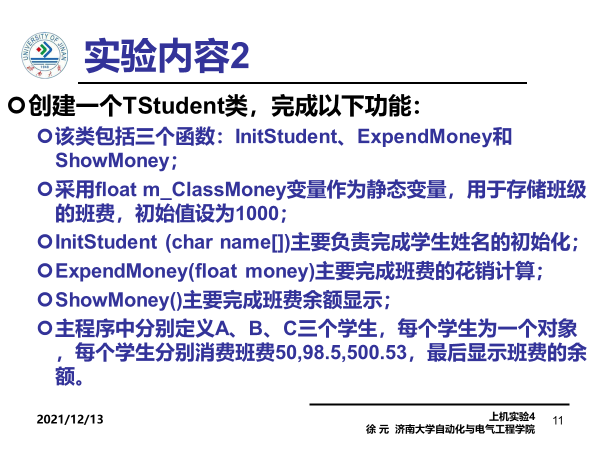
Copyright ©2021-2099 ChenJiacheng. All rights reserved

**实验要求：**

****

**程序代码：**

#include <iostream>

using namespace std;

static float m\_classMoney = 1000;

class TStudent

{

char\* name;

public:

void InitStudent(char name[])

{

name = name;

}

void ExpendMoney(float money, char name[]);

void showMoney();

};

void TStudent::ExpendMoney(float money, char name[])

{

m\_classMoney -= money;

cout << name << "花费班费" << money << "元" << endl;

}

void TStudent::showMoney()

{

cout << "班费还剩余" << m\_classMoney <<"\n" << endl;

}

int main()

{

TStudent stu[3];

char name[3][100];

int i;

double money[3] = { 50,98.5,500.53 };

for (i = 0; i < 3; i++)

{

cout << "请输入姓名:";

cin >> name[i];

stu[i].InitStudent(name[i]);

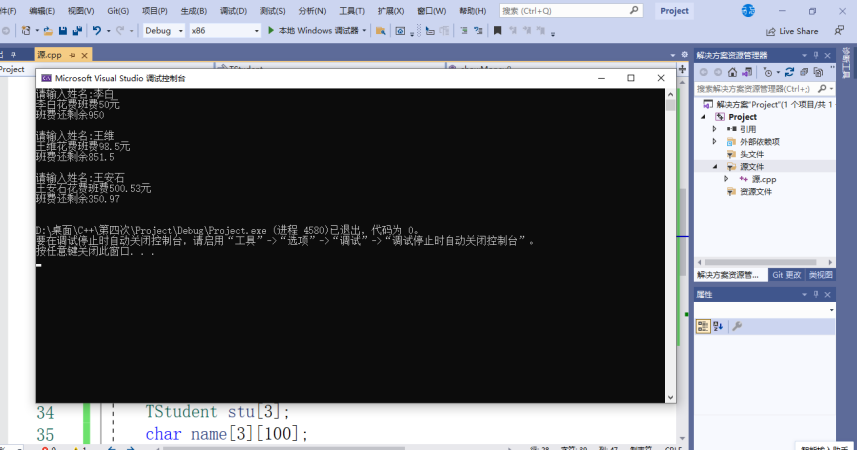
stu[i].ExpendMoney(money[i],name[i]);

stu[i].showMoney();

}

}

程序结果：



感想心得：记住了一个重要概念静态变量属于类而不是某一个对象。

静态数据成员是类的成员，而不是对象的成员，所有该类对象都共用该数据成员，可以实现同类对象之间进行数据共享。静态数据成员是静态存储的，必须对它进行初始化。

1.值传递：有一个形参向函数所属的栈拷贝数据的过程，如果值传递的对象是类对象或是大的结构体对象，将耗费一定的时间和空间。  
  
2.指针传递：同样有一个形参向函数所属的栈拷贝数据的过程，但拷贝的数据是一个固定为4字节的地址。  
  
3.引用传递：同样有上述的数据拷贝过程，但其是针对地址的，相当于为该数据所在的地址起了一个别名。  
  
效率上讲，指针传递和引用传递比值传递效率高。一般主张使用引用传递，代码逻辑上更加紧凑、清晰。