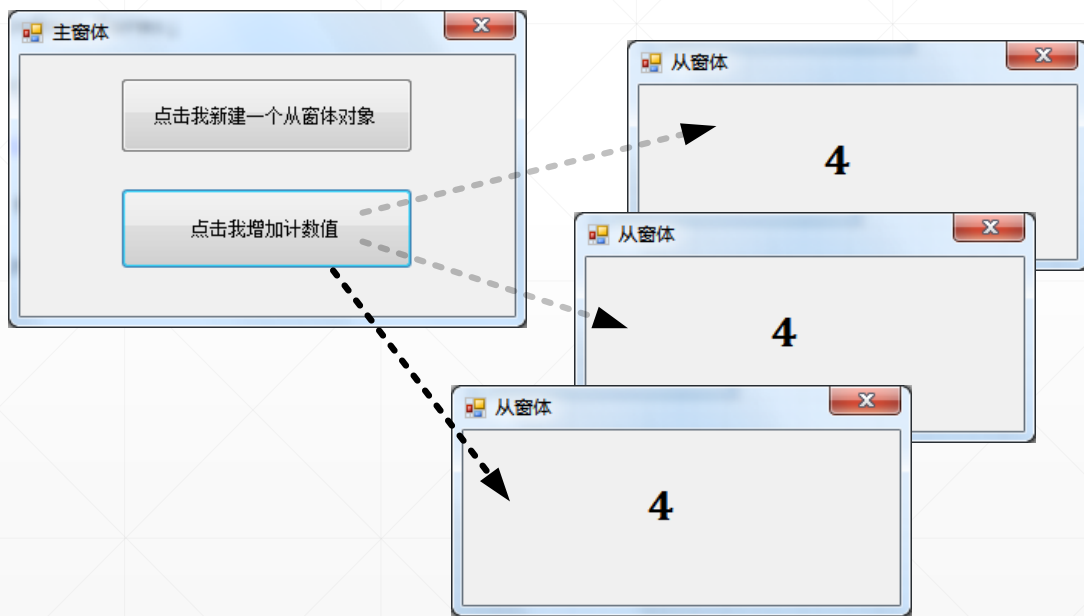


对象间的协作与信息交换（下）

北京理工大学计算机学院
金旭亮

在对象间“广播信息”

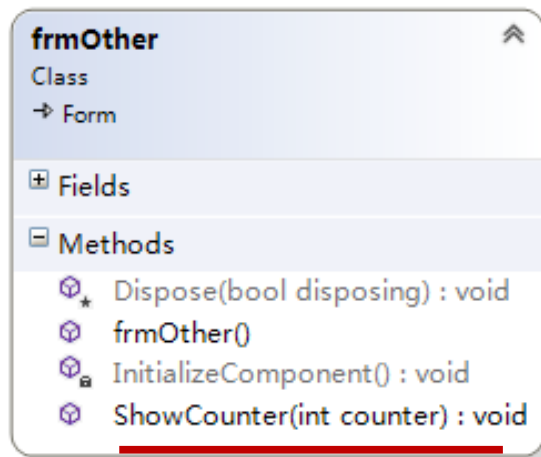
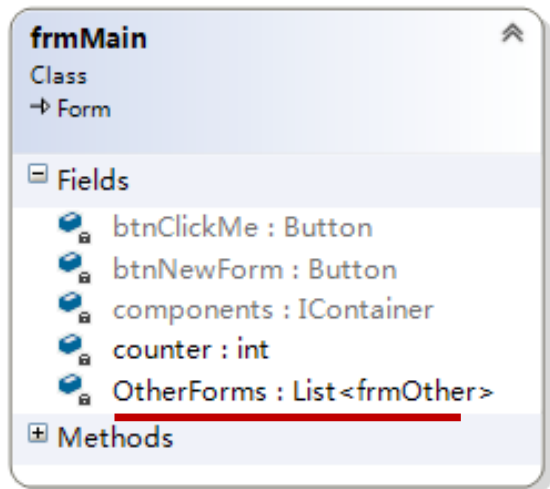
要实现的程序功能



特点:

信息从一个对象发出，多个对象接收。

使用对象集合实现“消息广播”

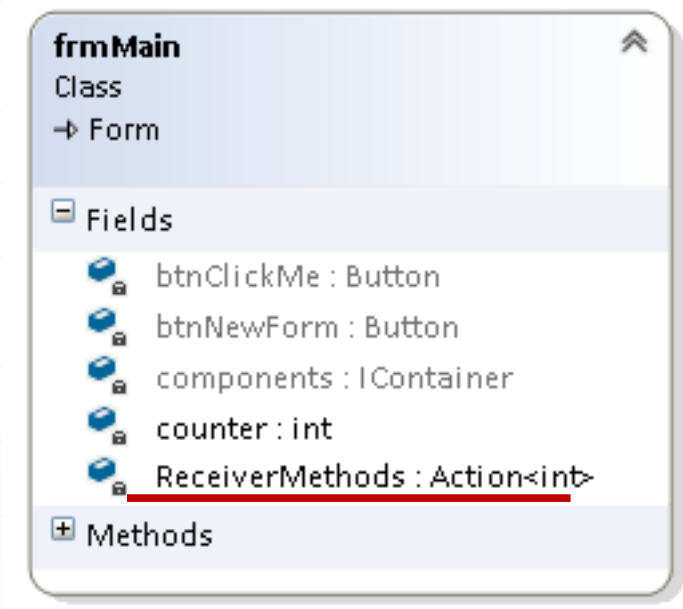


- 主窗体中使用一个对象集合保存所有已创建从窗体对象的引用
- 当点击按钮时，**遍历**这个对象集合，逐个调用从窗体对象的公有方法显示信息

```
private int counter = 0; //计数器
1 reference
private void btnClickMe_Click(object sender, EventArgs e)
{
    counter++;
    foreach (frmOther frm in OtherForms)
        frm.ShowCounter(counter);
}
```

- 这种方法是一种“通用”的方法，适用于各种面向对象编程语言。

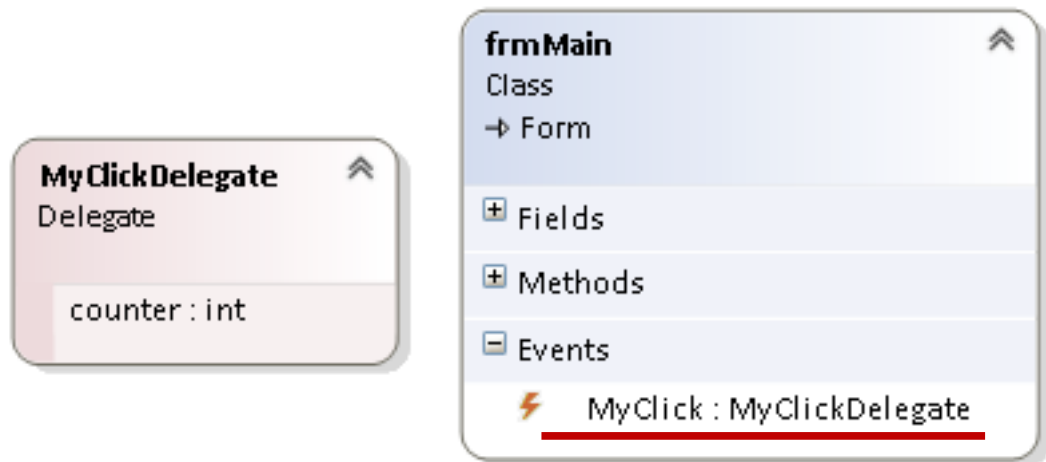
利用委托实现“消息广播”



示例项目 UseManyToOneDelegate

- C#支持一种称为“**委托 (Delegate)**”的特性，一个委托变量包容一个方法引用列表，能“包容”多个方法，调用一次委托变量，所有方法顺序执行。
- 主窗体定义一个ReceiverMethods委托变量，在创建从窗体对象时，将从窗体的公有方法ShowCounter“挂接”到这个字段上
- 主窗体点击按钮时，只需调用一次ReceiverMethods，所有从窗体都能更新显示
- 这本质上是主窗体“**回调**”从窗体的方法。

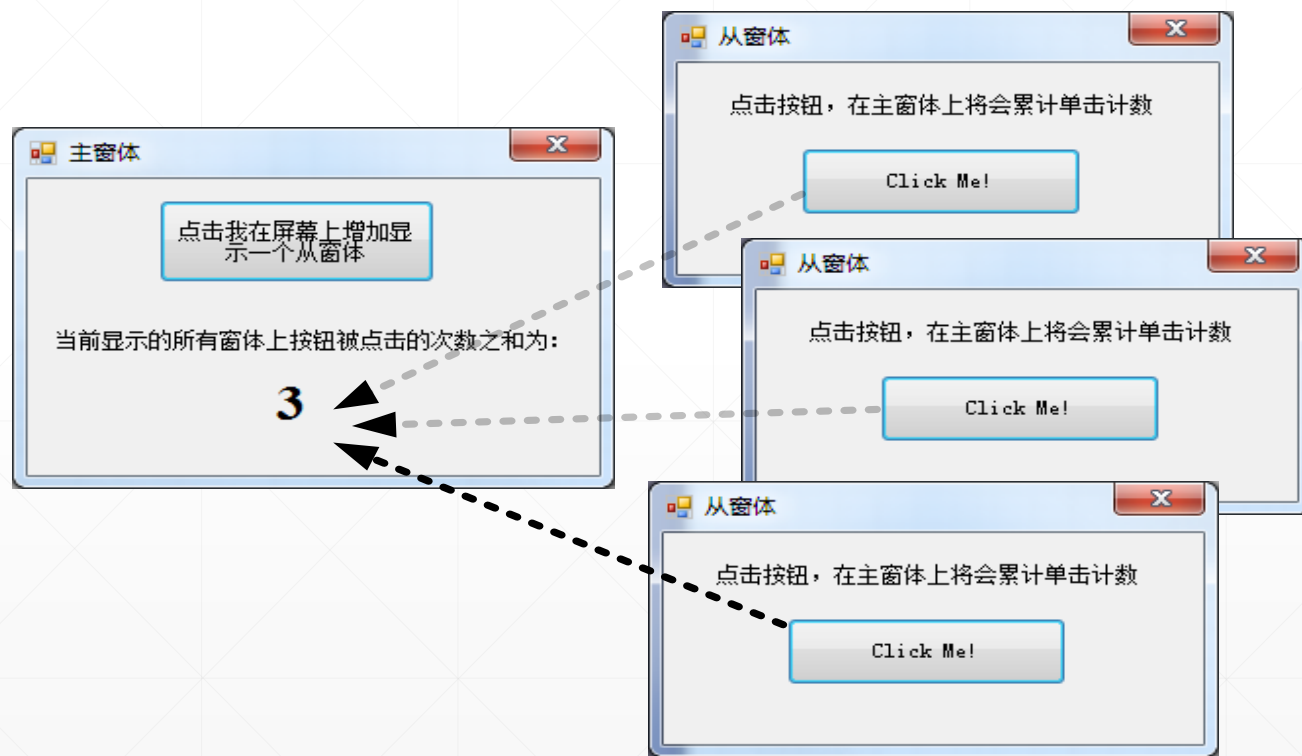
利用事件实现“消息广播”



示例项目 UseEvent

- C#可以很方便地自定义事件。
- 我们为主窗体定义一个MyClick事件，当点击按钮时，触发这个事件。
- 从窗体的ShowCounter () 方法响应这个事件，当事件触发时，刷新显示
- 这里我们实际上构建了一个“**事件驱动**”的面向对象软件系统。

一个对象“监控”多个对象

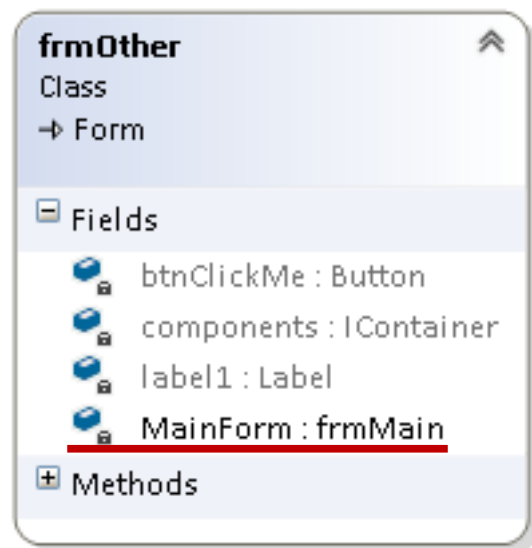


特点:

多个对象向一个对象发送信息

实现“对象监控”：利用对象引用回调

示例 ButtonCounterForMultiFormUseReference

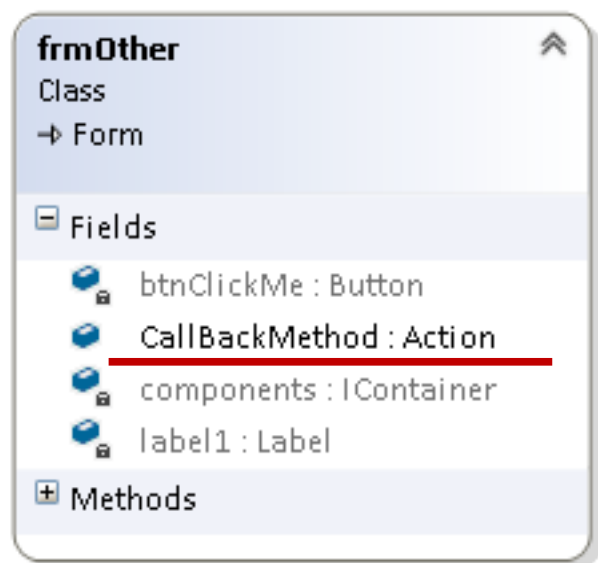


- 从窗体定义有一个MainForm字段，它引用主窗体对象
- 当从窗体点击按钮时，它通过对象引用调用主窗体的方法向其“主动汇报情况”

```
1 reference
private void btnClickMe_Click(object sender, EventArgs e)
{
    //调用主窗体的公有方法，显示按钮计数
    if (MainForm != null)
        MainForm.ShowCounter();
}
```

实现“对象监控”：利用委托回调

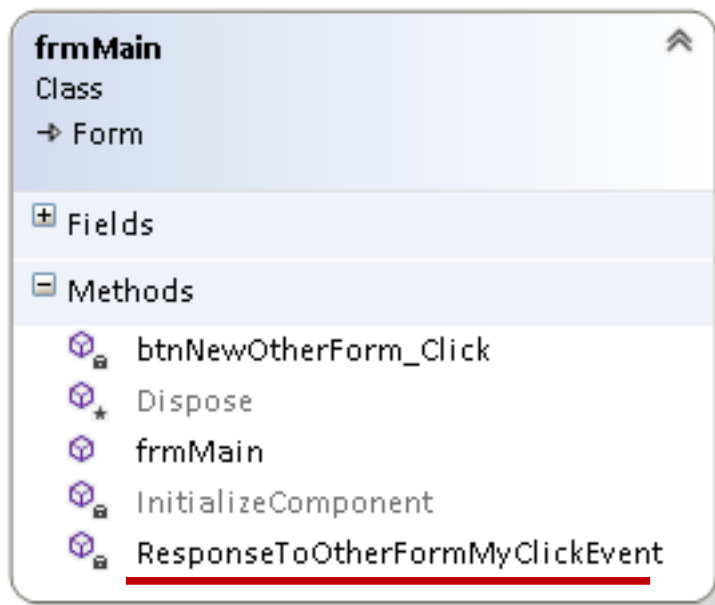
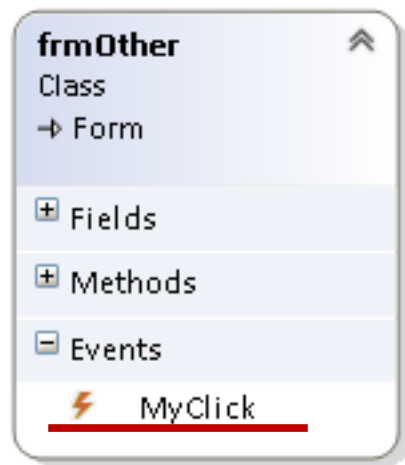
示例：ButtonCounterForMultiFormUseDelegate



- 从窗体定义有一个委托类型的字段
CallBackMethod
- 主窗体对象在创建从窗体对象时，把它自己的公有方法ShowCounter挂接到此字段上。
- 当从窗体点击按钮时，它通过委托回调主窗体的方法

```
frmOther frm = new frmOther();  
//将主窗体的方法“挂接”到从窗体对象上  
frm.CallBackMethod = this.ShowCounter;  
//在屏幕上显示从窗体  
frm.Show();
```


实现“对象监控”：自定义事件



- 从窗体定义了一个MyClick事件
 - 主窗体的ResponseTo...()方法响应这个事件，在事件响应代码中累加计数并更新显示
-
- 这种方法，实际上是一个对象响应多个对象触发的事件，是另一种“**事件驱动**”编程类型。

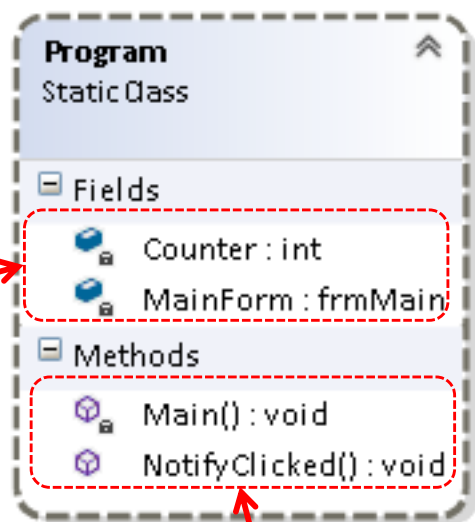
```
frmOther frm = new frmOther();  
//挂接事件响应函数  
frm.MyClick += this.ResponseToOtherFormMyClickEvent;  
frm.Show();
```

示例：ButtonCounterForMultiFormUseEvent

实现“对象监控”：全局静态字段和方法

示例：ButtonCounterForMultiFormUseStaticMethod

静态
字段



静态方法

- 在Program.cs中定义一个静态的字段引用主窗体，再定义一个静态NotifyClicked方法累加计数值并更新主窗体显示
 - 从窗体在单击按钮时，调用NotifyClicked方法。
-
- 这种方法构建了“全局”的字段和方法，不推荐使用

巩固练习

请编写一个程序，运行时可以在屏幕上显示多个从窗体，主窗体可以动态设置所有从窗体的背景色，即：

主窗体上使用相应控件生成或选择一种颜色，点击“确定”（或采用其他方式），当前所有显示的从窗体其背景色都变成这个颜色。

你能否“举一反三”？

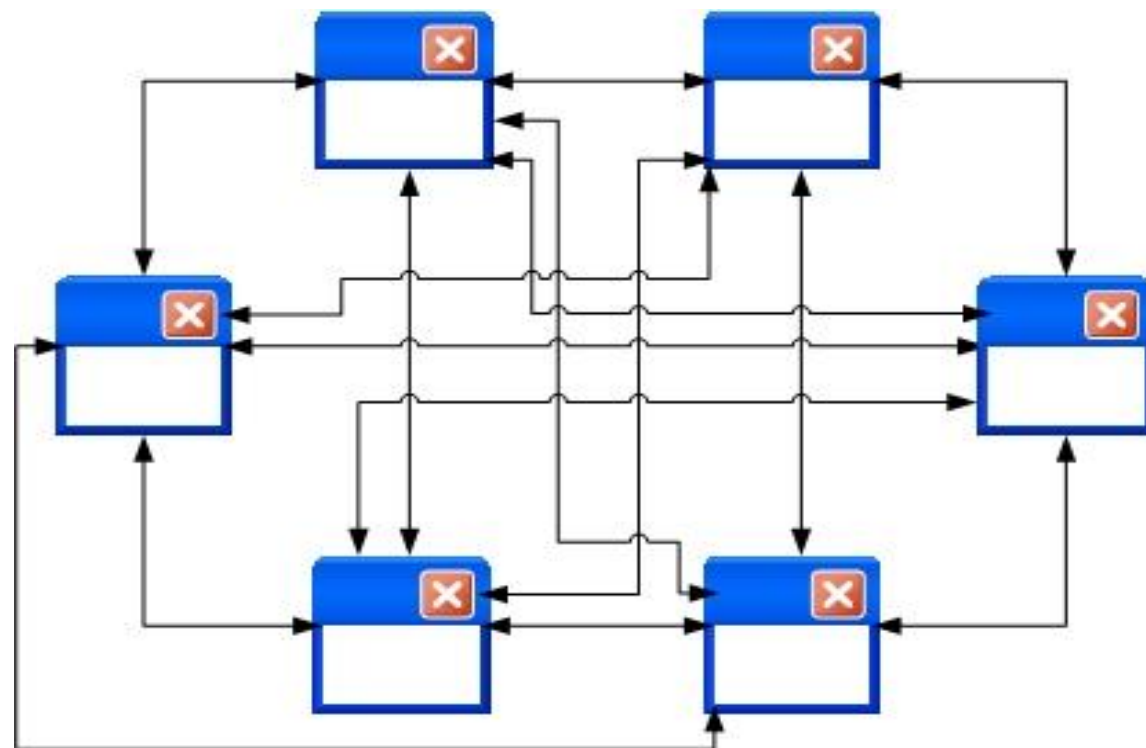
前面我们已经介绍了许多对象协作与信息交换的编程技巧，给出的例子都是主窗体与单种类型的从窗体之间的信息交换。

如果从窗体的类型不止一种，那么，你能否修改前面的示例，让它们支持主窗体与一种以上类型的从窗体交换信息？

“多对多”的对象信息交换

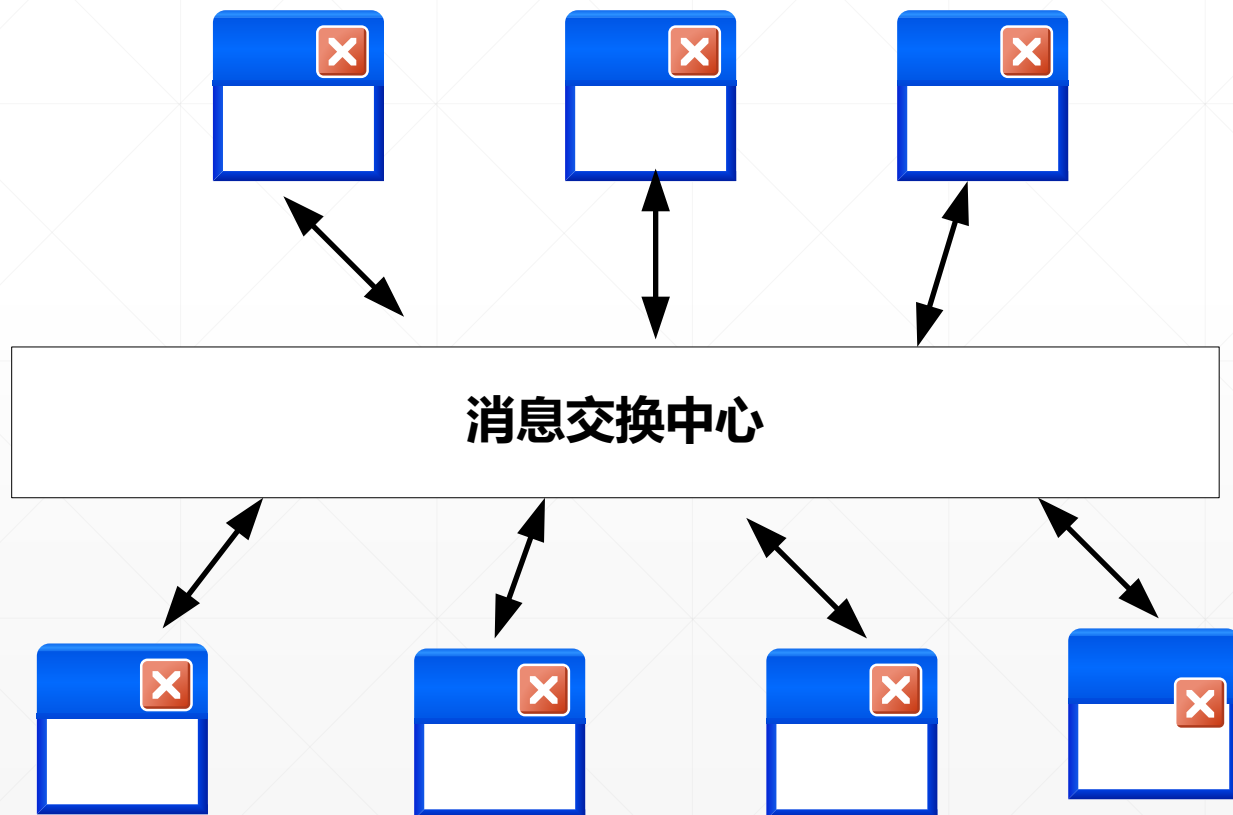
使用“**对象之间相互执有对方引用**”的方式实现对象间的相互通讯，当有“一堆”对象需要相互通讯时，我们面临着一个困境：

对象之间建立了一个复杂的关联网络，**牵一发，动全身**……



难题的解决思路

改“多对多”为“一对多”



如何编程实现左图的设计方案?
这个任务留为作业。