

품과 메뉴

HyoJoon Han
동국대학교
han6343@dongguk.edu



폼 다루기

메뉴 만들기

- 폼
 - 프로그램 사용자에게 보이는 화면(인터페이스)을 만드는 데 사용하는 창(윈도우)
 - 메뉴, 스크롤 바, 단축 아이콘 등 윈도우 응용 프로그램에서 볼 수 있는 요소를 모두 포함
 - 폼도 한 개체로 간주되어 고유한 속성과 이벤트, 메서드 등을 가진다

- 폼의 속성
 - 디자인 모드에서는 속성 창에서, 실행 모드에서는 코드로 속성을 설정할 수 있다.
 - 각 속성은 기본 값이 있기 때문에 변경이 필요한 속성의 값만 수정하면 된다.

표 9-1 폼의 주요 속성

파란색은 기본값

속성	설명
(Name)	이름을 설정한다. 여기서 지정한 이름을 프로그램 코드에서 사용한다. • 기본값 : Form1
BackgroundImage	배경에 이미지를 지정한다. • 기본값 없음
BackColor	배경 색을 지정한다. • 기본값 : Control
FormBorderStyle	경계 모양을 설정한다. • None : 테두리나 테두리 관련 요소가 없음 • FixedSingle : 단일선 테두리 • Fixed3D : 입체 테두리 • FixedDialog : 대화 상자 스타일의 굵은 테두리 • Sizable : 크기를 조정 가능한 테두리 • FixedToolWindow : 크기를 조정할 수 없는 테두리로 작업 표시 줄이나 창에는 나타나지 않음 • SizableToolWindow : 크기를 조정할 수 있는 테두리로 작업 표시 줄이나 창에는 나타나지 않음

속성	설명
Text	제목 표시 줄에 들어갈 문자열을 설정한다. • 기본값 : Form1
ControlBox	제목 표시 줄에 아이콘과 최소화/최대화/닫기 버튼의 표시 여부를 설정한다. • True : 표시 가능 • False : 표시 불가
Font	폼에 표시하는 글자의 모양을 설정한다. • 기본값 : 굴림, 9pt
ForeColor	폼에 출력할 글자나 도형의 색을 설정한다. • 기본값 : ControlText
Size	폼의 크기를 조절한다. • 기본값 : Width:300 Height:300
Icon	폼의 아이콘을 설정한다.
MaximizeBox, MinimizeBox	폼의 최대화, 최소화 버튼 사용 유무를 설정한다. 단, BorderStyle이 FixedSingle, Fixed3D, FixedDialog, Sizable이고 ControlBox 속성값도 True여야 적용이 가능하다. • True : 사용 가능 • False : 사용 불가
StartPosition	화면에 표시할 폼의 위치를 설정한다. • Manual : 폼 배치 창에서 출력 위치를 변경하면 속성값이 그 값으로 변경되는, 기본 설정 값이 없는 상태 • CenterScreen : 전체 화면의 중앙에 출력 • WindowsDefaultLocation : 윈도우 기본 위치에 출력하고 크기는 해당 폼의 크기 내에서 지정 • WindowsDefaultBounds : 윈도우 기본 위치에 출력하고 범위는 윈도우 기본값에 의해 결정 • CenterParent : 부모 폼의 중앙에 출력

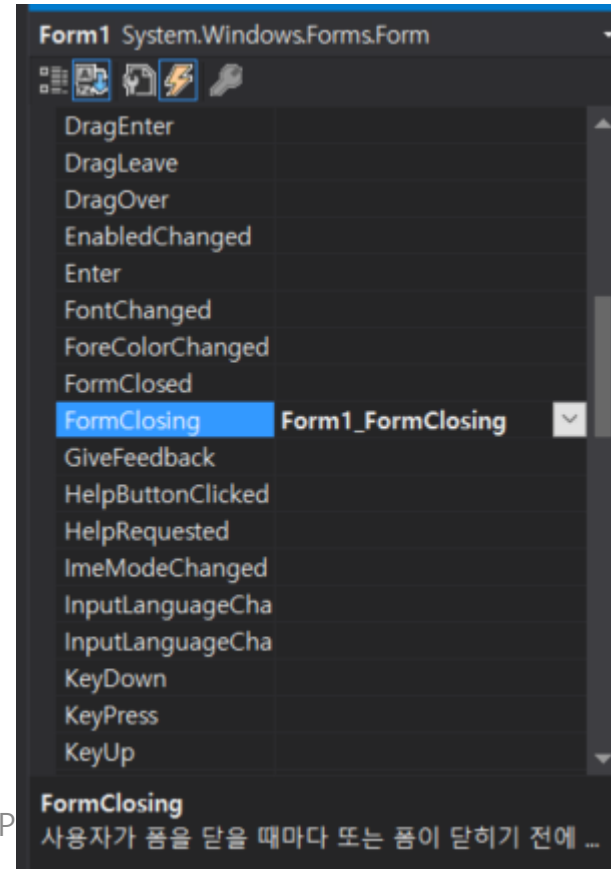
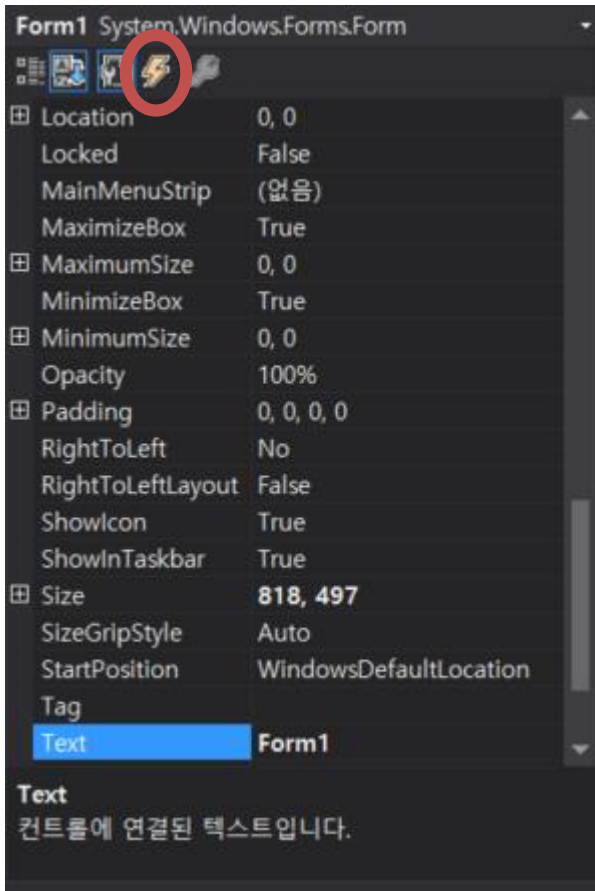
TopMost	폼을 화면 최상단에 나타낼지 여부를 설정한다. • True : 폼을 화면 최상단에 표시 • False : 폼을 화면 최상단에 표시하지 않음
WindowState	폼이 실행되었을 때의 상태(최대화/최소화/닫기)를 설정한다. • Normal : 디자인 모드에서 사용한 크기대로 실행 모드에서 표시 • Minimized : 아이콘으로 표시 • Maximized : 전체 화면으로 표시

표 9-2 폼의 주요 이벤트

click = MouseDown + MouseUp

이벤트	기능	
기본	Activated	폼이 활성화될 때 발생한다.
	Deactivate	폼이 비활성화될 때 발생한다.
	FormClosing	제목 표시 줄의 닫기 버튼을 클릭했을 때, close() 메서드를 호출했을 때 발생한다.
	Load	폼이 메모리에 로딩될 때 발생한다.
	Resize	사용자의 상호 작용이나 코드를 사용하여 폼의 크기를 변경할 때 발생한다.
마우스 관련	Click	폼에서 마우스 왼쪽 버튼을 눌렀을 때 발생한다.
	MouseMove	폼에서 마우스를 움직일 때 발생한다.
	MouseDown	폼에서 마우스 왼쪽 버튼을 눌렀을 때 발생한다.
	MouseUp	폼에서 마우스 왼쪽 버튼을 눌렀다 놓았을 때 발생한다.
키보드 관련	KeyPress	표준 아스키 코드 값이 부여된 키(숫자, 문자, 특수기호 등)를 눌렀다 놓을 때 발생한다.
	KeyDown	모든 키를 눌렀을 때 발생한다.
	KeyUp	모든 키를 눌렀다 놓았을 때 발생한다.

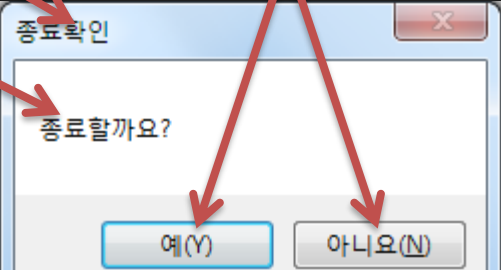
- 실습 – FormClosing 이벤트를 이용하여 프로그램 종료 확인 창 띄우기
 - 1) 새 프로젝트를 열고 폼을 클릭한 상태에서 속성창 옆의 이벤트를 클릭한다.
 - 폼 이벤트 리스트에서 FormClosing을 더블 클릭한다.
 - FormClosing 메소드에서는 폼에 종료 명령이 들어왔을 때 수행해야할 동작을 지정할 수 있다.



2) 코드 작성

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    var digMessage = MessageBox.Show("종료할까요?", "종료 확인", MessageBoxButtons.YesNo);

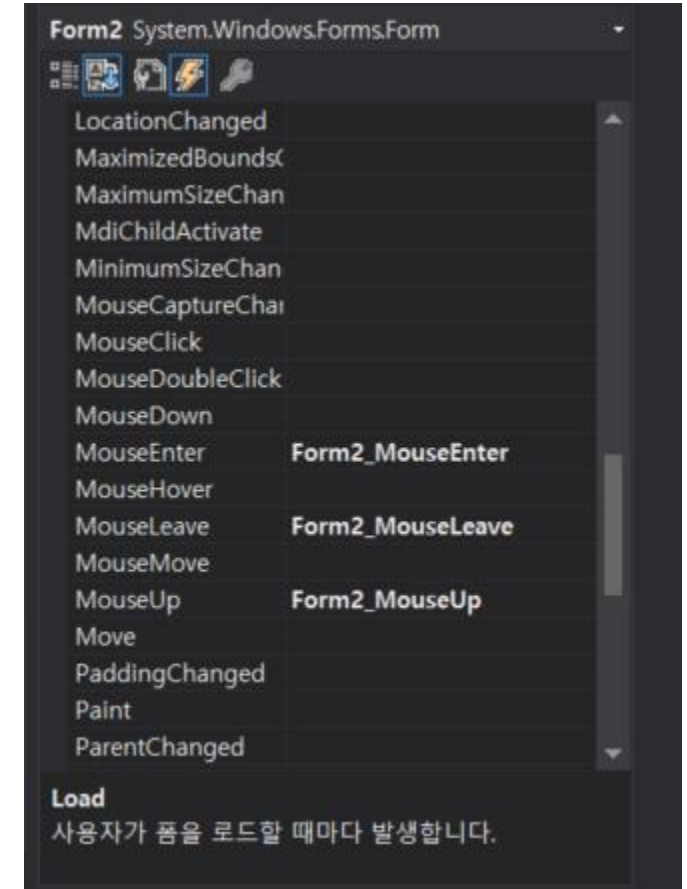
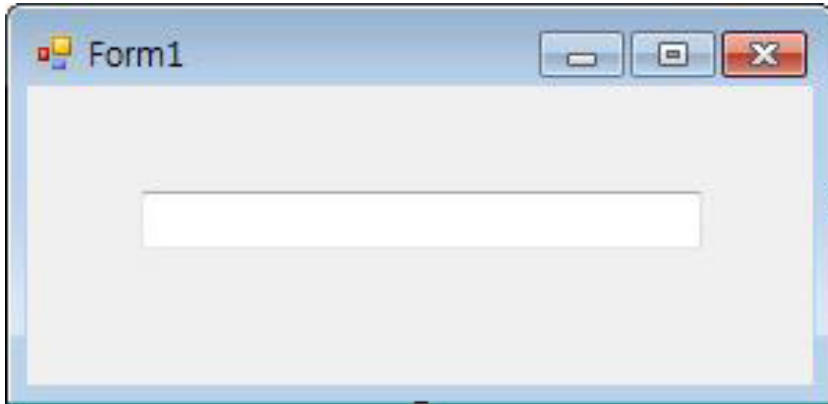
    if(digMessage == DialogResult.No)
    {
        e.Cancel = true;
    }
}
```



- Form1_FormClosing() 메서드를 포함한 다른 대부분의 메서드들은 자신의 처리 결과를 시스템에 전달 하기 위해 'System.Windows.Forms.FormClosingEventArgs e' 와 같은 형태의 인자를 갖는다.
- 이 경우 e.cancel = true의 의미는 시스템에 '종료 명령을 취소' 하겠다는 의미에 해당

- 실습 – MouseEnter, MouseLeave, MouseUp 이벤트 사용하기

1) 윈도우 폼 디자인



2) 이벤트 리스트에서 MouseEnter, MouseLeave, MouseUp 추가

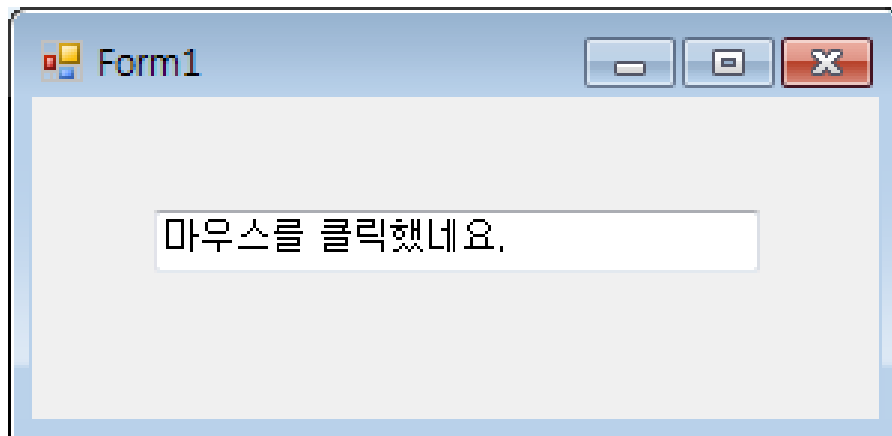
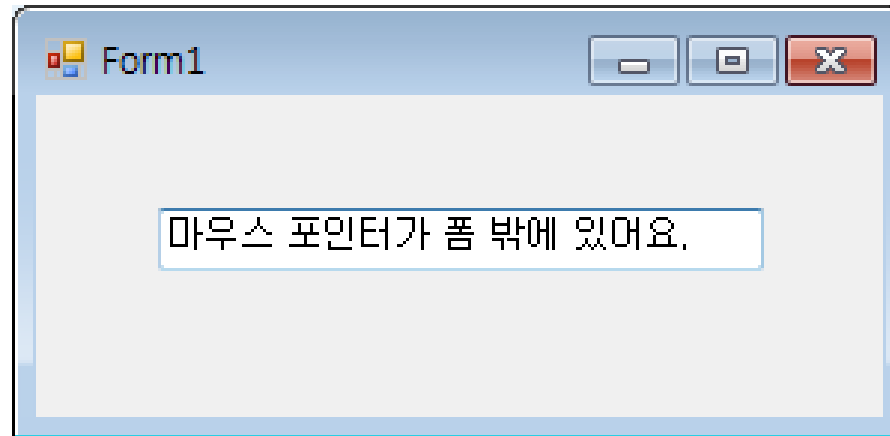
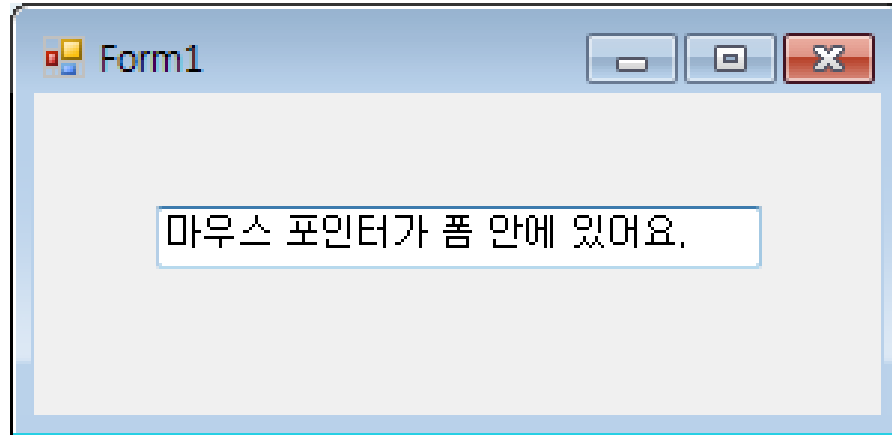
3) 코드 작성

```
private void Form2_MouseEnter(object sender, EventArgs e)
{
    textBox1.Text = "마우스 포인터가 폼 안에 있어요.";
}

private void Form2_MouseLeave(object sender, EventArgs e)
{
    textBox1.Text = "마우스 포인터가 폼 밖에 있어요.";
}

private void Form2_MouseUp(object sender, MouseEventArgs e)
{
    textBox1.Text = "마우스를 클릭했네요.";
}
```


4) 실행 결과 확인



- 폼의 메서드

표 9-3 폼의 주요 메서드

메서드	기능	사용 형식
Show	폼을 화면에 표시한다.	폼 이름.Show()
ShowDialog	폼을 Modal 대화 상자로 화면에 표시한다.	폼 이름.ShowDialog()
Close	폼을 닫는다.	폼 이름.Close()
Hide	폼을 화면에서 감춘다.	폼 이름.Hide()

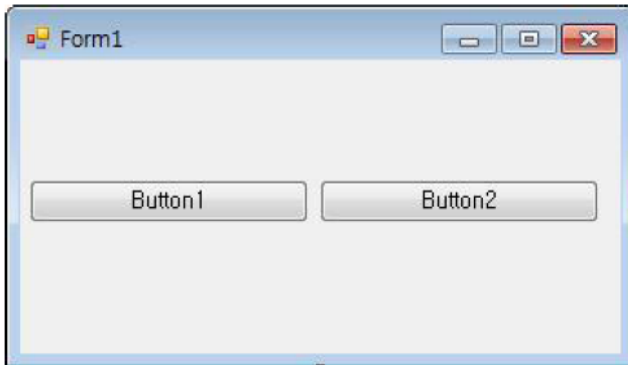
표 9-4 Show와 ShowDialog 메서드의 차이

메서드	설명
Show	모달리스 폼은 닫지 않아도 다른 폼으로 포커스를 옮길 수 있으므로 다른 작업을 수행할 수 있다.
ShowDialog	모달 형태로 표시된 폼은 그 폼이 제거되지 않는 한 다른 작업을 할 수 없다. 즉 모달 폼이 닫히기 전에는 다른 폼으로 포커스를 옮길 수 없다.

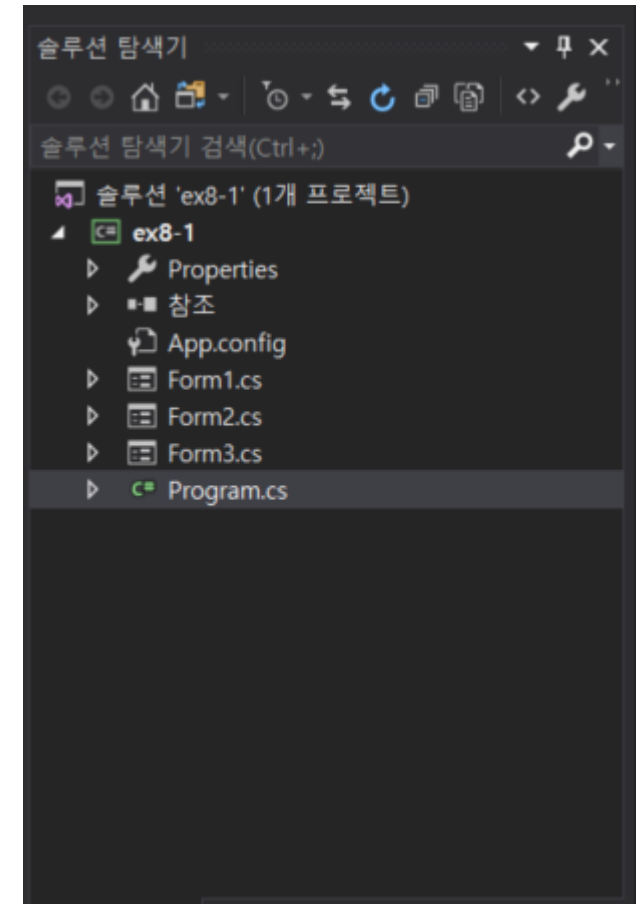
- 실습 – 폼의 Show, ShowDialog, Hide, Close 메서드 사용하기

1) 새 프로젝트를 열고 폼을 2개 추가한다.

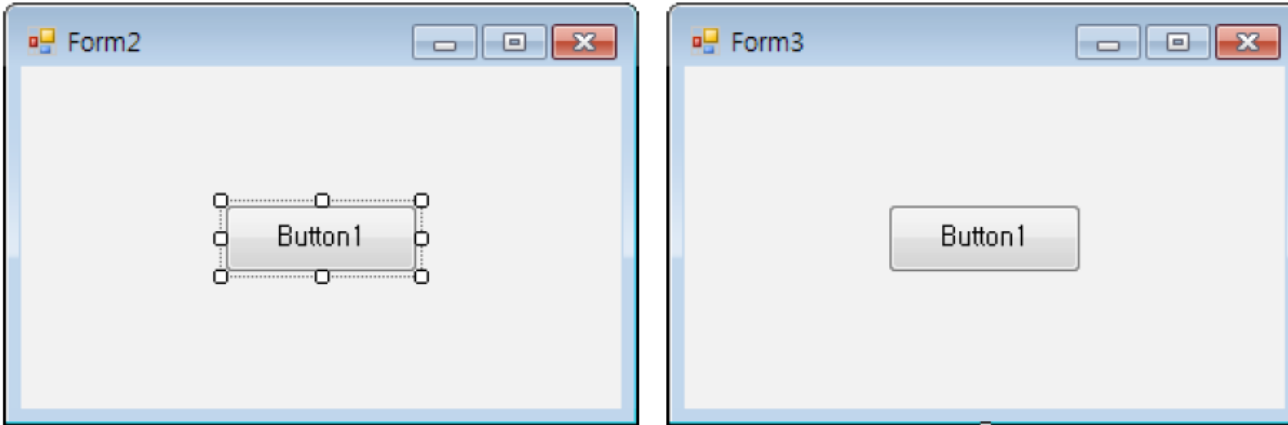
2) Form1을 다음과 같이 디자인하고 컨트롤의 속성을 설정한다.



컨트롤	속성	속성값
Form1	Text	첫 번째 창
Button1	Text	두 번째 창 보기
Button2	Text	세 번째 창(모델폼) 보기



3) Form2와 Form3도 같은 방법으로 디자인하고 컨트롤의 속성을 설정한다.



컨트롤	속성	속성값
Form2	Text	두 번째 창
Button1	Text	감추기
Form3	Text	세 번째 창
Button1	Text	닫기

4) 코드 작성

- 첫 번째 폼

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    form2.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    Form3 form3 = new Form3();
    form3.ShowDialog();
}
```

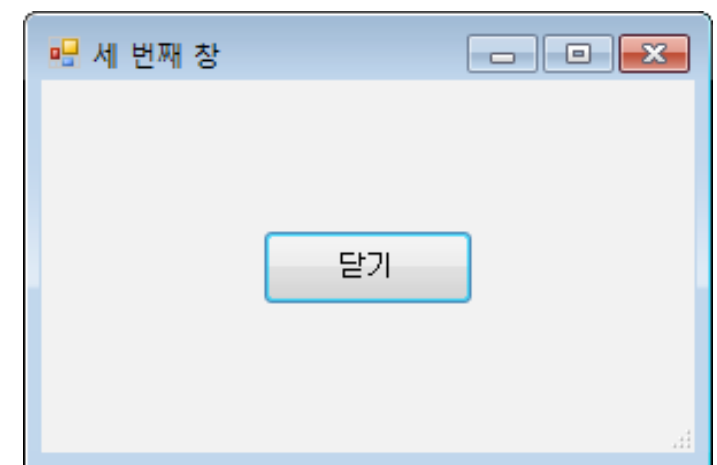
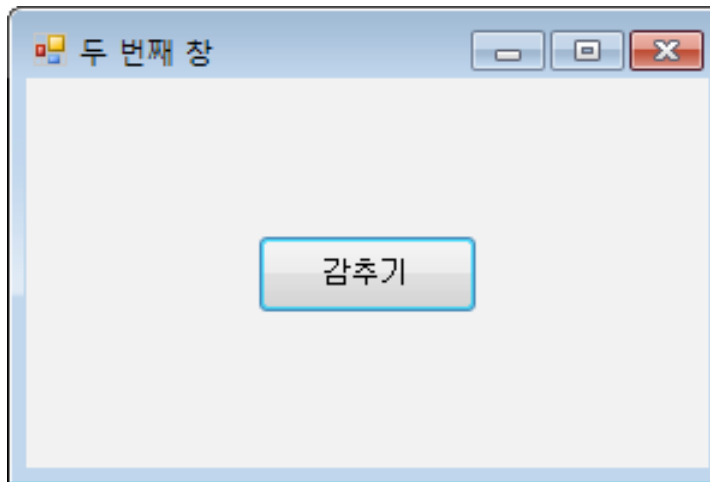
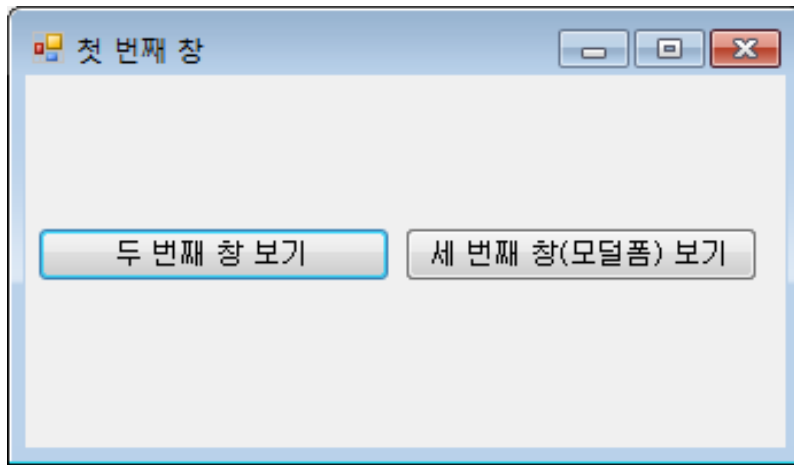
- 두 번째 폼

```
private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
}
```

- 세 번째 폼

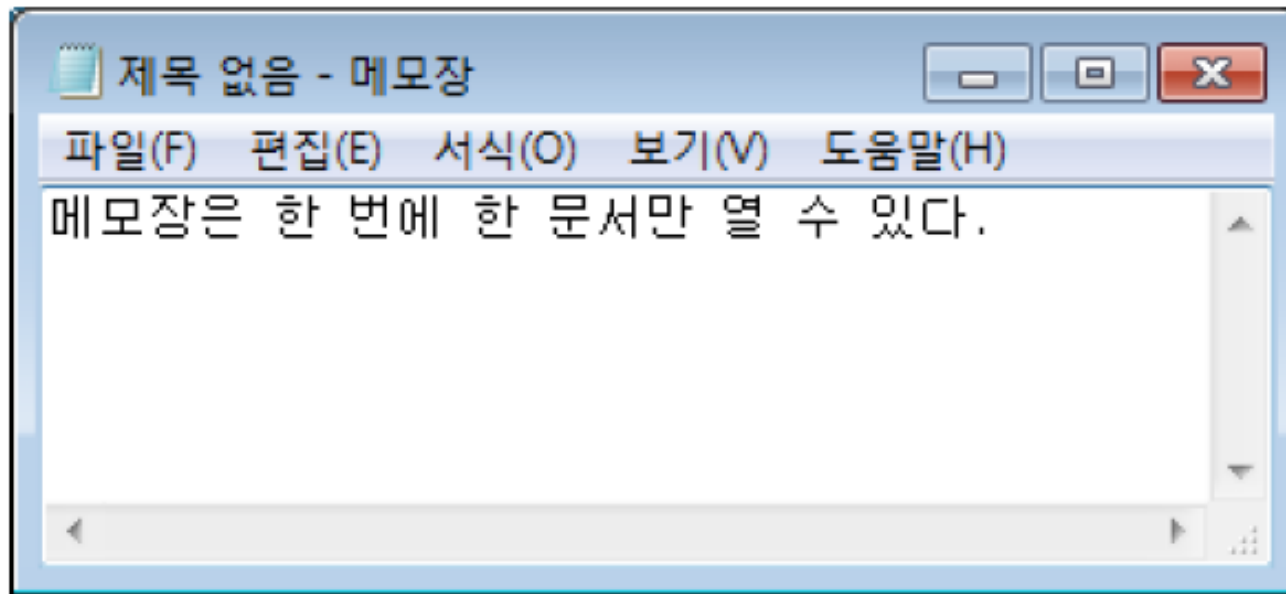
```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

5) 실행 결과 확인



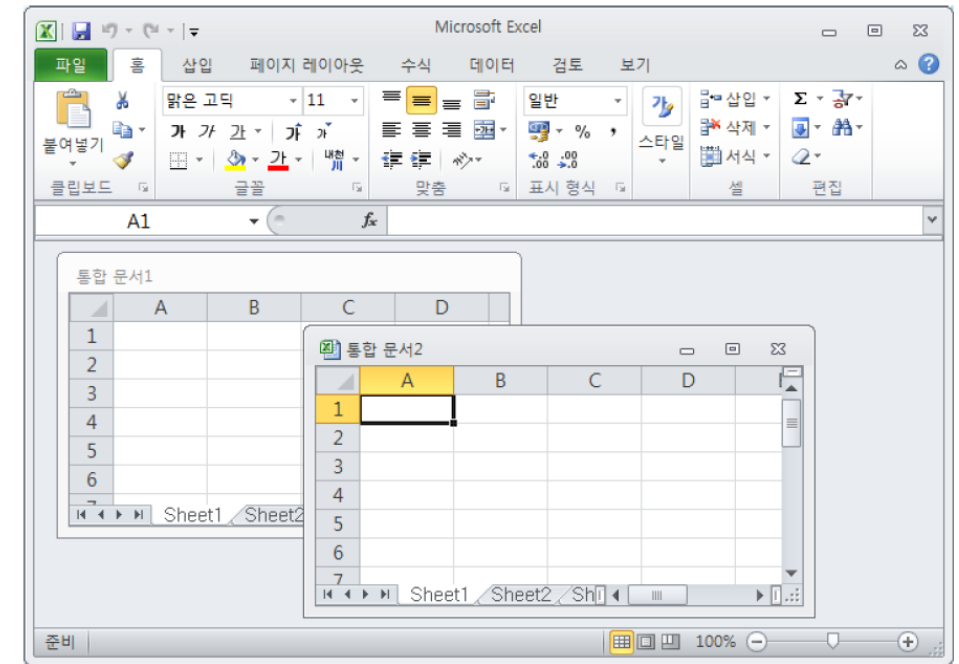
- 폼의 형태
 - SDI 폼 (Single Document Interface Form)
 - 하나의 폼을 독립적으로 사용하며 폼 안에 다른 폼이 있을 수 없다.
 - 메모장

그림 9-1 SDI 폼의 예(메모장)



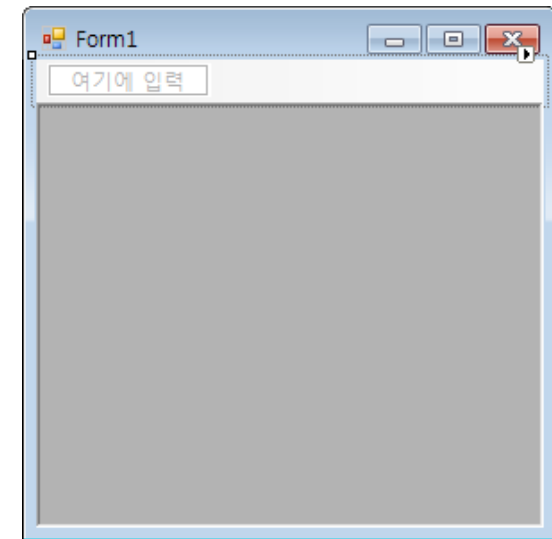
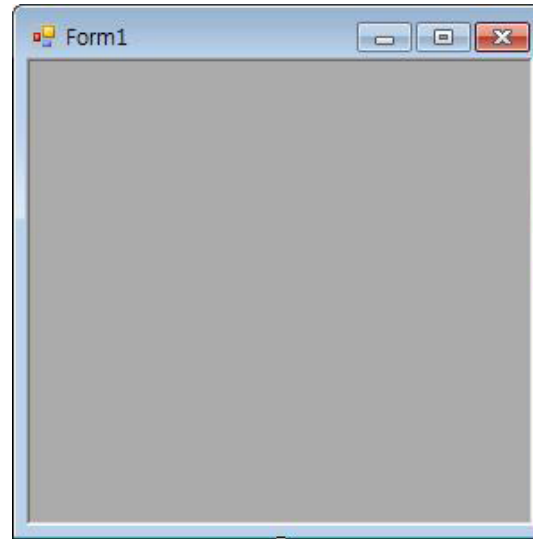
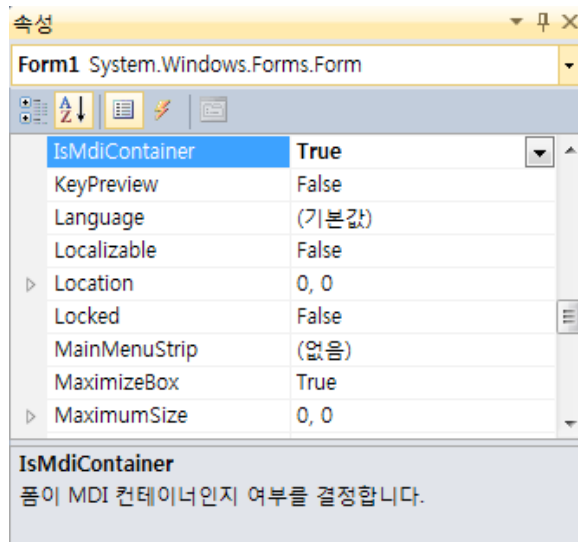
- MDI 폼 (Multiple Document Interface Form)
 - SDI와 달리 메인 폼(부모 폼) 하나에 여러 자식 폼을 포함한다.
 - Microsoft Word, Excel ...
 - 자식 폼을 부모 폼 영역 밖으로 이동할 수 없다.
 - 일반적으로 SDI 폼과 MDI 폼을 동시에 사용하지 않는다.
 - 자식 폼을 최소화하면 윈도우 작업 표시줄에 나타나지 않고 부모 폼의 하단에 표시된다.
 - 자식 폼을 최대화하면 부모 폼을 벗어나지 않으면서 부모 폼의 제목 표시 줄에 부모 폼의 제목과 함께 자식 폼의 제목이 표시된다.

그림 9-2 MDI 폼의 예(Microsoft Excel 2010)

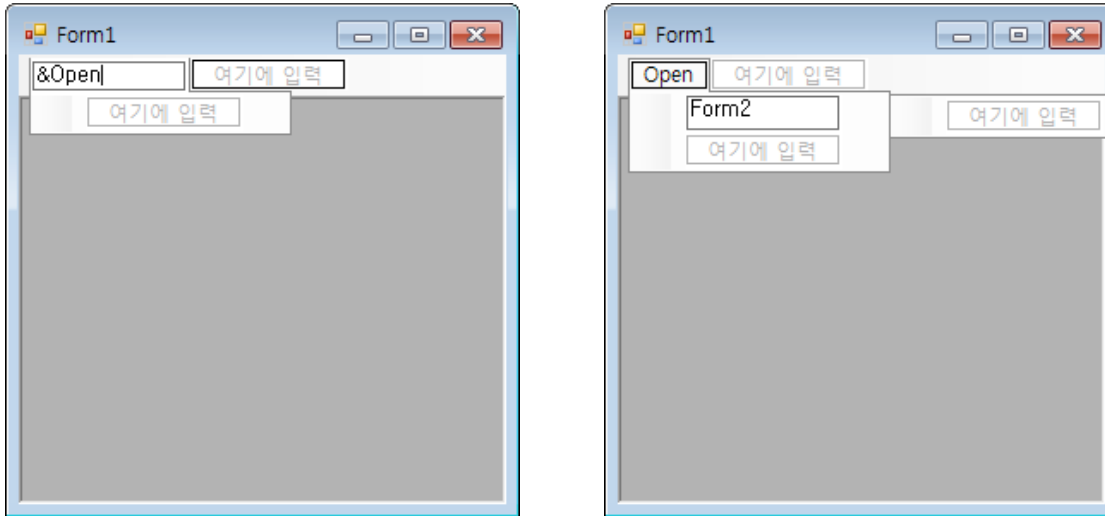


- 실습 – MDI 폼 만들고 자식 폼 추가하기

- 1) 새 프로젝트를 열고 자식 폼으로 사용할 폼을 하나 추가한다.
- 2) 2개의 폼 중에서 Form1을 부모 폼으로 사용하기 위해 IsMdiContainer 속성값을 True로 설정한다.
- 3) 메뉴스트립 컨트롤을 Form1로 끌어와 삽입한다.



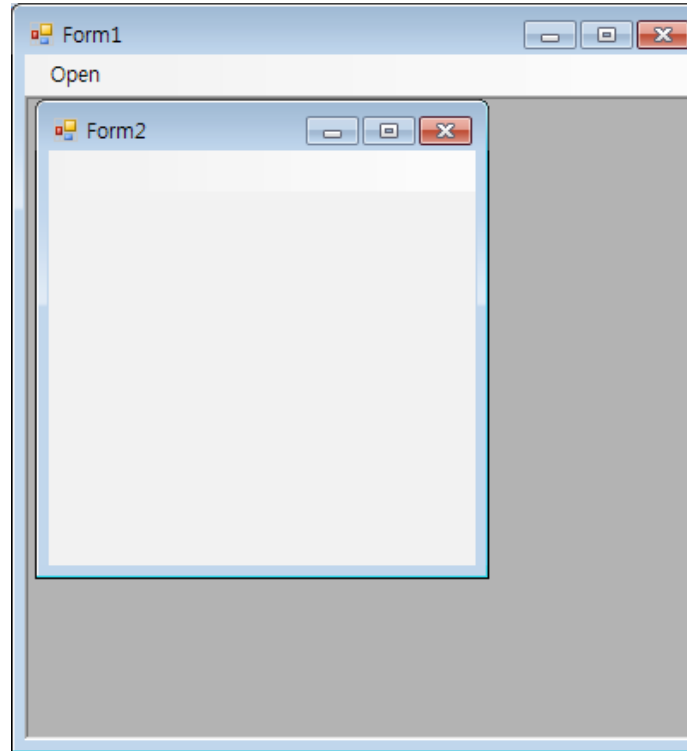
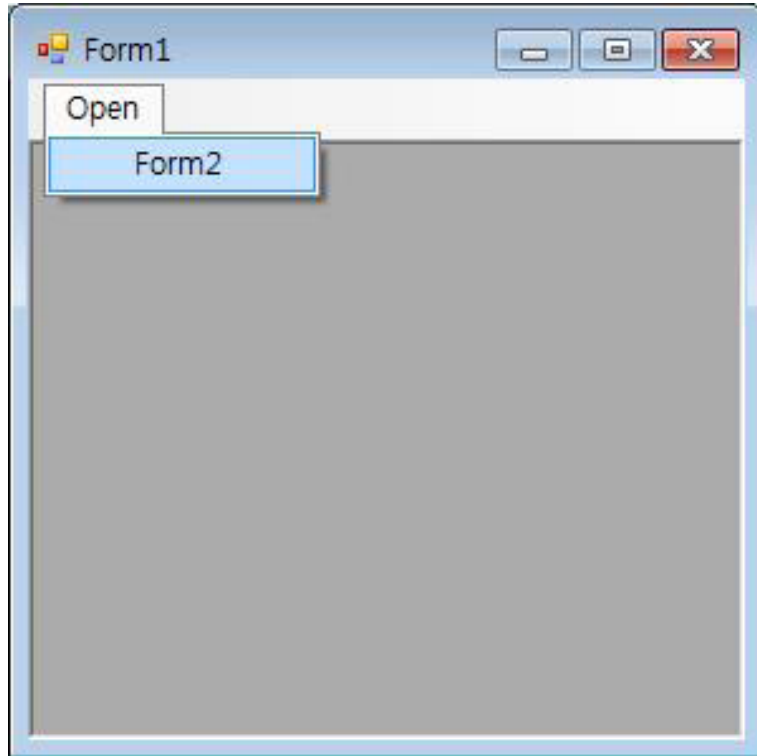
4) 새로 생긴 메뉴스트립에 '&Open' 을 입력한다. 같은 방법으로 Form2라는 하위 메뉴를 만든다. (기호 '&' 다음에 오는 문자가 해당 메뉴의 단축키가 된다.)



5) 코드 작성

```
private void form2ToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    form2.MdiParent = this;
    form2.Show();
}
```

6) 실행 결과 확인



- 메뉴의 종류와 구조
 - 메뉴
 - 사용자가 선택할 수 있는 명령을 주제별로 묶어 놓은 것
 - 메뉴 바에 위치하며 관련 명령어를 가지고 있다.
 - 주 메뉴, 하위 메뉴, 팝업 메뉴가 있다.

그림 9-3 메뉴의 종류와 기본 구조

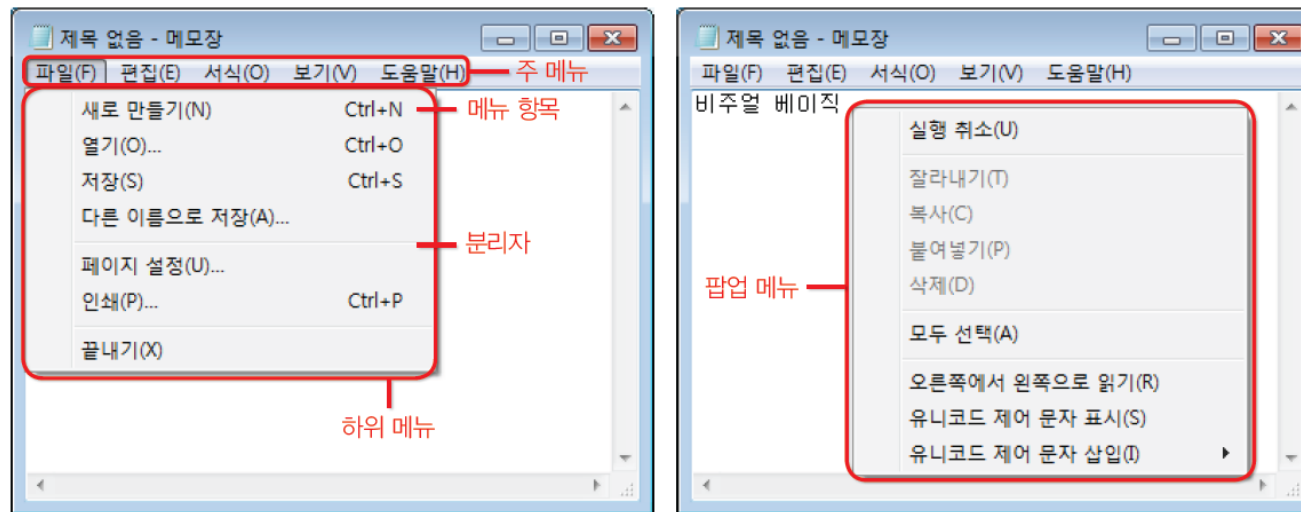


그림 9-4 메뉴에 있는 표시의 의미



- 메뉴스트립 컨트롤을 이용한 메뉴 작성
 - 메뉴스트립 컨트롤은 프로그램의 메뉴를 관리하는 데 사용한다.

표 9-5 메뉴스트립 컨트롤의 주요 속성

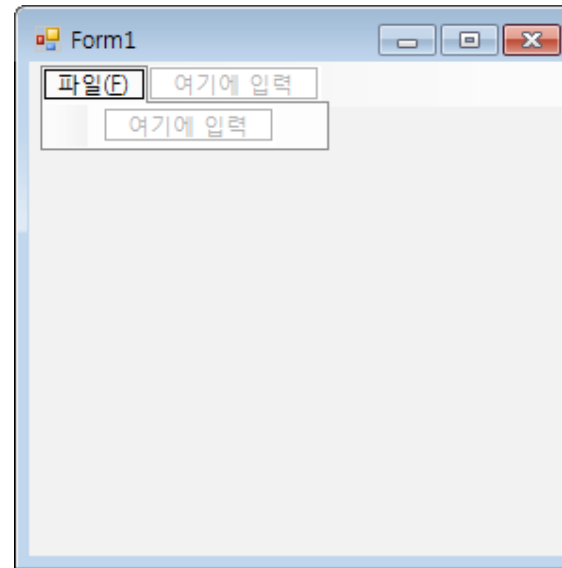
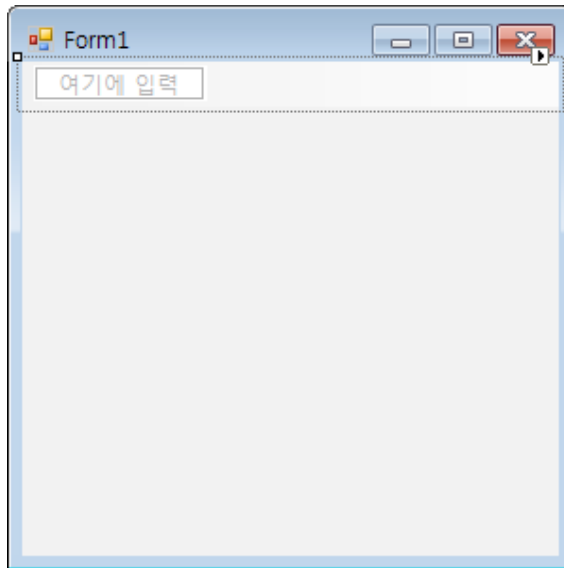
이름	설명
Text	메뉴 또는 메뉴 표시 줄에 나타낼 이름을 설정한다. 입력란에 -(줄표)를 삽입하면 메뉴에 구분 선이 들어간다. 해당 메뉴의 선택키를 지정하려면 메뉴 앞에 &를 삽입한다.
(Name)	메뉴 개체의 이름을 설정한다.
MergelIndex	같은 이름의 메뉴 가운데 일치하는 것을 찾고 위치를 지정하는 데 사용된다.
MergeAction	MergelIndex가 일치할 때 수행할 작업을 지정한다. <ul style="list-style-type: none">• Append : 일치 항목 메뉴 끝에 추가• Insert : 일치 항목 바로 앞에 삽입• Replace : 일치 항목을 해당 항목으로 교체• Remove : 일치 항목 제거• MatchOnly : 아무것도 하지 않음
ShortcutKeys	메뉴의 단축키를 지정한다.
ShortcutKeyDisplay	메뉴의 단축키로 표시될 문자열을 지정한다.
TextAlign	메뉴 항목 문자열의 위치를 설정한다.
Checked	메뉴 이름 앞에 체크 표시를 설정한다.
CheckOnClick	메뉴를 클릭했을 때 체크 표시가 변경되는지 여부를 설정한다.

이름	설명
CheckState	초기 체크 표시의 상태를 설정한다.
DisplayStyle	메뉴의 표시 방법을 설정한다. <ul style="list-style-type: none">• ImageAndText : 이미지와 텍스트 표시• Image : 이미지만 표시• Text : 텍스트만 표시• None : 표시하지 않음
DoubleClickEnabled	더블클릭했을 때 이벤트를 발생시킬지 여부를 설정한다.
DropDown	메뉴를 클릭했을 때 보일 ToolStrip을 설정한다.
DropDownItems	DropDown에서 설정한 ToolStrip의 항목을 설정한다.
Enabled	메뉴의 활성화 여부를 설정한다. 체크되어 있으면 사용자가 해당 메뉴를 선택할 수 있는 상태가 된다.
Visible	메뉴 항목의 표시 여부를 설정한다.
ToolTipText	해당 메뉴에 표시할 ToolTip 메시지를 설정한다.

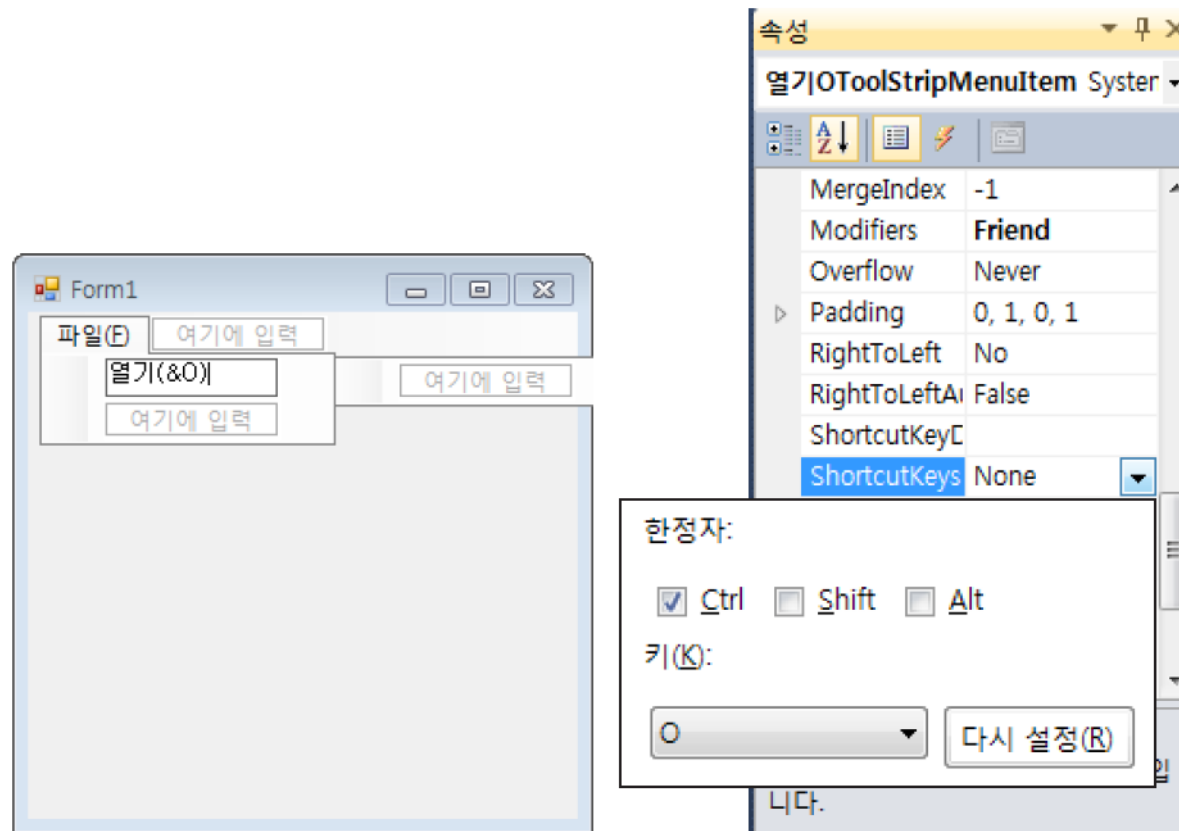
- 실습 – 메뉴스트립 컨트롤을 이용하여 메뉴 만들기

1) 새 폼에 메뉴스트립 컨트롤을 끌어와 추가한다.

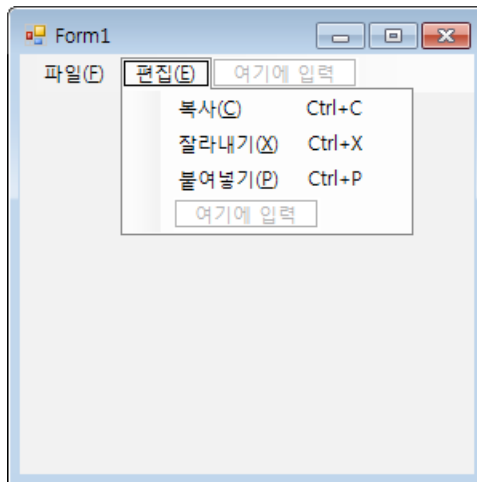
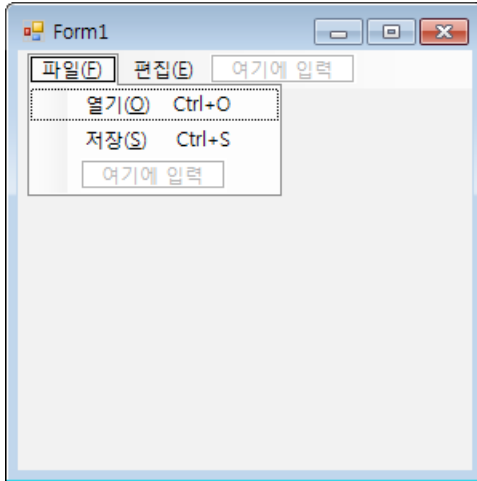
2) 첫 번째 칸에 '파일(&F)', Name 속성에 menuFile을 입력하고 다른 속성은 기본값으로 둔다.



3) ' 파일(F)' 하단 칸에 ' 열기(&O)'를 입력하고 Name 속성에 'menuOpen'을 입력한다. 단축키(ShortcutKeys)는 'Ctrl + O'로 설정한다.



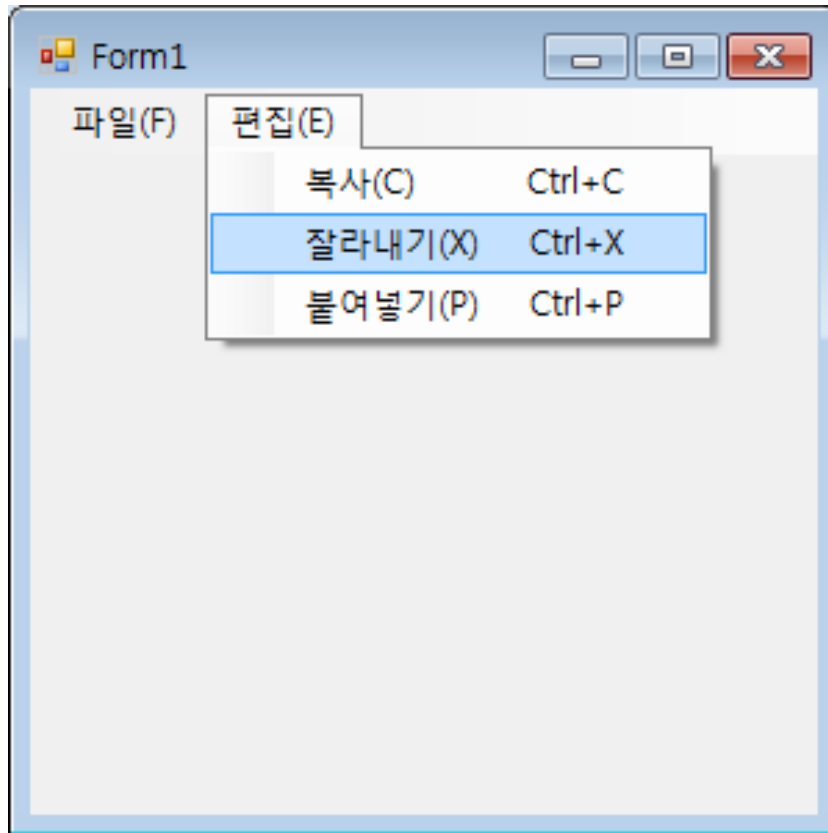
4) 다음 표를 참고하여 다른 메뉴도 모두 완성한다.



메뉴	하위 메뉴	속성	속성값
파일	열기	Text	파일(&F)
		(Name)	menuFile
		ShortcutKeys	(없음)
	저장	Text	열기(&O)
		(Name)	menuOpen
		ShortcutKeys	Ctrl+O
		Text	저장(&S)
		(Name)	menuSave
		ShortcutKeys	Ctrl+S

메뉴	하위 메뉴	속성	속성값
편집		Text	편집(&E)
		(Name)	menuEdit
		ShortCutKeys	(없음)
	복사	Text	복사(&C)
		(Name)	menuCopy
		ShortCutKeys	Ctrl+C
	잘라내기	Text	잘라내기(&X)
		(Name)	menuCut
		ShortCutKeys	Ctrl+X
	붙여넣기	Text	붙여넣기(&P)
		(Name)	menuPaste
		ShortCutKeys	Ctrl+P

5) 실행 결과 확인



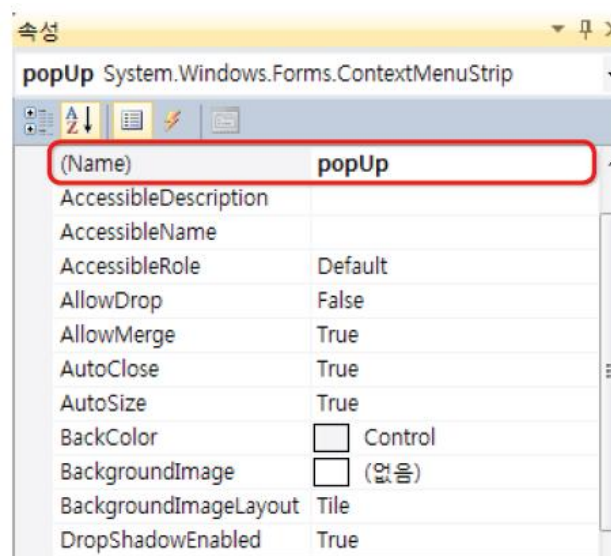
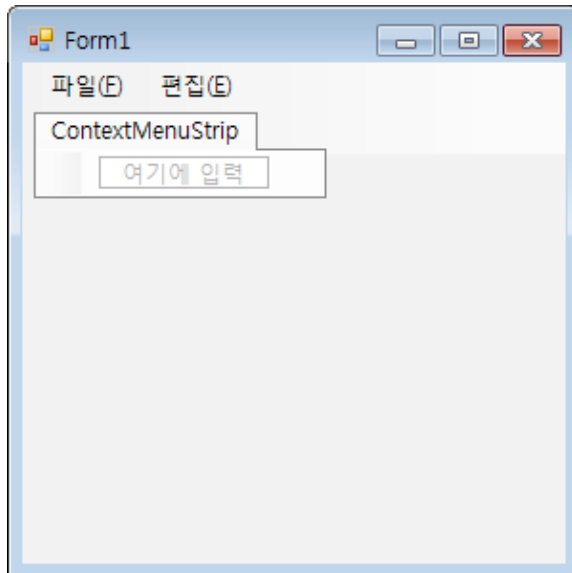
- 컨텍스트메뉴스트립 컨트롤을 이용한 팝업 메뉴 작성

표 9-6 컨텍스트메뉴스트립 컨트롤의 주요 속성

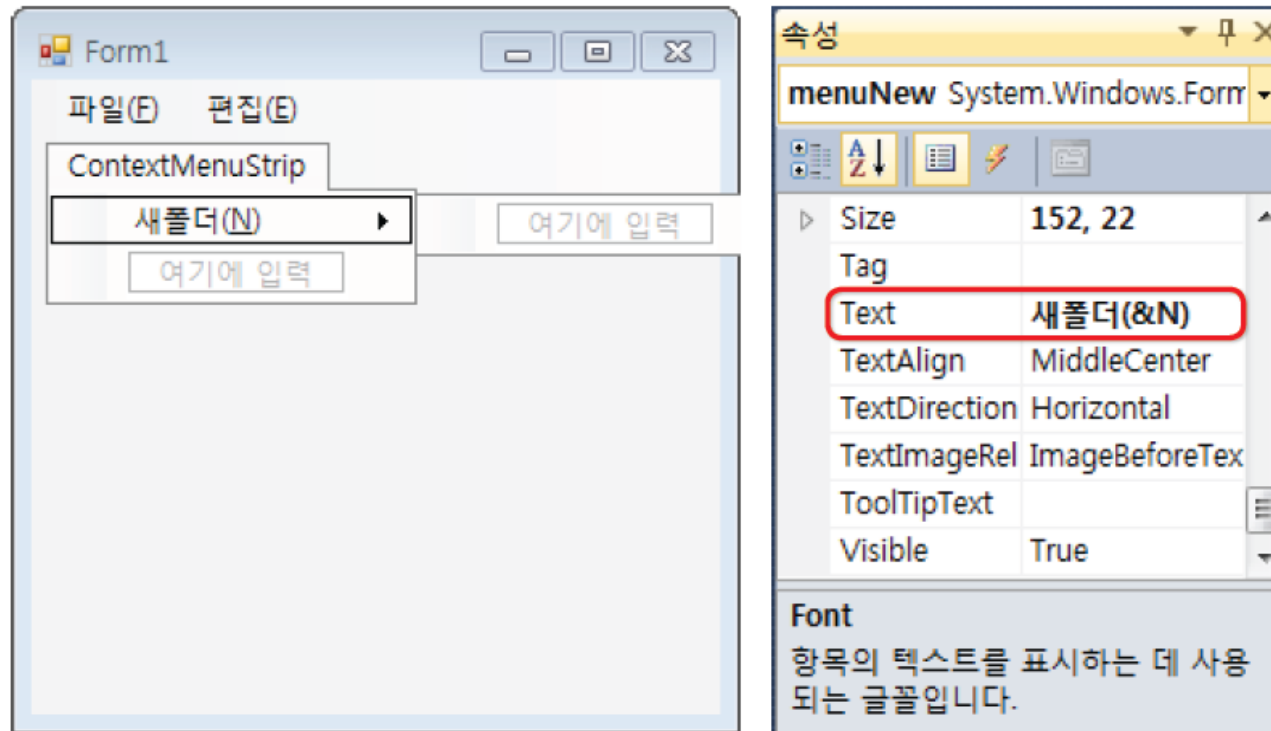
이름	설명
Text	메뉴에 나타낼 이름을 설정한다.
(Name)	메뉴 개체의 이름을 설정한다.
MergeIndex	같은 이름의 메뉴 가운데 일치하는 것을 찾고 위치를 지정하는 데 사용된다.
MergeAction	MergeIndex가 일치할 때 수행할 작업을 지정한다. <ul style="list-style-type: none">• Append : 일치 항목 메뉴 끝에 추가• Insert : 일치 항목 바로 앞에 삽입• Replace : 일치 항목을 해당 항목으로 교체• Remove : 일치 항목 제거• MatchOnly : 아무것도 하지 않음
ShortcutKeys	메뉴의 단축키를 지정한다.
ShortcutKeyDisplay	메뉴의 단축키로 표시될 문자열을 지정한다.
TextAlign	메뉴 항목 문자열의 위치를 설정한다.
Checked	메뉴 이름 앞에 체크 표시를 설정한다.
CheckOnClick	메뉴를 클릭했을 때 체크 표시가 변경되는지 여부를 설정한다.
CheckState	초기 체크 표시의 상태를 설정한다.

DisplayStyle	메뉴의 표시 방법을 설정한다. <ul style="list-style-type: none">• ImageAndText : 이미지와 텍스트 표시• Image : 이미지만 표시• Text : 텍스트만 표시• None : 표시하지 않음
DoubleClickEnabled	더블클릭했을 때 이벤트를 발생시킬지 여부를 설정한다.
DropDown	메뉴를 클릭했을 때 보일 ToolStrip을 설정한다.
DropDownItems	DropDown에서 설정한 ToolStrip의 항목을 설정한다.
Enabled	메뉴의 활성화 여부를 설정한다. 체크되어 있으면 사용자가 해당 메뉴를 선택할 수 있는 상태가 된다.
Visible	메뉴 항목의 표시 여부를 설정한다.
ToolTipText	해당 메뉴에 표시할 ToolTip 메시지를 설정한다.

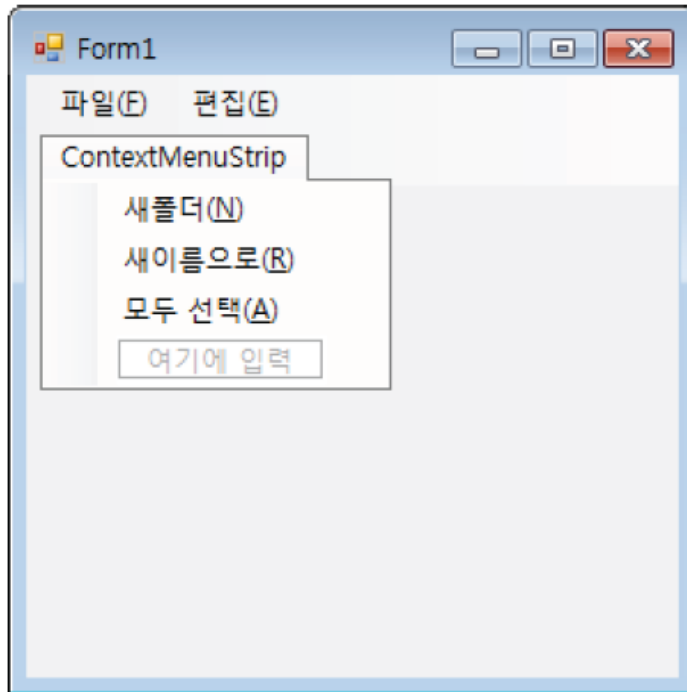
- 실습 – 컨텍스트메뉴스트립 컨트롤을 이용하여 팝업 메뉴 만들기
- 1) 위의 실습에서, 도구 상자의 '메뉴 및 도구 모음' 그룹에 있는 컨텍스트메뉴스트립 컨트롤을 추가한다.
 - 2) 최상위 메뉴의 Text를 설정하지 않고 (Name) 속성에 'popUp'을 입력한다.



3) 다음 하위 메뉴의 Text 속성에 '새폴더(&N)'를 속성에 'menuNew'를 입력하고 Visible 속성을 'True'로 설정한다.



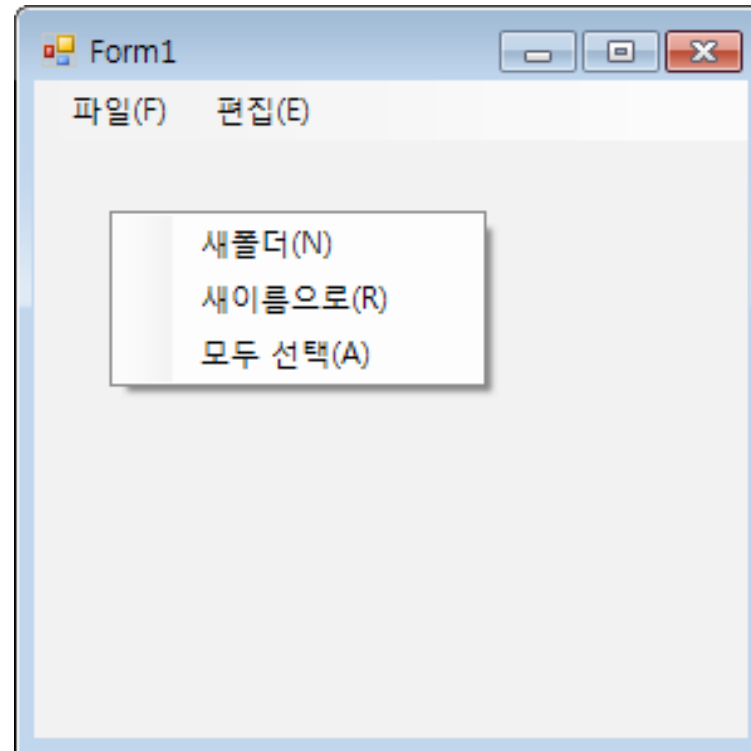
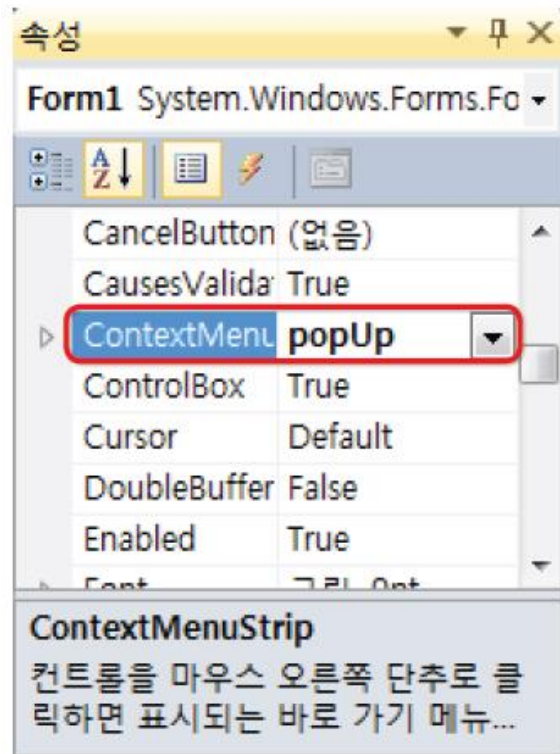
4) 다음 표를 참고하여 다른 팝업 메뉴도 모두 완성한다.



하위 메뉴	속성	속성값
새이름으로	Text	새이름으로(&R)
	Name	menuRename
모두 선택	Text	모두 선택(&A)
	Name	menuSelectAll

5) 마우스 오른쪽 버튼에 대한 이벤트 프로시저를 설정한다. Form의 ContextMenuStrip 속성에 새로 작성한 popUp을 할당한다.

6) 실행 결과 확인



• 툴스트립 컨트롤을 이용한 도구 모음 만들기

그림 9-5 Microsoft Visual Basic 2010 Express의 도구 모음

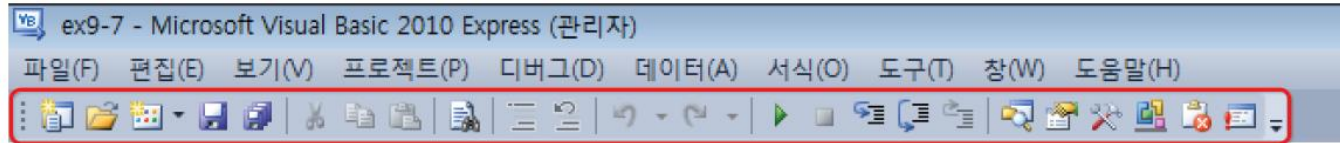


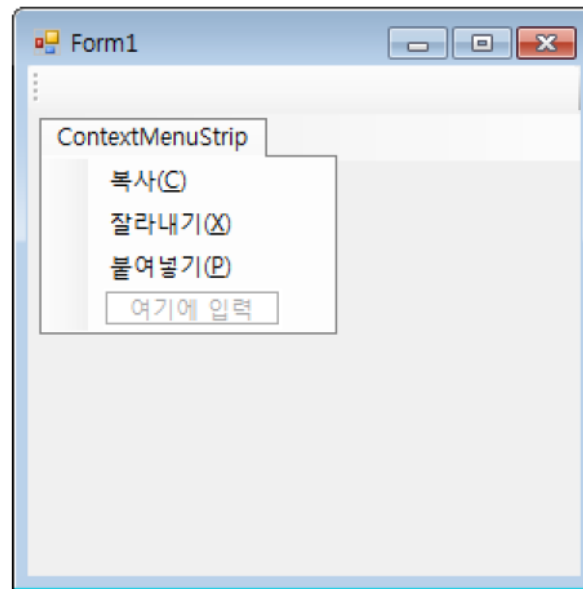
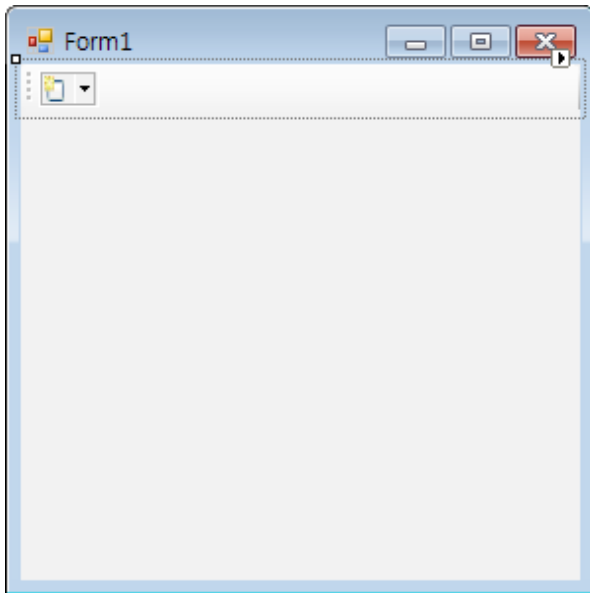
표 9-7 툴스트립 컨트롤의 주요 속성

이름	설명
(Name)	도구 모음의 이름을 설정한다.
ContextMenuStrip	도구 모음에서 마우스 오른쪽 버튼을 눌렀을 때 나타낼 팝업 메뉴를 지정한다.
Dock	도구 모음의 위치를 결정한다. • Top : 품의 상단 끝에 위치시킴 • Left : 품의 왼쪽 끝에 위치시킴 • Right : 품의 오른쪽 끝에 위치시킴 • Bottom : 품의 하단 끝에 위치시킴 • Fill : 품의 크기에 맞춰 위치시킴 • None : 사용자가 자유롭게 위치시킴
GripMargin	도구 모음 그림의 여백을 지정한다.
GripStyle	도구 모음 그림을 표시할지 여부를 설정한다.
Items	툴스트립에 나타낼 항목을 설정한다.
Text	컨트롤에 연결된 텍스트를 설정한다.
Enabled	도구 모음을 활성화할지 여부를 설정한다.
Visible	도구 모음의 표시 여부를 설정한다.

- 실습 - 툴스트립 컨트롤을 이용하여 도구 모음 만들기

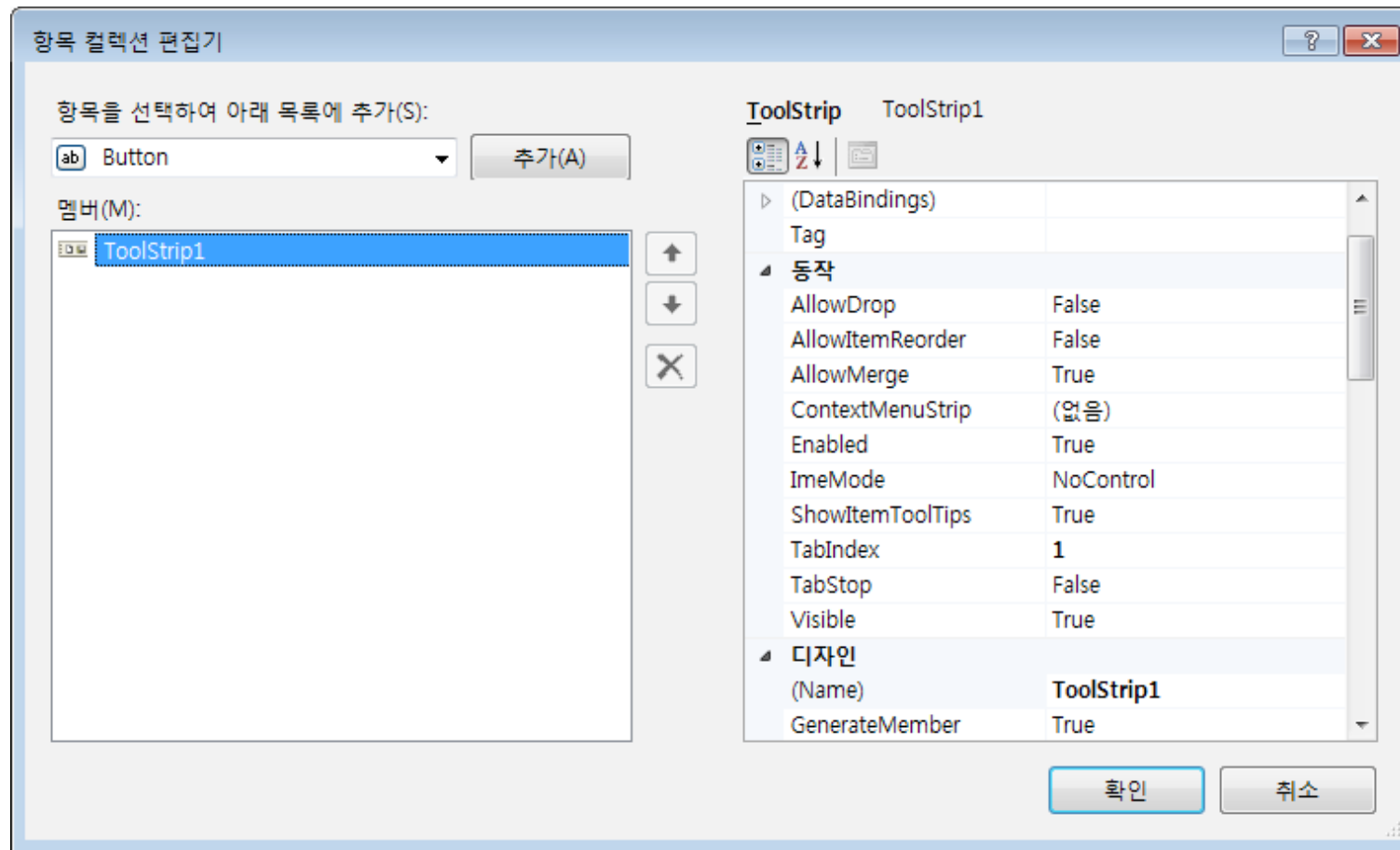
1) 새 프로젝트를 열고 Form1에 툴스트립 컨트롤을 추가한다.

2) DropDown 버튼에서 사용할 컨텍스트메뉴스트립 컨트롤을 하나 추가하고 다음 표와 같이 속성을 설정한다.

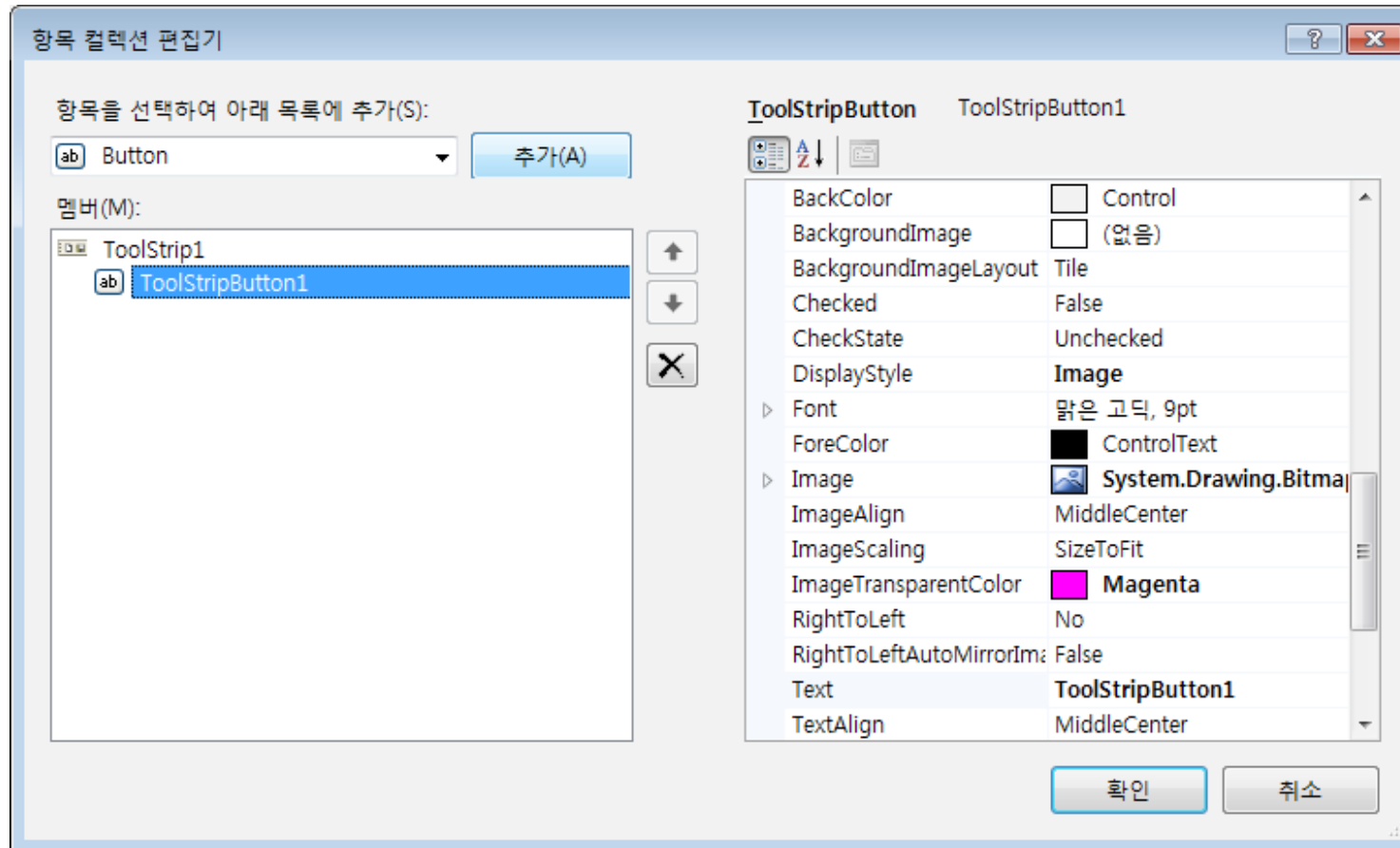


하위 메뉴	속성	속성값
복사	Text	복사(&C)
	Name	menuCopy
잘라내기	Text	잘라내기(&X)
	Name	menuCut
붙여넣기	Text	붙여넣기(&P)
	Name	menuPaste

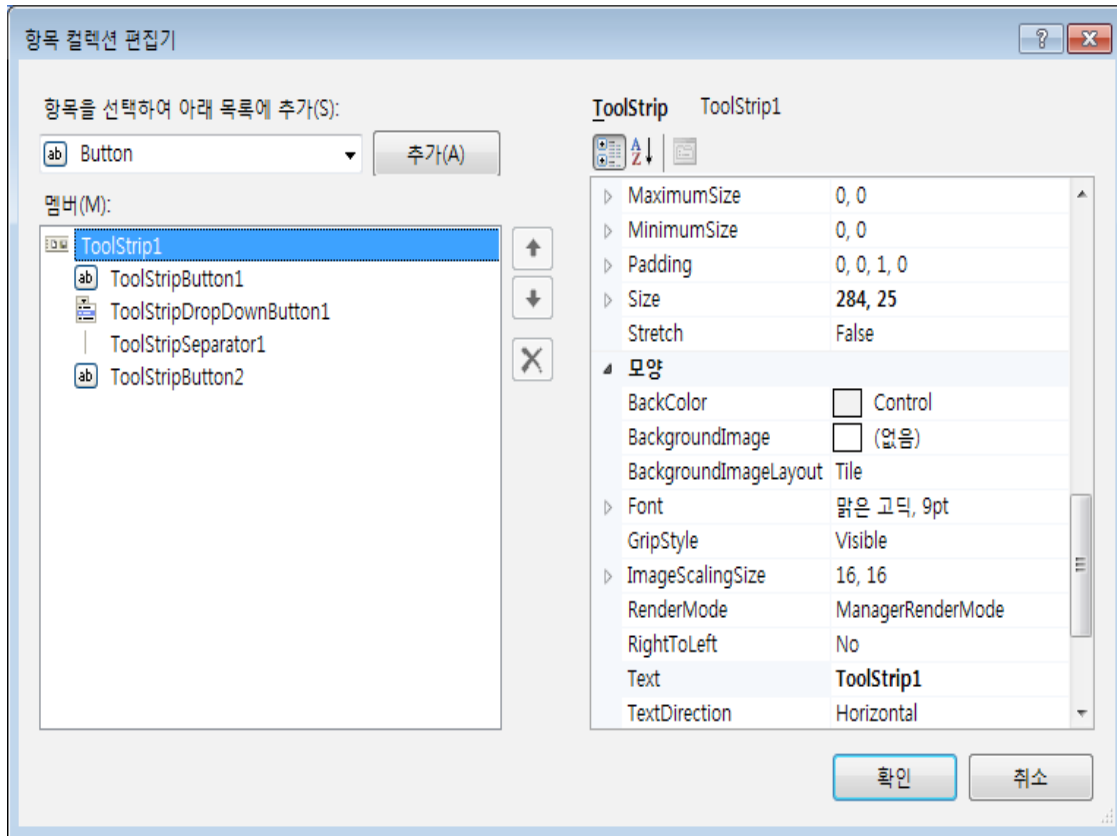
3) ToolStrip1 의 Items 속성에서 (컬렉션)의 옆  버튼을 클릭하여 [항목 컬렉션 편집기] 대화 상자로 이동한다.



4) Button 항목을 선택하여 <추가>를 클릭하고 도구 모음에 멤버를 추가한다.

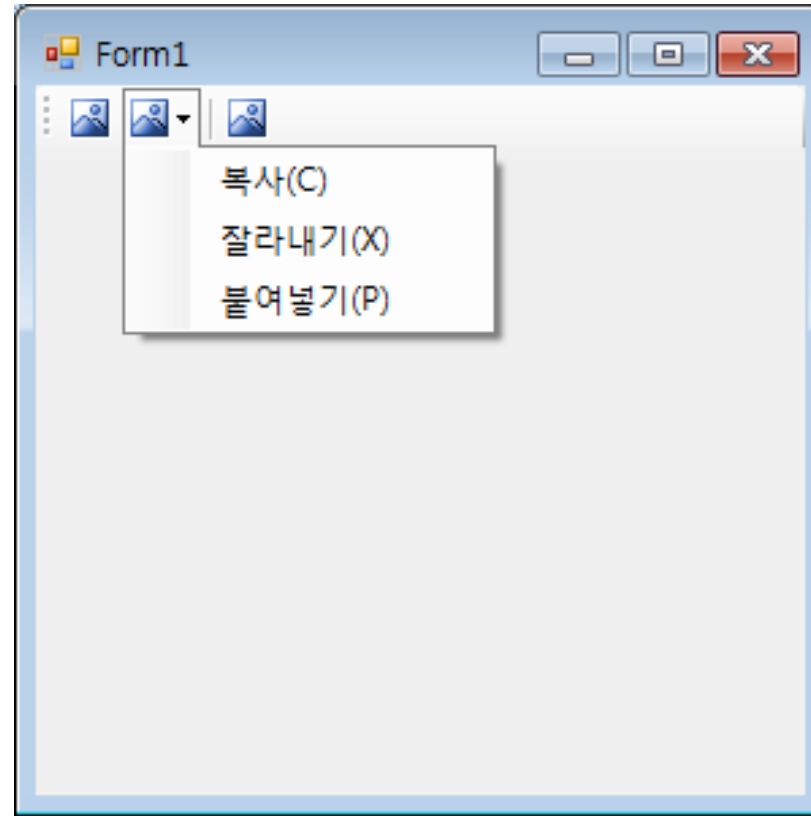
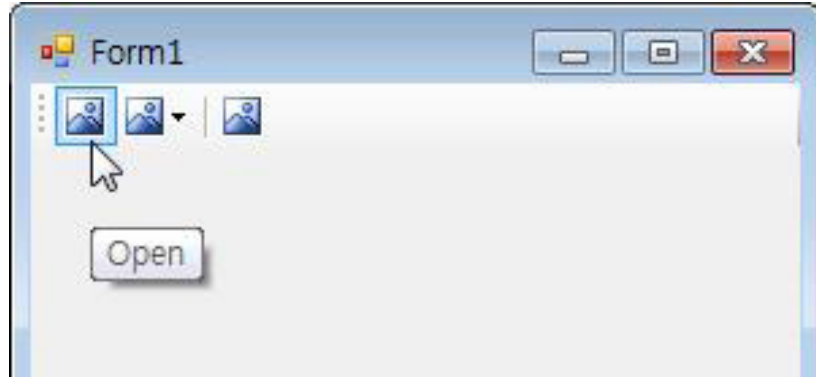


5) 항목을 바꿔가며 멤버를 추가하고 다음 표와 같이 속성을 설정한다.



멤버	속성	속성값
ToolStripButton1	(Name)	toolOpen
	Text	열기
	Image	버튼 선택 후 원하는 그림 사용
	ToolTipText	Open
ToolStripDropDownButton1	(Name)	toolEdit
	Text	편집
	Image	버튼 선택 후 원하는 그림 사용
	DropDown	popUp
ToolStripButton2	(Name)	toolExit
	Text	종료
	Image	버튼 선택 후 원하는 그림 사용
	ToolTipText	Exit

6) 실행 결과 확인



• 스테이터스스트립 컨트롤을 이용한 상태 표시 줄 만들기

그림 9-6 Microsoft Excel 2010의 상태 표시 줄

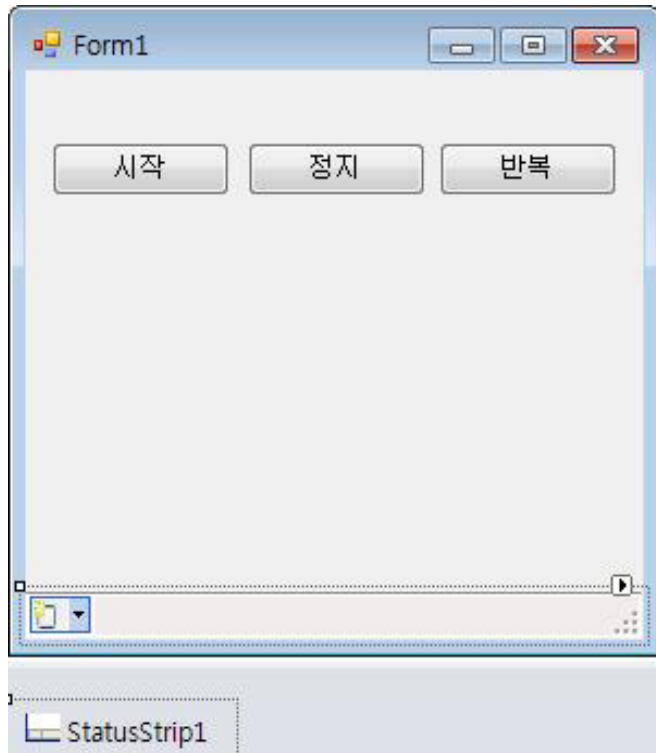


표 9-8 스테이터스스트립 컨트롤의 주요 속성

이름	설명
(Name)	상태 표시 줄의 이름을 설정한다.
ContextMenuStrip	상태 표시 줄에서 마우스 오른쪽 버튼을 눌렀을 때 나타낼 팝업 메뉴를 지정한다.
Dock	상태 표시 줄의 위치를 결정한다. <ul style="list-style-type: none"> • Top : 폼의 상단 끝에 위치시킴 • Left : 폼의 왼쪽 끝에 위치시킴 • Right : 폼의 오른쪽 끝에 위치시킴 • Bottom : 폼의 하단 끝에 위치시킴 • Fill : 폼의 크기에 맞춰 위치시킴 • None : 사용자가 자유롭게 위치시킴
GripMargin	상태 표시 줄 그림의 여백을 지정한다.
GripStyle	상태 표시 줄 그림을 표시할지 여부를 설정한다.
Items	스테이터스스트립에 나타낼 항목을 설정한다.
Text	컨트롤에 연결된 텍스트를 설정한다.
Enabled	상태 표시 줄을 활성화할지 여부를 설정한다.
Visible	상태 표시 줄의 표시 여부를 설정한다.


- 실습 - 스테이터스스트립을 이용하여 상태 표시 줄 만들기

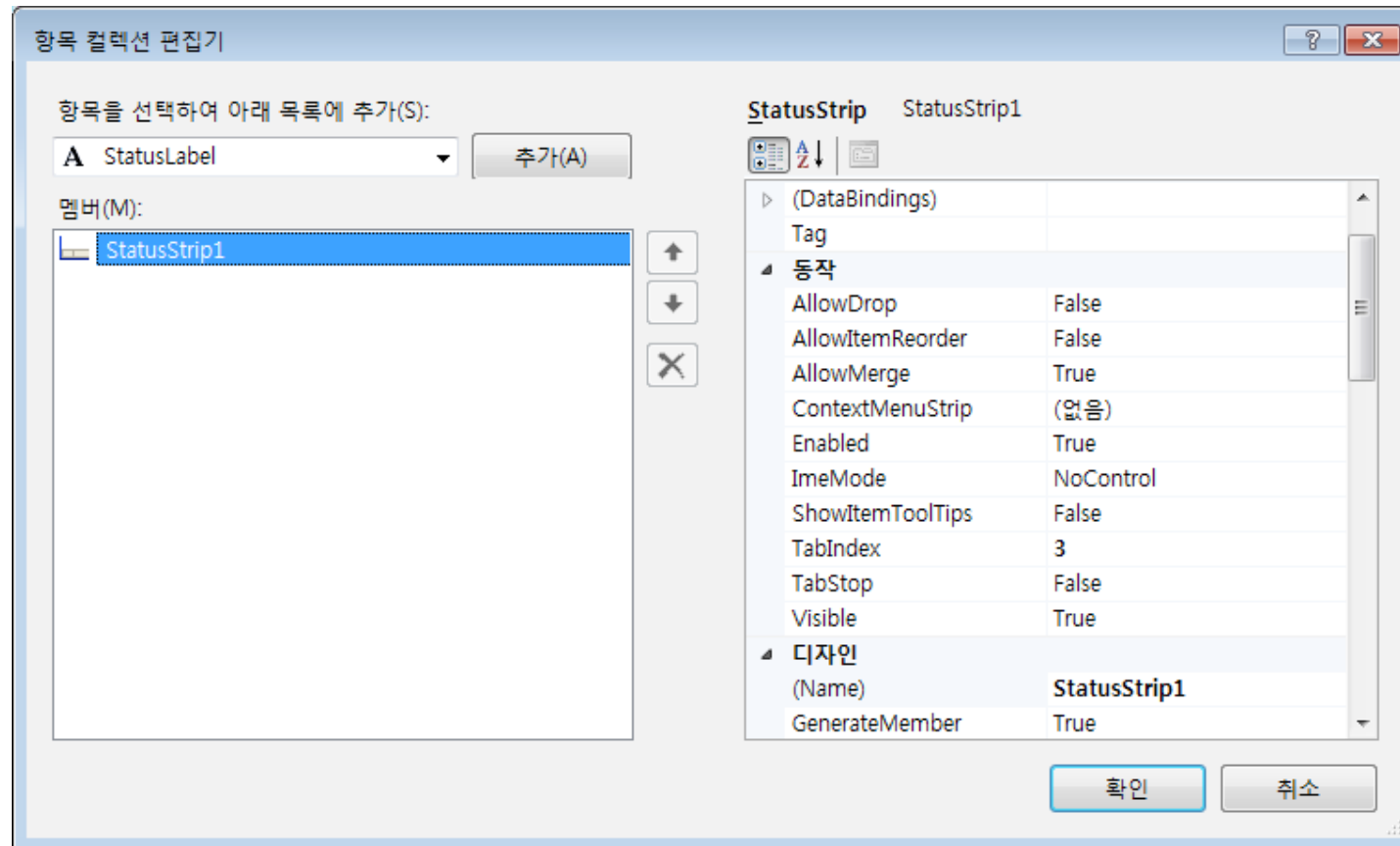
1) 윈도우 폼 디자인



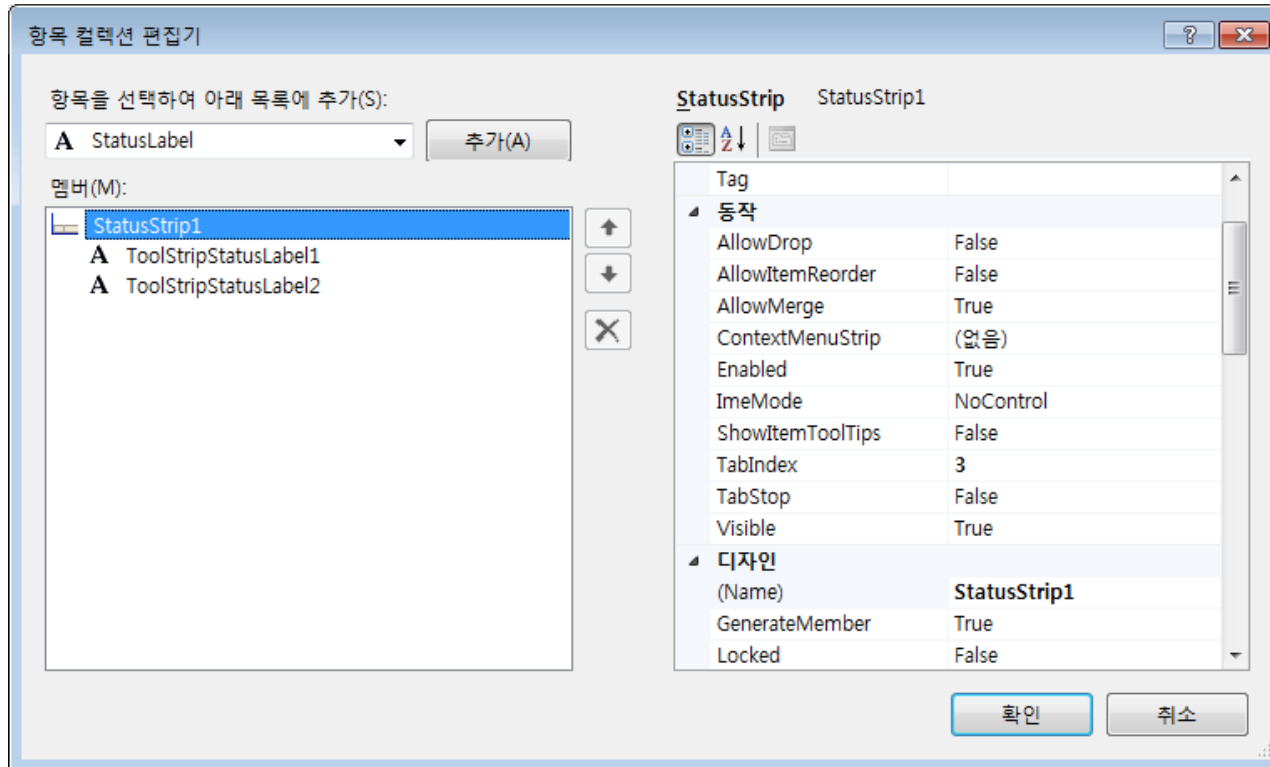
2) 속성 설정

컨트롤	속성	속성값
Button1	Text	시작
Button2	Text	정지
Button3	Text	반복

3) 스테이터스스트립 컨트롤의 Items 속성에서 (컬렉션) 옆의  버튼을 클릭하여 [항목 컬렉션 편집기] 대화 상자로 이동한다.



4) StatusLabel을 추가하고 아래 표와 같이 속성을 설정한다.



멤버	속성	속성값
ToolStripStatusLabel1	(Name)	ToolStripStatusLabel1
	Text	대기
ToolStripStatusLabel2	(Name)	ToolStripStatusLabel2
	Text	일방

5) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "재생";
}

private void button2_Click(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text = "정지";
}

private void button3_Click(object sender, EventArgs e)
{
    if (toolStripStatusLabel2.Text == "일방")
    {
        toolStripStatusLabel2.Text = "반복";
    }
    else
    {
        toolStripStatusLabel2.Text = "일방";
    }
}
```

6) 실행 결과 확인

