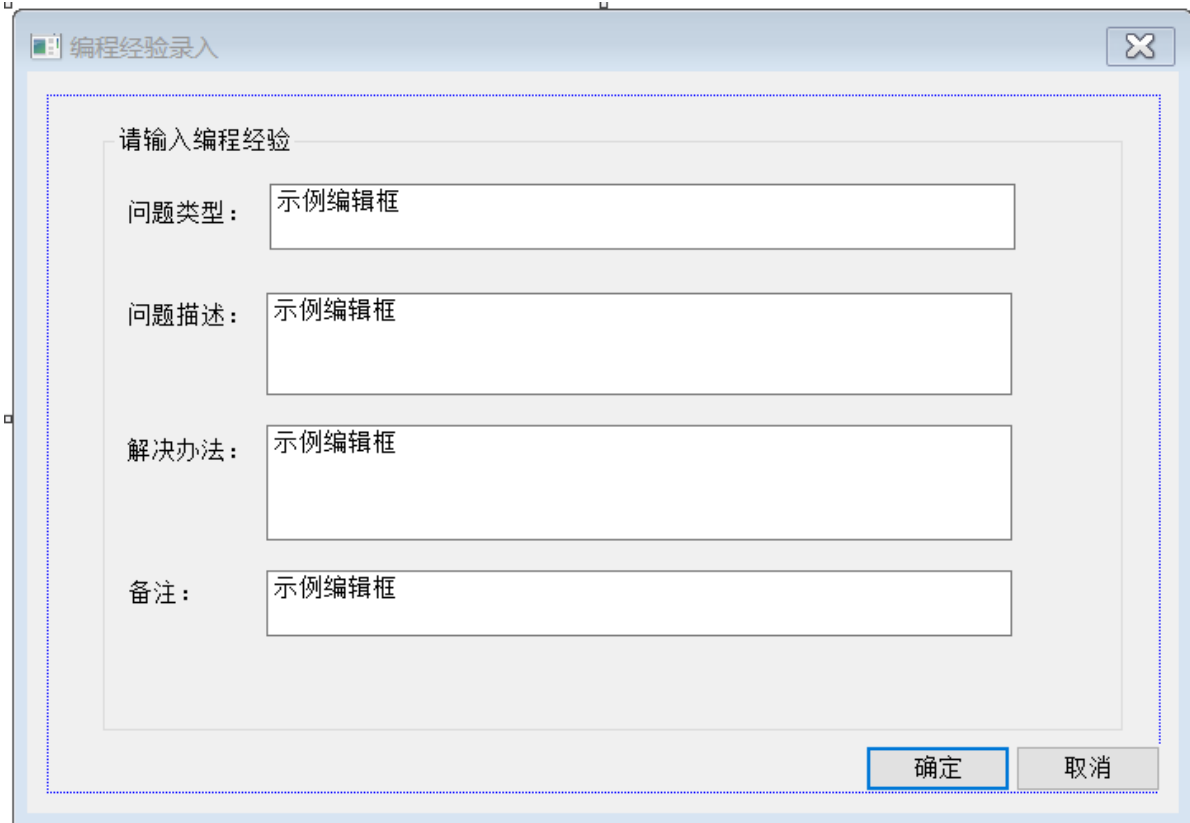


面向对象考试之对话框

1. 直接资源 -> 新建对话框：
2. 注意，**一定要改对话框的名字，一定要改，一定要改!!!**
3. 然后可以进行设计了，点击左侧工具栏，即可调出控件进行设计
4. 一个简单的对话框，基本就用到了 **Static Text**、**Edit Control**、**Group-Box**（看起来更美观）

比如设计如下：



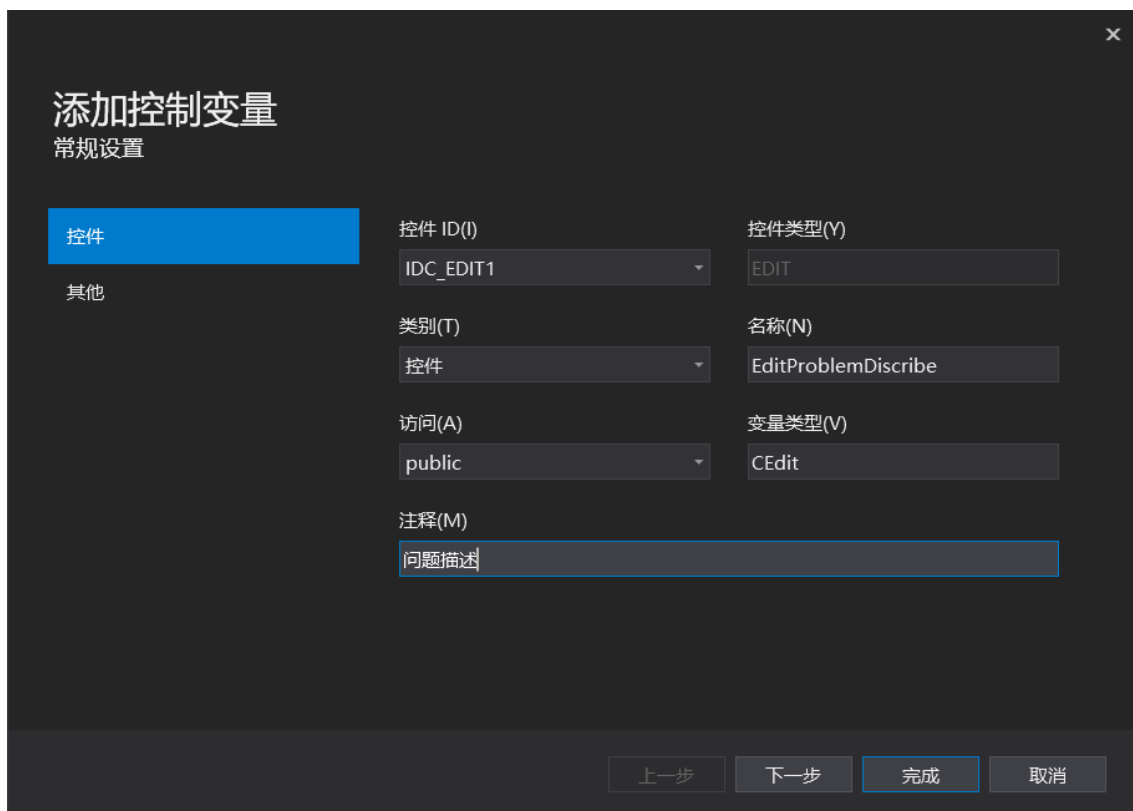
5. 在MFC中，这些对话框的这些控件的ID可以不用管，当然按钮的ID还是要改一下的。
6. 每一给对话框，都要建立一个类单独处理，右键，添加类即可（类名统一使用CDialogXXXXX）。
7. 然后可以在需要调用对话框的地方，先调用一下试一下：

注意要先在需要调用对话框的文件中引入该对话框的头文件!!!!

对话框的调用方法如下：

```
CDialogInput dialog;  
if (dialog.DoModal() == IDOK)  
{  
    InvalidateRect(NULL); //刷新  
}
```

8. ok，看到成功调用的时候，下一步就要开始设计把控件添加到类的成员变量中（统一以Edit命名），变量会自动添加并绑定。



9. 在数据结构定义好之后，我们可以尝试读取对话框控件的信息

假设在确认按下后，用类向导创建对话框按键的处理

10. 在处理中，有以下代码：

```
//这是个接收控件消息的实例
void CDialogInput::OnBnClickedOk()
{
    // TODO: 在此添加控件通知处理程序代码
    char tempType[100] = { 0 };
    char tempDecribe[100] = { 0 };
    char tempSolve[100] = { 0 };
    char tempOther[100] = { 0 };

    //接收控件消息（C类型的字符串，必须指定长度）
    EditProblemType.GetWindowTextA(tempType, 100);
    EditProblemDiscribe.GetWindowTextA(tempDecribe, 100);
    EditProblemSolve.GetWindowTextA(tempSolve, 100);
    EditProblemOther.GetWindowTextA(tempOther, 100);

    //增加一条数据
    AddOneDataToBuffer(tempType, tempDecribe, tempSolve, tempOther);

    //刷新界面
    InvalidateRect(NULL); //刷新

    CDialogEx::OnOK();
}

//注意接收到的是字符串，如果想要接收浮点数或者整数，应该使用格式化输入，如下：
sscanf_s(tempStr, "%d", &Int);
sscanf_s(tempStr, "%f", &Float);
```

```
//这是一个发送的实例
BOOL CDialogInput::OnInitDialog()
{
    CDialogEx::OnInitDialog();
    // TODO: 在此添加额外的初始化
    //向控件发送消息
    EditProblemType.SetWindowTextA("你1好");
    EditProblemDiscribe.SetWindowTextA("你2好");
    EditProblemSolve.SetWindowTextA("你3好");
    EditProblemOther.SetWindowTextA("你4好");

    return TRUE; // return TRUE unless you set the focus to a control
               // 异常：OCX 属性页应返回 FALSE
}

//如果有一个整形需要显示的话，应该用wsprintf转换一下
char TempStr[50] = { 0 };
wsprintf(TempStr, "%d", Int);
```

11. 注意刷新函数的使用 `InvalidateRect(NULL)`; 这句代码在主函数中调用，就是刷新主函数窗口，在对话框中调用，就是刷新对话框窗口，注意调用的地方。

12. 在最后，可以加上图片。空间是picture control, 选为bitmap形式，然后导入位图资源即可。



上面是自定义，对话框，下面是基本对话框的调用

1. 字体（包括颜色）选择对话框

```
void CWQAView::OnMenuFont()
{
    // TODO: 在此添加命令处理程序代码
    CFontDialog dialog;
    if (dialog.DoModal() == IDOK)
    {
        dialog.GetCurrentFont(&currentFont);
        currentTextColor = dialog.GetColor();
    }
}
```

2. 纯颜色选择对话框

```
void CWQAView::OnMenuColor()
{
    // TODO: 在此添加命令处理程序代码
    CColorDialog dialog;
    if (dialog.DoModal() == IDOK)
    {
        currentTextColor = dialog.GetColor();
    }
}
```