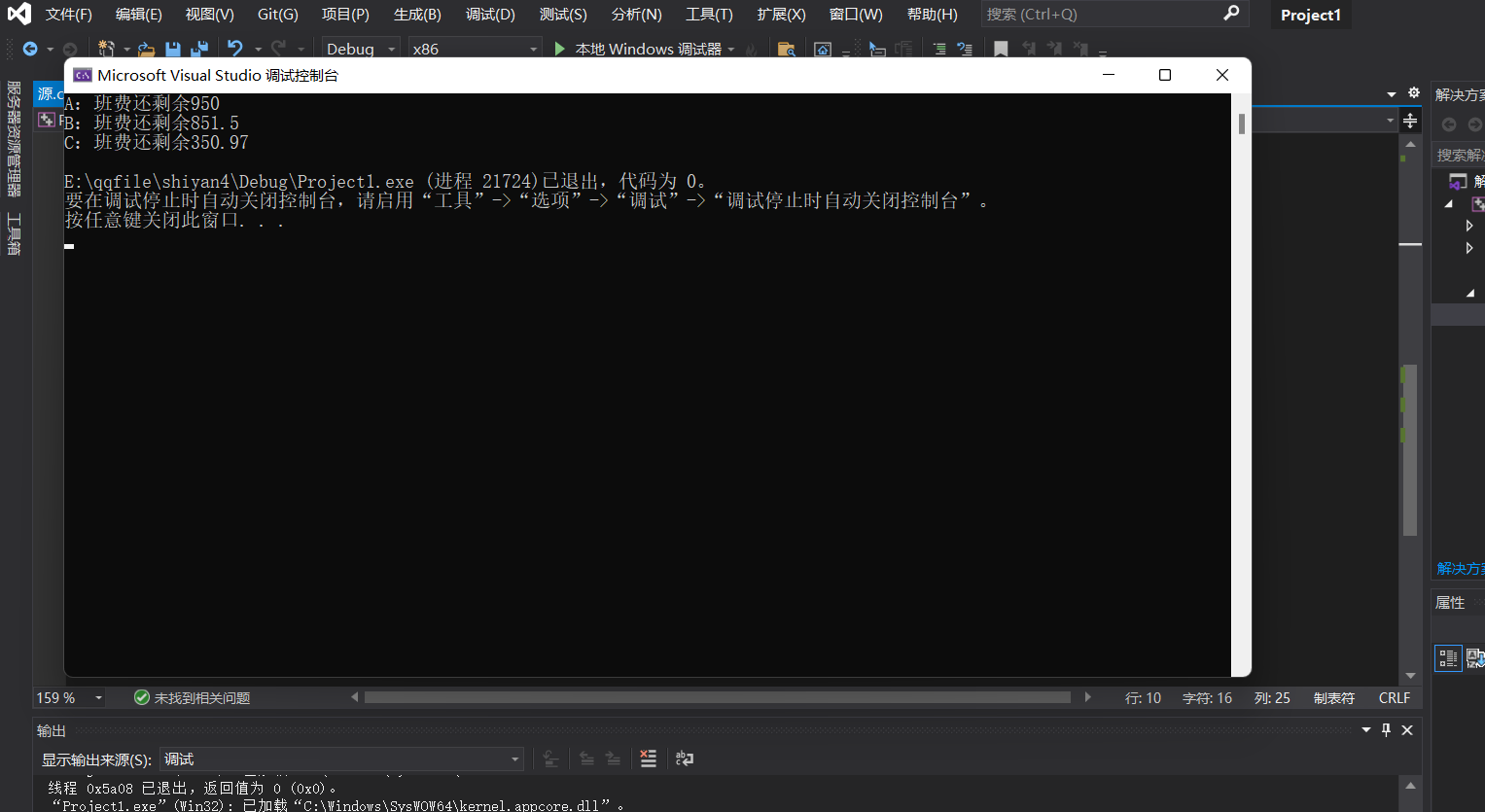
C.ExpendMoney(500.53);

C.ShowMoney();

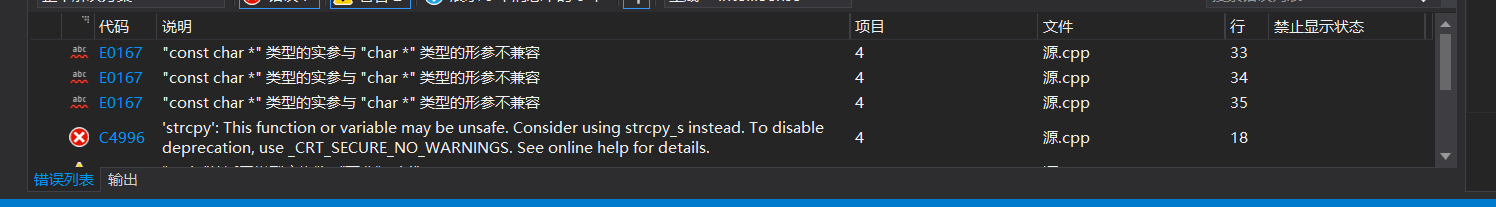
return 0;

}

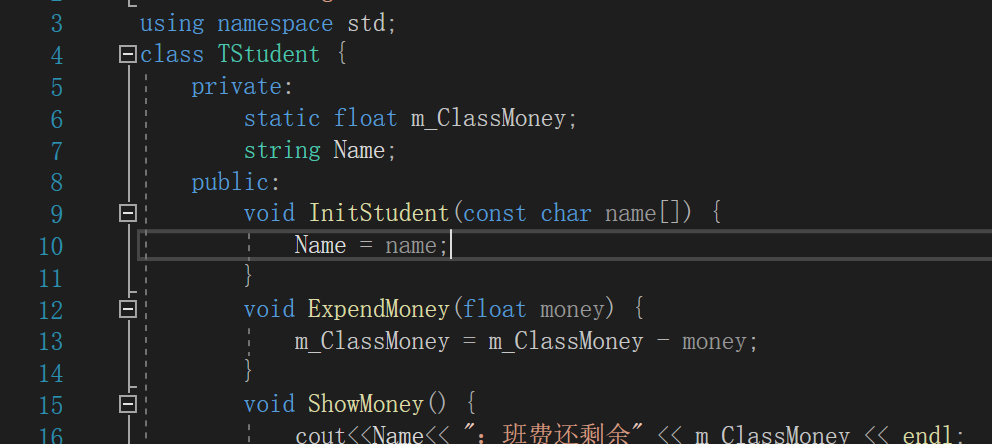
**代码结果**



**感想心得**



对名字的初始化无法完成，传参过程遇到问题，用const 解决了。



第一个实验通过三个函数的不同传递方式，告诉我们不同传递方式的特点，对象作为函数参数，那么改变的只是形参，形参没有发生变化，当对象指针作为函数参数时，实参发生了改变，因为传递的是地址，因而实参会随着形参的改变而发生变化。当对象引用作为函数参数时，改变了形参，从而改变了实参，因为形参就是实参的引用名，改变形参实际就是改变了实参。

第二个实验是静态成员变量的应用方法，实验较有难度。

Copyright ©2021-2099 ZhengGuangYing. All rights reserved