

---

## 实验二 Socket 通信中的前端 UI 开发

### Windows 标准控件的使用

#### 【实验目的】

1. 了解 MFC 消息循环机制；
2. 掌握 MFC 中部分常用控件的使用方法；
3. 基于 MFC 编写具备一定用户界面交互功能的应用程序。

#### 【实验任务】

1. 进一步理解 MFC 消息循环机制；
2. 进一步了解掌握 MFC 控件的使用方法，使用多种控件开发具有简单用户交互功能的 MFC 应用程序—个人信息简表；
3. 制作具有基本的数据处理能力的 MFC 应用程序—计算器。

#### 【实验设备】

个人 PC，WindowsXP 及以上操作系统，Visual C++ 6.0 及以上开发环境。

#### 【计划学时】

4-8 学时

#### 【实验内容】

1. 使用多种控件开发具有简单用户界面交互功能的 MFC 应用程序，显示你的个人信息，界面设计如图 1. 请将界面控件规则对齐排列，同类控件使用静态组合框归类布置。

##### 1.1 功能要求

1. “弹出模态对话框”按钮无效(该控件不使能)；
2. 点击“弹出非模态对话框”按钮，弹出一个你设计好的非模态对话框，注意关闭该对话框时对话框资源的销毁过程。

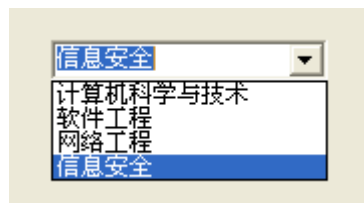
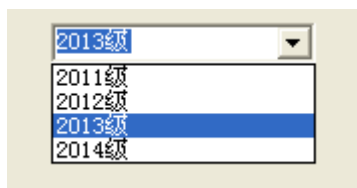


图 1 系统界面

3. 点击“打开文件”按钮，弹出打开文件对话框，选中你上次保存的 **txt** 文件，点击“确定”后打开该文件，从中读取你的姓名、学号、和本地网络信息，动态填入对应的编辑框中。

```
-----  
... .. //弹出打开文件对话框  
if(dlg.DoModal()==IDOK)           //选定文件，点击确定  
{  
    ... ..                          //得到文件路径  
    CStdioFile file;                //此方法能方便使用 CString 进行处理  
    CString str;                    //用来逐行保存文件内容的字符串(1 个或多个)  
    if(file.Open(m_path,CFile::modeRead))    //打开你选中的文件  
    {  
        while(file.ReadString(str))        //逐行读文件内容  
        {  
            //对每行(str)进行处理，把数据填入对应的编辑框  
        }  
    }  
}  
-----
```

- 
4. 从年级专业组合框中选中你的年级和专业，两个组合框应有如下选项和选中效果：



5. 使用两个互斥的 **Radio** 按钮，确定你的性别信息；
6. 使用一个滑动条 **Slider**，动态得到你的年龄；
7. 使用一个 **Picture** 控件，显示你的 **bmp** 位图照片；
8. 在界面底部，放置一个进度条控件，在你的程序运行过程中，始终动态滚动；
9. 手动输入补全剩余的编辑框，填写你的籍贯、政治面貌、联系方式等基本信息；
10. 所有信息填写完毕后，点击“保存文件”按钮，弹出保存文件对话框，将除图片以外的你的个人信息再次保存为一个 **txt** 文件。

```
-----  
... ..//打开保存文件对话框  
if(dlg.DoModal() == IDOK) // 确定输入的文件名  
{  
    ... .. //得到文件路径  
    CFile file;  
    file.Open(fileName,CFile::modeCreate | CFile::modeReadWrite); //打开文件  
  
    CString strlist;  
    CString str1 = "姓名:";  
    strlist += str1 + m_姓名编辑框变量值;  
    file.Write(strlist,strlist.GetLength()); //写入文件  
    ... ..  
    file.Close();  
}
```

-----  
(本题大部分内容与课堂授课内容完全一致，另请参照课堂授课内容)

## 2. MFC 下基本的数据运算和处理—计算器

### 2.1 功能要求

简易计算(必做): 实现两数的加、减、乘、除、、取模等简单计算;

科学计算: 实现包括(反)正弦、(反)余弦、(反)正切、(反)余切、乘方、开方、指数等函数运算;

提示功能: 具备数据的溢出错误提示, 无意义运算表达式提示等;

进制转换: 实现二进制、八进制、十进制、十六进制转换功能。

系统的用户界面如图 2(简单计算器)和图 3(科学计算器)所示。所有运算符号和数字使用按键输入, 实时显示。

考虑到同学们实际编程经验的差异, 本题做不同的要求, 具有开发经验的同学可实现简单计算、科学计算、提示功能、进制转换功能, 其他同学实现简单计

算功能。



图 2 简易计算器运行界面



图 3 科学计算器运行界面

---

**思路 1:** 先整体输入一个运算表达式，再从中区分运算的数字和运算符号，给出结果。

**思路 2:** 在输入每个字符时就区分输入的运算数字、运算符号等。

示例代码

计算过程的构思

```
CString m_input1;           //第一个操作数
CString m_input2;           //第二个操作数
BOOL m_bXX;                 //是否按下运算符
int  m_OperationIndicator;  //对应哪个运算符
```

数字按键的处理

```
void CXXXDlg::On_数字按键()
{
    if(!m_bXX)                //没按下运算符
    {
        m_input1 += "键值";
        m_display += "键值";
        UpdateData(FALSE);
    }
    else                       //运算符已按下
    {
        m_input2 += "键值";
        m_display += "键值";
        UpdateData(FALSE);
    }
}
```

运算符号按键的处理

```
void CXXXDlg::On_运算符号 ()
{
    m_bXX = TRUE;
    m_OperationIndicator = 1;    //用来标识哪个运算符号按下
    m_display = m_display + "运算符号";
    UpdateData(false);
}
```

特殊符号的处理

```
void CXXXDlg::OnButton_等号() //等于号的功能函数
{
    double X,Y, result;
    X=strtod(m_input1,NULL);    //把运算符号前后的两个字符串转化成浮点型
    Y=strtod(m_input2,NULL);
```

---

```
switch(m_OperationIndicator) //判断具体进行哪种计算
{
case 1:
    result = X + Y;
    m_display.Format("%f",result);    //显示运算结果
    UpdateData(false);
    break;

default :
    UpdateData(false);
}
m_input1 = m_display;
m_input2 = "";
}

void CXXxDlg::On_清除键_r() //清零键的功能函数
{
    m_input1 = "";
    m_input2 = "";
    m_bXX = FALSE;
    m_OperationIndicator = 0;
    m_display = "";
    UpdateData(FALSE);
}
```

-----

对进一步改进的思考:

如何直接输入长表达式求得计算结果?

## 【实验报告要求】

将编程题 1 中打开文件后的读文件、自动填充编辑框部分和将你的个人信息保存文件部分的代码写在实验报告上，要求给出详细注释。

---

## 附录：实验 2 中你可能会用到的 CString 类方法：

### 1. 两 CString 类对象比较

```
int Compare( LPCTSTR lpsz ) const;  
返回值 两者一样返回 0，区分大小字符  
CString s1( "abc" );  
CString s2( "abc" );  
s1.Compare( s2 );结果为 0
```

### 2. 两 CString 类对象连接

```
CString str1("Gray");  
CString str2("Cat");  
CString str3 = str1 + str2;  
str3 结果为 GrayCat
```

### 3. 在 CString 类对象中搜索特定字符，索引以 0 开始

```
int Find( TCHAR ch ) const;  
int Find( LPCTSTR lpszSub ) const;  
CString s( "abcdef" );  
s.Find( 'c' ), 结果为 2 ;  
s.Find( "de" )结果为 3 ;
```

### 4. 格式转化

```
CString str;  
Str.Format(“%d”,13);  
把数字转换成字符串 此时 Str 为 13
```

### 5. 求字符串长度

```
CString::GetLength  
CString s( "abcdef" );  
s.GetLength() ;长度为 6
```

### 6. 提取前几个字符

```
CString::Left  
CString Left( int nCount ) const;  
返回的字符串是前 nCount 个字符。  
CString s( _T("abcdef") );  
s.Left(2);结果为 ab
```

### 7. 去掉左右的空格字符

```
去掉字符串左边空格： CString.TrimLeft();
```

---

去掉字符串右边空格: `CString.TrimRight()`;

8. 提取数据中间某段

`CString::Mid`

`CString Mid( int nFirst ) const;`

`CString Mid( int nFirst, int nCount ) const;`

`nCount` 代表要提取的字符数, `nFirst` 代表要提取的开始索引位置

`CString s( _T("abcdef") );`

`s.Mid( 2, 3 );`结果为 cde