友元函数能够直接访问类的所有成员，包括private成员。

如何实现一个友元函数？

1，在头文件，类内部声明friend void f(class & ,……);

2，在头文件，类的外部再次声明void f(class &,…… );

3，在类实现cpp，实现void f(class &,…… ) { }

Account.h

#include<iostream>

usingnamespacestd;

classAccount{

friendvoidhandSetBalance(intbalance,Account&account);

public:

Account(intbalance){

this->balance=balance;

}

voidprintBalance()const{

cout<<balance<<endl;

}

private:

intbalance;

};

voidhandSetBalance(intbalance,Account&account);

Account.cpp

#include"Account.h"

voidhandSetBalance(intbalance,Account&account){

account.balance=balance;

account.printBalance();

}

main.cpp

intmain(){

Accountaccount(0);

handSetBalance(100,account);

system("pause");

}

友元类

声明一个类A是B类的友元，则类A的所有成员函数能够直接操作类B的数据成员，包括private。

步骤：

1. 类B接口中 friend class A;
2. 实现类B的接口
3. 实现类A的接口

---------------------------------------------------------------------------------------

也可以只把类A的某个成员函数声明成类B的友元函数。