实验报告（第四次上机）

自2001 郭锴

# 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);

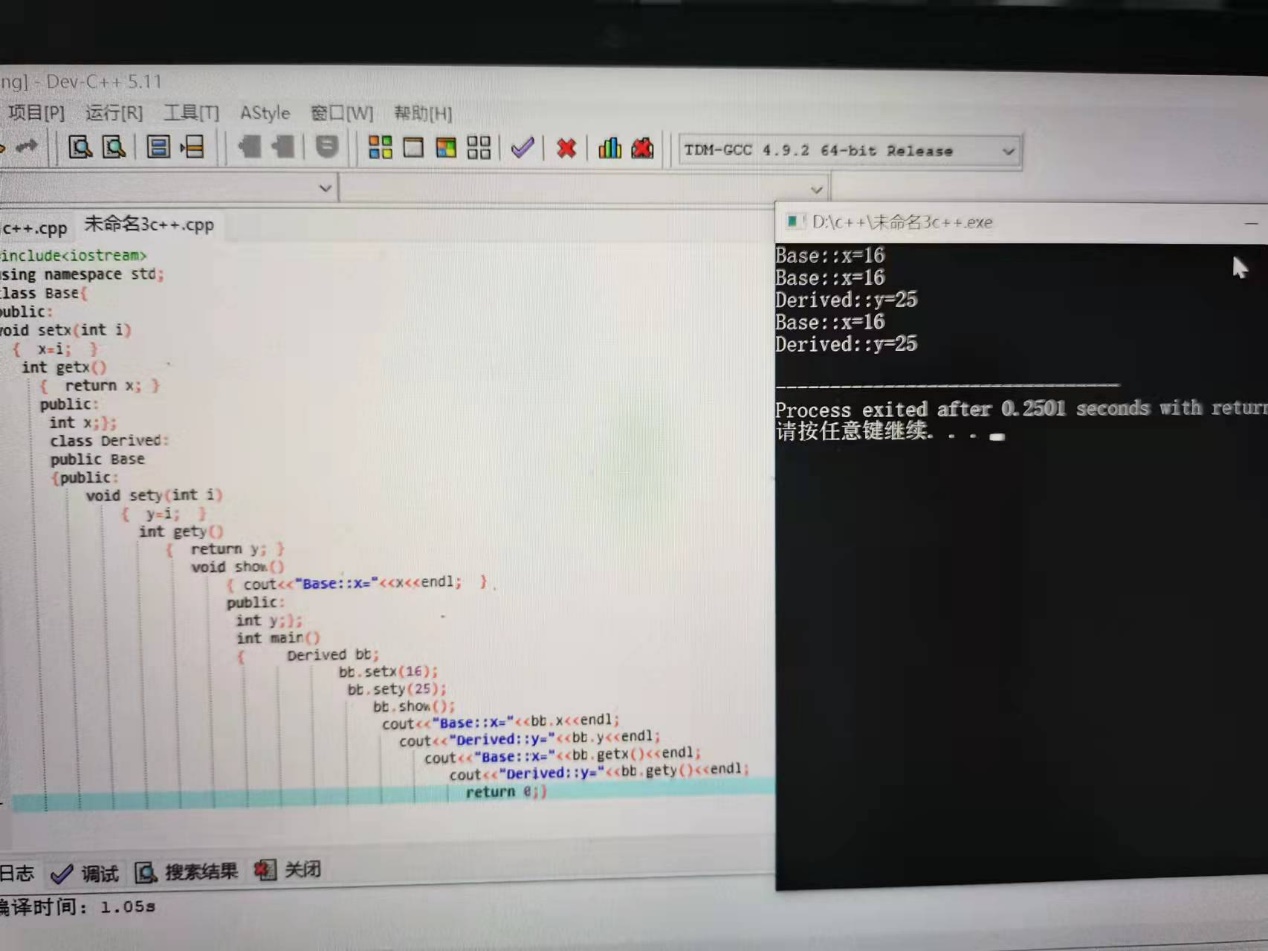
cout << "调用函数sqr\_it后, 实参对象obj的数据成员i的值为:";

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]);

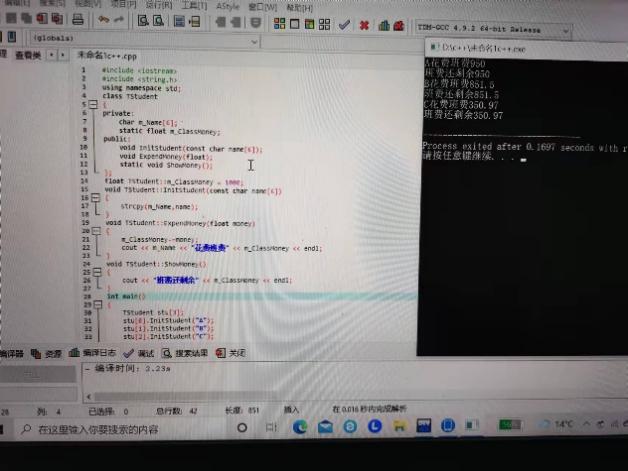
s3.expendmoney(b[2]);

s1.showmoney();

return 0;

}

图片：



心得：数据有了，我们就要进行数据的导入与保存，这时我们通常会采用‌重载输入输出，这是很严格的一个过程，必须保证数据与重载完全对应起来，否则可能无法导入数据，那这个程序就宣布作废了，一个没有数据、无法对数据进行操作的程序是没有意义的。很多时候重载会给我们带来很多便利，比如我们对时间类进行重载，作出时间差或者根据特定的值去推算时间等。但是又不得不说重载还是尽量少用一些， 因为有的时候万一错了可能会导致大的问题