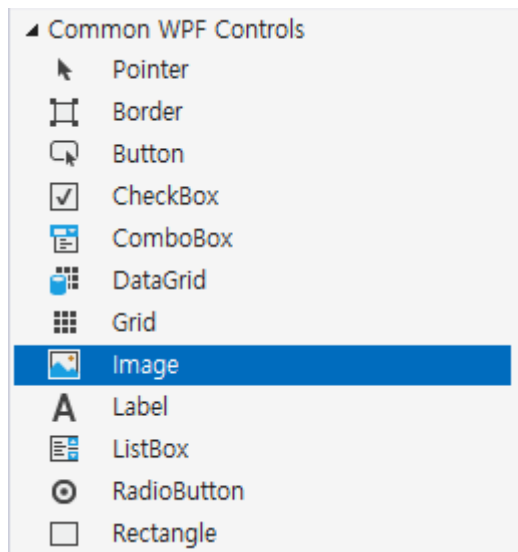
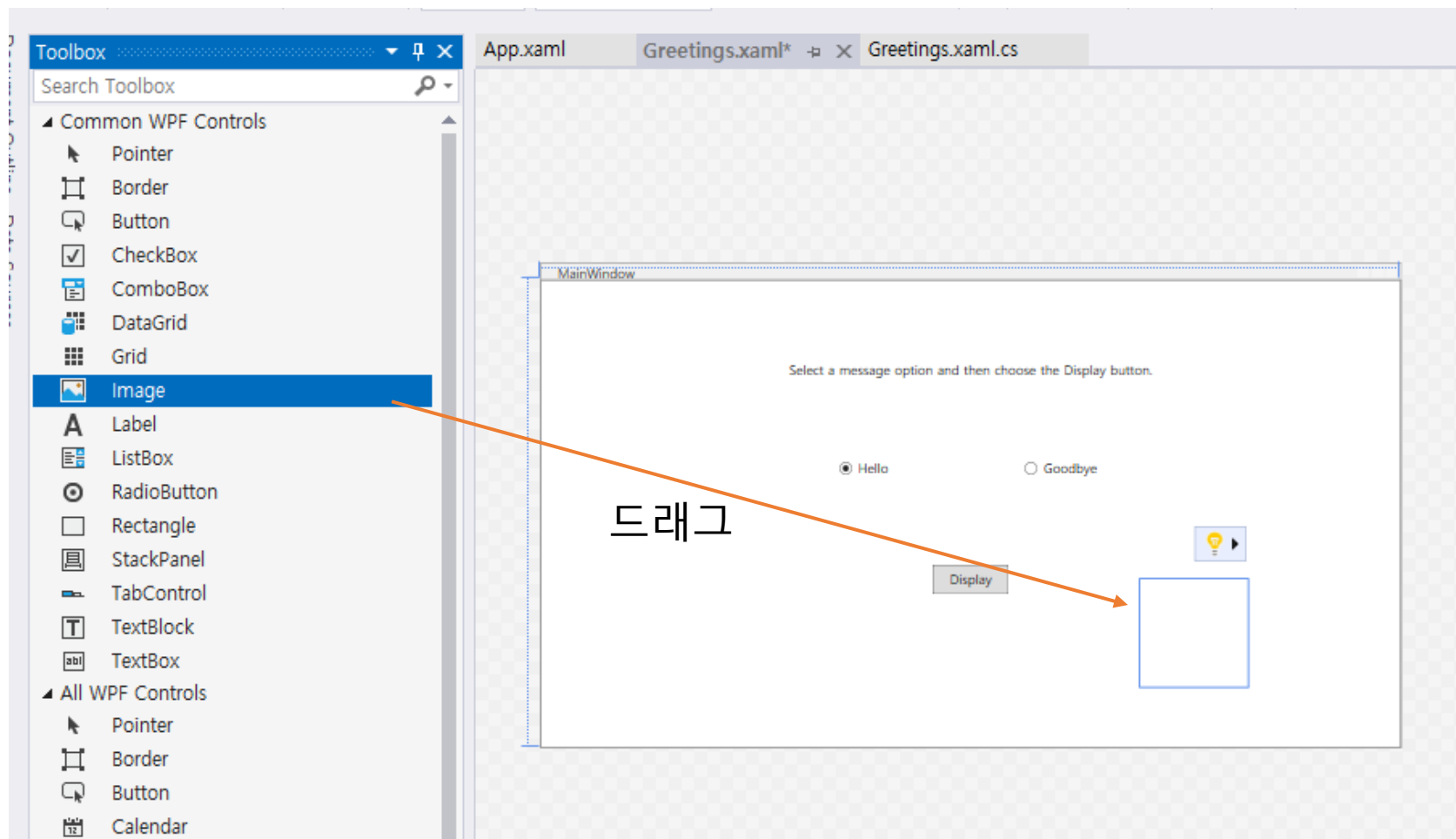


도구상자 - 심화

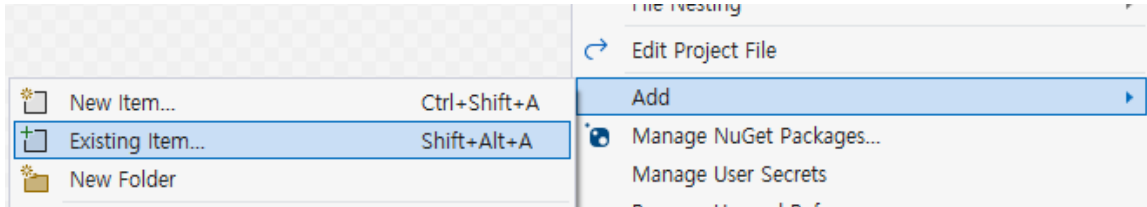
도구상자 - 이미지



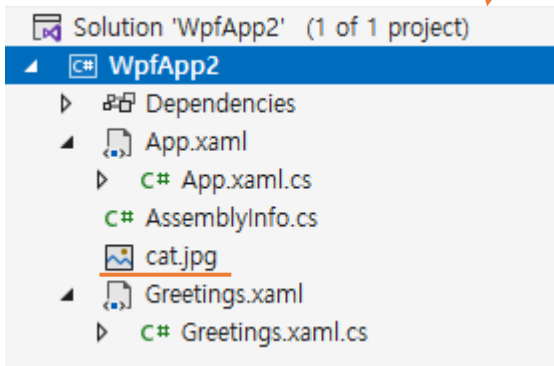
이미지를 표시하는데 사용



도구상자 - 이미지



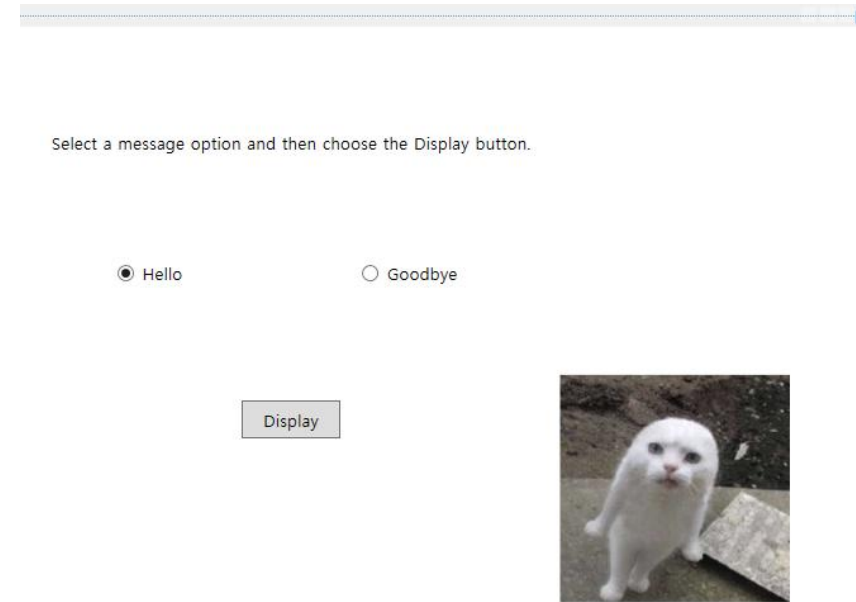
프로젝트 우클릭 > Add > Existing Item



이미지 파일 선택 시 프로젝트에 추가됨

```
<Button x:Name="button" Content="Display" HorizontalAlign="Center" Width="100" Height="30" Click="Display_Click" />
<Image x:Name="image" Source="cat.jpg" HorizontalAlignment="Center" Width="100" Height="100" />
```

이미지 태그의 Source를 이미지 파일 이름으로 수정



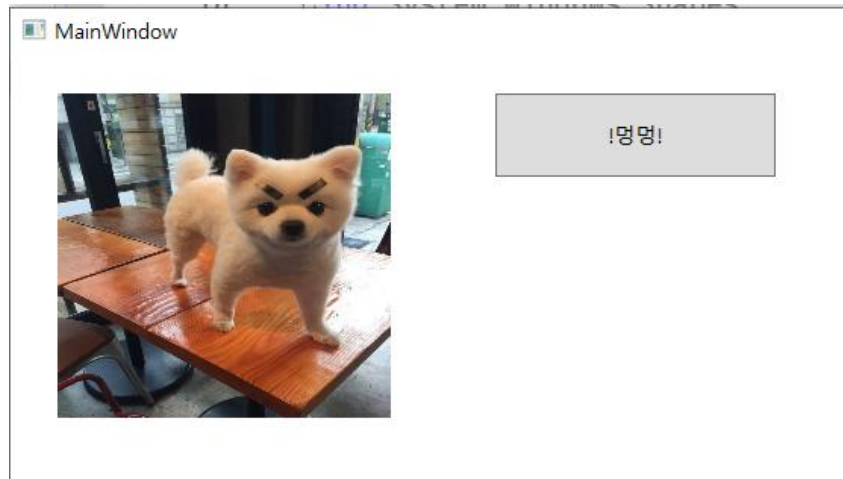
이미지가 추가됨

도구상자 - 이미지

- 코드를 사용하여 이미지 변경

1 reference

```
private void Btn1_Click(object sender, RoutedEventArgs e)
{
    var uriSource = new Uri(@"/WpfApp3;component/Resources/dog.jpg", UriKind.Relative);
    ImageBox1.Source = new BitmapImage(uriSource);
}
```



번외_ URI 란?

- Uniform Resource Identifier 의 약자
 - URL은 Uniform Resource **Locator**의 약자로 URI의 위치를 나타냄
- 프로그램 개발에 필요한 자원을 (주로 파일)을 "**식별**" 함

```
Uri siteUri = new Uri("http://www.contoso.com/");
```

```
Uri fileUri = new Uri("file:///C:/test/path/file.txt")
```

* 공식 매뉴얼 설명

<https://learn.microsoft.com/ko-kr/dotnet/api/system.uri?view=net-7.0>

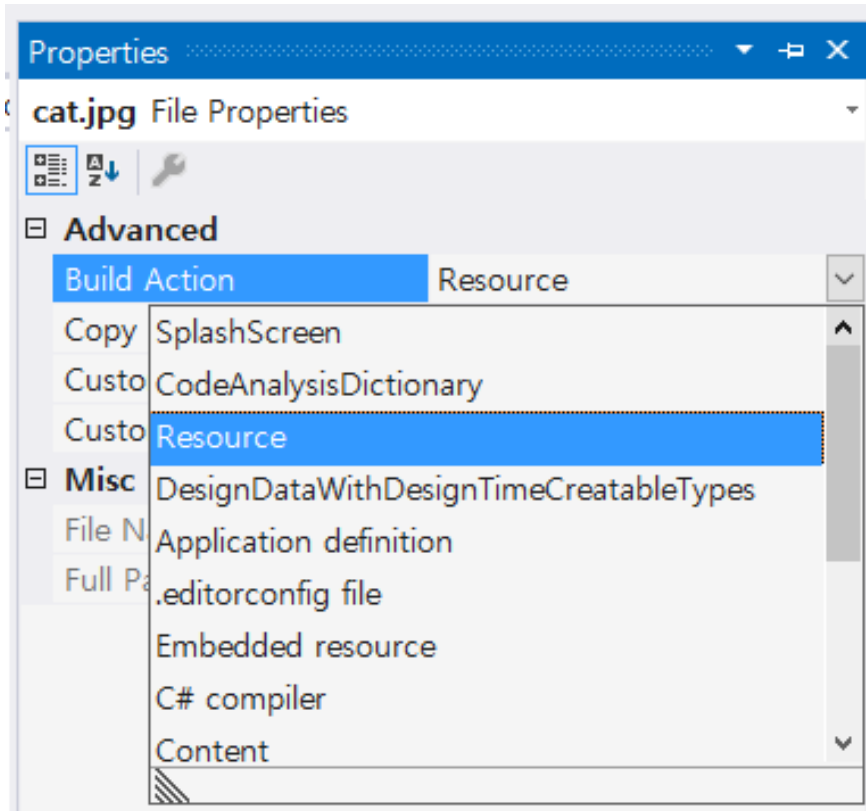
번외_ URI 란?

- 상대 경로, 절대 경로 또는 둘 다 사용 세 가지 조건을 옵션으로 줄 수 있음
- 파일 경로의 경우 옵션이 반드시 필요함
- 후술할 pack uri를 사용하는 것이 가장 좋음

내 프로젝트 이름

```
var uriSource1 = new Uri(@"WpfApp3;component/Resources/dog.jpg", UriKind.Relative);  
var uriSource2 = new Uri(@"D:/Source/WpfApp3/WpfApp3/Resources/dog.jpg", UriKind.Absolute);  
var uriSource3 = new Uri(@"WpfApp3;component/Resources/dog.jpg", UriKind.RelativeOrAbsolute);  
ImageBox1.Source = new BitmapImage(uriSource2);
```

번외_ Build Action 이란?



* 공식 매뉴얼

https://learn.microsoft.com/ko-kr/visualstudio/ide/build-actions?view=vs-2022&WT.mc_id=DT-MVP-5000651

- None
 - 실행 파일 또는 DLL에 포함되지 않음
- Embedded Resource
 - DLL에 포함됨
 - GetManifestResourceStream으로 불러옴
- Resource
 - DLL에 포함됨
 - URI를 통해 불러옴

* 참고 자료

<https://kaki104.tistory.com/878>

번외_ 버튼 이미지 바꾸기

- C# 코드에서 변경하기

2 references

```
public partial class MainWindow : Window  
{
```

```
    bool btn_image_cat = true;  
    Uri uriCatImage;  
    Uri uriDogImage;
```

0 references

```
public MainWindow()  
{
```

```
    InitializeComponent();
```

```
    this.uriCatImage = new Uri(@"pack://application:,,,/Resources/cat.jpg", UriKind.Absolute);  
    this.uriDogImage = new Uri(@"pack://application:,,,/Resources/dog.jpg", UriKind.Absolute);  
}
```


번외_ 버튼 이미지 바꾸기

1 reference

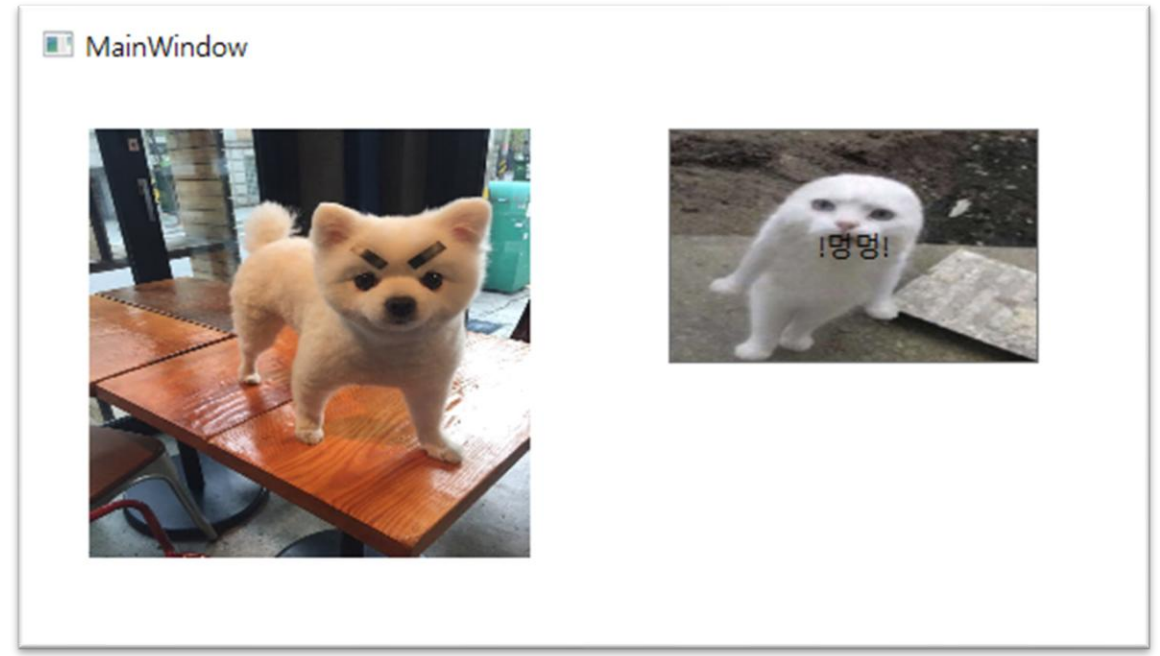
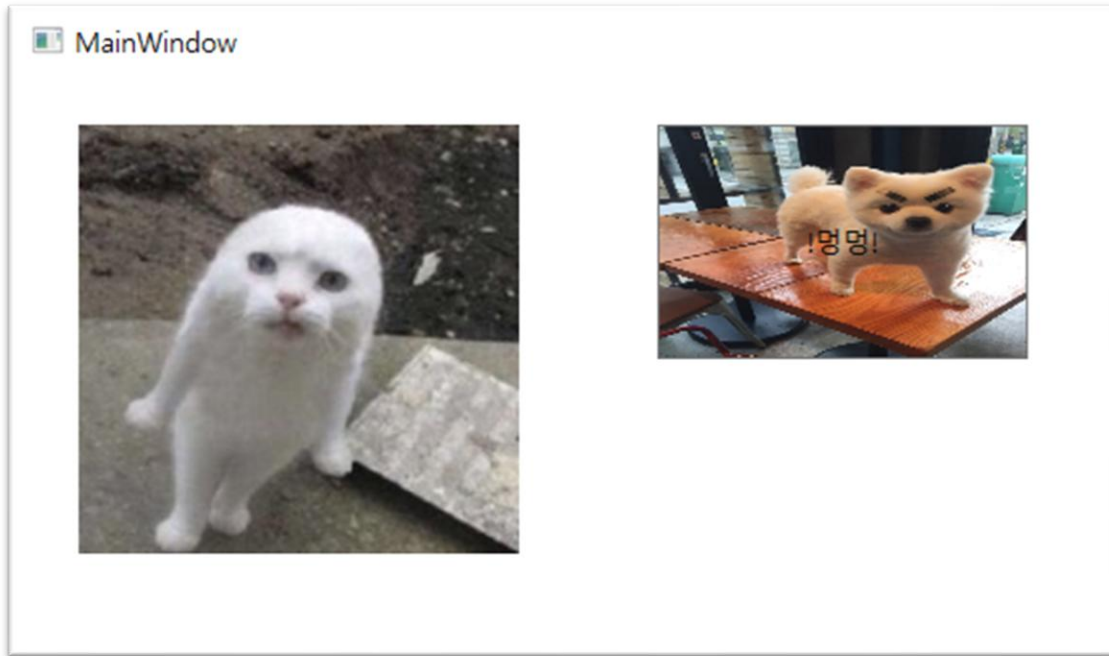
```
private void Btn1_Click(object sender, RoutedEventArgs e)
{
    if(this.btn_image_cat)
    {
        Btn1.Background = new ImageBrush(new BitmapImage(this.uriCatImage));
        ImageBox1.Source = new BitmapImage(this.uriDogImage);
        this.btn_image_cat = false;
    }
    else
    {
        Btn1.Background = new ImageBrush(new BitmapImage(this.uriDogImage));
        ImageBox1.Source = new BitmapImage(this.uriCatImage);
        this.btn_image_cat = true;
    }
}
```

번외_ 버튼 이미지 바꾸기

- xaml 코드에서 변경하기

```
<Grid>
    <Image x:Name="ImageBox1" Margin="31,26,533,171" Source="Resources/cat.jpg" Stretch="Fill"/>
    <Button x:Name="Btn1" Content="!멍멍!" HorizontalAlignment="Left" VerticalAlignment="Top"
        Margin="320,26,0,0" Height="117" Width="185" Click="Btn1_Click">
        <Button.Background>
            <ImageBrush ImageSource="pack://application:,,,/Resources/dog.jpg"/>
        </Button.Background>
    </Button>
</Grid>
```

번외_ 버튼 이미지 바꾸기



실습. 이미지 버튼

- 이미지 파일을 불러오면 버튼이 불러온 이미지로 바뀌는 프로그램 제작
 - OpenFileDialog를 구현
 - 파일 경로는 URI를 사용하여 가져오기
 - 버튼의 크기도 이미지 크기와 같게 변경되도록 하기.
-
- Git Repo 링크 업로드

도구상자 – DataGrid

Toolbox controls:

- CheckBox
- ComboBox
- ContentControl
- DataGrid**
- DatePicker
- DockPanel
- DocumentViewer

Application Window:

File path

Open

SampleStringA	SampleStringB	SampleBool
Sample String A - 1	Sample String B - 1	<input checked="" type="checkbox"/>
Sample String A - 2	Sample String B - 2	<input checked="" type="checkbox"/>
Sample String A - 3	Sample String B - 3	<input checked="" type="checkbox"/>
Sample String A - 4	Sample String B - 4	<input checked="" type="checkbox"/>
Sample String A - 5	Sample String B - 5	<input checked="" type="checkbox"/>

표(DataGrid) 형태의 데이터를 표현하기 위한 도구

도구상자 - DataGrid

- XAML 코드로 헤더 데이터(컬럼) 추가하기

MainWindow				
이름	학번	학과	학년	비고

```
<DataGrid x:Name="StudentList">
    <DataGrid.Columns>
        <DataGridTextColumn Header="이름" Binding="{Binding Path=name}" Width="130"/>
        <DataGridTextColumn Header="학번" Binding="{Binding Path=id}" Width="130"/>
        <DataGridTextColumn Header="학과" Binding="{Binding Path=major}" Width="180"/>
        <DataGridTextColumn Header="학년" Binding="{Binding Path=grade}" Width="100"/>
        <DataGridTextColumn Header="비고" Binding="{Binding Path=etc}" Width="*/>
    </DataGrid.Columns>
</DataGrid>
```

도구상자 - DataGrid

- AutoGenerateColumns 옵션
 - 입력값: True(기본값), False
 - DataGrid로 데이터가 입력될 때 자동으로 컬럼을 생성해주는 기능
- CanUserAddRows 옵션
 - 입력값: True(기본값), False
 - 사용자가 표에 직접 컬럼을 추가할 수 있게 해주는 기능

```
<DataGrid x:Name="StudentList" AutoGenerateColumns="False" CanUserAddRows="False">
```

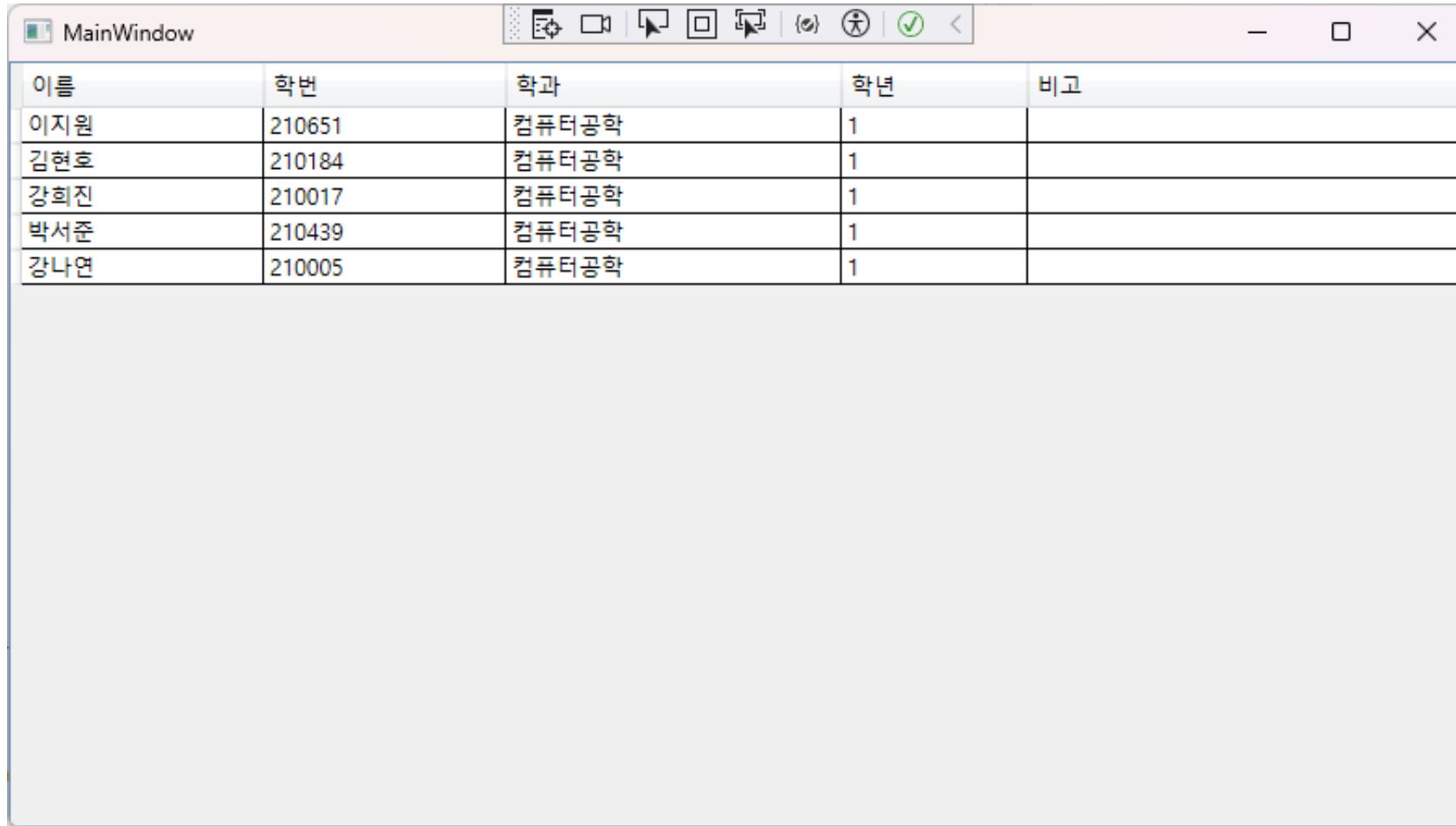
도구상자 - DataGridView

- C#으로 데이터 추가하기
 - List 생성후 DataGridView에 추가
 - List에 들어갈 데이터는 클래스, 구조체 등

```
0 references
public class Data
{
    0 references
    public string name { get; set; }
    0 references
    public string id { get; set; }
    0 references
    public string major { get; set; }
    0 references
    public int grade { get; set; }
    0 references
    public string etc { get; set; }
}
```

```
List<Data> list = new List<Data>();
list.Add(new Data { name = "이지원", id = "210651", major = "컴퓨터공학", grade = 1, etc = "" });
list.Add(new Data { name = "김현호", id = "210184", major = "컴퓨터공학", grade = 1, etc = "" });
list.Add(new Data { name = "강희진", id = "210017", major = "컴퓨터공학", grade = 1, etc = "" });
list.Add(new Data { name = "박서준", id = "210439", major = "컴퓨터공학", grade = 1, etc = "" });
list.Add(new Data { name = "강나연", id = "210005", major = "컴퓨터공학", grade = 1, etc = "" });
StudentList.ItemsSource = list; // List를 DataGridView에 리스트 추가
```


도구상자 - DataGrid



이름	학번	학과	학년	비고
이지원	210651	컴퓨터공학	1	
김현호	210184	컴퓨터공학	1	
강희진	210017	컴퓨터공학	1	
박서준	210439	컴퓨터공학	1	
강나연	210005	컴퓨터공학	1	

도구상자 - DataGridView

• DataGridView에서 원하는 값 가져오기

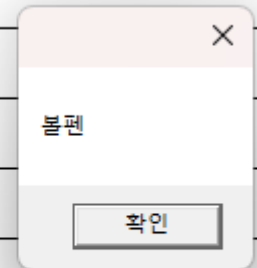
// 하나의 행만 가져오기

```
Person dataRow = (Person)DataGridView_Person.SelectedItem;
string data = dataRow.Name;
MessageBox.Show(data);
```

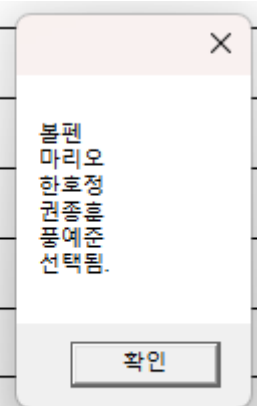
// 여러개의 행이 선택된 경우

```
string names = string.Empty;
foreach (Person person in DataGridView_Person.SelectedItems)
{
    names += person.Name + "\r\n";
}
MessageBox.Show(names + "선택됨.");
```

바나나	1	덜익음
볼펜	6	안나옴
마리오	40	외별 인생
한호정	15	AAA
권종훈	78	BBB
풍예준	54	CCC
추민호	65	DDD



바나나	1	덜익음
볼펜	6	안나옴
마리오	40	외별 인생
한호정	15	AAA
권종훈	78	BBB
풍예준	54	CCC
추민호	65	DDD



실습. DataGridView

- DataGridView, Button, Label을 사용하는 윈도우를 제작
- File Open 버튼으로 이름, 나이, 설명이 적힌 CSV 파일을 불러와서 내용을 DataGridView에 표시
 - OpenFileDialog를 이용할 것
 - CSV 파일을 가져오고 파싱(해석)할 때 try, catch 처리할 것
 - Label은 선택된 파일의 경로를 표시하는데 사용
- DataGridView에 있는 행을 선택 후 Get Data 버튼 클릭시 선택된 행의 이름을 출력하는 메시지박스 띄우기