第四次上机作业：对象传递与静态成员

#include<iostream>

using namespace std;

class Tr

{

public:

Tr(int n)

{i=n;}

void set\_(int n)

{i=n;}

int get\_i()

{return i;}

private:

int i;

};

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;

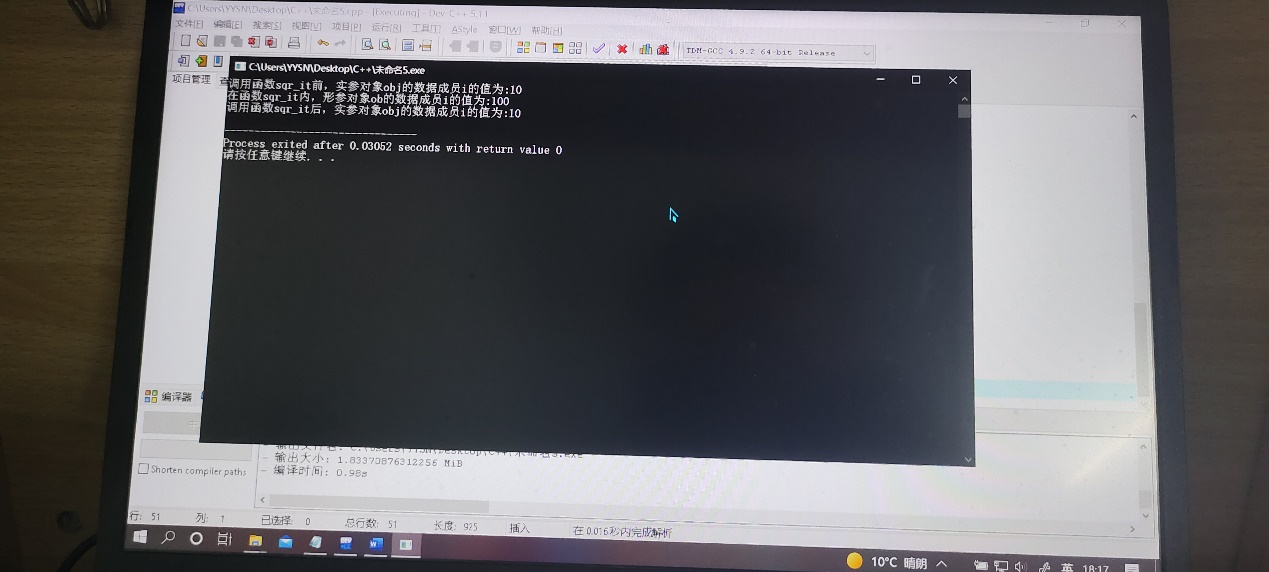
sqr\_it(obj);

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;

}



如果一个类包含了数据成员和成员函数，就要分别为数据和函数的代码分配存储空间。类是对具有相同属性和相同行为的一组对象的描述。例如,学生类可由学号、姓名、成绩等表示其属性的数据项和对这些数据的录人、修改和显示等操作组成。在C++语言中把类中数据称为数据成员。对象是类的一个实例,每个对象具有自己的数据成员。

#include<iostream>

using namespace std;

class Ctest

{

static int count;

public:

Ctest()

{++count;cout<<"对象数量="<<count<<'\n';}

};

int Ctest::count=0;

int main(void)

{

Ctest a[3];

return 0;

}

#include"iostream.h"

#include<string.h>

class TStudent

{

private:

char m\_Name[6];

static float m\_ClassMoney;

public:

void InitStudent(char\*);

void ExpendMoney(float);

static void ShowMoney();

};

float TStudent::m\_ClassMoney=1000;

void TStudent::InitStudent(char name[])

{strcpy(m\_Name,name);}

void TStudent::ExpendMoney(float money)

{

m\_ClassMoney=money;

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];

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

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

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

stu[0].ExpendMoney(50);

TStudent::ShowMoney();

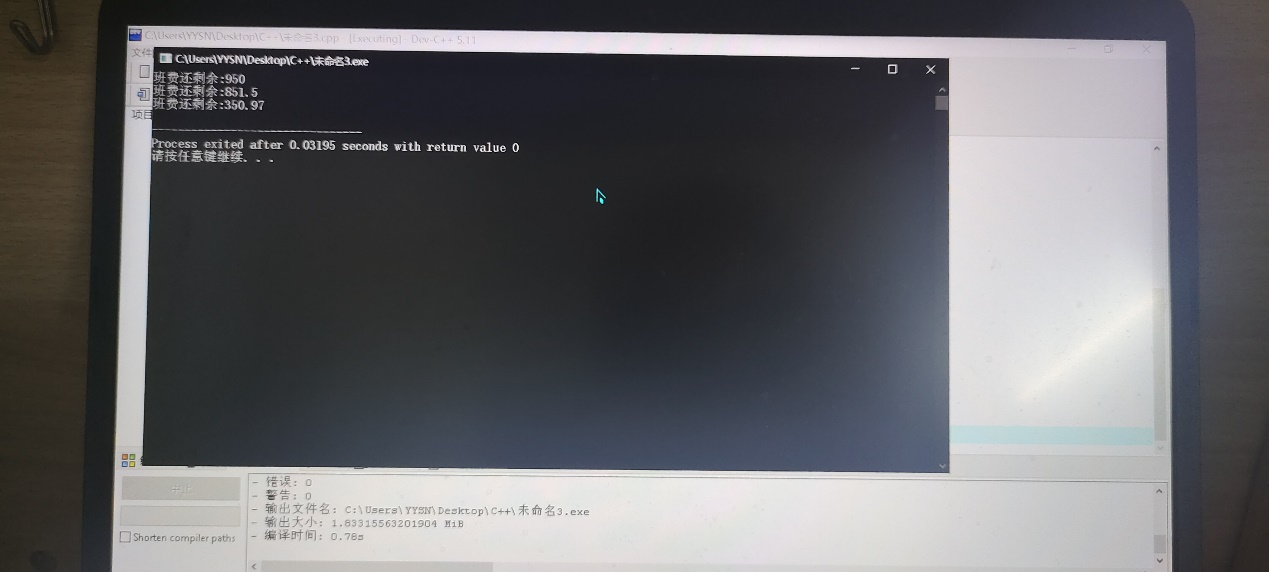
stu[1].ExpendMoney(98.5);

TStudent::ShowMoney();

stu[2].ExpendMoney(500.53);

TStudent::ShowMoney();

}



准确理解静态成员内容，才能更好的利用。