



```
//-----使用这些类-----
public class BankDemo {
    public static void Main(string [] args)
    {
        Bank bank = new Bank();
        bank.OpenAccount("2222", "2222", 20);
        bank.OpenAccount("3333", "3333", 50);
        ATM atm = new ATM(bank);

        for( int i=0; i<5; i++)
        {
            atm.Transaction();
        }
    }
}
```

```
//-----账号类-----
```

```
public class Account {

    double money; //decimal money;
```

```
string id;
string pwd;
//string name;

public Account( string id, string pwd, double money )
{
    //if( money < 0 ) throw new Exception("....");
    this.id = id;
    this.pwd = pwd;
    this.money = money;
}

public double getMoney()
{
    return money;
}

public void setMoney(double val)
{
    this.money = val;
}

public string getId()
{
    return id;
}

public void setId(string id)
{
    this.id = id;
}

public string getpwd()
{
    return pwd;
}

public void setPwd(string pwd)
{
    this.pwd = pwd;
}

public bool SaveMoney( double money)
{

```

```

        if( money < 0 ) return false; //卫语句

        this.money += money;
        return true;
    }

    public bool WithdrawMoney( double money)
    {
        if( this.money >= money )
        {
            this.money -= money;
            return true;
        }

        return false;
    }

    public bool IsMatch( string id, string pwd )
    {
        return id==this.id && pwd==this.pwd;
    }
}

//-----银行类-----

using System;
using System.Collections;
using System.Collections.Generic;

public class Bank {

    List<Account> accounts = new List<Account>();

    public Account OpenAccount(string id, string pwd,
double money)
    {
        Account account = new Account(id, pwd, money);
        accounts.Add( account );
    }
}

```

```

        return account;
    }

    public bool CloseAccount( Account account)
    {
        int idx = accounts.IndexOf(account);
        if( idx<0 )return false;
        accounts.Remove(account);
        return true;
    }

    public Account FindAccount(string id, string pwd)
    {
        foreach(Account account in accounts)
        {
            if( account.IsMatch(id, pwd))
            {
                return account;
            }
        }
        return null;
    }
}

//-----ATM-----
using System;
using System.Collections;

public class ATM {

    Bank bank;
    public ATM( Bank bank)
    {
        this.bank = bank;
    }

    public void Transaction()
    {
        Show("please insert your card");
        string id = GetInput();

        Show("please enter your password");
    }
}

```

```

string pwd = GetInput();

Account account = bank.FindAccount(id, pwd);

if( account == null)
{
    Show("card invalid or password not corrent");
    return;
}

Show("1: display; 2: save; 3: withdraw");
string op = GetInput();
if (op == "1")
{
    Show( "balance: " + account.getMoney() );
}
else if( op=="2")
{
    Show("save money");
    string smoney = GetInput();
    double money = double.Parse(smoney);

    bool ok = account.SaveMoney(money);
    if( ok ) Show("ok");
    else Show("eeer");

    Show("balance: " + account.getMoney());
}
else if( op=="3" )
{
    Show("withdraw money");
    string smoney = GetInput();
    double money = double.Parse(smoney);

    bool ok = account.WithdrawMoney(money);
    if( ok ) Show("ok");
    else Show("eeer");

    Show("balance: " + account.getMoney());
}
}

```

```
public void Show(string msg)
{
    Console.WriteLine(msg);
}
public string GetInput()
{
    return Console.ReadLine(); // 输入字符
}
}
```