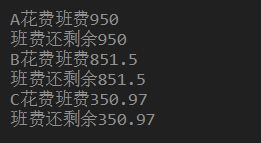
实验内容：#include <iostream>#include <string>using namespace std;class TStudent{private: string Name; static float ClassMoney;public: void InitStudent(string); void ExpendMoney(float); static void ShowMoney(); //静态成员函数};float TStudent::ClassMoney=1000;void TStudent::InitStudent(string name){ Name=name;}void TStudent::ExpendMoney(float money){ ClassMoney-=money; //班费为原先的减去花费的 cout<<Name<<"花费班费"<<ClassMoney<<endl;} void TStudent::ShowMoney() { cout<<"班费还剩余"<< ClassMoney<<endl; }int 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(); return 0;}程序结果：



感想

通过part c的题目练习，感受到了静态函数的优势，会被自动分配在一个一直使用的存储区，直到退出应用程序实例，避免了调用函数时压栈出栈，速度快很多。