实验报告（第四次上机）

电自2002 毛佳林

实验一：

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);

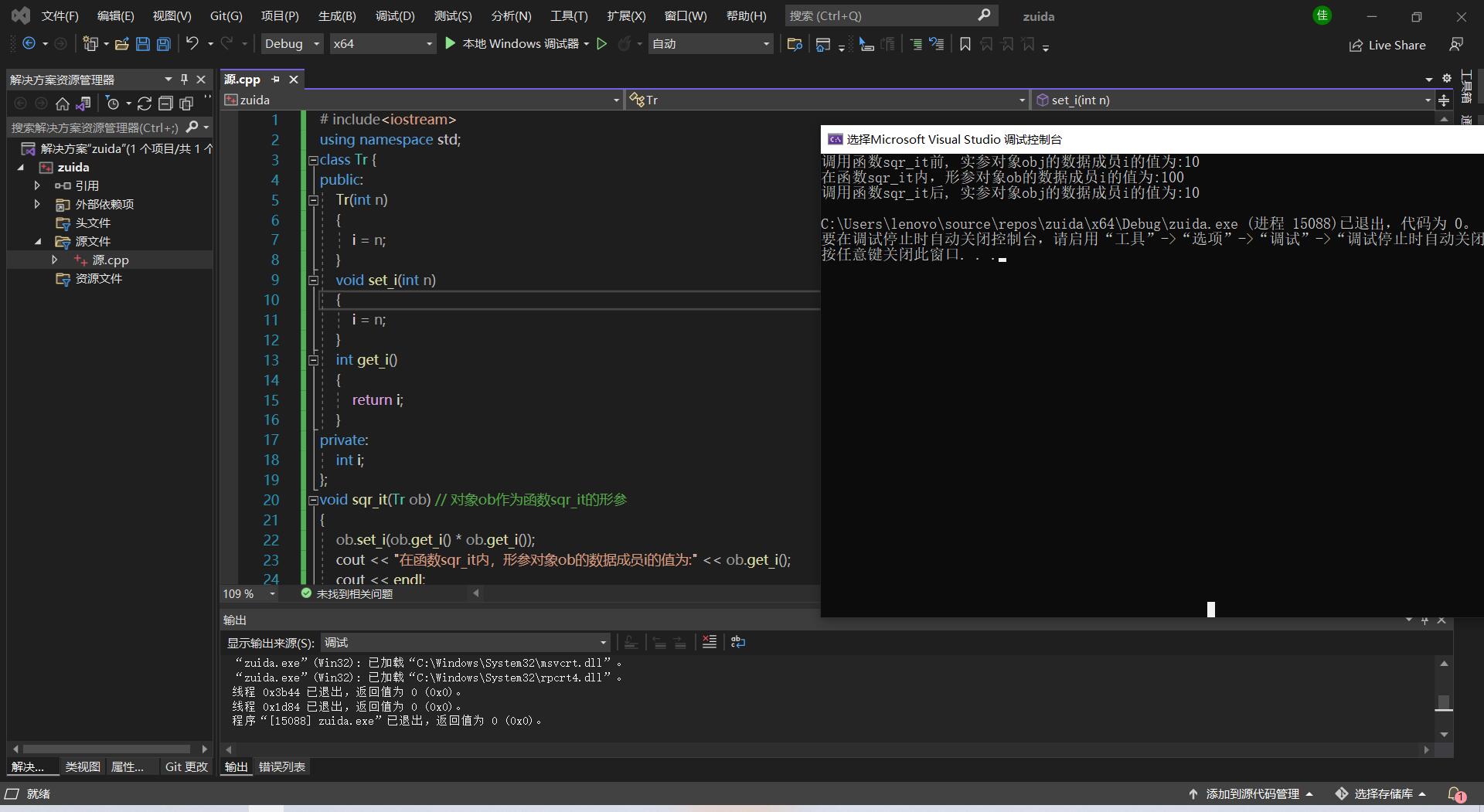
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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

return 0;

}

图片：



实验二：

代码：

#include<iostream>

#include<cmath>

#include<string>

using namespace std;

class tstudent {

private:

static float m\_classmoney;

string name;

public:

static void showmoney();

void initstudent(string name1);

void expendmoney(float money);

};

float tstudent::m\_classmoney = 1000;

void tstudent::initstudent(string name1)

{

name = name1;

}

void tstudent::expendmoney(float money)

{

m\_classmoney = m\_classmoney - money;

}

void tstudent::showmoney()

{

cout << "There is " << m\_classmoney << "yuan left" << endl;

}

int main()

{

int i;

float b[3];

string a[3];

tstudent s1, s2, s3;

for (i = 0; i < 3; i++)

{

cout << "Please input student's name: " << endl;

cin >> a[i];

cout << "Please input how much money did this student spend: " << endl;

cin >> b[i];

}

s1.initstudent(a[0]);

s1.expendmoney(b[0]);

s2.initstudent(a[1]);

s2.expendmoney(b[1]);

s3.initstudent(a[2]);

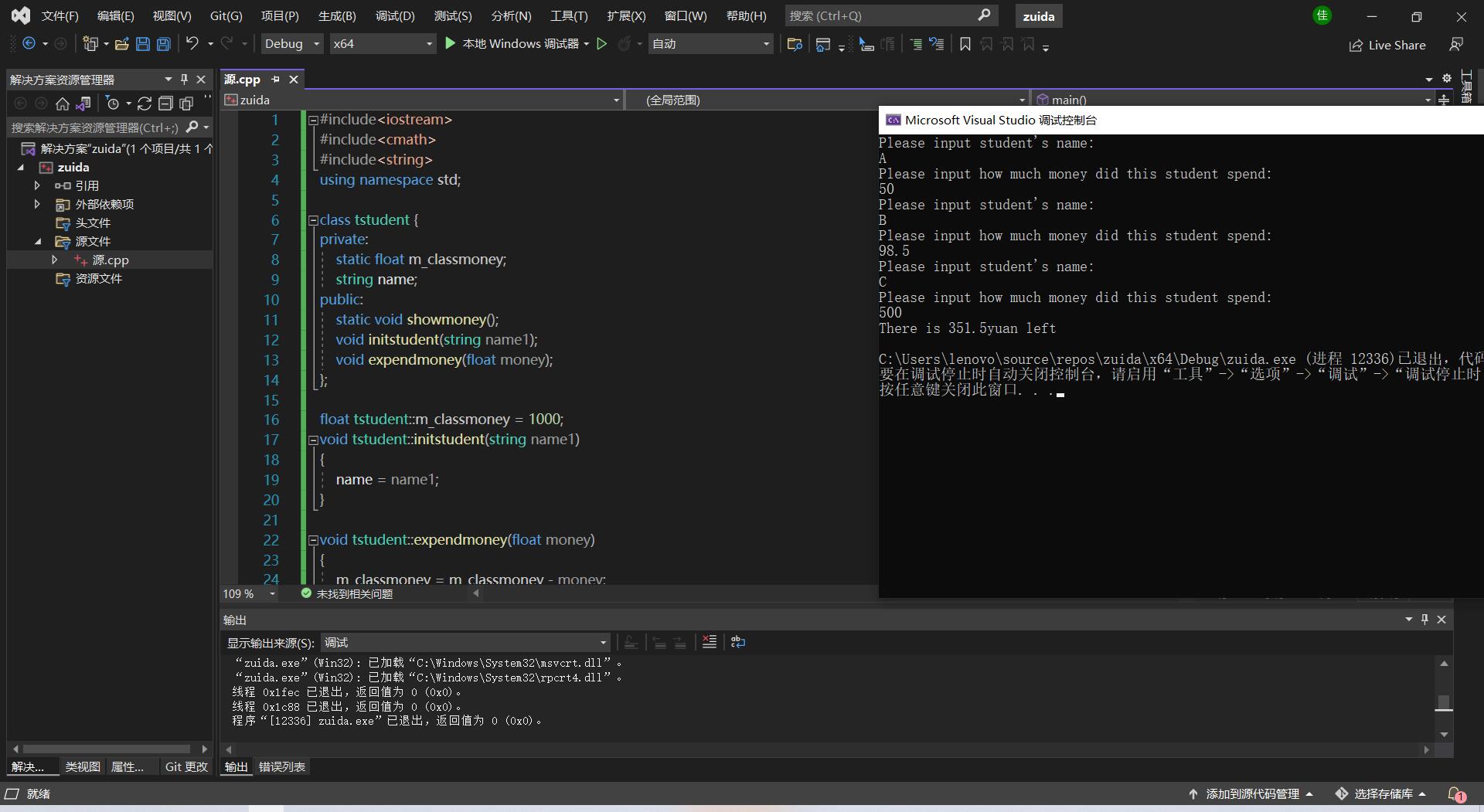
s3.expendmoney(b[2]);

s1.showmoney();

return 0;

}

图片：



总结：

本次实验我了解到C++有着三种不同的传递方式：值传递、指针传递以及引用传递，通过编写程序，使我更好地了解到其是如何工作的；对于实验二，我了解到静态成员的作用以及用法——静态成员为所有对象共享，只有一份存于公用内存当中。静态成员包括静态数据成员以及静态函数成员。对于类的多个不同对象，它们都能够通过对象的成员函数对静态成员进行修改，或者说，静态成员独立于对象而存在，它的存在不随对象的创建而创建，不随对象的删除而删除，我认为这便是静态成员相较于非静态成员最大的区别。