



# 对象间的协作与信息交换（上）

---

北京理工大学计算机学院  
金旭亮

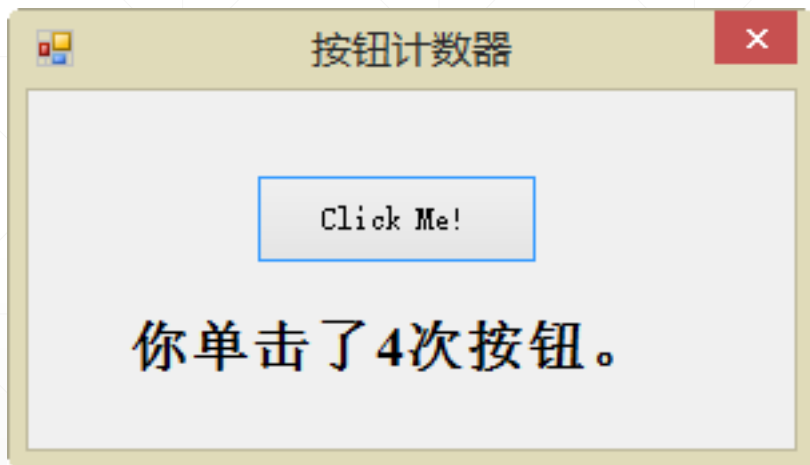
# 概述

对象协作的核心就是对象间信息的交换，体现为对象之间的相互访问，主要通过以下两种方式实现：

- 1 相互存取字段/属性
- 2 相互调用方法

# 同一对象内部的信息交换

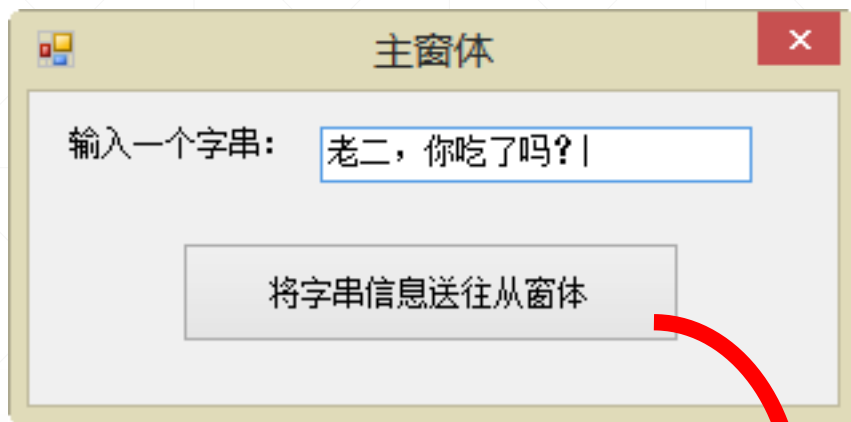
## 技术要点：



示例：ButtonCounterInSingleForm

- “点击次数”这个信息，在窗体这个对象的内部流动，最终被标签对象所接收。
- 信息在同一个对象内的流动，主要通过类中的方法直接访问类中的字段实现的。
- 用户鼠标的“单击”驱动这个信息的流动过程

# “主→从”对象间信息传送

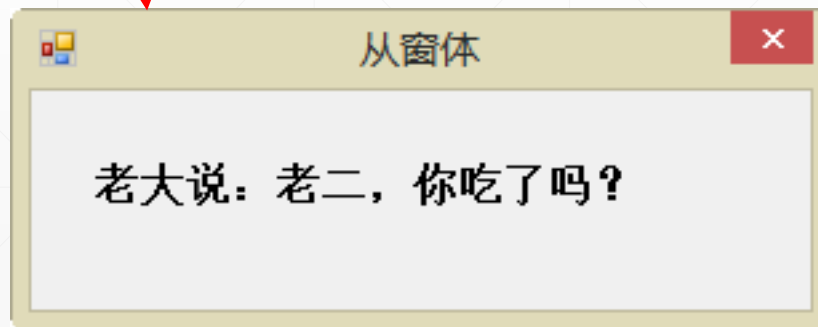


主窗体

输入一个字符串: 老二，你吃了吗？|

将字符串信息送往从窗体

主窗体通过从窗体对象的引用，调用其公有方法/属性传送信息。



从窗体

老大说：老二，你吃了吗？

示例：MainFormToOtherForm

# “从→主”对象信息传送（1）



主窗体必须知道从窗体关闭的原因，从而进行不同的处理工作

在这个示例中，主窗体是“主动”的，它主动地查询从窗体对象的状态并启动不同的处理流程。

# “从→主”对象信息传送（2）

可以让从窗体“主动”地向主窗体“汇报工作” .....

参看示例OtherFormToMainForm2

这个示例展示了极为重要的编程技巧：

- 1 **对象注入**：主窗体使用this把自己的引用传给从窗体
- 2 **“回调”的编程方法**：从窗体回调主窗体的公有方法

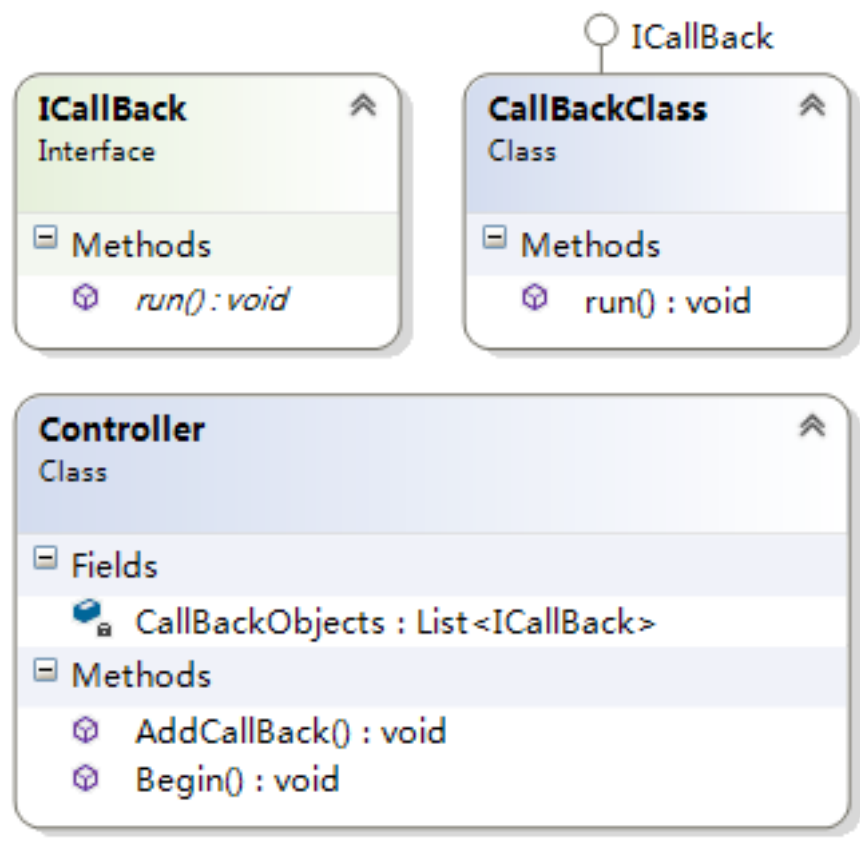
# 理解“回调（call back）”

通常情况下，A对象new一个对象B，然后会立即调用它的方法，这个过程中：**A是“主动”，B是“被动”的。**

然而，如果A对象在new了一个对象B之后，并不去调用B中的方法，而是告诉它：“**在XXX条件下，你可以调用我的YYY方法**”，之后A可以休息了，不再关心这件事。

在合适的时候，B调用A所告诉它的方法，这就叫“**回调（call back）**”。在这个过程中，**A是“被动”的，B是“主动”的。**

# 回调应用实例



Controller内部包容着一个ICallback对象的集合，Begin()方法启动对这个集合中对象的遍历操作



# CallbackExample程序为什么用接口实现？

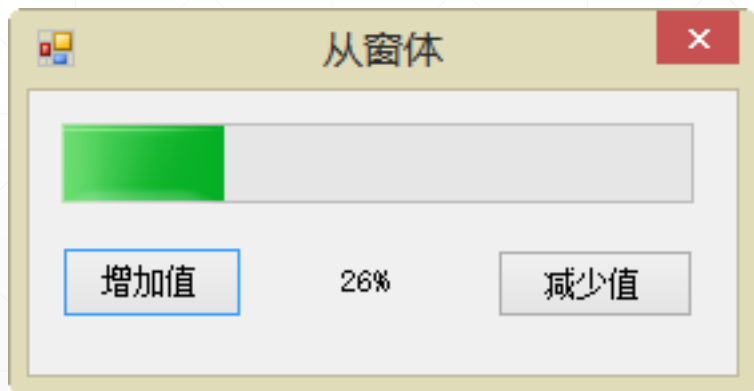
主要好处：

能让程序易于扩充。

程序扩充功能步骤：

新写一个类实现ICallBack接口，然后将其加入到Controller对象的内部集合中即可，其他代码不用作任何修改！

# 作业：两个对象间信息的双向传送



示例的特点：

两个窗体之间实现了即时的双向互动和同步。

基本实现方式：

两个对象之间互相执有对方的引用。

示例：TwoWayCommunicationInForms