1. 程序代码

#include <iostream>

#include <string>

using namespace std;

static float classmoney;

static int i = 1;

static string name[100];

class student {

public:

student()

{

classmoney = 1000;

}

static void initstudent()

{

cin >> name[i];

//name[i] = n;

++i;

}

void showmoney()

{

cout << "目前余额为" << classmoney << endl;

}

void expendmoney(float n)

{

classmoney = classmoney - n;

cout <<name[i] <<"花费了" << n << endl;

}

};

int main()

{

student A, B, C;

A.initstudent();

A.expendmoney(50);

A.showmoney();

B.initstudent();

B.expendmoney(98.5);

B.showmoney();

C.initstudent();

C.expendmoney(500.53);

C.showmoney();

return 0;

}

1. 实验结果



1. 实验感想

（1）在进行对象传递时，可以使用对象作为函数参数，使用对象指针作为函数参数，使用对象引用作为函数参数，这三种方法，它们各有不同的特点。其中，使用对象作为函数参数时，对象的值在调用函数后不改变，使用对象指针和对象引用作为函数参数时，可以修改对象本身的值，使用时可根据不同的程序功能来选用。

（2）在一个类中，将一个数据成员说明为 static，这种成员称为静态数据成员。无论建立多少个类的对象，都只有一个静态数据成员的拷贝。从而实现了同一个类的不同对象之间的数据共享。每次对对象变量 操作，其值都会在原有的基础上发生改变。所以能够达到跟随性的目的。

（3）当程序运行出错后，可采用多种方法进行修改，使用时，明白其错误原因，应用最合适的方法。