

## 윈도우 폼 컨트롤 02

HyoJoon Han  
동국대학교  
han6343@dongguk.edu



픽처박스 컨트롤

파일 관련 컨트롤

스크롤바 컨트롤

타이머 컨트롤과 뉴메릭 업다운 컨트롤

체크리스트박스와 데이트타임피커 컨트롤

- 이미지를 화면에 나타낸다.
- 다양한 속성과 이벤트, 메서드를 지원하며 Image 속성이나 Bitmap() 메서드를 이용하여 그림을 보여준다.

표 7-2 픽처박스 컨트롤의 주요 속성

파란색은 기본값

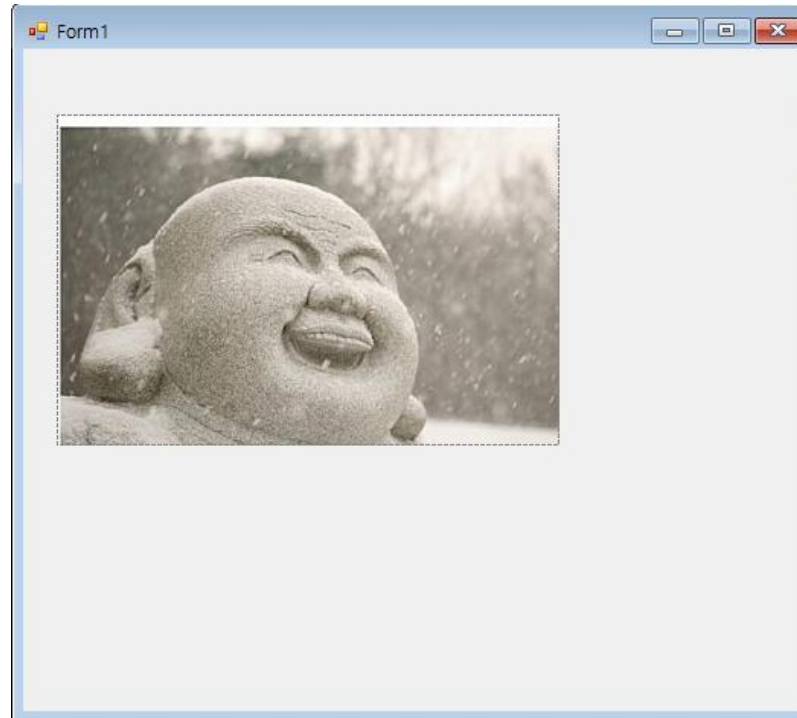
이름	설명
SizeMode	<p>이미지의 배치와 컨트롤의 크기 조정을 처리하는 방법을 선택한다.</p> <ul style="list-style-type: none"> <li>• Normal : 이미지를 컨트롤의 왼쪽 상단에 맞춤, 이미지가 컨트롤의 크기보다 크면 잘림</li> <li>• StretchImage : 이미지가 컨트롤의 크기에 맞게 늘어남</li> <li>• AutoSize : 컨트롤의 크기가 이미지의 크기에 맞게 조절됨</li> <li>• CenterImage : 이미지가 컨트롤보다 크면 컨트롤의 중앙에 정렬되는 동시에 잘림</li> <li>• Zoom : 이미지가 크기 비율에 맞게 확대 또는 축소됨</li> </ul>
BackgroundImage	<p>배경 이미지를 설정한다.</p> <ul style="list-style-type: none"> <li>• 기본값 없음</li> </ul>
BackgroundImageLayout	<p>배경 이미지의 위치를 지정한다.</p> <ul style="list-style-type: none"> <li>• None : 컨트롤의 왼쪽 상단에 맞춤</li> <li>• Tile : 바둑판식으로 배열</li> <li>• Center : 컨트롤의 중앙에 맞춤</li> <li>• Stretch : 컨트롤의 크기에 맞춤</li> <li>• Zoom : 컨트롤의 크기 내에서 확대됨</li> </ul>
BorderStyle	<p>테두리 모양을 설정한다.</p> <ul style="list-style-type: none"> <li>• None : 테두리나 테두리 관련 요소가 없음</li> <li>• FixedSingle : 테두리를 단일 선으로 표현하고 크기 조절 불가능</li> <li>• Fixed3D : 테두리를 입체로 표현</li> </ul>
Image	<p>나타낼 그림을 설정한다. 속성 창에서  버튼을 클릭하여 원하는 그림을 선택한다. 그림을 선택하고 나면 속성 창에 '비트맵'이라고 표시된다. '비트맵'을 지우면 그림이 삭제된다.</p> <ul style="list-style-type: none"> <li>• 기본값 없음</li> </ul>

- SizeMode : Normal

이미지의 크기가 컨트롤의 크기보다 작은 경우



이미지의 크기가 컨트롤의 크기보다 커서 잘린 경우



- SizeMode : StretchImage

이미지가 컨트롤의 크기에 맞게 조정되었음(축소)

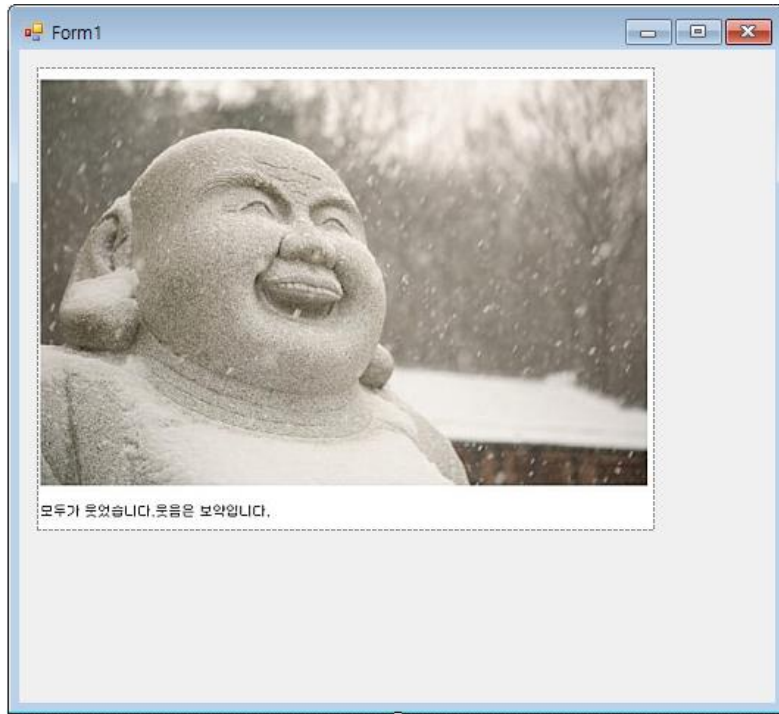


이미지가 컨트롤의 크기에 맞게 늘어남



- SizeMode : AutoSize

픽처박스 컨트롤의 크기가 이미지에 맞게 조정됨.



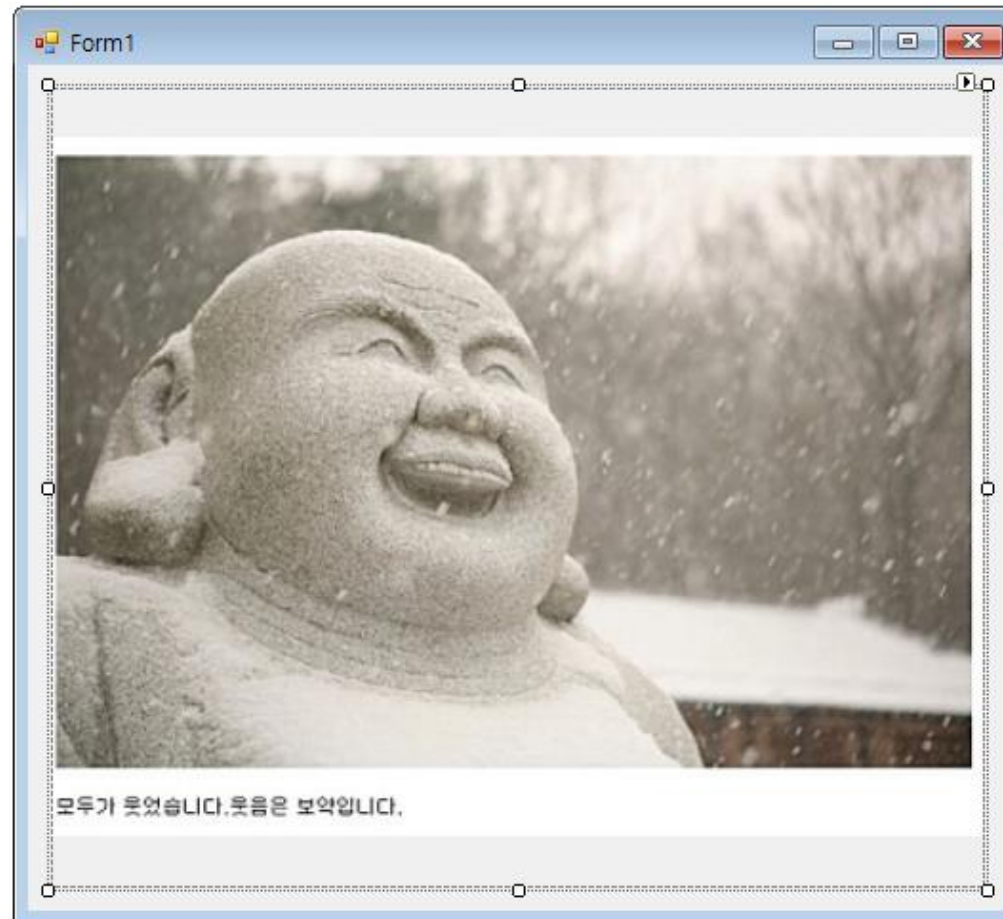
- SizeMode : CenterImage

이미지가 컨트롤의 가운데에 배치됨.



- SizeMode : Zoom

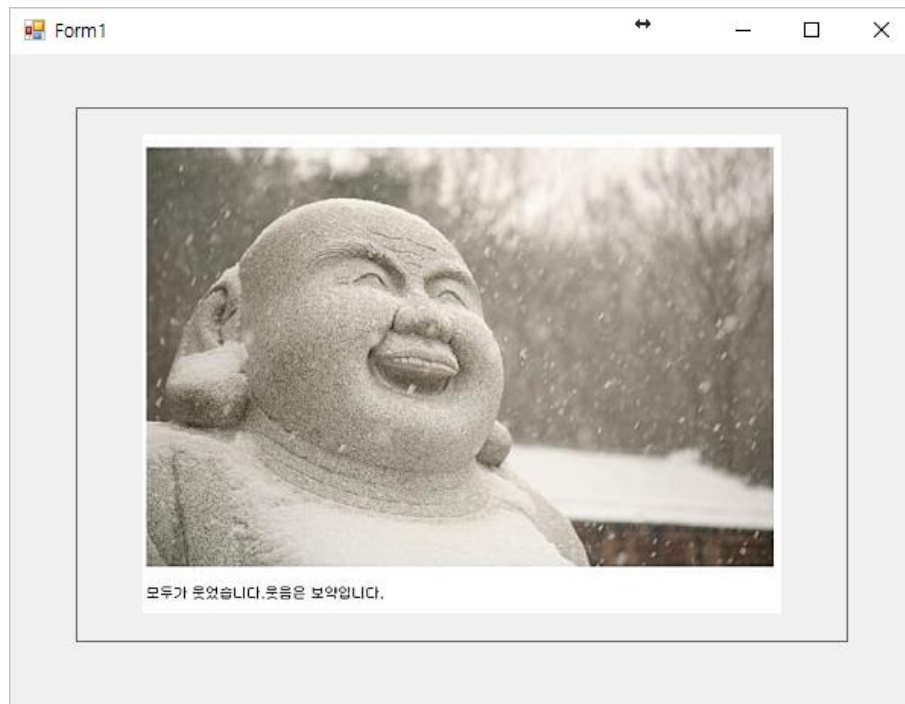
이미지의 비율을 유지하면서 컨트롤의 크기와 맞게 조정됨.



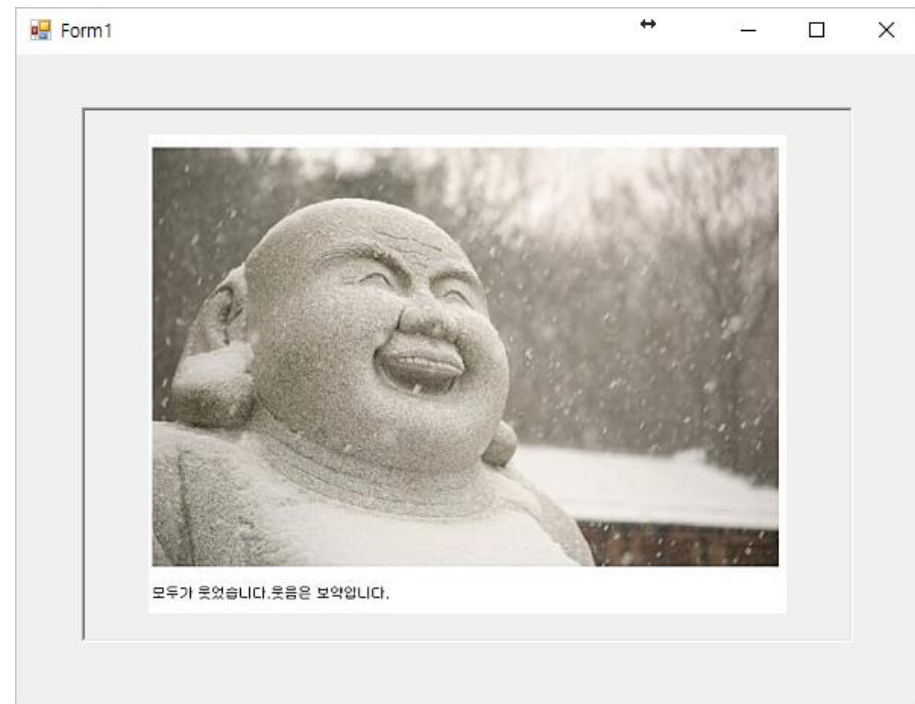


- BorderStyle

FixedSingle



Fixed3D



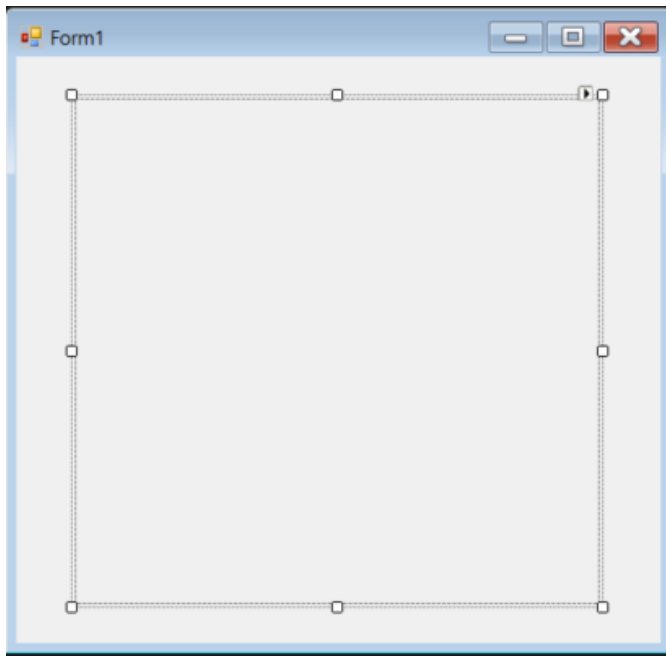


- 이클래스의 <image1.jpg>, <image2.jpg>, <image3.jpg> 파일 D드라이브에 저장


- 실습 - 픽처박스 컨트롤의 Image 속성을 이용하여 이미지 삽입

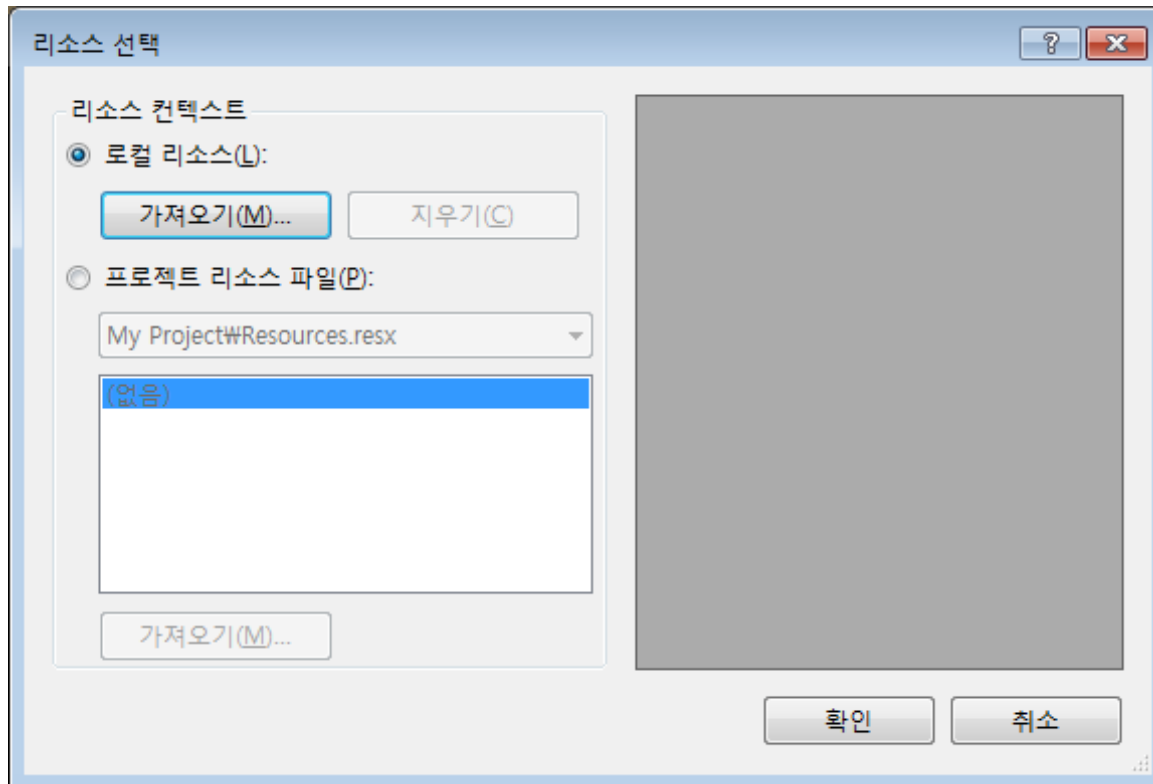
1) 윈도우 폼 디자인

2) 속성 설정



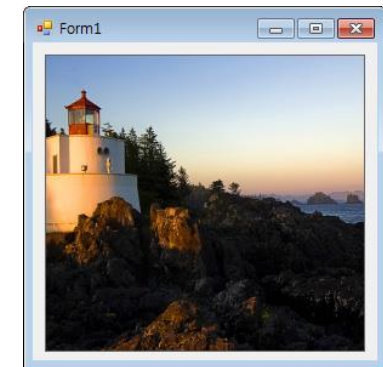
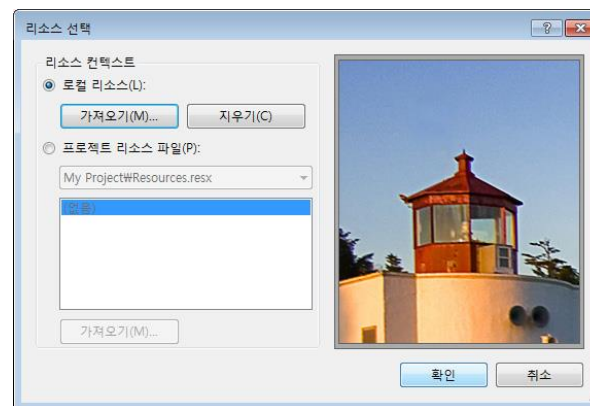
컨트롤	속성	속성값
PictureBox1	BorderStyle	FixedSingle
	SizeMode	StretchImage
	Image	아래 [3]~[6]번과 같이 설정

- 3 ) 픽처박스 컨트롤의 Image 속성 오른쪽에 있는  버튼을 클릭
- 4 ) [리소스 선택] 대화 상자가 나타나면 '로컬 리소스' 를 선택하고 <가져오기>를 클릭



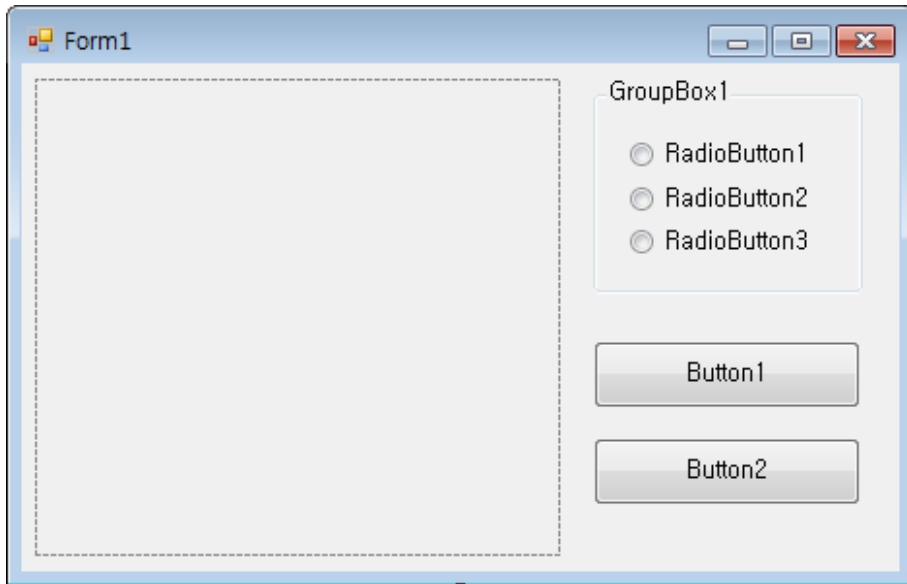
5 ) [열기] 대화 상자에서 원하는 이미지 파일을 선택한다.

6 ) 다시 나타난 [리소스 선택] 대화 상자에서 <확인>을 클릭한다. 픽처박스 컨트롤에 해당 이미지가 삽입되고 Image 속성에 'System.Drawing.Bitmap' 이라고 표시된다.



- 실습 – 픽처박스 컨트롤의 Bitmap() 메서드를 이용하여 이미지 나타내기

## 1) 윈도우 폼 디자인



## 2) 속성 설정

컨트롤	속성	속성값
PictureBox1	SizeMode	StretchImage
GroupBox1	Text	그림 선택
RadioButton1	Text	1번 그림
RadioButton2	Text	2번 그림
RadioButton3	Text	3번 그림
Button1	Text	그림 보기
Button2	Text	그림 삭제

## 3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked)
        pictureBox1.Image = new Bitmap(@"D:\Wimage1.jpg");
    else if (radioButton2.Checked)
        pictureBox1.Image = new Bitmap(@"D:\Wimage2.jpg");
    else
        pictureBox1.Image = new Bitmap(@"D:\Wimage3.jpg");
}

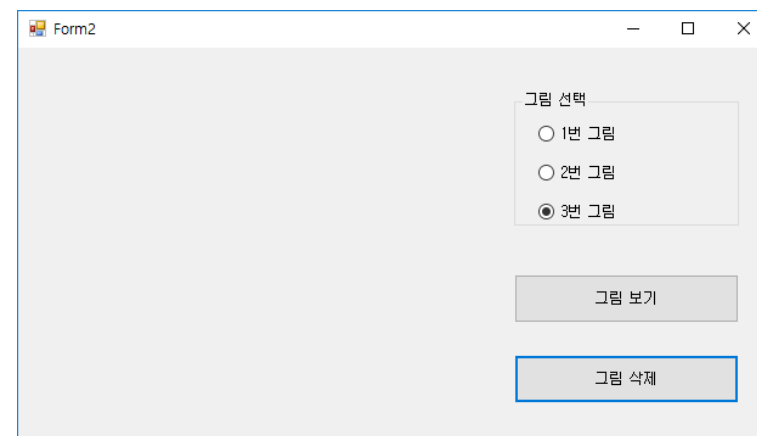
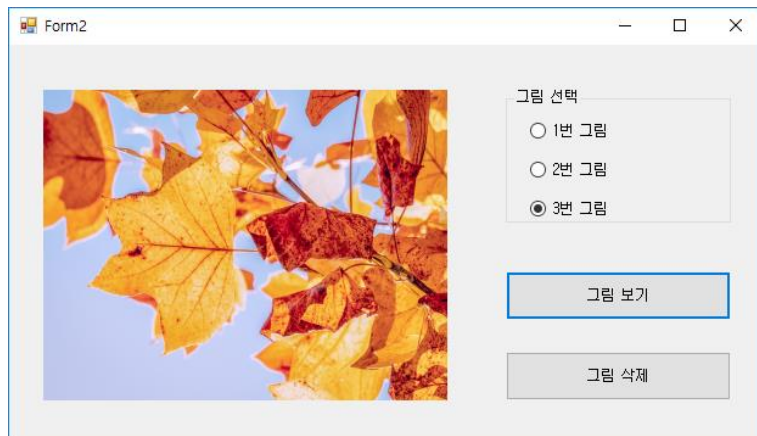
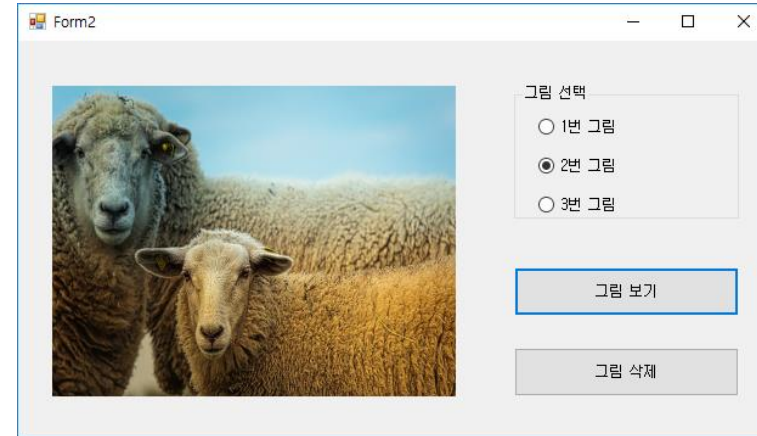
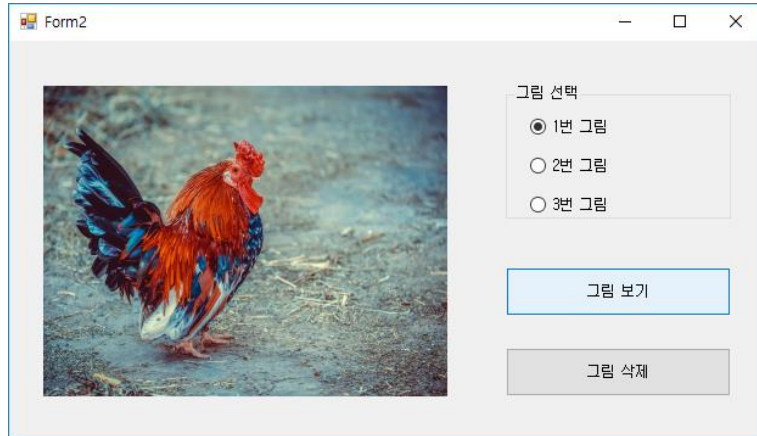
private void button2_Click(object sender, EventArgs e)
{
    pictureBox1.Image = null;
}
```

**New Bitmap("파일 경로 및 파일 명")**

과 같은 형식을 사용하여, 코드 상에서 픽처박스에 이미지를 삽입하는 것이 가능하다.

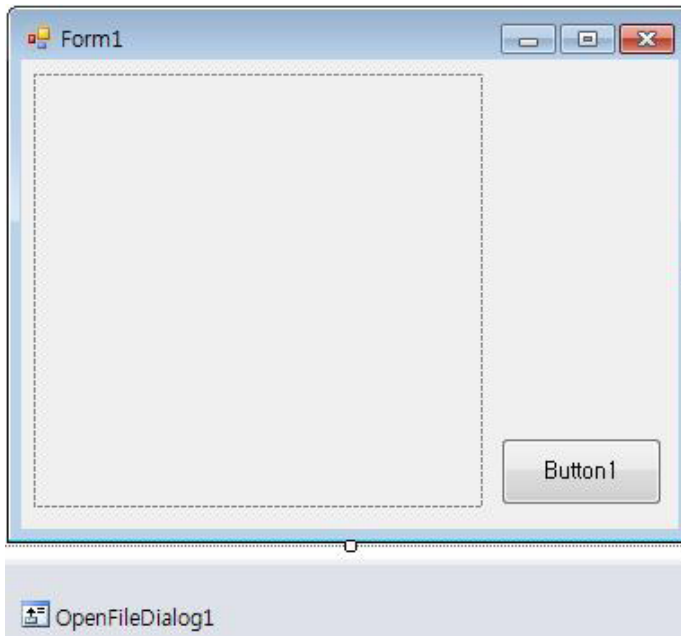


## 4) 결과 확인



- 실습 – [열기] 대화 상자를 이용하여 나타낼 그림 선택

## 1) 윈도우 폼 디자인



## 2) 속성 설정

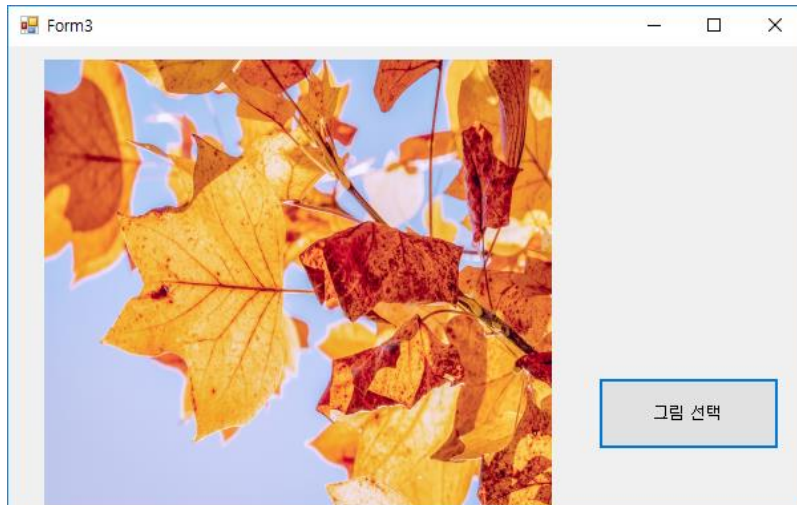
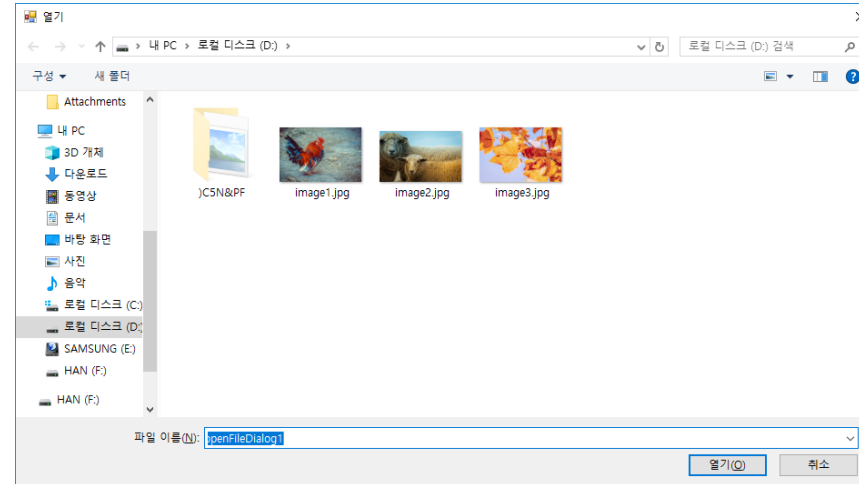
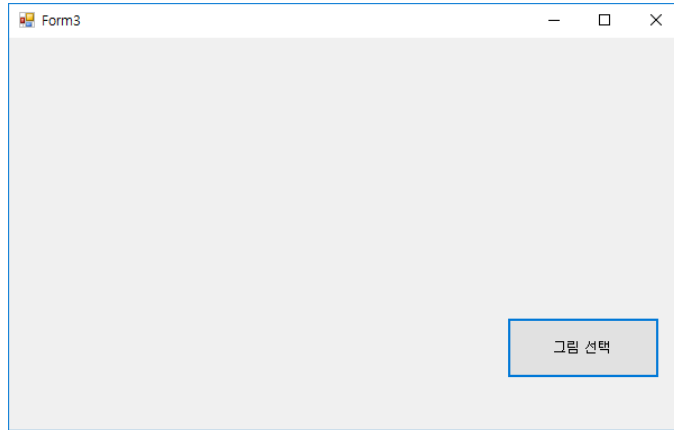
컨트롤	속성	속성값
PictureBox1	SizeMode	StretchImage
Button1	Text	그림 선택

## 3 ) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    var fileContent = openFileDialog1.ShowDialog();

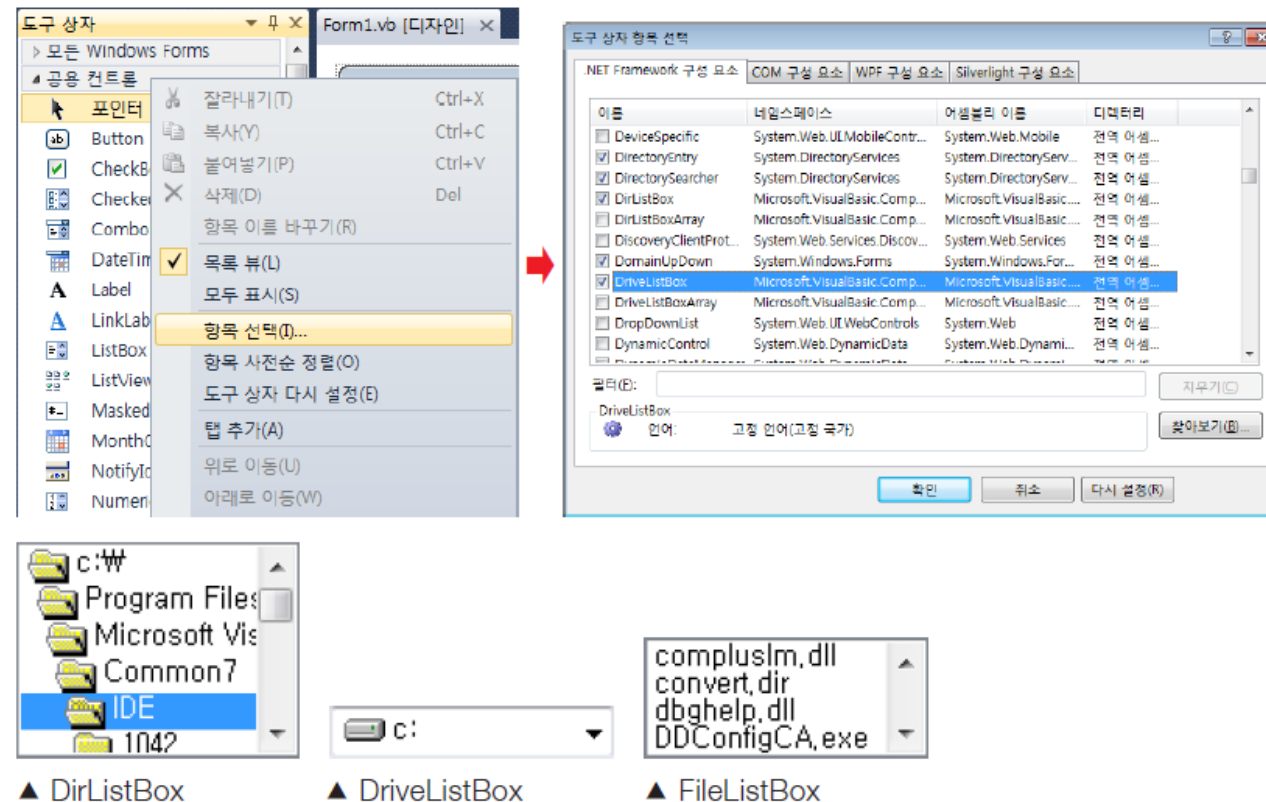
    if(fileContent == DialogResult.OK)
    {
        pictureBox1.ImageLocation = openFileDialog1.FileName;
    }
}
```

## 4) 실행 결과 확인



- C#에서는 드라이브 리스트박스, 디렉터리 리스트박스, 파일 리스트박스 등의 컨트롤을 사용하여 사용자가 파일 시스템을 통해 파일 목록을 살펴볼 수 있도록 해준다.

그림 7-1 파일 관련 컨트롤을 추가하는 방법과 각 컨트롤의 모습



- 드라이브 리스트박스 컨트롤

- 시스템에서 유효한 모든 드라이브의 정보를 콤보박스 형태로 나열해서 보여주며, 사용자가 원하는 드라이브를 선택할 수 있다.

표 7-6 드라이브 리스트박스 컨트롤의 주요 속성

이름	설명
Drive	드라이브를 설정하거나 선택한 드라이브를 반환한다. '형식 : 드라이브 리스트박스 개체명.Drive' 형식으로 사용한다.
List(Index)	인덱스를 사용하여 목록의 항목을 설정하거나 반환한다. 번호는 0부터 시작한다.
ListCount	목록에 포함된 드라이브의 개수를 반환한다.
ListIndex	선택한 드라이브의 인덱스 번호를 반환한다.

- 디렉터리 리스트박스 컨트롤

- 윈도우 탐색기처럼 시스템의 디렉터리 정보를 보여준다.

표 7-7 디렉터리 리스트박스 컨트롤의 주요 속성

이름	설명
List(Index)	인덱스를 사용하여 목록의 항목을 설정하거나 반환한다.
ListCount	목록에 포함된 디렉터리의 개수를 반환한다.
ListIndex	선택한 디렉터리의 인덱스 번호를 반환한다.
Path	선택한 디렉터리의 경로명을 반환한다.



- 파일 리스트박스 컨트롤
  - 파일의 정보를 보여준다.

표 7-8 파일 리스트박스 컨트롤의 주요 속성

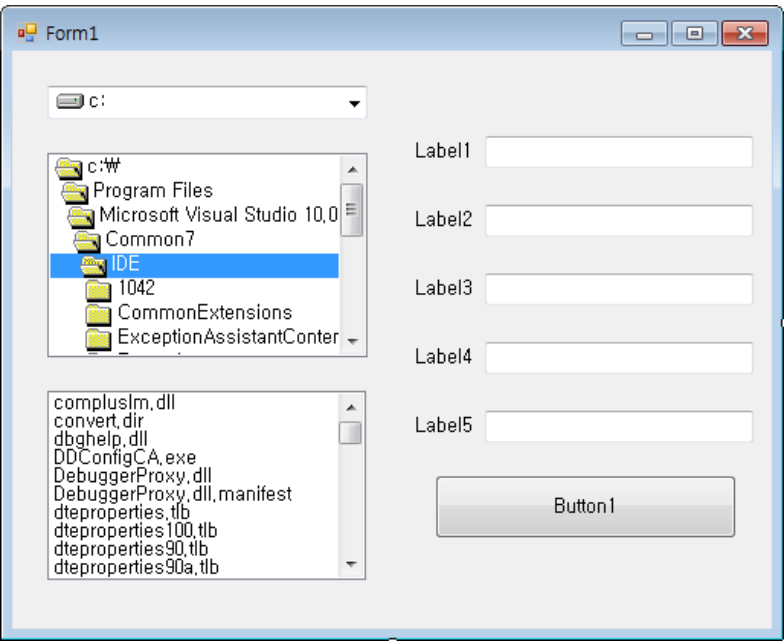
파란색은 기본값

이름	설명
Archive	목록에 일반 파일의 표시 여부를 설정한다. <ul style="list-style-type: none"><li>• True : 일반 파일 표시</li><li>• False : 일반 파일을 표시하지 않음</li></ul>
FileName	선택한 파일의 이름을 반환한다. 파일이 저장된 경로는 포함하지 않으며, 코드에서 설정한다.
Hidden	목록에 숨기기 속성을 가진 파일의 표시 여부를 설정한다. <ul style="list-style-type: none"><li>• True : 숨김 파일 표시</li><li>• False : 숨김 파일을 표시하지 않음</li></ul>
Normal	목록에 표준 속성을 가진 파일의 표시 여부를 설정한다. <ul style="list-style-type: none"><li>• True : 표준 속성 파일 표시</li><li>• False : 표준 속성 파일을 표시하지 않음</li></ul> <div>표준속성 파일 : system or hidden 속성이 아닌 모든 파일</div>
List(Index)	인덱스를 사용하여 해당 파일을 설정하거나 반환한다. 코드에서 설정한다.
ListCount	파일 리스트박스의 목록에 포함된 파일의 개수를 반환한다. 코드에서 설정한다.
ListIndex	선택한 파일의 인덱스 번호를 반환한다. 코드에서 설정한다.

SelectionMode	<p>여러 파일을 선택할지를 설정한다.</p> <ul style="list-style-type: none"><li>• None : 선택 불가</li><li>• One : <b>하나만 선택 가능</b></li><li>• MultiSimple : 마우스 클릭으로 여러 파일 선택</li><li>• MultiExtended : <b>Shift</b>나 <b>Ctrl</b>을 사용하여 여러 파일 선택</li></ul>
Path	<p>선택한 파일의 경로명을 반환한다. 코드에서 설정한다.</p>
Pattern	<p>특정 조건이나 여러 조건에 맞는 파일을 표시할 때 사용한다.</p> <ul style="list-style-type: none"><li>• <b>기본값 : *</b></li><li>• 예 : File1.Pattern = "*" → 모든 파일 표시</li><li>File1.Pattern = "*.hwp", "*.txt" → 확장자가 hwp, txt인 파일 표시</li></ul>
ReadOnly	<p>목록에 읽기 전용 속성을 가진 파일의 표시 여부를 설정한다.</p> <ul style="list-style-type: none"><li>• <b>True : 읽기 전용 파일 표시</b></li><li>• False : 읽기 전용 파일을 표시하지 않음</li></ul>
System	<p>목록에 시스템 관련 파일의 표시 여부를 설정한다.</p> <ul style="list-style-type: none"><li>• True : 시스템 관련 파일 표시</li><li>• <b>False : 시스템 관련 파일을 표시하지 않음</b></li></ul>

- 실습 – 드라이브/디렉터리/파일 리스트박스 컨트롤을 이용하여 파일 정보 보여주기

1 ) 윈도우 폼 디자인  
(컨트롤 추가에서 DirListBox,  
DriveListBox, FileListBox 추가)



2 ) 속성 설정

컨트롤	속성	속성값
FileListBox1	Hidden	True
	System	True
Label1	Text	드라이브
Label2	Text	디렉터리
Label3	Text	파일 경로
Label4	Text	파일명
Label5	Text	파일 확장자
TextBox1~TextBox5	Text	(빈칸)
Button1	Text	파일 목록 갱신

## 3 ) 코드 작성

```
private void Form4_Load(object sender, EventArgs e)
{
    changePathInfo();
}

private void driveListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    dirListBox1.Path = driveListBox1.Drive;
    changePathInfo();
}

private void dirListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    fileListBox1.Path = dirListBox1.Path;
    changePathInfo();
}
```

```
private void fileListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    changePathInfo();
}

private void button1_Click(object sender, EventArgs e)
{
    fileListBox1.Pattern = textBox5.Text;
    fileListBox1.Refresh();
    changePathInfo();
}

public void changePathInfo()
{
    textBox1.Text = driveListBox1.Drive;
    textBox2.Text = dirListBox1.Path;
    textBox3.Text = fileListBox1.Path;
    textBox4.Text = fileListBox1.FileName;
    textBox5.Text = fileListBox1.Pattern;
}
```

## 4) 실행 결과 확인

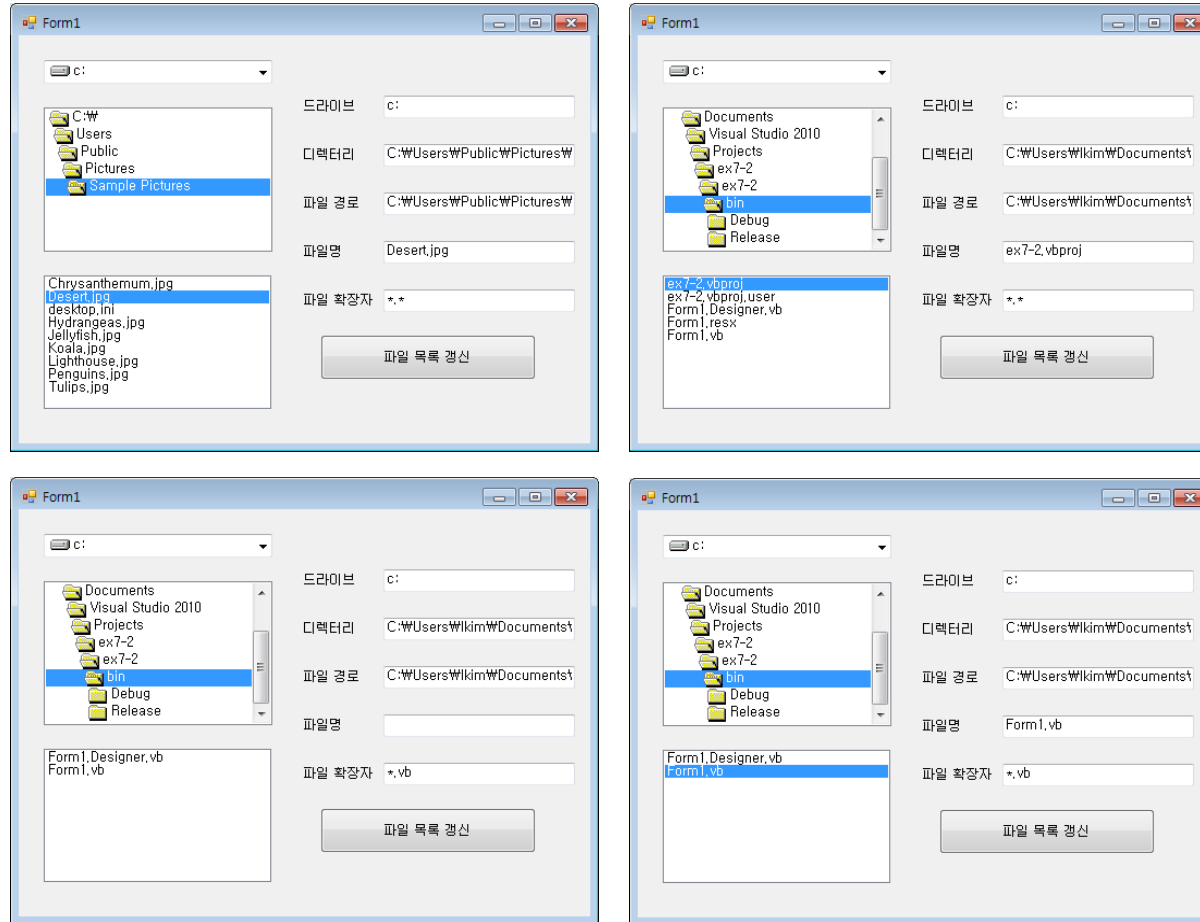
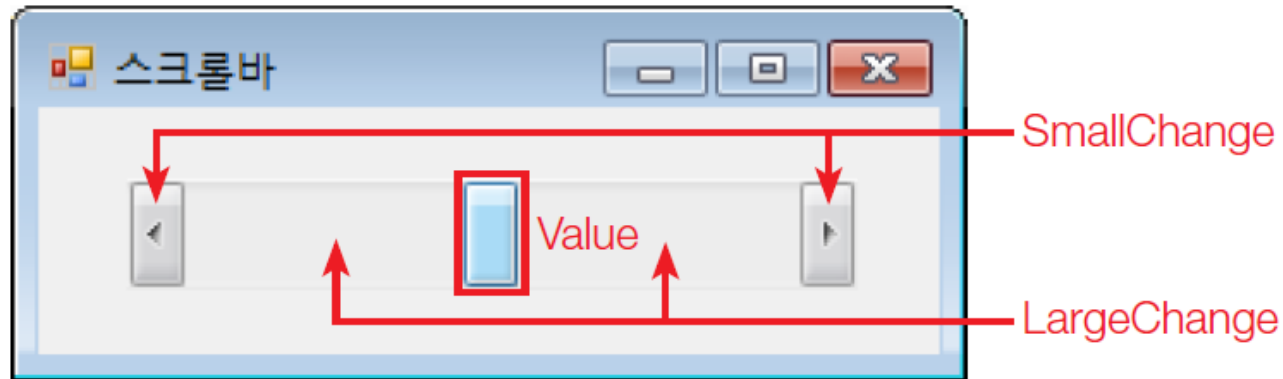


표 7-9 스크롤바 컨트롤의 주요 속성

이름	설명
LargeChange	스크롤바 안쪽을 클릭할 때마다 변경되는 스크롤바의 Value 속성을 설정하거나 반환한다.
Minimum/Maximum	속성값이 가질 수 있는 최솟값과 최댓값으로, 각각 스크롤바 맨 위와 맨 아래의 값이다.
SmallChange	스크롤바 양 끝의 화살표를 눌렀을 때 변경되는 스크롤바의 Value 속성을 설정하거나 반환한다.
Value	스크롤바의 현재 위치 값을 설정하거나 반환한다.

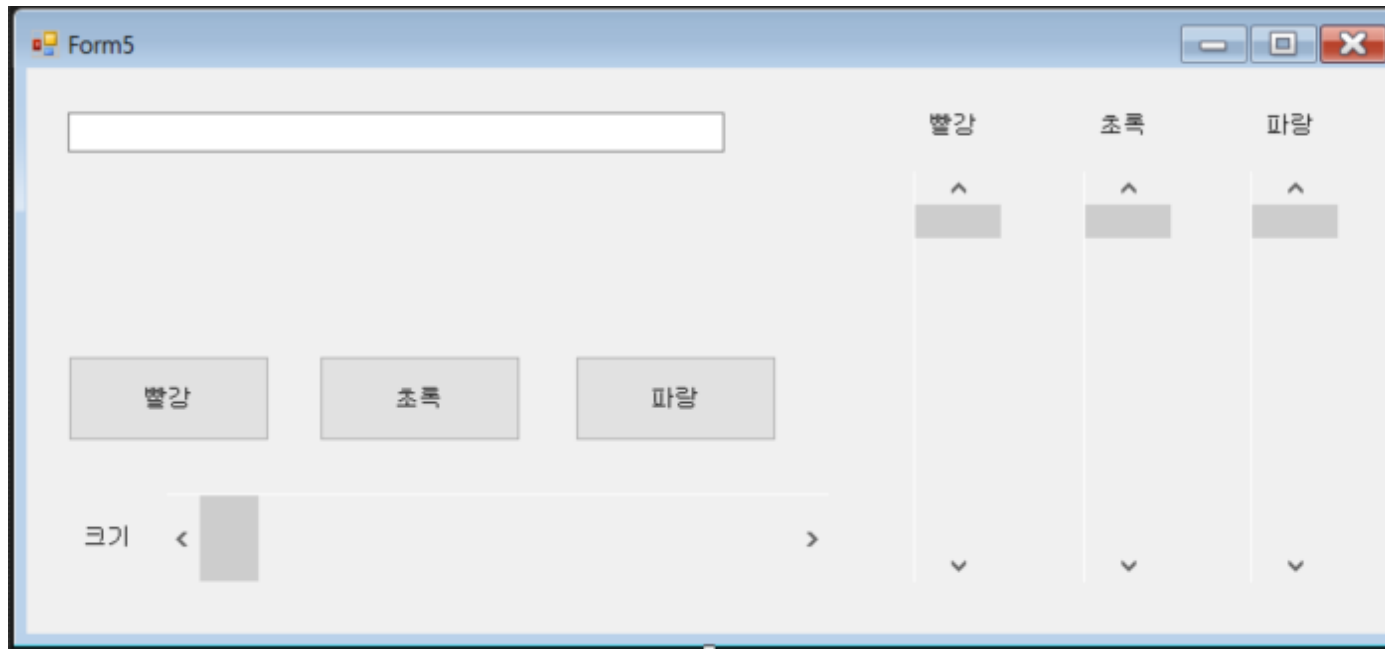
그림 7-2 스크롤바의 주요 속성





- 실습 - 글자의 크기와 색을 변경하는 스크롤바 만들기

## 1) 윈도우 폼 디자인



## 2) 속성 설정

컨트롤	속성	속성값
textBox1	Multiline	True
	Text	(빈칸)
button1	Text	빨강
button2	Text	초록
button3	Text	파랑
label1	Text	크기
label2	Text	빨강
label3	Text	초록
label4	Text	파랑

컨트롤	속성	속성값
HScrollBar1	LargeChange	10
	Maximum	72
	Value	5
	Minimum	
VScrollBar1	LargeChange	10
	Maximum	255
	Value	0
VScrollBar2	LargeChange	10
	Maximum	255
	Value	0
VScrollBar3	LargeChange	10
	Maximum	255
	Value	0

## 3 ) 코드 작성

```
private void Form5_Load(object sender, EventArgs e)
{
    setFont();
}

private void button1_Click(object sender, EventArgs e)
{
    vScrollBar1.Value = 255;
    vScrollBar2.Value = 0;
    vScrollBar3.Value = 0;

    setFont();
}

private void button2_Click(object sender, EventArgs e)
{
    vScrollBar1.Value = 0;
    vScrollBar2.Value = 255;
    vScrollBar3.Value = 0;

    setFont();
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    vScrollBar1.Value = 0;
    vScrollBar2.Value = 0;
    vScrollBar3.Value = 255;

    setFont();
}

private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)
{
    setFont();
}

private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
{
    setFont();
}

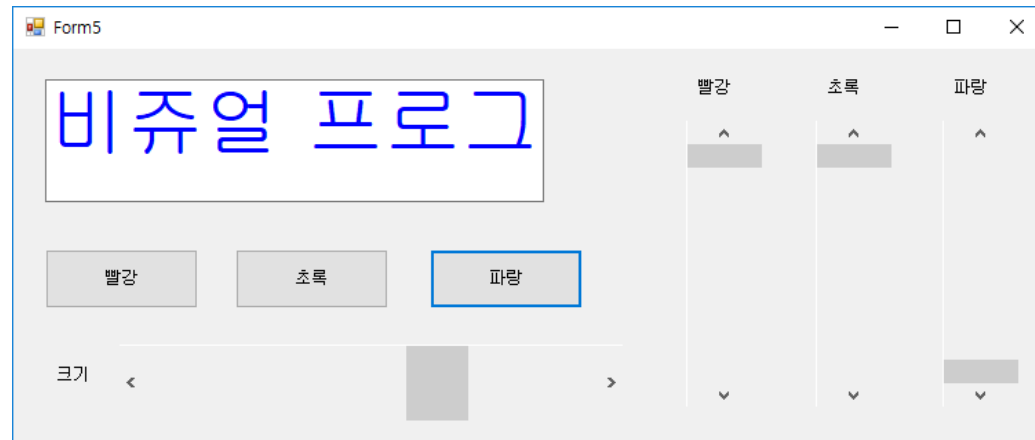
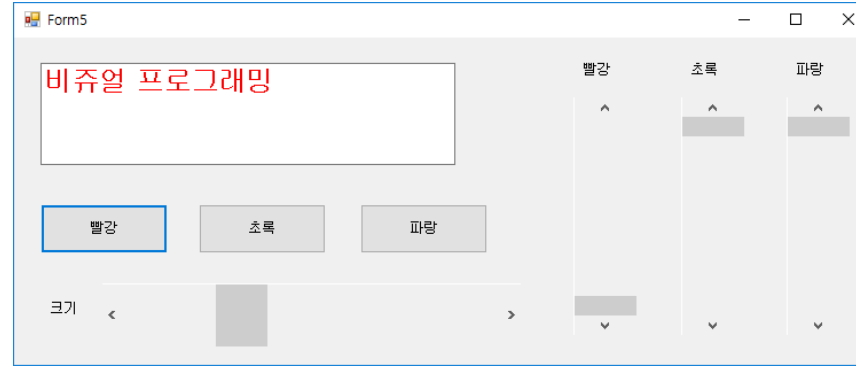
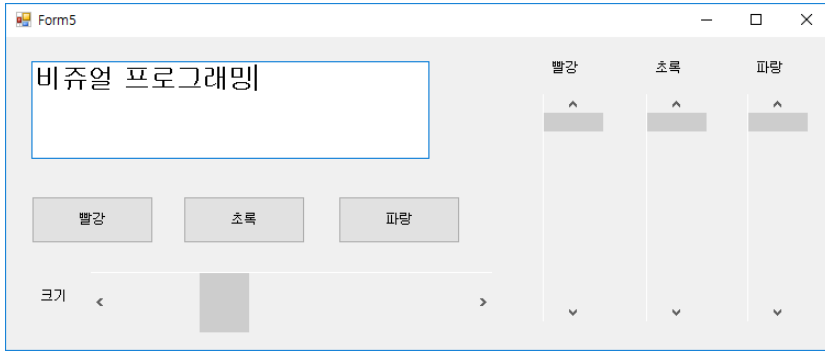
private void vScrollBar2_Scroll(object sender, ScrollEventArgs e)
{
    setFont();
}

private void vScrollBar3_Scroll(object sender, ScrollEventArgs e)
{
    setFont();
}
```

## 3 ) 코드 작성

```
public void setFont()  
{  
    int r = vScrollBar1.Value;  
    int g = vScrollBar2.Value;  
    int b = vScrollBar3.Value;  
    int size = hScrollBar1.Value;  
    if (size <= 0)  
        size = 1;  
    textBox1.Font = new Font("굴림", size);  
    textBox1.ForeColor = Color.FromArgb(r, g, b);  
}
```

## 4) 실행 결과 확인



- 타이머 컨트롤
  - 지정된 시간 단위로 타이머 이벤트 프로시저를 실행
  - 시간 표시나 애니메이션과 같이 일정 간격으로 설정된 내용을 반복 실행하는 데 주로 사용

표 7-10 타이머 컨트롤의 주요 속성

파란색은 기본값

이름	설명
Interval	타이머 이벤트의 실행 간격을 밀리초(1/1,000초)로 설정하거나 반환한다. 1초 간격으로 실행하려면 1000을 입력한다. 1~65535(약 1분)를 지정할 수 있으며 숫자가 클수록 긴 간격으로 실행된다. • 기본값 : 100
Enabled	타이머 이벤트를 실행하거나 중지한다. • True : Interval 간격으로 타이머 이벤트 발생 • False : 타이머 이벤트 중지



- 뉴메릭 (Numeric) 업다운 컨트롤
  - 위아래 버튼을 눌러 숫자를 증감

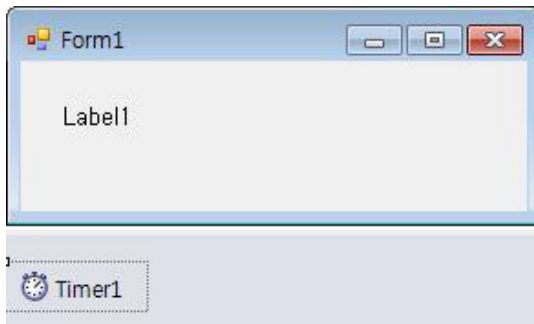
표 7-11 뉴메릭 업다운 컨트롤의 주요 속성

파란색은 기본값

이름	설명
Maximum	증가할 수 있는 숫자의 최댓값을 지정한다. • 기본값 : 100
Minimum	감소할 수 있는 숫자의 최솟값을 지정한다. • 기본값 : 0
Value	증감하는 숫자의 현재 값을 지정한다. • 기본값 : 0
Increment	증감 버튼을 누를 때마다 변화하는 값의 크기를 지정한다. • 기본값 : 1

- 실습 - 타이머 컨트롤을 이용하여 시간 표시하기

1) 윈도우 폼 디자인      2) 속성 설정



컨트롤	속성	속성값
Timer1	Interval	1000

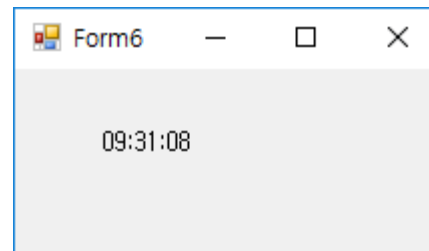
3) 코드 작성

4) 실행 결과 확인

```
private void Form6_Load(object sender, EventArgs e)
{
    timer1.Enabled = true;
    label1.Text = System.DateTime.Now.ToString("hh:mm:ss");
}

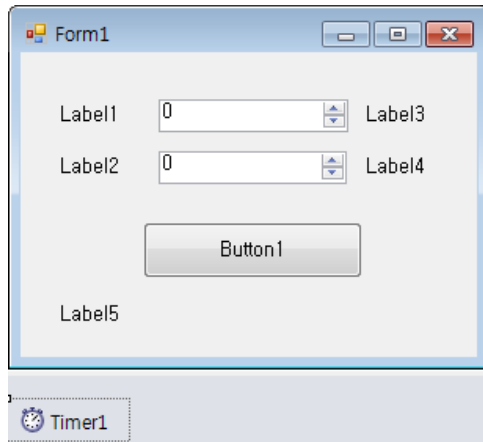
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = System.DateTime.Now.ToString("hh:mm:ss");
}
```

- 타이머 컨트롤의 Interval 속성을 1000으로 설정했기 때문에 1초마다 시간이 새롭게 표시



- 실습 – 타이머 컨트롤과 뉴메릭 업다운 컨트롤을 이용하여 운동 스톱워치 만들기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값	컨트롤	속성	속성값
Label1	Text	반복 간격	NumericUpDown1	Maximum	60
Label2	Text	반복 횟수		Minimum	1
Label3	Text	초		Value	1
Label4	Text	회	NumericUpDown2	Value	1
Button1	Text	시작			

## 3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    label5.Text = "시작!" + numericUpDown2.Value.ToString() + "회 남았습니다.";
    timer1.Interval = (int)numericUpDown1.Value * 1000;
    timer1.Start();
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (numericUpDown2.Value > 0)
    {
        numericUpDown2.Value--;
        label5.Text = numericUpDown2.Value.ToString() + "회 남았습니다.";
    }
    else
    {
        timer1.Stop();
    }
}
```

## 4) 실행 결과 확인