调用：

public static void TestAccount()

{

Account a1 = new Account();

a1.Money = 2000;

a1.AccountNo = "24234324324324";

float m1 = a1.Deposit(30000.00f); // 存30000， 32000;

Account a2 = new Account("3656455465", 50000.00f);

float m2 = a2.Withdraw(43000.00f); // 取43000， 7000

}

namespace NsDemo03

{

// 描述银行账户的类

public class Account

{

public string AccountNo; //对象字段，代表账户的账号

public float Money; //对象字段，代表账户的余额

// 系统提供默认无参构造方法，如果没有其他构造方法，可以不用写

public Account()

{

}

public Account(string accountNo, float amount)

{

AccountNo = accountNo;

Money = amount;

}

public Account(string accountNo)

{

AccountNo = accountNo;

}

/// <summary>

/// 存款

/// </summary>

/// <param name="amount">存款金额</param>

/// <returns>返回存款后的最新余额</returns>

public float Deposit(float amount)

{

Money += amount;

return Money;

}

/// <summary>

/// 取款

/// </summary>

/// <param name="amount">取款金额</param>

/// <returns>返回取款后的最新余额</returns>

public float Withdraw(float amount)

{

Money -= amount;

return Money;

}

/// <summary>

/// 查询余额，想想查询方法为什么不需要账号参数

/// </summary>

/// <returns>返回账户的当前余额</returns>

public float QuerySum()

{

return Money;

}

/// <summary>

/// 转账

/// </summary>

/// <param name="toAcccountNo">转入账号</param>

/// <param name="amount">转账金额</param>

/// <returns>转账后的余额</returns>

public float Transfer(string toAcccountNo, float amount)

{

// 根据转入账号创建一个账户对象

Account toAccount = new Account(toAcccountNo);

Withdraw(amount); // 自己账户上取走指定的金额

toAccount.Deposit(amount); // 向对方账户上存入指定的金额

return Money;

}

}

}