# 第四次上机

### 实验内容

1. 了解三种不同的对象传递方式
2. 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。

### 程序代码

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;

}

//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);

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;

}

2.

#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);

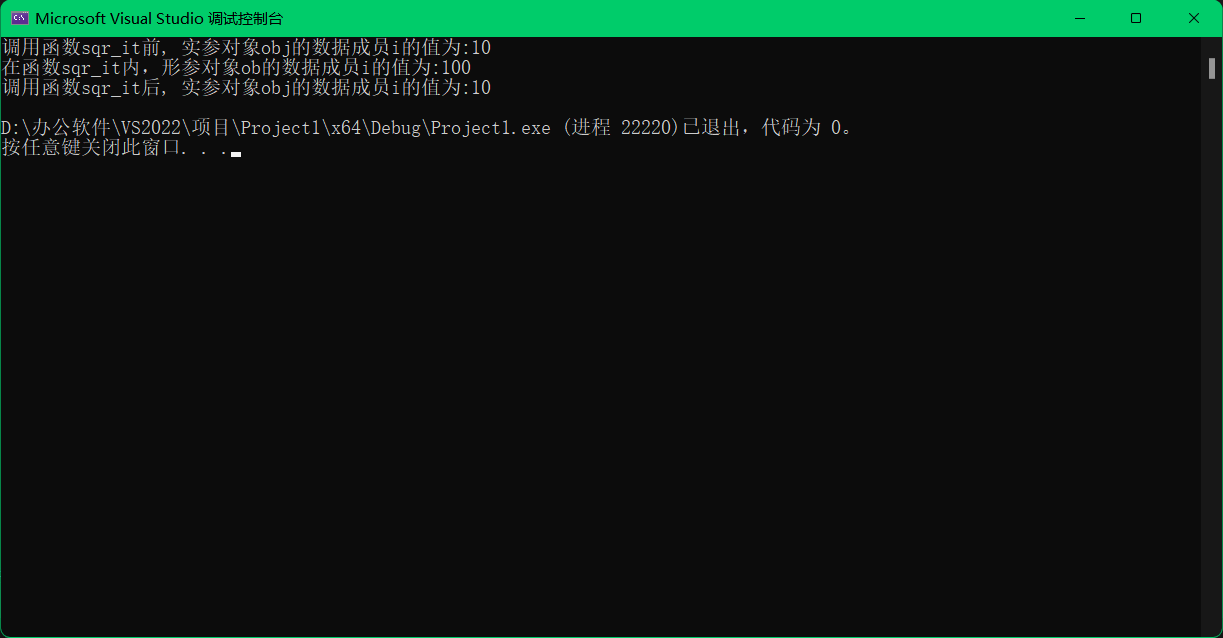
C.ShowMoney();

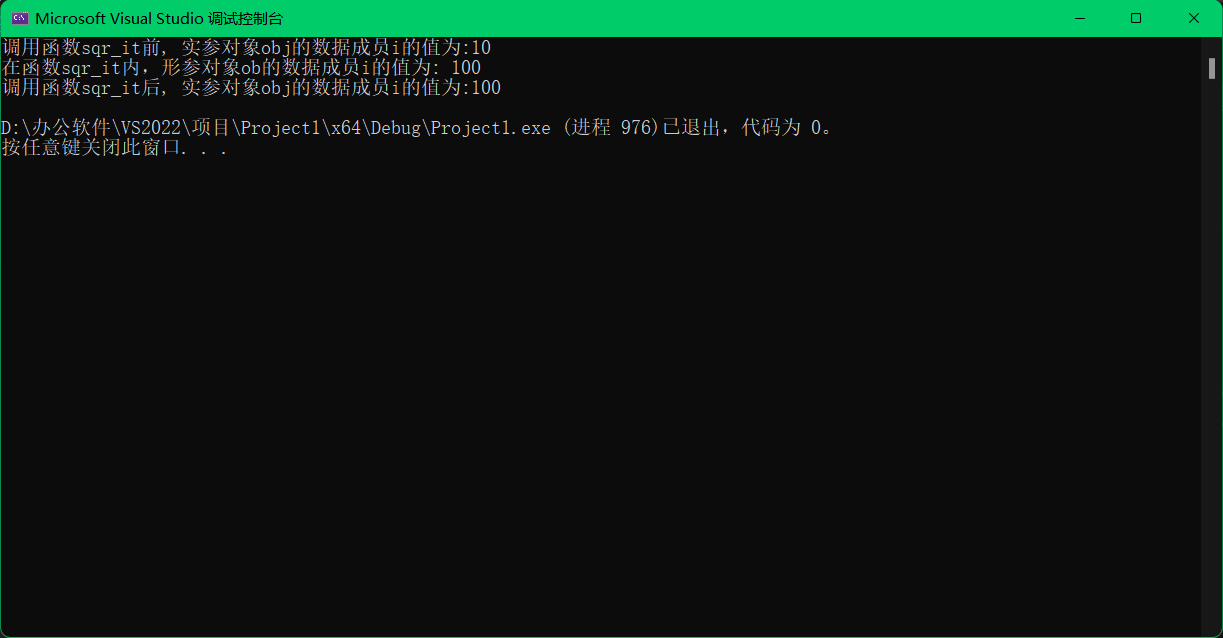
return 0;

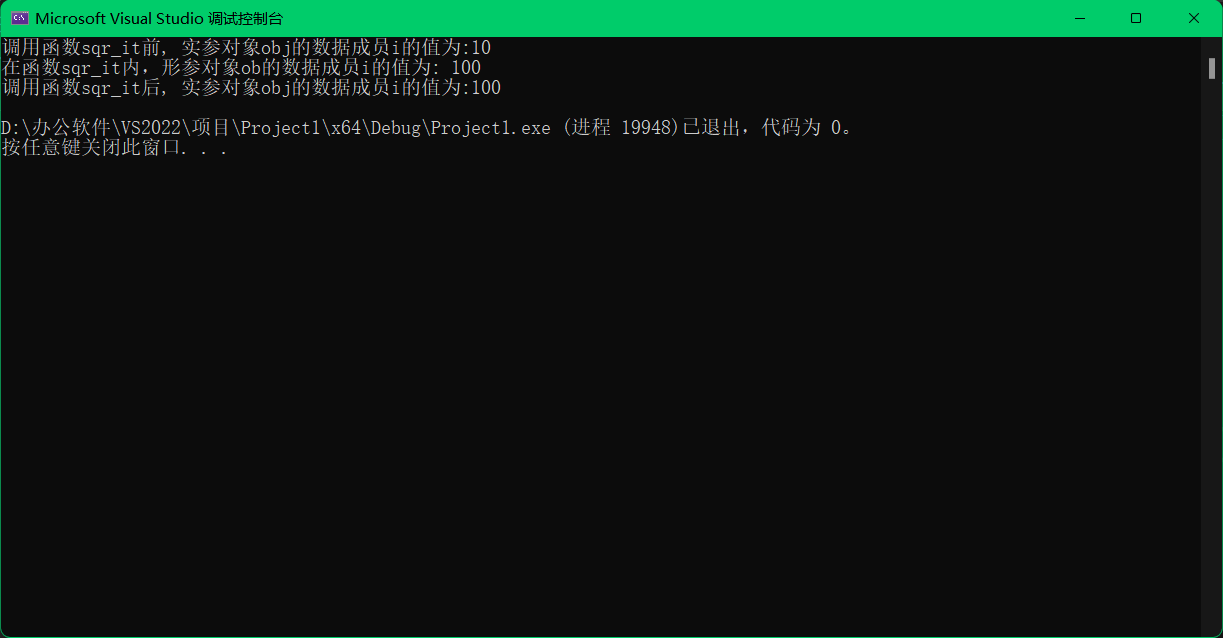
}

### 程序结果

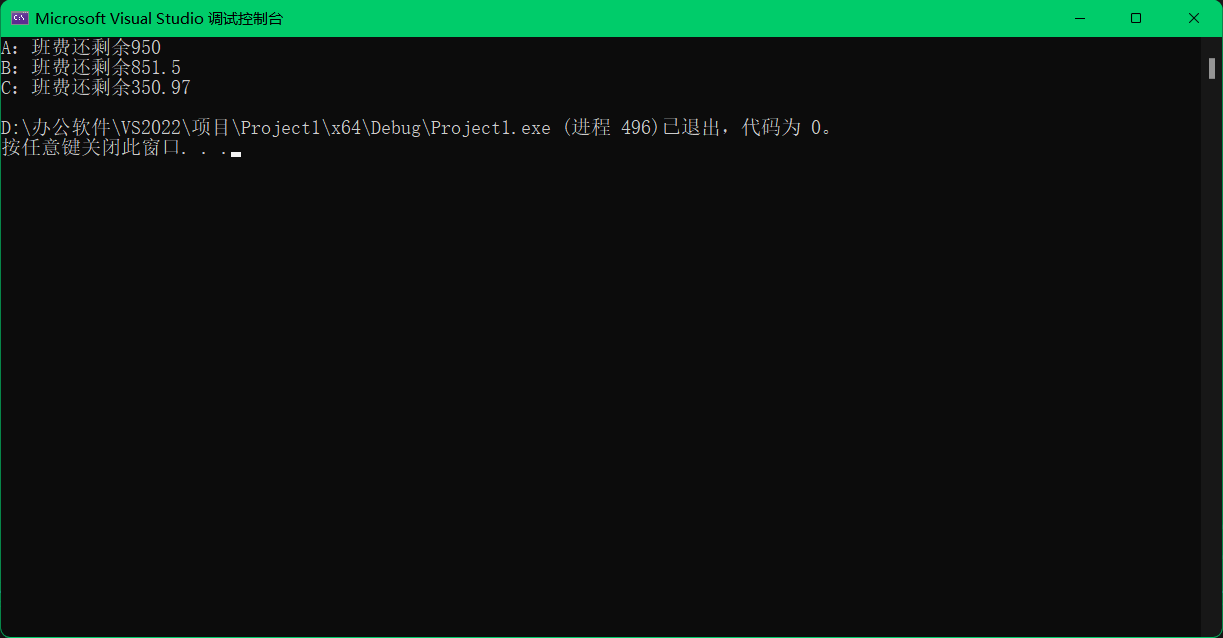
1.







2.



### 感想心得

静态成员函数一般不访问普通数据成员，它的作用主要是访问和操作同类中的静态数据成员或全局变量。

一个类的静态成员函数与非静态成员函数不同，它不需要创建任何该类的对象就可以被调用。

静态数据函数使用有点困难，需要思考得程序和信息很多，最后总结运行程序时也需要一定时间，做完后仍感觉不能很好掌握静态函数的使用。

Copyright ©2021-2099 qiuhaolong. All rights reserved