# 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;

}

其次

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;

}

最后

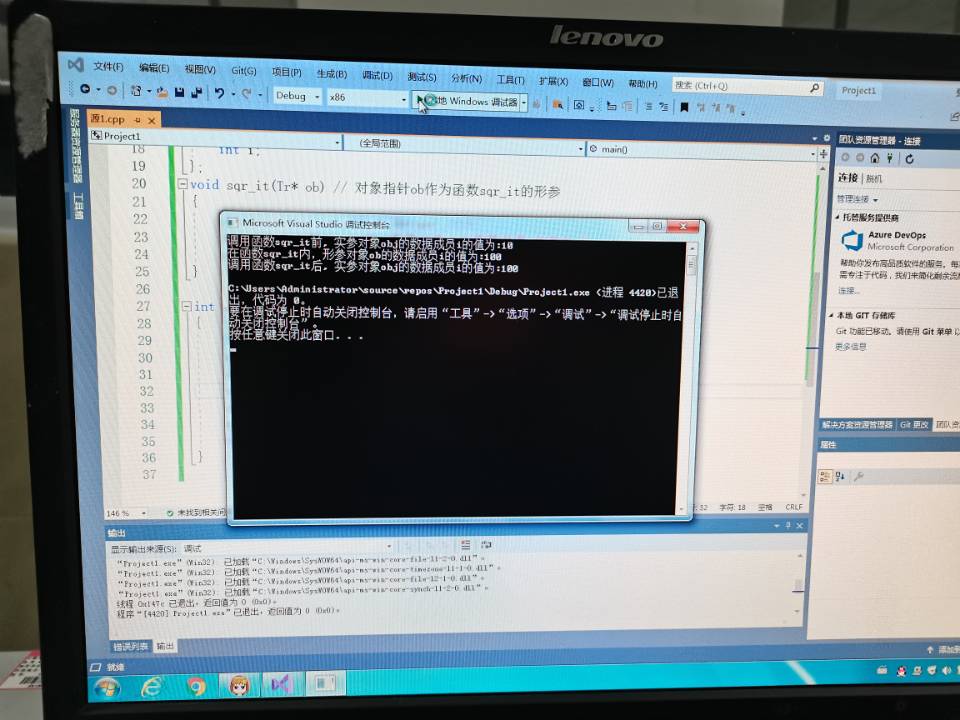
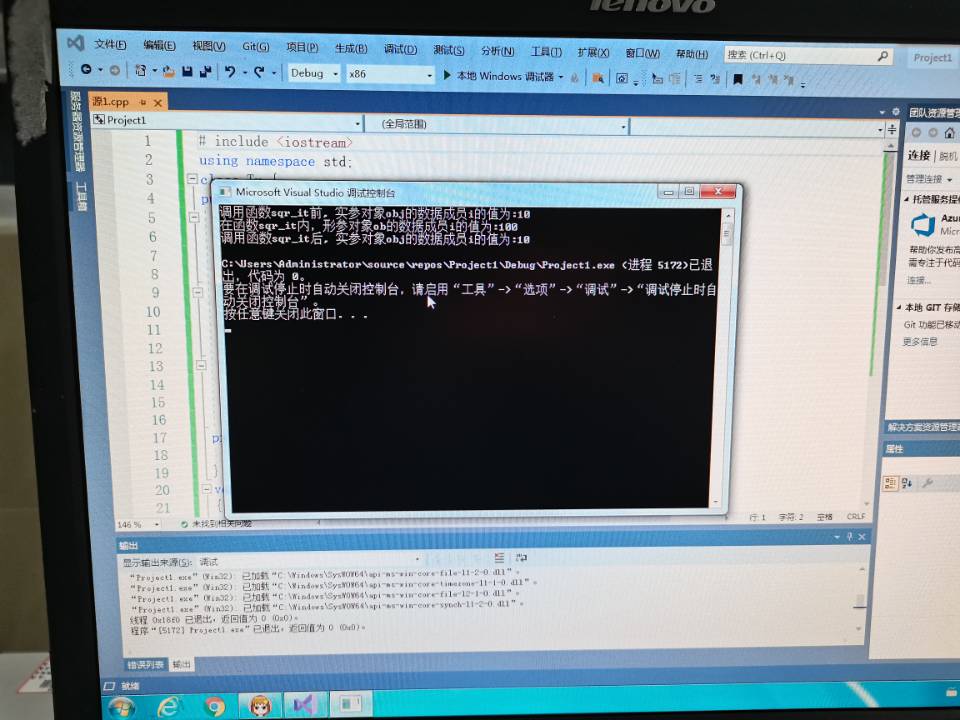
void sqr\_it(Tr& ob)

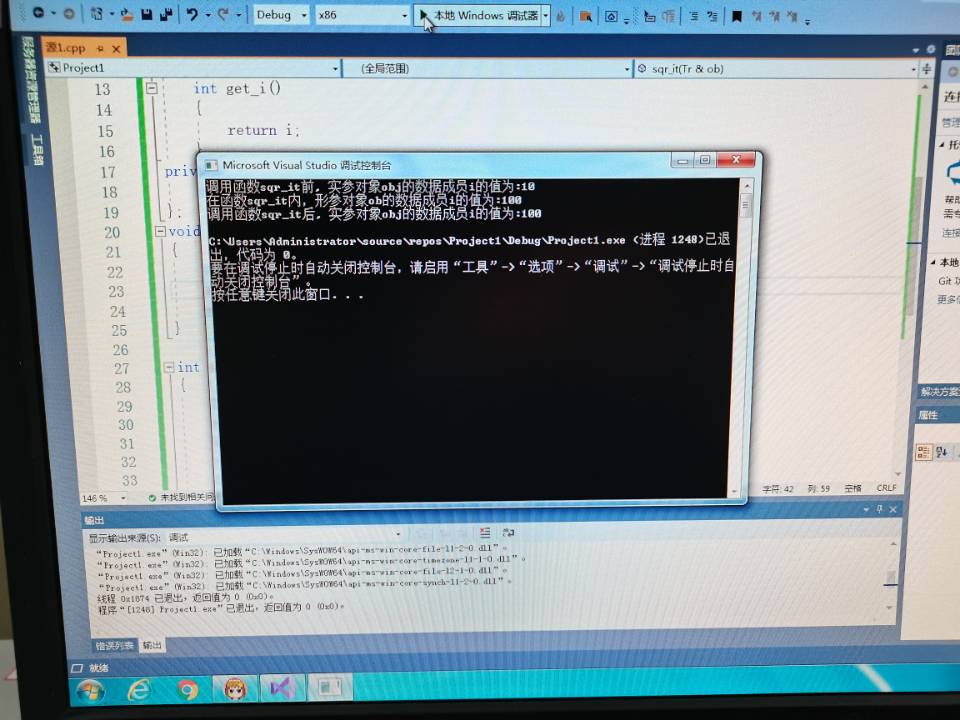
{ ob.set\_i(ob.get\_i()\*ob.get\_i());

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;

}





用静态数据成员计算由同一类建立的对象的数量

# include<iostream>

using namespace std;

class Ctest{

static int count; //私有成员

public:

Ctest() {

++count;cout<<"对象数量="<<count<<'\n';}

};

int Ctest::count=0; //A，对静态数据定义性说明

int main(void){

Ctest a[3];

return 0;

}

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。

#include "iostream.h"

#include <string.h>

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(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(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"<<m\_ClassMoney<<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();

}

上机心得：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 IMG_2562021-2099 wangzezheng. All rights reserved