

C# 시작하기

HyoJoon Han
동국대학교
han6343@dongguk.edu

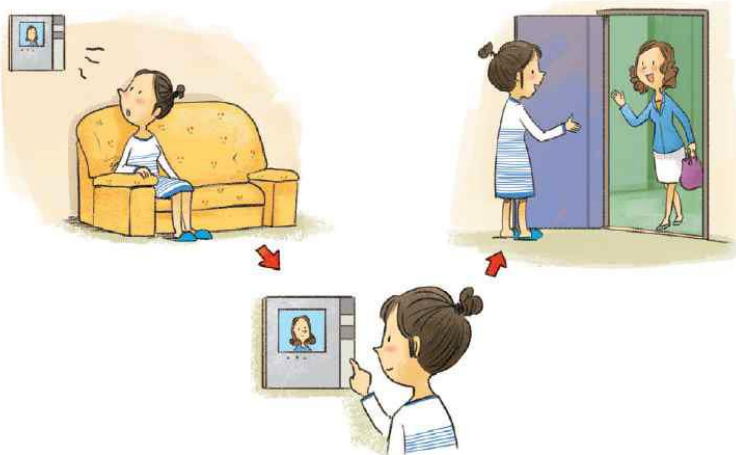


프로그래밍 기초 용어

C# 맛보기

- 이벤트 기반 프로그래밍
 - 절차형(순차적) 프로그래밍 방식
 - 도스 DOS 운영체제에서 사용
 - 프로그램을 작성할 때 명령어의 처리 순서를 고려
 - 순서들을 조금씩 구조화해서 함수라는 형태로 정의하고 호출해 사용
 - 이벤트 기반(처리형) 방식
 - 이벤트는 프로그램에서 감지한 어떤 행동이나 발생한 사건
 - 사용자가 발생시킨 이벤트에 따라 미리 정해진 명령을 처리

그림 2-1 이벤트 기반 프로그램과 손님 방문의 비교 예



- 1) '초인종이 울린다' 라는 이벤트 발생
- 2) 아는 사람인지 인터폰으로 확인하는 행동을 한다.
- 3) 아는 사람이면, 문을 열어 주는 행동을 한다.

- 개체와 컨트롤

- 개체

- 우리가 살아가는 세계에 실제로 존재하는 것과 생각할 수 있는 것
 - 특징, 기능(동작), 행위로 표현할 수 있다.

- 컨트롤

- C#에서는 도구 상자에 포함된 텍스트 박스, 리스트 박스 등의 각 컨트롤이 개체에 해당
 - 개체는 특징에 해당하는 속성(Property), 기능인 메소드(Method), 행위인 이벤트를 갖는다.
 - 동일한 컨트롤로 만든 개체는 기본적인 값은 같지만 각 개체의 특성에 맞게 변경할 수 있다.
 - 표준 컨트롤과 ActiveX 컨트롤이 있다.

실세계의 개체

자동차, 자전거 등

특징	흰색	자 동 차
기능	전진	
행위	운전	

C#의 개체

컨트롤(버튼, 텍스트박스 등)

속성	Caption	버 튼
메소드	SetFocus	
이벤트	Click	

그림 2-3 개체의 속성, 이벤트, 메소드

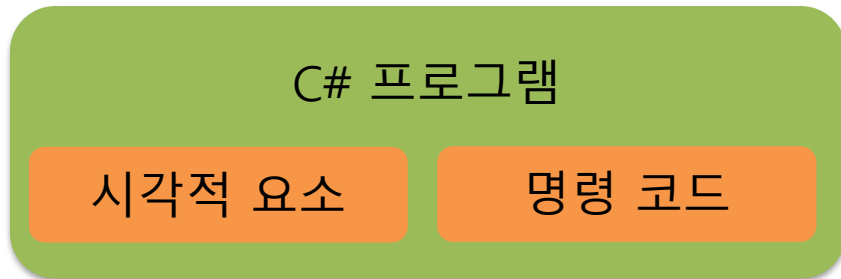


- 속성
 - 개체의 색상, 크기, 위치와 같은 외형을 정의하는 특성
 - 개체마다 속성이 다르다.
 - 레이블 – 제목용으로 사용되어 글자를 표시하는 속성이 대부분이므로 그림 속성이 없다.
 - 픽처 박스 – 그림을 넣는 컨트롤이므로 그림에 관한 속성이 다양하지만 글자에 대한 속성이 없다.
- 메소드
 - 개체가 수행하는 동작이나 기능을 정의
 - 각 컨트롤의 기능에 따라 정의된 메소드가 다르다.
 - C#에서는 각 컨트롤이 가진 속성과 메소드를 이용하여 어떤 기능을 하게 만드는 작업이다.

- 이벤트
 - 개체가 사용자나 시스템으로부터 받는 명령
 - 개체가 반응할 수 있는 특정 사건이다.
 - 사용자가 마우스를 움직이는 행위, 키보드를 문자를 입력하는 행위, 버튼을 클릭하는 행위 등
- 프로시저
 - 이벤트를 처리하는 단위로, 어떤 행동을 수행하기 위한 일련의 작업 순서
 - 특히 이벤트 프로시저는 사용자가 이벤트를 발생시키면 이를 처리하기 위해 자동으로 호출된다.
 - 프로시저는 이벤트를 처리하기 위한 메소드와 변수 제어문 등을 포함
 - 이벤트 프로시저 – 이벤트가 발생했을 때 자동으로 호출되는 프로시저

- C# 프로그램의 구조와 작성 과정
 - 화면(인터페이스)을 만들고 각 개체의 속성을 설정한 후 코드를 작성하여 프로그램을 완성한다.
 - 비주얼이라는 시각적 요소와 이 시각적 요소를 마우스로 클릭했을 때 어떤 동작을 하는 명령 코드로 구성된다.

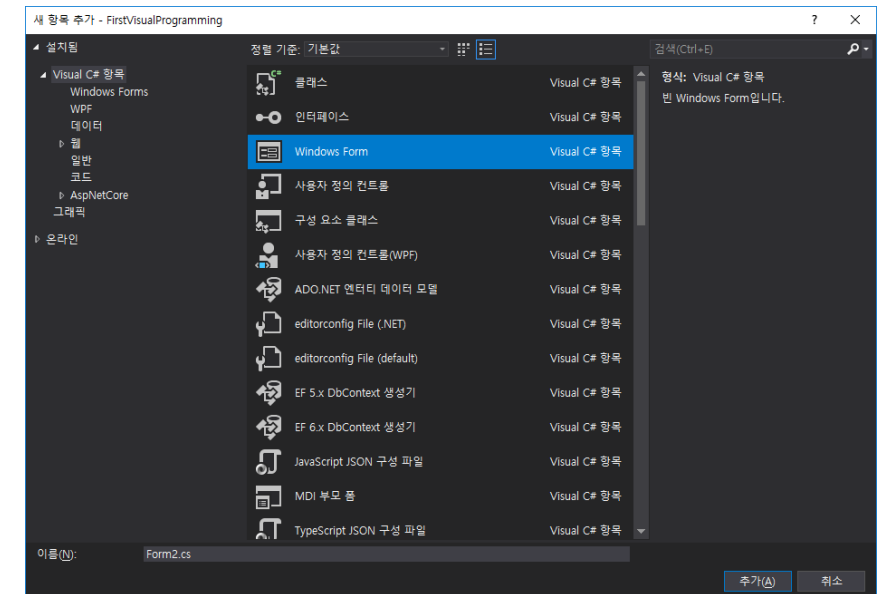
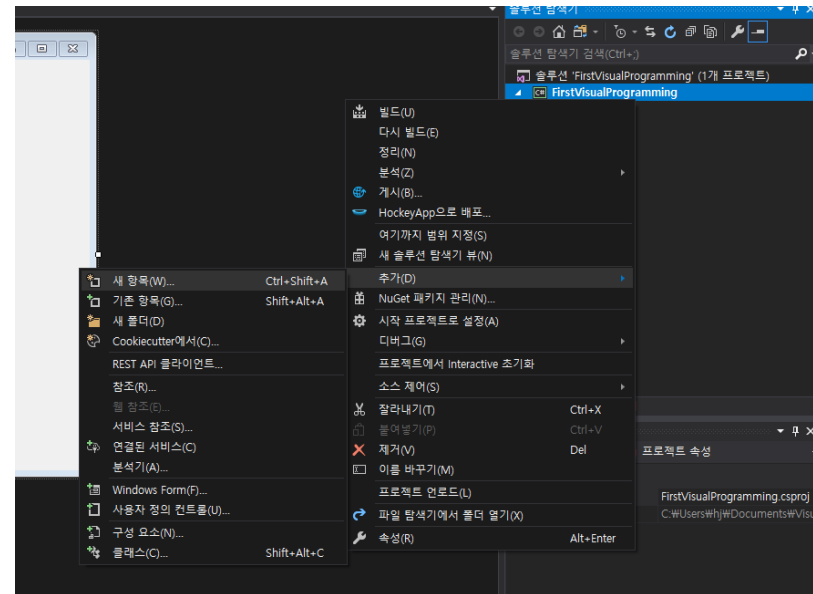
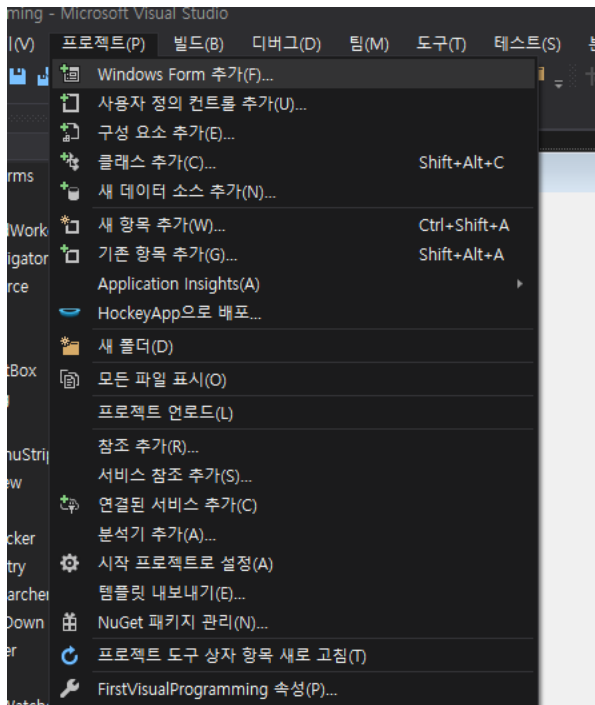
<그림> C# 프로그램의 구조



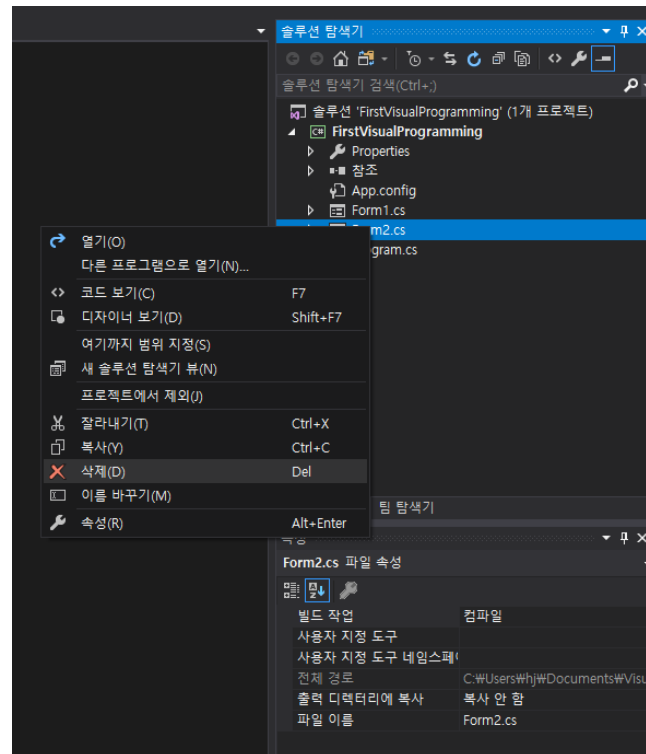
<그림> C# 프로그램 작성 과정



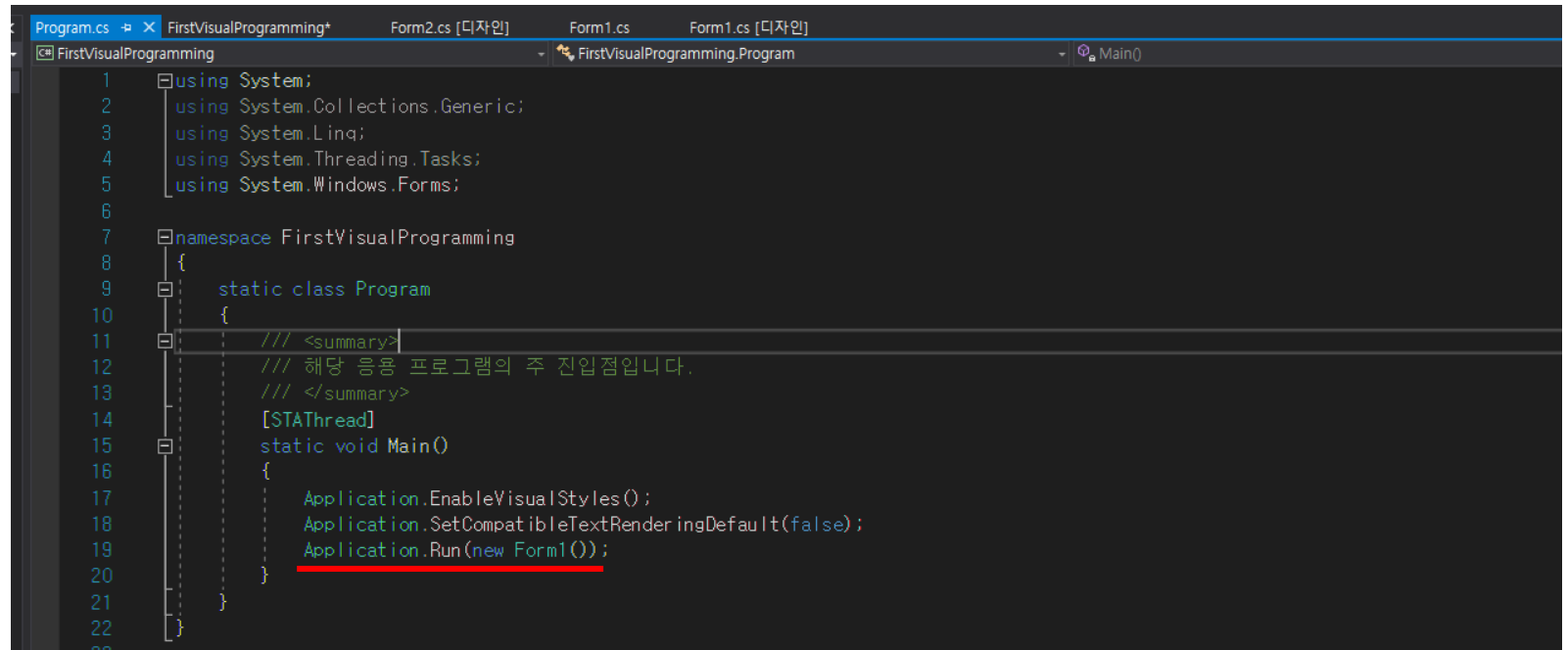
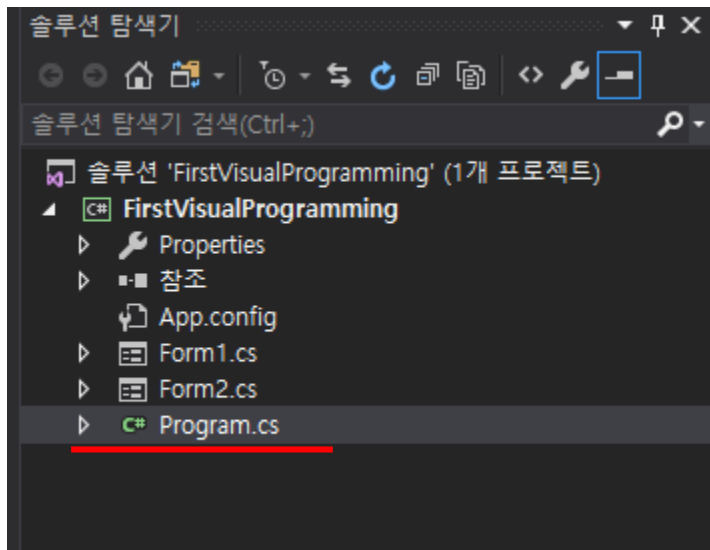
- 폼 기본 조작법 익히기
 - 프로젝트당 하나의 폼이 기본으로 생성되며 추가 및 삭제가 가능하다
 - 폼 추가 및 삭제하기
 - [프로젝트] - [Windows Form 추가] 또는 [프로젝트] - [새 항목 추가]를 클릭하거나, 솔루션 탐색기에서 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭하여 [추가] - [새 항목]을 선택한다.



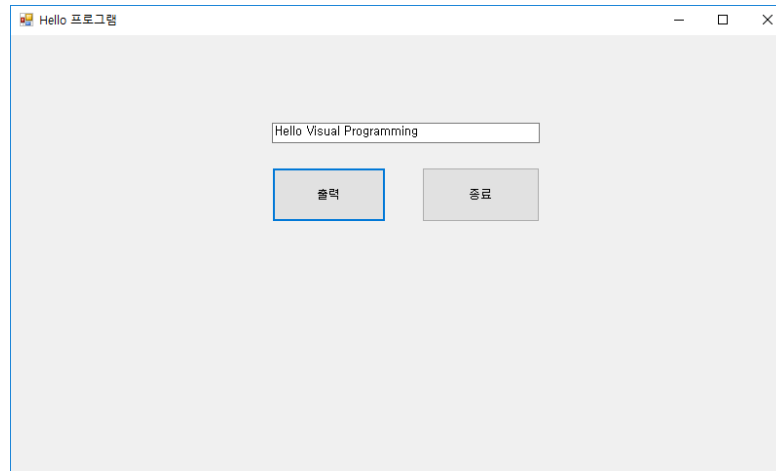
- 폼 기본 조작법 익히기
 - C#은 프로젝트를 구성하는 모듈, 클래스 등의 구성 요소를 별도로 저장하여 폼도 독립적이다.
 - 폼을 삭제하고 싶다면 솔루션 탐색기에서 해당 폼을 선택하고 마우스 오른쪽 버튼을 클릭하여 [삭제]를 선택한다.



- 폼 기본 조작법 익히기
 - 시작 폼 설정하기
 - 솔루션 탐색기에서 'Program.cs' 를 더블클릭한다.
 - 'Application.Run()' 메소드 안의 폼 이름을 적절하게 바꿔준다.

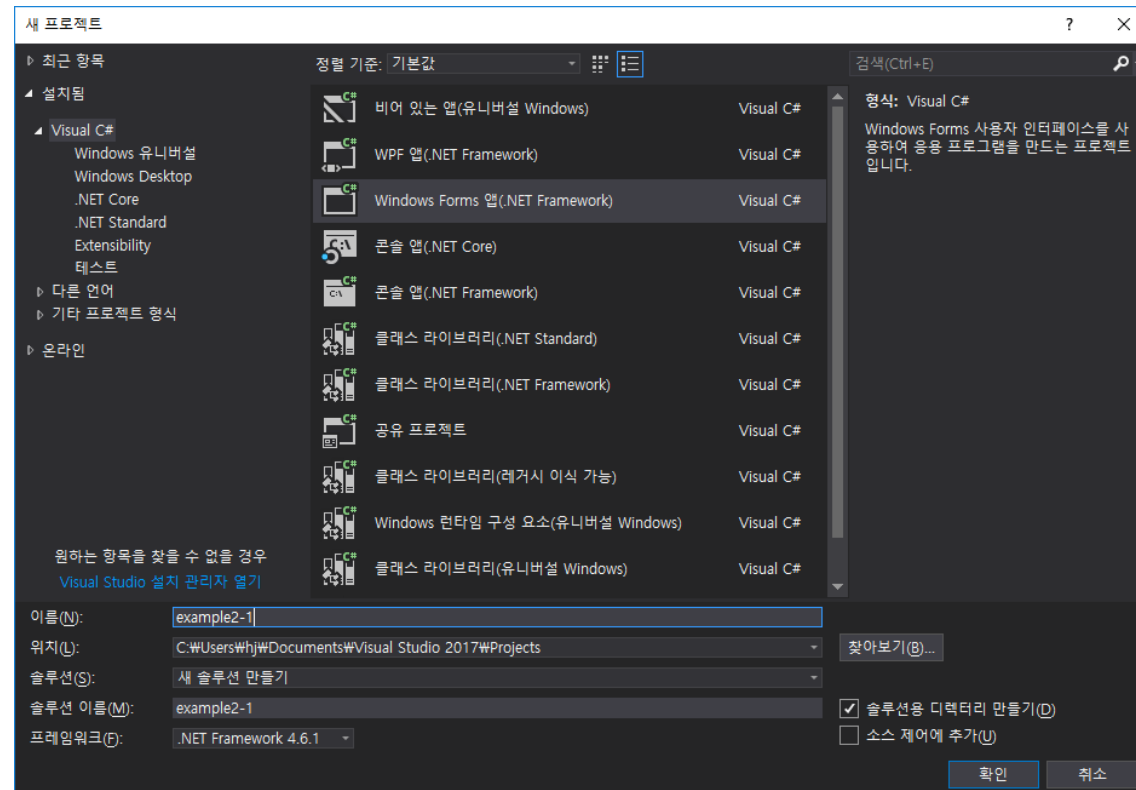


- 'Hello Visual Programming' 메시지를 출력하는 프로그램 작성하기
- 실행 결과

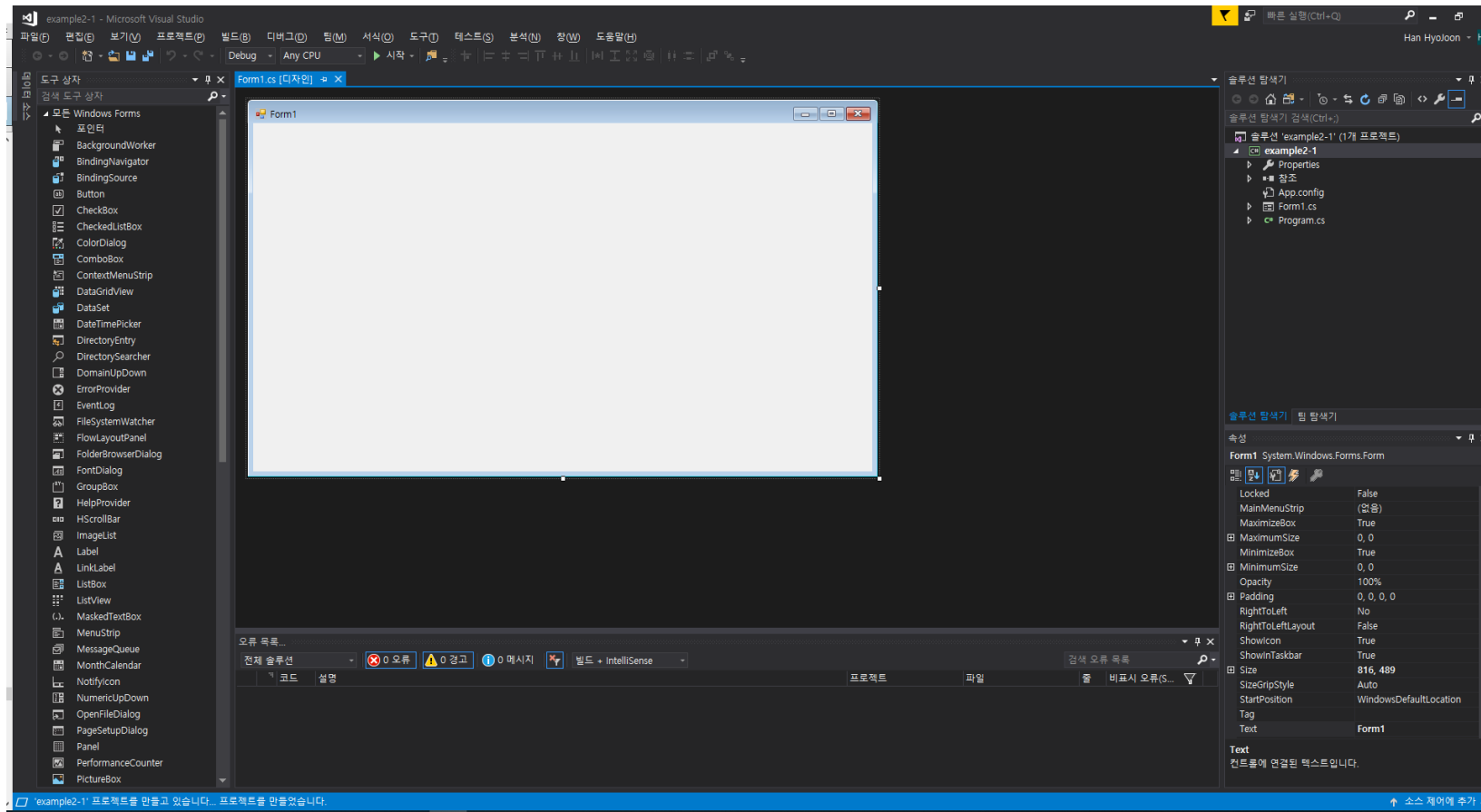


- 1) Visual Studio 실행 : [시작] – [모든 프로그램]을 선택한 후 Visual Studio 2017를 실행한다.
또는 바탕화면의 Visual Studio 2017 아이콘을 더블클릭한다.

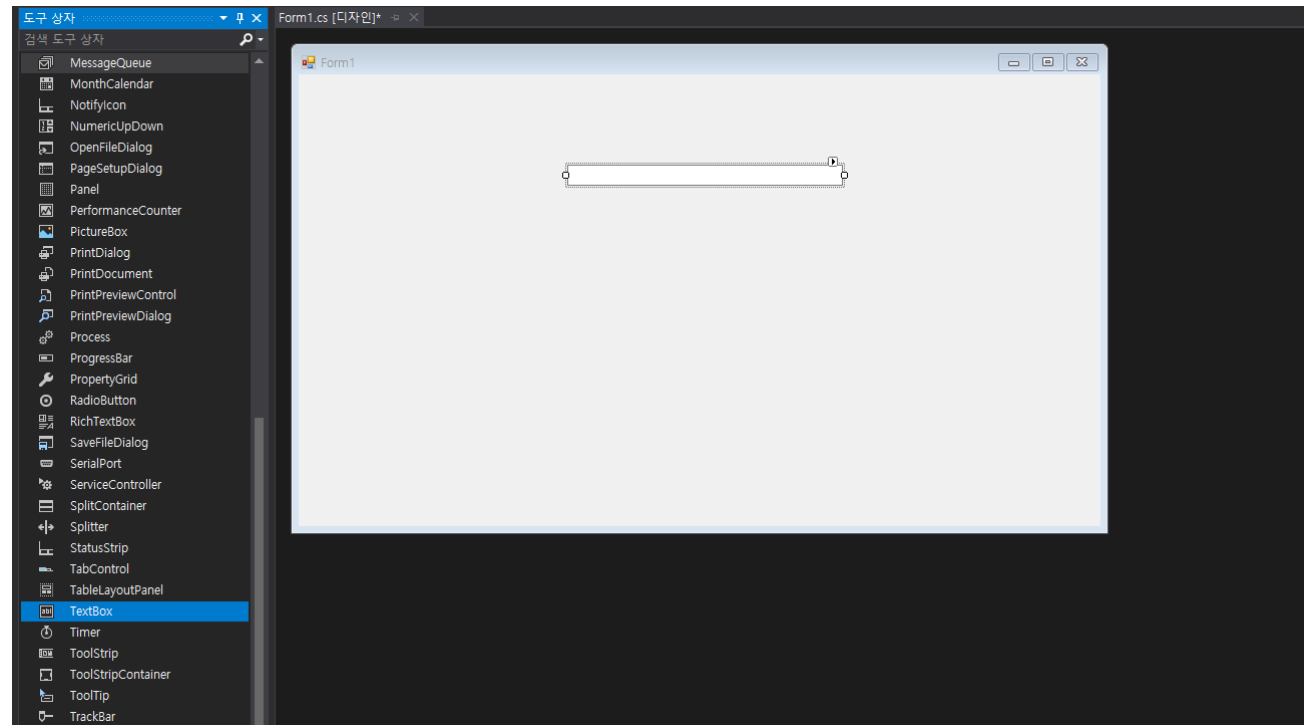
- 2) 새 프로젝트 시작하기 : [파일] - [새 프로젝트]를 선택한다.
- 3) 새 템플릿 시작 하기 : [새 프로젝트] 대화 상자에서 [Visual C#] 항목의 [Windows Forms 앱 (.Net Framework)] 을 선택하고 프로젝트 이름을 입력하고 확인을 클릭한다.



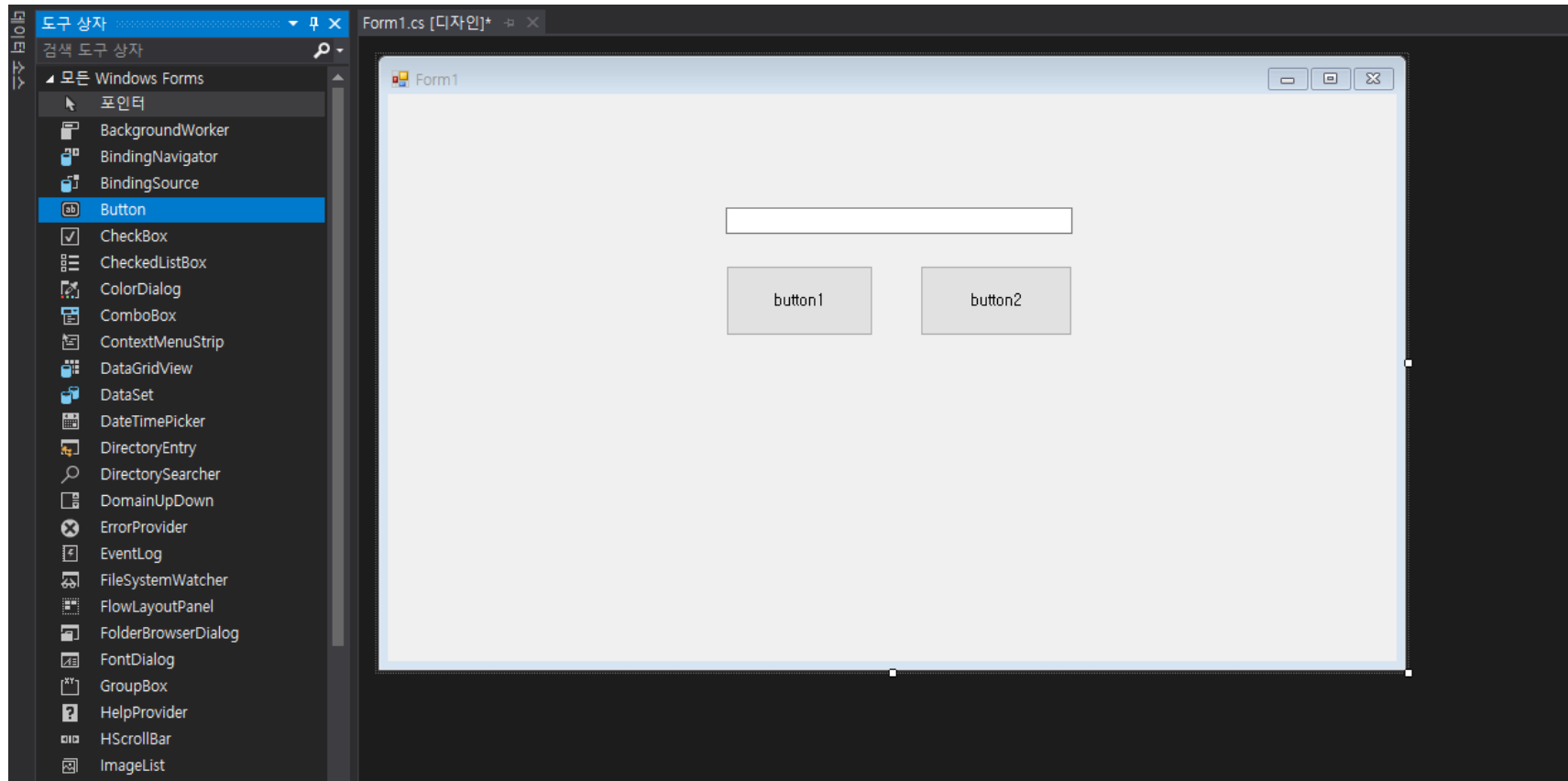
- 4) Windows Forms 응용 프로그램 작업 화면 : 다음 그림과 같은 응용 프로그램 작업 화면이 나온다.



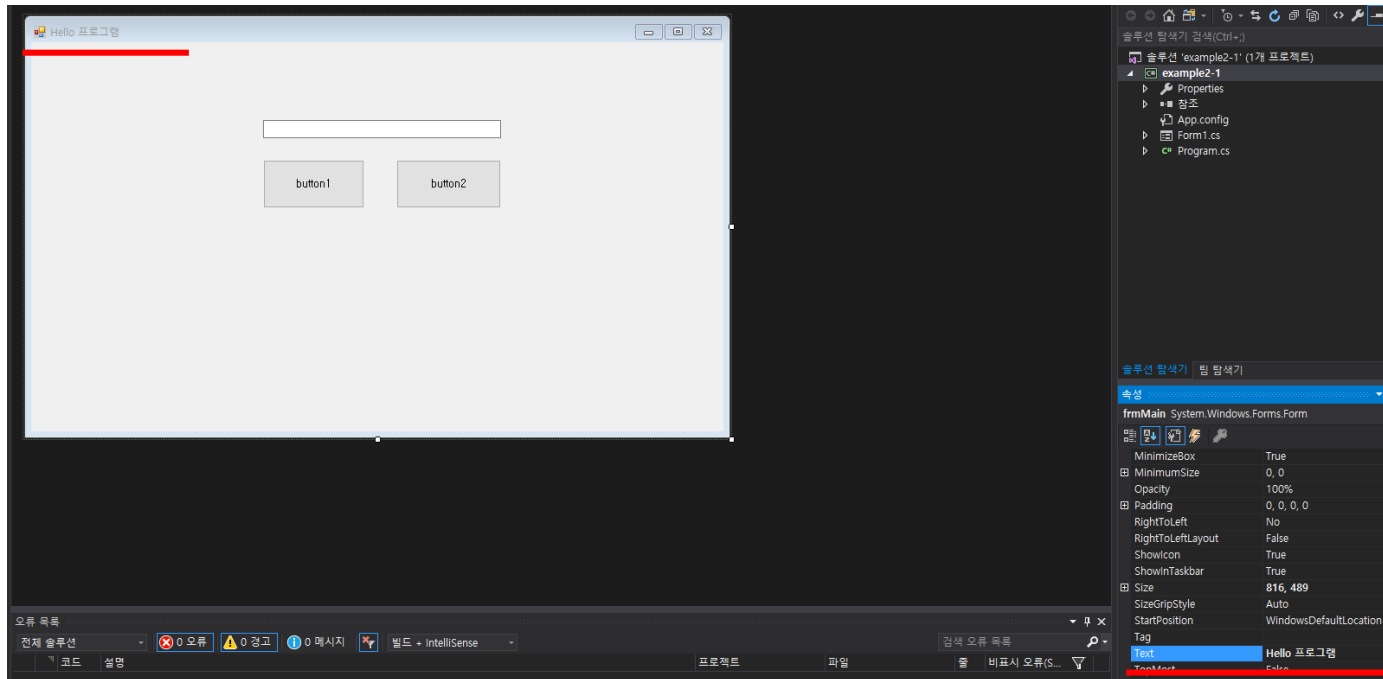
- 5) 윈도우 폼 디자인-컨트롤 배치하기 : 도구 상자에서 텍스트 박스 컨트롤을 마우스로 클릭하여 선택하고 마우스를 폼 위로 옮기면 포인터가 십자 모양(+)이 된다. 이 상태로 폼위에서 마우스를 드롭다운하여 텍스트 박스를 생성한다. 마우스를 텍스트 박스 컨트롤의 양쪽 끝으로 옮기면 양쪽 화살표 모양(↔) 이 된다. 이상태로 드래그 하여 텍스트 박스 컨트롤의 크기를 조절할 수 있다.



- 6) 윈도우 폼 디자인-컨트롤 배치하기 : 5번 과정을 반복하여 다음과 같이 버튼 2개를 추가하고 크기를 조절한다.



- 7) 윈도우 폼 디자인-컨트롤 속성 지정하기 : 각 컨트롤의 속성에는 이미 기본 값이 들어 있는데 이를 원하는 값으로 변경할 수 있다. 먼저 폼을 마우스로 클릭하여 선택하고, 속성 창에서 Name 속성 값과 Text 속성 값을 다음 표와 같이 변경한다. 이 값이 바로 반영되어 폼의 제목 표시 줄 내용이 변경되어 나타난다.



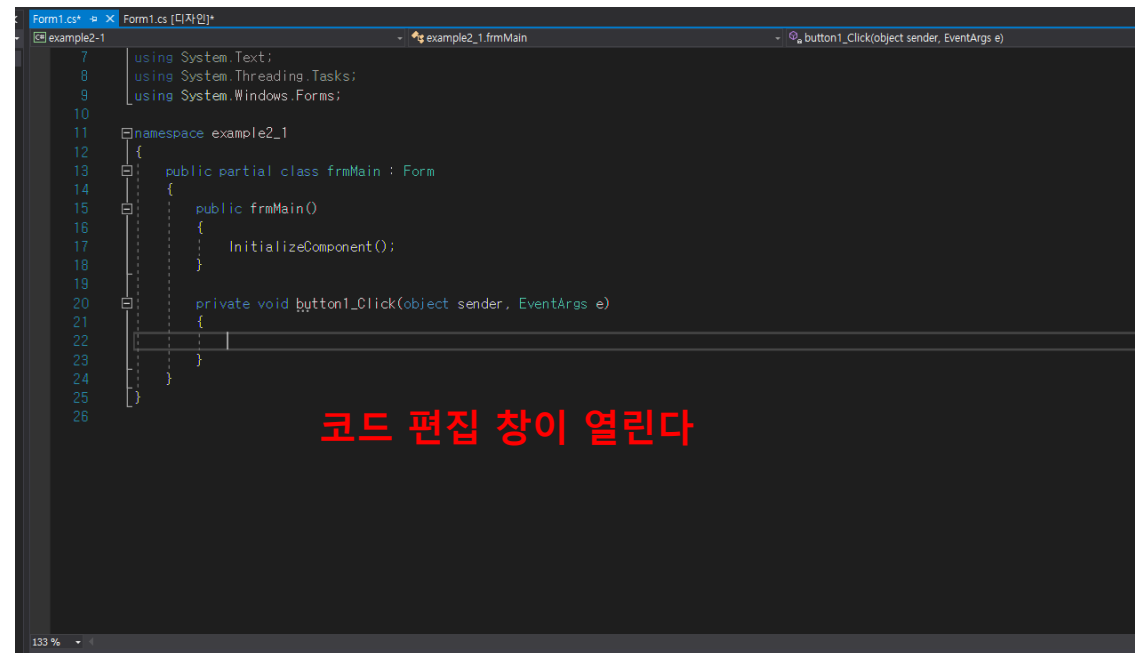
컨트롤	속성	속성 값
① Form	Name	frmMain
	Text	Hello 프로그램

- 8) 윈도우 폼 디자인-컨트롤 속성 지정하기 : 7번의 방법으로 폼, 텍스트 박스와 커맨드 버튼 컨트롤의 속성을 다음과 같이 설정한다.

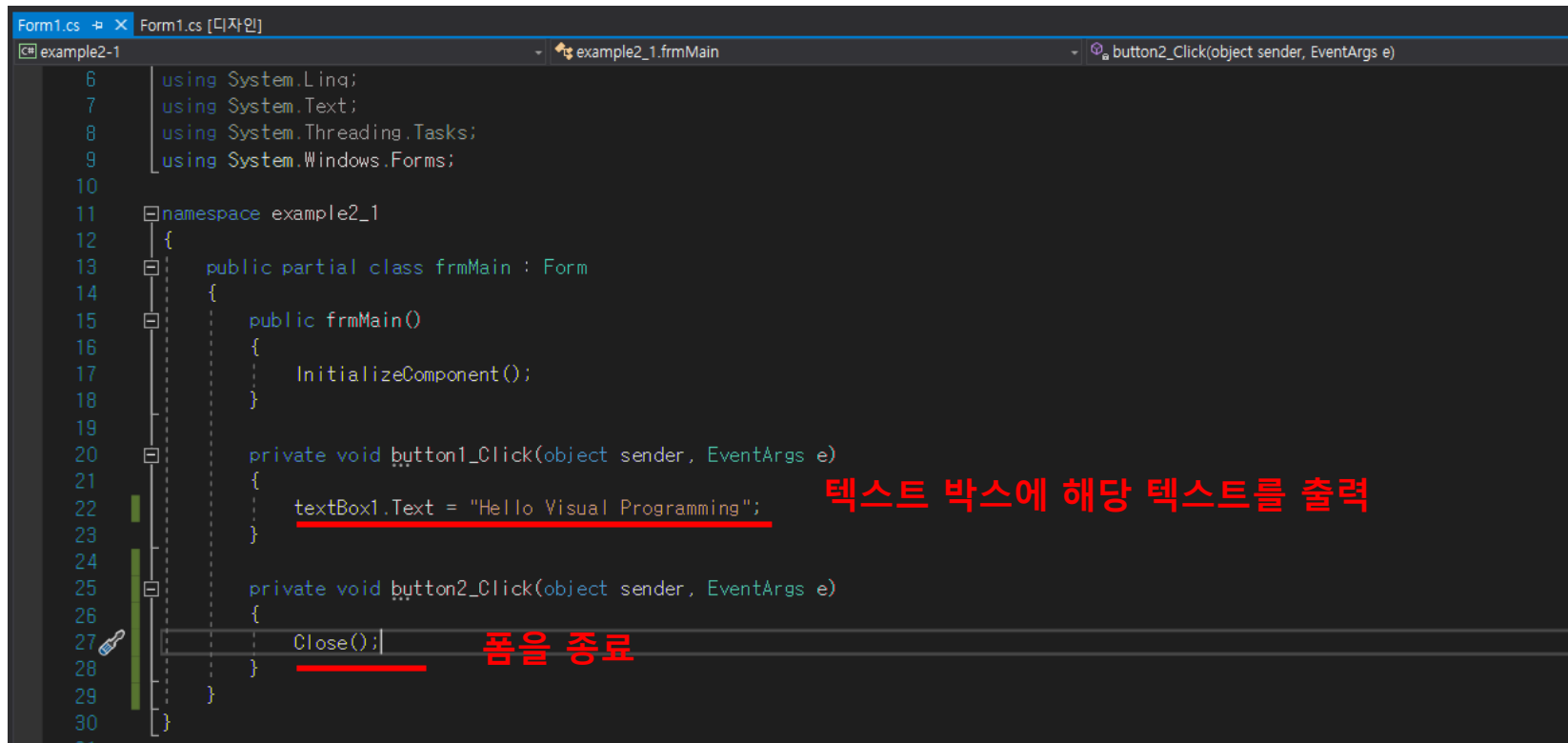


컨트롤	속성	속성값
① Text1	Name	TextBox1 (기본값 사용)
	Text	(빈칸)
② Button1	Name	Button1 (기본값 사용)
	Text	출력
③ Button2	Name	Button2 (기본값 사용)
	Text	종료

- 9) 프로그램 코드 편집 창 열기 : <출력> 버튼을 클릭하면 텍스트 박스에 'Hello Visual Programming'을 출력하도록 하는 프로그램을 작성한다. 먼저 <출력> 을 더블클릭하여 코드 편집 창을 연다. 또는 <출력>을 선택한 상태에서 마우스 오른쪽 버튼을 누르고 [코드 보기] 메뉴를 선택해도 된다.



- 10) 프로그램 코드 작성하기 : <출력>을 더블클릭한 다음 코드를 작성한다. 같은 방법으로 <종료>를 더블클릭한 다음 코드를 작성한다.



```
Form1.cs [디자인]
example2-1
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace example2_1
{
    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            textBox1.Text = "Hello Visual Programming";
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }
    }
}
```

텍스트 박스에 해당 텍스트를 출력

폼을 종료

- 11) 프로그램 실행하기 : [디버그] - [디버깅 시작] 또는 F5나 단축 아이콘에서 실행 버튼을 누르면 다음과 같은 실행 창이 뜬다. 결과 화면 에서 <출력>을 클릭하면 텍스트 박스에 텍스트가 출력되고, <종료>를 클릭하면 실행 프로그램이 종료된다.

