

제9장

C#으로 만들어보는 메모 프로그램

변영철

01.프로젝트 생성 및 사전작업

- 새로운 프로젝트 Memo 생성
 - Visual C# | Windows Forms 응용 프로그램
- 파일명 바꾸기
 - Program.cs → Memo.cs
 - Form1.cs → MainForm.cs

01.프로젝트 생성 및 사전작업

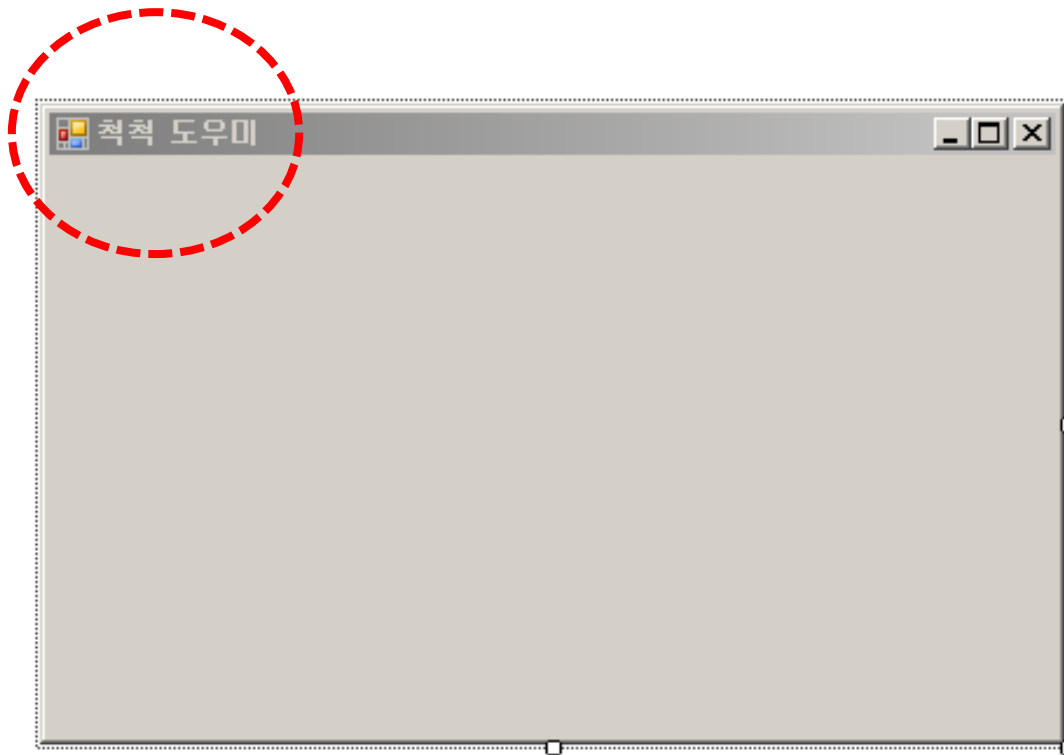
- 2개의 Partial 클래스 확인하기
 - Partial 클래스로 코드가 분리되어 있음
 - 사용자가 작성하는 코드(MainForm.cs)와 디자인 마술사에 의하여 자동으로 생성되는 코드(MainForm.Designer.cs)

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

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

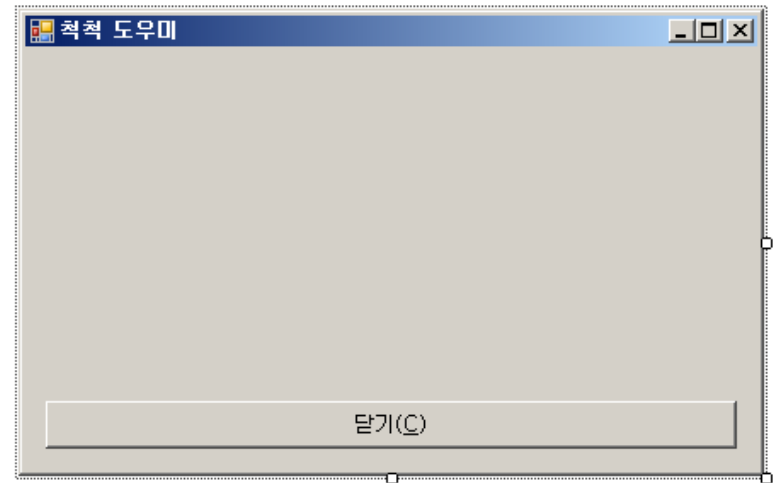
02. 비주얼 디자인

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



02. 비주얼 디자인

- 버튼 추가
 - 보기 | 도구상자
 - Button 드래그-앤-드롭
 - (Name) : **btnClose**
 - Text : 닫기(&C)



03. 프로그램 구현

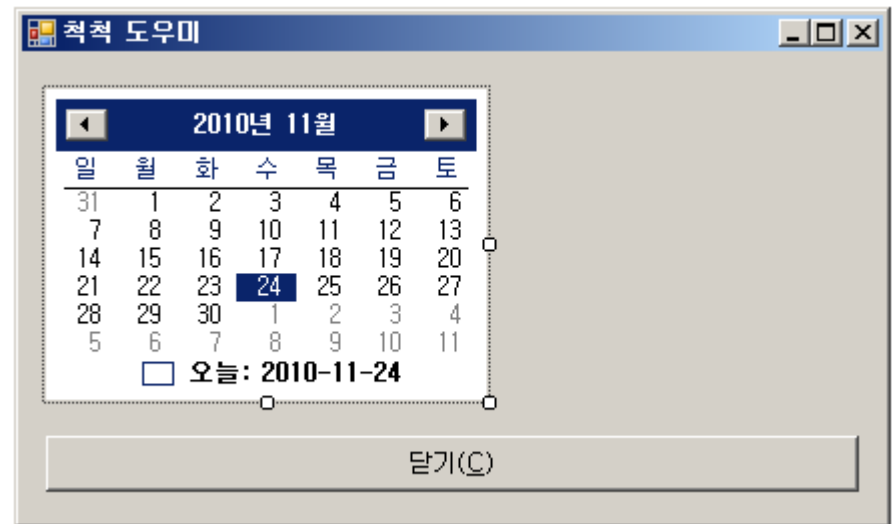
- 버튼 핸들러 함수 추가
 - 폼 윈도우에서 버튼 두 번 클릭→자동 추가

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

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

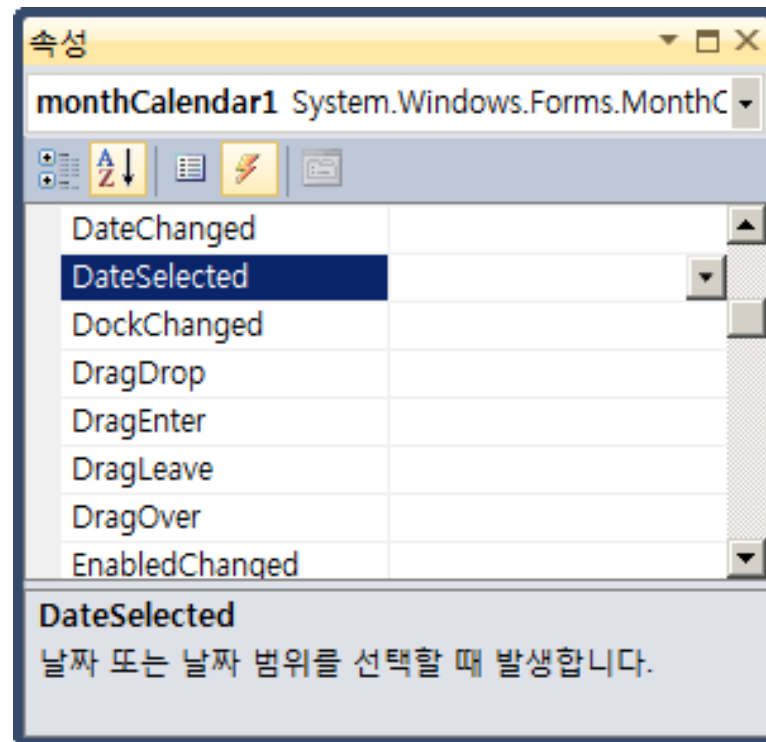
04. 비주얼 디자인 및 핸들러 함수 작성

- 캘린더 추가
 - (Name) : **monthCalendar** (길동이?)
 - 생성된 코드 확인
 - **MonthCalendar**



04. 비주얼 디자인 및 핸들러 함수 작성

- 날짜 선택 시 실행되는 핸들러 함수 추가
 - **DateSelected** 두 번 클릭→자동 추가

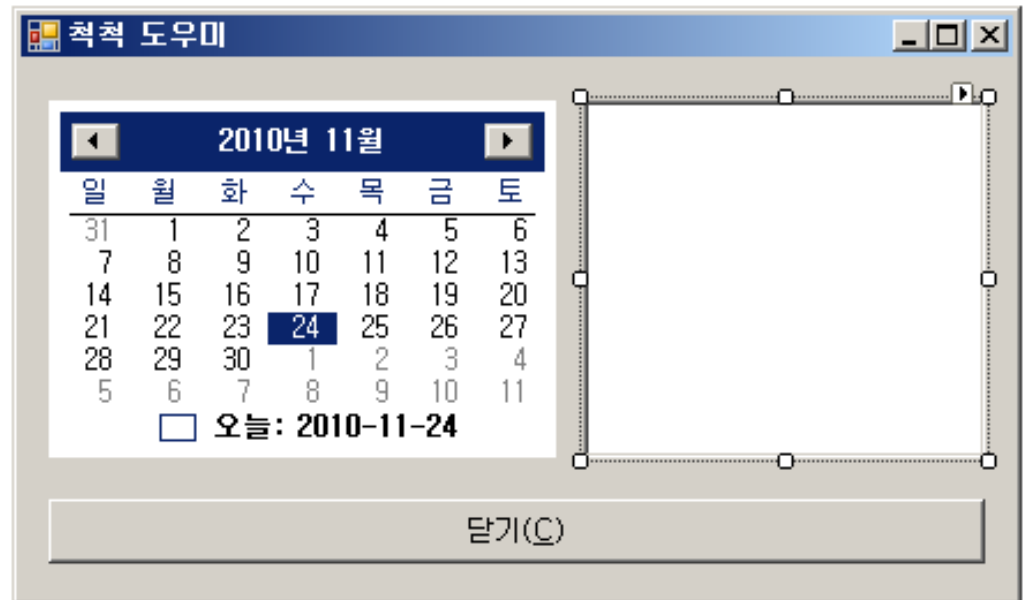


04. 비주얼 디자인 및 핸들러 함수 작성

```
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 상자에서 특정 날짜를 선택할 경우 자동으로 실행되는 핸들러 함수 추가

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

04. 비주얼 디자인 및 핸들러 함수 작성

- 필요 없는 문자열 빼고 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;
}
```

이거 뭐하는 코드?
“Date” 함수

10번째부터 끝까지 삭제



05. 새로운 클래스 모듈 작성(has-a)*

- 날짜만 표시하는 코드 세 줄을 나중에 쓸 수 있도록 추상화 : Date 함수
- 새로운 클래스(CalendarUtil)로 이동
- {오류 수정!!}
- 클래스를 새로운 파일에 작성(CalendarUtil.cs)
- 외부 폴더로 이동 (c:\cslib)
- 프로젝트 | 기존항목추가 | (파일 선택 후) 링크로 추가

05. 새로운 클래스 모듈 작성(is-a)*

- Date 멤버함수가 있다면 좋을 텐데...

하지만 없다. 멤버로 정의하라.

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

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

05. 새로운 클래스 모듈 작성(is-a)*

클래스는 뭐 하라고 있는 것?

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

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

05. 새로운 클래스 모듈 작성(is-a)*

- 객체 만들기

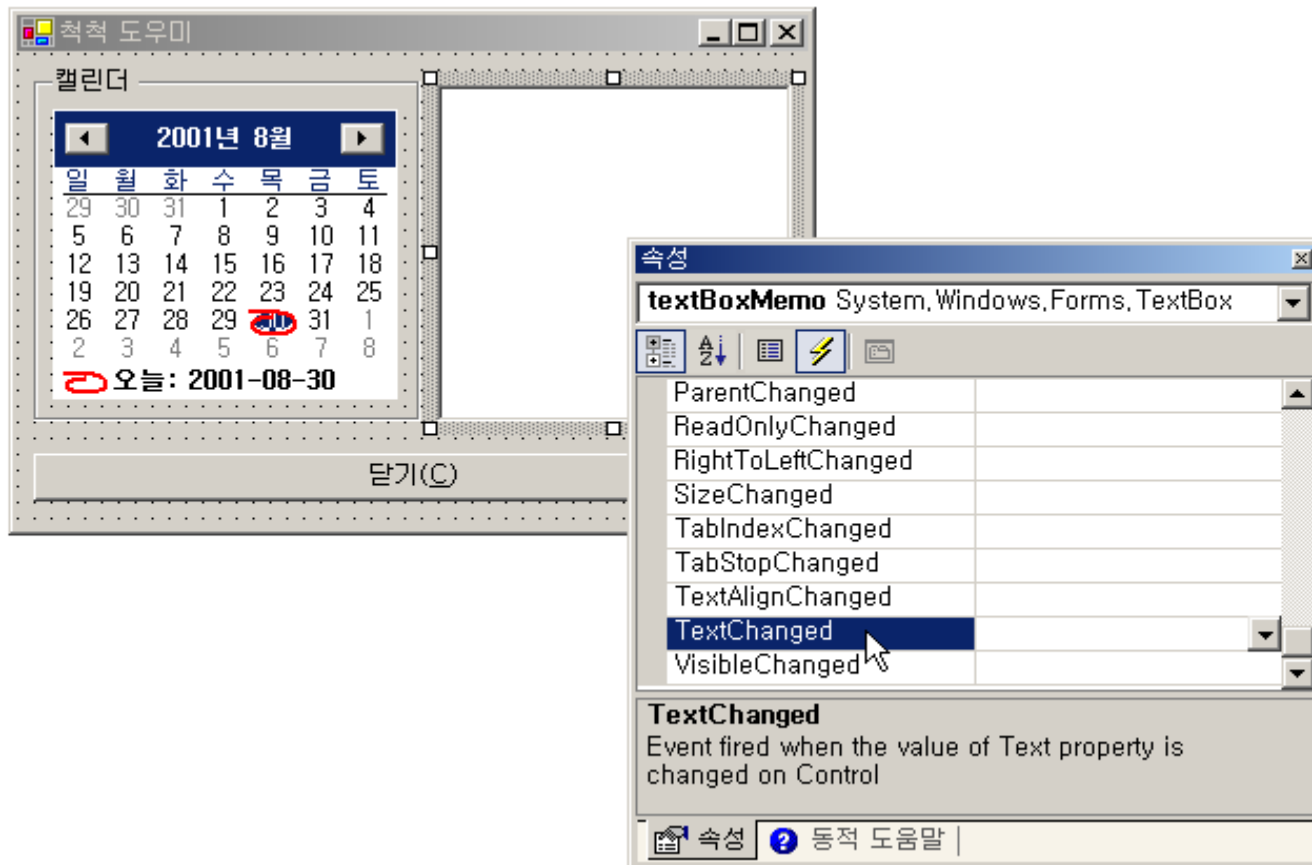
- MainForm.Designer.cs

```
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 입력 시 실행되는 핸들러 함수 추가



06. TextChanged 이벤트 핸들러 함수 작성

- 여러분이 입력한 메모를 해쉬 테이블에 저장

```
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

- 어떤 날짜를 선택하면 그 날짜에 저장된 메모 표시하기

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

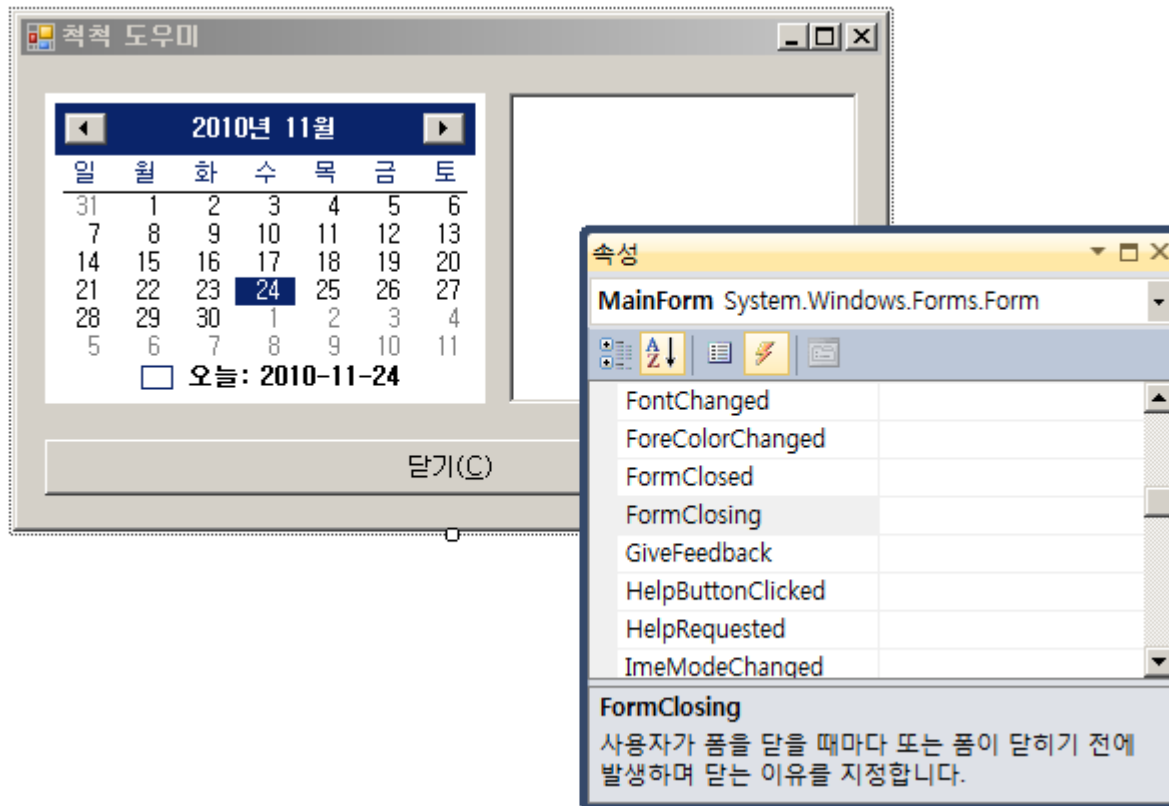
06. TextChanged 이벤트 핸들러 함수 작성

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

07. HashtableUtil 모듈 만들기*

08. 직렬화(파일 입출력)구현하기

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



08. 직렬화(파일 입출력)구현하기

- 프로그램 종료 시 데이터 **저장**하기

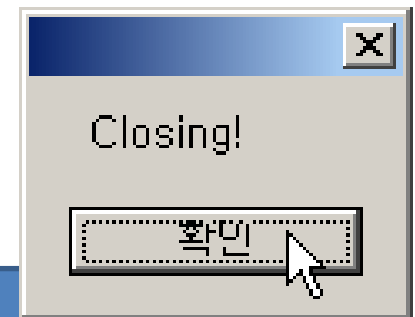
```
private void MainForm_FormClosing(object sender,  
    FormClosingEventArgs e)
```

```
{
```

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

```
}
```

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



08. 직렬화(파일 입출력)구현하기

- 프로그램 실행 시 데이터 **읽어오기**

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

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


08. 직렬화(파일 입출력)구현하기

- 맨 처음 프로그램 실행할 때는 파일이 없는데?

```
if (!File.Exists("memo.dat")) {  
    MessageBox.Show("No memo.dat");  
    m_MemoDB = new Hashtable();  
}  
else {  
    FileStream fs = File.Open("memo.dat",  
        FileMode.Open, FileAccess.Read);  
    BinaryFormatter gildong = new BinaryFormatter();  
    m_MemoDB = (Hashtable)gildong.Deserialize(fs);  
    fs.Close()  
}
```

09. HashtableDB 모듈 만들기*