

윈도우 폼 컨트롤 01

HyoJoon Han
동국대학교
han6343@dongguk.edu



윈도우 폼 컨트롤이란?

버튼 컨트롤

레이블 컨트롤

텍스트박스 컨트롤

체크박스, 라디오버튼, 그룹박스 컨트롤

리스트박스, 콤보박스 컨트롤

체크박스, 라디오버튼, 그룹박스 컨트롤

- 체크박스(CheckBox), 라디오버튼(RadioButton)
 - 데이터 입력 시 문자를 직접 입력하지 않고 사용자가 선택할 수 있는 예시를 보여주어 마우스 클릭만으로 필요한 데이터를 입력할 수 있도록 하는 컨트롤
- 그룹박스(GroupBox)
 - 여러 개의 컨트롤을 그룹으로 묶어 서로 연관된 항목임을 보여줄 때 사용

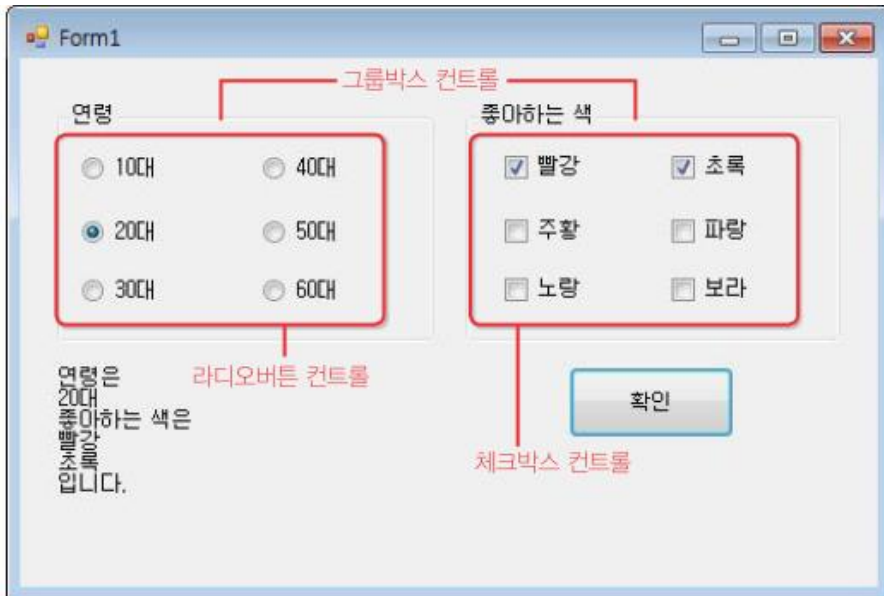
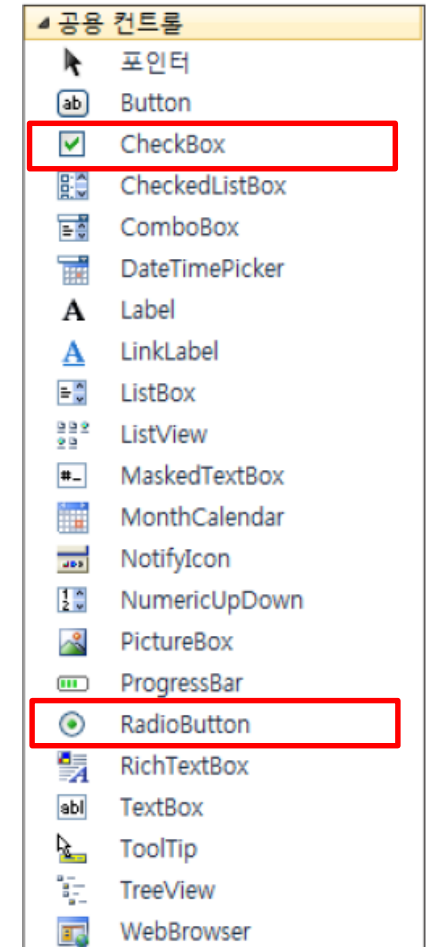


그림 3-2 공용 컨트롤의 종류



- 체크박스과 라디오버튼 컨트롤
 - 체크박스 컨트롤
 - 그룹으로 묶인 여러 항목 중 하나 이상을 선택할 때 사용
 - 참(True)과 거짓(False) 중 한 가지 상태만 가질 수 있다.
 - 라디오버튼 컨트롤
 - 여러 항목 중 하나만 선택할 때 사용
 - 동시에 여러 항목을 선택할 수 있는 체크박스 컨트롤과 달리 한 항목만 선택할 수 있다.

표 3-5 체크박스과 라디오버튼 컨트롤의 비교

체크박스 컨트롤	라디오버튼 컨트롤
여러 항목 선택 가능	한 항목만 선택 가능
세 가지 상태 사용	두 가지 상태 사용
선택의 의미는 True	선택의 의미는 True

• 체크박스과 라디오버튼 컨트롤

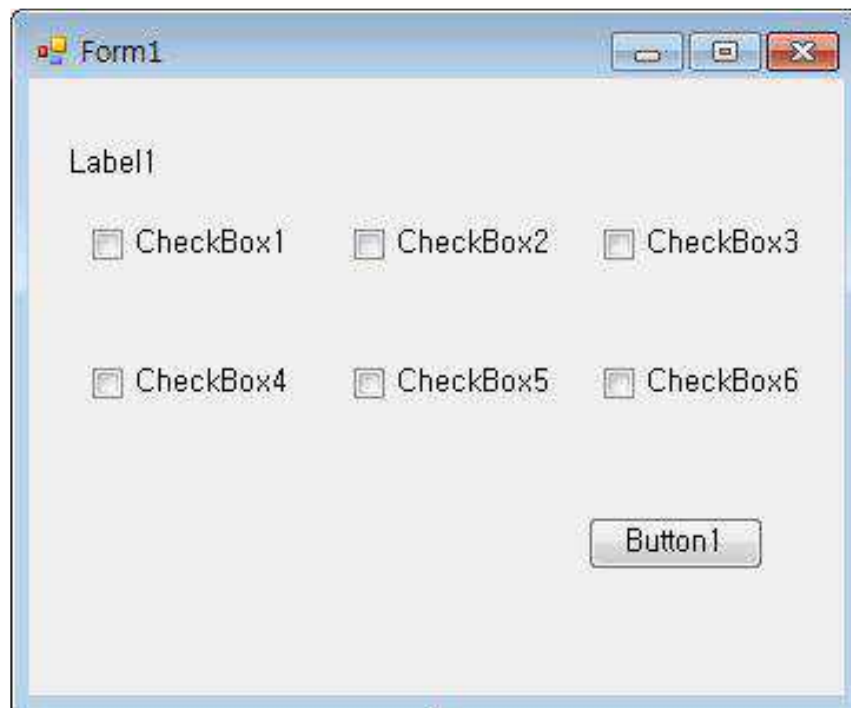
표 3-6 체크박스과 라디오버튼 컨트롤의 주요 속성

파란색은 기본값

속성	설명
AutoSize	체크박스과 라디오버튼의 크기 자동 조정 여부를 설정한다. · True : 자동 조정 가능 · False : 임의 조정 가능
BackColor	체크박스과 라디오버튼의 배경색을 설정한다.
BackgroundImage	체크박스과 라디오버튼에 사용할 배경 이미지를 설정한다.
Enabled	체크박스과 라디오버튼의 활성화 여부를 설정한다. · True : 체크박스과 라디오버튼 활성화 · False : 체크박스과 라디오버튼 비활성화
Text	체크박스과 라디오버튼 옆에 나오는 문자열을 설정한다.
Checked	체크박스과 라디오버튼의 선택 여부를 나타낸다. · True : 선택된 경우 · False : 선택되지 않은 경우
체크박스의 경우	<div> <div>CheckState</div> <div> 체크박스의 초기 선택 상태를 설정한다. · Unchecked : 체크되지 않은 상태로 설정 · Checked : 체크된 상태로 설정 · Indeterminate : 체크되고 회색인 상태(비활성화 상태) </div> </div>
	<div> <div>ThreeState</div> <div> 체크박스에 Indeterminate을 사용 가능하게 하는지 여부를 설정한다. · True : 세 가지 선택 상태 · False : 두 가지 선택 상태 </div> </div>

- 실습 - 체크박스로 원하는 색 출력하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Label1	Text	좋아하는 색을 선택하세요.
CheckBox1	Text	빨강
CheckBox2	Text	주황
CheckBox3	Text	노랑
CheckBox4	Text	초록
CheckBox5	Text	파랑
CheckBox6	Text	보라
Label2	Text	(빈칸)
Button1	Text	확인(&O)

3) 코드 작성

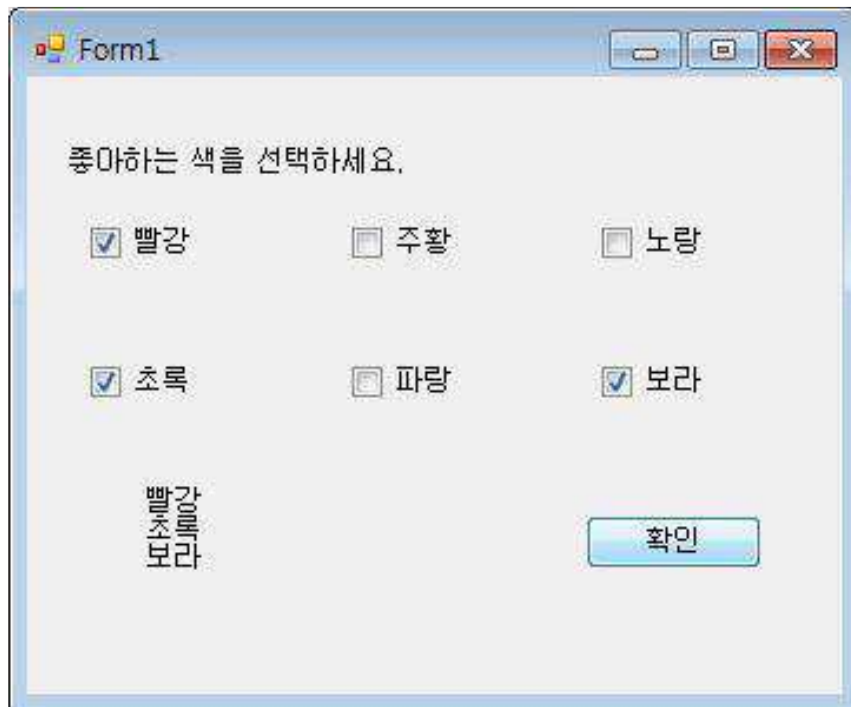
```
private void button1_Click(object sender, EventArgs e)
{
    String str="";

    if (checkBox1.Checked)
        str += checkBox1.Text + "Wn";
    if (checkBox2.Checked)
        str += checkBox2.Text + "Wn";
    if (checkBox3.Checked)
        str += checkBox3.Text + "Wn";
    if (checkBox4.Checked)
        str += checkBox4.Text + "Wn";
    if (checkBox5.Checked)
        str += checkBox5.Text + "Wn";
    if (checkBox6.Checked)
        str += checkBox6.Text + "Wn";

    label2.Text = str;
}
```

4) 실행 결과 확인

- 레이블에 선택한 색상명이 출력된다.



- 실습 – 라디오버튼으로 연령 선택하기

1) 윈도우 폼 디자인



2) 속성 설정

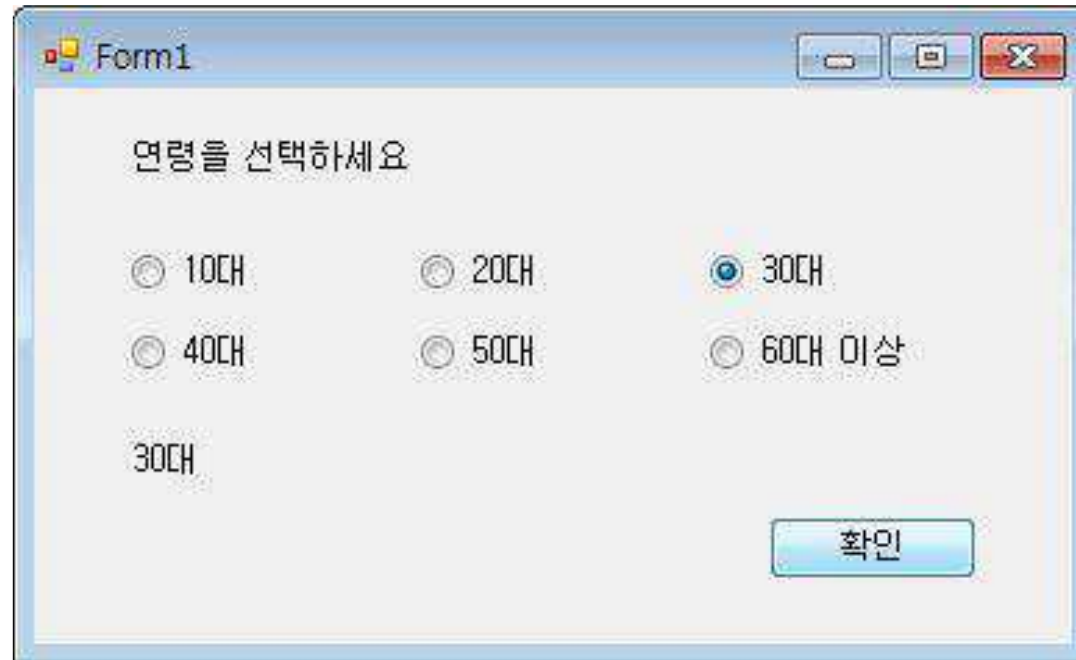
컨트롤	속성	속성값
Label1	Text	연령을 선택하세요.
RadioButton1	Text	10대
RadioButton2	Text	20대
RadioButton3	Text	30대
RadioButton4	Text	40대
RadioButton5	Text	50대
RadioButton6	Text	60대 이상
Label2	Text	(빈칸)
Button1	Text	확인(&O)

3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked)
        label2.Text = radioButton1.Text;
    if (radioButton2.Checked)
        label2.Text = radioButton2.Text;
    if (radioButton3.Checked)
        label2.Text = radioButton3.Text;
    if (radioButton4.Checked)
        label2.Text = radioButton4.Text;
    if (radioButton5.Checked)
        label2.Text = radioButton5.Text;
    if (radioButton6.Checked)
        label2.Text = radioButton6.Text;
}
```

4) 실행 결과 확인

- 선택한 라디오버튼에 따라 레이블의 문자열이 변경된다.



- 그룹박스 컨트롤
 - 관련된 컨트롤을 묶어 그룹을 만들 때 사용
 - 도구 상자의 컨테이너 그룹에서 찾을 수 있다.
- 컨테이너
 - 그룹박스과 같이 다른 개체를 하나 이상 포함하는 것
 - 컨테이너가 움직이면 안에 포함된 개체도 함께 움직여 컨테이너 밖으로 나오지 않는다.

표 3-7 그룹박스 컨트롤의 주요 속성

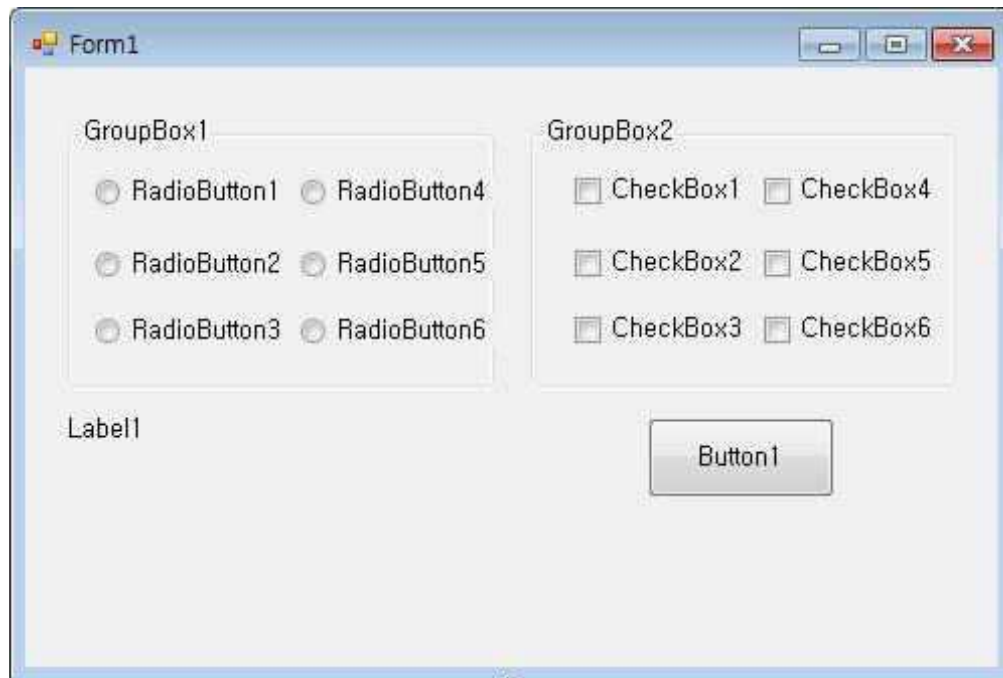
파란색은 기본값

속성	설명
Dock	그룹박스의 위치를 설정한다. · None : 임의 배치 가능 · Top, Bottom, Left, Right, Fill : 위, 아래, 왼쪽, 오른쪽, 화면 가득 등 위치 설정
Enabled	그룹박스의 사용 여부를 설정한다. · True : 그룹박스와 그 안에 포함된 컨트롤 사용 가능 · False : 그룹박스와 그 안에 포함된 컨트롤 사용 불가
Text	그룹박스의 제목 줄에 나타나는 문자열을 설정한다.
Visible	그룹박스를 화면에 표시할지 여부를 설정한다. · True : 그룹박스와 그 안에 포함된 모든 컨트롤을 화면에 표시함 · False : 그룹박스와 그 안에 포함된 모든 컨트롤을 화면에 표시하지 않음

- 그룹박스를 옮길 때 안에 포함된 컨트롤도 함께 움직이는 방법
 - 먼저 그룹박스 컨트롤을 만든다.
 - 그룹박스 컨트롤을 선택한 상태에서 그 안에 넣을 컨트롤을 선택하여 그룹박스 안에서 드래그하여 그려 놓는다(더블 클릭해서 넣으면 안된다).

- 실습 - 그룹박스 컨트롤로 컨트롤 묶기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
GroupBox1	Text	연령
GroupBox2	Text	좋아하는 색
RadioButton1~RadioButton6	Text	10대, 20대, 30대, 40대, 50대, 60대 이상
CheckBox1~CheckBox6	Text	빨강, 주황, 노랑, 초록, 파랑, 보라
Label1	Text	(빈칸)
Button1	Text	확인(&O)

3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    String str = "연령은\n";

    if (radioButton1.Checked)
        str += radioButton1.Text;
    if (radioButton2.Checked)
        str += radioButton2.Text;
    if (radioButton3.Checked)
        str += radioButton3.Text;
    if (radioButton4.Checked)
        str += radioButton4.Text;
    if (radioButton5.Checked)
        str += radioButton5.Text;
    if (radioButton6.Checked)
        str += radioButton6.Text;

    str += "\n좋아하는색은\n";

    if (checkBox1.Checked)
        str += checkBox1.Text + "\n";
    if (checkBox2.Checked)
        str += checkBox2.Text + "\n";
    if (checkBox3.Checked)
        str += checkBox3.Text + "\n";
    if (checkBox4.Checked)
        str += checkBox4.Text + "\n";
    if (checkBox5.Checked)
        str += checkBox5.Text + "\n";
    if (checkBox6.Checked)
        str += checkBox6.Text + "\n";
    label1.Text = str;
}
```

4) 실행 결과 확인

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains two main sections: "연령" (Age) and "좋아하는 색" (Favorite Color). The "연령" section has six radio buttons arranged in a 3x2 grid, with "20대" (20s) selected. The "좋아하는 색" section has six checkboxes arranged in a 3x2 grid, with "빨강" (Red) and "초록" (Green) checked. Below the "연령" section, there is a text label that reads "연령은 20대, 좋아하는 색은 빨강과 초록입니다." (Age is 20s, favorite colors are red and green). A blue "확인" (Confirm) button is located at the bottom center of the form.

연령		좋아하는 색	
<input type="radio"/> 10대	<input type="radio"/> 40대	<input checked="" type="checkbox"/> 빨강	<input checked="" type="checkbox"/> 초록
<input checked="" type="radio"/> 20대	<input type="radio"/> 50대	<input type="checkbox"/> 주황	<input type="checkbox"/> 파랑
<input type="radio"/> 30대	<input type="radio"/> 60대	<input type="checkbox"/> 노랑	<input type="checkbox"/> 보라

연령은 20대, 좋아하는 색은 빨강과 초록입니다.

확인

• 리스트박스 컨트롤

- 사용자가 직접 입력하지 않고 목록으로 제시된 항목 중에서 원하는 것을 선택하도록 유도할 때 사용
- 폼에 디자인된 목록의 크기보다 항목의 개수가 많으면 스크롤바가 자동으로 추가됨

그림 3-2 공용 컨트롤의 종류

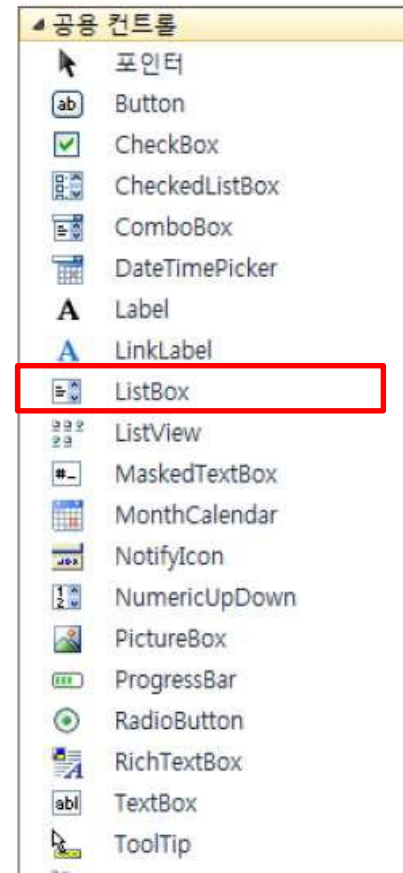


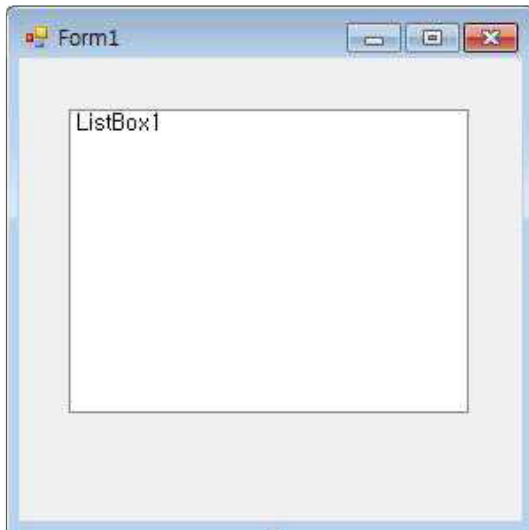
표 3-8 리스트박스 컨트롤의 주요 속성과 메소드


파란색은 기본값

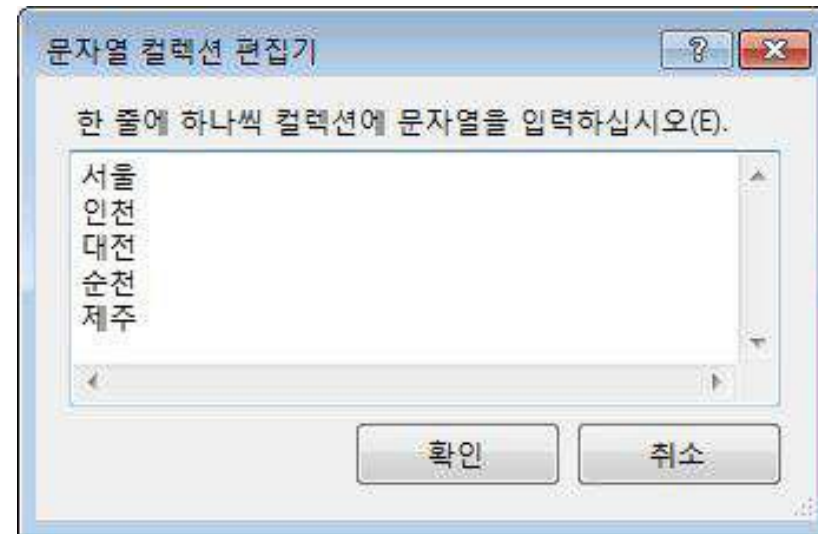
종류	이름	설명
속성	HorizontalScrollBar	가로 스크롤바를 사용할지 여부를 설정한다. · True : 수평 스크롤바를 사용함 · False : 수평 스크롤바를 사용하지 않음
	Items	리스트박스에 나타날 항목을 추가하거나 제거한다.
	SelectionMode	여러 항목의 선택 가능 여부를 설정한다. · None : 선택 불가 · One : 한 항목만 선택 가능 · MultiSimple : 마우스로 클릭할 때마다 다중 선택이 되고 선택된 항목을 다시 클릭하면 선택 해제 · MultiExtended : 마우스로 드래그하거나 [Ctrl] 또는 [Shift]를 누르고 마우스로 클릭하여 다중 선택 가능([Ctrl]은 비연속 항목을 다중 선택 및 해제, [Shift]는 연속 항목을 다중 선택)
	Sorted	목록을 사전 순으로 정렬한 것인지를 결정하며, 이는 속성 창에서만 설정할 수 있다. · False : 목록에 입력하는 순서대로 표시 · True : 목록에 입력하는 순서대로 표시되지 않고 자동 정렬, 한글을 먼저 오름차순으로 정렬한 다음 영문을 정렬하며 대·소문자를 구분하지 않음
	Enabled	리스트박스의 사용 여부를 설정한다. · True : 사용 · False : 불가
	Add	목록에 문자열을 추가한다.
메소드	Remove	인덱스 번호를 사용하여 항목을 삭제한다.
	Refresh	목록을 최근에 입력한 항목으로 수정한다.
	Clear	모든 목록을 삭제한다.

- 실습 - 리스트박스에 항목 추가하기

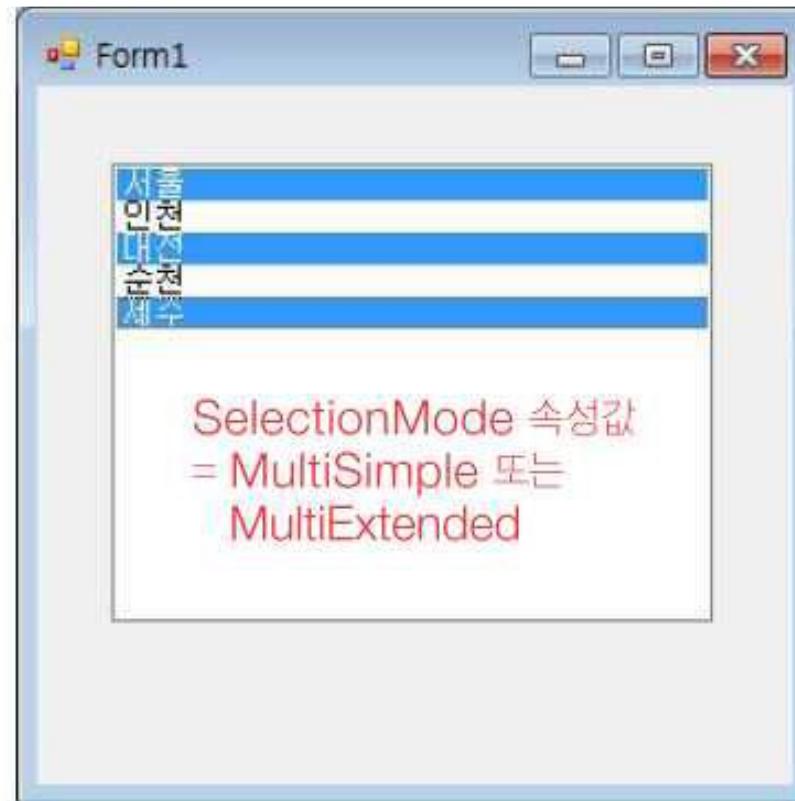
1) 윈도우 폼 디자인



2) 속성 창에서 items의  을 클릭한다.
'문자열 컬렉션 편집기' 에 리스트박스
컨트롤에서 보여줄 항목을 입력한 후
<확인>을 클릭한다.



3) F5를 눌러 프로그램을 실행한다. 실행 화면에서 마우스로 클릭하여 항목을 선택할 수 있다. 리스트 컨트롤의 'SelectionMode' 속성 값을 'MultiSimple'이나 'Shift', 'Ctrl'을 이용하여 다중 선택할 수 있는 'MultiExtended'로 설정하여 여러 항목을 선택할 수 있다.



• 콤보박스 컨트롤

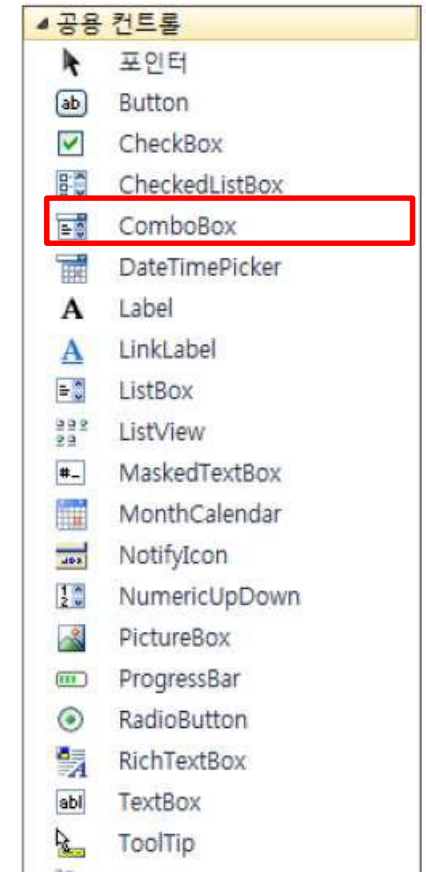
- 텍스트박스와 리스트박스의 기능이 합쳐진 형태
- 리스트박스처럼 목록을 선택하고 텍스트박스처럼 내용을 입력할 수도 있다.
- 화살표 모양의 목록 단추를 누르면 드롭다운(펼침) 목록이 나타나는데, 사용자는 그 목록 중 원하는 내용을 선택하여 입력한다.

표 3-9 콤보박스 컨트롤의 주요 속성

파란색은 기본값

속성	설명
Text	콤보박스에 입력되거나 선택된 값을 표시한다.
DropDownStyle	<p>콤보박스의 형태를 설정한다.</p> <ul style="list-style-type: none"> · DropDown : 일반적인 콤보박스 생성, 드롭다운 목록을 이용하여 항목을 선택하거나 직접 입력 가능 · Simple : 드롭다운 목록 버튼을 사용할 수 없어 단순 텍스트박스처럼 보임, 다음 목록을 보려면 키보드의 상하 방향 키를 사용해야 하고 목록 부분이 모두 보이게 하려면 Height 속성 값을 충분히 크게 주어야 함 · DropDownList : 직접 입력하지 못하고 드롭다운 목록에서만 선택 가능

그림 3-2 공용 컨트롤의 종류



- 실습 – 콤보박스를 이용하여 주소 입력하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값	컨트롤	속성	속성값
Label1	Text	성명	ComboBox1	Items	Mr., Ms.
Label2	Text	호칭		Text	(빈칸)
Label3	Text	거주지	ComboBox2	Sorted	True
TextBox1	Text	(빈칸)		Text	(빈칸)
Button1	Text	확인(&O)			

3) 코드 작성

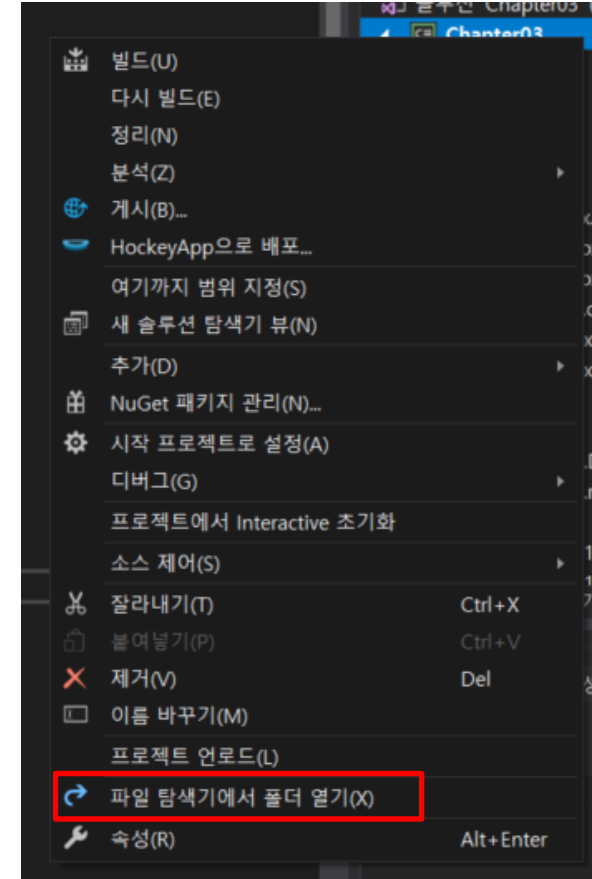
```
private void comboBoxEx_Load(object sender, EventArgs e)
{
    /*
    comboBox2.Items.Add("서울특별시");
    comboBox2.Items.Add("인천광역시");
    comboBox2.Items.Add("대전광역시");
    comboBox2.Items.Add("대구광역시");
    comboBox2.Items.Add("부산광역시");
    comboBox2.Items.Add("광주광역시");
    comboBox2.Items.Add("울산광역시");
    comboBox2.Items.Add("경기도");
    comboBox2.Items.Add("충청도");
    comboBox2.Items.Add("강원도");
    comboBox2.Items.Add("경상도");
    comboBox2.Items.Add("전라도");
    comboBox2.Items.Add("제주도");*/
    String[] initValue = new String[] { "서울특별시", "인천광역시",
    "대전광역시", "대구광역시", "부산광역시", "광주광역시", "울산광역시",
    "경기도", "충청도", "강원도", "경상도", "전라도", "제주도" };
    comboBox2.Items.AddRange(initValue);
    comboBox2.SelectedIndex = 0;
}

private void button1_Click(object sender, EventArgs e)
{
    String str = comboBox1.Text + textBox1.Text + "님은 " + comboBox2.Text + "에 거주하십니다.";
    MessageBox.Show(str);
}
```

- 비주얼프로그래밍 01강좌 과제 – 제출기한 (9월 30일 저녁 11시 59분)
- 실습 – 리스트박스를 활용하여 항목 추가/제거하기(캠핑 물품 정리)
- 과제 1 – 문자열 변환 프로그램
- 과제 2 – 기차표 정보 확인 프로그램

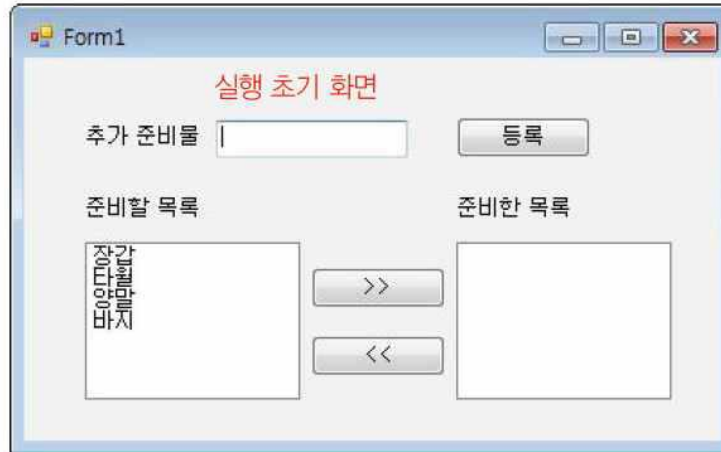
- 프로젝트 생성 시 - 프로젝트 명 => 학번_이름_과제1 ex) 2018115478_홍길동_과제1
- 1개의 프로젝트 안에 실습과 과제 모두 Form 추가하여 작성
- 작성 후 프로그램 확인 시 Form 변경 방법
 - 솔루션탐색기에서 'program.cs' 더블클릭
 - 코드 창 편집기에서 'Application.Run(new 폼이름());
 - 폼 이름 부분을 실행하고자 하는 폼 이름으로 변경하여 실행
- 이벤트 추가 시 반드시 디자인 부분에서 더블 클릭하여 추가 할 것
- 더블 클릭하여 추가한 이벤트 코드 부분을 지우면 에러가 발생할 수 있으니, 이벤트를 잘못 생성하여도 절대 지우지 마세요

- 과제 완료 후 제출 방법
 - 솔루션 탐색기에서 솔루션 클릭 후 오른쪽 마우스 클릭
 - 나타나는 메뉴에서 '파일 탐색기에서 폴더 열기' 클릭
 - 탐색기가 열리면 상위 폴더로 두 번 이동 한 후 폴더 전체를 압축
 - 압축한 파일을 이클래스에 제출하세요.
 - Ex) 2018115478_홍길동_과제1.zip

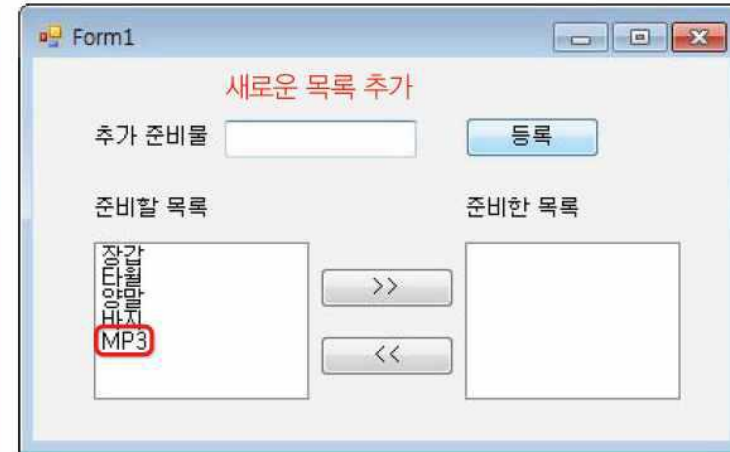


- 실습 - 리스트박스를 활용하여 항목 추가/제거하기(캠핑 물품 정리)

- 실행 결과



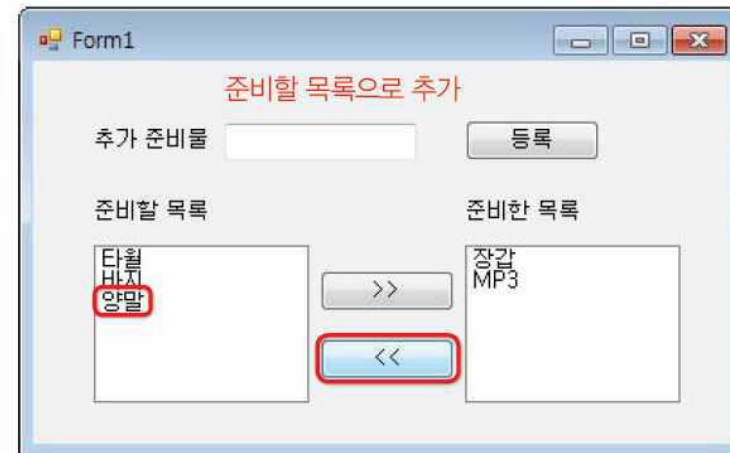
Form1 실행 초기 화면. 상단 제목은 '실행 초기 화면'. '추가 준비물' 텍스트 옆에 빈 입력 필드와 '등록' 버튼이 있다. '준비할 목록'과 '준비한 목록' 두 리스트박스가 있다. '준비할 목록'에는 '장갑', '양말', '바지'가 수직으로 나열되어 있다. 두 리스트박스 사이에는 '>>'와 '<<' 버튼이 있다.



Form1 새로운 목록 추가. 상단 제목은 '새로운 목록 추가'. '추가 준비물' 텍스트 옆에 빈 입력 필드와 '등록' 버튼이 있다. '준비할 목록'과 '준비한 목록' 두 리스트박스가 있다. '준비할 목록'에는 '장갑', '양말', '바지', 'MP3'가 수직으로 나열되어 있다. 'MP3' 항목이 빨간색 사각형으로 강조되어 있다. 두 리스트박스 사이에는 '>>'와 '<<' 버튼이 있다.

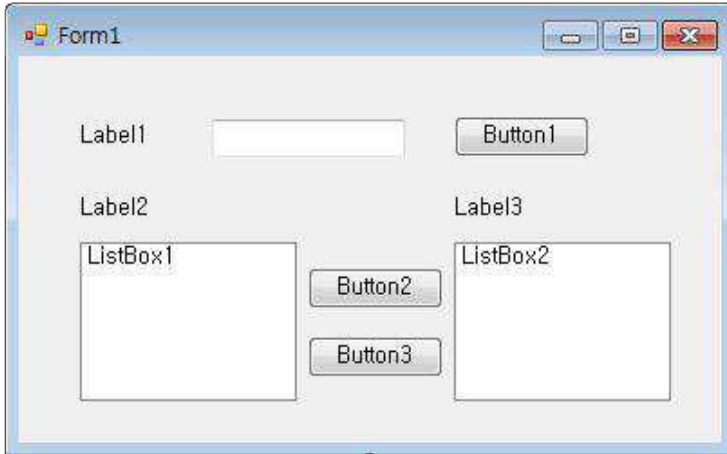


Form1 준비한 목록으로 이동. 상단 제목은 '준비한 목록으로 이동'. '추가 준비물' 텍스트 옆에 빈 입력 필드와 '등록' 버튼이 있다. '준비할 목록'과 '준비한 목록' 두 리스트박스가 있다. '준비할 목록'에는 '타월', '바지'가 수직으로 나열되어 있다. '준비한 목록'에는 '장갑', '양말', 'MP3'가 수직으로 나열되어 있다. '장갑' 항목이 빨간색 사각형으로 강조되어 있다. 두 리스트박스 사이에는 '>>'와 '<<' 버튼이 있다. '>>' 버튼이 빨간색 사각형으로 강조되어 있다.



Form1 준비할 목록으로 추가. 상단 제목은 '준비할 목록으로 추가'. '추가 준비물' 텍스트 옆에 빈 입력 필드와 '등록' 버튼이 있다. '준비할 목록'과 '준비한 목록' 두 리스트박스가 있다. '준비할 목록'에는 '타월', '바지', '양말'이 수직으로 나열되어 있다. '양말' 항목이 빨간색 사각형으로 강조되어 있다. '준비한 목록'에는 '장갑', 'MP3'가 수직으로 나열되어 있다. 두 리스트박스 사이에는 '>>'와 '<<' 버튼이 있다. '<<' 버튼이 빨간색 사각형으로 강조되어 있다.

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값	컨트롤	속성	속성값
Label1	Text	추가 준비물	ListBox2	SelectionMode	MultiExtended
Label2	Text	준비할 목록	Button1	Text	등록
Label3	Text	준비한 목록	Button2	Text	»
TextBox1	Text	(빈칸)	Button3	Text	«
ListBox1	SelectionMode	One			

3) 코드 작성

```
private void listBoxEx1_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("장갑");
    listBox1.Items.Add("타월");
    listBox1.Items.Add("양말");
    listBox1.Items.Add("바지");
}

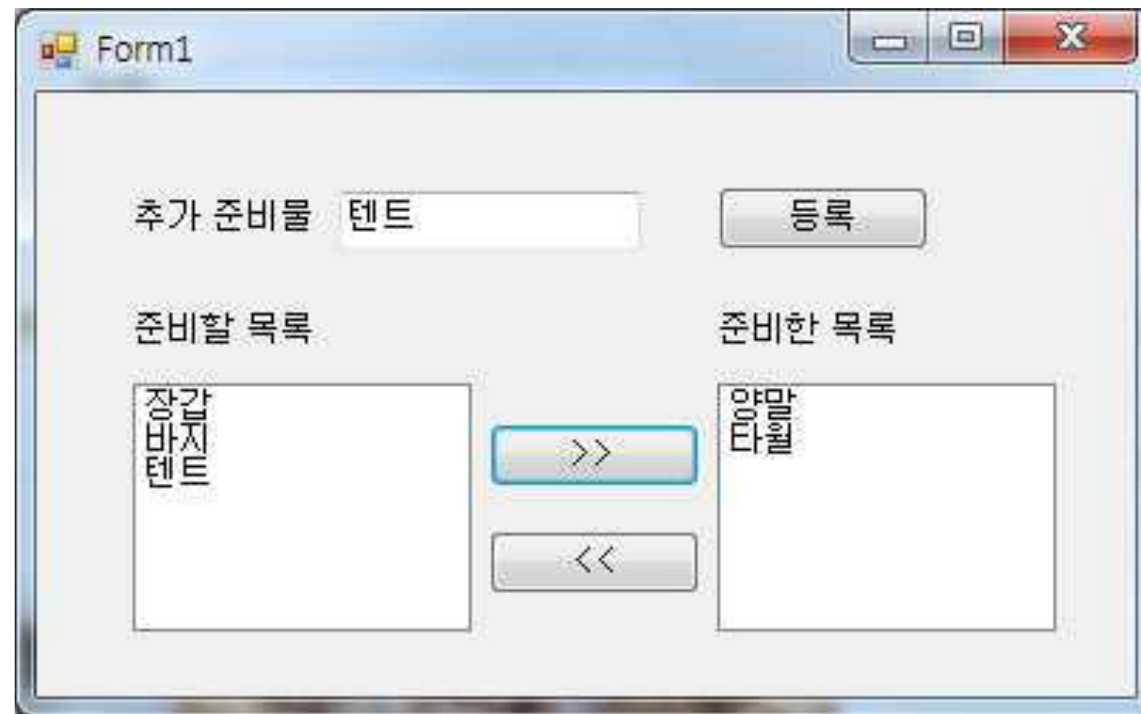
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(textBox1.Text);
    textBox1.Text = "";
    textBox1.Focus();
}

private void button2_Click(object sender, EventArgs e)
{
    listBox2.Items.Add(listBox1.Text);
    listBox1.Items.Remove(listBox1.Text);
}

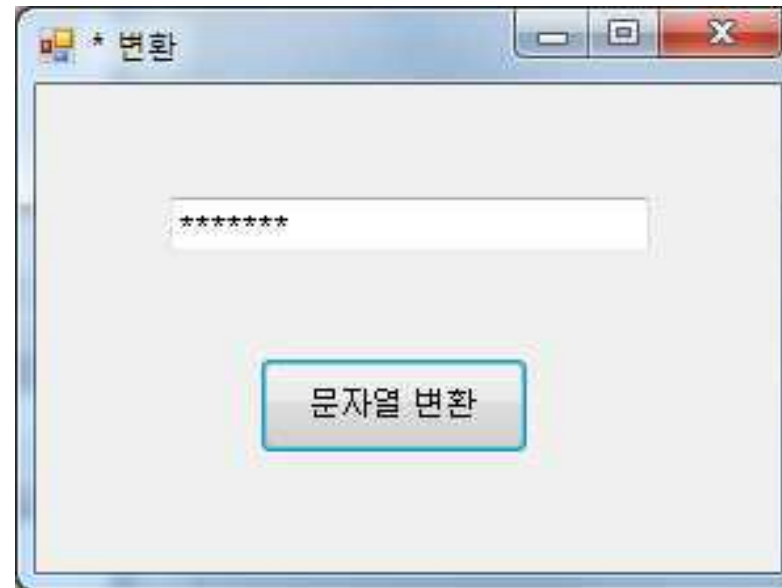
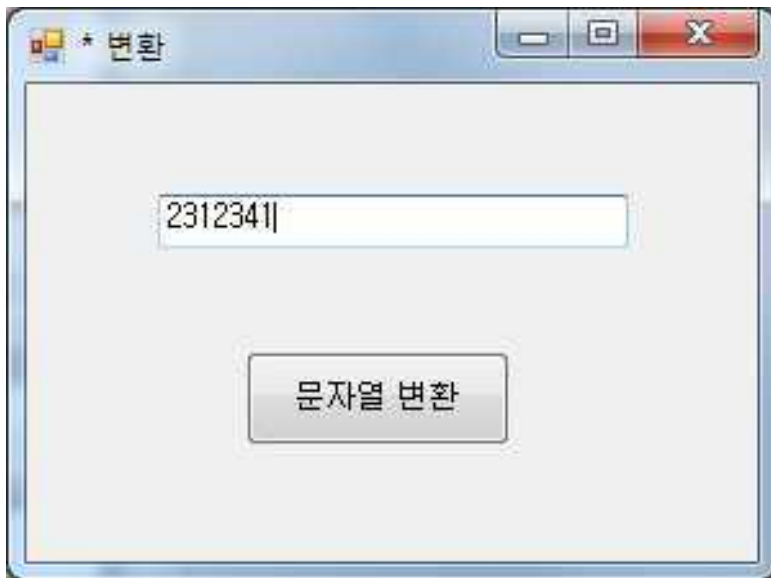
private void button3_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(listBox2.Text);
    listBox2.Items.Remove(listBox2.Text);
}
```

4) 실행 결과 확인

- 준비물을 새로 추가할 수도 있고 준비할 목록에서 준비한 목록으로, 준비한 목록에서 준비할 목록으로 옮길 수도 있다.



- 과제 1 – 문자열 변환 프로그램
 - 텍스트박스 컨트롤에 입력된 문자열을 *로 바꾸는 프로그램을 작성하시오.



- 과제 2 – 기차표 정보 확인 프로그램
 - 사용자가 선택한 기차표의 정보를 알려주는 프로그램을 리스트박스 컨트롤과 콤보박스 컨트롤을 사용하여 작성하시오.

