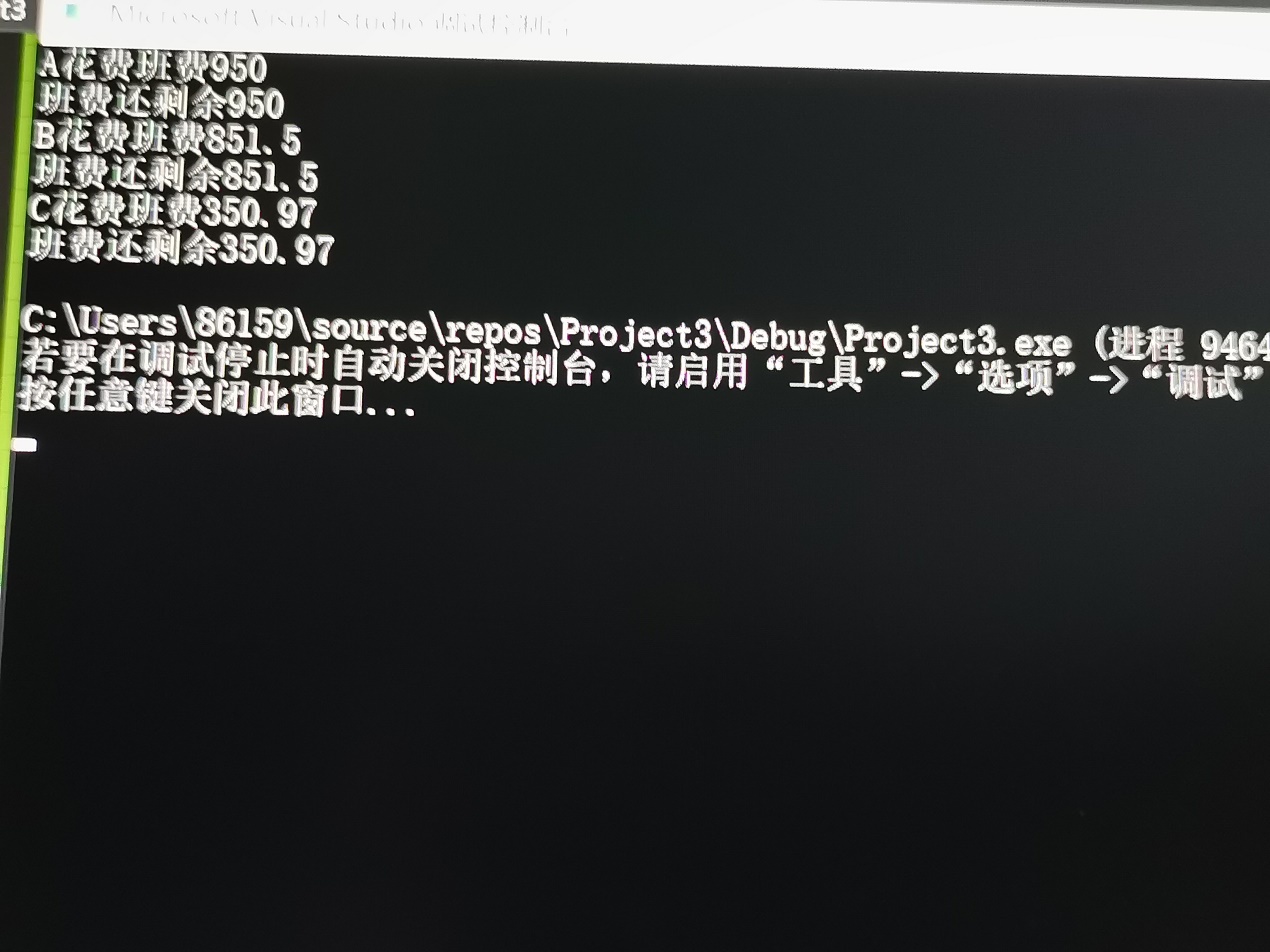
第四次实验报告：对象传递与静态成员

1. 程序代码
2. #include "iostream.h"
3. #include <string.h>
4. class TStudent
5. {
6. private:
7. char m\_Name[6];
8. static float m\_ClassMoney;
9. //m\_ClassMoney为静态成员变量，保存班费
10. public:
11. void InitStudent(char \*);
12. void ExpendMoney(float);
13. static void ShowMoney(); //静态成员函数
14. };
15. float TStudent::m\_ClassMoney=1000; //静态成员变量的初始化
16. void TStudent::InitStudent(char name[])
17. {
18. strcpy(m\_Name,name);
19. }
20. void TStudent::ExpendMoney(float money)
21. {
22. m\_ClassMoney-=money; //班费为原先的减去花费的
23. cout<<m\_Name<<"花费班费"<<m\_ClassMoney<<endl;
24. }
25. void TStudent::ShowMoney()
26. {
27. cout<<"班费还剩余"<< m\_ClassMoney<<endl;
28. }
29. void main()
30. {
31. TStudent stu[3];//定义三个学生
32. stu[0].InitStudent("A");
33. stu[1].InitStudent("B");
34. stu[2].InitStudent("C");
35. stu[0].ExpendMoney(50);
36. TStudent::ShowMoney();
37. stu[1].ExpendMoney(98.5);
38. TStudent::ShowMoney();
39. stu[2].ExpendMoney(500.53);
40. TStudent::ShowMoney();
41. }

2.程序结果

3.心得体会

学习本次内容，最重要的是搞清楚静态成员变量。静态成员变量最大的作用是可以实现在同类中的多个对象之间实现数据共享。但是静态成员并不属于某一个对象，而是在所有对象之外单独开辟空间，在一个类中可以有一个或者多个静态数据成员，所有对象共享这些静态成员，并且都可以引用。但需要注意的是，静态数据成员只能在类外初始化，这是我做实验时所忽略的一点，通过这次实验，我更好的理解明白了静态数据成员的用法。