삼성전기 : C#언어 고급(Step1)

『3일차 』: WinForm으로 만드는 사용자 인터페이스

2025. 11.17-2025.11.19(3일, 24시간)

Prepared by DaeKyeong K

Ph.D.

학습일정 및 내용

시간	1일차	2일차	3일차		Total
1교시 08:00~08:50	『10장』 배열과 컬렉션, 그리고 인덱서 1. 배열이란? 2. 배열을 초기화하는 세 가지 방법	『11장』 일반화 프로그래밍 1. 일반화 프로그래밍이란? 2. 일반화 메소드 실습 1: 일반화 메서드로 배열 검색 실습 2: 일반화 메서드로 배열 필터링	『13장』 대리자와 이벤트 1. 대리자란? 2. 대리자는 왜,언제 사용하나요? 3. 일반화 대리자 4. 대리자 체인 실습 1: 계산기 with 대리자		
2교시 09:00~09:50	3. System.Array 4. 배열 분할하기	3. 일반화 클래스 4. 형식 매개 변수 제약 실습 3: 제약이 있는 일반화 클래스	5. 익명 메소드 6. 이벤트: 객체에 일어난 사건 알리기 실습 2: 온도 모니터링 시스템 실습 3: 주문 처리 시스템		
3교시 10:00~10:5	5. 2차원 배열 실습 1: 학생 성적 관리-EduScore 실습 2: 좌석 예약 시스템 -Seat-Now	5. 일반화 컬렉션 실습 4: 단어 빈도수 계산기 실습 5: 학생 관리 시스템	『20장』 WinForm으로 만드는 사용자 인터페이스 1. WinForm 소개 및 개발 환경 설정 2. WinForm 윈도우 만들기 3. Application 클래스 4. 메시지 필터링		
4교시 11:00~11:50	6. 다차원 배열 7. 가변배열(Jagged Array) 실습 3: 월별 일수 관리 실습 4: 시험 점수 관리 실습 5: 조직도 관리 시스템-TeamStructure	6. Foreach를 사용할 수 있는 일반화 클래스	5. 윈도우를 표현하는 Form 클래스		
5교시 13:00~13:50	8. 컬렉션(Collection) 컬렉션- ArrayList 컬렉션- Queue 컬렉션- Stack 컬렉션- Hashtable	『12장』 예외 처리하기 1. 예외에 대하여 2. try ~ catch로 예외 받기 실습 1: 안전한 계산기	6. 폼 디자이너를 이용한 WinForm UI 구성		
6교시 14:00~14:50	실습 6: Queue 활용 - 고객 대기열-WaitingON 실습 7: Queue 활용 - 프린터 출력 대기열- Qprint	3. System.Exception 클래스	7. 사용자 인터페이스와 비동기 작업		
7교시 15:00~15:50	실습 8: 단어 빈도수 계산- WordCount 실습 9: Hashtable로 전화번호부 만들기-HashPhone	4. 예외 던지기 5. try~catch와 finally 6. 사용자 정의 예외 클래스 7. 예외 필터(Exception Filter)	실습 1: 윈폼으로 계산기 만들기		
8교시 16:00~16:50	컬렉션 초기화 9. 인덱서(Indexer) 10. Foreach가 가능한 객체 만들기	8. 예외 처리 다시 생각해보기 실습 2: 사용자 등록 시스템	필기 평가		
Total Time	8	8	8		24

* 커리큘럼은 협의에 따라 변경 가능

목차

- WinForm 소개
- WinForm으로 윈도우 띄우기
- Application 클래스
- 윈도우 메시지
- 윈도우 컨트롤(with 이벤트)

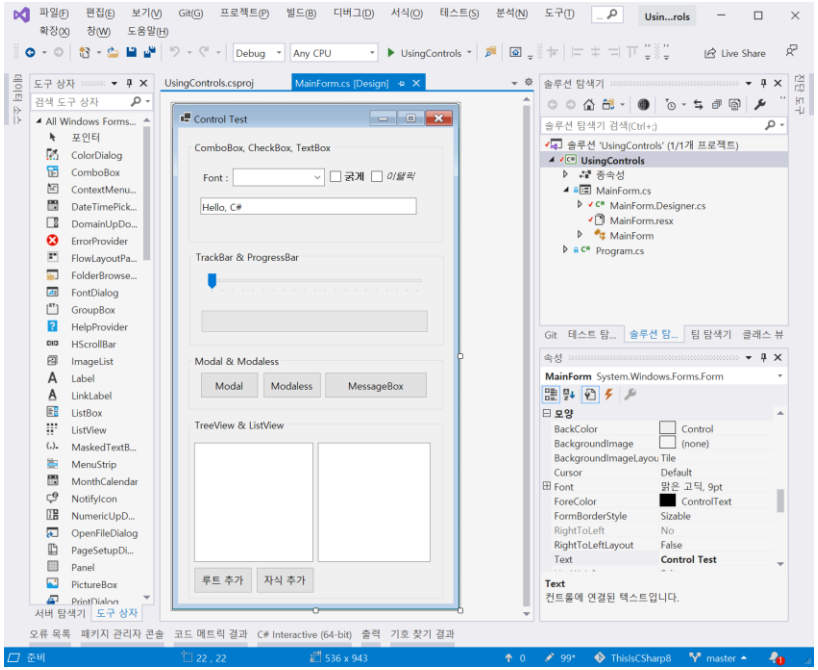
1. WinForm 소개 및 개발 환경 설정

WinForm이란?

- **Windows Forms (WinForm)는 Microsoft .NET Framework에서 제공하는 Windows 데스크톱 애플리케이션 개발 플랫폼입니다.**

WinForm 소개

- .NET UI 라이브러리 두 가지
  - WinForm(Windows Form) : 익히기 쉽고 높은 생산성 제공
  - WPF(Windows Presentation Foundation) : 세련된 UI와 화려한 효과를 제공하나, 학습곡선이 가파름.
- WYSIWYG 방식의 GUI 프로그램 개발



1. WinForm 소개 및 개발 환경 설정

WinForm의 특징

- **장점:**
  - 빠른 개발 속도 (드래그 앤 드롭 디자이너)
  - 풍부한 컨트롤 라이브러리
  - 이벤트 기반 프로그래밍
  - .NET Framework 완전 통합
  - 레거시 시스템과 호환성

- **단점:**
  - 구식 UI 스타일
  - 크로스 플랫폼 지원 제한적
  - 최신 UI 트렌드 반영 어려움

1. WinForm 소개 및 개발 환경 설정

WinForm vs 다른 기술

특징	WinForm	WPF	UWP	WinUI 3
출시 시기	2002	2006	2015	2021
학습 난이도	쉬움	중간	중간	중간
디자인 유연성	낮음	높음	높음	높음
성능	좋음	매우 좋음	좋음	매우 좋음
권장 용도	내부 도구, 레거시	데스크톱 앱	스토어 앱	최신 앱

1. WinForm 소개 및 개발 환경 설정

WinForm 애플리케이션 구조

```
WinForm Application
├── Program.cs          // 진입점
├── Form1.cs            // 폼 클래스 (코드)
├── Form1.Designer.cs   // 폼 디자이너 코드 (자동 생성)
└── Form1.resx          // 리소스 파일
```

1. WinForm 소개 및 개발 환경 설정

기본 구조 설명

```
// Program.cs - 애플리케이션 진입점
using System;
using System.Windows.Forms;

namespace MyFirstWinForm
{
    static class Program
    {
        [STAThread]  // Single-Threaded Apartment (COM 상호 운용)
        static void Main()
        {
            // 애플리케이션 초기화
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            // 메인 폼 실행
            Application.Run(new Form1());
        }
    }
}
```

[STAThread] 특성(Attribute)

STA(Single Threaded Apartment) 모델로 COM(Component Object Model)과 상호작용하기 위한 설정입니다.
윈도우 폼, 클립보드, 드래그앤드롭, 메시지 루프 등 UI 관련 기능들이 STA 스레드를 필요로 합니다.
즉, UI 스레드가 단일 스레드로 동작하도록 지정하는 표시입니다.
만약 이 속성을 생략하면 일부 WinForms 기능이 비정상 동작하거나 예외가 발생할 수 있습니다.

Application.EnableVisualStyles()
윈도우의 테마(시각 스타일) 를 적용합니다.
→ 버튼, 체크박스, 텍스트박스 등이 OS 스타일로 렌더링됨.
예:
적용 전 → 회색 고전 스타일
적용 후 → 윈도우 10/11의 깔끔한 현대적 UI

Application.SetCompatibleTextRenderingDefault(false)
텍스트 렌더링 방식을 설정합니다.
false → GDI+ 기반 (새로운 렌더링 방식, 기본값)
true → GDI 기반 (이전 .NET 버전 호환용)
일반적으로 false로 설정하는 것이 권장됩니다.

1. WinForm 소개 및 개발 환경 설정

기본 구조 설명

```
// Form1.cs - 폼 클래스
using System;
using System.Windows.Forms;

namespace MyFirstWinForm
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();  // 디자이너 코드 초기화
        }

        // 이벤트 핸들러 등을 여기에 작성
    }
}
```

partial

Form1 클래스가 두 개 이상의 파일로 나뉘어 있음을 의미합니다.

실제로 WinForms에서는 Form1.cs와 Form1.Designer.cs 두 파일이 함께 작동합니다.

Form1.cs ← 코드 로직, 이벤트 처리
Form1.Designer.cs ← 버튼, 레이블, 레이아웃 등 UI 요소 정의

1. WinForm 소개 및 개발 환경 설정

Visual Studio 설치 및 설정

1. Visual Studio 2019/2022
2. .NET desktop development 워크로드
3. .NET Framework 4.7.2 이상

2. WinForm 윈도우 만들기

C# 코드로 WinForm 윈도우 만들기

- 윈도우를 띄우는 과정 비교

Win32	① 윈도우 클래스를 정의 ② 정의된 윈도우 클래스를 등록 ③ 윈도우를 생성한 다음 ④ 윈도우를 사용자에게 표시 ⑤ 메시지 루프 실행
WinForm	① System.Windows.Forms.Form 클래스에서 파생된 윈도우 폼 클래스를 선언 ② ① 번에서 만든 클래스의 인스턴스를 인수로 넘겨 System.Windows.Forms.Application.Run() 메소드 호출

SimpleWindow

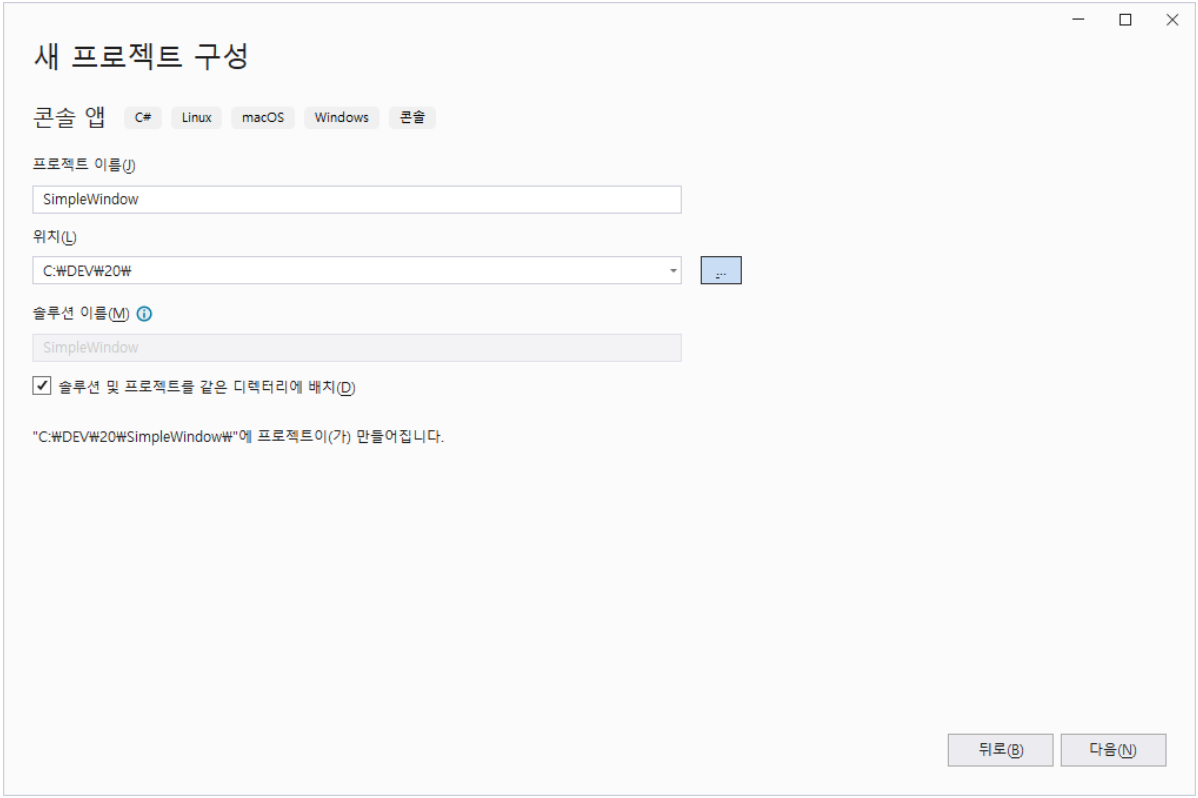
WinForm 윈도우 띄우기

```
using System;

namespace SimpleWindow
{
    class MainApp : System.Windows.Forms.Form
    {
        static void Main(string[] args)
        {
            System.Windows.Forms.Application.Run(new MainApp());
        }
    }
}
```

2. WinForm 윈도우 만들기

SimpleWindow

새 프로젝트 구성

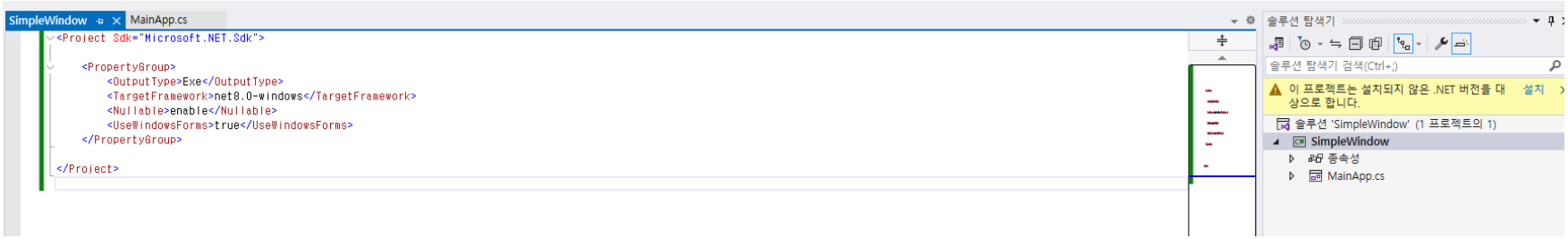


2. WinForm 윈도우 만들기

SimpleWindow

SimpleWindow.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

        <PropertyGroup>
                <OutputType>Exe</OutputType>
                <TargetFramework>net8.0-windows</TargetFramework>
                <Nullable>enable</Nullable>
                <UseWindowsForms>true</UseWindowsForms>
        </PropertyGroup>

</Project>
```



SimpleWindow

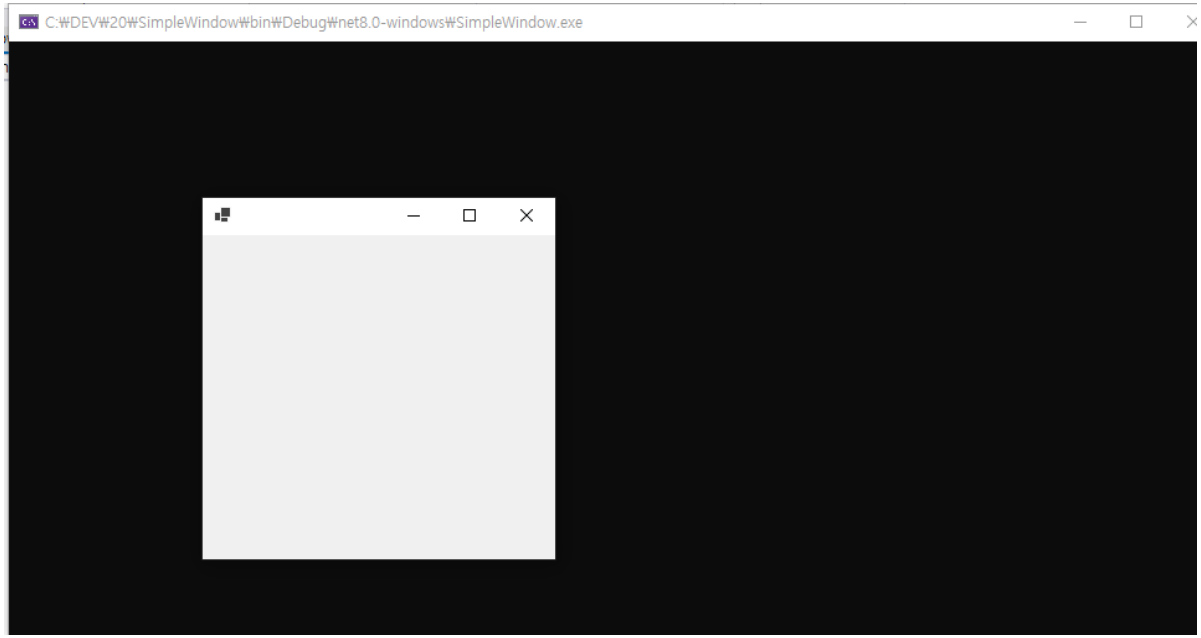
MainApp.cs

```
using System;

namespace SimpleWindow
{
    class MainApp : System.Windows.Forms.Form
    {
        static void Main(string[] args)
        {
            System.Windows.Forms.Application.Run(new MainApp());
        }
    }
}
```

2. WinForm 윈도우 만들기

SimpleWindow

실행 결과



Application 클래스

- 윈도우 응용 프로그램 **시작/종료** 메소드 제공
  - Application.Run()
  - Application.Exit()
- **윈도우 메시지 처리**하는 것

```
class MyForm : System.Windows.Forms.Form
{
}

class MainApp
{
   static void Main(string[] args)
   {
      MyForm form = new MyForm();
      form.Click += new EventHandler(
         (sender, eventArgs) =>
         {
            Application.Exit();
         });
      Application.Run(form);
   }
}
```

3. Application 클래스

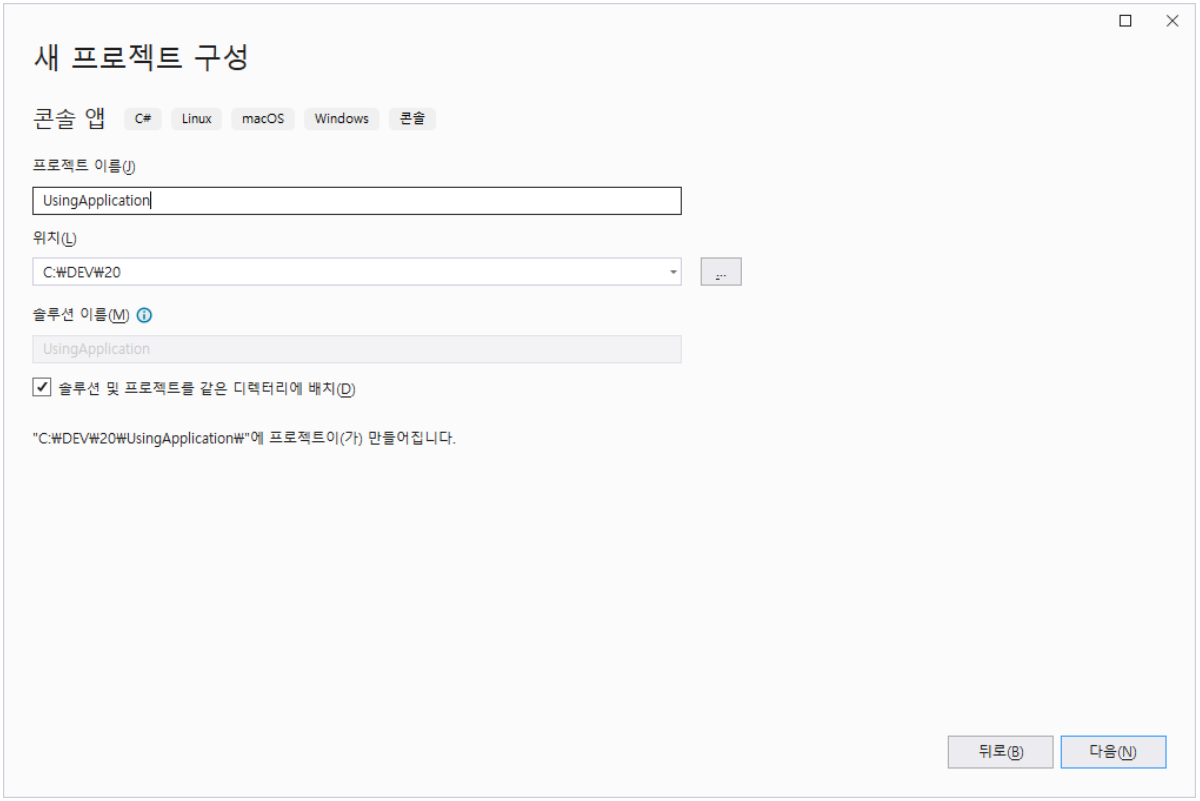
Application 클래스 주요 멤버

멤버	설명	예시
Run()	메시지 루프 시작	Application.Run(new Form1())
Exit()	애플리케이션 종료	Application.Exit()
DoEvents()	대기 중인 메시지 처리	Application.DoEvents()
StartupPath	실행 파일 경로	Application.StartupPath
ProductName	제품 이름	Application.ProductName
ProductVersion	제품 버전	Application.ProductVersion

3. Application 클래스

UsingApplication

새 프로젝트 구성



UsingApplication

UsingApplication.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

UsingApplication

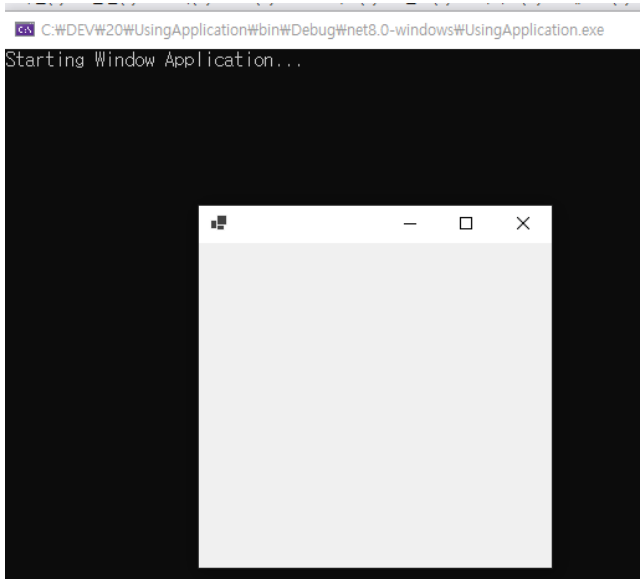
MainApp.cs

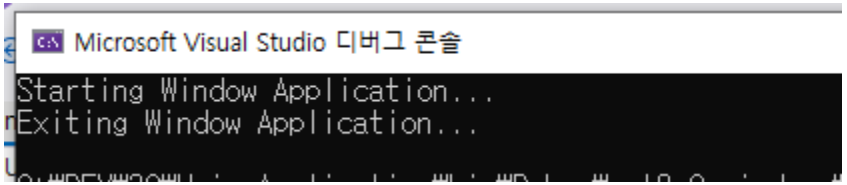
```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

3. Application 클래스

UsingApplication

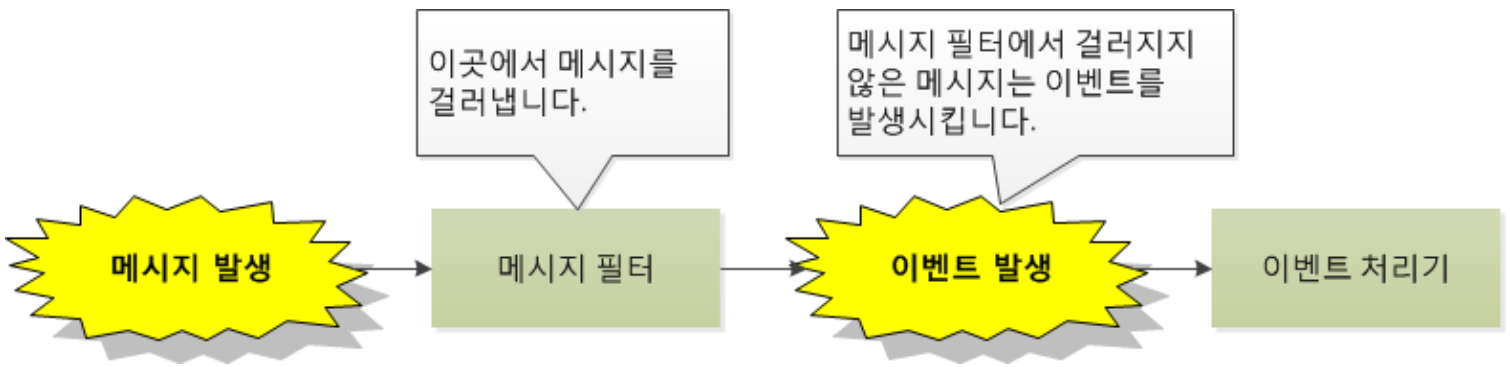
실행 결과 > 종료 버튼 클릭 시





메시지 필터링(Message Filtering) (1/2)

- 윈도우 메시지의 생성 및 처리 예
  - 하드웨어(마우스나 키보드)로부터 인터럽트 발생
  - 해당 인터럽트는 윈도우에 의해 윈도우 메시지(Windows Message) 생성
  - 윈도우 메시지를 응용 프로그램에게 전송
  - 응용 프로그램은 해당 메시지를 처리
- 메시지 필터는 관심 있는 메시지를 정제
- Application 클래스가 메시지 필터 제공



메시지 필터링(Message Filtering) (2/2)

- Application.AddMessageFilter() 메소드를 이용하여 응용프로그램에 메시지 필터 설치
- 메시지 필터는 다음과 같이 IMessageFilter 인터페이스 상속

```
public interface IMessageFilter
{
    bool PreFilterMessage(ref Message m);
}
```

마우스 이동부터 마우스의 왼쪽, 오른쪽, 가운데 버튼 동작, 마우스 휠 굴림 메시지를 모두 정제

```
public class MessageFilter : IMessageFilter
{
    public bool PreFilterMessage(ref Message m)
    {
        if (m.Msg >= 0x200 && m.Msg <= 20E)
        {
            Console.WriteLine("발생한 메시지: " + m.Msg);
            return true;
        }
        return false;
    }
}

// ...
Application.AddMessageFilter( new MessageFilter() );
```

4. 메시지 필터링

메시지 처리 흐름

사용자 동작 (마우스 클릭)

↓

Windows OS가 메시지 생성

↓

메시지 큐에 추가

↓

Application.Run()의 메시지 루프

↓

해당 윈도우로 메시지 전달

↓

WndProc() 메서드에서 처리

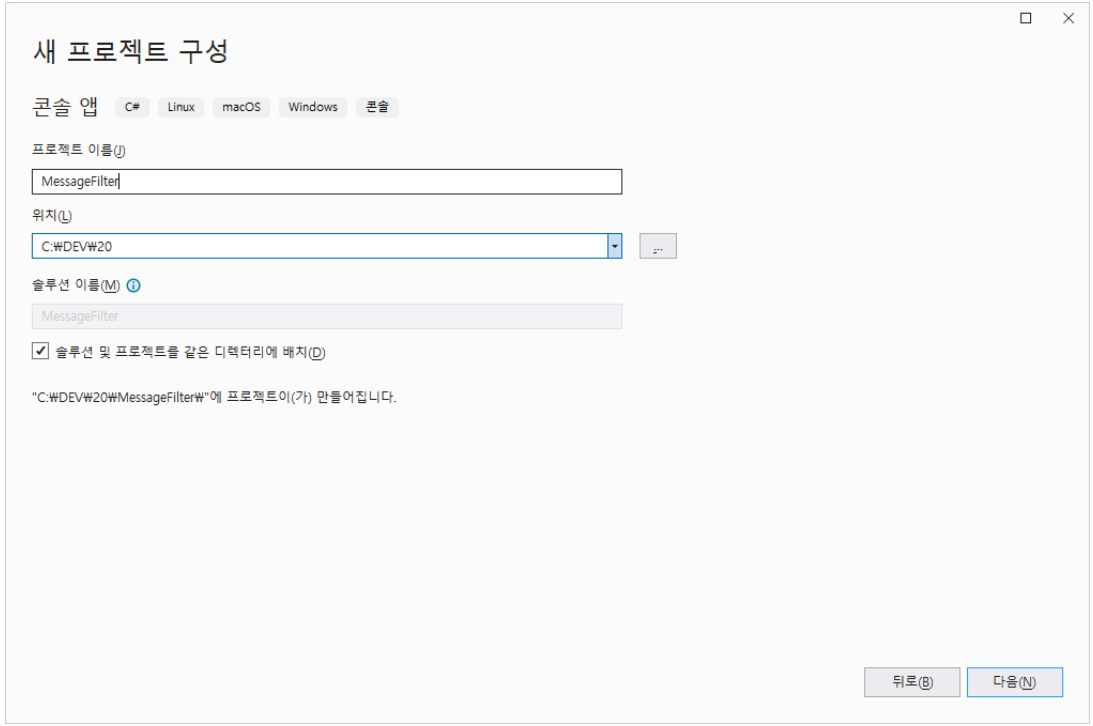
↓

이벤트 발생 (Click, KeyDown 등)

4. 메시지 필터링

MessageFilter

새 프로젝트 구성



MessageFilter

MessageFilter.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

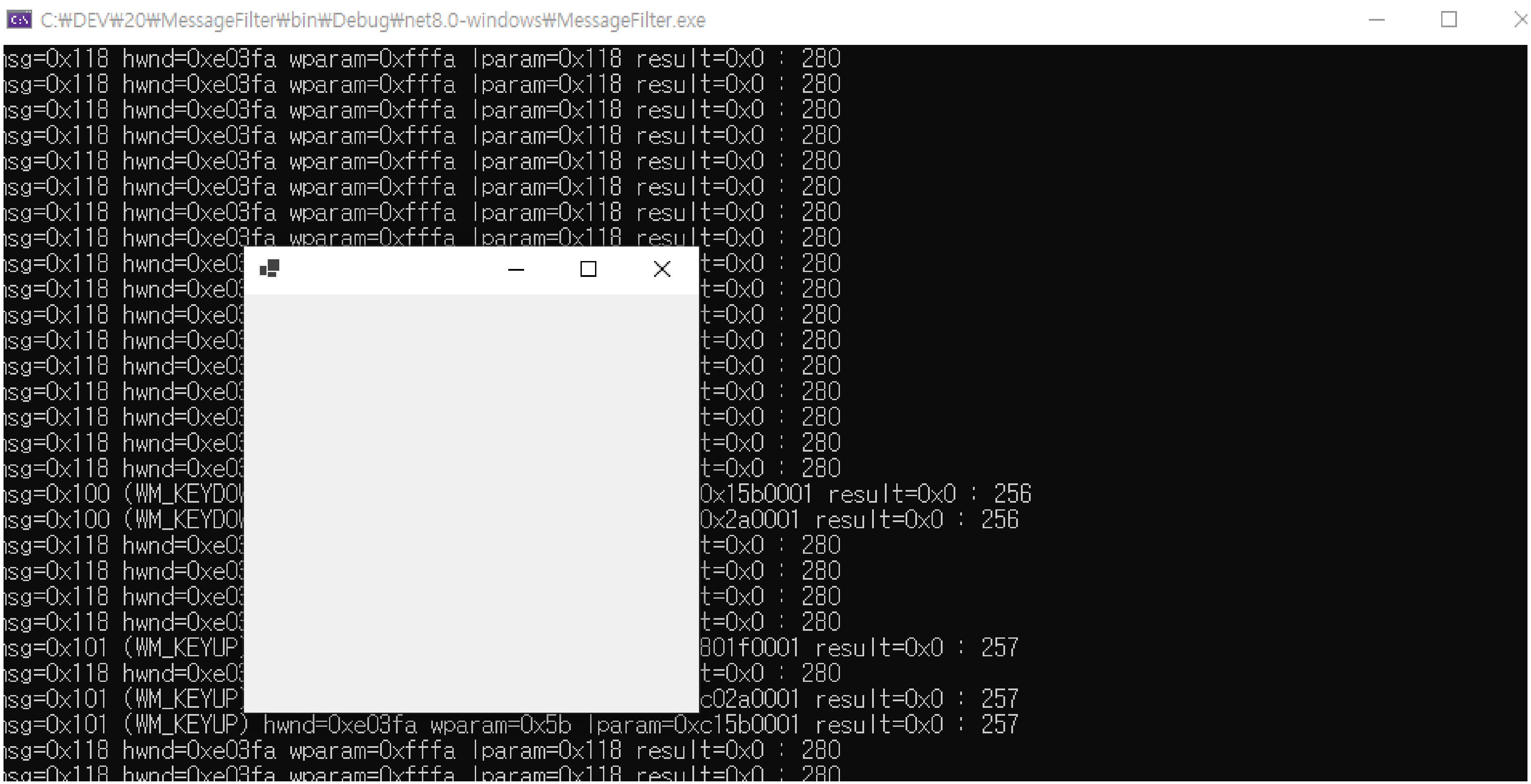
MessageFilter

MainApp.cs

```
using System;
using System.Windows.Forms;

namespace MessageFilter
{
    class MessageFilter : IMessageFilter
    {
        public bool PreFilterMessage(ref Message m)
        {
            if (m.Msg == 0x0F || m.Msg == 0xA0 ||
                m.Msg == 0x200 || m.Msg == 0x113)
                return false;

            Console.WriteLine($"{m.ToString()} : {m.Msg}");

            if (m.Msg == 0x201)
                Application.Exit();

            return true;
        }
    }

    class MainApp : Form
    {
        static void Main(string[] args)
        {
            Application.AddMessageFilter(new MessageFilter());
            Application.Run(new MainApp());
        }
    }
}
```

MessageFilter

실행 결과



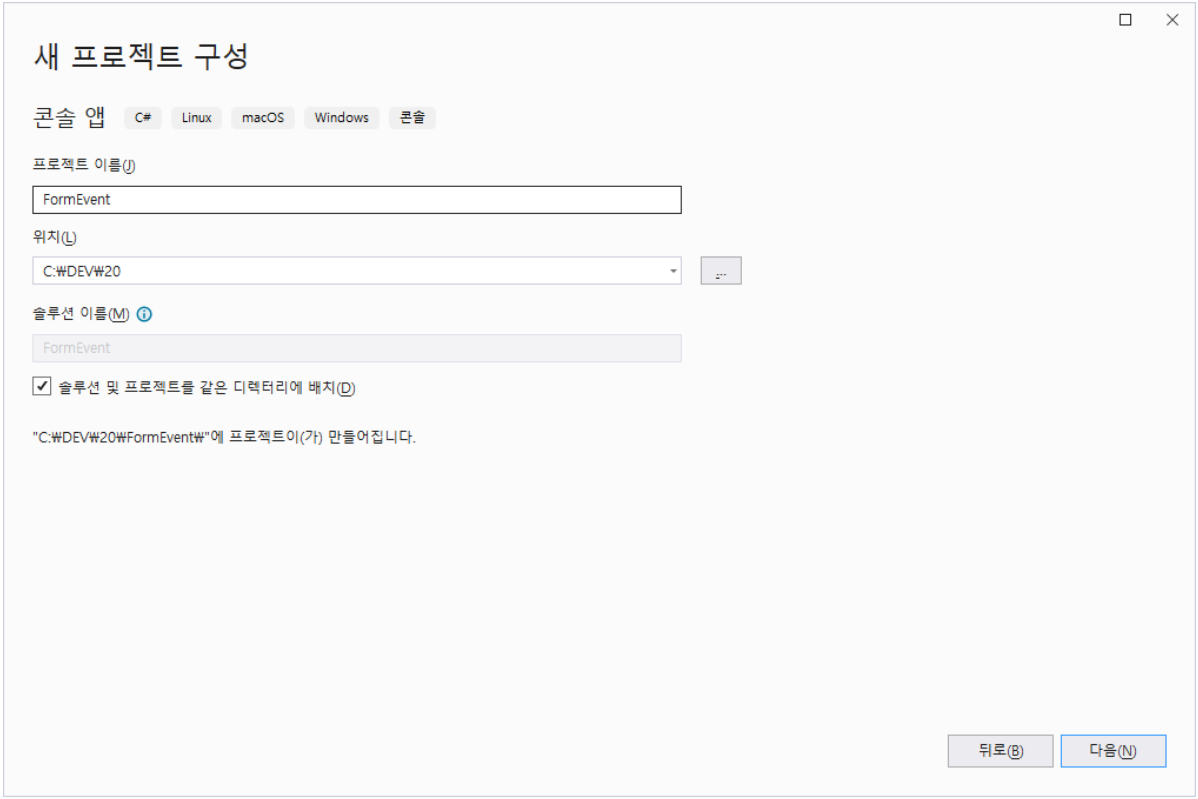
System.Windows.Forms.Form 클래스

- 다양한 컨트롤을 올려 사용할 수 있는 도화지
- 다양한 윈도우 메시지에 대응하는 이벤트 정의
  - 이벤트에 대한 처리 코드 구현이 간단해짐

```
class MyForm : Form
{
   //이벤트 처리기 선언
   private void Form_MouseDown(object sender, System.Windows.Forms.MouseEventArgs e)
   {
      MessageBox.Show("안녕하세요!");
   }

   public MyForm()
   {
      //이벤트 처리기를 이벤트에 연결
      this.MouseDown += new System.Windows.Forms.MouseEventHandler (this.Form_MouseDown);
   }
}
```

Form에 정의된이벤트-FormEvent

새 프로젝트 구성



Form에 정의된이벤트-FormEvent

FormEvent.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

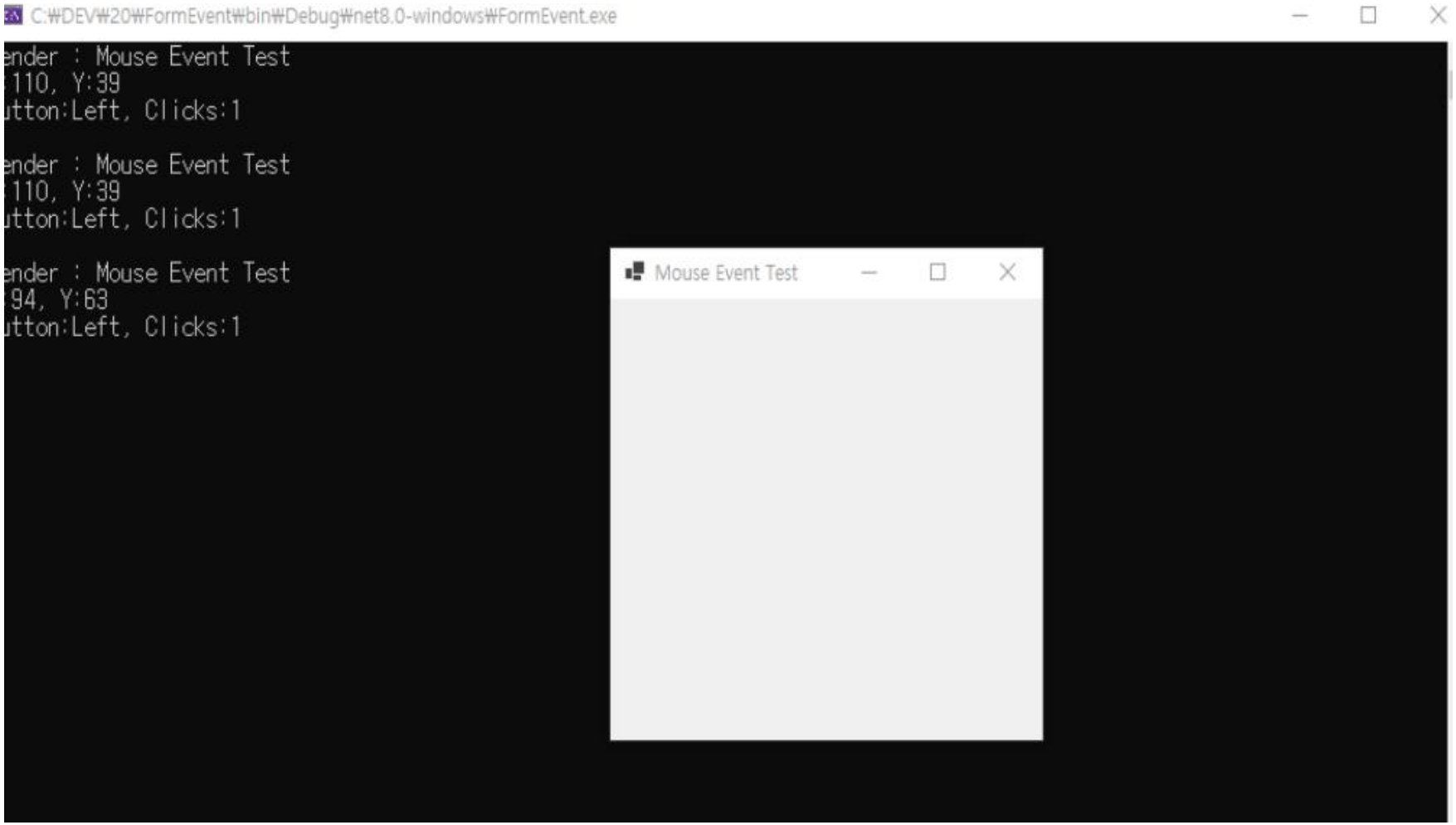
Form에 정의된이벤트-FormEvent

MainApp.cs

```
using System;
using System.Windows.Forms;

namespace FormEvent
{
    class MainApp : Form
    {
        public void MyMouseHandler(object sender, MouseEventArgs e)
        {
            Console.WriteLine($"Sender : {((Form)sender).Text}");
            Console.WriteLine($"X:{e.X}, Y:{e.Y}");
            Console.WriteLine($"Button:{e.Button}, Clicks:{e.Clicks}");
            Console.WriteLine();
        }

        public MainApp(string title)
        {
            this.Text = title;
            this.MouseDown +=
                new MouseEventHandler(MyMouseHandler);
        }

        static void Main(string[] args)
        {
            Application.Run(new MainApp("Mouse Event Test"));
        }
    }
}
```

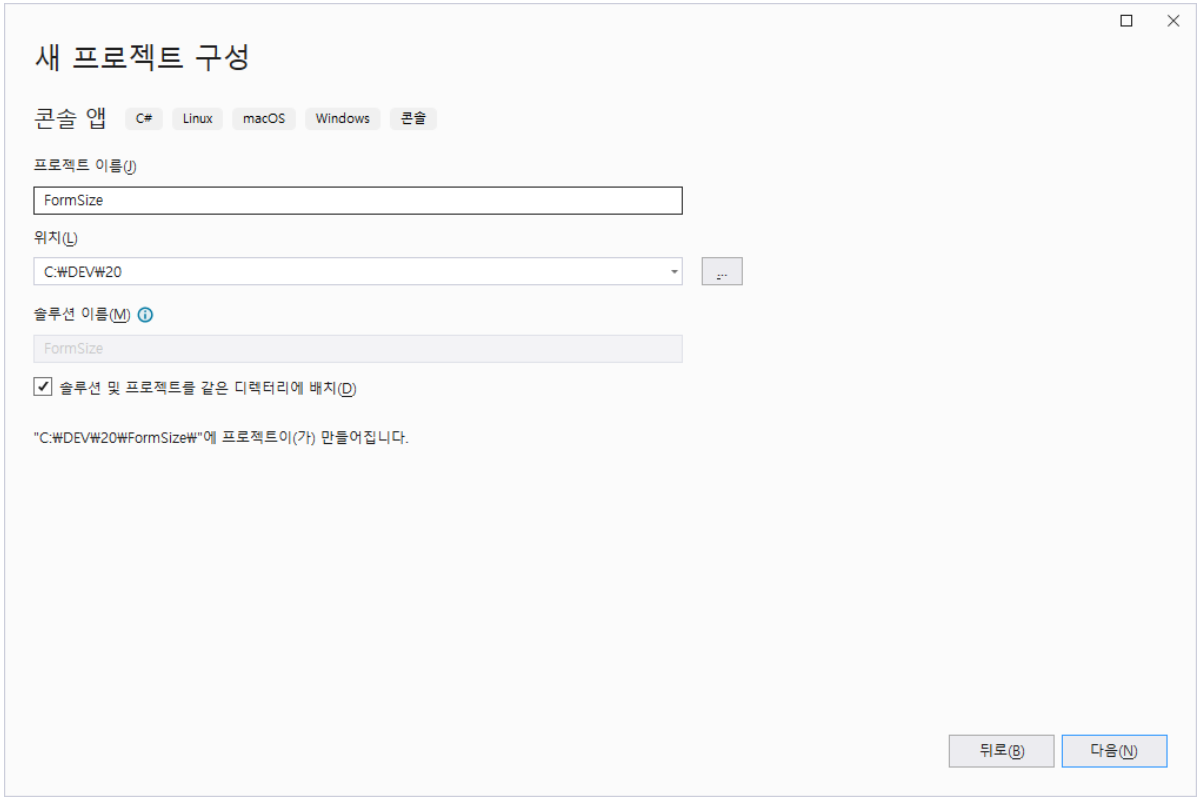
Form에 정의된이벤트-FormEvent

실행 결과



윈도우 모양 바꾸기-FormSize

새 프로젝트 구성



윈도우 모양 바꾸기-FormSize

FormSize.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

윈도우 모양 바꾸기-FormSize

MainApp.cs

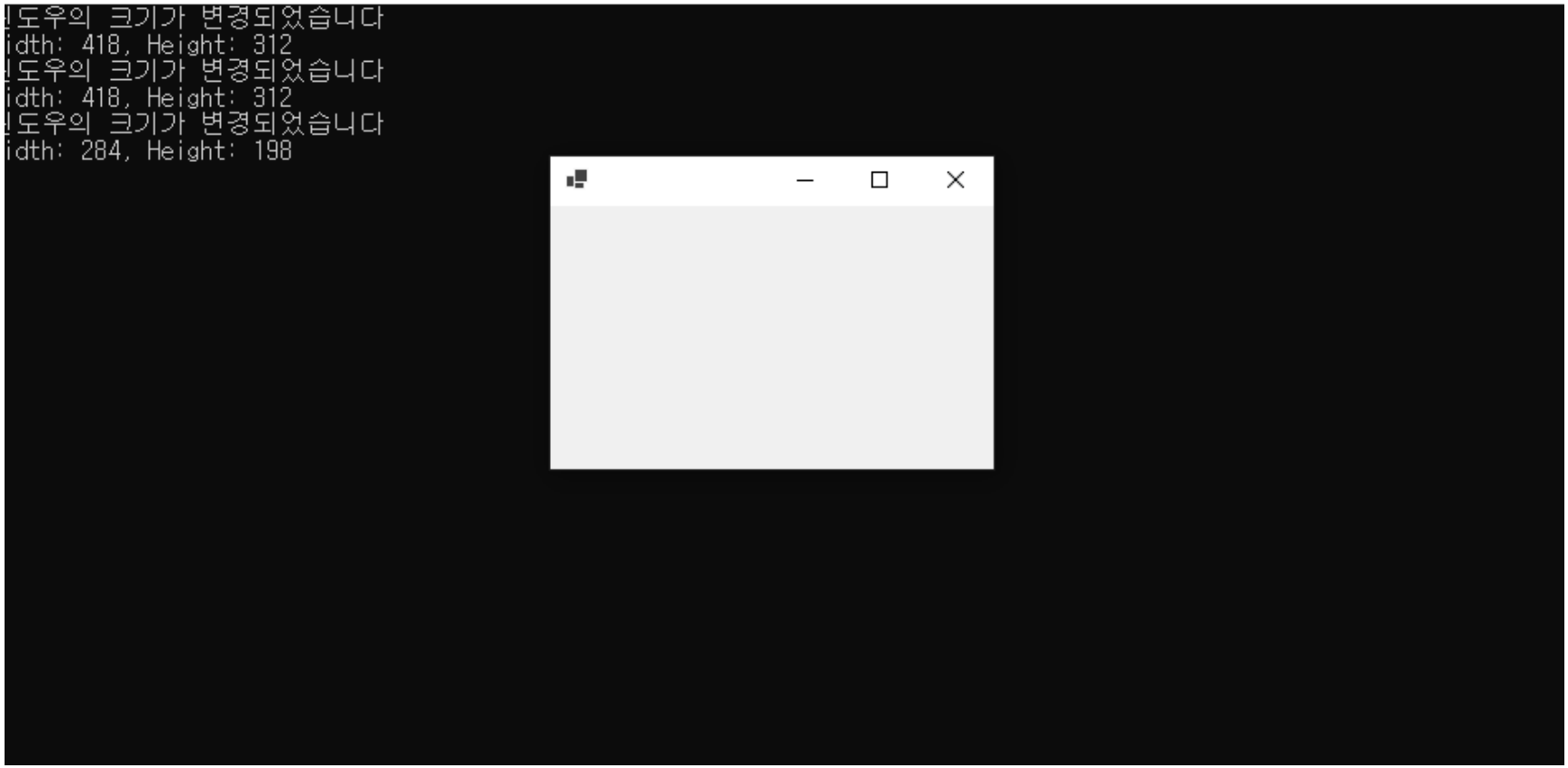
```
using System;
using System.Windows.Forms;

namespace FormSize
{
    class MainApp : Form
    {
        static void Main(string[] args)
        {
            MainApp form = new MainApp();
            form.Width = 300;
            form.Height = 200;

            form.MouseDown += new MouseEventHandler(form_MouseDown);

            Application.Run(form);
        }
```

윈도우 모양 바꾸기-FormSize

```
    static void form_MouseDown(object sender, MouseEventArgs e)
    {
        Form form = (Form)sender;
        int oldWidth  = form.Width;
        int oldHeight = form.Height;

        if (e.Button == MouseButtons.Left)
        {
            if (oldWidth < oldHeight)
            {
                form.Width  = oldHeight;
                form.Height = oldWidth;
            }
        }
        else if (e.Button == MouseButtons.Right)
        {
            if (oldHeight < oldWidth)
            {
                form.Width = oldHeight;
                form.Height = oldWidth;
            }
        }
        Console.WriteLine("윈도우의 크기가 변경되었습니다");
        Console.WriteLine($"Width: {form.Width}, Height: {form.Height}");
    }
  }
}
```

윈도우 모양 바꾸기-FormSize

실행 결과

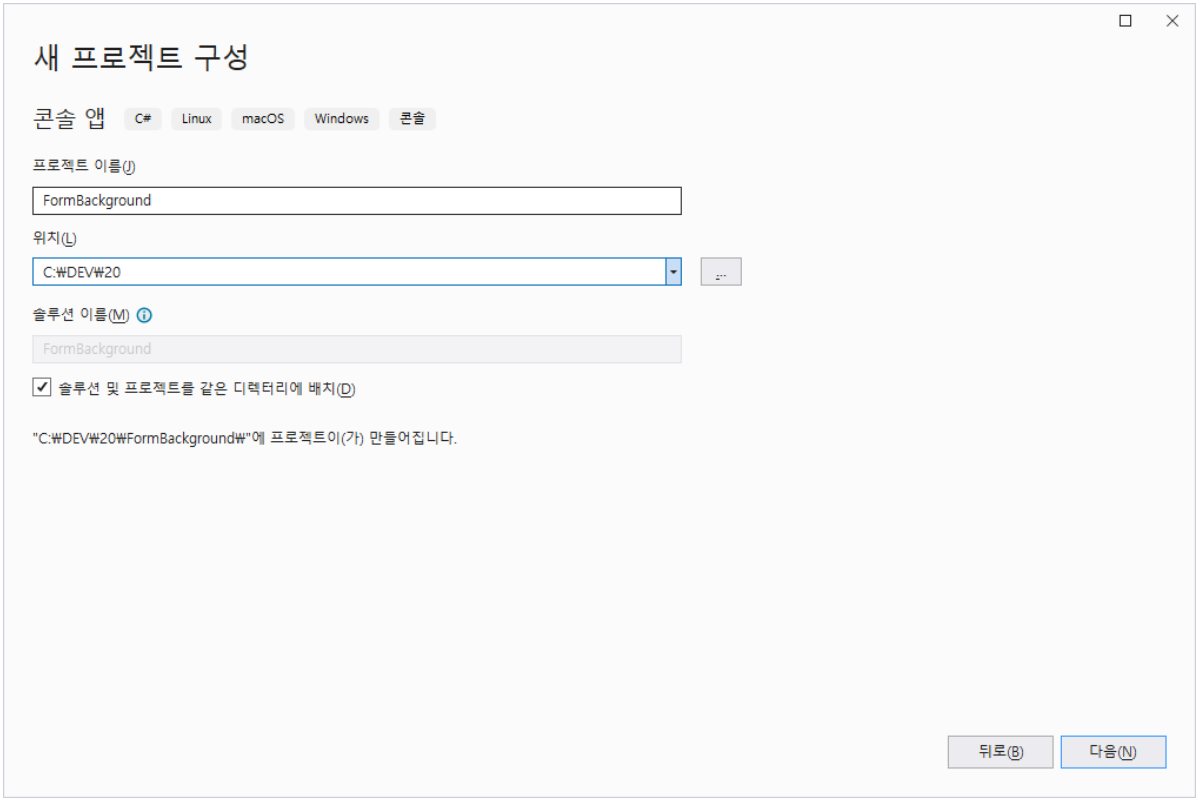


배경 바꾸기-FormBackground

Cosmos.jpg- 그림 하나 준비

배경 바꾸기-FormBackground

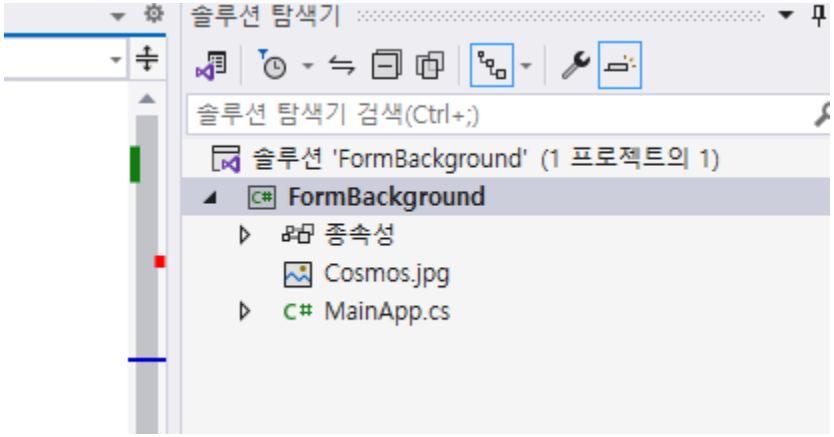
새 프로젝트 구성

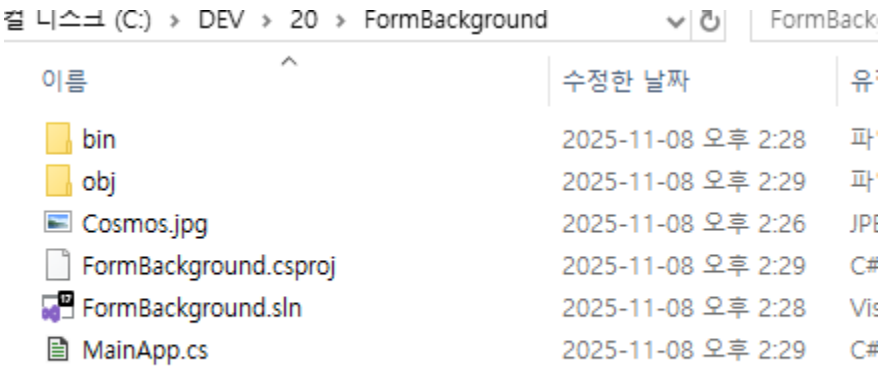


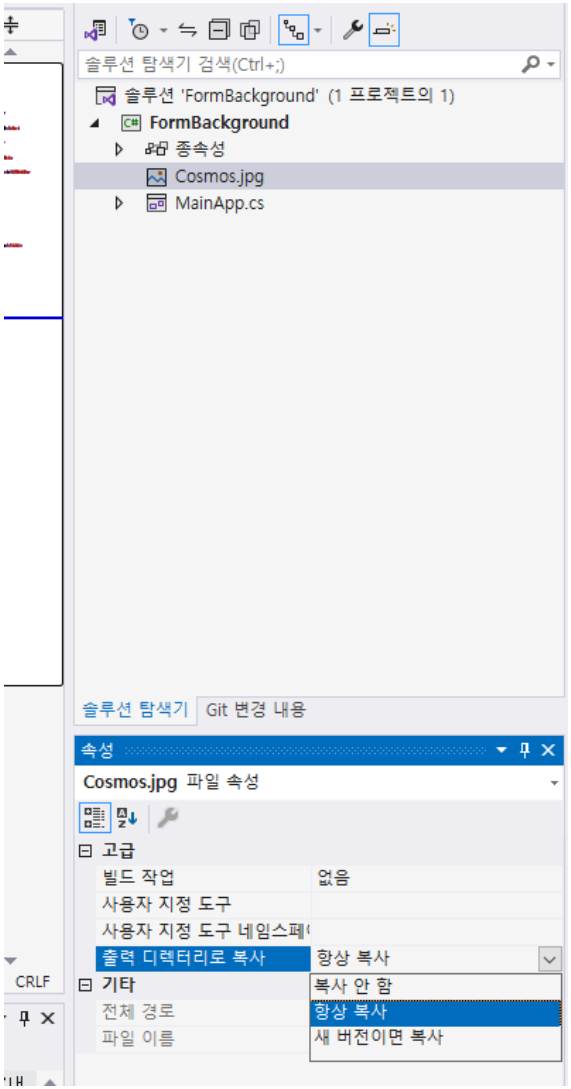
5. 윈도우를 표현하는 Form 클래스

배경 바꾸기-FormBackground

Cosmos.jpg- 속성 변경







배경 바꾸기-FormBackground

FormBackground.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
		<DisableWinExeOutputInference>true</DisableWinExeOutputInference>
	</PropertyGroup>

	<ItemGroup>
	  <None Update="Cosmos.jpg">
	    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
	  </None>
	</ItemGroup>

</Project>
```

배경 바꾸기-FormBackground

MainApp.cs

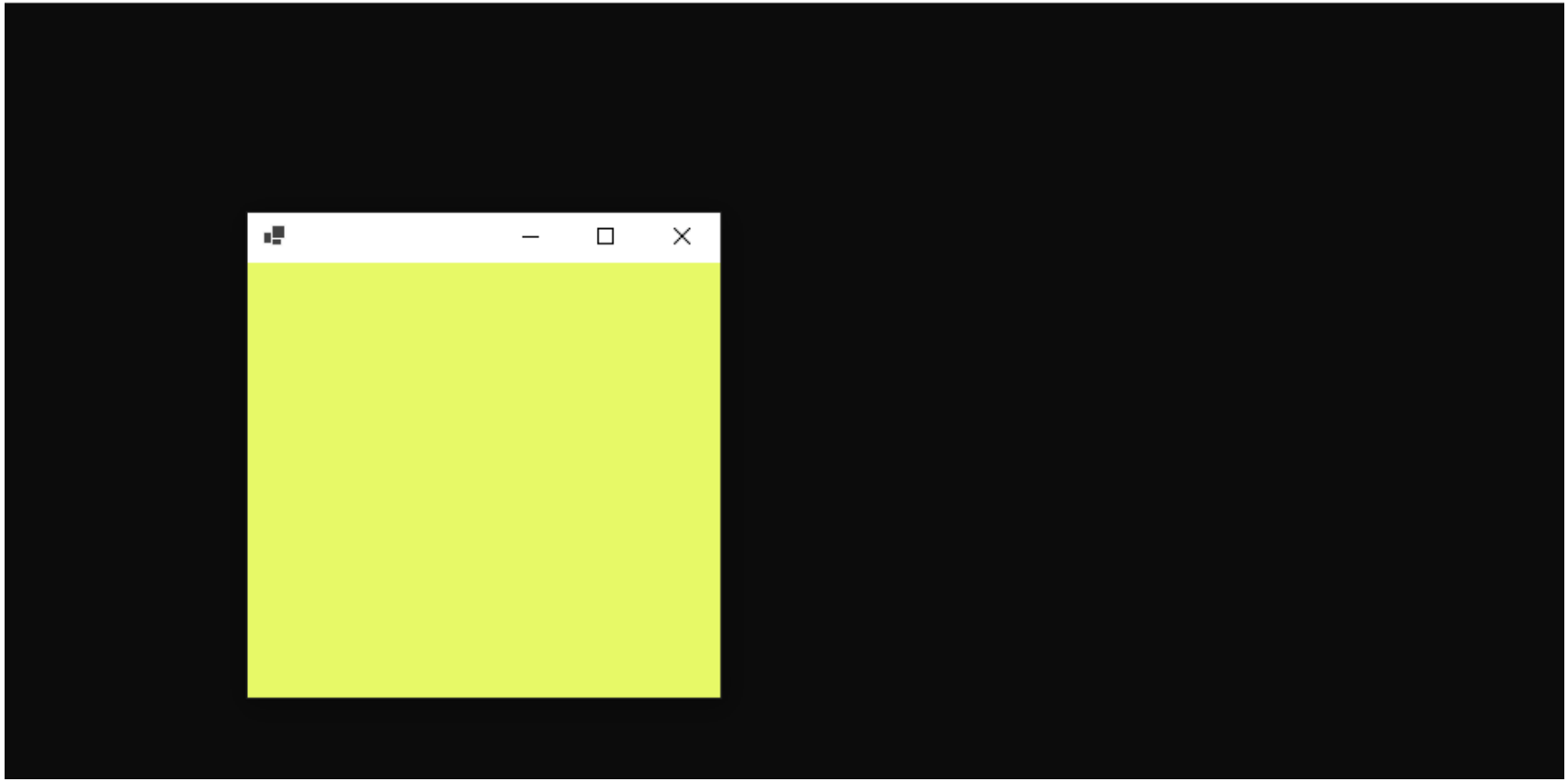
```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace FormBackground
{
    class MainApp : Form
    {
        Random rand;
        public MainApp()
        {
            rand = new Random();

            this.MouseWheel += new MouseEventHandler(MainApp_MouseWheel);
            this.MouseDown += new MouseEventHandler(MainApp_MouseDown);
        }
```

배경 바꾸기-FormBackground

```
    void MainApp_MouseDown(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {
            Color oldColor = this.BackColor;
            this.BackColor = Color.FromArgb(rand.Next(0, 255),
                                            rand.Next(0, 255),
                                            rand.Next(0, 255));
        }
        else if (e.Button == MouseButtons.Right)
        {
            if (this.BackgroundImage != null)
            {
                this.BackgroundImage = null;
                return;
            }

            string file = "Cosmos.jpg";
            if (System.IO.File.Exists(file) == false)
                MessageBox.Show("이미지 파일이 없습니다.");
            else
                this.BackgroundImage = Image.FromFile(file);
        }
    }

    void MainApp_MouseWheel(object sender, MouseEventArgs e)
    {
        this.Opacity = this.Opacity + ( e.Delta>0?0.1:-0.1 );
        Console.WriteLine($"Opacity: {this.Opacity}");
    }

    static void Main(string[] args)
    {
        Application.Run(new MainApp());
    }
  }
}
```

배경 바꾸기-FormBackground

실행 결과



배경 이미지 채우기-FormBackground

MainApp.cs

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace SimpleBackground
{
    public class MainApp : Form
    {
        Image background;

        public MainApp()
        {
            // Cosmos.jpg 불러오기
            string file = "Cosmos.jpg";
            if (!System.IO.File.Exists(file))
            {
                MessageBox.Show("이미지 파일을 찾을 수 없습니다: " + System.IO.Path.GetFullPath(file));
                Environment.Exit(0);
            }

            background = Image.FromFile(file);

            // 폼 크기를 이미지 크기에 맞게 설정
            this.ClientSize = background.Size;
            this.Text = "Original Size Background Example";
            this.DoubleBuffered = true; // 깜빡임 방지
        }
```

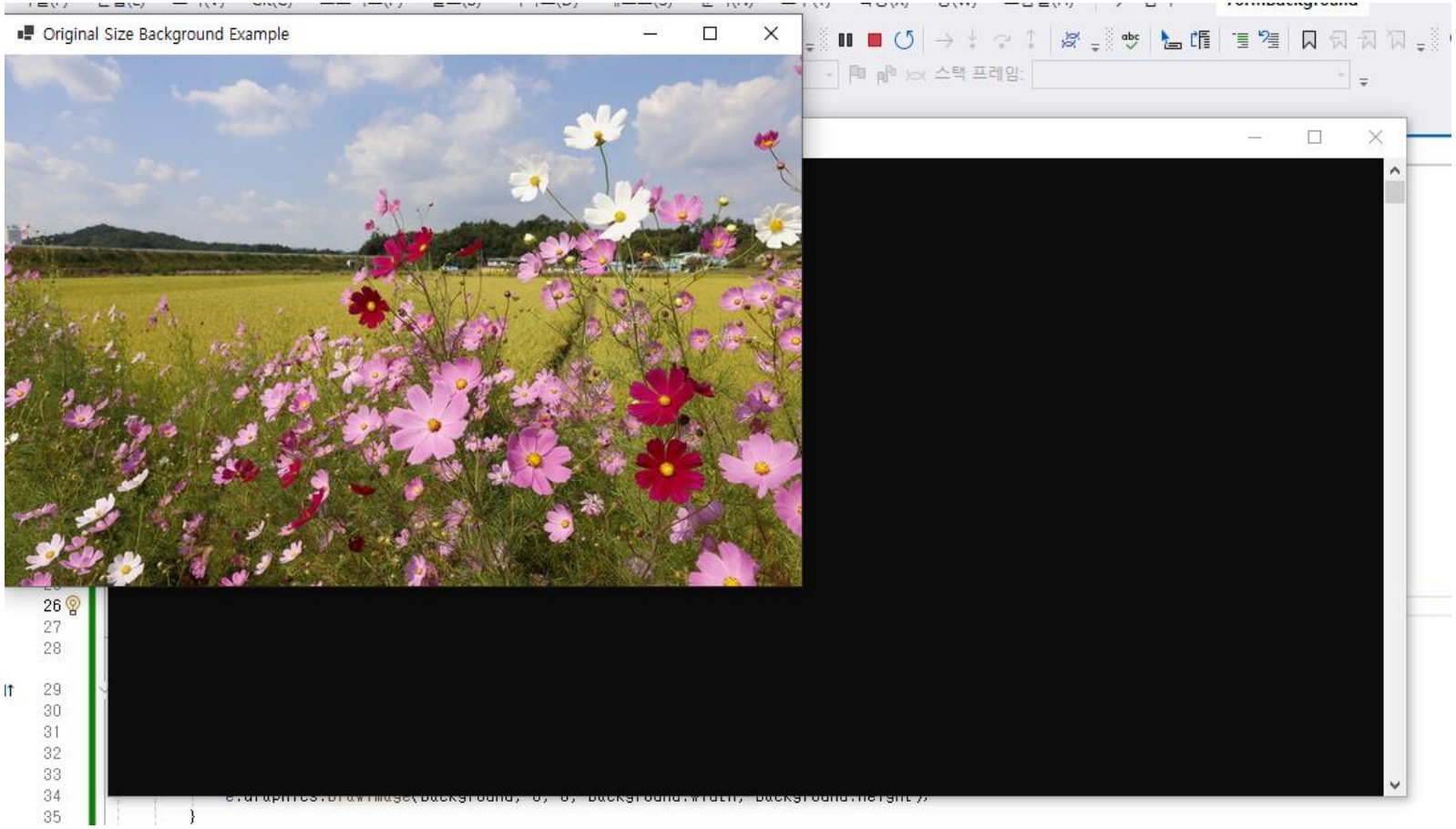
배경 이미지 채우기-FormBackground

MainApp.cs

```
        protected override void OnPaint(PaintEventArgs e)
        {
            base.OnPaint(e);

            // 이미지 원본 크기 그대로 (좌측 상단 기준) 그리기
            e.Graphics.DrawImage(background, 0, 0, background.Width, background.Height);
        }

        [STAThread]
        static void Main()
        {
            Application.Run(new MainApp());
        }
    }
}
```

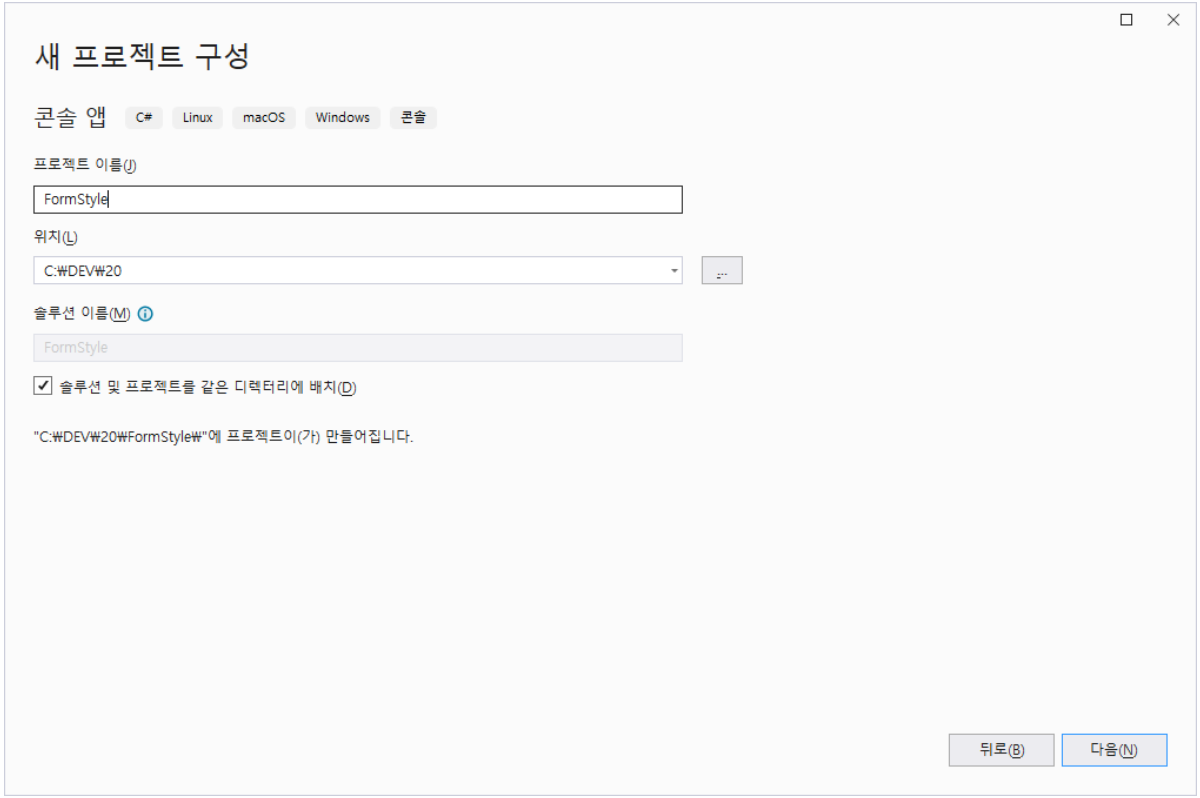
배경 바꾸기-FormBackground

실행 결과



최소화/최대화버튼-FormStyle

새 프로젝트 구성



최소화/최대화버튼-FormStyle

FormStyle.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

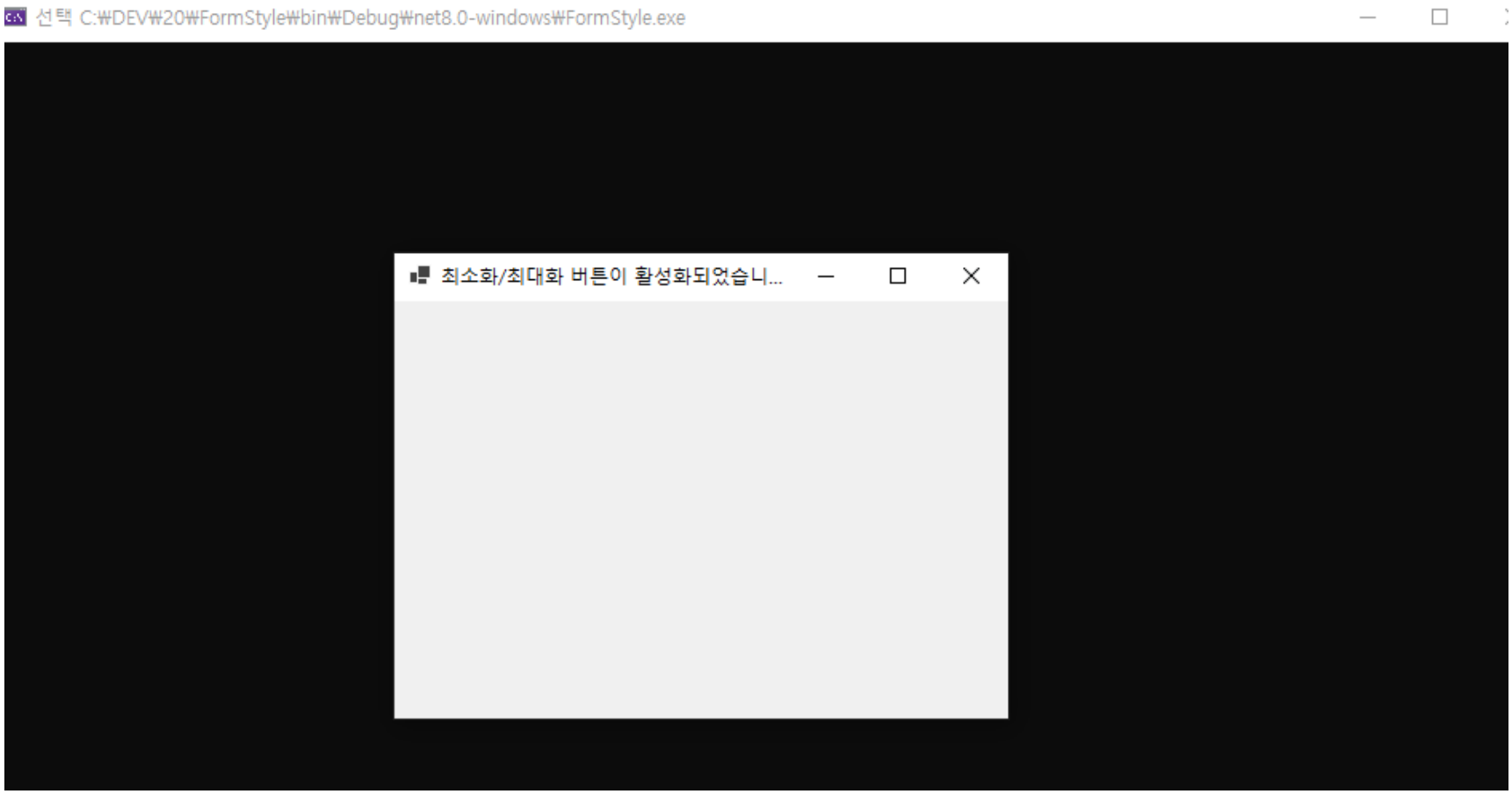
최소화/최대화버튼-FormStyle

MainApp.cs

```
using System;
using System.Windows.Forms;

namespace FormStyle
{
   class MainApp : Form
   {
      static void Main(string[] args)
      {
         MainApp form = new MainApp();

         form.Width = 400;
         form.MouseDown += new MouseEventHandler(form_MouseDown);

         Application.Run(form);
      }

      static void form_MouseDown(object sender, MouseEventArgs e)
      {
         Form form = (Form)sender;

         if (e.Button == MouseButtons.Left)
         {
            form.MaximizeBox = true;
            form.MinimizeBox = true;
            form.Text = "최소화/최대화 버튼이 활성화되었습니다.";
         }
         else if (e.Button == MouseButtons.Right)
         {
            form.MaximizeBox = false;
            form.MinimizeBox = false;
            form.Text = "최소화/최대화 버튼이 비활성화되었습니다.";
         }
      }
   }
}
```

최소화/최대화버튼-FormStyle

실행 결과



Form위에 컨트롤 올리기

컨트롤 (1/2)

- 윈도우 운영체제가 제공하는 사용자 인터페이스 요소
- 메뉴, 콤보박스, 리스트뷰, 버튼, 텍스트박스 등과 같은 표준 컨트롤 제공
- 컨트롤은 다음과 같은 절차로 Form 위에 배치
  1. 컨트롤의 인스턴스 생성
  2. 컨트롤의 프로퍼티에 값 지정
  3. 컨트롤의 이벤트에 이벤트 처리기 등록
  4. 폼에 컨트롤 추가

Form위에 컨트롤 올리기

컨트롤 (2/2)

1. 컨트롤의 인스턴스 생성	`Button button = new Button();`

2. 컨트롤의 프로퍼티에 값 지정	`button.Text = "Click Me!";`
	`button.Left = 100;`
	`button.Top = 50;`

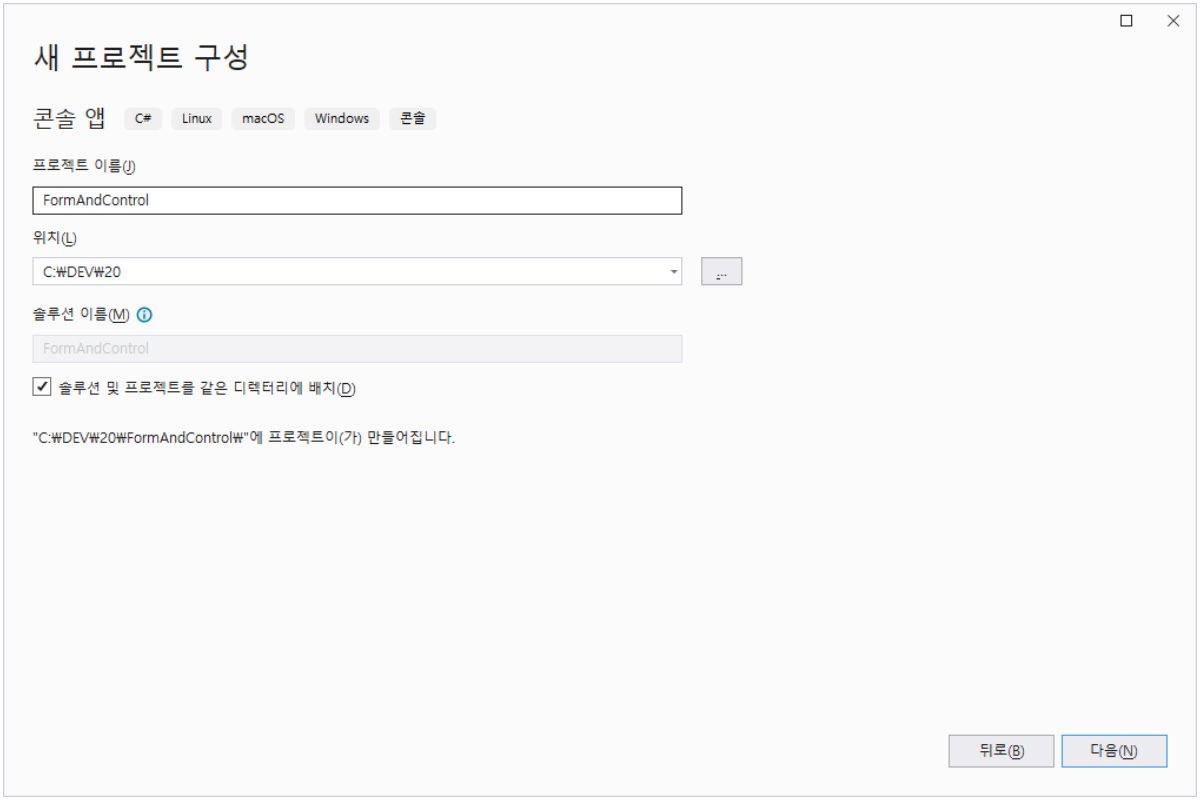
3. 컨트롤의 이벤트에 이벤트 처리기 등록	`button.Click +=`
	`(object sender, EventArgs e) =>`
	`{`
	`MessageBox.Show("딸깍!");`
	`};`

4. 폼에 컨트롤 추가	`MainApp form = new MainApp();`
	
	`form.Controls.Add(button);`

5. 윈도우를 표현하는 Form 클래스

Form위에 컨트롤 올리기- FormAndControl

새 프로젝트 구성



Form위에 컨트롤 올리기- FormAndControl

FormAndControl.csproj 편집

```
<Project Sdk="Microsoft.NET.Sdk">

	<PropertyGroup>
		<OutputType>Exe</OutputType>
		<TargetFramework>net8.0-windows</TargetFramework>
		<Nullable>enable</Nullable>
		<UseWindowsForms>true</UseWindowsForms>
	</PropertyGroup>

</Project>
```

Form위에 컨트롤 올리기- FormAndControl

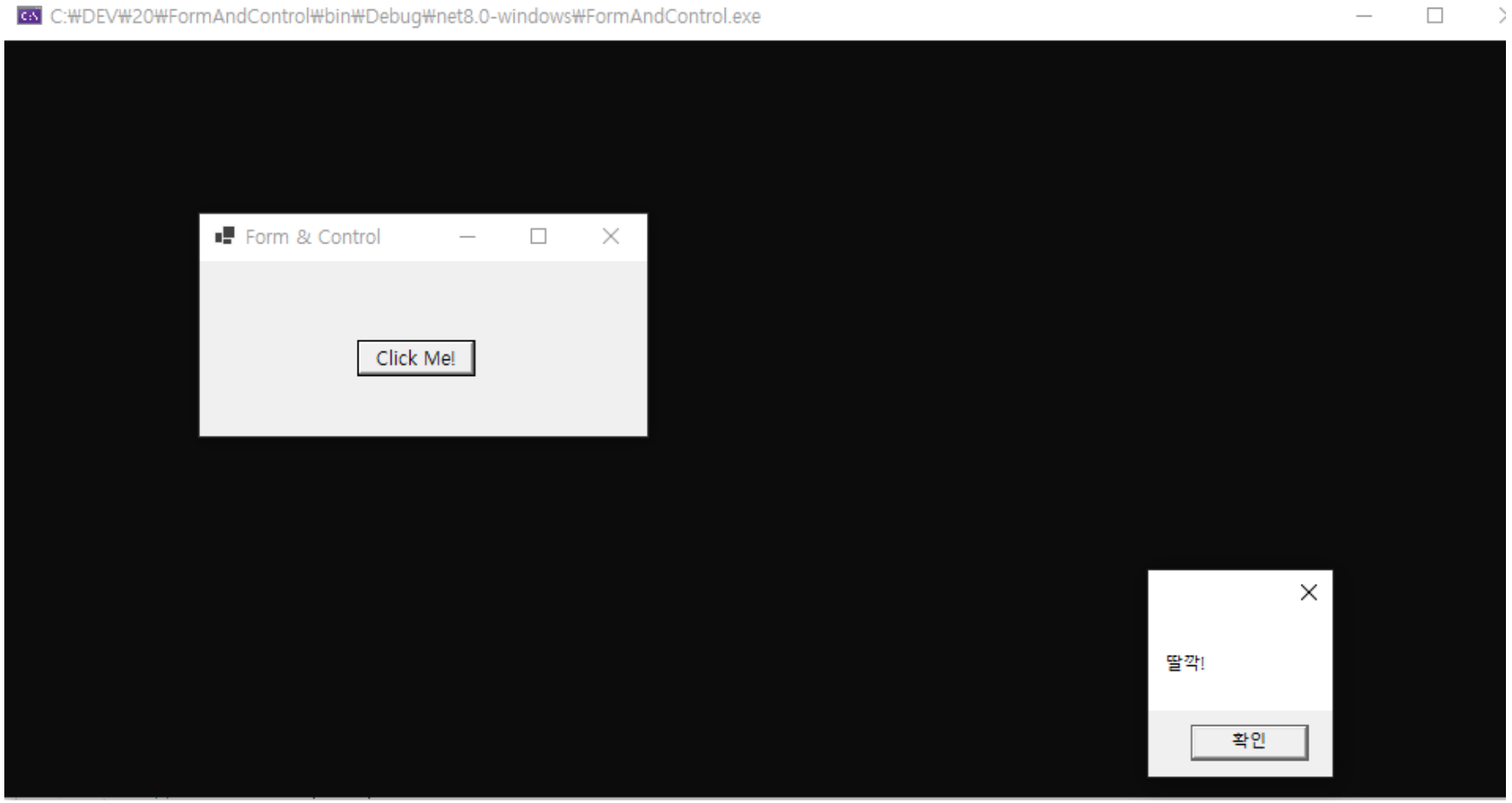
MainApp.cs

```
using System;
using System.Windows.Forms;

namespace FormAndControl
{
   class MainApp : Form
   {
      static void Main(string[] args)
      {
         Button button = new Button();

         button.Text = "Click Me!";
         button.Left = 100;
         button.Top = 50;

         button.Click +=
            (object sender, EventArgs e) =>
            {
               MessageBox.Show("딸깍!");
            };

         MainApp form = new MainApp();
         form.Text = "Form & Control";
         form.Height = 150;

         form.Controls.Add(button);

         Application.Run(form);
      }
   }
}
```

5. 윈도우를 표현하는 Form 클래스

Form위에 컨트롤 올리기- FormAndControl

실행 결과



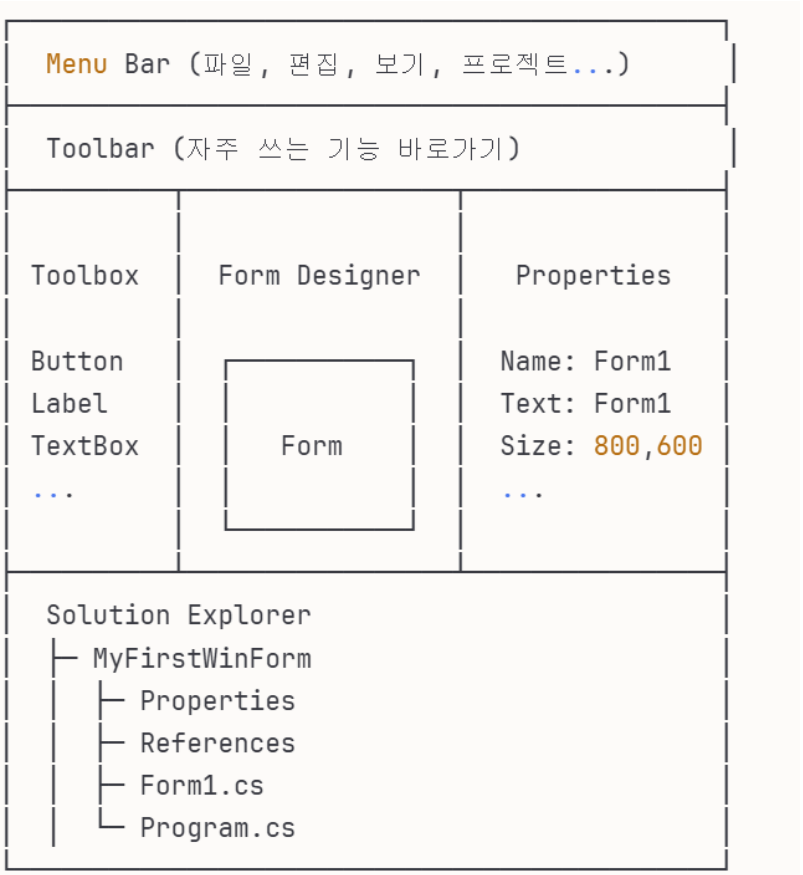
WinForm 디자이너 인터페이스

Toolbox (도구 상자)
- 사용 가능한 컨트롤 목록
- 드래그 앤 드롭으로 폼에 추가

Form Designer (폼 디자이너)
- 시각적으로 UI 디자인
- 컨트롤 배치 및 크기 조절

Properties (속성 창)
- 선택한 컨트롤의 속성 편집
- 이벤트 핸들러 연결

Solution Explorer (솔루션 탐색기)
- 프로젝트 파일 구조
- 파일 관리



Button 컨트롤

버튼은 사용자의 클릭을 받아 동작을 수행합니다.

기본 사용법

```
Button button1 = new Button();
button1.Text = "클릭하세요";
button1.Location = new Point(50, 50);
button1.Size = new Size(100, 30);
button1.Click += Button1_Click;

this.Controls.Add(button1);

private void Button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("버튼이 클릭되었습니다!");
}
```

Button 컨트롤

Button 주요 속성

```
Button btn = new Button
{
    Text = "확인",                          // 버튼 텍스트
    Size = new Size(100, 40),               // 크기
    Location = new Point(50, 50),           // 위치
    BackColor = Color.LightBlue,            // 배경색
    ForeColor = Color.White,                // 글자색
    Font = new Font("맑은 고딕", 12, FontStyle.Bold),
    FlatStyle = FlatStyle.Flat,             // 스타일
    Image = Properties.Resources.Icon,      // 이미지
    ImageAlign = ContentAlignment.MiddleLeft,   // 이미지 정렬
    TextAlign = ContentAlignment.MiddleRight,   // 텍스트 정렬
    Cursor = Cursors.Hand,                  // 커서 모양
    TabIndex = 0,                           // 탭 순서
    Enabled = true,                         // 활성화 여부
    Visible = true                          // 표시 여부
};
```

Button 컨트롤

Button 이벤트

```
// 클릭 이벤트
btn.Click += (s, e) => MessageBox.Show("클릭!");

// 마우스 이벤트
btn.MouseEnter += (s, e) => btn.BackColor = Color.Yellow;
btn.MouseLeave += (s, e) => btn.BackColor = Color.LightBlue;
btn.MouseDown += (s, e) => Console.WriteLine("마우스 다운");
btn.MouseUp += (s, e) => Console.WriteLine("마우스 업");

// 키보드 이벤트
btn.KeyDown += (s, e) =>
{
    if (e.KeyCode == Keys.Enter)
    {
        MessageBox.Show("Enter 키 감지");
    }
};
```

Label 컨트롤

텍스트를 표시하는 컨트롤입니다.

```
Label label1 = new Label
{
    Text = "사용자 이름:",
    Location = new Point(20, 20),
    Size = new Size(100, 20),
    Font = new Font("맑은 고딕", 10),
    ForeColor = Color.Black,
    BackColor = Color.Transparent,
    TextAlign = ContentAlignment.MiddleRight,  // 텍스트 정렬
    AutoSize = true,                           // 자동 크기 조절
    BorderStyle = BorderStyle.FixedSingle      // 테두리
};
```

TextBox 컨트롤

사용자 입력을 받는 컨트롤입니다.

기본 TextBox

```
TextBox txtName = new TextBox
{
    Location = new Point(120, 20),
    Size = new Size(200, 20),
    Font = new Font("맑은 고딕", 10),
    MaxLength = 50,                    // 최대 길이
    PlaceholderText = "이름을 입력하세요"  // .NET 6 이상
};

// 텍스트 변경 이벤트
txtName.TextChanged += (s, e) =>
{
    Console.WriteLine($"현재 텍스트: {txtName.Text}");
};
```

CheckBox 컨트롤

체크박스는 여러 옵션을 독립적으로 선택할 수 있습니다.

기본 사용법

```
CheckBox chkBold = new CheckBox
{
    Text = "굵게",
    Location = new Point(20, 20),
    AutoSize = true,
    Checked = false  // 초기 상태
};

// 체크 상태 변경 이벤트
chkBold.CheckedChanged += (s, e) =>
{
    Console.WriteLine($"굵게: {chkBold.Checked}");
};
```

CheckBox 컨트롤

3가지 상태 CheckBox

```
CheckBox chkTriState = new CheckBox
{
    Text = "3-State 체크박스",
    Location = new Point(20, 50),
    ThreeState = true,  // 3가지 상태 활성화
    CheckState = CheckState.Indeterminate  // 초기 상태: 불확정
};

chkTriState.CheckStateChanged += (s, e) =>
{
    switch (chkTriState.CheckState)
    {
        case CheckState.Unchecked:
            Console.WriteLine("체크 안됨");
            break;
        case CheckState.Checked:
            Console.WriteLine("체크됨");
            break;
        case CheckState.Indeterminate:
            Console.WriteLine("불확정 상태");
            break;
    }
};
```

RadioButton 컨트롤

라디오버튼은 그룹 내에서 하나만 선택할 수 있습니다.

기본 사용법

```
// 라디오 버튼 그룹
RadioButton rbSmall = new RadioButton
{
    Text = "작게",
    Location = new Point(20, 20),
    Checked = true  // 기본 선택
};

RadioButton rbMedium = new RadioButton
{
    Text = "보통",
    Location = new Point(20, 45)
};

RadioButton rbLarge = new RadioButton
{
    Text = "크게",
    Location = new Point(20, 70)
};
```

RadioButton 컨트롤

```
// 같은 컨테이너(폼 또는 GroupBox)에 있으면 자동으로 그룹화됨
this.Controls.AddRange(new Control[] { rbSmall, rbMedium, rbLarge });

// 선택 변경 이벤트
rbSmall.CheckedChanged += OnSizeChanged;
rbMedium.CheckedChanged += OnSizeChanged;
rbLarge.CheckedChanged += OnSizeChanged;

void OnSizeChanged(object sender, EventArgs e)
{
    RadioButton rb = sender as RadioButton;
    if (rb.Checked)
    {
        Console.WriteLine($"{rb.Text} 선택됨");
    }
}
```

ComboBox 컨트롤

드롭다운 목록에서 항목을 선택합니다.

기본 사용법

```
ComboBox cmbCountry = new ComboBox
{
    Location = new Point(120, 20),
    Size = new Size(200, 25),
    DropDownStyle = ComboBoxStyle.DropDownList  // 편집 불가
};

// 항목 추가 방법 1: 개별 추가
cmbCountry.Items.Add("대한민국");
cmbCountry.Items.Add("미국");
cmbCountry.Items.Add("일본");

// 항목 추가 방법 2: 배열로 추가
string[] countries = { "중국", "영국", "프랑스" };
cmbCountry.Items.AddRange(countries);

// 기본 선택
cmbCountry.SelectedIndex = 0;  // 첫 번째 항목
```

ComboBox 컨트롤

```
// 선택 변경 이벤트
cmbCountry.SelectedIndexChanged += (s, e) =>
{
    if (cmbCountry.SelectedIndex >= 0)
    {
        string selected = cmbCountry.SelectedItem.ToString();
        Console.WriteLine($"선택: {selected}");
    }
};
```

ComboBox 컨트롤

ComboBox 스타일

```
// 1. DropDown (기본) - 편집 가능, 드롭다운 가능
ComboBox cmbDropDown = new ComboBox
{
    DropDownStyle = ComboBoxStyle.DropDown
};

// 2. DropDownList - 편집 불가, 드롭다운만 가능 (권장)
ComboBox cmbDropDownList = new ComboBox
{
    DropDownStyle = ComboBoxStyle.DropDownList
};

// 3. Simple - 항상 목록 표시, 편집 가능
ComboBox cmbSimple = new ComboBox
{
    DropDownStyle = ComboBoxStyle.Simple,
    Size = new Size(200, 100)  // 높이가 목록 크기
};
```

ListBox 컨트롤

여러 항목을 목록으로 표시하고 선택할 수 있습니다.

기본 사용법

```
ListBox lstItems = new ListBox
{
    Location = new Point(20, 20),
    Size = new Size(200, 150),
    SelectionMode = SelectionMode.One  // 단일 선택
};

// 항목 추가
lstItems.Items.Add("항목 1");
lstItems.Items.Add("항목 2");
lstItems.Items.Add("항목 3");

// 선택 이벤트
lstItems.SelectedIndexChanged += (s, e) =>
{
    if (lstItems.SelectedIndex >= 0)
    {
        string selected = lstItems.SelectedItem.ToString();
        Console.WriteLine($"선택: {selected}");
    }
};
```

ListView 컨트롤

아이콘, 상세 정보 등 다양한 형태로 항목을 표시합니다.

View 모드

```
ListView listView1 = new ListView
{
    Location = new Point(20, 20),
    Size = new Size(400, 300),
    View = View.Details,  // 상세 보기
    FullRowSelect = true,  // 전체 행 선택
    GridLines = true       // 그리드 라인 표시
};

// 열 추가
listView1.Columns.Add("이름", 100);
listView1.Columns.Add("나이", 50);
listView1.Columns.Add("이메일", 150);

// 항목 추가
ListViewItem item1 = new ListViewItem("홍길동");
item1.SubItems.Add("25");
item1.SubItems.Add("hong@example.com");
listView1.Items.Add(item1);

// 또는
ListViewItem item2 = new ListViewItem(new[] { "김철수", "30", "kim@example.com" });
listView1.Items.Add(item2);
```

ListView 컨트롤

View 모드 전환

```
// LargeIcon: 큰 아이콘
listView1.View = View.LargeIcon;

// SmallIcon: 작은 아이콘
listView1.View = View.SmallIcon;

// List: 목록
listView1.View = View.List;

// Details: 상세 보기 (가장 많이 사용)
listView1.View = View.Details;

// Tile: 타일 (Windows Vista 이상)
listView1.View = View.Tile;
```

DataGridView 컨트롤

표 형식으로 데이터를 표시하고 편집합니다.

기본 사용법

```
DataGridView dgvData = new DataGridView
{
    Location = new Point(20, 20),
    Size = new Size(600, 300),
    AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill,
    AllowUserToAddRows = false,  // 새 행 추가 금지
    AllowUserToDeleteRows = false,  // 행 삭제 금지
    ReadOnly = true  // 읽기 전용
};

// 열 추가
dgvData.Columns.Add("Name", "이름");
dgvData.Columns.Add("Age", "나이");
dgvData.Columns.Add("Email", "이메일");

// 행 추가
dgvData.Rows.Add("홍길동", 25, "hong@example.com");
dgvData.Rows.Add("김철수", 30, "kim@example.com");
```

컨테이너 컨트롤

Panel

```
Panel panel1 = new Panel
{
    Location = new Point(20, 20),
    Size = new Size(300, 200),
    BorderStyle = BorderStyle.FixedSingle,
    AutoScroll = true  // 스크롤바 자동 표시
};

// Panel에 컨트롤 추가
Button btn1 = new Button { Text = "버튼 1", Location = new Point(10, 10) };
Button btn2 = new Button { Text = "버튼 2", Location = new Point(10, 50) };

panel1.Controls.AddRange(new Control[] { btn1, btn2 });
this.Controls.Add(panel1);
```

컨테이너 컨트롤

GroupBox

```
GroupBox grpSettings = new GroupBox
{
    Text = "설정",
    Location = new Point(20, 20),
    Size = new Size(200, 150)
};

RadioButton rb1 = new RadioButton
{
    Text = "옵션 1",
    Location = new Point(20, 30),
    Checked = true
};

RadioButton rb2 = new RadioButton
{
    Text = "옵션 2",
    Location = new Point(20, 60)
};

grpSettings.Controls.AddRange(new Control[] { rb1, rb2 });
this.Controls.Add(grpSettings);
```

컨테이너 컨트롤

TabControl

```
TabControl tabControl1 = new TabControl
{
    Location = new Point(20, 20),
    Size = new Size(500, 350)
};

// 탭 페이지 추가
TabPage tab1 = new TabPage("일반");
TabPage tab2 = new TabPage("고급");
TabPage tab3 = new TabPage("정보");

// 각 탭에 컨트롤 추가
tab1.Controls.Add(new Label
{
    Text = "일반 설정",
    Location = new Point(20, 20)
});

tab2.Controls.Add(new Label
{
    Text = "고급 설정",
    Location = new Point(20, 20)
});

tabControl1.TabPages.AddRange(new TabPage[] { tab1, tab2, tab3 });
this.Controls.Add(tabControl1);

// 탭 변경 이벤트
tabControl1.SelectedIndexChanged += (s, e) =>
{
    Console.WriteLine($"현재 탭: {tabControl1.SelectedTab.Text}");
};
```

컨테이너 컨트롤

SplitContainer

```
SplitContainer splitContainer1 = new SplitContainer
{
    Dock = DockStyle.Fill,
    Orientation = Orientation.Vertical,  // 세로 분할
    SplitterDistance = 200  // 분할 위치
};

// 왼쪽 패널
TreeView treeView1 = new TreeView { Dock = DockStyle.Fill };
splitContainer1.Panel1.Controls.Add(treeView1);

// 오른쪽 패널
TextBox textBox1 = new TextBox
{
    Multiline = true,
    Dock = DockStyle.Fill
};
splitContainer1.Panel2.Controls.Add(textBox1);

this.Controls.Add(splitContainer1);
```

대화상자 (Dialog)

MessageBox

```
// 기본 메시지
MessageBox.Show("저장되었습니다.");

// 제목과 아이콘
MessageBox.Show("파일을 찾을 수 없습니다.", "오류",
    MessageBoxButtons.OK, MessageBoxIcon.Error);

// 버튼 선택
DialogResult result = MessageBox.Show("정말 삭제하시겠습니까?", "확인",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)
{
    // 삭제 수행
}

// 다양한 버튼 조합
MessageBox.Show("저장하시겠습니까?", "저장",
    MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

// MessageBoxButtons 종류:
// - OK
// - OKCancel
// - YesNo
// - YesNoCancel
// - RetryCancel
// - AbortRetryIgnore

// MessageBoxIcon 종류:
// - None
// - Information
// - Warning
// - Error
// - Question
```

대화상자 (Dialog)

OpenFileDialog

```
Button btnOpen = new Button { Text = "파일 열기" };
btnOpen.Click += (s, e) =>
{
    OpenFileDialog ofd = new OpenFileDialog
    {
        Title = "파일 선택",
        Filter = "텍스트 파일|*.txt|모든 파일|*.*",
        FilterIndex = 1,  // 기본 선택 필터
        InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments),
        Multiselect = false  // 다중 선택
    };

    if (ofd.ShowDialog() == DialogResult.OK)
    {
        string fileName = ofd.FileName;
        string content = System.IO.File.ReadAllText(fileName);
        MessageBox.Show($"파일: {fileName}\n내용: {content}");
    }
};

// 다중 선택
OpenFileDialog ofdMulti = new OpenFileDialog { Multiselect = true };
if (ofdMulti.ShowDialog() == DialogResult.OK)
{
    foreach (string file in ofdMulti.FileNames)
    {
        Console.WriteLine(file);
    }
}
```

대화상자 (Dialog)

SaveFileDialog

```
Button btnSave = new Button { Text = "파일 저장" };
btnSave.Click += (s, e) =>
{
    SaveFileDialog sfd = new SaveFileDialog
    {
        Title = "파일 저장",
        Filter = "텍스트 파일|*.txt|모든 파일|*.*",
        DefaultExt = "txt",  // 기본 확장자
        FileName = "untitled.txt",  // 기본 파일명
        OverwritePrompt = true  // 덮어쓰기 확인
    };

    if (sfd.ShowDialog() == DialogResult.OK)
    {
        string content = "저장할 내용";
        System.IO.File.WriteAllText(sfd.FileName, content);
        MessageBox.Show($"파일이 저장되었습니다:\n{sfd.FileName}");
    }
};
```

대화상자 (Dialog)

FolderBrowserDialog

```
Button btnFolder = new Button { Text = "폴더 선택" };
btnFolder.Click += (s, e) =>
{
    FolderBrowserDialog fbd = new FolderBrowserDialog
    {
        Description = "저장할 폴더를 선택하세요.",
        RootFolder = Environment.SpecialFolder.MyComputer,
        ShowNewFolderButton = true
    };

    if (fbd.ShowDialog() == DialogResult.OK)
    {
        string folderPath = fbd.SelectedPath;
        MessageBox.Show($"선택한 폴더: {folderPath}");
    }
};
```

대화상자 (Dialog)

ColorDialog

```
Button btnColor = new Button { Text = "색상 선택" };
btnColor.Click += (s, e) =>
{
    ColorDialog cd = new ColorDialog
    {
        AllowFullOpen = true,  // 사용자 정의 색상 허용
        FullOpen = true,  // 대화상자 완전히 열기
        Color = Color.Blue  // 초기 색상
    };

    if (cd.ShowDialog() == DialogResult.OK)
    {
        btnColor.BackColor = cd.Color;
    }
};
```

대화상자 (Dialog)

FontDialog

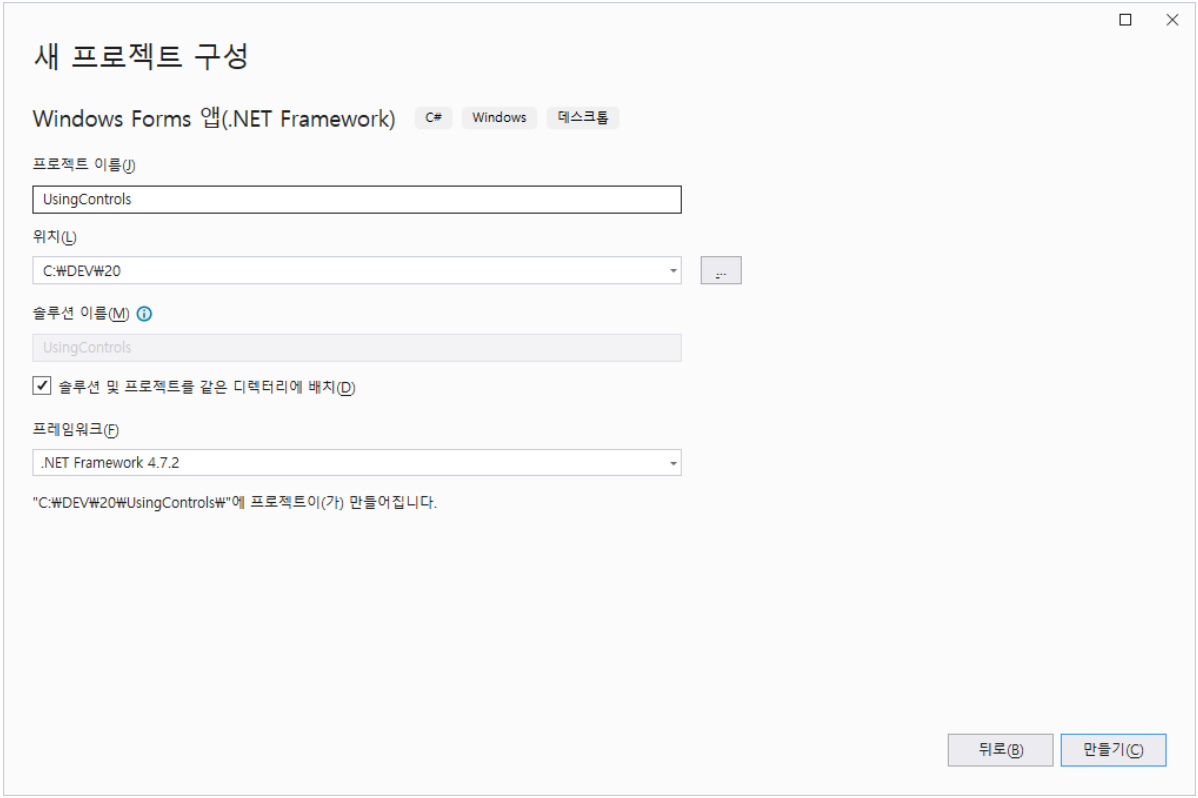
```
TextBox txtEditor = new TextBox { Multiline = true, Dock = DockStyle.Fill };

Button btnFont = new Button { Text = "글꼴 선택" };
btnFont.Click += (s, e) =>
{
    FontDialog fd = new FontDialog
    {
        Font = txtEditor.Font,
        ShowColor = true,  // 색상 선택 표시
        ShowEffects = true  // 효과 옵션 표시
    };

    if (fd.ShowDialog() == DialogResult.OK)
    {
        txtEditor.Font = fd.Font;
        txtEditor.ForeColor = fd.Color;
    }
};
```

새 프로젝트 만들기

- 새 프로젝트 만들기 창에서 C#용 Windows Forms 앱 (.NET Framework) 템플릿을 선택합니다.

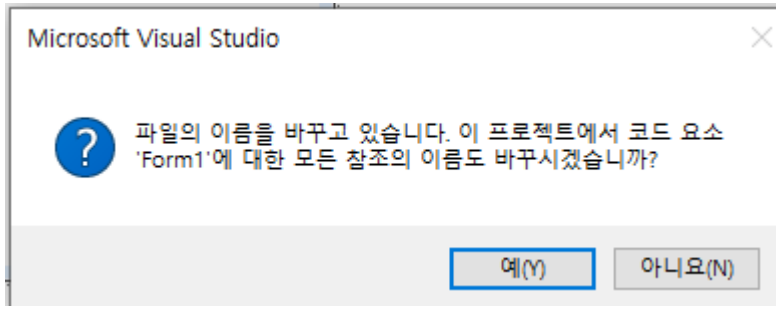
새 프로젝트 만들기

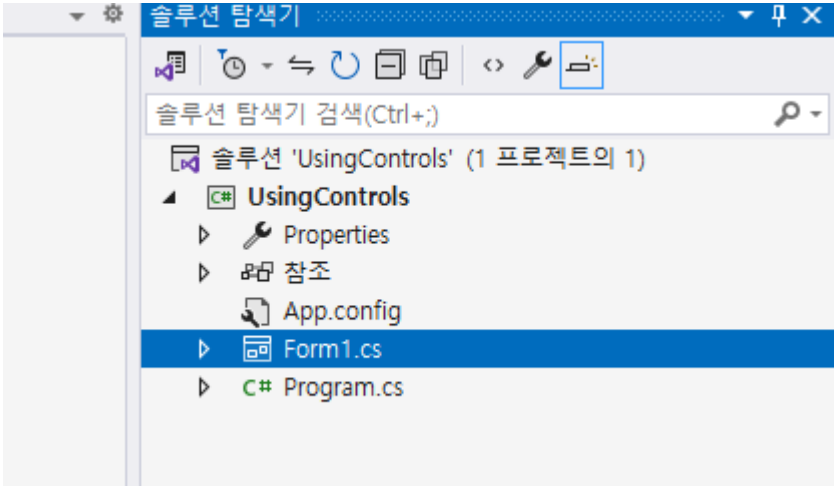
새 프로젝트 구성

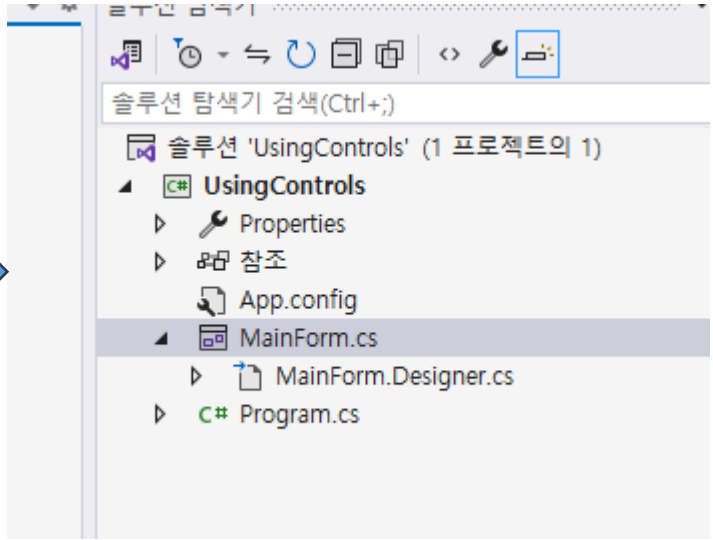


새 프로젝트 만들기

Form1.cs를 MainForm.cs 로 변경



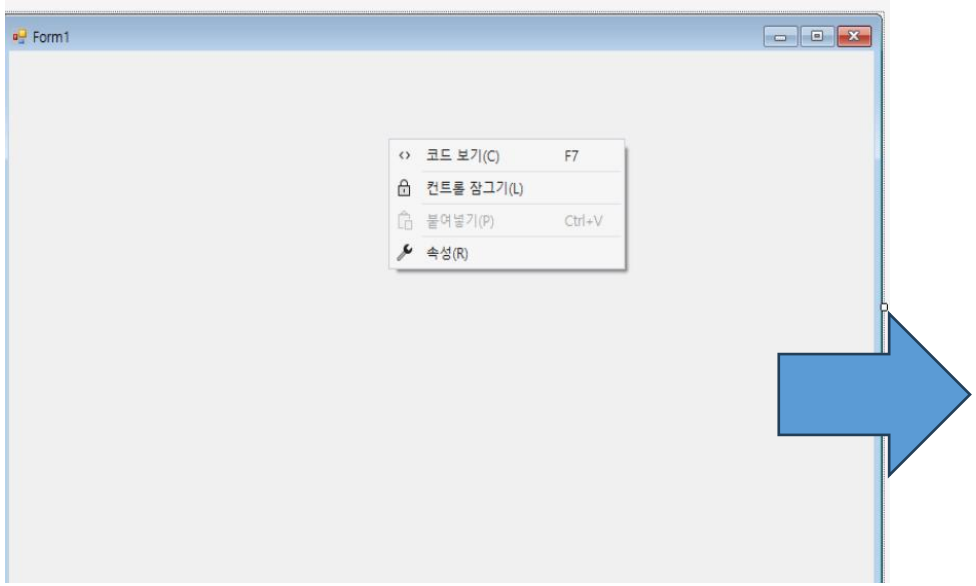


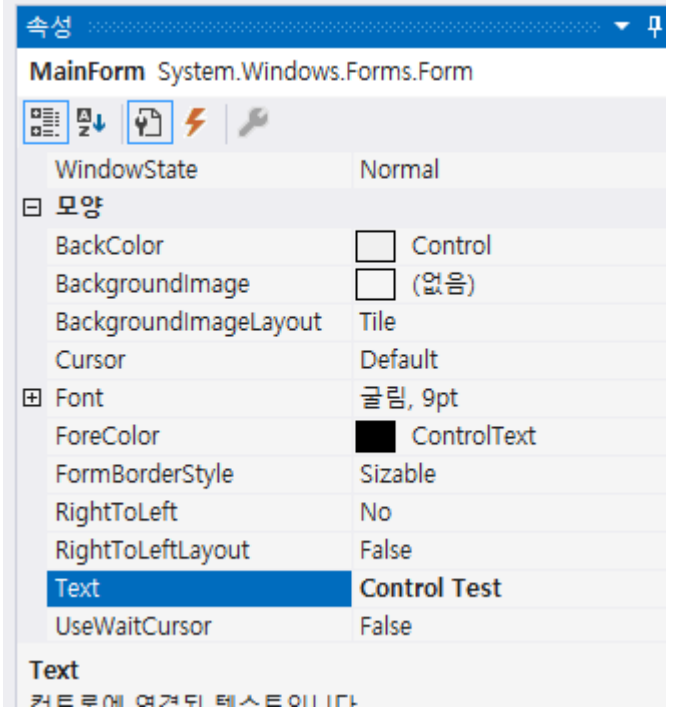


6. 폼 디자이너를 이용한 WinForm UI 구성

GroupBox, Label, ComboBox, CheckBox, TextBox

Form 마우스 우 클릭 > 속성 > Text -> Control Test 로 변경

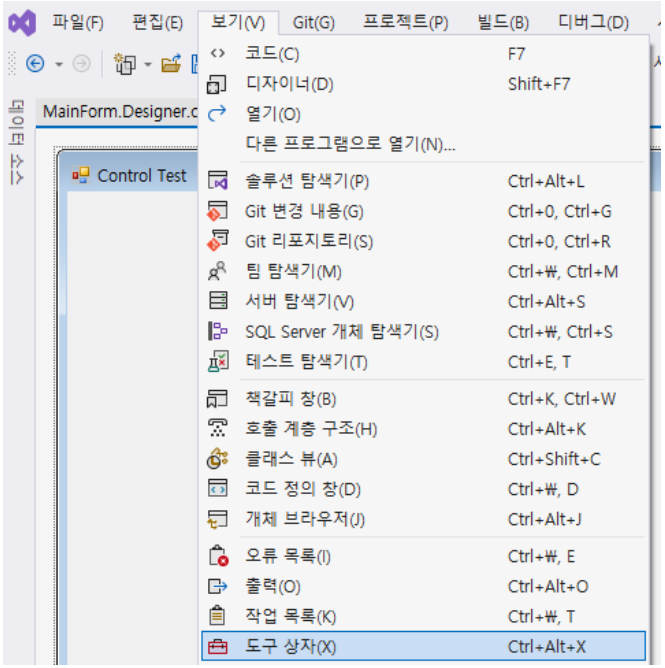




6. 폼 디자이너를 이용한 WinForm UI 구성

GroupBox, Label, ComboBox, CheckBox, TextBox

GroupBox, Label, ComboBox, CheckBox, TextBox 배치 >
먼저, 도구 상자 메뉴 또는 단축키로 열기:
메뉴 모음에서 보기 > 도구 상자를 선택

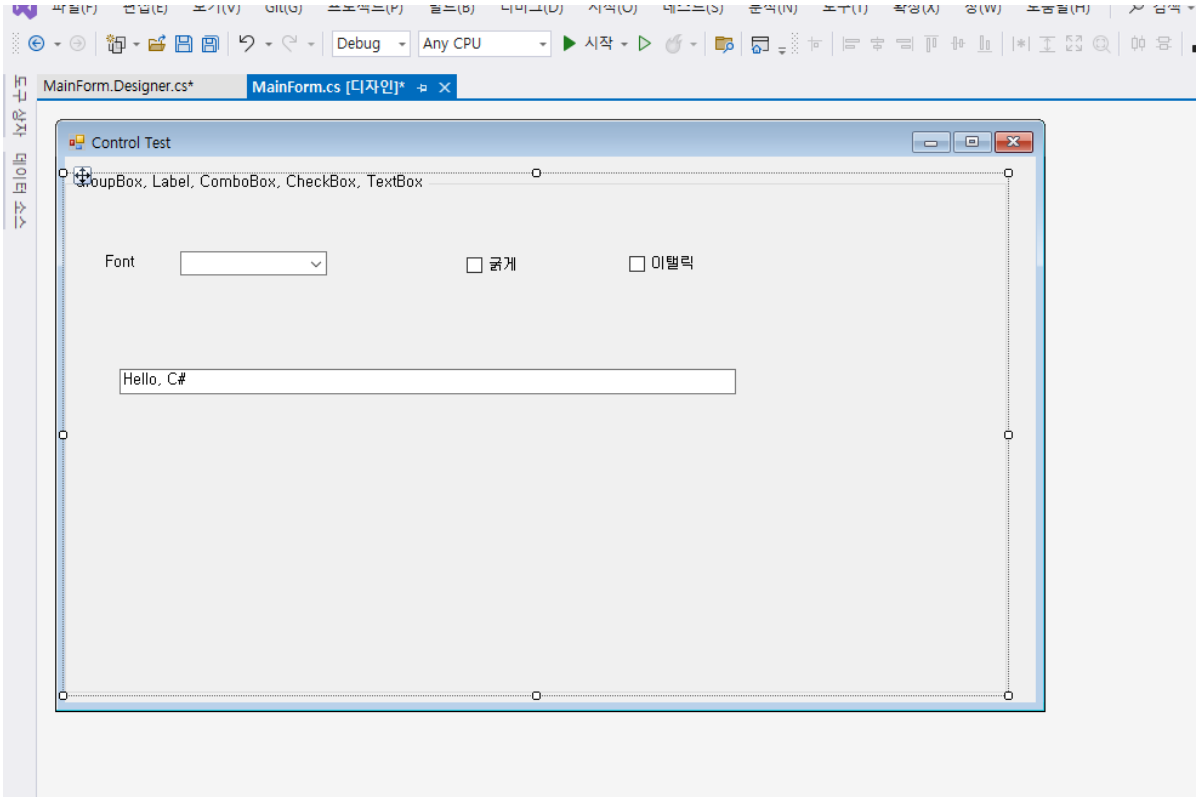


GroupBox, Label, ComboBox, CheckBox, TextBox

폼 이름 : MainForm
주요 컨트롤 그룹 :

GroupBox (grpFont)
 ├── Label (lblFont)
 ├── ComboBox (cboFont)
 ├── CheckBox (chkBold)
 ├── CheckBox (chkItalic)
 └── TextBox (txtSampleText)

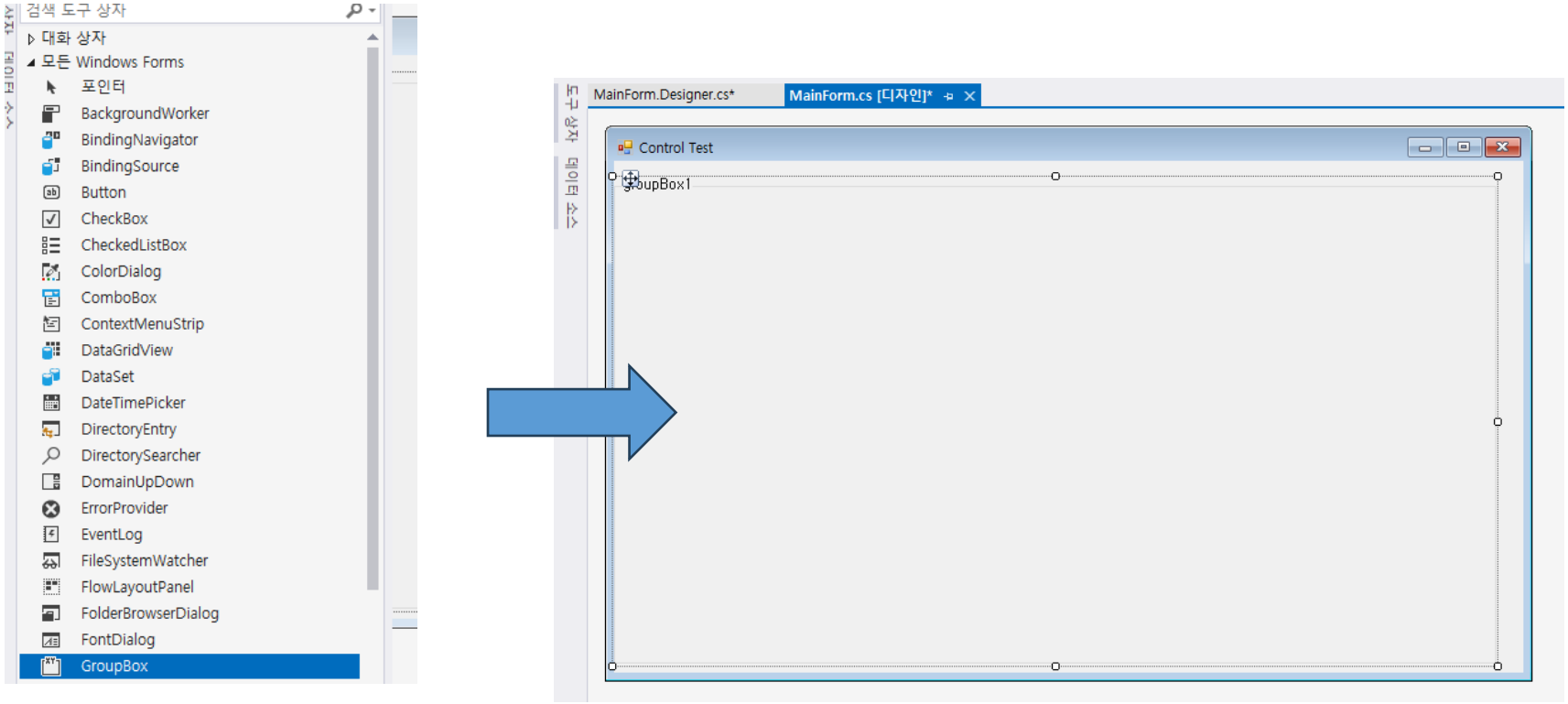
GroupBox, Label, ComboBox, CheckBox, TextBox 컨트롤 배치 완료 예시



6. 폼 디자이너를 이용한 WinForm UI 구성

GroupBox, Label, ComboBox, CheckBox, TextBox

도구 상자 활용: Visual Studio 디자인 화면 왼쪽 상단의 도구 상자에서 원하는 컨트롤(예: Button, Label)을 마우스로 드래그하여 폼에 놓습니다.



GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm (폼 자체) 속성

속성	값	설명
Name	"MainForm"	폼의 내부 이름
Text	"Control Test"	윈도우 제목 표시줄 텍스트

GroupBox (grpFont)

속성	값	설명
Name	"grpFont"	컨트롤 이름
Text	"GroupBox, Label, Combo Box, CheckBox, TextBox"	그룹 제목 (UI에 표시됨)

GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm.Designer.cs을 보면, grpFont 추가됨.

```
private void InitializeComponent()
    {
        this. grpFont = new System.Windows.Forms.GroupBox();
        this.SuspendLayout();
        //
        // grpFont
        //
        this.groupBox1.Location = new System.Drawing.Point(12, 12);
        this.groupBox1.Name = "grpFont";
        this.groupBox1.Size = new System.Drawing.Size(776, 347);
        this.groupBox1.TabIndex = 0;
        this.groupBox1.TabStop = false;
        this.groupBox1.Text = "groupBox1";
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(800, 450);
        this.Controls.Add(this. grpFont);
        this.Name = "MainForm";
        this.Text = "Control Test";
        this.ResumeLayout(false);

    }

    #endregion

private System.Windows.Forms.GroupBox grpFont;
}
```

MainForm에 grpFont가 추가됨

GroupBox, Label, ComboBox, CheckBox, TextBox

Label (lblFont)

속성	값	설명
Name	"lblFont"	컨트롤 이름
Text	"Font:"	표시 텍스트

GroupBox, Label, ComboBox, CheckBox, TextBox

```
private void InitializeComponent()
    {
        this.grpFont = new System.Windows.Forms.GroupBox();
        this.lblFont = new System.Windows.Forms.Label();
        this.grpFont.SuspendLayout();
        this.SuspendLayout();
        //
        // grpFont
        //
        this.grpFont.Controls.Add(this.lblFont);
        this.grpFont.Location = new System.Drawing.Point(12, 12);
        this.grpFont.Name = "grpFont";
        this.grpFont.Size = new System.Drawing.Size(776, 347);
        this.grpFont.TabIndex = 0;
        this.grpFont.TabStop = false;
        this.grpFont.Text = "GroupBox, Label, ComboBox, CheckBox, TextBox";
        //
        // lblFont
        //
        this.lblFont.AutoSize = true;
        this.lblFont.Location = new System.Drawing.Point(19, 65);
        this.lblFont.Name = "lblFont";
        this.lblFont.Size = new System.Drawing.Size(33, 12);
        this.lblFont.TabIndex = 0;
        this.lblFont.Text = "Font:";
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(800, 450);
        this.Controls.Add(this.grpFont);
        this.Name = "MainForm";
        this.Text = "Control Test";
        this.grpFont.ResumeLayout(false);
        this.grpFont.PerformLayout();
        this.ResumeLayout(false);

    }

    #endregion

    private System.Windows.Forms.GroupBox grpFont;
    private System.Windows.Forms.Label lblFont;
```

grpFont 에 lblFont 가 추가됨

GroupBox, Label, ComboBox, CheckBox, TextBox

ComboBox (cboFont)

속성	값	설명
Name	"cboFont"	컨트롤 이름

CheckBox (chkBold)

속성	값	설명
Name	"chkBold"	컨트롤 이름
Text	"굵게"	표시 텍스트 (Bold 선택용)

CheckBox (chkItalic)

속성	값	설명
Name	"chkItalic"	컨트롤 이름
Text	"이탤릭"	표시 텍스트 (Italic 선택용)

GroupBox, Label, ComboBox, CheckBox, TextBox

TextBox (txtSampleText)

속성	값	설명
Name	"txtSampleText"	컨트롤 이름
Text	"Hello, C#"	기본 표시 텍스트

GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm.Designer.cs을 보면, groupBox1 추가됨.

```
private void InitializeComponent()
      {
          this.grpFont = new System.Windows.Forms.GroupBox();
          this.lblFont = new System.Windows.Forms.Label();
          this.cboFont = new System.Windows.Forms.ComboBox();
          this.chkBold = new System.Windows.Forms.CheckBox();
          this.chkItalic = new System.Windows.Forms.CheckBox();
          this.txtSampleText = new System.Windows.Forms.TextBox();
          this.grpFont.SuspendLayout();
          this.SuspendLayout();
```

this.grpFont.SuspendLayout(); 추가 됨

GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm.Designer.cs을 보면, groupBox1 추가됨.

```
private void InitializeComponent()
      {
         //
         // grpFont
         //
         this.grpFont.Controls.Add(this.txtSampleText);
         this.grpFont.Controls.Add(this.chkItalic);
         this.grpFont.Controls.Add(this.chkBold);
         this.grpFont.Controls.Add(this.cboFont);
         this.grpFont.Controls.Add(this.lblFont);
         this.grpFont.Location = new System.Drawing.Point(12, 12);
         this.grpFont.Name = "grpFont";
         this.grpFont.Size = new System.Drawing.Size(776, 347);
         this.grpFont.TabIndex = 0;
         this.grpFont.TabStop = false;
         this.grpFont.Text = "GroupBox, Label, ComboBox, CheckBox, TextBox";
```

추가 됨

GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm.Designer.cs을 보면, groupBox1 추가됨.

```
private void InitializeComponent()
        {
            #endregion

        private System.Windows.Forms.GroupBox grpFont;
        private System.Windows.Forms.Label lblFont;
        private System.Windows.Forms.CheckBox chkItalic;
        private System.Windows.Forms.CheckBox chkBold;
        private System.Windows.Forms.ComboBox cboFont;
        private System.Windows.Forms.TextBox txtSampleText;
```

추가 됨

GroupBox, Label, ComboBox, CheckBox, TextBox

MainForm.Designer.cs

```
namespace UsingControls
{
    partial class MainForm
    {
        /// <summary>
        /// 필수 디자이너 변수입니다.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// 사용 중인 모든 리소스를 정리합니다.
        /// </summary>
        /// <param name="disposing">관리되는 리소스를 삭제해야 하면 true이고, 그렇지 않으면 false입니다.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form 디자이너에서 생성한 코드
```

GroupBox, Label, ComboBox, CheckBox, TextBox

```
/// <summary>
/// 디자이너 지원에 필요한 메서드입니다.
/// 이 메서드의 내용을 코드 편집기로 수정하지 마세요.
/// </summary>
private void InitializeComponent()
{
    this.grpFont = new System.Windows.Forms.GroupBox();
    this.lblFont = new System.Windows.Forms.Label();
    this.cboFont = new System.Windows.Forms.ComboBox();
    this.chkBold = new System.Windows.Forms.CheckBox();
    this.chkItalic = new System.Windows.Forms.CheckBox();
    this.txtSampleText = new System.Windows.Forms.TextBox();
    this.grpFont.SuspendLayout();
    this.SuspendLayout();
    //
    // grpFont
    //
    this.grpFont.Controls.Add(this.txtSampleText);
    this.grpFont.Controls.Add(this.chkItalic);
    this.grpFont.Controls.Add(this.chkBold);
    this.grpFont.Controls.Add(this.cboFont);
    this.grpFont.Controls.Add(this.lblFont);
    this.grpFont.Location = new System.Drawing.Point(12, 12);
    this.grpFont.Name = "grpFont";
    this.grpFont.Size = new System.Drawing.Size(600, 180);
    this.grpFont.TabIndex = 0;
    this.grpFont.TabStop = false;
    this.grpFont.Text = "GroupBox, Label, ComboBox, CheckBox, TextBox";
    //
```

GroupBox, Label, ComboBox, CheckBox, TextBox

```
//
        // lblFont
        //
        this.lblFont.AutoSize = true;
        this.lblFont.Location = new System.Drawing.Point(20, 40);
        this.lblFont.Name = "lblFont";
        this.lblFont.Size = new System.Drawing.Size(36, 15);
        this.lblFont.TabIndex = 0;
        this.lblFont.Text = "Font:";
        //
        // cboFont
        //
        this.cboFont.FormattingEnabled = true;
        this.cboFont.Location = new System.Drawing.Point(70, 37);
        this.cboFont.Name = "cboFont";
        this.cboFont.Size = new System.Drawing.Size(180, 23);
        this.cboFont.TabIndex = 1;
        //
        // chkBold
        //
        this.chkBold.AutoSize = true;
        this.chkBold.Location = new System.Drawing.Point(280, 39);
        this.chkBold.Name = "chkBold";
        this.chkBold.Size = new System.Drawing.Size(51, 19);
        this.chkBold.TabIndex = 2;
        this.chkBold.Text = "굵게";
        this.chkBold.UseVisualStyleBackColor = true;
        //
```

GroupBox, Label, ComboBox, CheckBox, TextBox

```
//
        // chkItalic
        //
        this.chkItalic.AutoSize = true;
        this.chkItalic.Location = new System.Drawing.Point(350, 39);
        this.chkItalic.Name = "chkItalic";
        this.chkItalic.Size = new System.Drawing.Size(63, 19);
        this.chkItalic.TabIndex = 3;
        this.chkItalic.Text = "이탤릭";
        this.chkItalic.UseVisualStyleBackColor = true;
        //
        // txtSampleText
        //
        this.txtSampleText.Location = new System.Drawing.Point(20, 90);
        this.txtSampleText.Name = "txtSampleText";
        this.txtSampleText.Size = new System.Drawing.Size(550, 23);
        this.txtSampleText.TabIndex = 4;
        this.txtSampleText.Text = "Hello, C#";
        //
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(640, 210);
        this.Controls.Add(this.grpFont);
        this.Name = "MainForm";
        this.Text = "Control Test";
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.grpFont.ResumeLayout(false);
        this.grpFont.PerformLayout();
        this.ResumeLayout(false);

    }
```

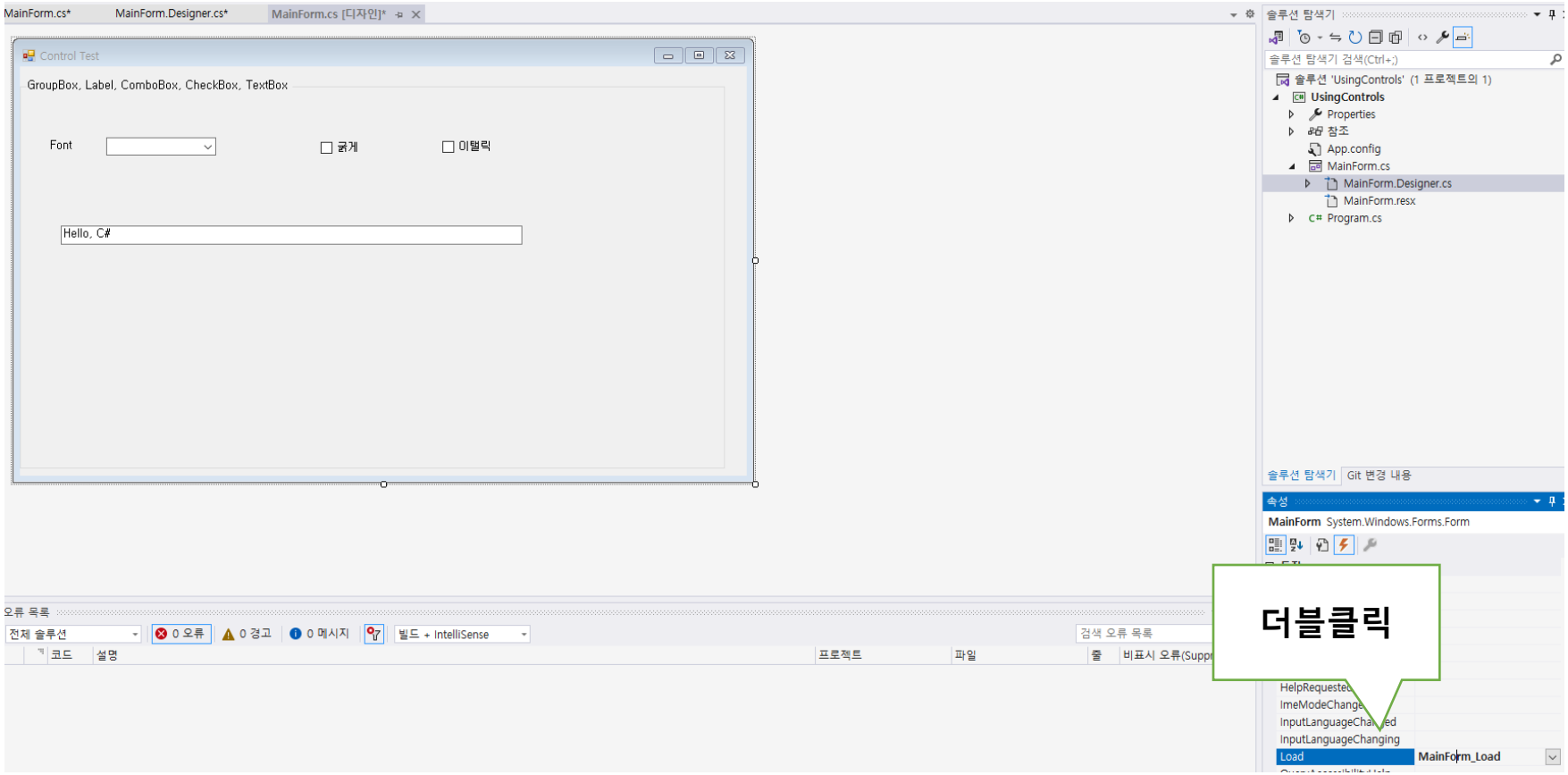
GroupBox, Label, ComboBox, CheckBox, TextBox

```
        #endregion

        private System.Windows.Forms.GroupBox grpFont;
        private System.Windows.Forms.ComboBox cboFont;
        private System.Windows.Forms.Label lblFont;
        private System.Windows.Forms.TextBox txtSampleText;
        private System.Windows.Forms.CheckBox chkItalic;
        private System.Windows.Forms.CheckBox chkBold;
    }
}
```

6. 폼 디자이너를 이용한 WinForm UI 구성

GroupBox, Label, ComboBox, CheckBox, TextBox

각 컨트롤에 이벤트 처리기 등록> MainForm > Load 이벤트 처리기 등록



GroupBox, Label, ComboBox, CheckBox, TextBox

MainApp.cs

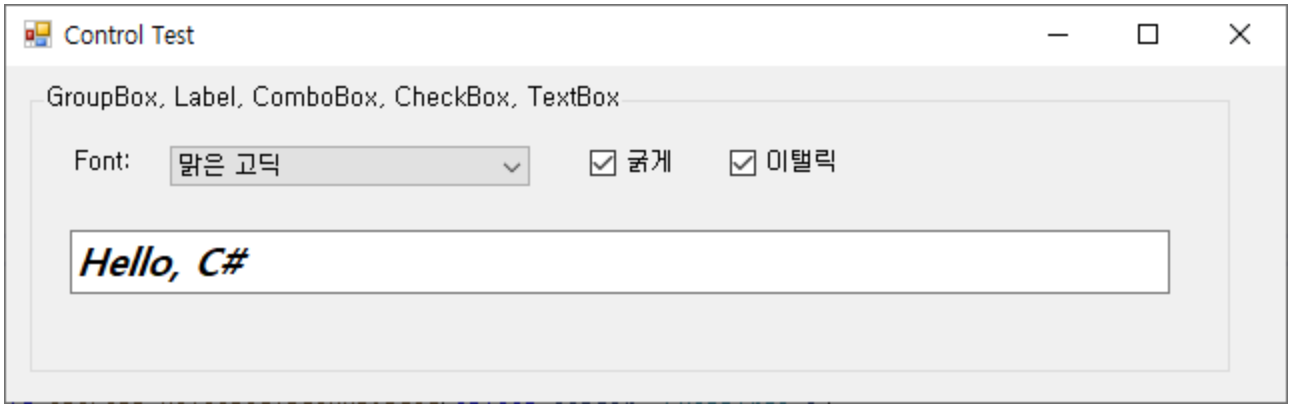
```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace UsingControls
{
   public partial class MainForm : Form
   {
      public MainForm()
      {
         InitializeComponent();

         // 이벤트 연결
         cboFont.SelectedIndexChanged += cboFont_SelectedIndexChanged;
         chkBold.CheckedChanged += chkBold_CheckedChanged;
         chkItalic.CheckedChanged += chkItalic_CheckedChanged;
      }

      private void MainForm_Load(object sender, EventArgs e)
      {
         // 폰트 목록 불러오기
         foreach (FontFamily font in FontFamily.Families)
            cboFont.Items.Add(font.Name);

         // 콤보박스 기본 선택값 설정
         cboFont.DropDownStyle = ComboBoxStyle.DropDownList;
         cboFont.SelectedItem = this.Font.FontFamily.Name;

         // 초기 텍스트 박스 폰트 설정
         ChangeFont();
      }
```

GroupBox, Label, ComboBox, CheckBox, TextBox

```
        private void ChangeFont()
        {
            if (cboFont.SelectedIndex < 0)
                return;

            // 폰트 스타일 설정
            FontStyle style = FontStyle.Regular;
            if (chkBold.Checked)
                style |= FontStyle.Bold;
            if (chkItalic.Checked)
                style |= FontStyle.Italic;

            // 텍스트 박스 폰트 적용
            txtSampleText.Font = new Font((string)cboFont.SelectedItem, 14, style);
        }

        // 이벤트 핸들러
        private void cboFont_SelectedIndexChanged(object sender, EventArgs e)
        {
            ChangeFont();
        }

        private void chkBold_CheckedChanged(object sender, EventArgs e)
        {
            ChangeFont();
        }

        private void chkItalic_CheckedChanged(object sender, EventArgs e)
        {
            ChangeFont();
        }
    }
}
```

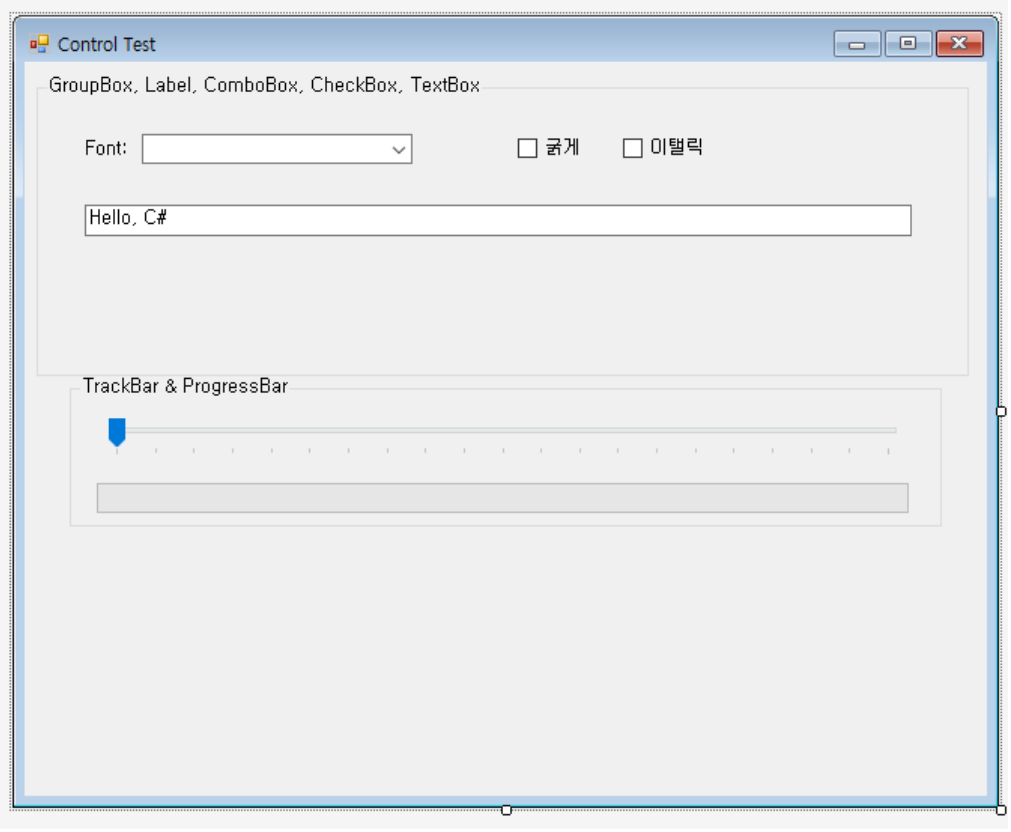
GroupBox, Label, ComboBox, CheckBox, TextBox

실행 결과



TrackBar, ProgressBar

컨트롤 배치



TrackBar, ProgressBar

제공된 소스 코드를 참고해, 작업해 봅니다.

```
this.grpFont = new System.Windows.Forms.GroupBox();
this.lblFont = new System.Windows.Forms.Label();
this.cboFont = new System.Windows.Forms.ComboBox();
this.chkBold = new System.Windows.Forms.CheckBox();
this.chkItalic = new System.Windows.Forms.CheckBox();
this.txtSampleText = new System.Windows.Forms.TextBox();
this.grpFont.SuspendLayout();
this.tbDummy = new System.Windows.Forms.TrackBar();
this.pgDummy = new System.Windows.Forms.ProgressBar();
this.grpBar = new System.Windows.Forms.GroupBox();
this.grpBar.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.tbDummy)).BeginInit();
this.SuspendLayout();
```

TrackBar, ProgressBar

```
// grpBar
        this.grpBar.Controls.Add(this.tbDummy);
        this.grpBar.Controls.Add(this.pgDummy);
        this.grpBar.Location = new System.Drawing.Point(30, 210);
        this.grpBar.Margin = new System.Windows.Forms.Padding(2);
        this.grpBar.Name = "grpBar";
        this.grpBar.Padding = new System.Windows.Forms.Padding(2);
        this.grpBar.Size = new System.Drawing.Size(580, 100);
        this.grpBar.TabIndex = 1;
        this.grpBar.TabStop = false;
        this.grpBar.Text = "TrackBar && ProgressBar";

        // tbDummy
        this.tbDummy.Location = new System.Drawing.Point(18, 25);
        this.tbDummy.Margin = new System.Windows.Forms.Padding(2);
        this.tbDummy.Maximum = 20;
        this.tbDummy.Name = "tbDummy";
        this.tbDummy.Size = new System.Drawing.Size(540, 45);
        this.tbDummy.TabIndex = 0;
        this.tbDummy.Scroll += new System.EventHandler(this.tbDummy_Scroll);

        //  pgDummy (위치 아래로 조정)
        this.pgDummy.Location = new System.Drawing.Point(18, 70);
        this.pgDummy.Margin = new System.Windows.Forms.Padding(2);
        this.pgDummy.Maximum = 20;
        this.pgDummy.Name = "pgDummy";
        this.pgDummy.Size = new System.Drawing.Size(540, 20);
        this.pgDummy.TabIndex = 1;
```

TrackBar, ProgressBar

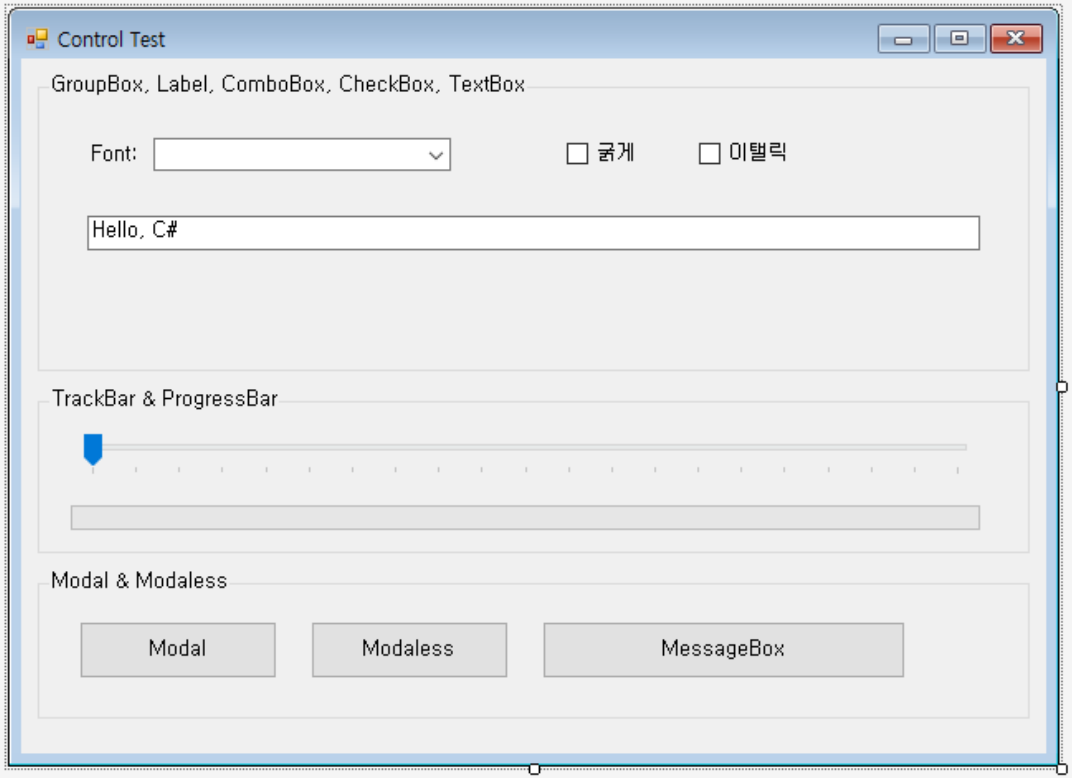
```
//
        // MainForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(800, 450);
        this.Controls.Add(this.grpBar);
        this.Controls.Add(this.grpFont);
        this.Name = "MainForm";
        this.Text = "Control Test";
        this.Load += new System.EventHandler(this.MainForm_Load);
        this.grpFont.ResumeLayout(false);
        this.grpFont.PerformLayout();
        this.grpBar.ResumeLayout(false);
        this.grpBar.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.tbDummy)).EndInit();
        this.ResumeLayout(false);
```

TrackBar, ProgressBar

```
        #endregion

        private System.Windows.Forms.GroupBox grpFont;
        private System.Windows.Forms.Label lblFont;
        private System.Windows.Forms.CheckBox chkItalic;
        private System.Windows.Forms.CheckBox chkBold;
        private System.Windows.Forms.ComboBox cboFont;
        private System.Windows.Forms.TextBox txtSampleText;
        private System.Windows.Forms.ProgressBar pgDummy;
        private System.Windows.Forms.TrackBar tbDummy;
        private System.Windows.Forms.GroupBox grpBar;
```

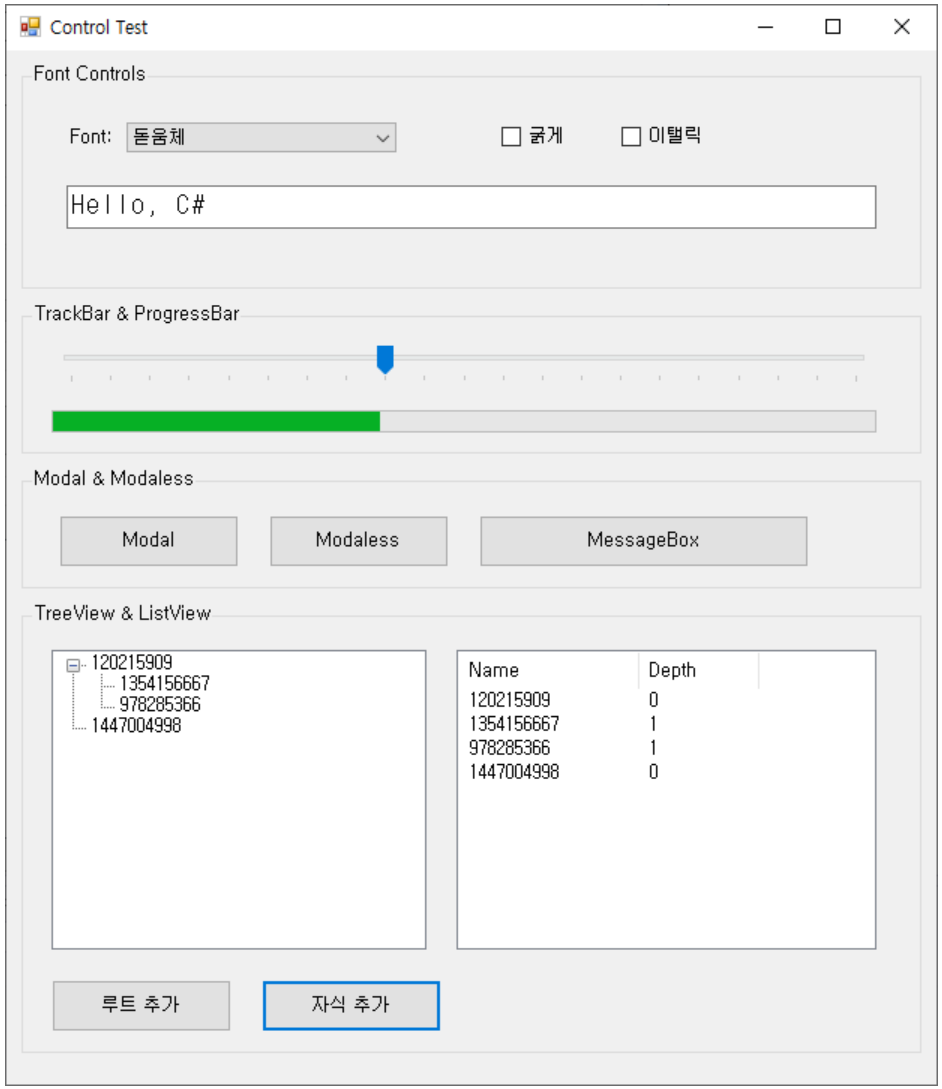
Button, Form, Dialog

컨트롤 배치



Treeview, ListView

전체 컨트롤 배치



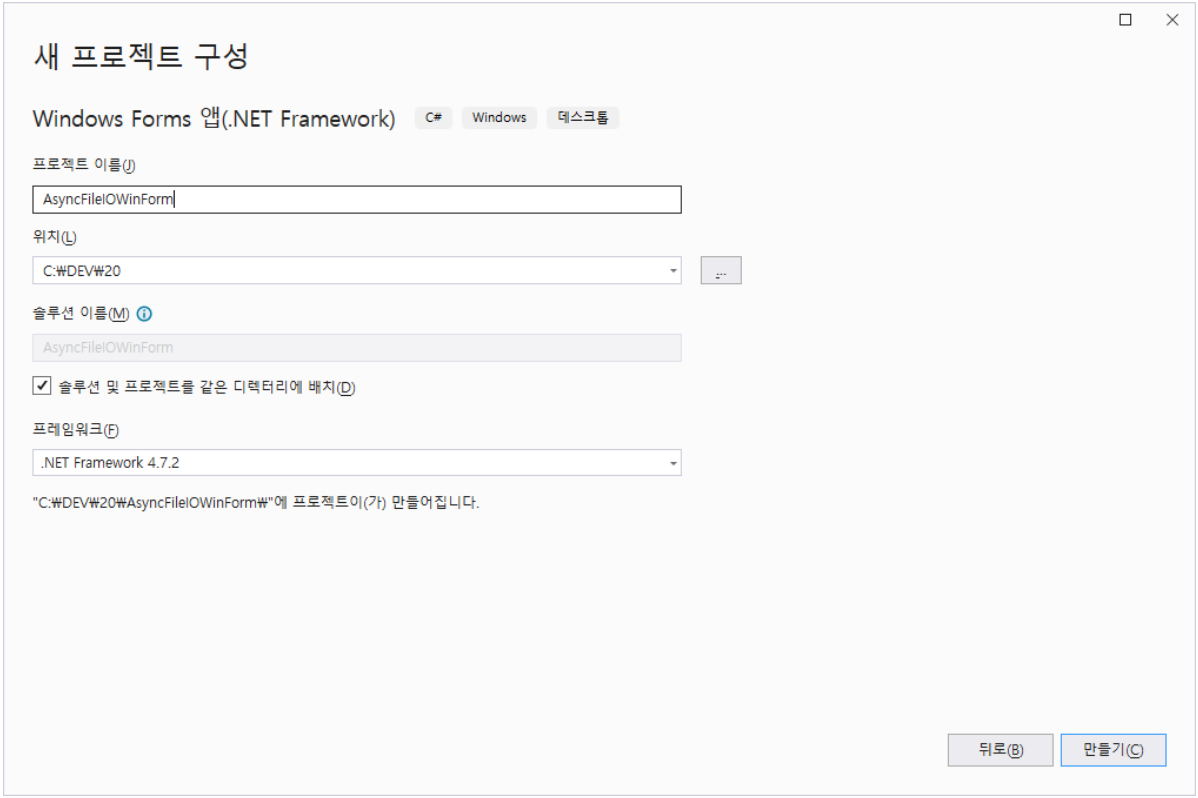
7. 사용자 인터페이스와 비동기 작업

새 프로젝트 만들기

- 새 프로젝트 만들기 창에서 C#용 Windows Forms 앱 (.NET Framework) 템플릿을 선택합니다.

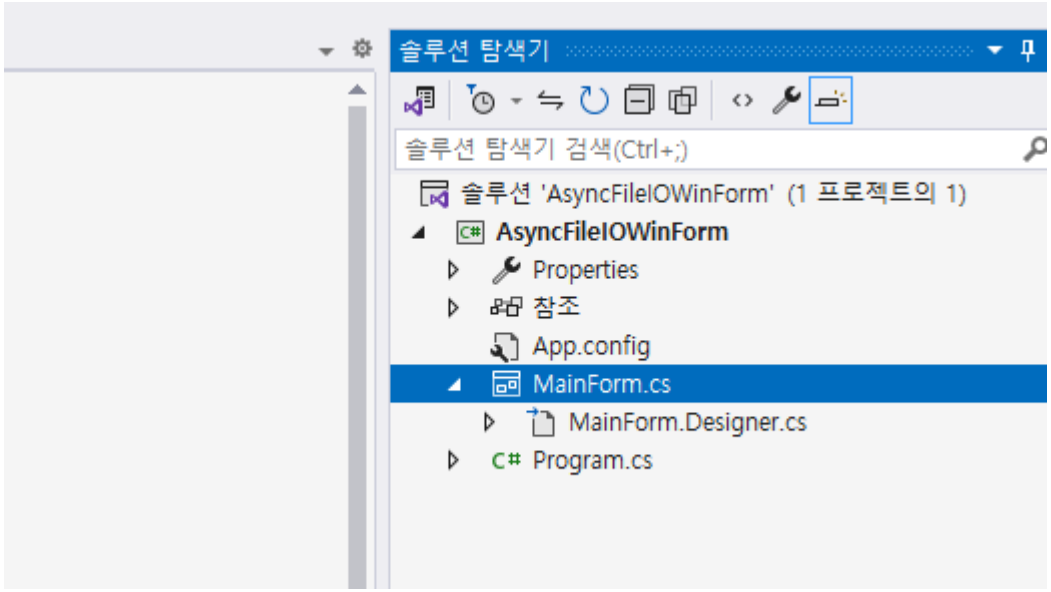
새 프로젝트 만들기

AsyncFileIOWinForm



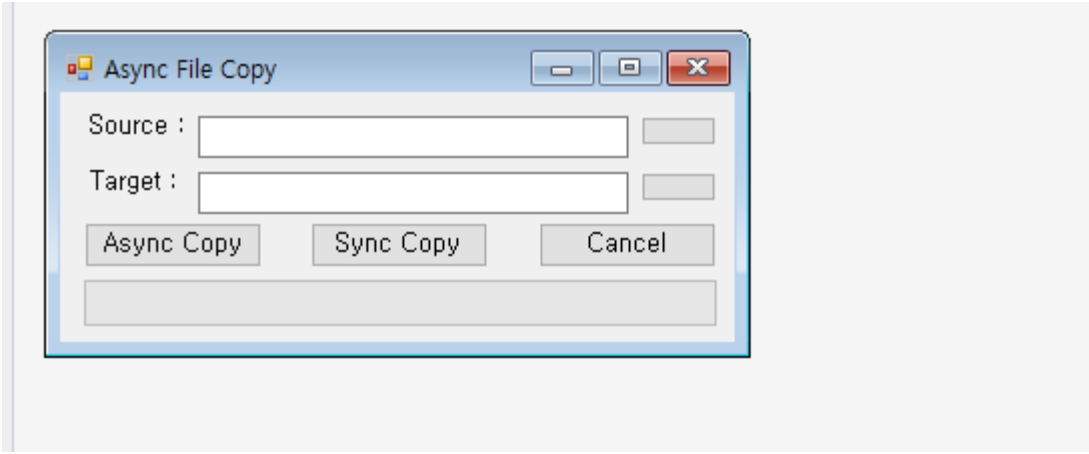
UI 구성하기

Form1.cs를 MainForm.cs 로 변경



UI 구성하기

MainForm.Designer.cs



UI 구성하기

```
namespace AsyncFileIOWinForm
{
    partial class MainForm
    {
        /// <summary>
        /// 필수 디자이너 변수입니다.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// 사용 중인 모든 리소스를 정리합니다.
        /// </summary>
        /// <param name="disposing">관리되는 리소스를 삭제해야 하면 true이고, 그렇지 않으면 false입니다.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form 디자이너에서 생성한 코드

        /// <summary>
        /// 디자이너 지원에 필요한 메서드입니다.
        /// 이 메서드의 내용을 코드 편집기로 수정하지 마세요.
        /// </summary>
        private void InitializeComponent()
        {
            this.lblSource = new System.Windows.Forms.Label();
            this.txtSource = new System.Windows.Forms.TextBox();
            this.btnFindSource = new System.Windows.Forms.Button();
            this.btnFindTarget = new System.Windows.Forms.Button();
            this.txtTarget = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.btnAsyncCopy = new System.Windows.Forms.Button();
            this.pbCopy = new System.Windows.Forms.ProgressBar();
            this.btnCancel = new System.Windows.Forms.Button();
            this.btnSyncCopy = new System.Windows.Forms.Button();
            this.SuspendLayout();
```

UI 구성하기

```
//
        // lblSource
        //
        this.lblSource.AutoSize = true;
        this.lblSource.Location = new System.Drawing.Point(17, 23);
        this.lblSource.Name = "lblSource";
        this.lblSource.Size = new System.Drawing.Size(78, 25);
        this.lblSource.TabIndex = 0;
        this.lblSource.Text = "Source :";
        //
        // txtSource
        //
        this.txtSource.Location = new System.Drawing.Point(99, 24);
        this.txtSource.Name = "txtSource";
        this.txtSource.Size = new System.Drawing.Size(305, 31);
        this.txtSource.TabIndex = 1;
        //
        // btnFindSource
        //
        this.btnFindSource.Location = new System.Drawing.Point(414, 24);
        this.btnFindSource.Name = "btnFindSource";
        this.btnFindSource.Size = new System.Drawing.Size(54, 31);
        this.btnFindSource.TabIndex = 2;
        this.btnFindSource.Text = "...";
        this.btnFindSource.UseVisualStyleBackColor = true;
        this.btnFindSource.Click += new System.EventHandler(this.btnFindSource_Click);
        //
        // btnFindTarget
        //
        this.btnFindTarget.Location = new System.Drawing.Point(414, 83);
        this.btnFindTarget.Name = "btnFindTarget";
        this.btnFindTarget.Size = new System.Drawing.Size(54, 31);
        this.btnFindTarget.TabIndex = 5;
        this.btnFindTarget.Text = "...";
        this.btnFindTarget.UseVisualStyleBackColor = true;
        this.btnFindTarget.Click += new System.EventHandler(this.btnFindTarget_Click);
```

UI 구성하기

```
//
        // txtTarget
        //
        this.txtTarget.Location = new System.Drawing.Point(99, 83);
        this.txtTarget.Name = "txtTarget";
        this.txtTarget.Size = new System.Drawing.Size(305, 31);
        this.txtTarget.TabIndex = 4;
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Location = new System.Drawing.Point(17, 82);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(74, 25);
        this.label1.TabIndex = 3;
        this.label1.Text = "Target :";
        //
        // btnAsyncCopy
        //
        this.btnAsyncCopy.Location = new System.Drawing.Point(17, 135);
        this.btnAsyncCopy.Name = "btnAsyncCopy";
        this.btnAsyncCopy.Size = new System.Drawing.Size(127, 48);
        this.btnAsyncCopy.TabIndex = 6;
        this.btnAsyncCopy.Text = "Async Copy";
        this.btnAsyncCopy.UseVisualStyleBackColor = true;
        this.btnAsyncCopy.Click += new System.EventHandler(this.btnAsyncCopy_Click);
        //
        // pbCopy
        //
        this.pbCopy.Location = new System.Drawing.Point(17, 195);
        this.pbCopy.Name = "pbCopy";
        this.pbCopy.Size = new System.Drawing.Size(451, 48);
        this.pbCopy.TabIndex = 7;
        //
        // btnCancel
        //
        this.btnCancel.Location = new System.Drawing.Point(341, 135);
        this.btnCancel.Name = "btnCancel";
        this.btnCancel.Size = new System.Drawing.Size(127, 48);
        this.btnCancel.TabIndex = 8;
        this.btnCancel.Text = "Cancel";
        this.btnCancel.UseVisualStyleBackColor = true;
        this.btnCancel.Click += new System.EventHandler(this.btnCancel_Click);
```

UI 구성하기

```
//
            // btnSyncCopy
            //
            this.btnSyncCopy.Location = new System.Drawing.Point(179, 135);
            this.btnSyncCopy.Name = "btnSyncCopy";
            this.btnSyncCopy.Size = new System.Drawing.Size(127, 48);
            this.btnSyncCopy.TabIndex = 9;
            this.btnSyncCopy.Text = "Sync Copy";
            this.btnSyncCopy.UseVisualStyleBackColor = true;
            this.btnSyncCopy.Click += new System.EventHandler(this.btnSyncCopy_Click);
            //
            // MainForm
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(10F, 25F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(482, 261);
            this.Controls.Add(this.btnSyncCopy);
            this.Controls.Add(this.btnCancel);
            this.Controls.Add(this.pbCopy);
            this.Controls.Add(this.btnAsyncCopy);
            this.Controls.Add(this.btnFindTarget);
            this.Controls.Add(this.txtTarget);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.btnFindSource);
            this.Controls.Add(this.txtSource);
            this.Controls.Add(this.lblSource);
            this.Name = "MainForm";
            this.Text = "Async File Copy";
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label lblSource;
        private System.Windows.Forms.TextBox txtSource;
        private System.Windows.Forms.Button btnFindSource;
        private System.Windows.Forms.Button btnFindTarget;
        private System.Windows.Forms.TextBox txtTarget;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Button btnAsyncCopy;
        private System.Windows.Forms.ProgressBar pbCopy;
        private System.Windows.Forms.Button btnCancel;
        private System.Windows.Forms.Button btnSyncCopy;
    }
}
```

MainForm

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace AsyncFileIOWinForm
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private async Task<long> CopyAsync(string FromPath, string ToPath)
        {
            btnSyncCopy.Enabled = false;
            long totalCopied = 0;

            using (FileStream fromStream = new FileStream(FromPath, FileMode.Open))
            {
                using (FileStream toStream = new FileStream(ToPath, FileMode.Create))
                {
                    byte[] buffer = new byte[1024 * 1024];
                    int nRead = 0;
                    while ((nRead = await fromStream.ReadAsync(buffer, 0, buffer.Length)) != 0)
                    {
                        await toStream.WriteAsync(buffer, 0, nRead);
                        totalCopied += nRead;

                        // 프로그레스바에 현재 파일 복사 상태 표시
                        pbCopy.Value =
                            (int)(((double)totalCopied / (double)fromStream.Length)
                                                * pbCopy.Maximum);
                    }
                }
            }

            btnSyncCopy.Enabled = true;
            return totalCopied;
        }
```

MainForm

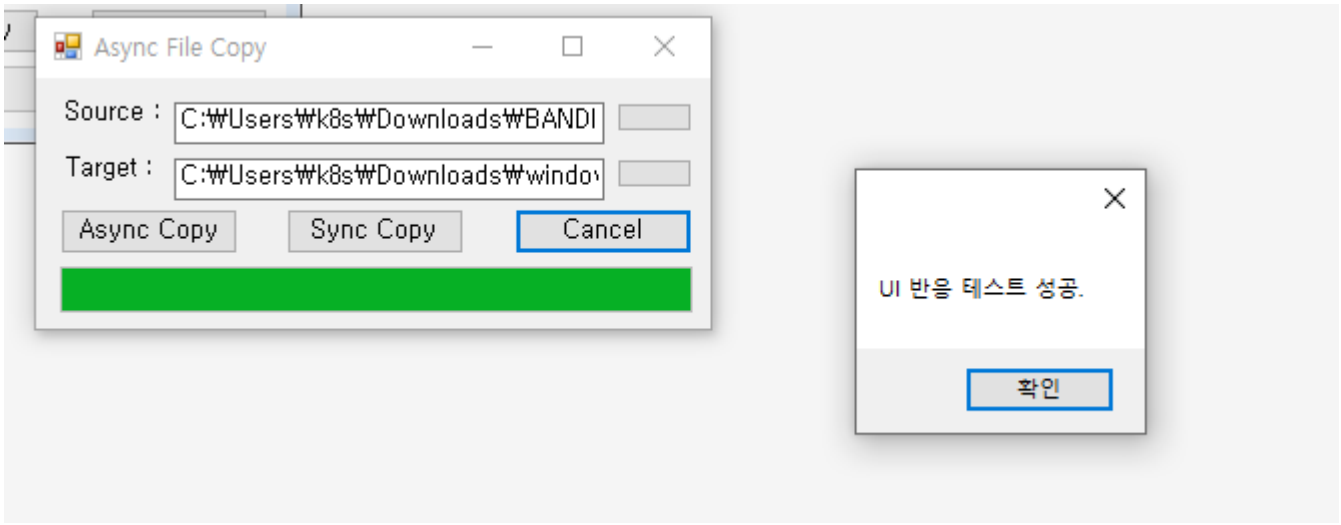
```
private long CopySync(string FromPath, string ToPath)
{
    btnAsyncCopy.Enabled = false;
    long totalCopied = 0;

    using (FileStream fromStream = new FileStream(FromPath, FileMode.Open))
    {
        using (FileStream toStream = new FileStream(ToPath, FileMode.Create))
        {
            byte[] buffer = new byte[1024 * 1024];
            int nRead = 0;
            while ((nRead = fromStream.Read(buffer, 0, buffer.Length)) != 0)
            {
                toStream.Write(buffer, 0, nRead);
                totalCopied += nRead;

                // 프로그레스바에 현재 파일 복사 상태 표시
                pbCopy.Value =
                 (int)(((double)totalCopied / (double)fromStream.Length)
                                        * pbCopy.Maximum);
            }
        }
    }

    btnAsyncCopy.Enabled = true;
    return totalCopied;
}

private void btnFindSource_Click(object sender, EventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    if (dlg.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        txtSource.Text = dlg.FileName;
    }
}
```

MainForm

```
        private void btnFindTarget_Click(object sender, EventArgs e)
        {
            SaveFileDialog dlg = new SaveFileDialog();
            if (dlg.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                txtTarget.Text = dlg.FileName;
            }
        }

        private async void btnAsyncCopy_Click(object sender, EventArgs e)
        {
            long totalCopied = await CopyAsync(txtSource.Text, txtTarget.Text);
        }

        private void btnSyncCopy_Click(object sender, EventArgs e)
        {

            long totalCopied = CopySync(txtSource.Text, txtTarget.Text);

        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            MessageBox.Show("UI 반응 테스트 성공.");
        }
    }
}
```

결과

원본 파일과 대상 파일을 선택후, 복사 하다 취소 버튼을 눌러 봅니다.

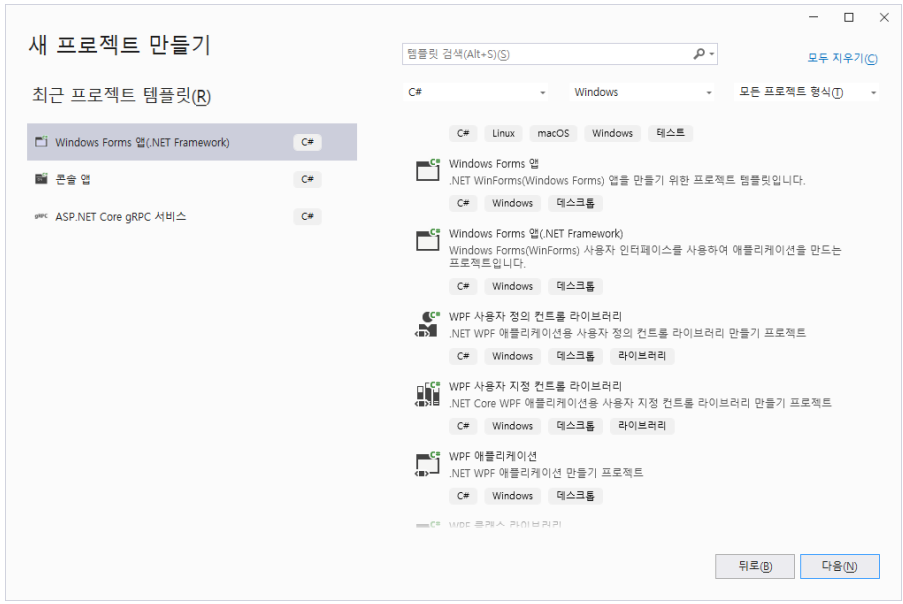


8. 실습 1: 윈폼으로 계산기 만들기

개요

●앞서 만든, 콘솔 계산기를 윈폼을 이용하여 만들어 보자.

- +, -, ×, ÷ 수행
- 프로젝트 명 : Calculator

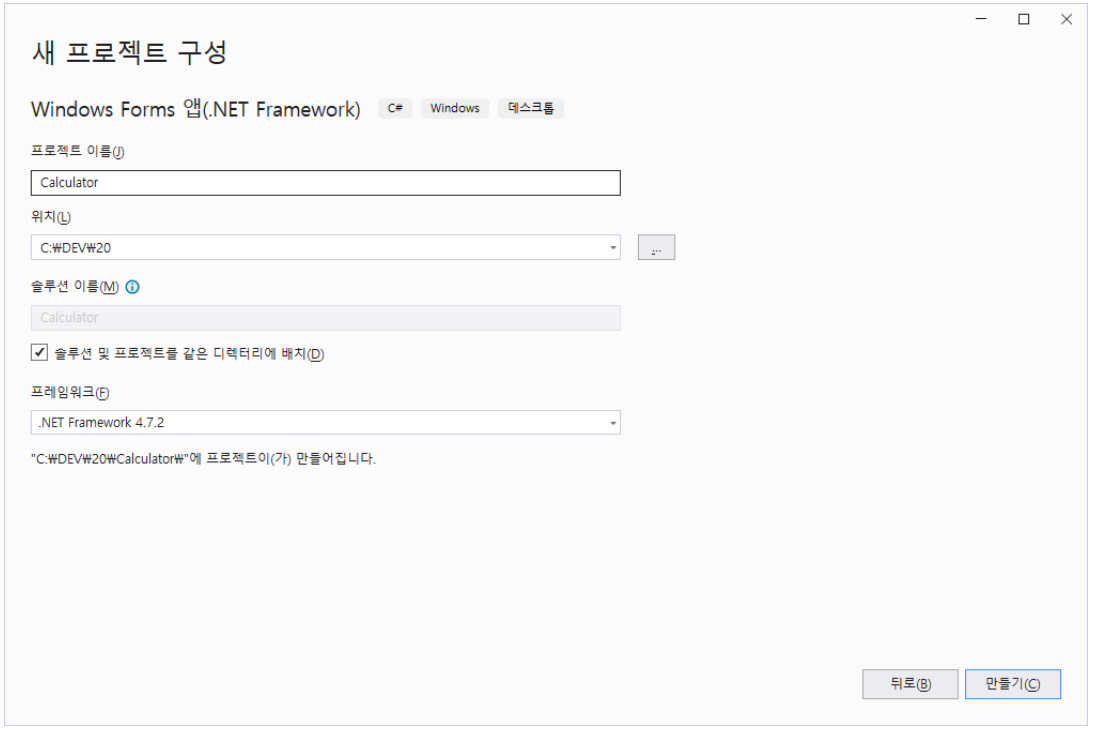


8. 실습 1: 윈폼으로 계산기 만들기

개요

●앞서 만든, 콘솔 계산기를 윈폼을 이용하여 만들어 보자.

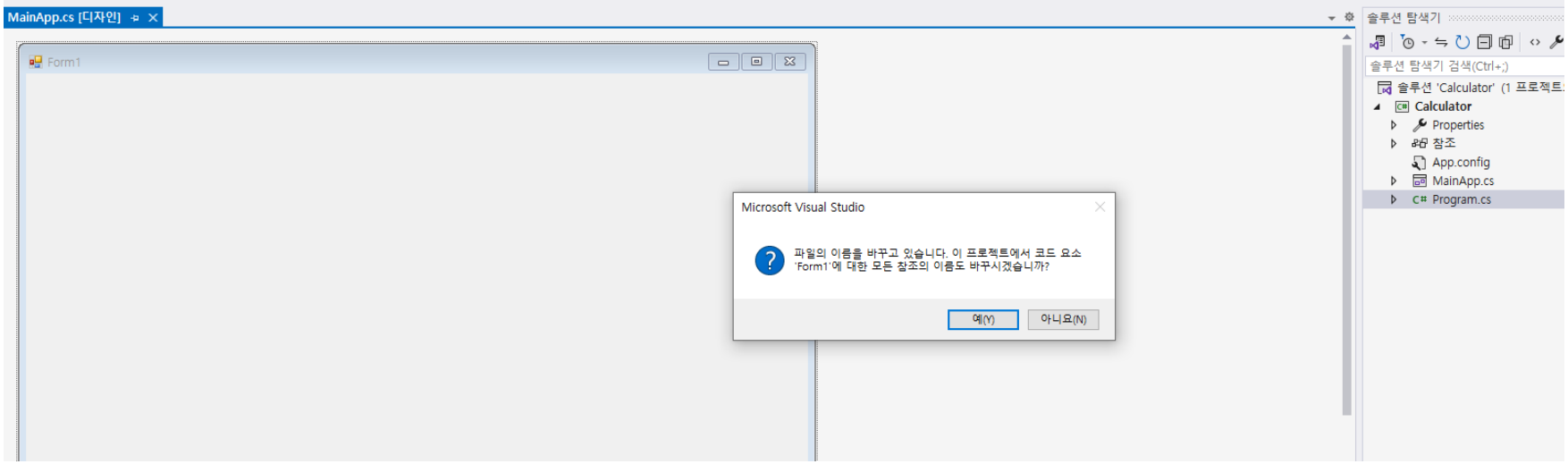
- +, -, ×, ÷ 수행
- 프로젝트 명 : Calculator



8. 실습 1: 윈폼으로 계산기 만들기

개요

- Form1.cs를 MainApp.cs로 변경



8. 실습 1: 윈폼으로 계산기 만들기

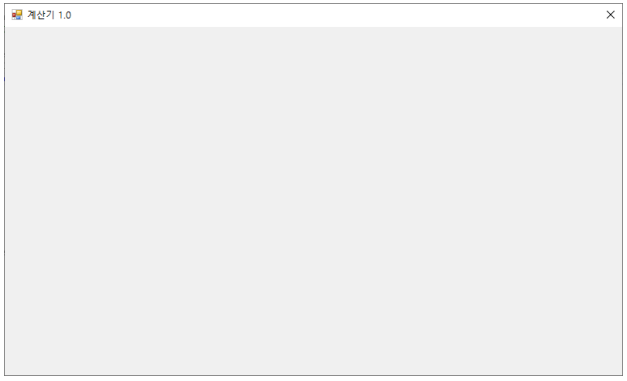
개요

●앞서 만든, 콘솔 계산기를 윈폼을 이용하여 만들어 보자.

- MainApp.Designer.cs, 다음처럼 속성 변경

```
private void InitializeComponent()
 {
    this.components = new System.ComponentModel.Container();
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(800, 450);
    this.Text = "Form1";

    this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
    this.Name = "MainApp";
    this.Text = "계산기 1.0";
 }
```

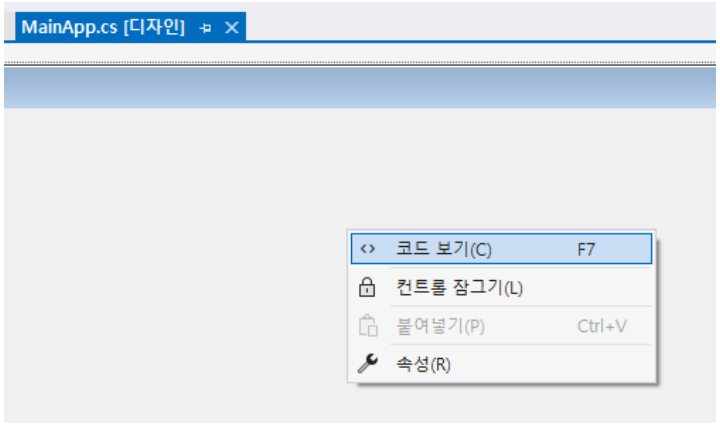
추가



8. 실습 1: 윈폼으로 계산기 만들기

버튼 배치

- MainApp.cs에 다음 코드 작성




```
namespace Calculator
{
   public partial class MainApp : Form
   {
      Button[] btnNums = new Button[10];

      public MainApp()
      {
         InitializeComponent();

         for (int n = 0; n < btnNums.Length; n++)
         {
            int m = 9 - n;
            btnNums[n] = new Button();
            btnNums[n].Text = n.ToString();
            btnNums[n].Size = new Size(50, 50);
            btnNums[n].Location = new Point(124 - ((m - (((int)(m / 3)) * 3)) * btnNums[n].Width) - ((m - (((int)(m / 3)) * 3)) * 6) - ((int)(m / 9) * 112), (((int)(m /
3)) * btnNums[n].Height) + (((int)(m / 3)) * 6) + 90);
            btnNums[n].Tag = n;
            btnNums[n].Click += new EventHandler(btnNums_Click);

            this.Controls.Add(btnNums[n]);
         }
      }

      void btnNums_Click(object sender, EventArgs e)
      {
         Button btnNum = sender as Button;
         int index = (int)btnNum.Tag;
      }
   }
}
```

버튼 배치

- MainApp.Designer.cs 에 나머지 버튼 배치

```
using System.Drawing;
using System.Windows.Forms;

namespace Calculator
{
    partial class MainApp
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

8. 실습 1: 윈폼으로 계산기 만들기

버튼 배치

- MainApp.Designer.cs 에 나머지 버튼 배치

```
#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.btnSum = new System.Windows.Forms.Button();
    this.btnMinus = new System.Windows.Forms.Button();
    this.btnDivide = new System.Windows.Forms.Button();
    this.btnMulti = new System.Windows.Forms.Button();
    this.btnClear = new System.Windows.Forms.Button();
    this.btnResult = new System.Windows.Forms.Button();
    this.laDisplay = new System.Windows.Forms.Label();
    this.SuspendLayout();
    //
    // btnSum
    //
    this.btnSum.Location = new System.Drawing.Point(180, 90);
    this.btnSum.Name = "btnSum";
    this.btnSum.Size = new System.Drawing.Size(50, 50);
    this.btnSum.TabIndex = 0;
    this.btnSum.Tag = "Sum";
    this.btnSum.Text = "+";
    this.btnSum.UseVisualStyleBackColor = true;
    this.btnSum.Click += new System.EventHandler(this.btnCalculates_Click);
    //
    // btnMinus
    //
    this.btnMinus.Location = new System.Drawing.Point(180, 146);
    this.btnMinus.Name = "btnMinus";
    this.btnMinus.Size = new System.Drawing.Size(50, 50);
    this.btnMinus.TabIndex = 1;
    this.btnMinus.Tag = "Minus";
    this.btnMinus.Text = "-";
    this.btnMinus.UseVisualStyleBackColor = true;
    this.btnMinus.Click += new System.EventHandler(this.btnCalculates_Click);
```

8. 실습 1: 윈폼으로 계산기 만들기

버튼 배치

- MainApp.Designer.cs 에 나머지 버튼 배치

```
//
// btnDivide
//
this.btnDivide.Location = new System.Drawing.Point(180, 258);
this.btnDivide.Name = "btnDivide";
this.btnDivide.Size = new System.Drawing.Size(50, 50);
this.btnDivide.TabIndex = 2;
this.btnDivide.Tag = "Divide";
this.btnDivide.Text = "/";
this.btnDivide.UseVisualStyleBackColor = true;
this.btnDivide.Click += new System.EventHandler(this.btnCalculates_Click);
//
// btnMulti
//
this.btnMulti.Location = new System.Drawing.Point(180, 202);
this.btnMulti.Name = "btnMulti";
this.btnMulti.Size = new System.Drawing.Size(50, 50);
this.btnMulti.TabIndex = 3;
this.btnMulti.Tag = "Multi";
this.btnMulti.Text = "X";
this.btnMulti.UseVisualStyleBackColor = true;
this.btnMulti.Click += new System.EventHandler(this.btnCalculates_Click);
//
// btnClear
//
this.btnClear.Location = new System.Drawing.Point(68, 258);
this.btnClear.Name = "btnClear";
this.btnClear.Size = new System.Drawing.Size(50, 50);
this.btnClear.TabIndex = 4;
this.btnClear.Tag = "Clear";
this.btnClear.Text = "C";
this.btnClear.UseVisualStyleBackColor = true;
this.btnClear.Click += new System.EventHandler(this.btnCalculates_Click);
```

8. 실습 1: 윈폼으로 계산기 만들기

버튼 배치

- MainApp.Designer.cs 에 나머지 버튼 배치

```
//
// btnResult
//
this.btnResult.Location = new System.Drawing.Point(124, 258);
this.btnResult.Name = "btnResult";
this.btnResult.Size = new System.Drawing.Size(50, 50);
this.btnResult.TabIndex = 5;
this.btnResult.Tag = "Result";
this.btnResult.Text = "=";
this.btnResult.UseVisualStyleBackColor = true;
this.btnResult.Click += new System.EventHandler(this.btnCalculates_Click);
//
// laDisplay
//
this.laDisplay.BackColor = System.Drawing.Color.White;
this.laDisplay.Font = new System.Drawing.Font("Dotum", 27F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(129)));
this.laDisplay.Location = new System.Drawing.Point(12, 9);
this.laDisplay.Name = "laDisplay";
this.laDisplay.Size = new System.Drawing.Size(218, 78);
this.laDisplay.TabIndex = 6;
this.laDisplay.Text = "0";
this.laDisplay.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
```

8. 실습 1: 윈폼으로 계산기 만들기

버튼 배치

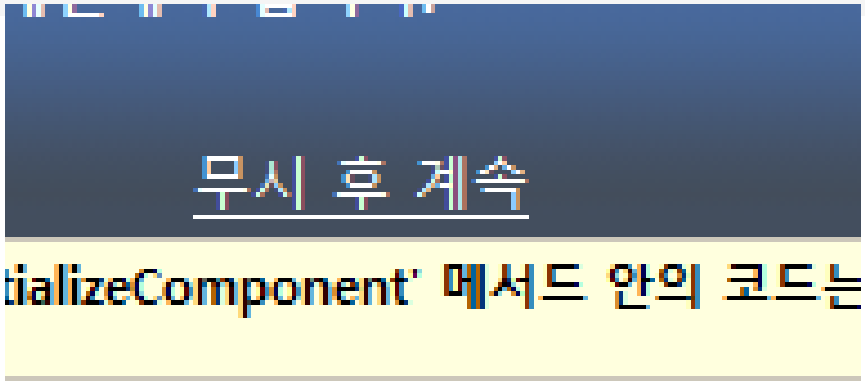
- MainApp.Designer.cs 에 나머지 버튼 배치

```
        //
        // MainApp
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(259, 333);
        this.Controls.Add(this.laDisplay);
        this.Controls.Add(this.btnResult);
        this.Controls.Add(this.btnClear);
        this.Controls.Add(this.btnMulti);
        this.Controls.Add(this.btnDivide);
        this.Controls.Add(this.btnMinus);
        this.Controls.Add(this.btnSum);
        this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
        this.MaximizeBox = false;
        this.MinimizeBox = false;
        this.Name = "MainApp";
        this.Text = "계산기 1.0";
        this.ResumeLayout(false);

    }

    #endregion

    private System.Windows.Forms.Button btnSum;
    private System.Windows.Forms.Button btnMinus;
    private System.Windows.Forms.Button btnDivide;
    private System.Windows.Forms.Button btnMulti;
    private System.Windows.Forms.Button btnClear;
    private System.Windows.Forms.Button btnResult;
    private System.Windows.Forms.Label laDisplay;
  }
}
```

8. 실습 1: 윈폼으로 계산기 만들기

코드 추가

- **MainApp.cs에 코드 갱신 > 무시 후 계속**



8. 실습 1: 윈폼으로 계산기 만들기

코드 추가

- MainApp.cs에 코드 갱신

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Calculator
{
    public partial class MainApp : Form
    {
        Button[] btnNums = new Button[10];
        int[] val = new int[2];
        int step, result;
        string oper;

        public MainApp()
        {
            InitializeComponent();

            for (int n = 0; n < btnNums.Length; n++)
            {
                int m = 9 - n;
                btnNums[n] = new Button();
                btnNums[n].Text = n.ToString();
                btnNums[n].Size = new Size(50, 50);
                btnNums[n].Location = new Point(124 - ((m - (((int)(m / 3)) * 3)) * btnNums[n].Width) - ((m - (((int)(m / 3)) * 3)) * 6) - ((int)(m / 9) * 112), (((int)(m /
3)) * btnNums[n].Height) + (((int)(m / 3)) * 6) + 90);
                btnNums[n].Tag = n;
                btnNums[n].Click += new EventHandler(btnNums_Click);

                this.Controls.Add(btnNums[n]);
            }
        }
```

8. 실습 1: 윈폼으로 계산기 만들기

코드 추가

- MainApp.cs에 코드 갱신

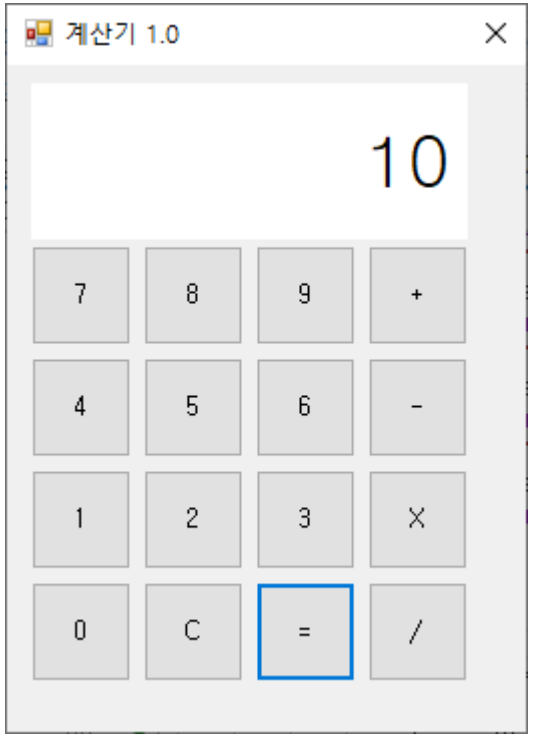
```
void btnNums_Click(object sender, EventArgs e)
{
    Button btnNum = sender as Button;
    int index = (int)btnNum.Tag;

    if (laDisplay.Text == "0" || result != 0)
        laDisplay.Text = index.ToString();
    else
        laDisplay.Text += index.ToString();
}

private void btnCalculates_Click(object sender, EventArgs e)
{
    Button btnCal = sender as Button;
    string tag = (string)btnCal.Tag;
    result = 0;
    val[step] = int.Parse(laDisplay.Text);
    step = (step + 1) % 3;
    if (tag == "Result" || step >= 2)
    {
        switch (oper)
        {
            case "Sum":
                result = val[0] + val[1];
                break;
            case "Minus":
                result = val[0] - val[1];
                break;
            case "Multi":
                result = val[0] * val[1];
                break;
            case "Divide":
                result = val[0] / val[1];
                break;
        }
        laDisplay.Text = result.ToString();
        step = 0;
    }
```

8. 실습 1: 윈폼으로 계산기 만들기

코드 추가

- MainApp.cs에 코드 갱신

```
        else if (tag == "Clear")
        {
            step = 0;
            laDisplay.Text = (0).ToString();
        }
        else
        {
            laDisplay.Text = (0).ToString();
        }
        oper = tag;
    }
  }
}
```



감사합니다.

❖ Mobile: 010-9591-1401
❖ E-mail: onlooker2zip@naver.com