

본 강의에서 수업자료로 이용되는 저작물은
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2024. 8. 30.

부천대학교·한국복제전송저작권협회

C#

3주차 3차시

8장 컨트롤(3)

3주차 학습 내용

- * 1차시
 - * 2주차 3교시 과제 피드백
 - * 4장 클래스
- * 2차시
 - * 7장 폼
 - * 원폼 어플리케이션의 구성
 - * 폼 클래스
 - * 폼 클래스의 계층도
 - * 폼 클래스의 프로퍼티, 메소드, 이벤트
 - * 컨트롤 클래스
 - * 컨트롤 클래스의 계층도
 - * 컨트롤 클래스의 프로퍼티, 메소드, 이벤트
- * 3차시
 - * 8장 컨트롤(3)
 - * 리스트
 - 리스트 상자
 - 콤보 상자
 - 체크 리스트 상자

[클래스 멤버의 종류]

-자료 멤버

필드, 상수, 이벤트

-함수멤버

메서드, 생성자, 소멸자, **프로퍼티 (속성)**,
인덱서(색인자), 연산자 중복

-내포형 멤버

클래스형, 구조형, 열거형, 인터페이스형,
대리(델리게이트)형

[객체 지향 프로그램 구성]

-클래스 정의

-객체 선언

-객체 생성

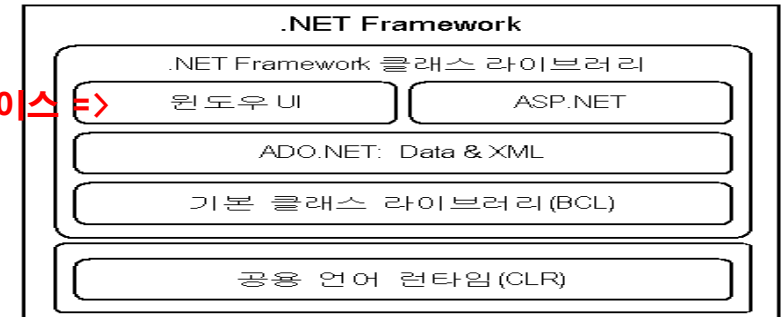
-객체의 멤버 접근 및 활용

7장 폼

System.Windows.Forms 네임스페이스 =>

* 7장 폼

- * 원폼 애플리케이션 작성 순서(2주차 2교시 참고)
- * 원폼 어플리케이션의 구성
- * 폼 클래스
 - * 폼 클래스의 계층도
 - * 폼 클래스의 프로퍼티, 메소드, 이벤트
- * 컨트롤 클래스
 - * 컨트롤 클래스의 계층도
 - * 컨트롤 클래스의 프로퍼티, 메소드, 이벤트



[클래스 멤버의 종류]

-자료 멤버

필드, 상수, 이벤트

-함수멤버

메서드, 생성자, 소멸자, 프로퍼티 (속성), 인덱서(색인자), 연산자 중복

-내포형 멤버

클래스형, 구조형, 열거형, 인터페이스형, 대리(델리게이트)형

[객체 지향 프로그램 구성]

-클래스 정의

-객체 선언

-객체 생성

-객체의 멤버 접근 및 활용

C# 프로그래밍



8. 컨트롤 (3)



목차

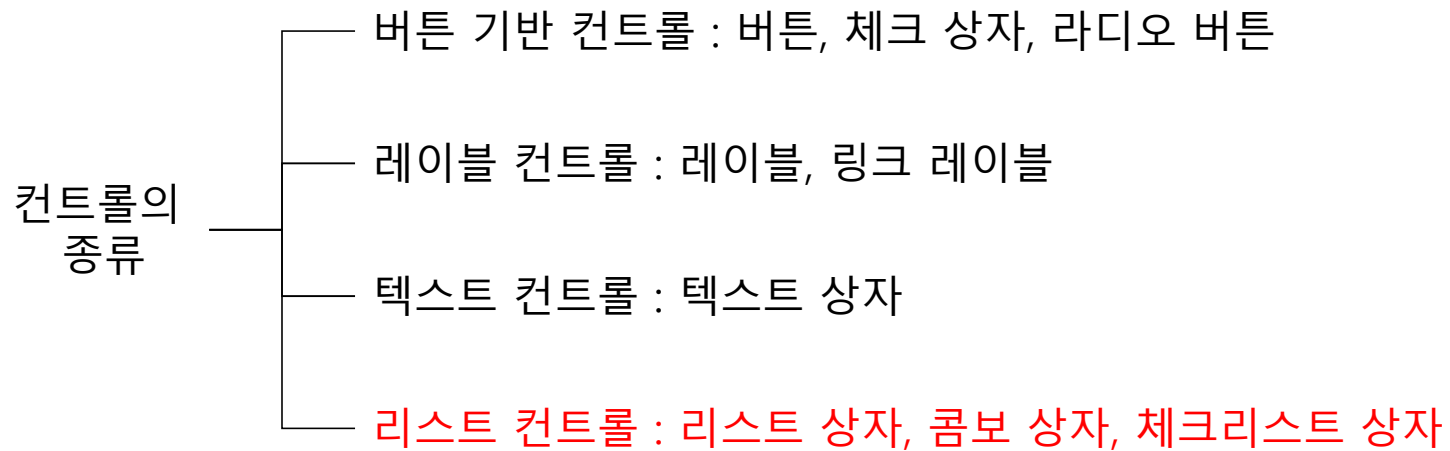
- 버튼 기반 컨트롤
- 레이블과 링크 레이블
- 텍스트 상자
- 리스트



컨트롤이란?

■ 컨트롤

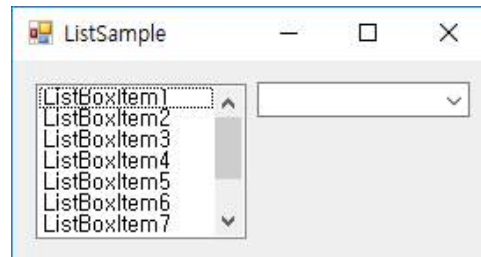
- 화면에 표시되어 사용자와 상호작용을 수행하는 컴포넌트를 의미.
- 종류 : 버튼, 레이블, 텍스트, 리스트





리스트

- 리스트 기반 컨트롤
 - ListControl 클래스를 상속받은 컨트롤을 가리킴.
- 리스트 기반 컨트롤의 종류
 - 리스트 상자
 - 콤보 상자
 - 체크리스트 상자
- 리스트 기반 컨트롤의 기본적인 형태





리스트 상자 [1/5]

■ 리스트 상자

- 사용자가 선택할 수 있는 항목들의 목록을 표시해 주는 컨트롤

■ 리스트 상자의 **프로퍼티**

프로퍼티	설명
Items	리스트 상자의 항목을 설정.
MultiColumn	리스트 상자가 여러 열을 표시할 수 있도록 설정.
ScrollAlwatsVisible	항상 세로 스크롤바가 표시되도록 설정.
SelectionMode	리스트 상자에서 항목을 선택하는 방법을 설정.
SelectedIndex	리스트 상자에서 현재 선택된 아이템이 인덱스를 반환(0부터 시작).
SelectedItem	리스트 상자에서 현재 선택된 아이템을 반환.



리스트 상자 [2/5]

- 리스트 상자의 항목을 선택하는 방법
 - SelectionMode 프로퍼티 값에 따라 변경
 - SelectionMode 열거형

기호상수	설명
None	항목을 선택할 수 없음.
One	하나의 항목만 선택할 수 있음.
MultiSimple	여러 항목을 선택할 수 있음.
MultiExtended	여러 항목을 선택할 수 있으며, <Shift>, <Ctrl>, 마우스 포인터를 이용해 선택할 수 있음.

- 리스트 상자의 항목
 - **Items** 프로퍼티를 통하여 **추가 및 삭제 가능.**
 - 문자열 컬렉션 편집기
 - ObjectCollection 클래스 메소드



리스트 상자 [3/5]

■ ObjectCollection 클래스의 메소드

메소드	설명
Add(object item)	리스트 상자에 항목을 추가한다.
Clear()	리스트 상자의 모든 항목을 제거한다.
FindString(string s, int index)	리스트 상자의 항목 중 지정된 인덱스 다음부터 지정된 문자열로 시작하는 항목의 인덱스를 반환한다.
IndexOf(object item)	지정한 항목의 인덱스를 반환한다.
Insert(int index, object item)	지정된 인덱스에 항목을 추가한다.
Remove(object item)	지정된 항목을 제거한다.
RemoveAt(int index)	지정된 인덱스의 항목을 제거한다.



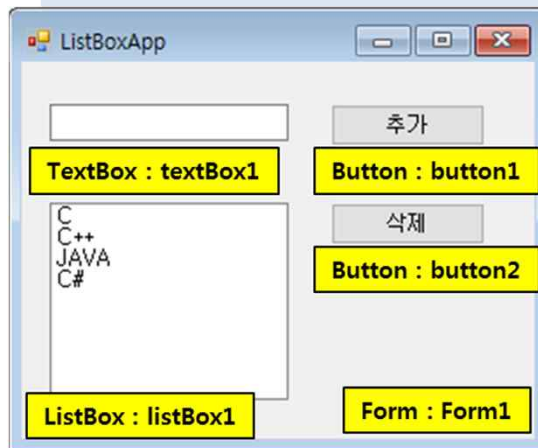
리스트 상자 [4/5]

■ 예제 8.7 [w3_ex8_7_xxxx]

- 텍스트 상자를 통해 입력받은 텍스트를 ObjectCollection 클래스의 메소드를 사용하여 리스트 상자에 추가하고 리스트 상자에서 선택된 항목을 삭제하는 예제.

예제 8.7 [w3_ex8_7_xxxx]

1) 디자인



컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	ListBoxApp
TextBox : textBox1	Text	
ListBox : listBox1	Items	C C++ JAVA C#
Button : button1	Text	추가
Button : button2	Text	삭제

컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	Button1_Click()
Button : button2	Click	Button2_Click()



리스트 상자 [5/5]

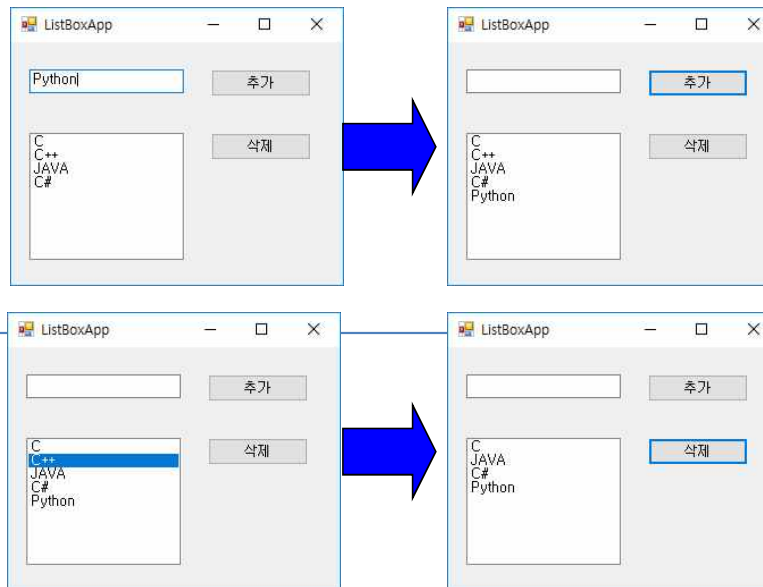
2) 코드

```
private void button1_Click(object sender, EventArgs e) {  
    if (textBox1.Text != " ") {  
        listBox1.Items.Add(textBox1.Text); // listBox1.Items.Insert(0, textBox1.Text);  
        textBox1.Text = ""; } }  
private void button2_Click(object sender, EventArgs e) {  
    if (listBox1.SelectedIndex > -1)  
        listBox1.Items.Remove(listBox1.SelectedItem); // listBox1.Items.RemoveAt(listBox1.SelectedIndex); }
```

실행 방법 : ① 텍스트 상자에 문자열을 입력한 후, 추가 버튼을 클릭한다.

② 리스트 상자의 항목을 선택한 후, 삭제 버튼을 클릭한다.

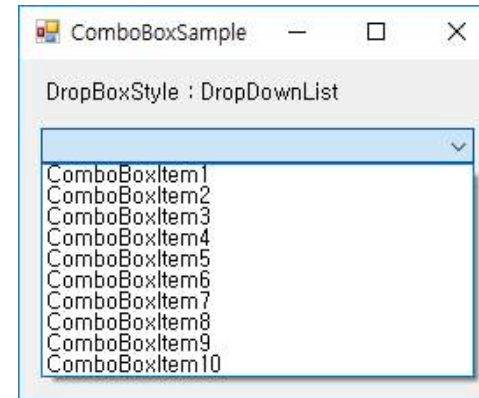
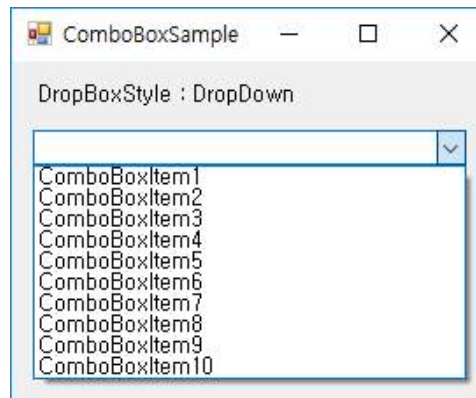
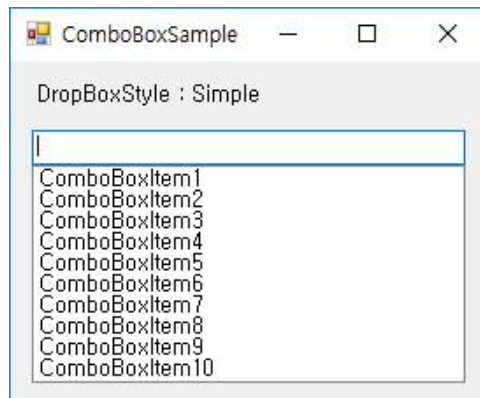
실행 결과 :





콤보 상자 [1/4]

- 콤보 상자
 - 사용자가 상자를 클릭하면 목록이 나타나는 드롭다운(drop-down) 형식의 컨트롤
- 콤보 상자의 기본적인 형태





콤보 상자 [2/4]

■ DropDownStyle

- 콤보 상자의 형태를 결정.
- ComboBoxStyle을 열거형으로 가짐.
- ComboBoxStyle 열거형

기호상수	설명
Simple	[그림 8.11]의 (1)과 같이 선택 항목을 항상 볼 수 있음.
DropDown	[그림 8.11]의 (2)와 같이 화살표 버튼을 클릭해야 선택항목을 볼 수 있음.
DropDownList	[그림 8.11]의 (3)과 같이 화살표 버튼뿐만 아니라 텍스트 부분을 클릭하여도 선택 항목을 볼 수 있음.

■ 콤보 상자의 항목

- Items 프로퍼티를 통하여 추가 및 삭제 가능.
 - 문자열 컬렉션 편집기
 - ObjectCollect 클래스 메소드



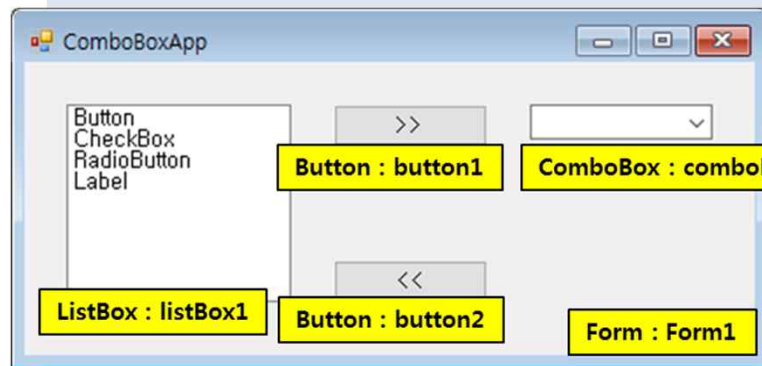
콤보 상자 [3/4]

■ 예제 8.8 [w3_ex8_8_xxxx]

- 리스트 상자와 콤보 상자를 폼에 배치한 후 두 개의 버튼을 이용하여 리스트 상자와 콤보 상자 사이에 항목을 주고 받을 수 있는 예제.

예제 8.8 [w3_ex8_8_xxxx]

1) 디자인



컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	ComboBoxApp
ListBox : listBox1	Items	Button CheckBox RadioButton Label
ComboBox : comboBox1	Items	LinkLabel TextBox ListBox ComboBox
Button : button1	Text	>>
Button : button2	Text	<<

컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	Button1_Click()
Button : button2	Click	Button2_Click()



콤보 상자 [4/4]

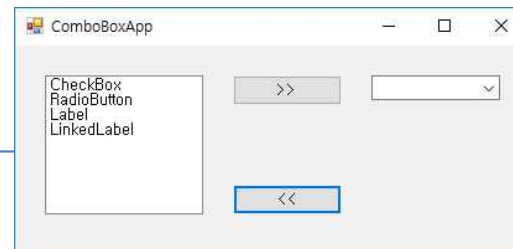
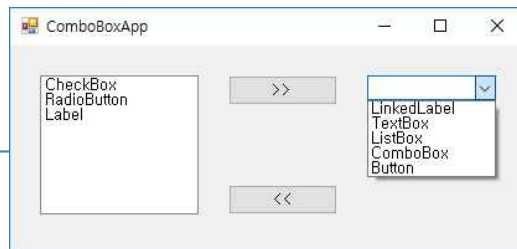
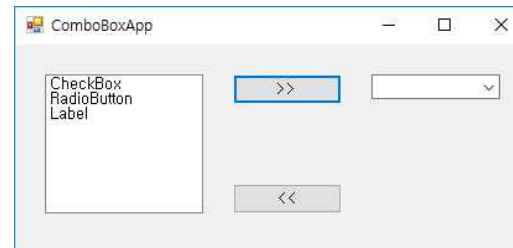
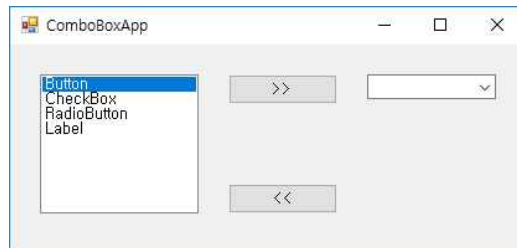
2) 코드

```
private void button1_Click(object sender, EventArgs e) {  
    if (listBox1.SelectedItem != null) {  
        comboBox1.Items.Add(listBox1.SelectedItem); // comboBox1.Items.Insert(0, listBox1.SelectedItem);  
        listBox1.Items.Remove(listBox1.SelectedItem); // listBox1.Items.RemoveAt(listBox1.SelectedIndex); } }  
  
private void button2_Click(object sender, EventArgs e) {  
    if (comboBox1.SelectedItem != null) {  
        listBox1.Items.Add(comboBox1.SelectedItem);  
        comboBox1.Items.Remove(comboBox1.SelectedItem); } }
```

실행 방법 : ① 리스트 상자의 항목을 선택한 후, >> 버튼을 클릭한다.

② 콤보 상자의 항목을 선택한 후, << 버튼을 클릭한다.

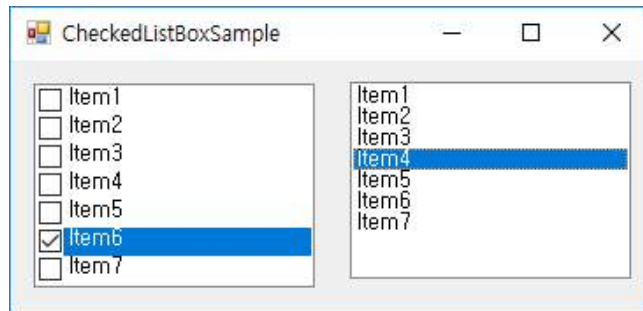
실행 결과 :





체크리스트 상자 [1/3]

- 체크 리스트 상자
 - 리스트 상자의 항목에 체크 상자를 추가한 형태의 컨트롤
- 체크 리스트 상자의 기본적인 형태



- 체크 리스트 상자의 프로퍼티
 - 리스트 상자과 대부분 동일
 - CheckOnClick 프로퍼티가 추가됨.
 - 항목을 클릭했을때 체크 상자에 '✓' 표시가 나타나도록 설정.



퀴즈 8.1

■ 퀴즈 8.1 [w3_quiz8_1_xxxx]

- 레이블과 라디오, 체크박스를 이용하여 글꼴 프로그램을 작성하시오
- 체크 박스를 취소하였을 때 레이블에 나타나는 문자열에도 효과가 취소가 되도록 작성하시오.

퀴즈 8.1 [w2_quiz8_2_xxxx]

1) 디자인

정렬	스타일	크기
<input checked="" type="radio"/> 왼쪽	<input type="checkbox"/> 굵게	<input type="radio"/> 12
<input type="radio"/> 가운데	<input type="checkbox"/> 밑줄	<input type="radio"/> 16
<input type="radio"/> 오른쪽	<input type="checkbox"/> 이탤릭	<input checked="" type="radio"/> 20
	<input type="checkbox"/> 취소선	

저의 모양을 바꾸세요!!



퀴즈 8.1

2) 코드

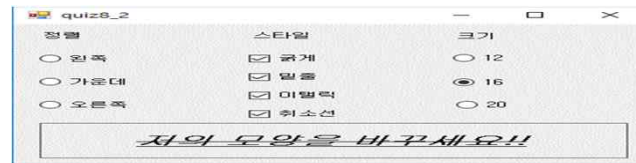
```
private void radioButton1_CheckedChanged(object sender, System.EventArgs e)
{
    label3.TextAlign = ContentAlignment.MiddleLeft;
}

private void checkBox1_CheckedChanged(object sender, System.EventArgs e)
{
    if(checkBox1.Checked == true)
    {
        Font font = new Font(label3.Font, label3.Font.Style | FontStyle.Bold);
        label3.Font = font;
    }
    else
    {
        Font font = new Font(label3.Font, label3.Font.Style ^ FontStyle.Bold);
        label3.Font = font;
    }
}

private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    Font font = new Font(label3.Font.FontFamily, 12, label3.Font.Style );
    label3.Font = font;
}
```

실행 방법 : 레이블과 라디오, 체크박스를 이용하여 레이블의 문자의 글꼴을 변경하시오.

실행 결과 :



3주차 학습 내용 정리

- * 1차시
 - * 2주차 3교시 과제 피드백
 - * 4장 클래스
- * 2차시
 - * 7장 폼
 - * 원폼 어플리케이션의 구성
 - * 폼 클래스
 - * 폼 클래스의 계층도
 - * 폼 클래스의 프로퍼티, 메소드, 이벤트
 - * 컨트롤 클래스
 - * 컨트롤 클래스의 계층도
 - * 컨트롤 클래스의 프로퍼티, 메소드, 이벤트
- * 3차시
 - * 8장 컨트롤(3)
 - * 리스트
 - 리스트 상자
 - 콤보 상자
 - 체크 리스트 상자

[클래스 멤버의 종류]

-자료 멤버

필드, 상수, 이벤트

-함수멤버

메서드, 생성자, 소멸자, **프로퍼티 (속성)**,
인덱서(색인자), 연산자 중복

-내포형 멤버

클래스형, 구조형, 열거형, 인터페이스형,
대리(델리게이트)형

[객체 지향 프로그램 구성]

-클래스 정의

-객체 선언

-객체 생성

-객체의 멤버 접근 및 활용