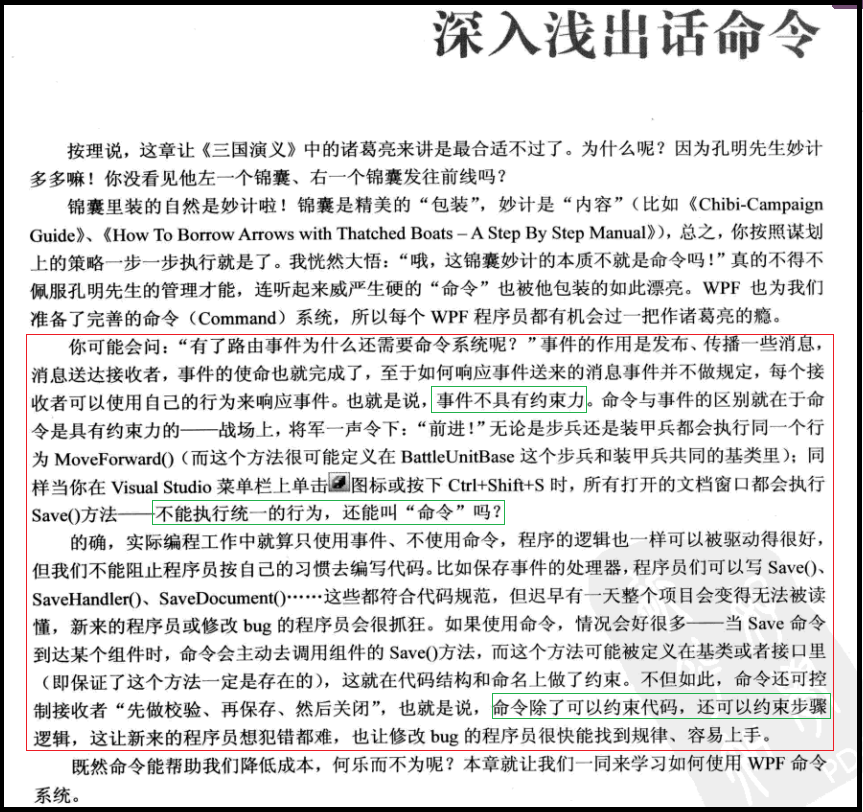
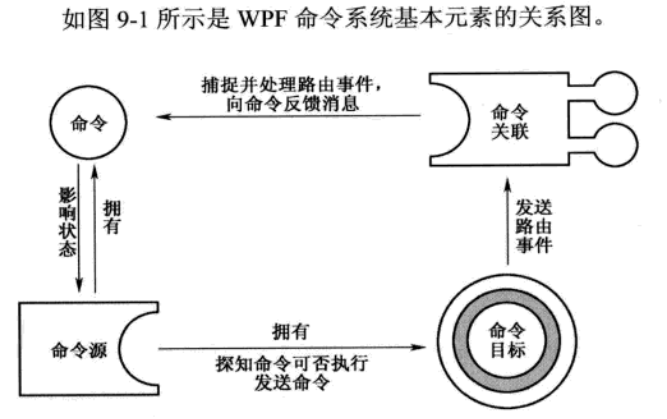
### 为什么需要命令



* 统一执行动作
* 代码结构、命名等编程规范上的约束
* 约束步骤逻辑

### 命令的基本元素



命令的主体：

* 命令的载体，类比于电报、军符、令箭等命令的载体、WPF中实现了Icommand接口的类对象
* 命令的来源，类比于指挥官、领导、WPF中实现了IcommandSource接口的类对象
* 命令执行者，类比于小兵、职员、WPF中实现了IinputElement接口的类对象

命令的传递(binding)：

XML：控件.Command=\*\*Cmd;

后台：控件.CommandBindings.Add(\*\*cmd的binding对象);

命令的执行：

* 命令的执行判断：Command. CanExecuted+=执行判断的回调函数；对是否可执行进行判断
* 命令的执行：Command. Executed+=执行命令的回调函数；具体的命令执行过程
* 备注：命令是不断循环触发的，程序会周期性的调用CanExecuted、Executed事件，为了效率，在函数结尾加上 **arg.Handled=true：**结束事件的后续路由。

### 命令的用例



### 命令的深入探究

以button控件为例子

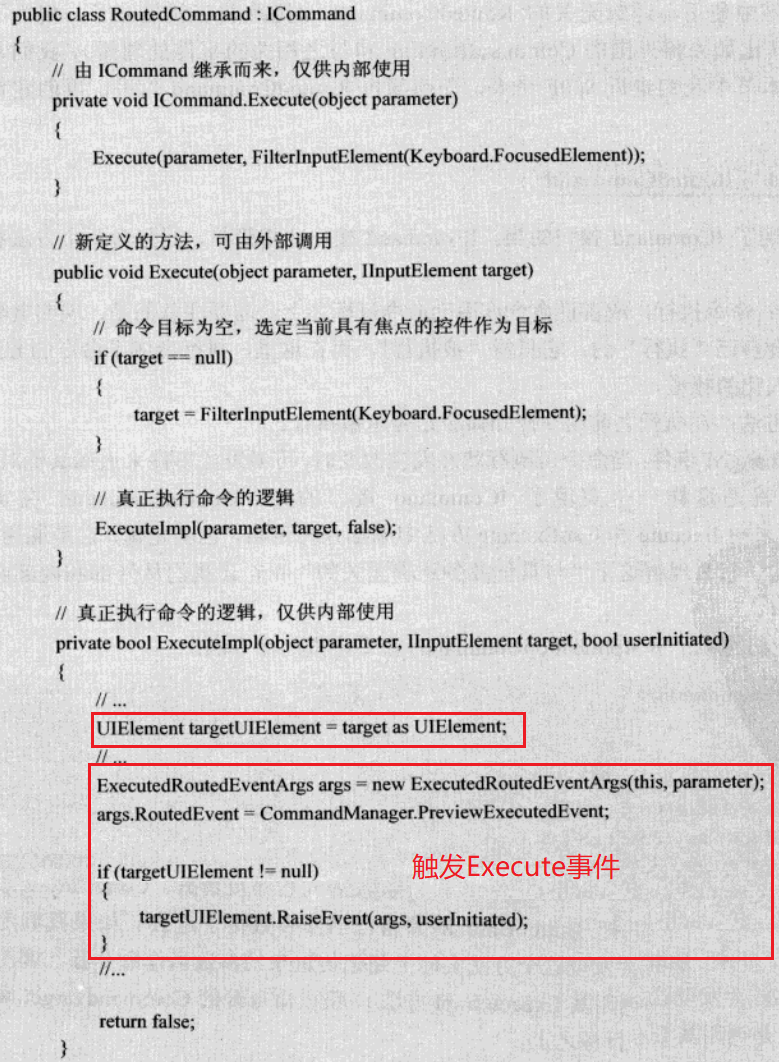
RouteCommand cmd=new RoutedCommand; //继承了Icommand的命令

Button btn=new Button(); //继承了IcommandSource的控件

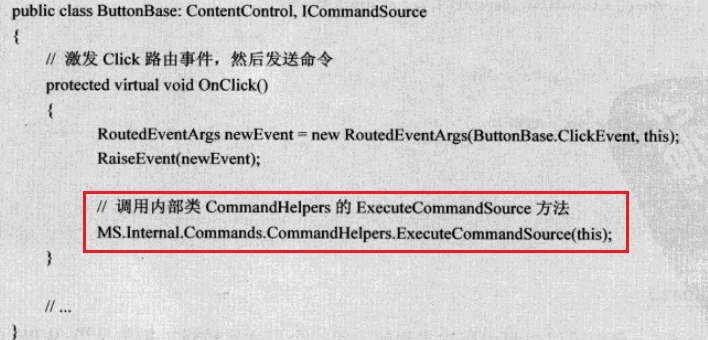
Btn.Command=cmd;

Btn.CommandTarget=this.textBox; //继承了IinputElement的目标控件textBox

#### RoutedCommand：生成Execute事件参数，触发Execute事件



#### Button控件内部：



在button控件的OnClick()事件处理函数中，根据button.commnd和button.commandTarget触发相应的命令

ExecuteCommandSource的具体处理如下图

