

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#

2주차 2교시

8장 컨트롤(1)

# 2주차 학습 내용

- \* 1교시

- \* 2장 C# 언어 구조 2

- \* 자료형(값형:열거형, 참조형: 배열)
    - \* 형 변환
    - \* 박싱과 언박싱

- \* 3장 문장 종류

- \* 제어문

- \* 2교시

- \* 8장 컨트롤

- \* 원폼 애플리케이션 작성 순서(7장 참조)
    - \* 버튼 기반 컨트롤(버튼, 라디오 버튼)

- \* 3교시

- \* 8장 컨트롤

- \* 버튼 기반 컨트롤(체크 상자)
    - \* 레이블과 링크 레이블
    - \* 텍스트 상자



## 목차

- 원폼 애플리케이션 작성 순서(7장 참조)
- 버튼 기반 컨트롤
- 레이블과 링크 레이블
- 텍스트 상자
- 리스트

# C# 프로그래밍



원폼 애플리케이션 작성 순서(7장 참조)



# 1. 윈도우 프로그래밍

## ■ 윈도우즈용 프로그래밍을 작성하는 방법

### ■ Win32 응용 프로그램

- 운영체제가 제공하는 Win32API를 사용하는 방법으로 섬세하고 강력하지만 일일이 손으로 코드를 작성해야 하므로 생산성이 떨어짐
- Win32의 원래 의미는 "Microsoft 32bit Windows API" 혹은 "Win32 API"라고 하고 Win32 API를 이용해 만든 프로그램을 Win32 응용 프로그램이라 함.

### ■ MFC라이브러리 활용

- Win32보다 좀 더 객체 지향적이며 편리함.

### ■ 비주얼 베이직이나 델파이 같은 비주얼 툴

### ■ 원폼 애플리케이션

#### ■ CLR의 Windows 응용 프로그램

- 원폼은 C#이라는 강력하고도 깔끔한 언어와 BCL이라는 방대한 라이브러리, 그리고 비주얼 스튜디오라는 편리한 개발 환경을 활용하여 멋지고도 안정적인 윈도우즈용 응용프로그램을 신속하게 작성함.

### ■ WPF(Windows Presentation Foundation) 고급 라이브러리

- 시각적으로 뛰어난 차세대 프레젠테이션 시스템



## 2. 윈폼 애플리케이션

### ■ 윈폼 애플리케이션

- .NET 프레임워크에서 제공하는 클래스 라이브러리로 만들어진 윈도우 GUI응용프로그램
- .NET 프레임워크를 지원하는 통합 개발 환경(Visual Studio 2019)에서 c#언어로 윈폼 응용 프로그램을 작성한다.
- 컨트롤을 사용하여 프로그래머가 원하는 화면을 구성하고 이벤트가 발생했을 때 처리하고자 하는 작업을 이벤트 처리기에 기술하는 방식으로 프로그래밍된 프로그램.
- 윈도우 폼에 컨트롤 또는 컴포넌트를 배치하고 이벤트 처리기를 등록하여 사용자의 다양한 요구를 입력으로 받아 처리한 후 실행 결과를 응답해 주는 방식으로 작동.



### 3. 원폼 애플리케이션 만들기

- 통합 개발 환경 시작하기
- 원폼 응용 프로그램 작성하기
  - 폼 설계
  - 컴포넌트
  - 프로퍼티
  - 멤버
  - 이벤트처리기
  - 응용 프로그램 실행
- 소스코드 보기

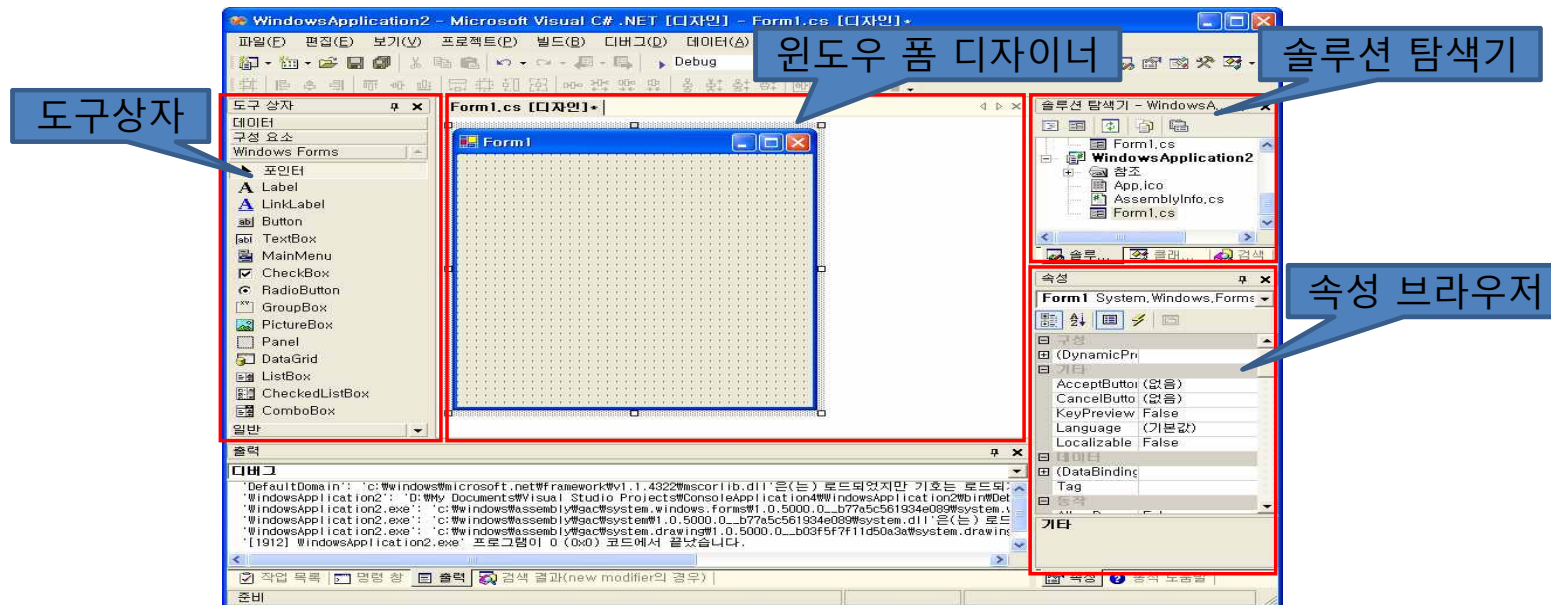




## 3.1 통합 개발환경 사용하기

### ■ 1. 새 프로젝트 시작

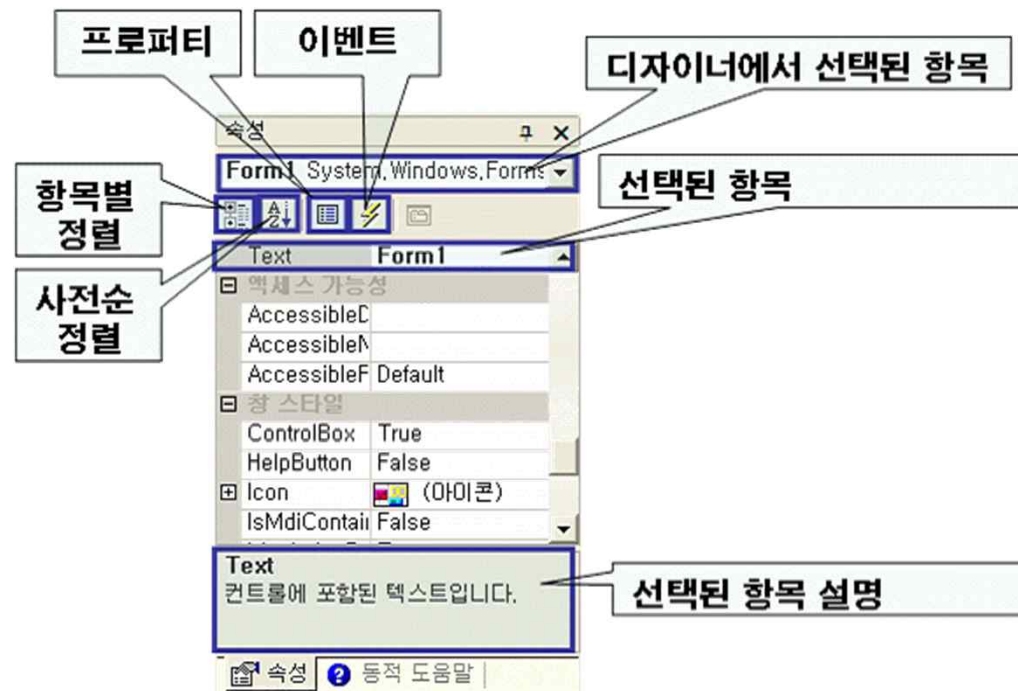
- **【파일(F)】⇒【새로 만들기(N)】⇒【프로젝트(P)】**를 선택
- **【Visual c#/Windows Desktop】⇒【Windows Forms 앱(.NET Framework)】**을 선택





## 3.1 통합 개발환경 사용하기

### ■ 속성 브라우저



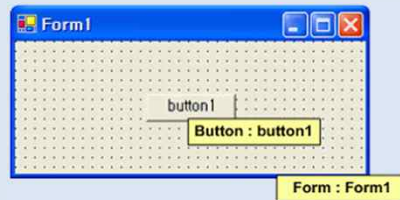


## 3.2 원폼 애플리케이션 작성하기

- 폼 설계
  - 컴포넌트
  - 프로퍼티
  - 멤버
- 
- 이벤트처리기
  - 응용 프로그램 실행

### 예제 7. 1 [w2\_ex7\_1\_xxxx]

#### 1) 폼 설계



#### 2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	DisplayStartDateApp
Button : button1	Text	Display

#### 3) 멤버

```
public partial class Form1 : Form
{
    public Form1( )
    {
        InitializeComponent();
        startDateTime = DateTime.Now;
    }
    DateTime startDateTime;
    public DateTime GetStartDateTime( )
    {
        return startDateTime;
    }
}
```



## 3.2 원폼 애플리케이션 작성하기

- 폼 설계
- 컴포넌트
- 프로퍼티
- 멤버
- 이벤트처리기
- 응용 프로그램 실행
- 컴파일과 실행

### 4) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	button1_Click()

```
public partial class Form1 : Form
{
    private void button1_Click(object sender, EventArgs e) {
        MessageBox.Show(GetStartDate().ToString());
    }
}
```

실행 방법 : 응용 프로그램을 실행한 후, 폼에 있는 "Display" 버튼을 클릭한다.  
실행 결과 :



- 컴파일하는 방법
  - 메인 메뉴를 통하여 **【빌드(B)】** ➞ **【솔루션 빌드(B)】**를 선택
- 컴파일한 후 바로 실행하는 방법
  - **디버그(D)】** ➞ **【시작(S)】**이나 **【디버그(D)】** ➞ **【디버깅하지 않고 시작(G)】**을 선택

# C# 프로그래밍



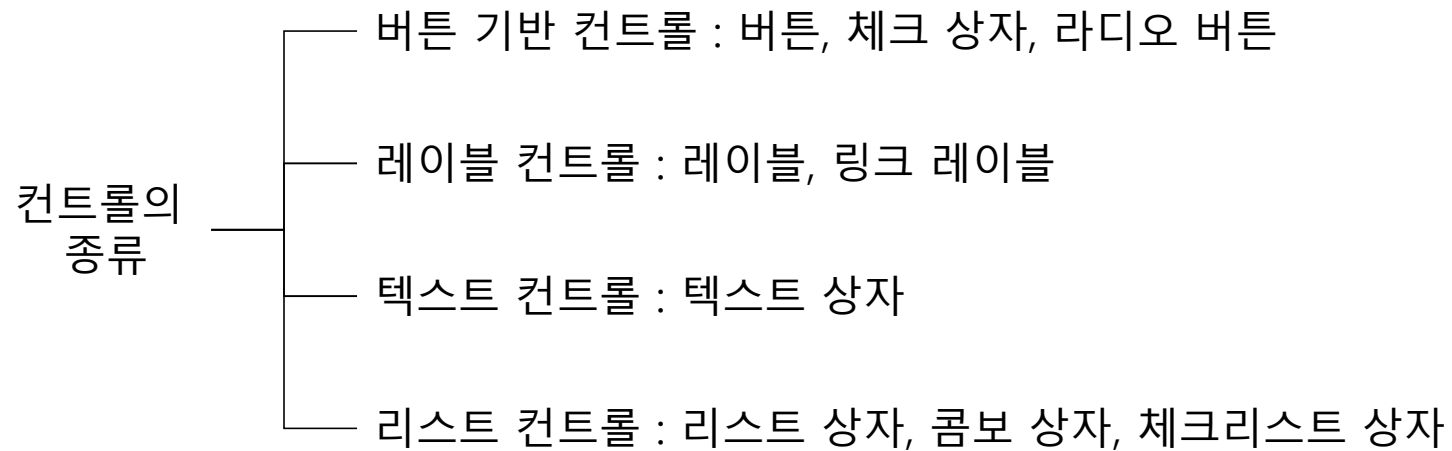
## 8. 컨트롤 (1)



## 컨트롤이란?

### ■ 컨트롤

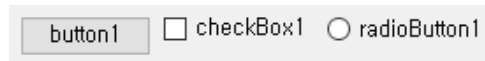
- 화면에 표시되어 사용자와 상호작용을 수행하는 컴포넌트를 의미.
- 종류 : 버튼, 레이블, 텍스트, 리스트





## 버튼 기반 컨트롤

- 버튼 기반 컨트롤
  - ButtonBase 추상 클래스를 상속받은 컨트롤을 의미.
- 버튼 기반 컨트롤의 종류
  - 버튼
  - 체크 상자
  - 라디오 버튼
- 버튼 기반 컨트롤의 기본적인 형태





## 버튼 [1/4]

### ■ 버튼

- 사용자 입력을 받는 가장 간단한 방법으로 버튼을 클릭함으로써 이벤트를 발생.

### ■ 버튼의 프로퍼티

- 이름과 글자색등을 설정할 수 있음.
- 버튼의 프로퍼티

프로퍼티	설명
Text	버튼 이름을 설정.
Font	버튼 이름에 해당하는 문자열의 글꼴을 설정.
ForeColor	버튼 이름에 해당하는 문자열의 색상을 설정.
TextAlign	버튼 이름에 해당하는 문자열을 정렬.
FlatStyle	버튼의 스타일(모양)을 설정.
Size	버튼의 크기(픽셀 단위)를 설정
BackColor	버튼의 배경색을 설정.
BackgroundImage	버튼의 배경 이미지를 설정.
Image	버튼 형태 위에 이미지를 설정.
ImageAlign	버튼 형태 위에 이미지를 정렬.

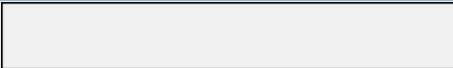
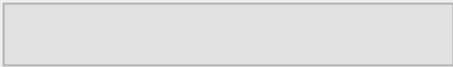
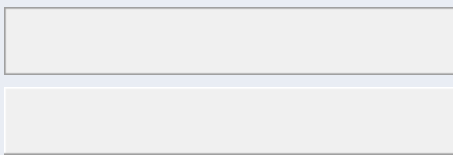
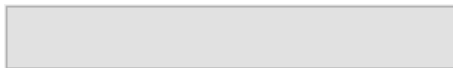




## 버튼 [2/4]

### ■ 버튼의 스타일

- 버튼의 스타일을 설정하기 위해서는 FlatStyle 프로퍼티를 지정해야 함.
- System.Windows.Forms 네임스페이스에 포함되어 있음.
- 4가지 종류가 있음.
  - FlatStyle 열거형

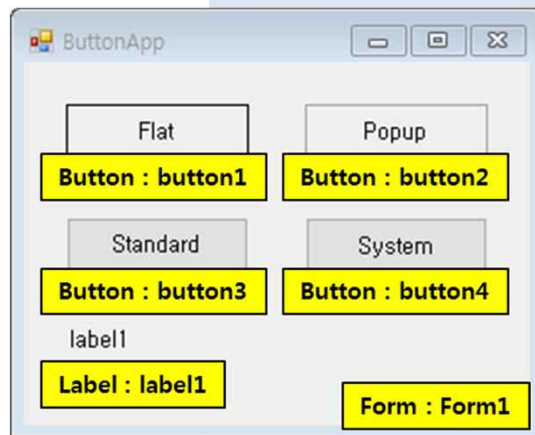
기호상수	버튼모양	설명
Flat		버튼의 모양을 평면으로 표시한다.
Standard		버튼의 모양을 입체적으로 표시한다.
Popup		버튼의 모양이 처음에는 평면으로 표시되었다가 마우스 포인터가 버튼 위로 이동하면 입체적으로 변경한다.
System		버튼의 모양을 시스템이 결정. 일반적으로 System으로 지정한 버튼의 모양은 Standard로 지정한 것과 같다.



## 버튼 [3/4]

### 예제 8.1 [w2\_ex8\_1\_xxxx]

#### 1) 디자인



컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	ButtonApp
Button : Button1	FlatStyle, Text	Flat
Button : Button2	FlatStyle, Text	Popup
Button : Button3	FlatStyle, Text	Standard
Button : Button4	FlatStyle, Text	System

컨트롤 : (Name)	이벤트	메소드명
Button : Button1	Click	Button1_Click()
Button : Button2	Click	Button2_Click()
Button : Button3	Click	Button3_Click()
Button : Button4	Click	Button4_Click()



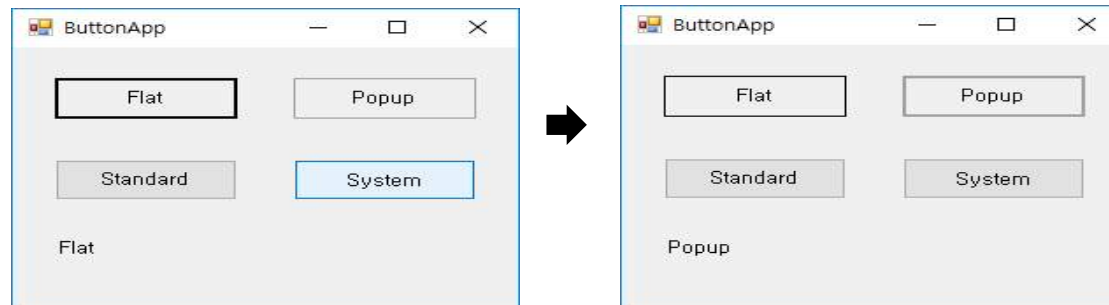
## 버튼 [4/4]

2) 코드

```
private void button1_Click(object sender, EventArgs e) {  
    label1.Text = FlatStyle.Flat.ToString();  
}  
private void button2_Click(object sender, EventArgs e) {  
    label1.Text = FlatStyle.Popup.ToString();  
}  
private void button3_Click(object sender, EventArgs e) {  
    label1.Text = FlatStyle.Standard.ToString();  
}  
private void button4_Click(object sender, EventArgs e) {  
    label1.Text = FlatStyle.System.ToString();  
}
```

실행 방법 : 각각의 버튼을 클릭한다.

실행 결과 :



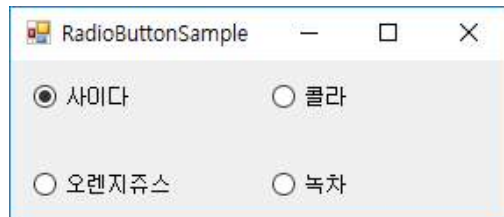


## 라디오 버튼 [1/3]

### ■ 라디오 버튼

- 주어진 항목들 중에서 오직 한 개만을 선택할 수 있는 컨트롤

### ■ 라디오 버튼의 기본적인 형태



### ■ 라디오 버튼의 프로퍼티

- 체크 상자와 동일

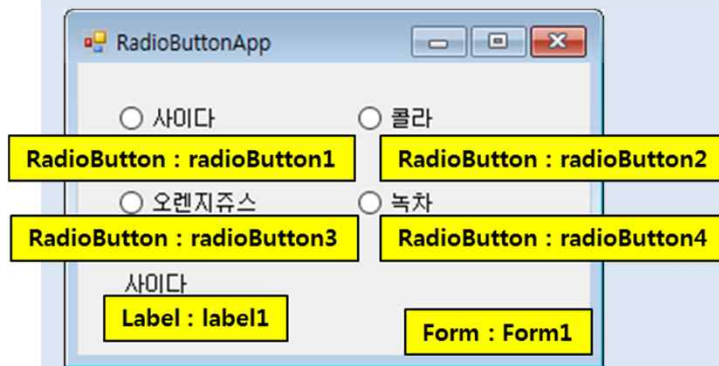


## 라디오 버튼 [2/3]

- 예제 8.3 [w2\_ex8\_3\_xxxx]
- 예제 8.2의 체크 상자를 라디오 버튼으로 대체한 예제

예제 8.3 [w2\_ex8\_3\_xxxx]

1) 디자인



컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	RadioButtonApp
RadioButton : radioButton1	Text	사이다
	Checked	True
RadioButton : radioButton2	Text	콜라
	Checked	False
RadioButton : radioButton3	Text	오렌지쥬스
	Checked	False
RadioButton : radioButton4	Text	녹차
	Checked	False
Label : label1	Text	사이다



## 라디오 버튼 [3/3]

컨트롤 : (Name)	이벤트	메소드명
RadioButton : radioButton1	CheckedChange	radioButton1_CheckedChanged()
RadioButton : radioButton2	CheckedChange	radioButton2_CheckedChanged()
RadioButton : radioButton3	CheckedChange	radioButton3_CheckedChanged()
RadioButton : radioButton4	CheckedChange	radioButton4_CheckedChanged()

### 2) 코드

```
private void radioButton1_CheckedChanged(object sender, EventArgs e) {  
    label1.Text = radioButton1.Text; }  
private void radioButton2_CheckedChanged(object sender, EventArgs e) {  
    label1.Text = radioButton2.Text; }  
private void radioButton3_CheckedChanged(object sender, EventArgs e) {  
    label1.Text = radioButton3.Text; }  
private void radioButton4_CheckedChanged(object sender, EventArgs e) {  
    label1.Text = radioButton4.Text; }
```

실행 방법 : 라디오 버튼을 클릭하여 선택

실행 결과 :

