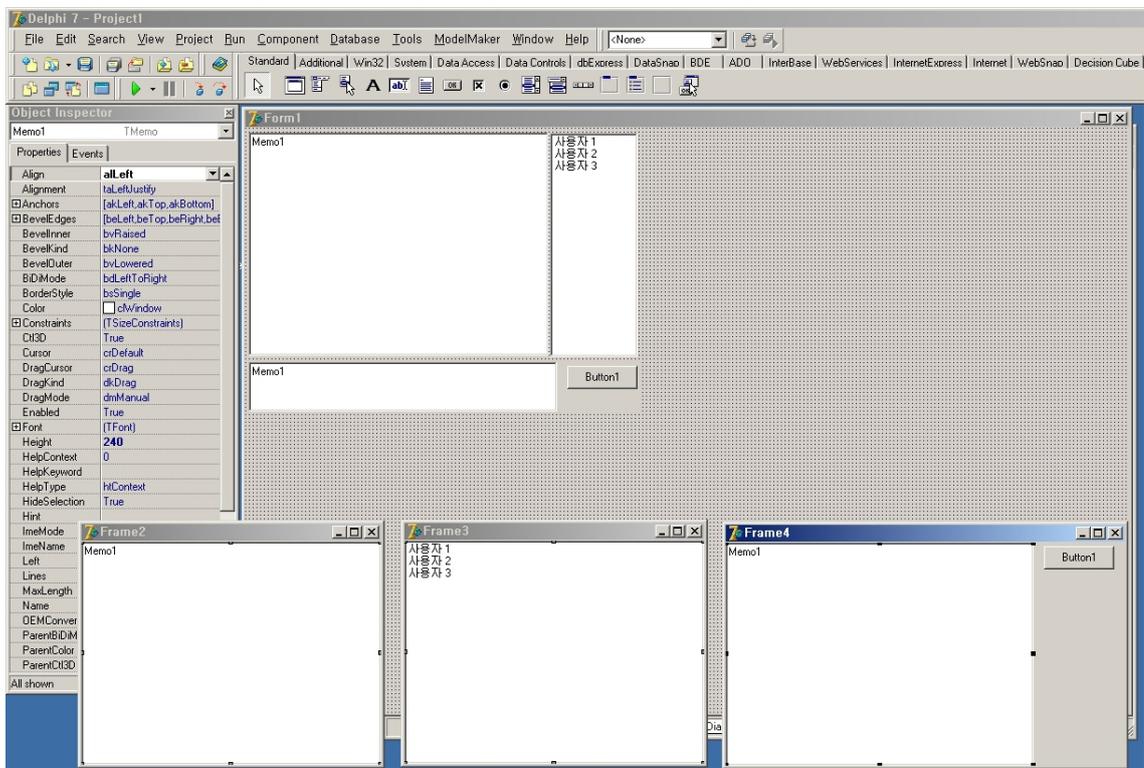


### 화면의 분할 설계

프로그램을 짜다 보면 한 폼에 여러가지 기능이 그룹으로 나누어질 때가 있습니다. 이런 때 TFrame 컴포넌트를 활용하면 보다 이해하기 쉬운 코드를 작성할 수가 있습니다.



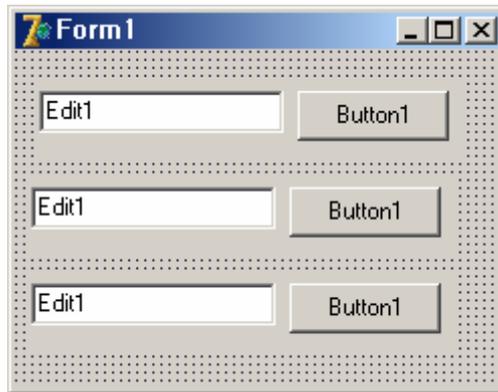
[그림 1] Frame 사용의 예

프레임을 이용할 경우에는 코드가 각각 프레임속에 숨겨지기 때문에 캡슐화의 장점을 이용하여 보다 쉽고 간결한 코드를 작성할 수 있습니다.

프레임은 폼을 이용해서 프로그래밍하는 것과 같은 방식으로 작업하시면 됩니다. 따라서 마치 하나의 폼을 여러 개의 폼으로 나누어서 작업하고 이를 합쳐서 화면에 표시해주는 것으로 생각하시면 됩니다.

### “새로운 컴포넌트 작성하기”의 대응책

프로젝트를 진행하다 보면 기존의 컴포넌트로는 부족하여 새로운 컴포넌트를 작성해야 할 경우가 종종 있습니다. 이럴 때 컴포넌트를 제작하고 등록하는 것은 상당히 번거롭습니다. 이럴 때 Frame을 컴포넌트 대신으로 사용할 수가 있습니다.



[그림 2] 프레임을 컴포넌트처럼 사용



[그림 3] 컴포넌트로 사용된 프레임

```
1 : unit Unit2;  
2 :  
3 : interface  
4 :  
5 : uses  
6 :   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
7 :   Dialogs, StdCtrls;  
8 :
```

```

9 : type
10 :   TFrame2 = class(TFrame)
11 :     Edit1: TEdit;
12 :     Button1: TButton;
13 :     procedure Button1Click(Sender: TObject);
14 :   private
15 :     { Private declarations }
16 :   public
17 :     { Public declarations }
18 :   end;
19 :
20 : implementation
21 :
22 :   {$R *.dfm}
23 :
24 :   procedure TFrame2.Button1Click(Sender: TObject);
25 :   begin
26 :     ShowMessage(Edit1.Text);
27 :   end;
28 :
29 : end.

```

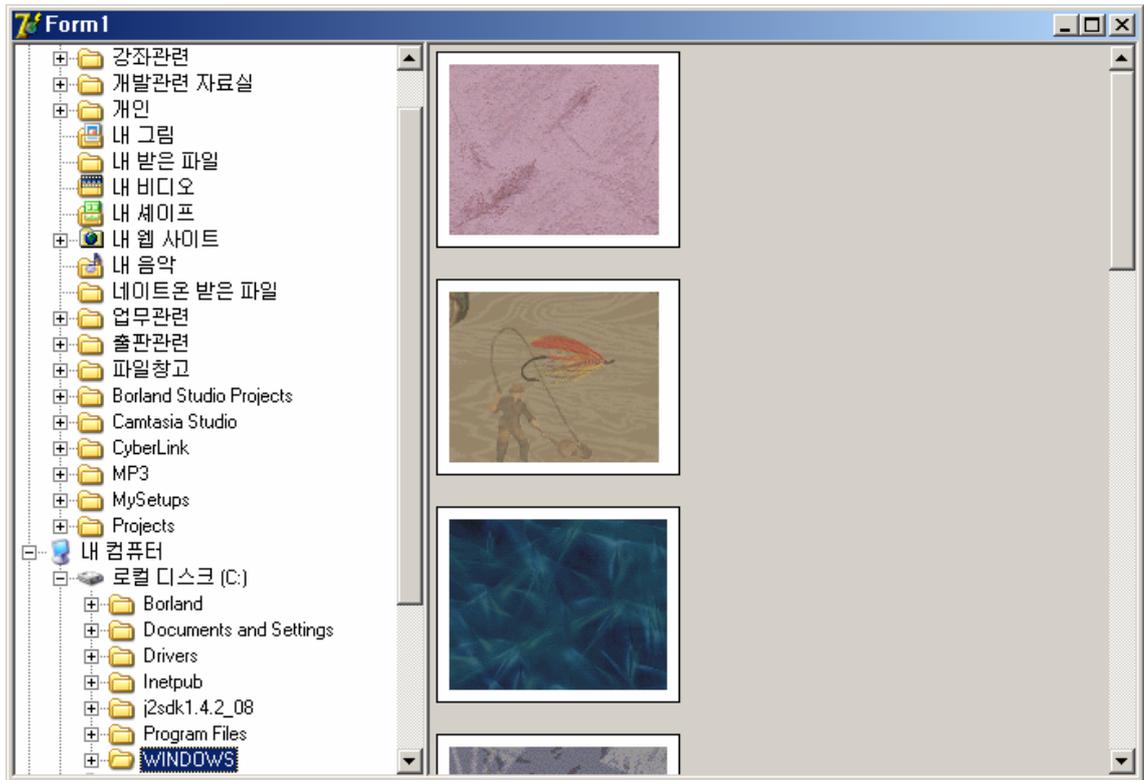
위의 소스에서 보시는 것처럼 프레임 안에 한 번만 코딩을 하시면 폼 위에 올려놓은 모든 프레임 오브젝트에 동시에 적용이됩니다. 이 프로그램을 실행한 후에 버튼 세 개를 클릭해 보면 버튼 왼쪽에 있는 TEdit 컴포넌트에 입력된 값이 화면에 표시됩니다.

## Image Viwer 만들기

---

간단한 Image Viewer를 통해서 TFrame을 활용하는 프로그램을 작성해 보도록 하겠습니다. 우선 Frame을 부분적으로 적용한 예제와 Frame을 적용한 경우를 비교해서 코딩이 얼마나 간결해지는 지 확인해 보도록 하겠습니다. 만약 전혀 Frame을 사용하지 않은 경우라면 더욱 심난한 코딩이 되겠지요.

우선 실행화면은 [그림 4]와 같습니다.



[그림 4] Image Viewer 실행의 예

일단 메인 폼의 소스만 살펴보도록 하겠습니다.

```

1 : unit Unit1;
2 :
3 : interface
4 :
5 : uses
6 :   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
7 :   Dialogs, StdCtrls, ComCtrls, ShellCtrls, frmImageList;
8 :
9 : type
10 :   TForm1 = class(TForm)
11 :     ShellTreeView1: TShellTreeView;
12 :     ScrollBox1: TScrollBox;
13 :     procedure ShellTreeView1Change(Sender: TObject; Node: TTreeNode);
14 :     procedure FormCreate(Sender: TObject);
15 :     procedure FormDestroy(Sender: TObject);
16 :   private

```

```
17 :      { Private declarations }
18 :      FList : TList;
19 :      Procedure GetFiles(Path:String);
20 :      Procedure Clear;
21 :      Procedure AddFile(FileName:String);
22 :      public
23 :      { Public declarations }
24 :      end;
25 :
26 :      var
27 :      Form1: TForm1;
28 :
29 :      implementation
30 :
31 :      Uses
32 :      frmImage;
33 :
34 :      {$R *.dfm}
35 :
36 :      Procedure TForm1.GetFiles(Path:String);
37 :      var
38 :      Di : TSearchRec;
39 :      Found : Integer;
40 :      Begin
41 :      Clear;
42 :      Found:= FindFirst(Path+'\*.*', faDirectory, Di);
43 :      While Found = 0 do
44 :      Begin
45 :      If ((Di.Attr and faDirectory) <> faDirectory) then Begin
46 :      AddFile(Path+'\'+Di.Name);
47 :      End;
48 :
49 :      Found:= FindNext(Di);
50 :      End;
51 :
52 :      FindClose(Di);
```

```
53 : End;
54 :
55 : procedure TForm1.ShellTreeView1Change(Sender: TObject; Node: TTreeNode);
56 : begin
57 :   If Node = Nil then Exit;
58 :
59 :   GetFiles(ShellTreeView1.SelectedFolder.PathName);
60 : end;
61 :
62 : procedure TForm1.FormCreate(Sender: TObject);
63 : begin
64 :   FList:= TList.Create;
65 :   ShellTreeView1.OnChange:= ShellTreeView1Change;
66 : end;
67 :
68 : procedure TForm1.FormDestroy(Sender: TObject);
69 : begin
70 :   FList.Free;
71 : end;
72 :
73 : Procedure TForm1.Clear;
74 : var
75 :   Loop : Integer;
76 : Begin
77 :   For Loop:= 0 to FList.Count-1 do
78 :     TfrImage(FList.Items[Loop]).Free;
79 :   FList.Clear;
80 : End;
81 :
82 : Procedure TForm1.AddFile(FileName:String);
83 : var
84 :   frImage : TfrImage;
85 : Begin
86 :   If LowerCase(ExtractFileExt(FileName)) <> '.bmp' then Exit;
87 :
88 :   frImage:= TfrImage.Create(Self);
```

```

89 :   frImage.Name:= 'Name'+IntToStr(FList.Count);
90 :   frImage.Parent:= ScrollBox1;
91 :   frImage.Top:= FList.Count*140;
92 :   frImage.LoadFromFile(FileName);
93 :   FList.Add(frImage);
94 : End;
95 :
96 : end.

```

소스 코드에 다양한 기능이 함께 어울려져 있기 때문에 상당히 읽기가 난해한 코드가 되어 있습니다. 이것을 프레임을 활용해서 만든 경우와 비교해 보도록 하겠습니다.

프레임을 적용한 경우의 메인 폼 소스

```

1 : unit Unit1;
2 :
3 : interface
4 :
5 : uses
6 :   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
7 :   Dialogs, StdCtrls, ComCtrls, ShellCtrls, frmImageList, frmFolder;
8 :
9 : type
10 :   TForm1 = class(TForm)
11 :     frImageList: TfrImageList;
12 :     frFolder1: TfrFolder;
13 :   private
14 :     { Private declarations }
15 :   public
16 :     { Public declarations }
17 :   end;
18 :
19 : var
20 :   Form1: TForm1;
21 :
22 : implementation
23 :

```

```
24 : {$R *.dfm}
25 :
26 : end.
```

#### frmFolder.pas 소스

```
1 : unit frmFolder;
2 :
3 : interface
4 :
5 : uses
6 :   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
7 :   Dialogs, ComCtrls, ShellCtrls;
8 :
9 : type
10 :   TfrFolder = class(TFrame)
11 :     ShellTreeView1: TShellTreeView;
12 :     procedure ShellTreeView1Change(Sender: TObject; Node: TTreeNode);
13 :   private
14 :     { Private declarations }
15 :     Procedure GetFiles(Path:String);
16 :   public
17 :     { Public declarations }
18 :   end;
19 :
20 : implementation
21 :
22 : uses
23 :   Unit1;
24 :
25 : {$R *.dfm}
26 :
27 : Procedure TfrFolder.GetFiles(Path:String);
28 : var
29 :   Di : TSearchRec;
30 :   Found : Integer;
31 : Begin
```

```

32 : TForm1(Parent).frImageList.Clear;
33 : Found:= FindFirst(Path+'\*.*', faDirectory, Di);
34 : While Found = 0 do
35 :     Begin
36 :         If ((Di.Attr and faDirectory) <> faDirectory) then Begin
37 :             TForm1(Parent).frImageList.AddFile(Path+'\'+Di.Name);
38 :         End;
39 :
40 :         Found:= FindNext(Di);
41 :     End;
42 :
43 :     FindClose(Di);
44 : End;
45 :
46 : procedure TfrFolder.ShellTreeView1Change(Sender: TObject; Node: TTreeNode);
47 : begin
48 :     If Node = Nil then Exit;
49 :
50 :     GetFiles(ShellTreeView1.SelectedFolder.PathName);
51 : end;
52 :
53 : end.

```

frImageList.pas의 소스

```

1 : unit frmImageList;
2 :
3 : interface
4 :
5 : uses
6 :     Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
7 :     Dialogs;
8 :
9 : type
10 : TfrImageList = class(TFrame)
11 : private
12 :     { Private declarations }

```

```
13 :     FList : TList;
14 :     public
15 :     { Public declarations }
16 :     Constructor Create(AOwner:TComponent); override;
17 :     Destructor Destroy; override;
18 :
19 :     Procedure Clear;
20 :     Procedure AddFile(FileName:String);
21 :     end;
22 :
23 : implementation
24 :
25 : Uses
26 :     frmImage;
27 :
28 :     {$R *.dfm}
29 :
30 : Constructor TfrImageList.Create(AOwner:TComponent);
31 : Begin
32 :     Inherited Create(AOwner);
33 :
34 :     FList:= TList.Create;
35 : End;
36 :
37 : Destructor TfrImageList.Destroy;
38 : Begin
39 :     FList.Free;
40 :
41 :     Inherited Destroy;
42 : End;
43 :
44 : Procedure TfrImageList.Clear;
45 : var
46 :     Loop : Integer;
47 : Begin
48 :     For Loop:= 0 to FList.Count-1 do
```

```

49 :   TfrImage(FList.Items[Loop]).Free;
50 :   FList.Clear;
51 : End;
52 :
53 : Procedure TfrImageList.AddFile(FileName:String);
54 : var
55 :   frImage : TfrImage;
56 : Begin
57 :   If LowerCase(ExtractFileExt(FileName)) <> '.bmp' then Exit;
58 :
59 :   frImage:= TfrImage.Create(Self);
60 :   frImage.Name:= 'Name'+IntToStr(FList.Count);
61 :   frImage.Parent:= Self;
62 :   frImage.Top:= FList.Count*140;
63 :   frImage.LoadFromFile(FileName);
64 :   FList.Add(frImage);
65 : End;
66 :
67 : end.

```

프레임을 사용한 경우에는 위와 같이 각각 프레임마다 기능을 분업화하여 해당 프레임이 맡은 바 임무에 대해서만 생각할 수 있도록 코드가 분리되어 있습니다. 정리정돈이 잘된 책상처럼 코드를 읽고 이해하고 그리고 수정/변경이 쉬운 장점이 생겼습니다.

frFolder 프레임은 PC의 폴더를 트리구조로 보여주고 어느 폴더를 선택하면 해당 폴더에 있는 파일만 찾아서 frImageList에 알려주는 기능만 담당합니다.

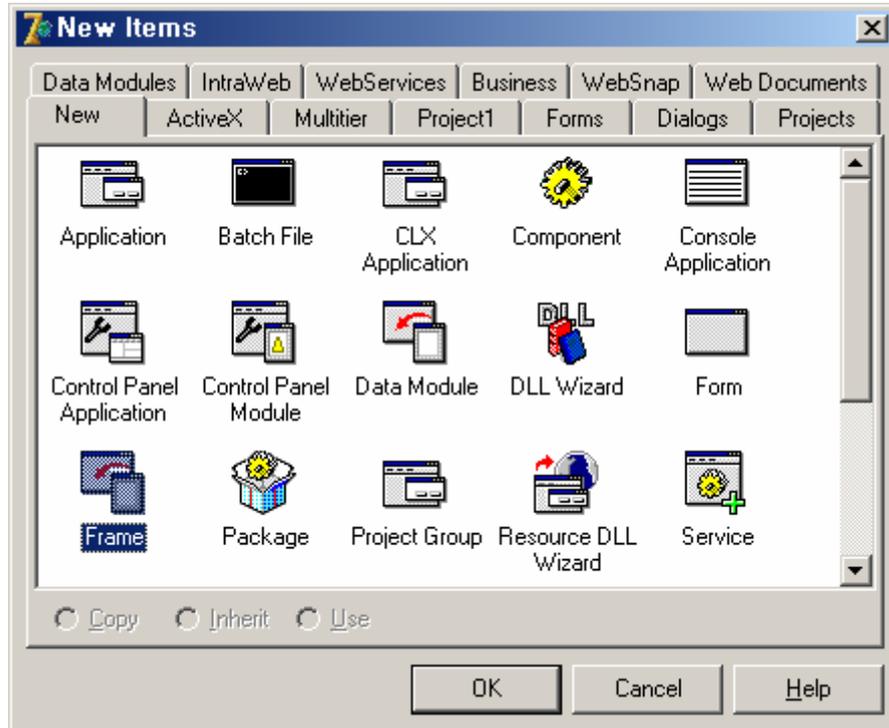
frImageList 프레임은 파일이름이 입력되면 화면에 이미지를 일렬로 늘어트려서 정렬해주는 기능만 담당합니다.

frImage 프레임은 각각의 이미지를 화면에 표시해주는 기능만 담당합니다.

이처럼 코드그 여러 파일로 나누어서 각자의 임무에만 전념하는 분업체계가 되어, 코드를 읽기 쉽고 변경이 쉬운 상태로 변경하였습니다.

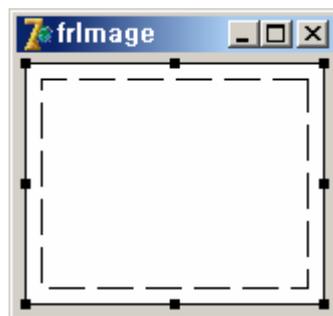
## 프로그래밍 실습

메인 메뉴에서 “File -> New -> Application” 을 실행합니다. 이어서 아래와 같이 프레임 추가합니다.



[그림 5]

프레임 위에 TShape와 TImage 컴포넌트를 각각 내려놓고 소스박스에 있는 것처럼 프로퍼티를 설정하여 줍니다. 이 프레임은 개별 이미지를 화면에 표시하여 주는 기능을 담당합니다.



[그림 6]

```
1 : object frImage: TfrImage
2 :   Width = 157
3 :   Height = 129
```

```

4 : object Shap1: TShape
5 :   Left = 4
6 :   Top = 4
7 :   Width = 149
8 :   Height = 121
9 : end
10 : object Imagel: TImage
11 :   Left = 12
12 :   Top = 12
13 :   Width = 133
14 :   Height = 105
15 : end
16 : end

```

이어서 F12 버튼을 클릭하고 아래의 소스박스처럼 Interface 부분에 코딩하여 줍니다.

```

1 : type
2 :   TfrImage = class(TFrame)
3 :     Imagel: TImage;
4 :     Shap1: TShape;
5 :   private
6 :     { Private declarations }
7 :     FSelected : Boolean;
8 :     Procedure SetSelected(Value:Boolean);
9 :   public
10 :    { Public declarations }
11 :    Procedure LoadFromFile(FileName:String);
12 :    Property Selected : Boolean read FSelected write SetSelected;
13 : end;

```

이어서 implementation 부분에도 코딩을 추가하여 줍니다.

```

1 : implementation
2 :
3 :   {SR *.dfm}
4 :
5 :   Procedure TfrImage.LoadFromFile(FileName:String);
6 :   Begin

```

```

7 : Image1.Picture.LoadFromFile(FileName);
8 : End;
9 :
10 : Procedure TfrImage.SetSelected(Value:Boolean);
11 : Begin
12 :   FSelected:= Value;
13 :   If Value = True then Shapel.Brush.Color:= clNavy
14 :   Else Shapel.Brush.Color:= clWhite;
15 : End;

```

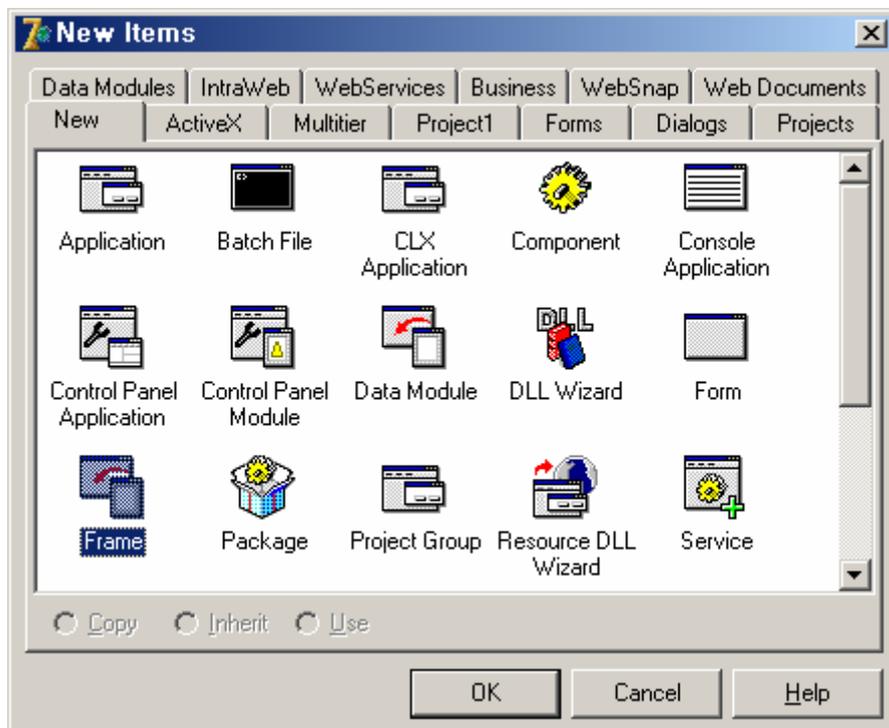
F12를 다시 클릭하시고 Image1 컴포넌트를 선택하신 후에 오브젝트 인스펙터에서 OnClick을 더블클릭 하신 후에 아래와 같이 코딩하여 줍니다.

```

1 : procedure TfrImage.Image1Click(Sender: TObject);
2 : begin
3 :   Selected:= not Selected;
4 : end;

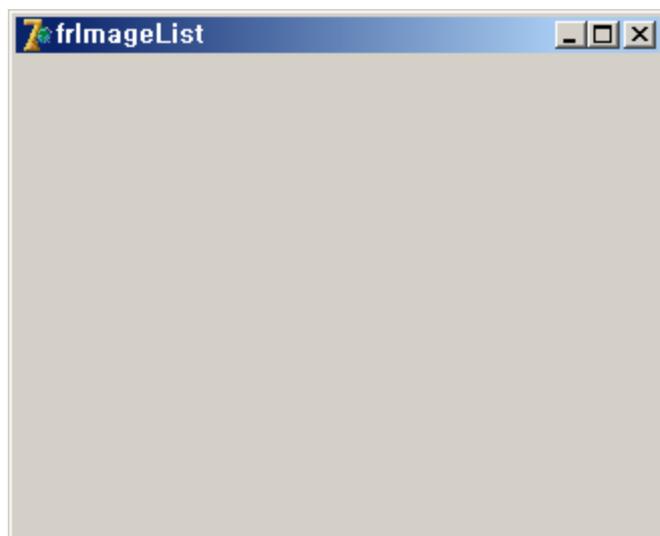
```

새로 프레임 하나를 더 추가합니다.



[그림 7]

이번에 생성된 프레임에서는 프로퍼티 설정을 변경하실 필요는 없습니다. 이 프레임은 이미지를 순서대로 나열하는 기능을 담당하게 됩니다.



[그림 8]

이어서 F12를 누르고 아래의 소스박스처럼 코딩하여 줍니다.

```
1 : unit frmImageList;  
2 :  
3 : interface  
4 :  
5 : uses  
6 :   Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
7 :   Dialogs;  
8 :  
9 : type  
10 :   TfrmImageList = class(TFrame)  
11 :   private  
12 :     { Private declarations }  
13 :     FList : TList;  
14 :   public  
15 :     { Public declarations }  
16 :     Constructor Create(AOwner:TComponent); override;  
17 :     Destructor Destroy; override;  
18 :  
19 :     Procedure Clear;
```

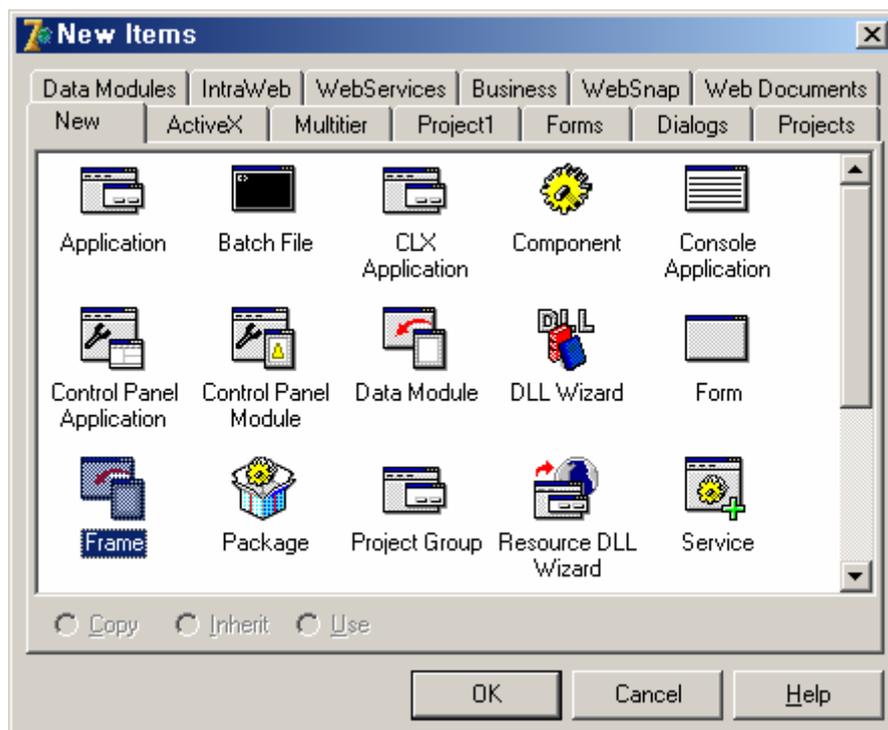
```
20 :   Procedure AddFile(FileName:String);
21 :   end;
22 :
23 : implementation
24 :
25 : Uses
26 :   frmImage;
27 :
28 :   {$R * .dfm}
29 :
30 : Constructor TfrImageList.Create(AOwner:TComponent);
31 : Begin
32 :   Inherited Create(AOwner);
33 :
34 :   FList:= TList.Create;
35 : End;
36 :
37 : Destructor TfrImageList.Destroy;
38 : Begin
39 :   FList.Free;
40 :
41 :   Inherited Destroy;
42 : End;
43 :
44 : Procedure TfrImageList.Clear;
45 : var
46 :   Loop : Integer;
47 : Begin
48 :   For Loop:= 0 to FList.Count-1 do
49 :     TfrImage(FList.Items[Loop]).Free;
50 :   FList.Clear;
51 : End;
52 :
53 : Procedure TfrImageList.AddFile(FileName:String);
54 : var
55 :   frImage : TfrImage;
```

```

56 : Begin
57 : If LowerCase(ExtractFileExt(FileName)) <> '.bmp' then Exit;
58 :
59 : frImage:= TfrImage.Create(Self);
60 : frImage.Name:= 'Name'+IntToStr(FList.Count);
61 : frImage.Parent:= Self;
62 : frImage.Top:= FList.Count*140;
63 : frImage.LoadFromFile(FileName);
64 : FList.Add(frImage);
65 : End;
66 :
67 : end.

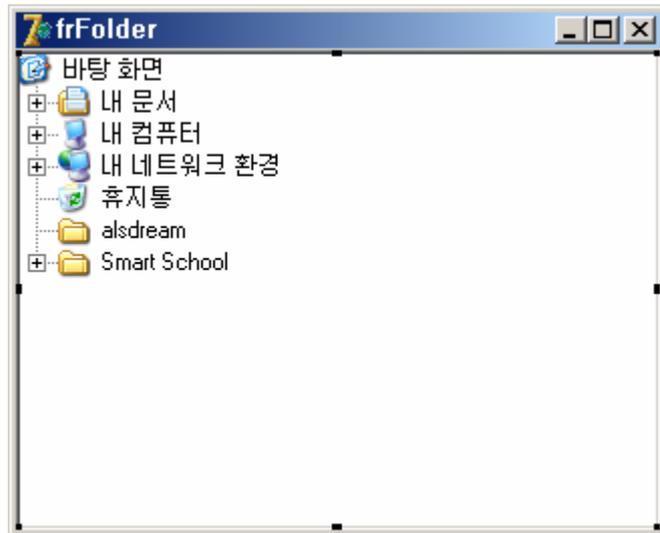
```

이어서 프레임 하나를 더 추가합니다. 이 프레임은 폴더를 선택하고 해당 폴더에 있는 파일 목록을 TfrImageList에 넘겨주는 역할을 합니다.



[그림 9]

[그림 10]과 같이 TShellTreeView 컴포넌트를 올려놓고 소스박스에서처럼 프로퍼티를 설정하여 줍니다.



[그림 10]

```

1 : object frFolder: TfrFolder
2 :   object ShellTreeView1: TShellTreeView
3 :     Align = alClient
4 :   end
5 : end

```

이어서 F12 버튼을 누르고 아래와 같이 코딩하여 줍니다.

```

1 : type
2 :   TfrFolder = class(TFrame)
3 :     ShellTreeView1: TShellTreeView;
4 :   private
5 :     { Private declarations }
6 :     Procedure GetFiles(Path:String);
7 :   public
8 :     { Public declarations }
9 :   end;

```

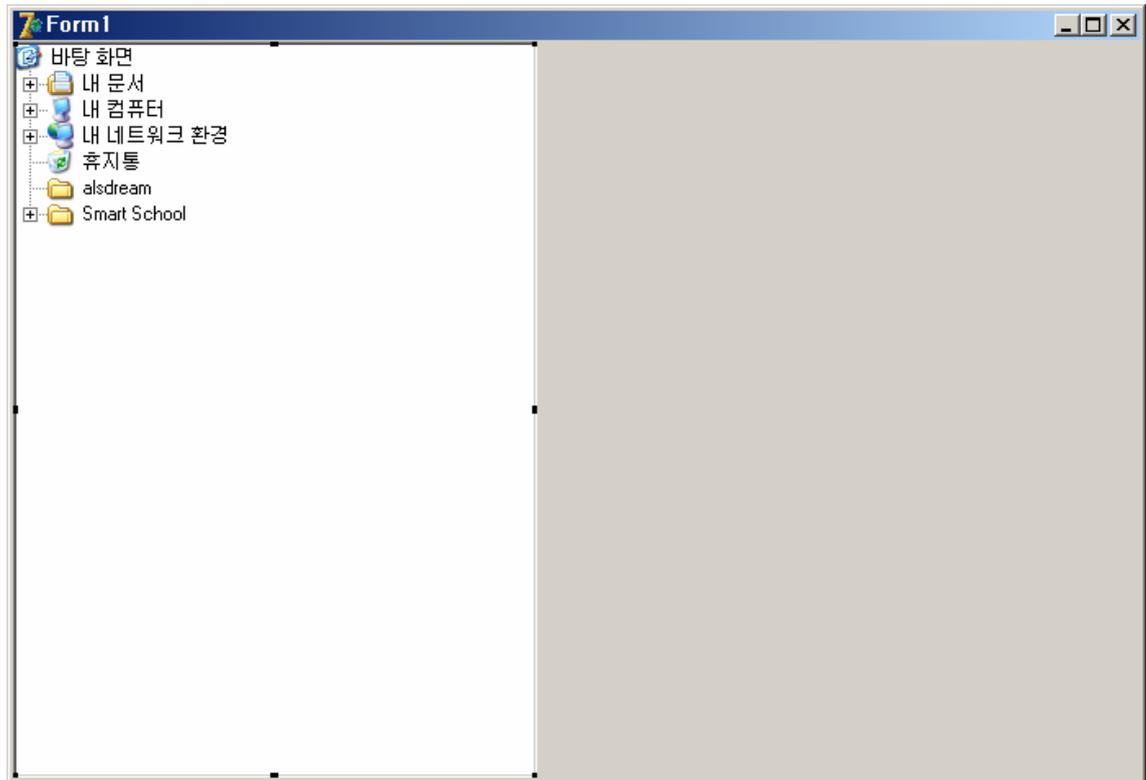
TShellTreeView의 OnChange 이벤트를 더블 클릭하시고 아래 소스박스처럼 코딩하여 줍니다.

```

1 : implementation
2 :
3 : uses

```

```
4 : Unit1;
5 :
6 : {$R *.dfm}
7 :
8 : Procedure TfrFolder.GetFiles(Path:String);
9 : var
10 : Di : TSearchRec;
11 : Found : Integer;
12 : Begin
13 : TForm1(Parent).frImageList.Clear;
14 : Found:= FindFirst(Path+'\\*.*', faDirectory, Di);
15 : While Found = 0 do
16 :     Begin
17 :         If ((Di.Attr and faDirectory) <> faDirectory) then Begin
18 :             TForm1(Parent).frImageList.AddFile(Path+'\\'+Di.Name);
19 :         End;
20 :
21 :         Found:= FindNext(Di);
22 :     End;
23 :
24 :     FindClose(Di);
25 : End;
26 :
27 : procedure TfrFolder.ShellTreeView1Change(Sender: TObject; Node: TTreeNode);
28 : begin
29 :     If Node = Nil then Exit;
30 :
31 :     GetFiles(ShellTreeView1.SelectedFolder.PathName);
32 : end;
```



[그림 11] 메인화면

메인 폼에는 TfrFolder와 TfrImageList를 내려 놓고 아래와 같이 프로퍼티만 설정하여 주면 됩니다.

```
1 : object Form1: TForm1
2 :   inline frFolder1: TfrFolder
3 :     Width = 320
4 :     Align = alLeft
5 :   end
6 :   inline frImageList: TfrImageList
7 :     Align = alClient
8 :   end
9 : end
```

보다 자세한 내용은 <http://cafe.naver.com/codeway>에서 동영상 자료를 참고하여 주시기 바랍니다.