

제8장

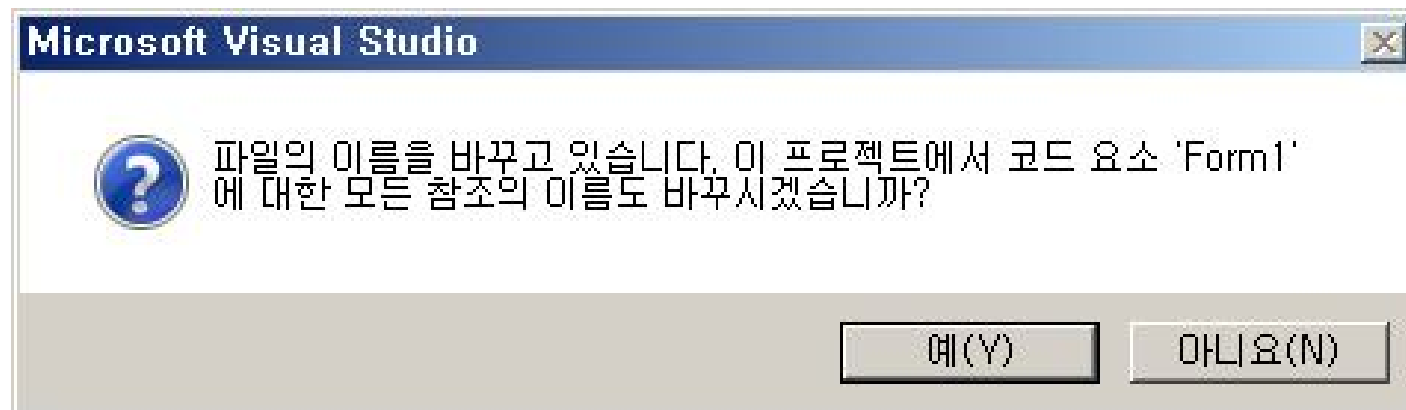
메모관리 프로그램으로 배우는 C#

제주대학교 컴퓨터공학과
변영철 교수

<http://groups.google.com/group/jnu-cp2>

01 프로젝트 생성 및 사전작업

- 새로운 프로젝트 (Memo) 생성
 - Visual C#
 - Windows
 - Windows Forms 응용 프로그램
- 이름 바꾸기
 - 파일 이름 Form1.cs -> MainForm.cs



01 프로젝트 생성 및 사전작업

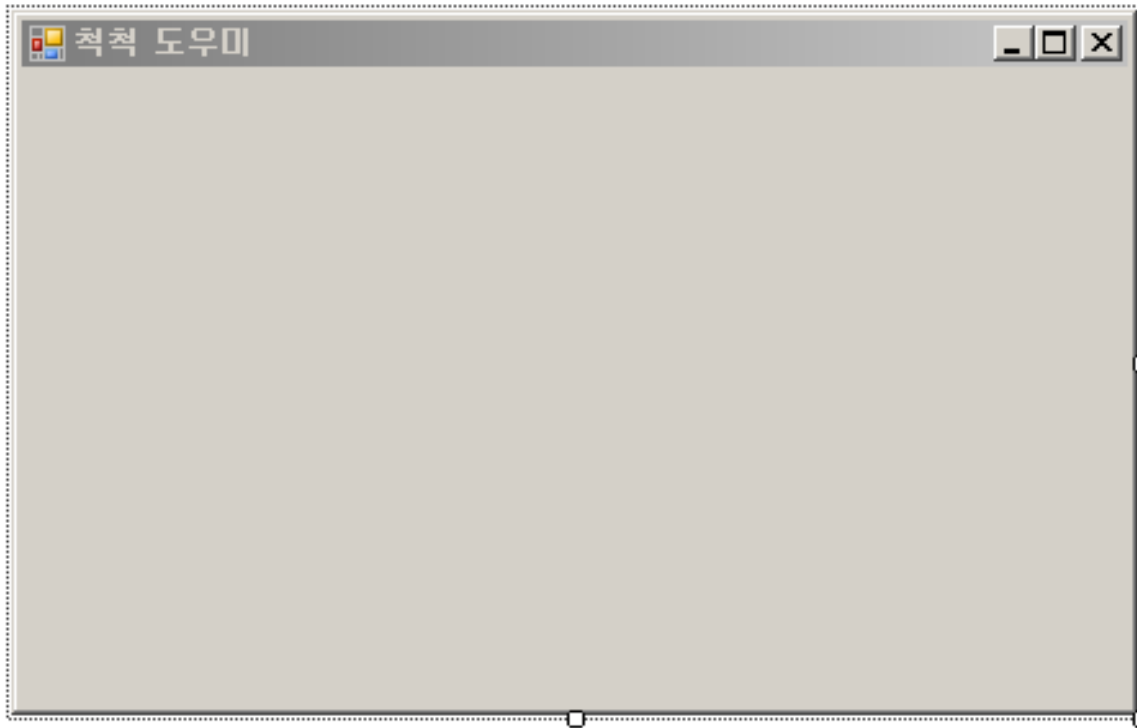
- 2개의 Partial 코드 확인하기
 - Partial 클래스로 코드가 분리되어 있음
 - 사용자가 작성하는 코드(Main.cs)와 디자인 마술사에 의하여 자동으로 생성하는 코드(MainForm.Designer.cs)

```
namespace Memo
{
    partial class MainForm
    {
        //.....
    }
}
```

```
namespace Memo
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }
    }
}
```

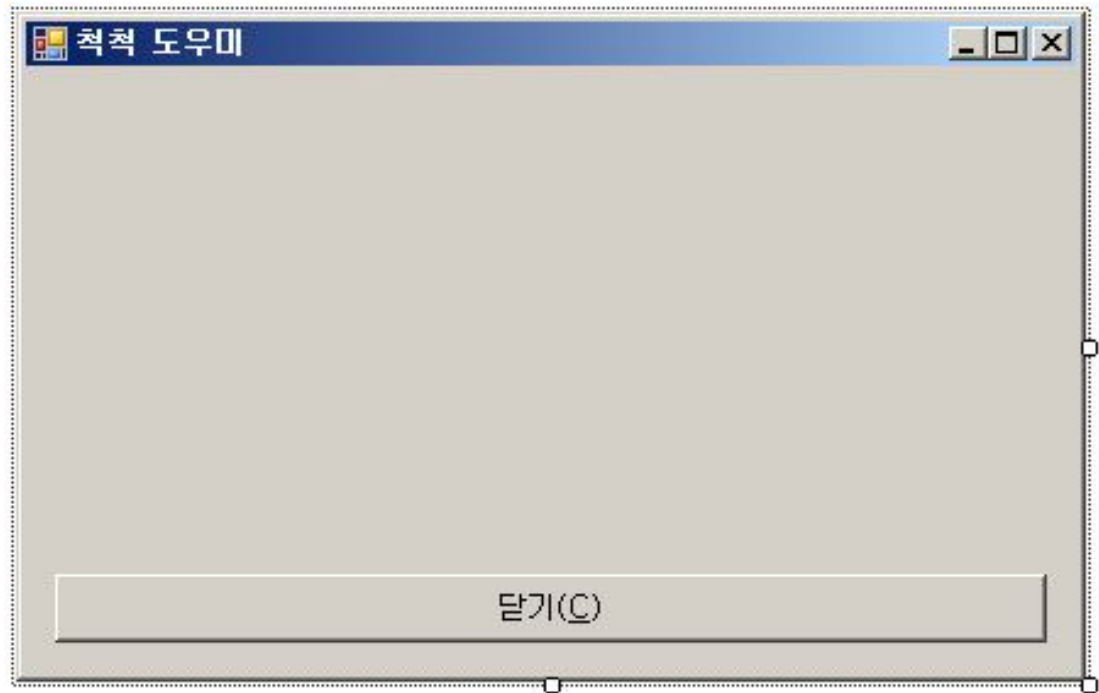
02. 비주얼 디자인

- 폼 윈도우 크기 변경
- 속성 창을 이용한 타이틀 변경 : Text 속성



02. 비주얼 디자인

- 도구상자를 이용하여 닫기 버튼 디자인
 - (Name) : **btnClose**
 - Text : 닫기(&C)



03. 프로그램 구현

- '닫기' 버튼 핸들러 함수 추가
 - 폼 윈도우에서 버튼 두 번 클릭

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
    }

    private void btnClose_Click(object sender, EventArgs e)
    {
        Close();
    }
}
```

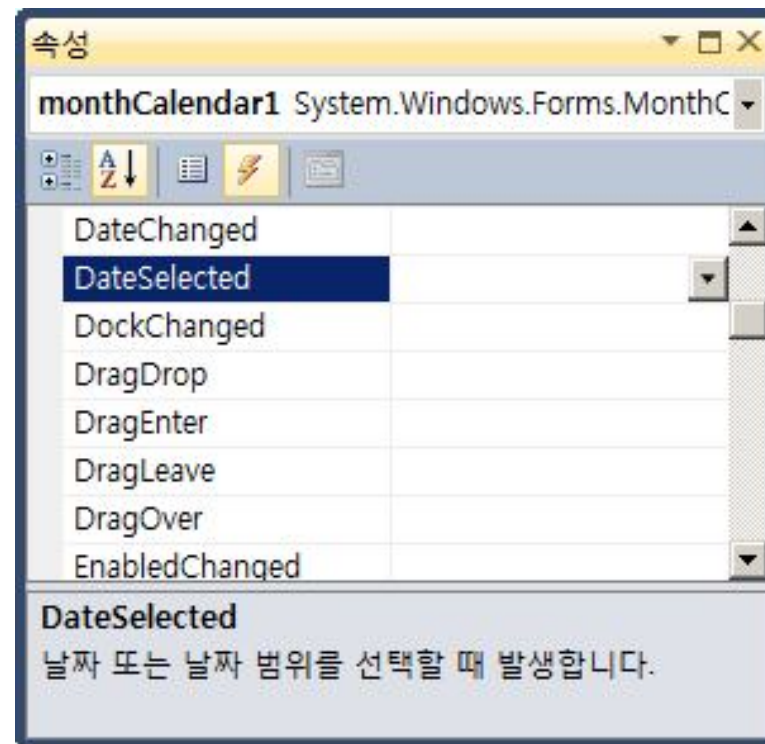
04. 비주얼 디자인 및 핸들러 함수 작성

- 캘린더 디자인
 - (Name) : **monthCalendar**



04. 비주얼 디자인 및 핸들러 함수 작성

- 날짜 선택시 실행되는 핸들러 함수 추가
 - DateSelected 두 번 클릭



04. 비주얼 디자인 및 핸들러 함수 작성

```
private void monthCalendar_DateSelected(object sender,  
DateRangeEventArgs e)  
{  
    MessageBox.Show(monthCalendar.SelectionStart.ToString());  
}
```

04. 비주얼 디자인 및 핸들러 함수 작성

- 텍스트 상자(TextBox) 디자인
 - (Name) : **tbMemo**
 - Multiline : True



04. 비주얼 디자인 및 핸들러 함수 작성

- 텍스트 상자에 날짜 출력

```
private void monthCalendar_DateSelected(object sender,  
DateRangeEventArgs e)  
{  
    tbMemo.Text = monthCalendar.SelectionStart.ToString();  
}
```

- 날짜만 표시

```
private void monthCalendar_DateSelected(object sender,  
DateRangeEventArgs e)  
{  
    string Tmp = monthCalendar.SelectionStart.ToString();  
    Tmp = Tmp.Remove(10, Tmp.Length - 10);  
    tbMemo.Text = Tmp;  
}
```

05. 캘린더 클래스 모듈 작성하기

- TellMeTodayDate 멤버함수가 있다면...

```
private void monthCalendar_DateSelected(object sender, DateRangeEventArgs e)
{
    tbMemo.Text = monthCalendar.TellMeTodayDate();
}
```

- MyMonthCalendar 클래스 추가하기
 - 솔루션 탐색기 > Memo 위에서 오른쪽 버튼 클릭 > 새 항목 추가

05. 캘린더 클래스 모듈 작성하기

```
namespace Memo
{
    class MyMonthCalendar : MonthCalendar
    {
        public string TellMeTodayDate()
        {
            return "2001-08-31";
        }
    }
}
```

```
string Tmp = SelectionStart.ToString();
Tmp = Tmp.Remove(10, Tmp.Length - 10);
return Tmp;
```

05. 캘린더 클래스 모듈 작성하기

- 객체 만들기
 - MainForm.Designer.cs

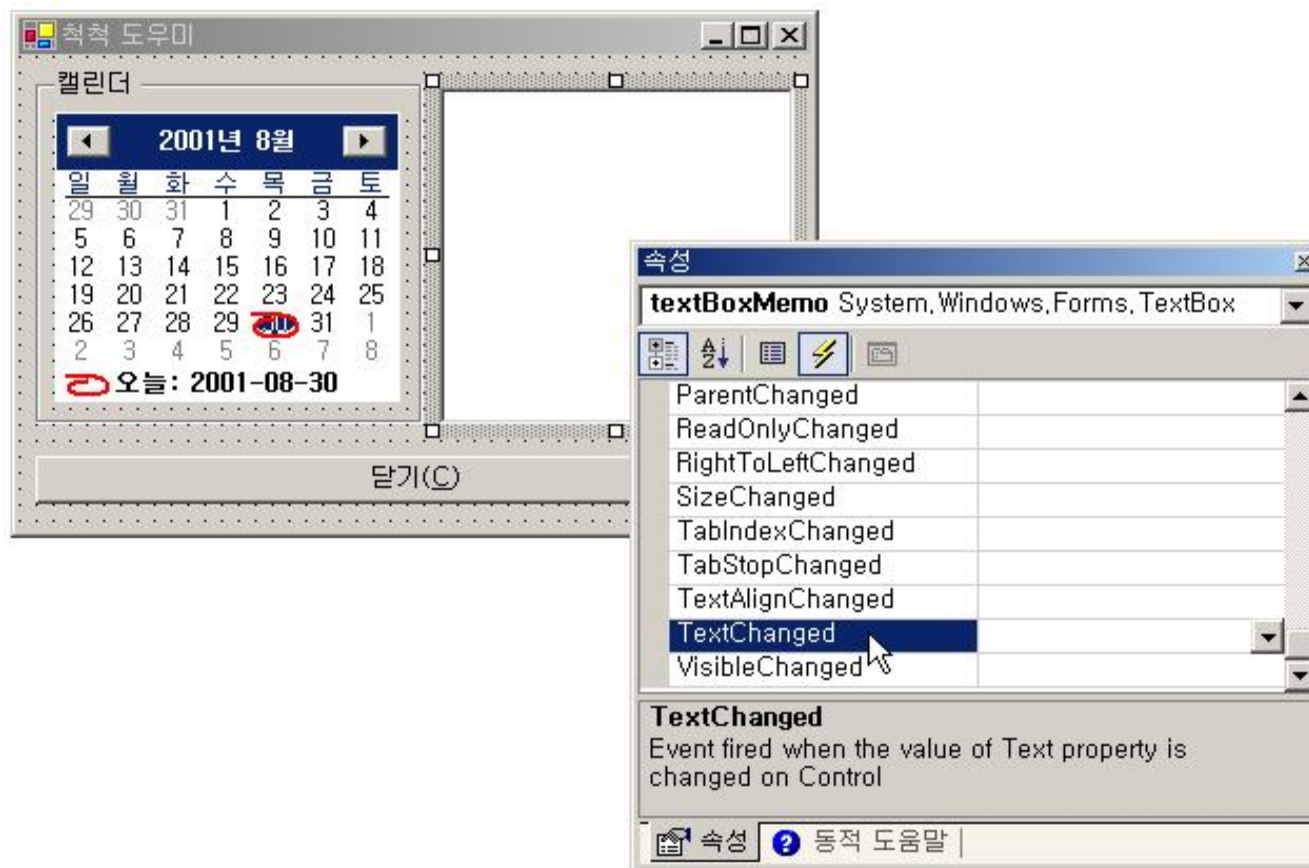
```
private void InitializeComponent()  
{  
    this.btnClose = new System.Windows.Forms.Button();  
    this.monthCalendar = new MyMonthCalendar();
```

```
...  
}
```

```
private MyMonthCalendar monthCalendar;
```

06. TextChanged 이벤트 핸들러 함수 작성

- 텍스트 상자 문자 입력시 실행되는 핸들러 함수



06. TextChanged 이벤트 핸들러 함수 작성

- 텍스트 상자에 입력할 경우 해쉬 테이블에 저장하기

```
private void tbMemo_TextChanged(object sender,  
    EventArgs e)  
{  
    string Key = monthCalendar.TellMeTodayDate();  
    m_MemoDB[Key] = tbMemo.Text;  
}
```

```
Hashtable m_MemoDB = new Hashtable();
```


06. TextChanged 이벤트 핸들러 함수 작성

- 날짜를 선택할 경우 해쉬 테이블에 저장되어있는 내용 표시하기

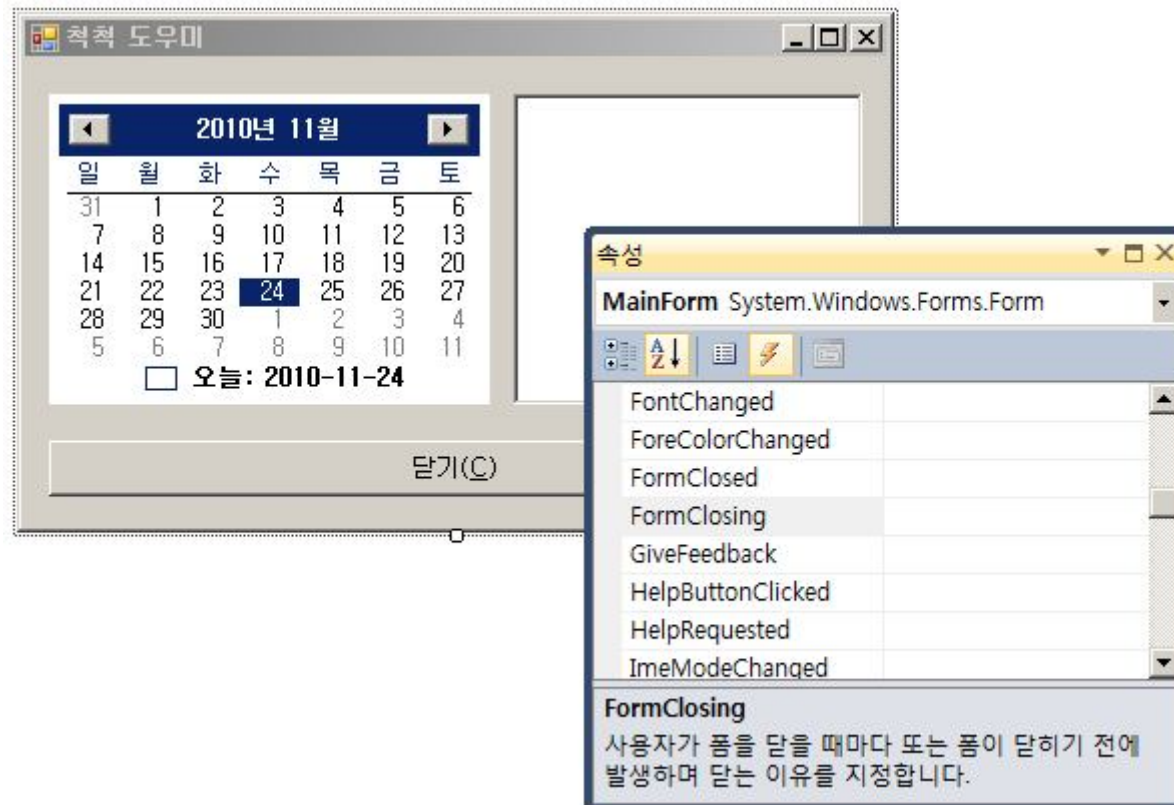
```
private void monthCalendar_DateSelected(object  
    sender, DateRangeEventArgs e)  
{  
    string Key = monthCalendar.TellMeTodayDate();  
    tbMemo.Text = (string)m_MemoDB[Key];  
}
```

06. TextChanged 이벤트 핸들러 함수 작성

```
private void tbMemo_TextChanged(object  
    sender, EventArgs e)  
{  
  
}
```

07. 직렬화 구현하기

- 폼 윈도우가 닫힐 때 실행되는 핸들러 함수 작성하기 : **FormClosing**



07. 직렬화 구현하기

- 프로그램 종료시 데이터 저장하기

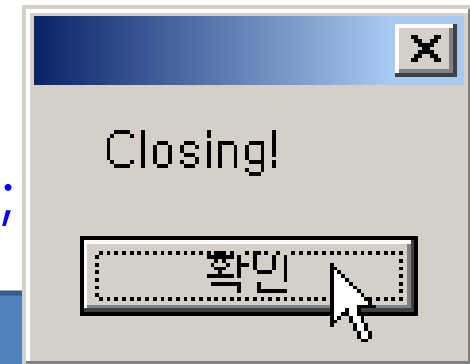
```
private void MainForm_FormClosing(object sender,  
    FormClosingEventArgs e)
```

```
{
```

```
    MessageBox.Show("Form Closing!");
```

```
}
```

```
    FileStream fs = File.Create("c:\\memo.dat");  
    BinaryFormatter gildong = new BinaryFormatter();  
    gildong.Serialize(fs, m_MemoDB);  
    fs.Close();
```



07. 직렬화 구현하기

- 프로그램 실행 시 데이터 읽어오기

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();

        FileStream fs = File.Open("c:\\memo.dat",
            FileMode.Open, FileAccess.Read);
        BinaryFormatter gildong = new BinaryFormatter();
        m_MemoDB = (Hashtable)gildong.Deserialize(fs);
        fs.Close();
    }
}
```