第四次上机实验

1. **程序代码**

**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;

};

①、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;

}

②、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;

}

用指针时，主函数中调用sqr\_it函数时，参数表需要用地址，如&obj。

③、void sqr\_it(Tr& ob) // 对象引用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;

}

**实验二**

#include<iostream>

#include<string>

#pragma warning(disable: 4996)

using namespace std;

class TStudent

{

private:

char m\_Name[6];

static float m\_ClassMoney;

//m\_ClassMoney为静态成员变量，保存班费

public:

void InitStudent(const char\*);

void ExpendMoney(float);

static void ShowMoney(); //静态成员函数

};

float TStudent::m\_ClassMoney = 1000; //静态成员变量的初始化

void TStudent::InitStudent(const char name[])

{

strcpy(m\_Name , name);

}

void TStudent::ExpendMoney(float money)

{

m\_ClassMoney -= money; //班费为原先的减去花费的

cout << m\_Name << "花费班费" << money << endl;

}

void TStudent::ShowMoney()

{

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

}

void main()

{

TStudent stu[3];//定义三个学生

stu[0].InitStudent("A");

stu[1].InitStudent("B");

stu[2].InitStudent("C");

stu[0].ExpendMoney(50);

TStudent::ShowMoney();

stu[1].ExpendMoney(98.5);

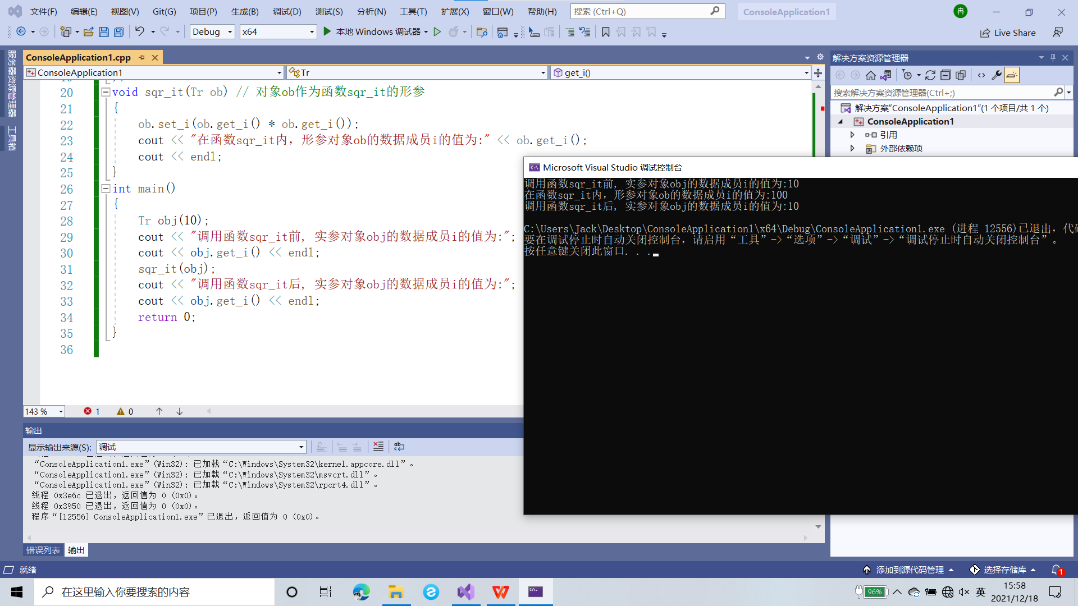
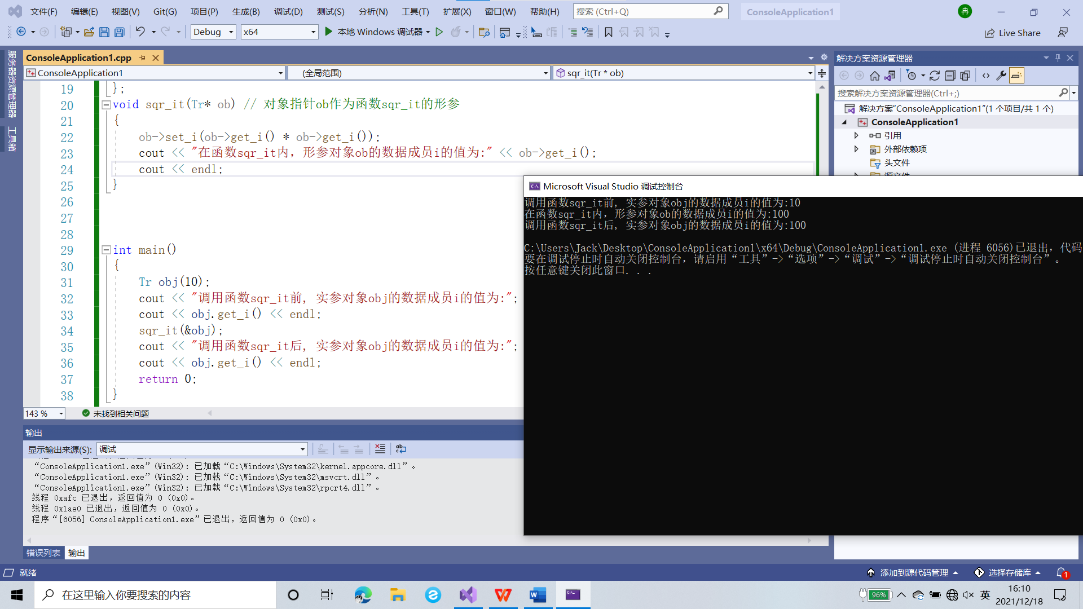
TStudent::ShowMoney();

stu[2].ExpendMoney(500.53);

TStudent::ShowMoney();

}

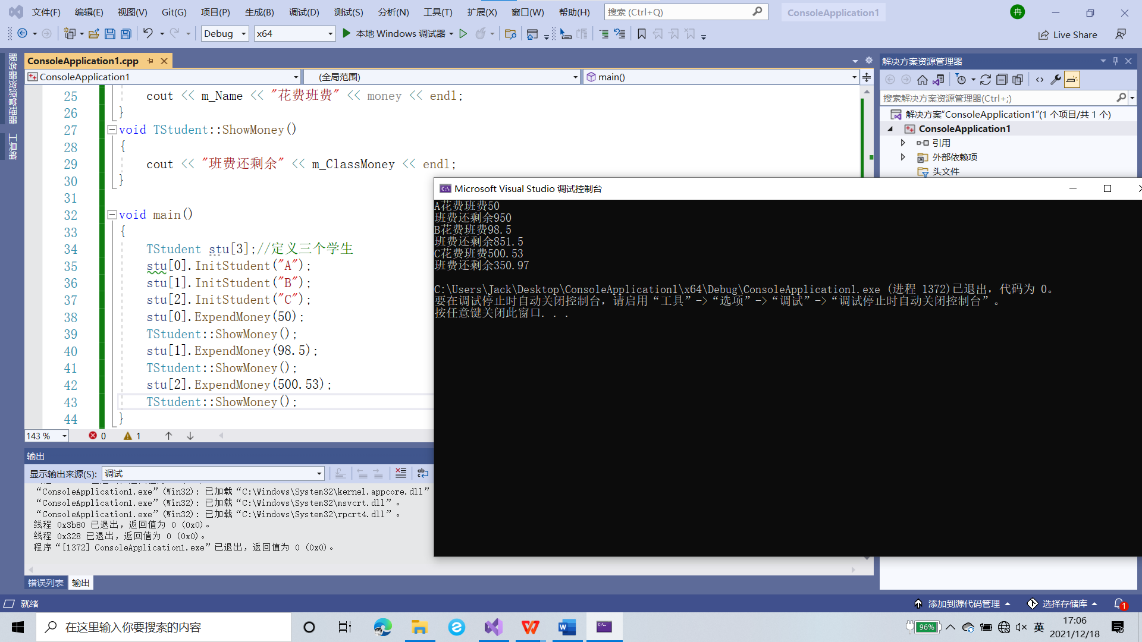
**2、程序结果**

**实验一**

①

②

③ 引用和指针的结果一致

**实验二**

**3、感想心得**

本次上机一共操作了两个实验，第一个主要是为了让我们了解三种不同的函数传递方式，了解实参传递给形参后，三种方式会有哪些变化，结果可以表明，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。

第二个实验则是为了了解静态成员变量与函数，理解他们的使用方法与特性，静态成员变量在定义主函数之前，定义类之后就应该被定义，而静态成员函数在主函数建立对象之前就可以引用，以便处理静态成员变量。

在第二个上机实验中我遭遇了许多困难，比如在在定义学生姓名时，没有想到如何去定义，在之后老师讲解答案之后，我才明白方法。在运用strcpy时，总是出现错误，最后才搞清楚这不是编译错误，是C++为了安全不允许使用strcpy，要想使用需要加协议，忽略此错误。