



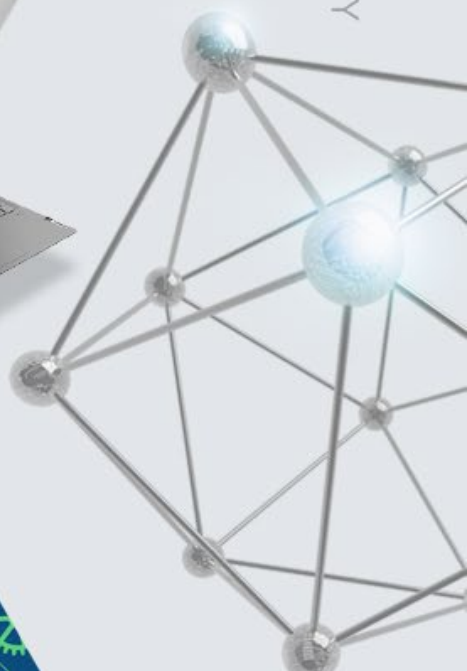
한국기술교육대학교  
온라인평생교육원

The 4th Industrial Revolution is characterized by super connectivity and super intelligence, where various products and services are connected to the network, and artificial intelligence and information communication technologies are used in 3D printing, unmanned transportation, robotics, Of the world's most advanced technologies.

T  
E  
C  
H  
N  
O  
L  
O  
G  
Y

# C# 프로그래밍

품



The 4th Industrial Revolution is characterized by super connectivity and super intelligence, where various products and services are connected to the network, and artificial intelligence and information communication technologies are used in 3D printing, unmanned transportation, robotics, Of the world's most advanced technologies.



### 폼



#### 학/습/목/표

1. 윈폼 애플리케이션을 생성하고 작성할 수 있다.
2. 폼 클래스의 계층도, 프로퍼티, 메소드 및 이벤트를 이해하고 설명할 수 있다.
3. 컨트롤 클래스의 계층도, 프로퍼티, 메소드 및 이벤트를 이해하고 설명할 수 있다.



#### 학/습/내/용

1. 윈폼 애플리케이션
2. 폼 클래스
3. 컨트롤 클래스



## 1. 윈폼 애플리케이션

### 1) 윈폼 애플리케이션의 개요

#### (1) 정의

##### ■ 윈폼 애플리케이션

- 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 작동됨

## 1. 윈폼 애플리케이션

### 2) 윈도우 폼

#### (1) 정의

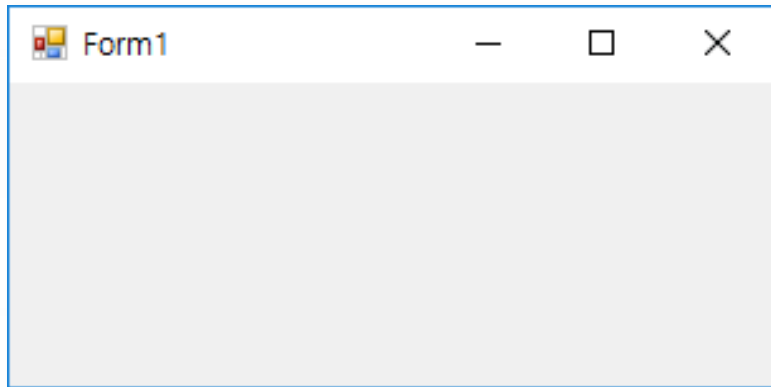
- 윈도우 폼 개요

- 단순히 폼이라고 부름

- 운영체제에서 제공하는 기본적인 화면 단위인 창을 말하는 개념

- 사각형 모양의 작은 화면 영역을 의미함

- 사용자에게 정보를 제공하고 사용자가 입력하는 자료를 받음





## 1. 윈폼 애플리케이션

### 3) 윈폼 애플리케이션의 핵심 클래스

#### (1) 폼 클래스

- 폼 클래스

- 윈도우 폼을 나타내는 클래스

- System.Windows.Forms 네임스페이스 속함

#### (2) 컴포넌트 클래스

- 컴포넌트 클래스

- 컨트롤 클래스의 베이스 클래스이며 화면에 직접적으로 나타나지 않으나 개념적인 부분을 나타내는 클래스

#### (3) 컨트롤 클래스

- 컨트롤 클래스

- 폼에 직접 표시되는 컨트롤을 위한 클래스



# 1. 원폼 애플리케이션

## 4) 원폼 애플리케이션 생성하기

### (1) 파일 구조

- 생성된 프로젝트 파일

파일명	설명
Program.cs	원폼 응용 프로그램의 시작점을 포함하는 C# 소스 파일
Form1.cs	원폼 응용 프로그램의 C# 소스파일
Form1.Designer.cs	원폼 응용 프로그램 폼 디자인 정의를 포함하는 C# 소스파일
WindowsApplication1.csproj	원폼 응용프로그램 프로젝트 파일
Properties/AssemblyInfo.cs	프로젝트가 생성하는 어셈블리를 설명하고 버전 관리 정보를 지정하는 데 사용하며, 애트리뷰트 정의를 포함하는 C# 소스파일
Properties/Resources.Designer.cs	원폼 응용 프로그램의 자원에 대한 C# 정의를 포함하는 C# 소스파일
Properties/Resources.resx	원폼 응용 프로그램의 자원 파일
Properties/Settings.Designer.cs	프로젝트 설정에 대한 C# 정의를 포함하는 C# 소스 파일
Properties/Settings.settings	프로젝트에 대한 설정 파일



## 1. 원폼 애플리케이션

### 4) 원폼 애플리케이션 생성하기

#### (2) 주요 프로젝트 파일

##### ▪ 주요 프로젝트 파일

파일명	설명
Program.cs	원폼 응용 프로그램의 시작점을 포함하는 C# 소스파일

#### → Main( ) 메소드

- 응용 프로그램의 시작점
- Application 클래스의 Run( ) 메소드를 호출하여 응용 프로그램을 실행함



## 1. 원폼 애플리케이션

### 4) 원폼 애플리케이션 생성하기

#### (2) 주요 프로젝트 파일

- 예제 : Program.cs

#### 예제

```
static class Program
{
    /// <summary>
    /// 해당 응용 프로그램의 주 진입점입니다.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();

        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1()); // --- ①
    }
}
```





## 1. 원폼 애플리케이션

### 4) 원폼 애플리케이션 생성하기

#### (2) 주요 프로젝트 파일

##### ▪ 주요 프로젝트 파일

파일명	설명
Form1.cs	원폼 응용 프로그램의 C# 소스파일

→ 생성자

- 폼에서 사용하는 각종 컴포넌트와 클래스의 멤버 초기화
- 폼에 있는 각종 컴포넌트들 초기화하는  
InitializeComponent() 메소드 호출함

##### ▪ 예제 : Form1.cs

예제

```
public partial class Form1: Form
{
    public Form1() // --- ②
    {
        InitializeComponent();
    }
}
```

### 1. 윈폼 애플리케이션

#### 5) 윈폼 애플리케이션 작성하기

##### (1) 윈폼 애플리케이션 작성

- 윈폼 애플리케이션 작성

- 디자인 : 컨트롤, 컴포넌트, 프로퍼티, 이벤트

- 코드 : 멤버, 이벤트 처리

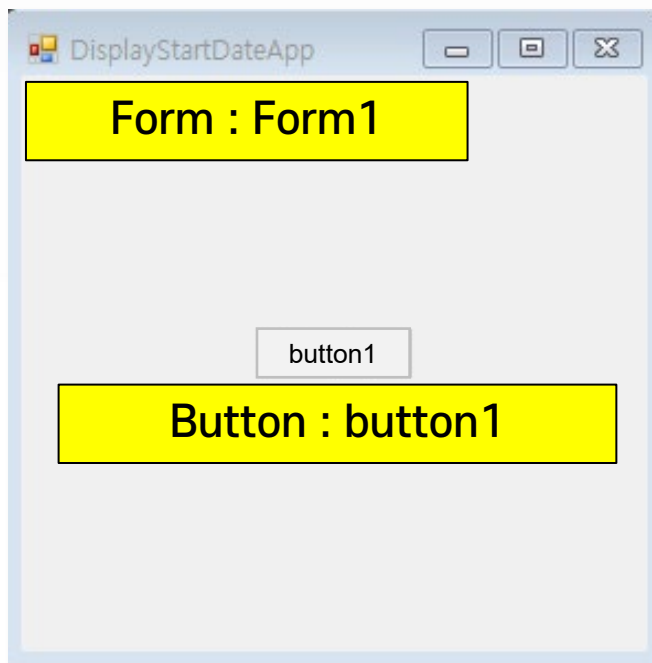
- 애플리케이션 실행

##### (2) 예제

- 예제 : DisplayStartDateApp.cs

- 디자인

- 폼 설계





## 1. 원폼 애플리케이션

### 5) 원폼 애플리케이션 작성하기

#### (2) 예제

- 예제 : DisplayStartDateApp.cs

→ 디자인

- 프로퍼티

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

→ 코드

- 멤버

코드

```
public Form1() {
    // ...
    startDateTime = DateTime.Now;
}
DateTime startDateTime;
public DateTime GetStartDateTime() {
    return startDateTime;
}
private void button1_Click(object sender, EventArgs e) {
    MessageBox.Show(GetStartDateTime().ToString());
}
```



1. 윈폼 애플리케이션

5) 윈폼 애플리케이션 작성하기

(2) 예제

▪ 예제 : DisplayStartDateApp.cs

→ 코드

- 이벤트 처리기

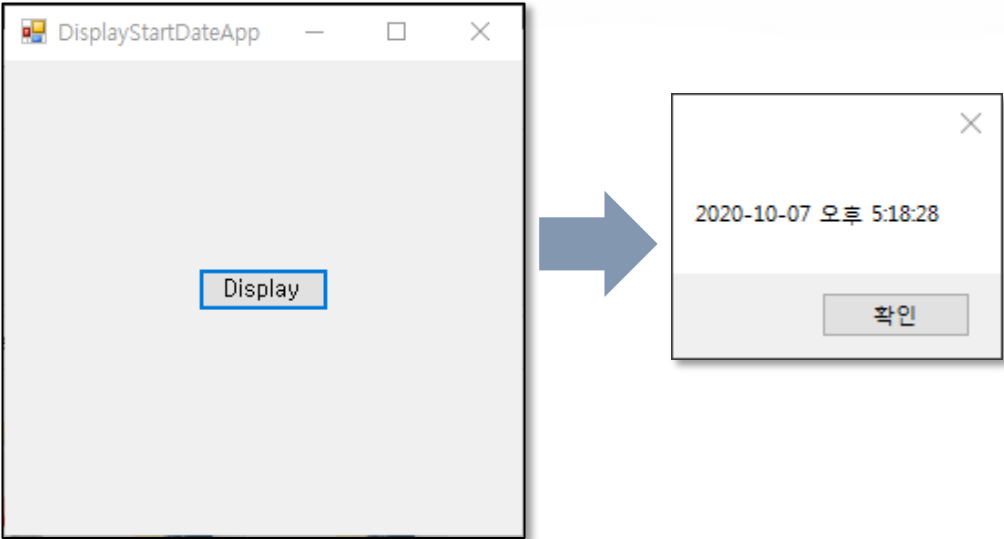
컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	Button1_Click( )

→ 애플리케이션 실행

실행 방법

애플리케이션을 실행한 후, 폼에 있는 “Display” 버튼을 클릭함

실행 결과





## 1. 윈폼 애플리케이션

### 5) 윈폼 애플리케이션 작성하기

#### (3) 프로퍼티

##### ▪ 프로퍼티 목록

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

#### (4) 멤버

##### ▪ 멤버

→ 컴포넌트와 컨트롤을 제외한 클래스 멤버에 대한 선언과 초기화 등을 소스 코드로 작성하는 곳

##### ▪ 멤버 코드 추가

→ 생성자에 초기화 부분 추가함

#### 정의 형태

```
public Form1() {
    InitializeComponent();

    //
    startDateTime = DateTime.Now;
}
```

→ 멤버 선언이나 메소드 등의 소스 코드 추가함

- 폼 클래스내의 적당한 곳에 추가함



## 1. 원폼 애플리케이션

### 5) 원폼 애플리케이션 작성하기

#### (5) 이벤트 처리기

##### ▪ 이벤트 처리기

→ 통합 환경에서 생성하는 처리기의 이름 :

<이벤트 처리기를 등록하려는 객체의 이름>\_<이벤트 이름>

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

#### 정의 형태

```
private void button1_Click(object sender, EventArgs e) {  
    MessageBox.Show(GetStartDate().ToString());  
}
```

#### → 매개변수

- 이벤트를 발생시킨 객체
- 이벤트에 관련된 정보를 가진 객체
- EventArgs 클래스형이나 이의 파생 클래스형

## 2. 폼 클래스

### 1) 폼 클래스 개요

#### (1) 개요

- 폼 클래스의 개요
  - Form 클래스를 나타냄
  - 폼의 외형을 설정하는 프로퍼티, 폼의 동작을 정의하는 메소드, 그리고 사용자와 상호작용을 처리하는 이벤트 등이 정의되어 있는 클래스
  - 윈도우 폼을 다루는 핵심 클래스로 윈도우 폼을 사용하는 모든 클래스의 베이스 클래스

#### 정의 형태

```
using System.Windows.Forms;  
public class UserDefinedForm : Form {  
    // 새 클래스 멤버 정의  
    // Form 클래스의 멤버 재정의  
}
```

## 2. 폼 클래스

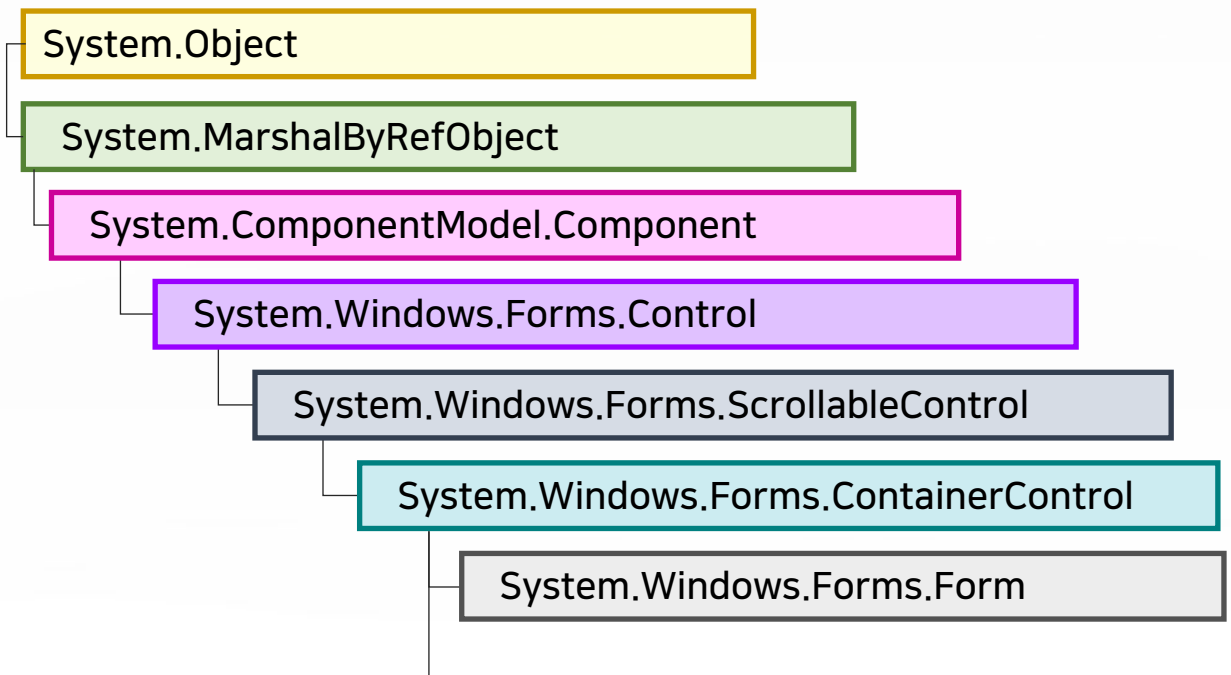
### 2) 폼 클래스의 계층도

#### (1) 계층도

- 폼 클래스의 계층도

→ 윈도우 폼을 사용하는 모든 클래스의 베이스 클래스

→ System.Windows.Forms 네임스페이스에 포함





## 2. 폼 클래스

### 3) 폼 클래스의 베이스 클래스

#### (1) Component 클래스

- Component 클래스

- .NET 프레임워크에서 컴포넌트 기반 프로그래밍 기법에서의 컴포넌트 개념을 지원하는 클래스
- System.ComponentModel 네임스페이스에 포함됨
- 윈폼 애플리케이션뿐만 아니라 컴포넌트 개념이 필요한 다른 곳에서도 사용함
- 윈폼에서 제공하는 화면을 구성하는 여러 요소 뿐만 아니라 화면에 표시되지 않는 요소도 컴포넌트로 표현함

#### (2) Control 클래스

- Control 클래스

- 윈폼 애플리케이션에서 화면에 표시되는 구성요소를 나타내기 위해서 사용되는 컨트롤들의 베이스 클래스
- 폼과 폼에 배치되는 여러 요소에서 공통적으로 필요한 멤버를 정의함

## 2. 폼 클래스

### 3) 폼 클래스의 베이스 클래스

#### (3) ScrollableControl 클래스

- ScrollableControl 클래스

→ 스크롤 개념이 필요한 컨트롤을 정의하기 위해서 사용되는 베이스 클래스

#### (4) ContainerControl 클래스

- ContainerControl 클래스

→ 여러 컨트롤이나 컴포넌트를 포함할 수 있는 컨트롤에서의 포커스 관리를 하기 위한 베이스 클래스

## 2. 폼 클래스

### 4) 폼 클래스의 멤버

#### (1) InitializeComponent() 메소드

- InitializeComponent() 메소드

- 폼에 있는 각종 컴포넌트들을 초기화

- 통합 개발 환경이 관리하는 메소드

#### (2) Form 클래스 멤버의 종류

Form 클래스의  
자체 정의 멤버

Control 클래스의  
멤버



## 2. 폼 클래스

### 5) 폼 클래스의 주요 프로퍼티

#### (1) Form BorderStyle 프로퍼티

- Form BorderStyle 프로퍼티

프로퍼티	자료형	속성	설명
FormBorderStyle	FormBorderStyle	get/set	폼의 테두리 모양을 설정하거나 참조
StartPosition	FormStartPosition	get/set	폼이 처음 나타나는 위치를 설정하거나 참조

기호상수	설명
None	테두리가 없음
FixedSingle	고정된 단일선 테두리
Fixed3D	고정된 3차원 테두리
FixedDialog	고정된 대화상자 스타일의 굵은 테두리
Sizable	크기를 조정할 수 있는 테두리(기본값)
FixedToolWindow	크기를 조정할 수 없는 도구 창 테두리
SizableToolWindow	크기를 조정할 수 있는 도구 창 테두리



2. 폼 클래스

5) 폼 클래스의 주요 프로퍼티

(2) StartPosition 프로퍼티

- StartPosition 프로퍼티

프로퍼티	자료형	속성	설명
FormBorderStyle	FormBorderStyle	get/set	폼의 테두리 모양을 설정하거나 참조
StartPosition	FormStartPosition	get/set	폼이 처음 나타나는 위치를 설정하거나 참조

기호상수	설명
CenterParent	폼이 해당 부모 폼의 범위 내에서 가운데에 맞춰짐
CenterScreen	폼이 현재 디스플레이의 가운데에 맞춰지며 크기는 해당 폼의 크기 내에서 지정됨
Manual	폼의 위치는 Location 프로퍼티에 의해 결정됨
WindowsDefaultLocation	폼의 위치는 윈도우 운영 체제가 결정함
WindowsDefaultBounds	폼의 위치와 크기는 윈도우 운영 체제가 결정함



## 2. 폼 클래스

### 6) 폼 클래스의 메소드

#### (1) 주요 메소드

▪ 주요 메소드 목록

메소드	설명
Close( )	폼을 닫음
Activate( )	폼을 활성화 함
AddOwnedForm( )	다른 폼을 현재 폼에 소유시킴
RemoveOwnedForm( )	현재 폼에 소유된 폼을 제거함
SetDesktopBounds( )	바탕 화면에서 폼의 범위를 설정함
SetDesktopLocation( )	바탕 화면에서 폼의 위치를 설정함



2. 폼 클래스

7) 폼 클래스의 이벤트

(1) 주요 이벤트

▪ 주요 이벤트 목록

이벤트	설명
Activated	폼이 활성화될 때 발생함
FormClosed	폼이 종료된 후 발행함
FormClosing	폼의 종료 과정 중에 발생함
Deactivate	폼이 비활성화될 때 발생함
Load	폼이 로드될 때 발생함

(2) Load/FormClosed 이벤트

Load 이벤트

- 폼이 메모리로 로딩 될 때 발생하는 이벤트
- 주로 폼에서 사용하는 자원을 할당하고 내부 구성 요소를 동적으로 초기화하는데 사용함

FormClosed 이벤트

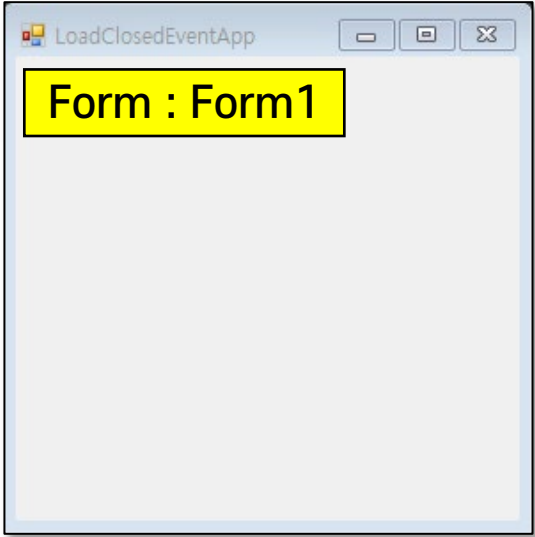
- 폼이 닫히면 발생하는 이벤트
- FormClosing 이벤트의 FormClosingEventArgs 객체의 Cancel 프로퍼티의 값이 거짓일 때만 발생함



2. 폼 클래스

7) 폼 클래스의 이벤트

- (3) 예제
- 예제 : LoadClosedApp.cs
  - 디자인
    - 폼 설계



- 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	LoadClosedApp

- 이벤트

컨트롤 : (Name)	이벤트	메소드명
Form : Form1	Load	Form1_Load( )
	FormClosed	Form1_FormClosed( )



## 2. 폼 클래스

### 7) 폼 클래스의 이벤트

#### (3) 예제

- 예제 : LoadClosedApp.cs - 코드

→ 코드

- 멤버

#### 코드

```
private void Form1_Load(object sender, EventArgs e)
{
    MessageBox.Show("This form is loaded");
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    MessageBox.Show("This form is closed");
}
```

## 2. 폼 클래스

### 7) 폼 클래스의 이벤트

#### (3) 예제

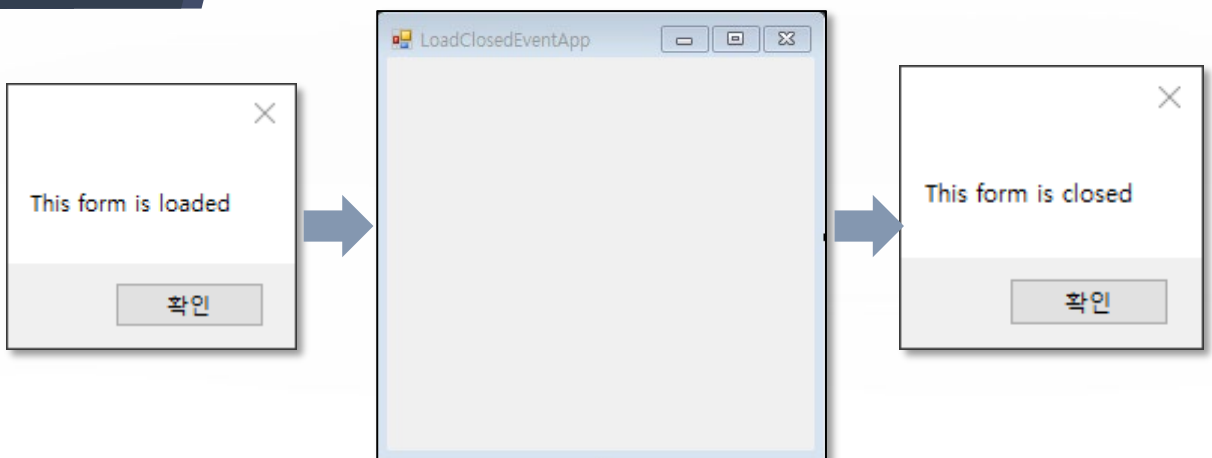
- 예제 : LoadClosedApp.cs - 코드

→ 애플리케이션 실행

#### 실행 방법

폼을 종료하면 폼의 종료를 알리는 메시지 상자가 나타남

#### 실행 결과



## 2. 폼 클래스

### 7) 폼 클래스의 이벤트

#### (4) FormClosing 이벤트

- FormClosing 이벤트

- 용도 : 폼이 닫히는 도중에 발생함

- 매개변수 : FormClosingEventArgs 형

- Cancel 프로퍼티 : 닫기 작업의 취소 여부를 설정

- 설명 : Cancel 프로퍼티는 닫기 작업의 취소 여부를 결정하기 위해 사용함

- 예

- 문서편집기에서작업 중인 문서를 저장하지 않고 종료
    - 현재작업 중인 문서를 저장하고 종료할지, 저장하지 않고 종료할지, 또는 종료 작업을 취소할지 여부를 사용자에게 물어보는 대화상자를 표시함
    - 대화상자에서 종료 작업에 대한 취소를 선택하면 프로그램 종료 작업은 취소됨

## 2. 폼 클래스

### 7) 폼 클래스의 이벤트

#### (5) 예제

- 예제 : PromptClosedApp.cs

#### 예제

```
private void Form1_FormClosing
(object sender, FormClosingEventArgs e)
{
    if(MessageBox.Show("Do you want to close?","Prompt",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        e.Cancel = false;
    else
        e.Cancel = true;
}
private void button1_Click(object sender, EventArgs e)
{
    Close();
}
```

## 2. 폼 클래스

## 7) 폼 클래스의 이벤트

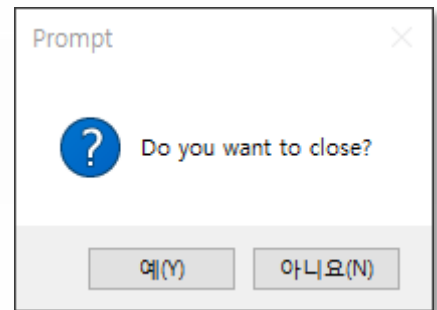
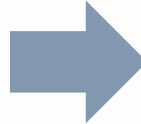
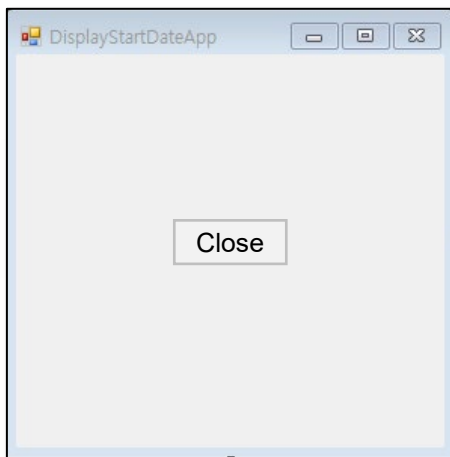
## (5) 예제

- 예제 : PromptClosedApp.cs

## 실행 방법

폼의 바탕에 있는 버튼을 누르면, 폼의 닫기 여부를 물어보는 대화상자가 나타남

## 실행 결과



## 3. 컨트롤 클래스

### 1) 컨트롤 클래스 개요

#### (1) 정의

##### ■ 컨트롤 클래스

→ Control 클래스를 의미

→ 윈폼 애플리케이션에서 화면에 표시되어 사용자와 상호작용을 수행하는 컨트롤들을 위한 베이스 클래스

→ 폼과 폼에 배치되는 여러 컨트롤에서 공통적으로 필요한 멤버들이 정의

#### (2) 컴포넌트와 컨트롤의 특징

##### 컴포넌트

화면에 자신을 그리지 않는  
컴포넌트

- 이미지 리스트나 타이머 등이 있음
- 폼에 표시되지 않는 컴포넌트는 Component 클래스에서 파생된 클래스로 정의함

##### 컨트롤

화면에 자신을 그릴 수 있는  
컴포넌트

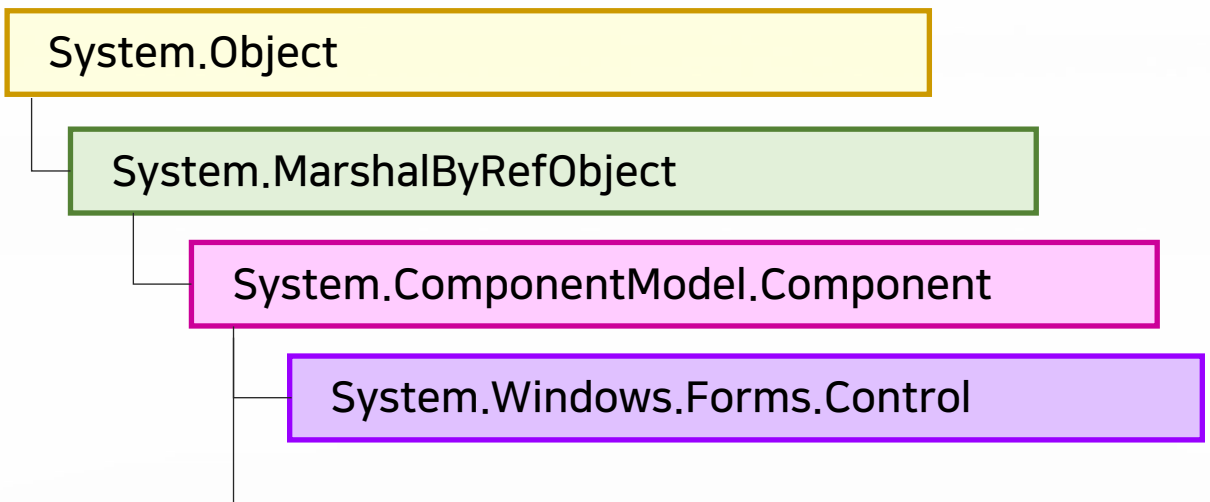
- 버튼과레이블,체크박스 등이있음
- 폼에 표시되는 컴포넌트인 컨트롤은 Control 클래스에서 파생된 클래스로 정의함

### 3. 컨트롤 클래스

#### 2) 컨트롤 클래스의 계층도

##### (1) 계층도

- System.Windows.Forms 네임스페이스





### 3. 컨트롤 클래스

#### 3) 컨트롤 클래스의 프로퍼티

##### (1) 컨트롤 클래스 프로퍼티

■ 컨트롤 클래스의 프로퍼티 목록

1/2

프로퍼티	자료형	속성	설명
Name	String	get/set	컨트롤의 이름으로 소스 파일에서 참조할 때 사용
Text	String	get/set	컨트롤에 표시되는 문자열을 설정하거나 참조
Font	Font	get/set	컨트롤에 표시되는 문자열의 글꼴을 설정
ForeColor	Color	get/set	컨트롤에 표시되는 문자열의 글자색을 설정
BackColor	Color	get/set	컨트롤의 배경색을 설정
BackgroundImage	Image	get/set	컨트롤의 배경 이미지를 설정
Visible	Bool	get/set	컨트롤의 화면 표시 여부를 설정하거나 참조
Enabled	Bool	get/set	컨트롤의 활성화 여부를 설정하거나 참조
Size	Size	get/set	컨트롤의 크기를 설정하거나 참조
ClientSize	Size	get/set	컨트롤에서 클라이언트 영역의 크기
ClientRectangle	Rectangle	get/set	컨트롤에서 클라이언트 영역의 위치와 크기
Dock	DockStyle	get/set	상위 컨트롤 내에서 컨트롤의 크기와 위치를 설정





3. 컨트롤 클래스

3) 컨트롤 클래스의 프로퍼티

(1) 컨트롤 클래스 프로퍼티

컨트롤 클래스의 프로퍼티 목록

2/2

프로퍼티	자료형	속성	설명
Width/Height	int	get/set	컨트롤의 폭과 높이를 설정하거나 참조
Location	Point	get/set	컨트롤의 위치를 설정하거나 참조
Left/Top	int	get/set	컨트롤의 왼쪽과 위쪽 좌표를 설정하거나 참조
Right/Bottom	int	get/set	컨트롤의 오른쪽/아래쪽 좌표 (읽기 전용 프로퍼티)
TabIndex	int	get/set	탭 키에 의한 포커스 이동 순서를 설정
TabStop	bool	get/set	탭 키에 의한 포커스 설정 가능 여부를 설정
Parent	Control	get/set	현재 컨트롤이 포함된 상위 컨트롤
Tag	object	get/set	컨트롤에 연관된 객체를 설정하거나 참조
Cursor	Cursor	get/set	컨트롤에서 사용하는 커서를 설정하거나 참조
ContextMenu	ContextMenu	get/set	컨트롤에 해당하는 상황메뉴를 설정

## 3. 컨트롤 클래스

## 4) 컨트롤 클래스의 주요 메소드

## (1) 컨트롤 클래스 메소드

## ▪ 컨트롤 클래스의 메소드 목록

메소드	설명
Show()	컨트롤을 화면에 표시함
Hide()	컨트롤을 화면에 표시하지 않음
Invalidate()	컨트롤의 영역을 무효화하여 다시 그려지도록 함(Paint Event)
BringToFront()	컨트롤의 화면 표시 순서를 가장앞으로 변경함
SendToBack()	컨트롤의 화면 표시 순서를 가장뒤로 변경함
PointToClient()	화면상의 좌표를 컨트롤 내부의 상대좌표로 변환함
PointToScreen()	컨트롤 내부의 좌표를 화면상의 좌표로 변환함
RectangleToClient()	사각형의 좌표를 컨트롤 내부의 상대좌표로 변환함
RectangleToScreen()	사각형의 좌표를 전체 화면상의 좌표로 변환함
ResetXxx()	Xxx이름을 가진 프로퍼티의 값을 초기값으로 설정함

## 3. 컨트롤 클래스

## 5) 컨트롤 클래스의 주요 이벤트

## (1) 컨트롤 클래스 이벤트

## ▪ 컨트롤 클래스의 주요 이벤트 목록

이벤트	설명
Click	컨트롤을 마우스로 클릭할 때 발생하는 이벤트
Move	컨트롤의 위치가 변경되었을 때 발생하는 이벤트
Paint	컨트롤을 다시 그려야 할 때 발생하는 이벤트
Resize	컨트롤의 크기가 변경되었을 때 발생하는 이벤트
XxxChanged	Xxx 이름을 가진 프로퍼티가 변경되었을 때 발생하는 이벤트

## (2) 주요 이벤트

## ▪ 컨트롤 클래스의 주요 이벤트 목록

이벤트	설명
Paint	컨트롤을 다시 그려야 할 때 발생하는 이벤트

→ PaintEventArgs 클래스형의 매개변수

→ ClipRectangle: 새로 그려야 하는 영역

→ Graphics: 컨트롤을 그리는데 사용하는 Graphics 객체



## 1. 윈폼 애플리케이션

- 윈폼 애플리케이션의 개요

컨트롤을 사용하여 화면을 구성하고 이벤트 발생 시, 이벤트 처리기로 기술하는 프로그램

- 윈도우 폼

운영체제에서 제공하는 기본적인 화면 단위인 창을 말하는 개념

- 윈폼 애플리케이션의 핵심 클래스

폼 클래스

컴포넌트 클래스

컨트롤 클래스

## 2. 폼 클래스

- 폼의 외형을 설정하는 프로퍼티, 폼의 동작을 정의하는 메소드, 그리고 사용자와 상호작용을 처리하는 이벤트 등이 정의되어 있는 클래스

- 윈도우 폼을 다루는 핵심 클래스로 윈도우 폼을 사용하는 모든 클래스의 베이스 클래스



## 3. 컨트롤 클래스

- 원폼 애플리케이션에서 화면에 표시되어 사용자와 상호작용을 수행하는 컨트롤들을 위한 베이스 클래스
- 폼과 폼에 배치되는 여러 컨트롤에서 공통적으로 필요한 멤버들이 정의