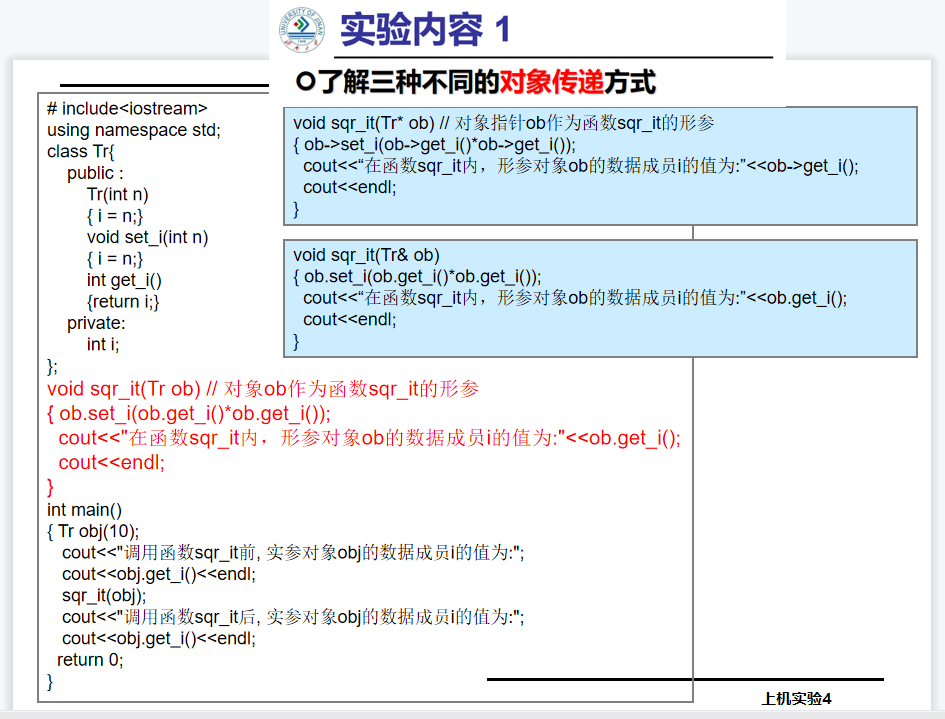
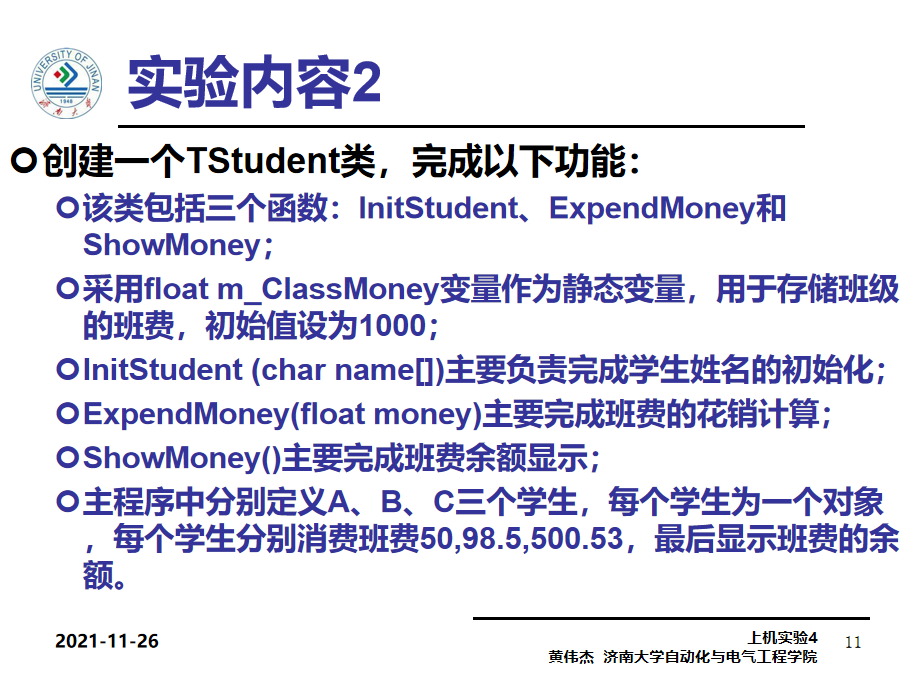
1. 实验题目

上机实验4：对象传递与静态成员





二、实验过程

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 实验一 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

————————————————原始代码：——————————————

# 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);

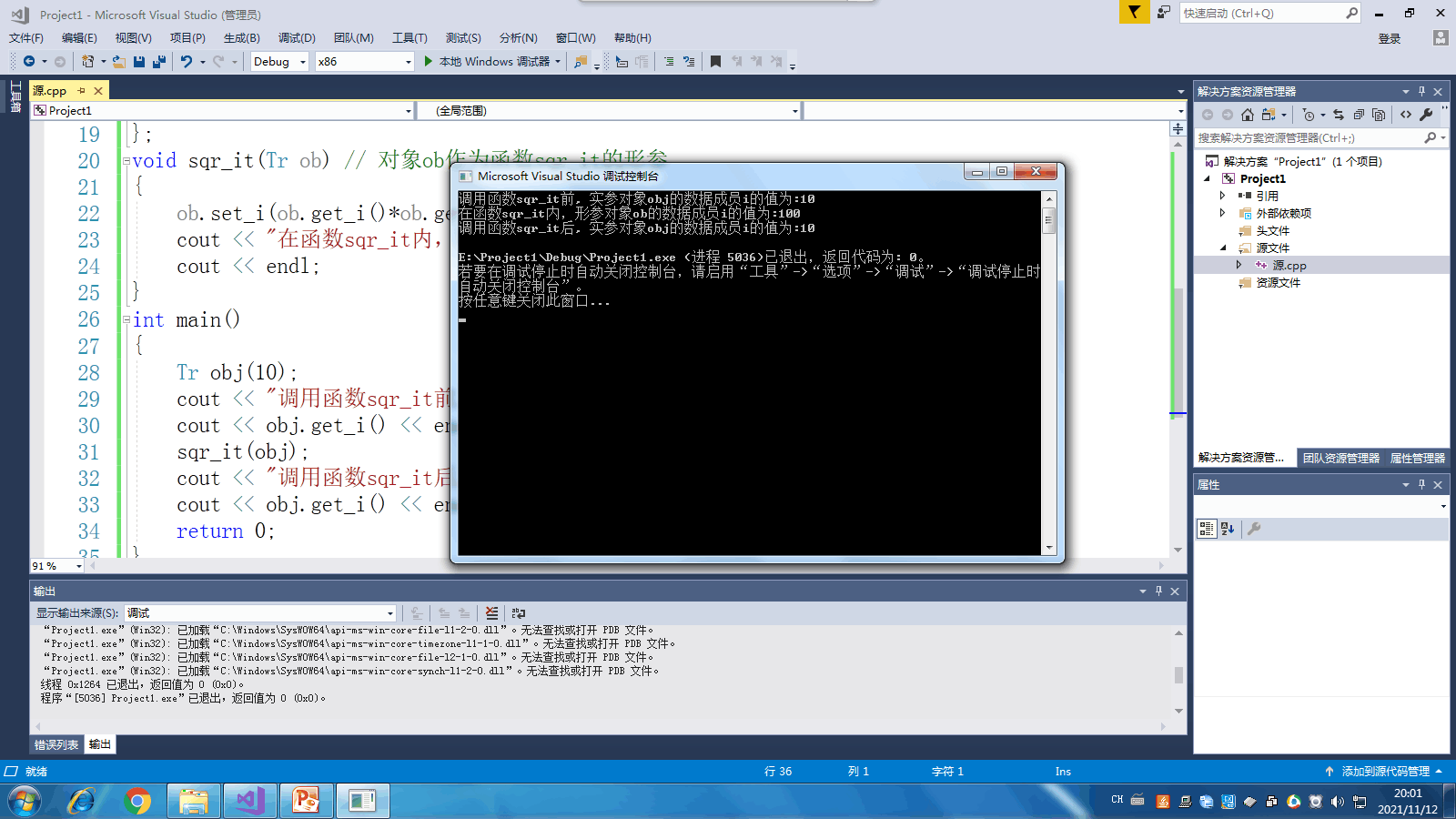
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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

return 0;

}

———————————————运行结果：———————————————



——————————①函数sqr\_i和主函数更换为：———————————

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;

}

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);

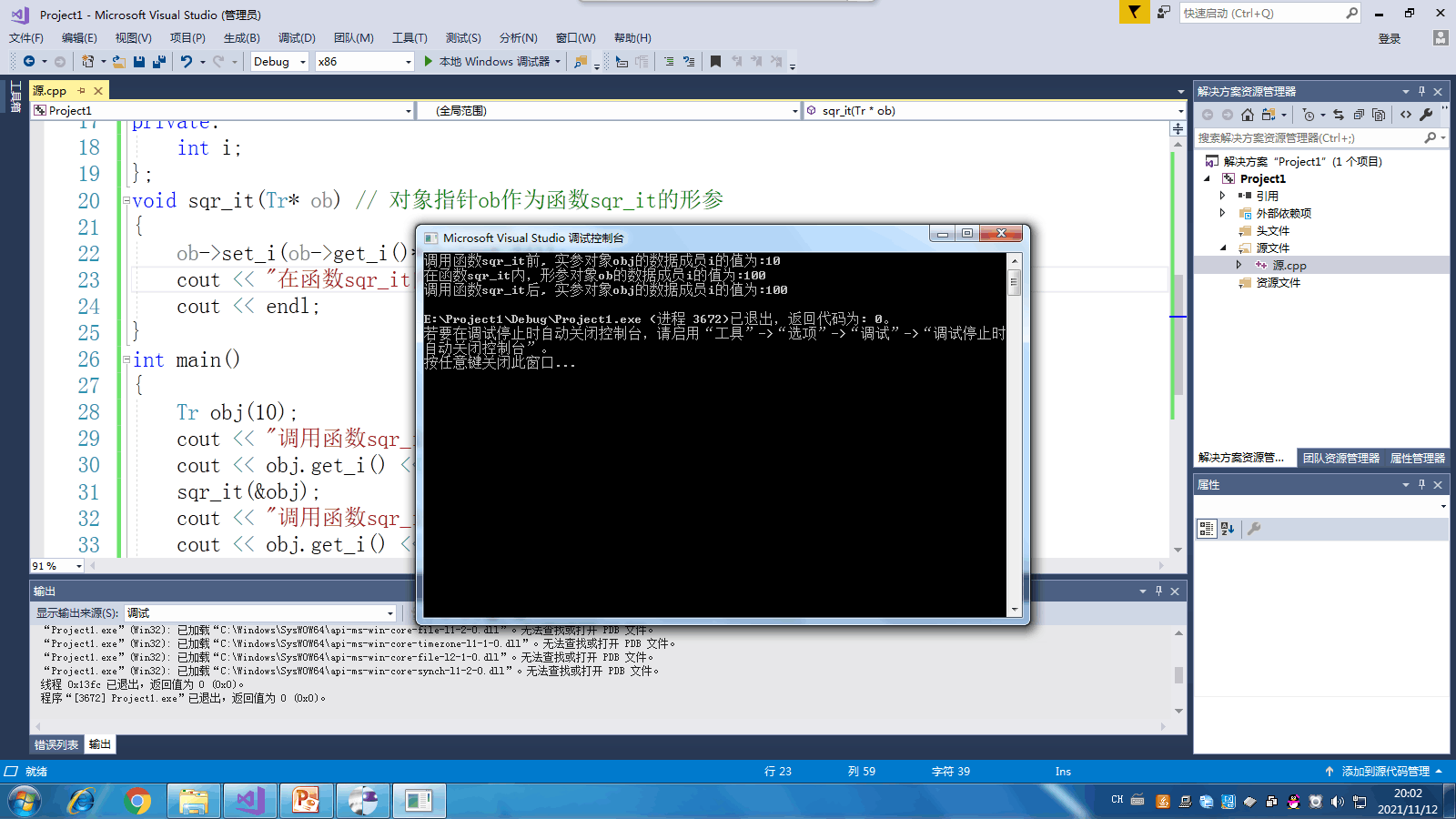
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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

return 0;

}

———————————————①运行结果：——————————————



———————————②函数sqr\_i和主函数更换为：——————————

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);

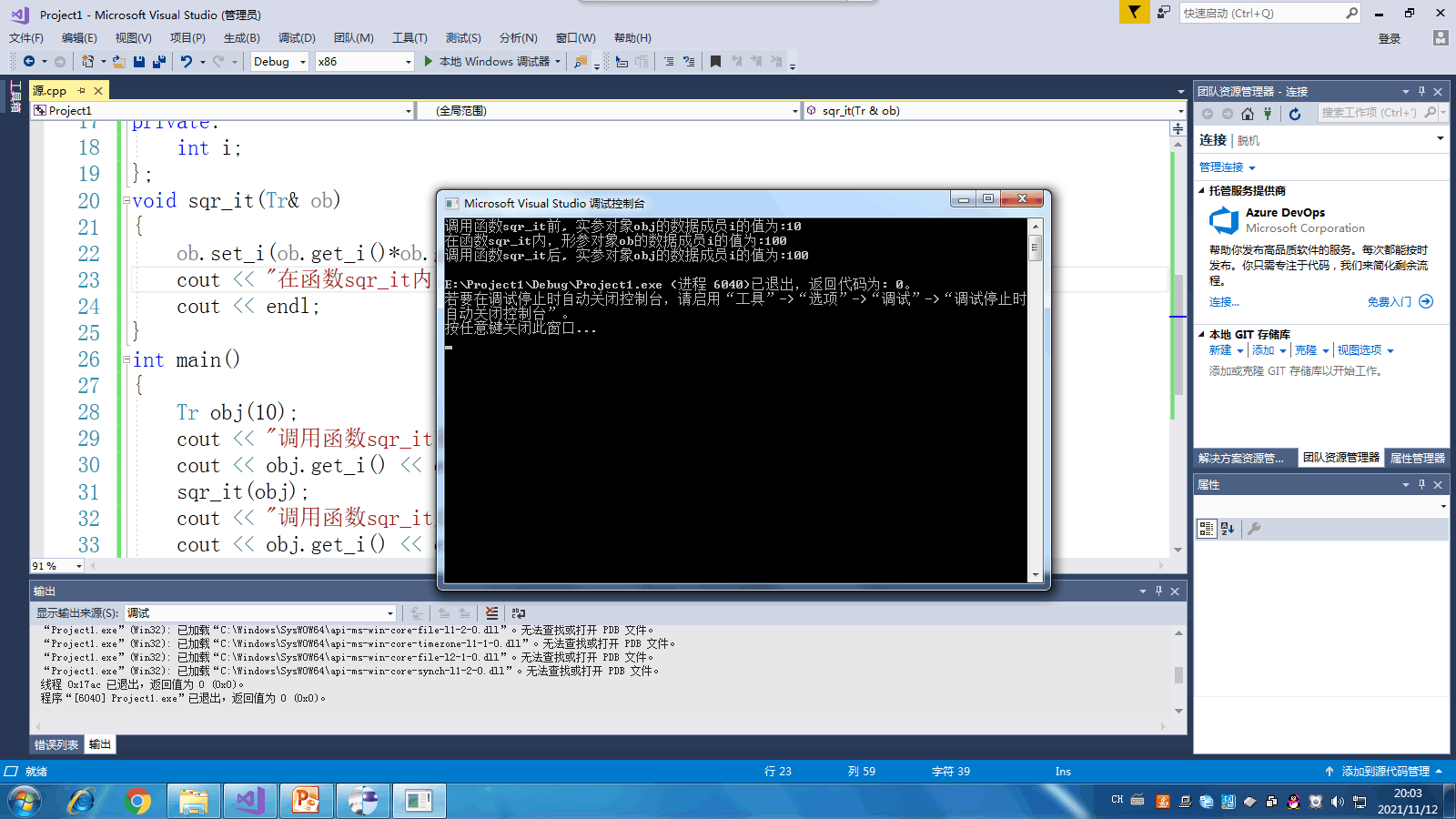
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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

return 0;

}

———————————————②运行结果：——————————————

****

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 实验二 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

—————————————————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("aa");

A.ExpendMoney(50);

A.ShowMoney();

B.InitStudent("bb");

B.ExpendMoney(98.5);

B.ShowMoney();

C.InitStudent("cc");

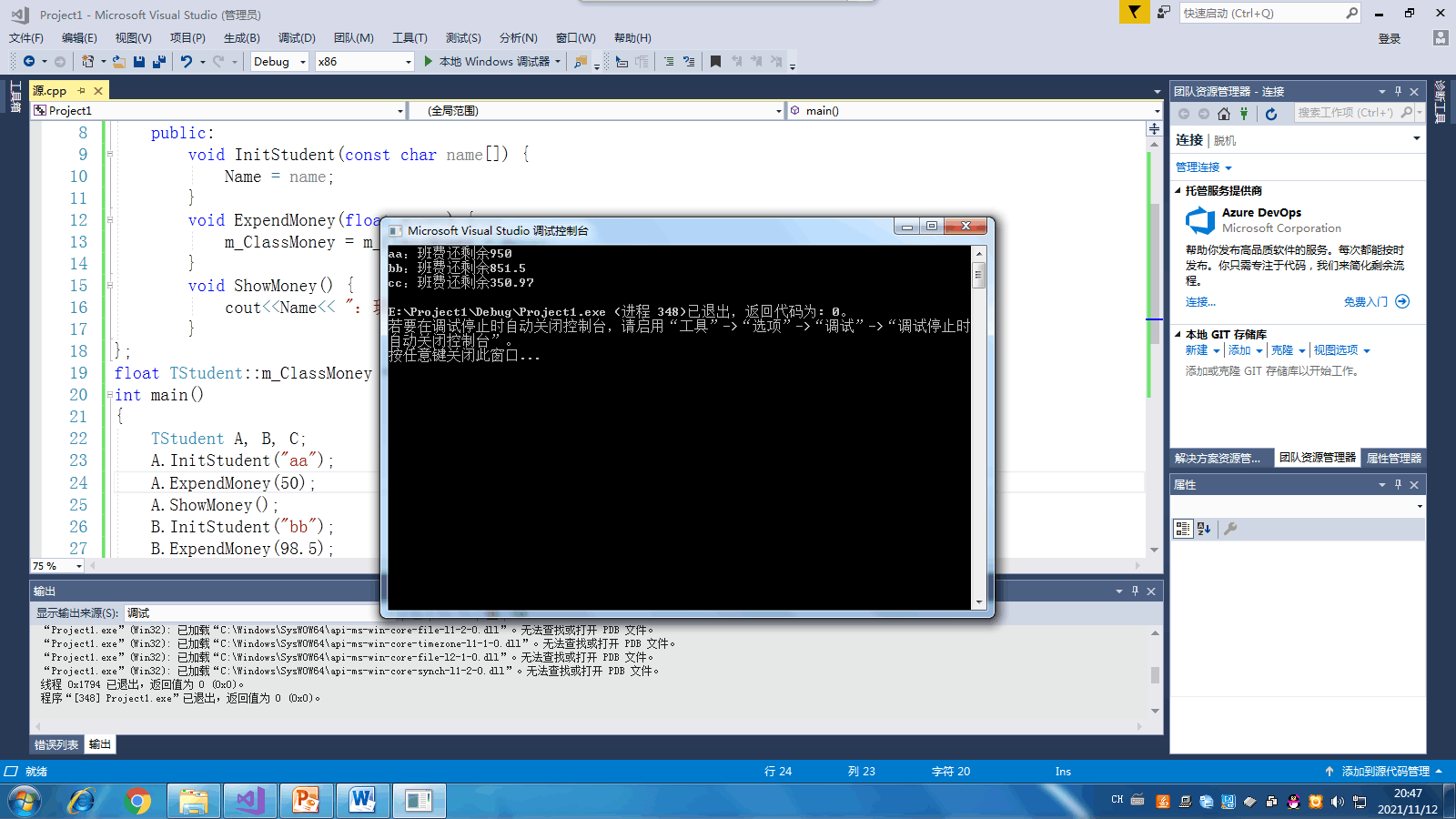
C.ExpendMoney(500.53);

C.ShowMoney();

return 0;

}

————————————————运行结果：——————————————



二、问题分析与感想

1、在char name[]前面需要加上const来表明这是一个不变的量，否则会出现报错的问题

2、在输出学生姓名时，要在前面加上#include<string>，否则无法正常输出string类型的变量

3、实验一中，初始程序里的sqr\_it函数没有返回值，所以只有在函数内改变了；改成指针之后，对原地址对应的值改变，所以函数调用完毕后，值改变；改成引用后，变量始终只有这一个，所以函数调用完毕后，值也改变。

电自2002 许吴瑕 202030310293

Copyright ©2021-2099 WuxiaXu. All right reserved