

- 该解决方案为 Windows 第三次实验内容
- 包含了两个 WinForm 项目和两个 WPF 项目
- 其中两个 WinForm 项目分别为 MsgEvent 和 MsgRev
 - a) MsgEvent 用于消息的发送和对事件的处理
 - i. 消息的发送使用了 User32.dll 中的 SendMessage 函数, 通过 FindWindow 函数找到对应的窗口再发送消息。
 - ii. 事件的处理采用了委托的方式, 将事件绑定在委托上, 将事件的触发封装在函数里面, 通过调用函数来触发事件, 执行相应的事件处理。
 - b) MsgRev 用于对消息的接收和显示
 - i. 消息的接收通过重载 DefWndProc 函数来实现自定义消息的接收和处理。
- 其中两个 WPF 项目分别为 MsgEventWPF 和 MsgRevWPF
 - a) MsgEventWPF 用于消息的发送和对事件的处理。
 - i. 消息的发送使用了 User32.dll 中的 SendMessage 函数, 通过 FindWindow 函数找到对应的窗口再发送消息。
 - ii. 事件的处理采用了委托的方式, 将事件绑定在委托上, 将事件的触发封装在函数里面, 通过调用函数来触发事件, 执行相应的事件处理。
 - b) MsgRevWPF 用于对消息的接收和显示
 - i. 消息的接收通过添加钩子函数 MainWindowProc 来实现自定义消息的接收和处理。