



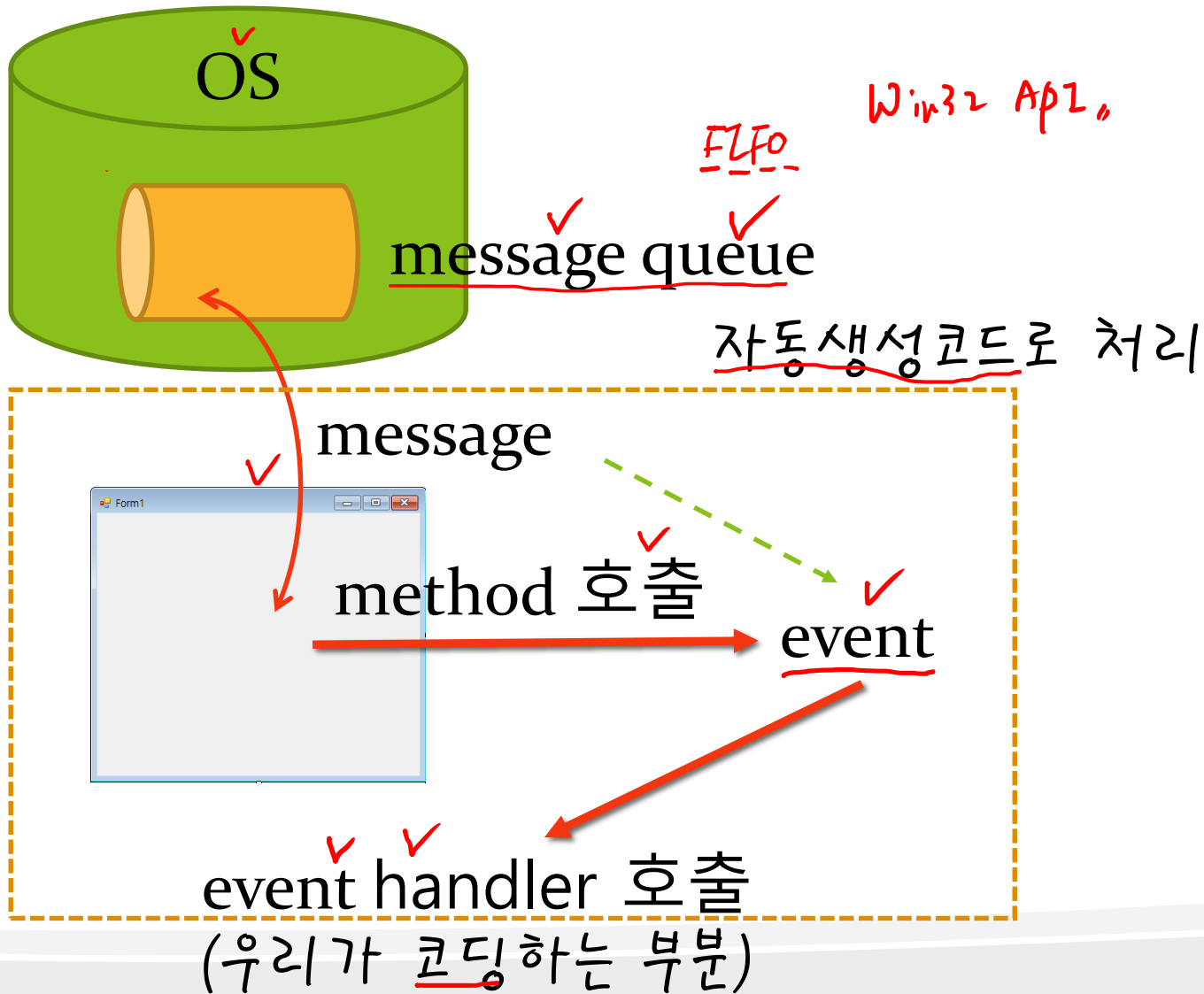
# 14강. 윈도우 프로그래밍



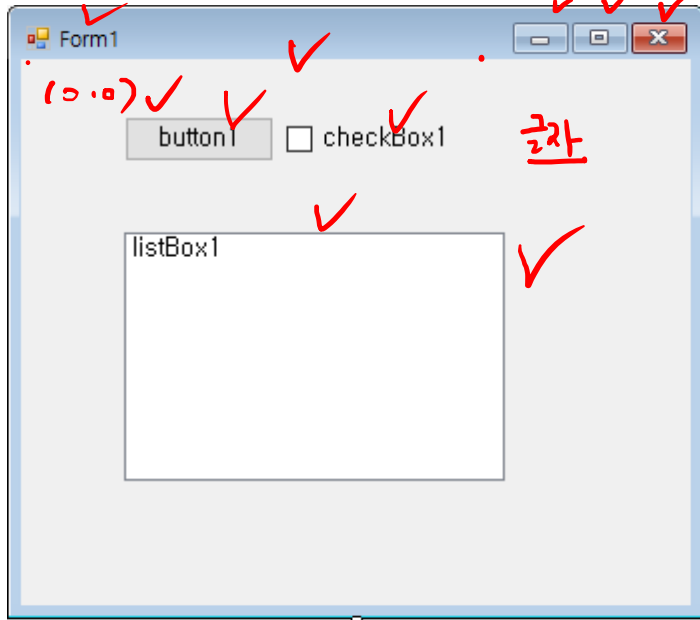
# 목차

- 윈도우 폼과 코드 구조
- 자동 생성<sup>✓</sup> 코드 분석
- Form<sup>✓</sup> 클래스

# 운영체제와 윈도우 프로그램과의 관계



# 윈도우 폼



Program.cs : 윈폼 생성과 실행 → 자동

Form.cs : 폼 프로그래밍 ←

Form.Designer.cs : 자동생성코드

(이벤트처리, 컨트롤 속성 등...)





# Program.cs

```
static class Program
{
    /// <summary>
    /// 해당 응용 프로그램의 주 진입점입니다.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}
```

static

}

응용 프로그램 생성 이전에 호출해야 할 메서드

# Application class

- 상속구조

System.Object

System.Windows.Threading.DispatcherObject

System.Windows.Application

- public sealed class Application

응용 프로그램을 관리하는 속성  
및 정보를 얻기 위한 클래스

MSDN

# EnableVisualStyles()

- 품의 스타일 설정  
✓색, ✓글꼴, ✓테마 등으로 시각적인 요소  
설정의 허가  
➔ 운영체제<sup>✓</sup>로부터 가져옴



## SetCompatibleTextRenderingDefault(false)

- 폼에 출력할 글자의 렌더링을 GDI vs GDI+ 결정

false → GDI

true → GDI+

- public static void SetCompatibleTextRenderingDefault();

## Run() ✓

loop

- 메시지 루프를 실행과 출력할 폼 설정 ✓
- `public static void Run( Form MainForm )`

```
Application.EnableVisualStyles();
```

```
Application.SetCompatibleTextRenderingDefault(false);
```

```
Application.Run(new Form1());
```



# Form.cs

```
namespace WindowsFormsApp1
{
    // →
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

- partial

- 클래스나 구조체, 인터페이스, 메서드의 정의를 둘 이상의 소스 파일로 분할
- 규모가 큰 프로젝트일 때 하나의 파일에 다수가 접근할 때 사용



Form.Designer.cs



## Form.Designer.cs

- 폼 위에 올려지는 컨트롤과 폼의 속성을 설정하는 자동생성코드
- 되도록 편집하지 않는 것이 좋다

코드보기



Form class

# Form class

- 폼에 대한 클래스
- 상속관계

System.Object

System.MarshalByRefObject

System.ComponentModel.Component

System.Windows.Forms.Control

System.Windows.Forms.ScrollableControl

System.Windows.Forms.ContainerControl

System.Windows.Forms.Form



## Form 속성 → get, set,

- Handle  
윈도우 **핸들** → 고위번호 (값)
- Text  
컨트롤 또는 윈도우 타이틀바에 텍스트 출력 or 입력
- Visible  
폼 또는 컨트롤의 출력 여부 결정
- Enable  
비활성 또는 활성
- ClientRectangle  
클라이언트 크기 영역을 가져오는 역할

# 메서드

- Invalidate() ✓  
특정 영역 또는 무효화 영역을 다시 출력하는 역할
- Update() ✓  
현재 무효화 영역만 다시 출력
- Activate() ✓  
폼을 활성화
- Close() ✓ → WM\_CLOSE → WM\_DESTROY  
폼을 닫는다
- ShowDialog()  
폼을 모달 대화상자로 출력