고급 프로그래밍 언어 및 실습

C# (환경구축)

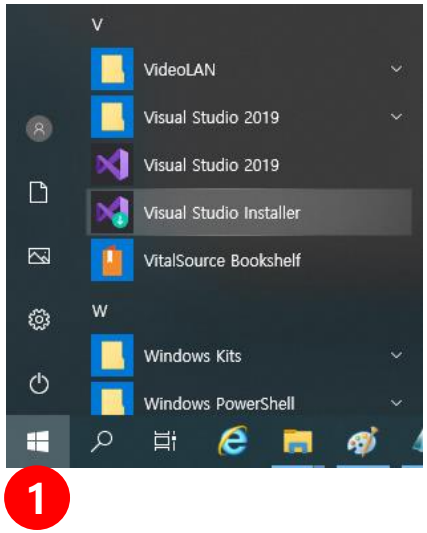
Ch1, Sec4. 실습환경구축

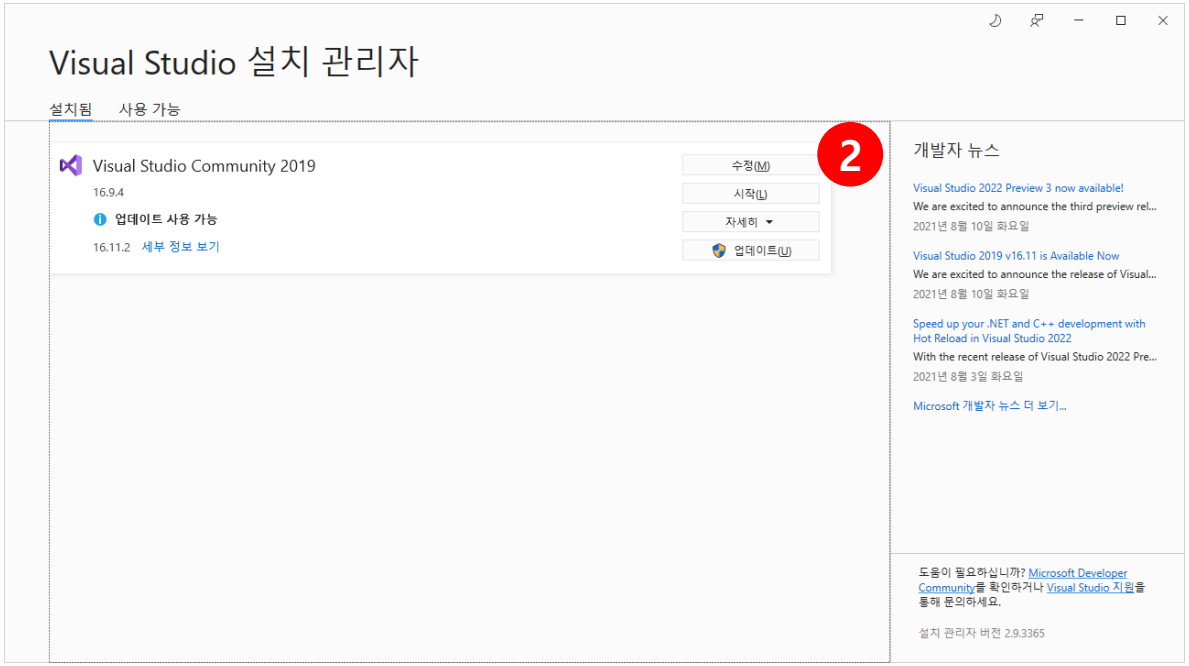
1. C# 실행 환경 구축
2. C# 프로젝트를 생성하고 코드를 실행
3. 오류 확인

C# 실행 환경 구축

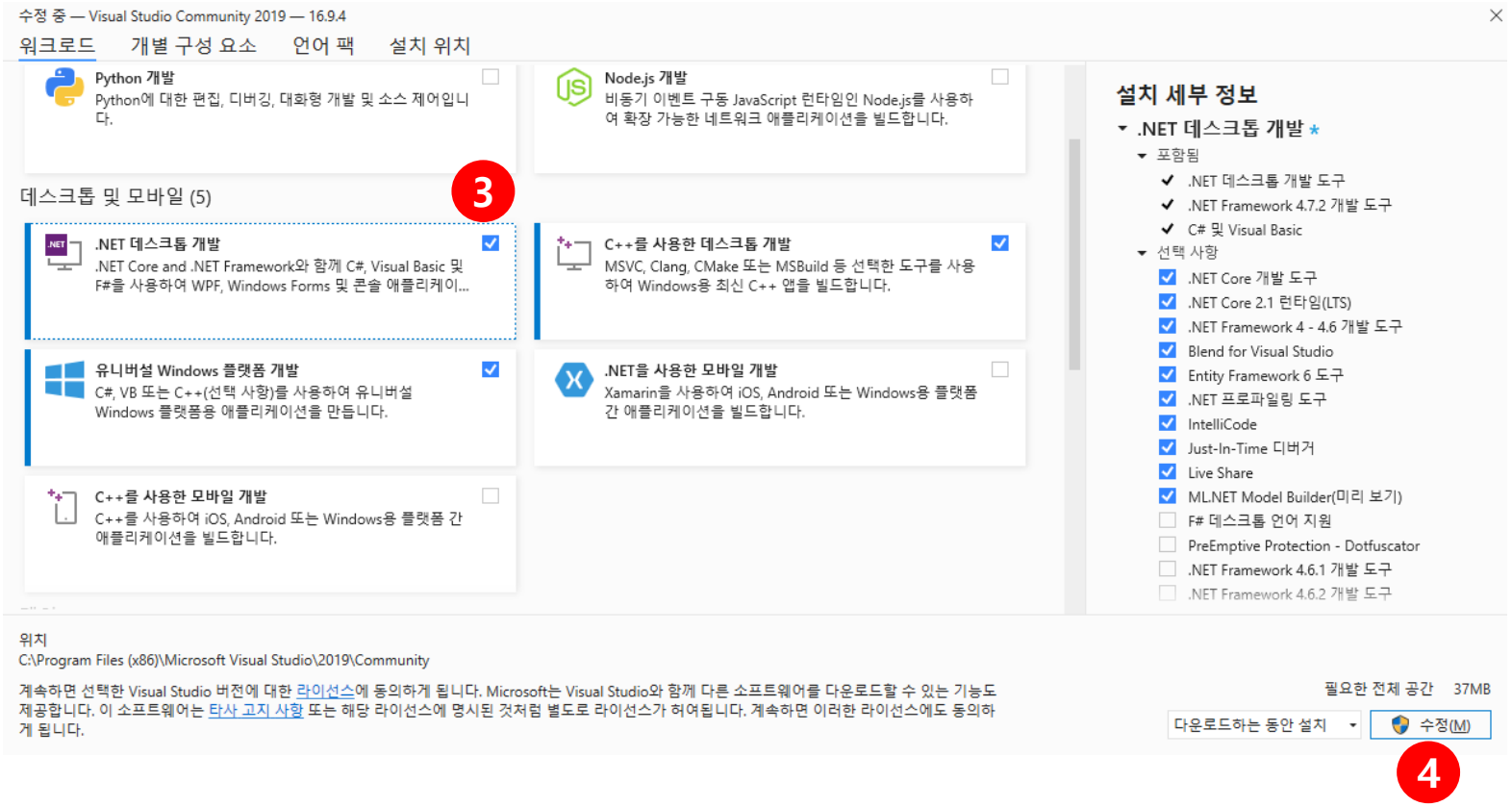
- **개발 환경 설치**
  - 비주얼 스튜디오 사용 Visual Studio 2019

(1) 기존에 비쥬얼 스튜디오가 설치되어 있는 경우 (C언어)

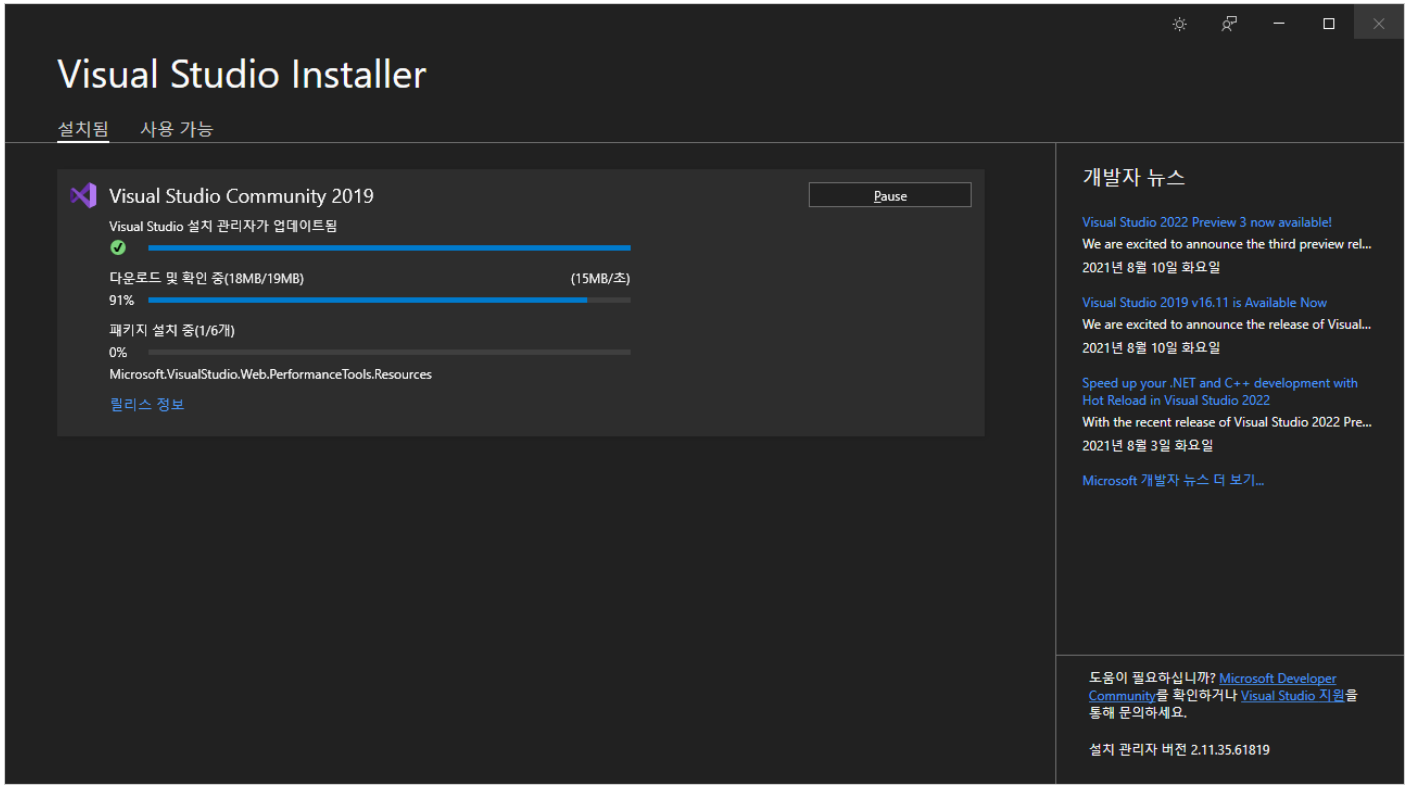




- 워크로드 [데스크톱 및 모바일]에서 [.NET 데스크톱 개발]을 클릭하여 체크
- [수정] 버튼 클릭

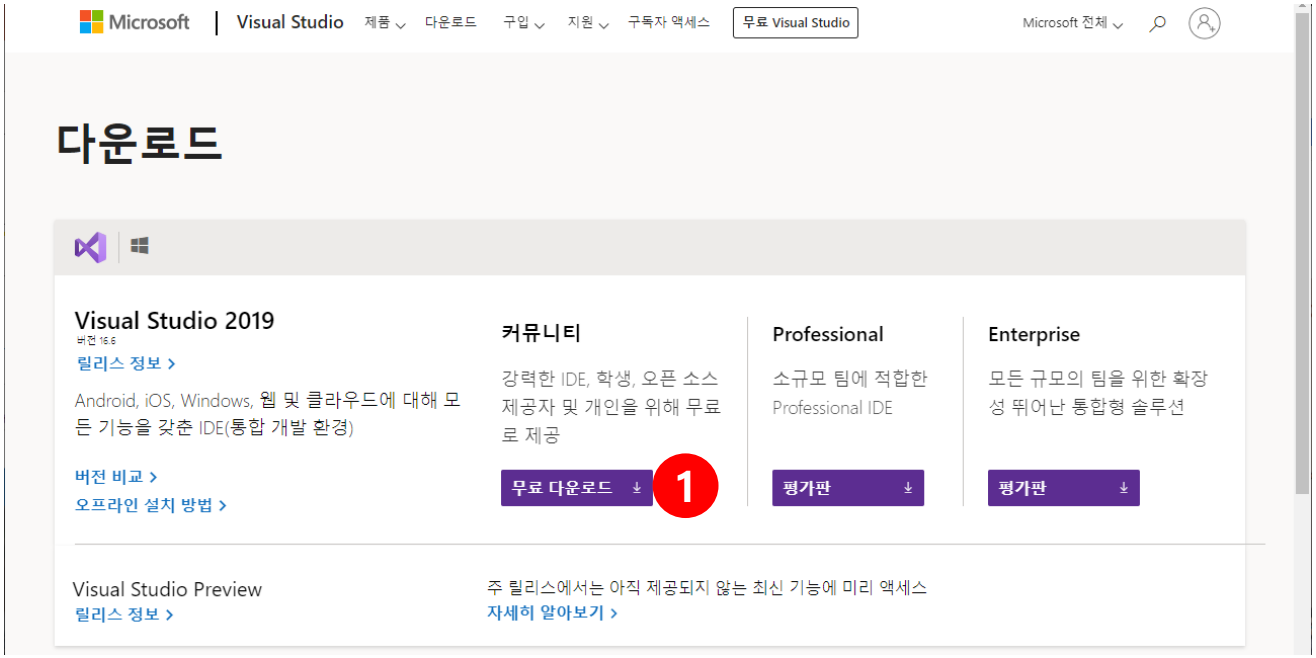


- 설치관리자 업데이트

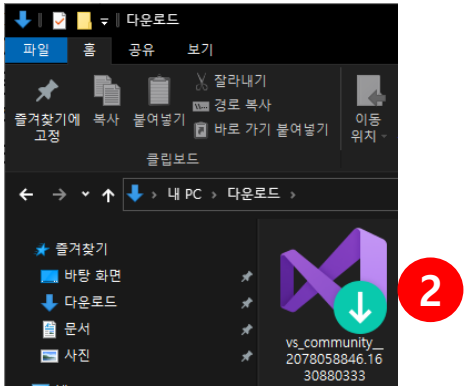


(2) 비쥬얼 스튜디오가 설치되어 있지 않은 경우

- Visual Studio 2019 Community Download
  - https://visualstudio.microsoft.com/ko/downloads/



- 설치파일을 더블클릭하여 설치



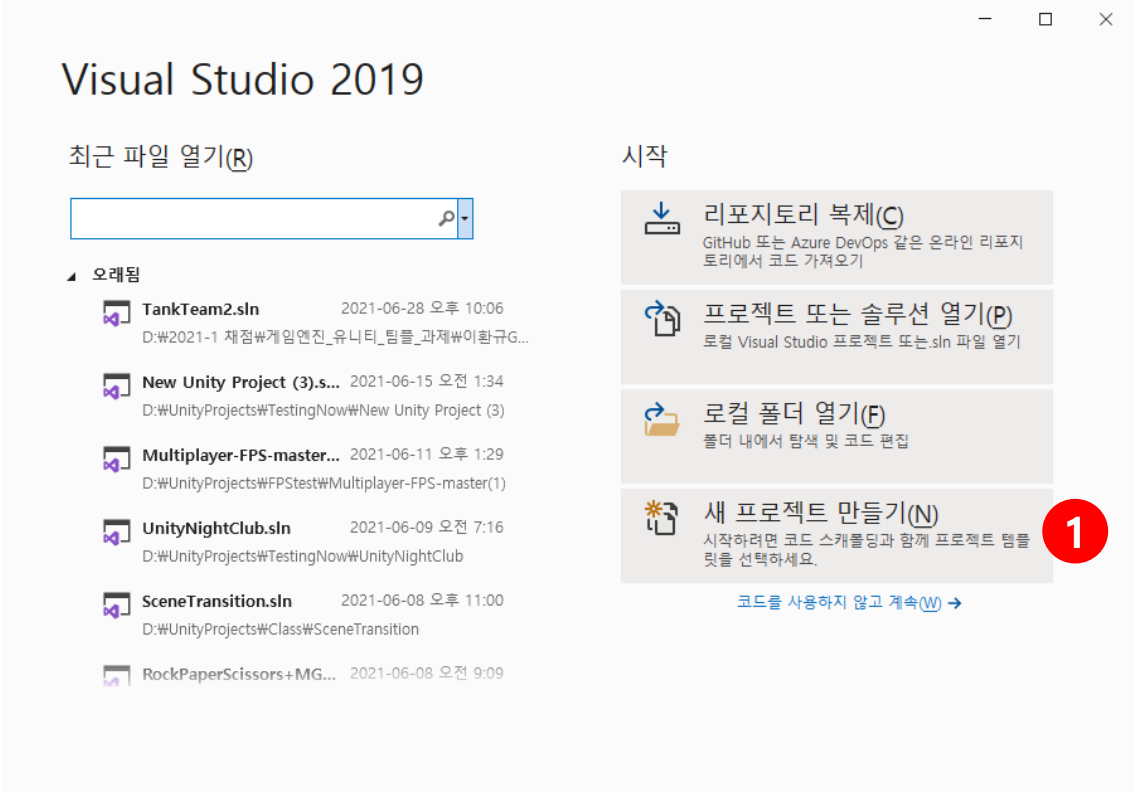
  - 설치 시, 워크로드 선택 창에서 [.NET 데스크톱 개발]을 클릭하여 체크 (앞 슬라이드 참조)

C# 프로젝트를 생성하고 코드를 실행

- **프로젝트 생성**

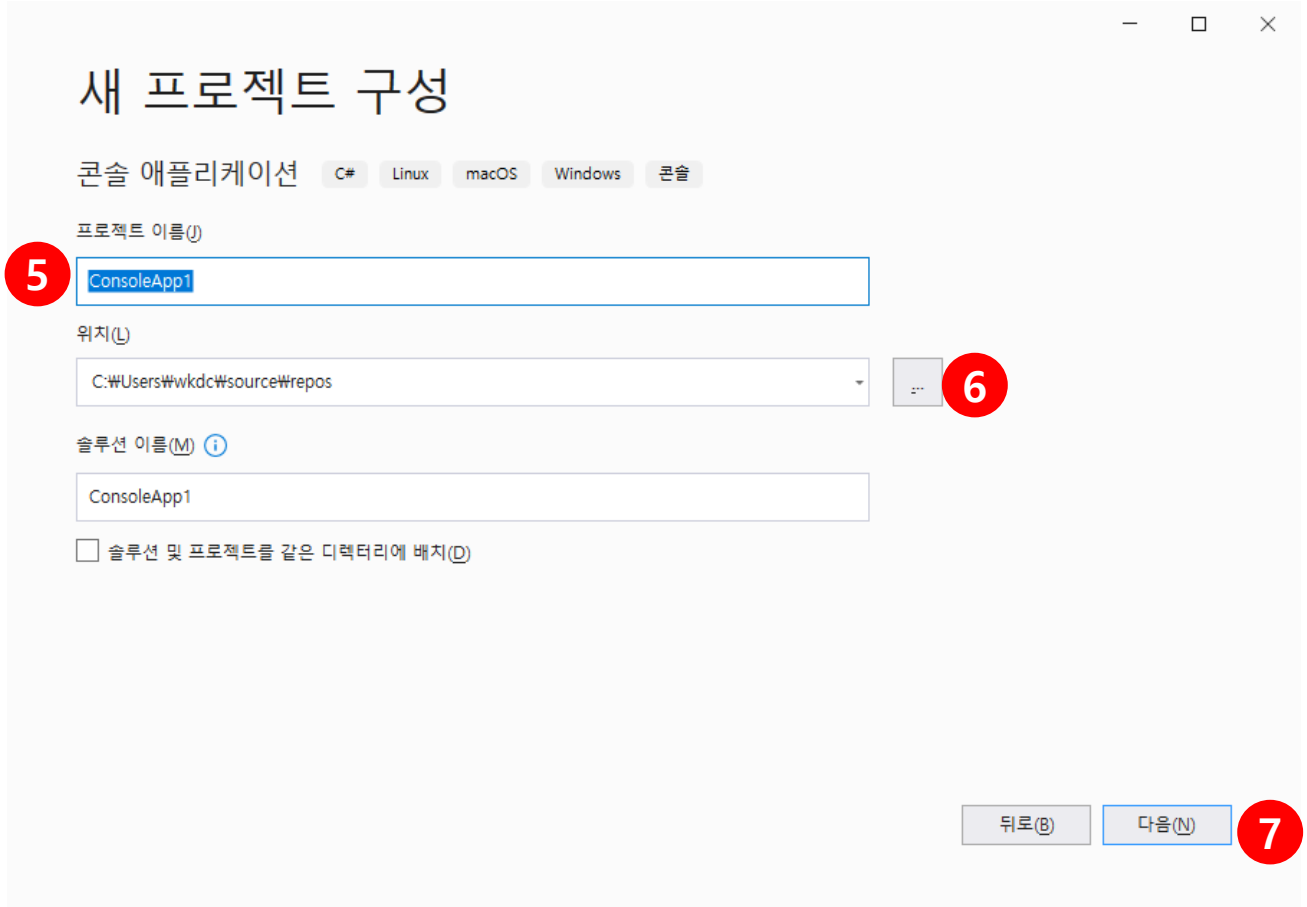
  ① [파일] - [새 프로젝트] 메뉴 선택

  ② [새 프로젝트] 대화상자 [새 프로젝트 만들기] 클릭

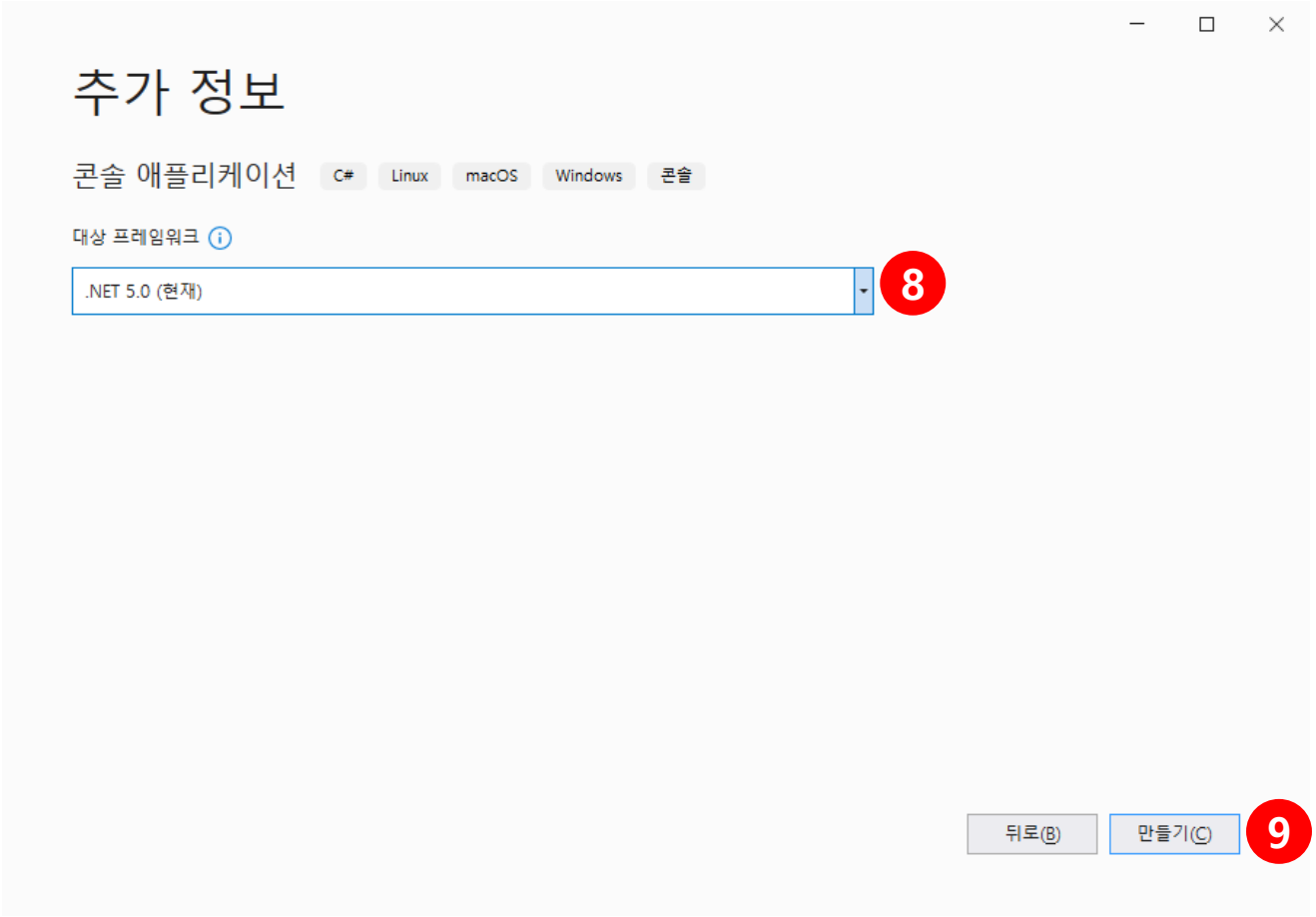


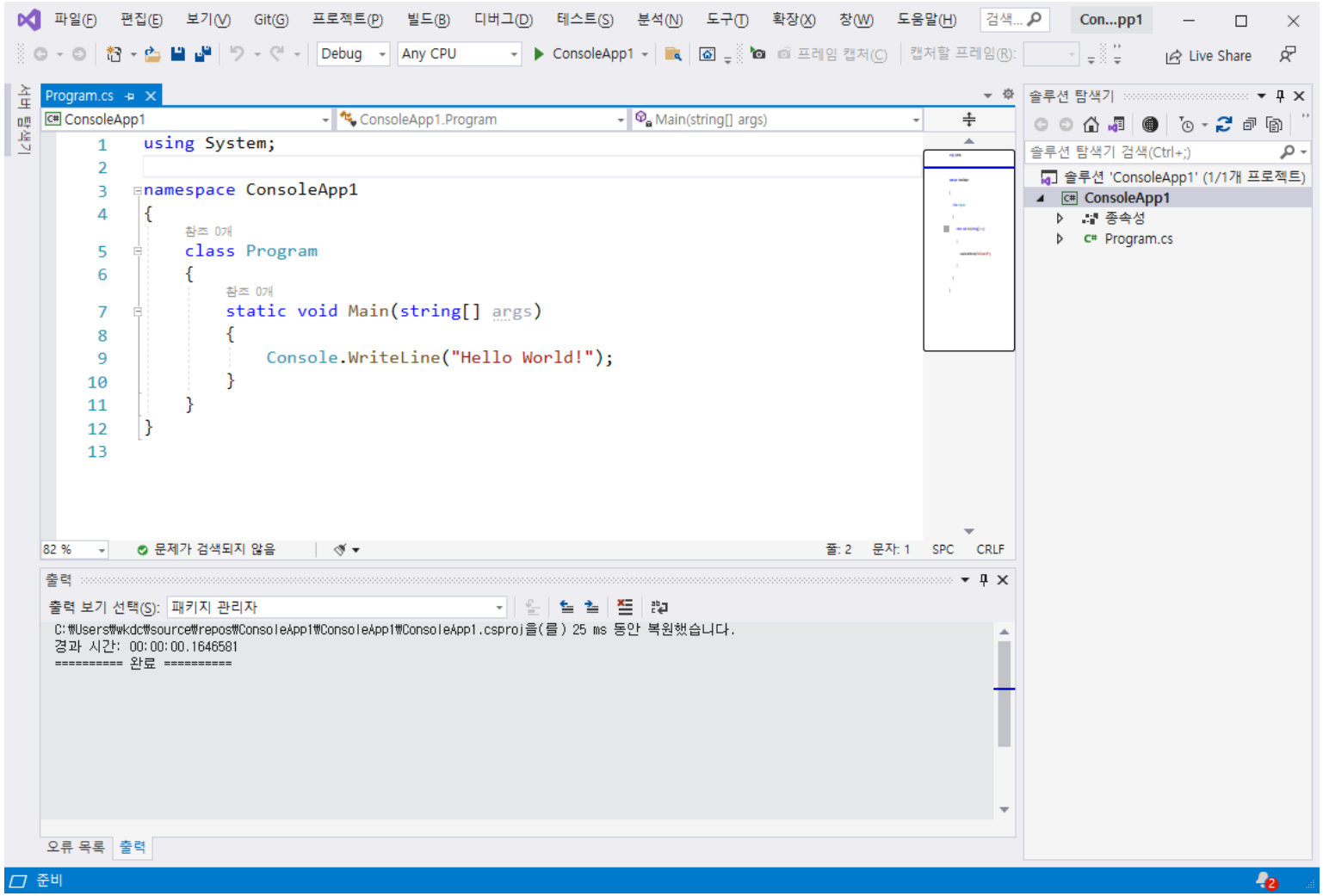
- C# 언어를 고르고 템플릿 프로젝트를 선택한 다음 [다음] 버튼 클릭

- 프로젝트 이름을 입력하고 저장 위치를 선택하고 [다음] 버튼 클릭



- 대상 프레임워크를 선택하고 [만들기] 버튼 클릭





- **프로젝트 실행**
  - 방법1 : [시작] 버튼 또는 [디버그] - [디버깅 시작] 메뉴 또는 단축키(F5) 누름
    - 프로그램이 한 번 실행되었다가 바로 종료되어 아무것도 안보임
    - 디버그 모드 실행(오류 등을 자세히 확인하기 위해 사용되는 모드)

    오류 없이 정상적으로 프로그램이 끝났다고 확인되면, 실행된 화면 꺼버려 실행 결과 보기 어려움

  - 방법2 : 콘솔 응용 프로그램 예제를 진행 시 [디버그] - [디버그하지 않고 시작] 메뉴 또는 Ctrl + 단축키(F5) 누름

- 예제 프로그램

네임스페이스

[예제 1.1 – HelloWorld.cs]

```
using System;
class HelloWorld {
    public static void Main() {
        Console.WriteLine("Hello World!");
    }
}
```

출력 메소드

실행 결과 :

```
Hello World!
```

- 실행 방법

```
C:\temp>csc HelloWorld.cs
C:\temp>HelloWorld
Hello World!
```

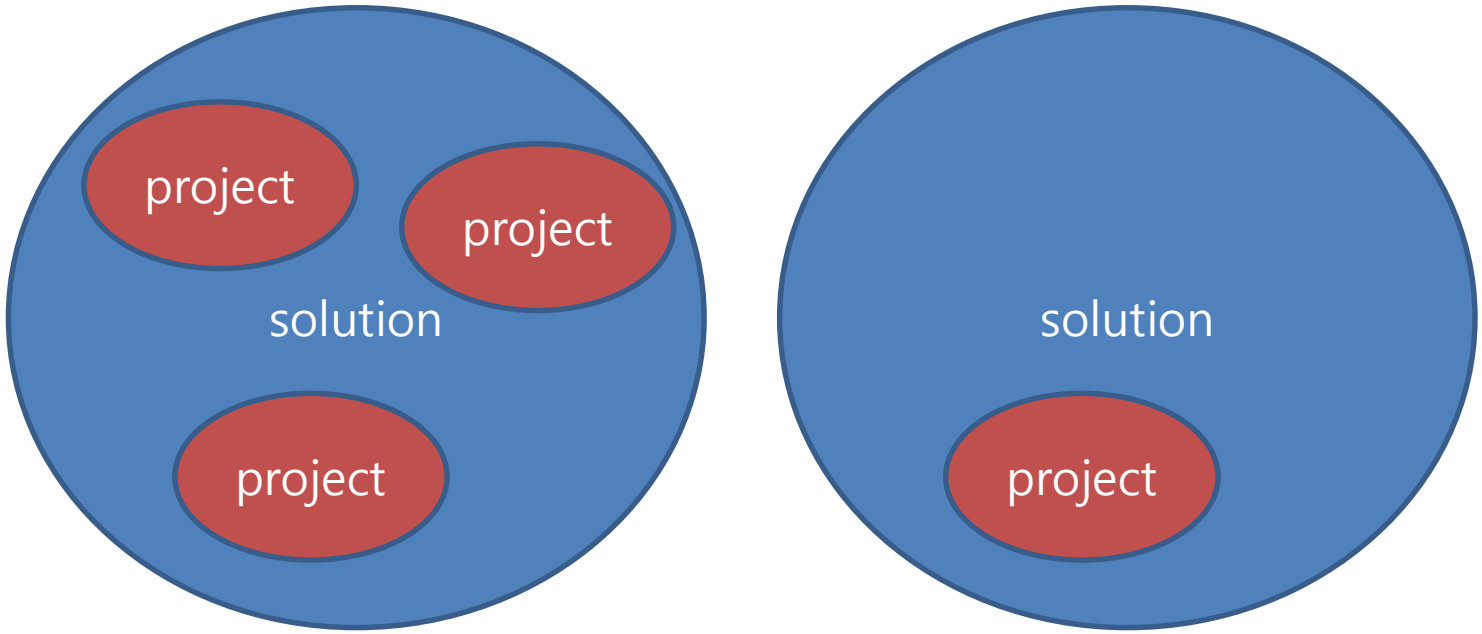
- **첫번째 프로젝트**

```
Using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FirstProgram
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Hello World .. !");
      }
   }
}
```

```
선택 Microsoft Visual Studio 디버그 콘솔
Hello World!
```

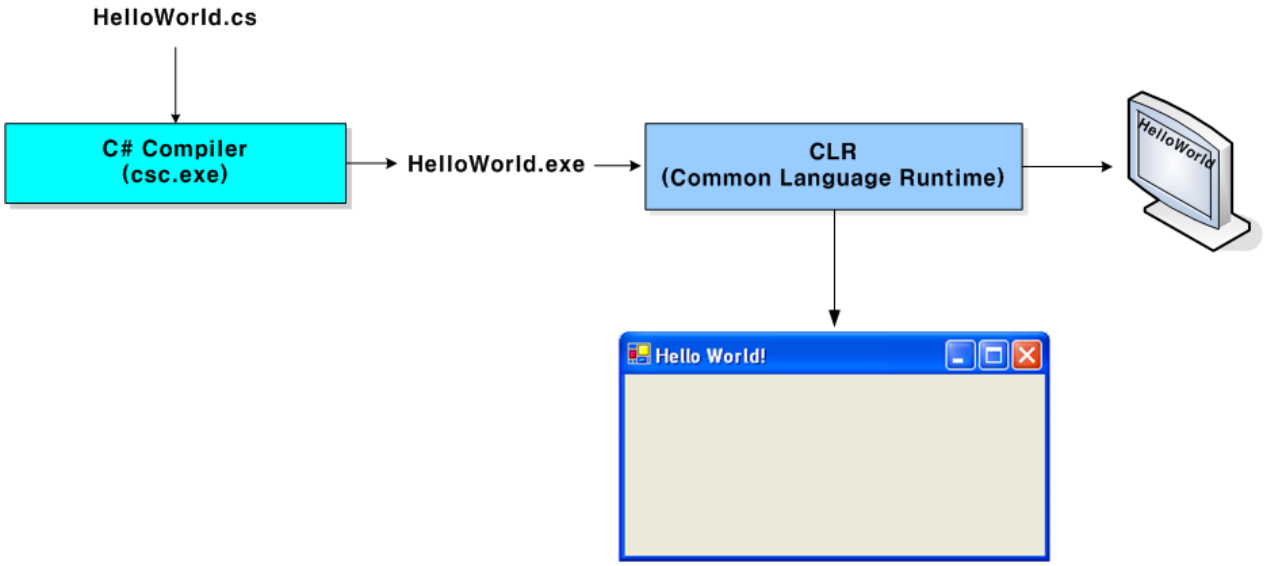
– Solution VS. Project



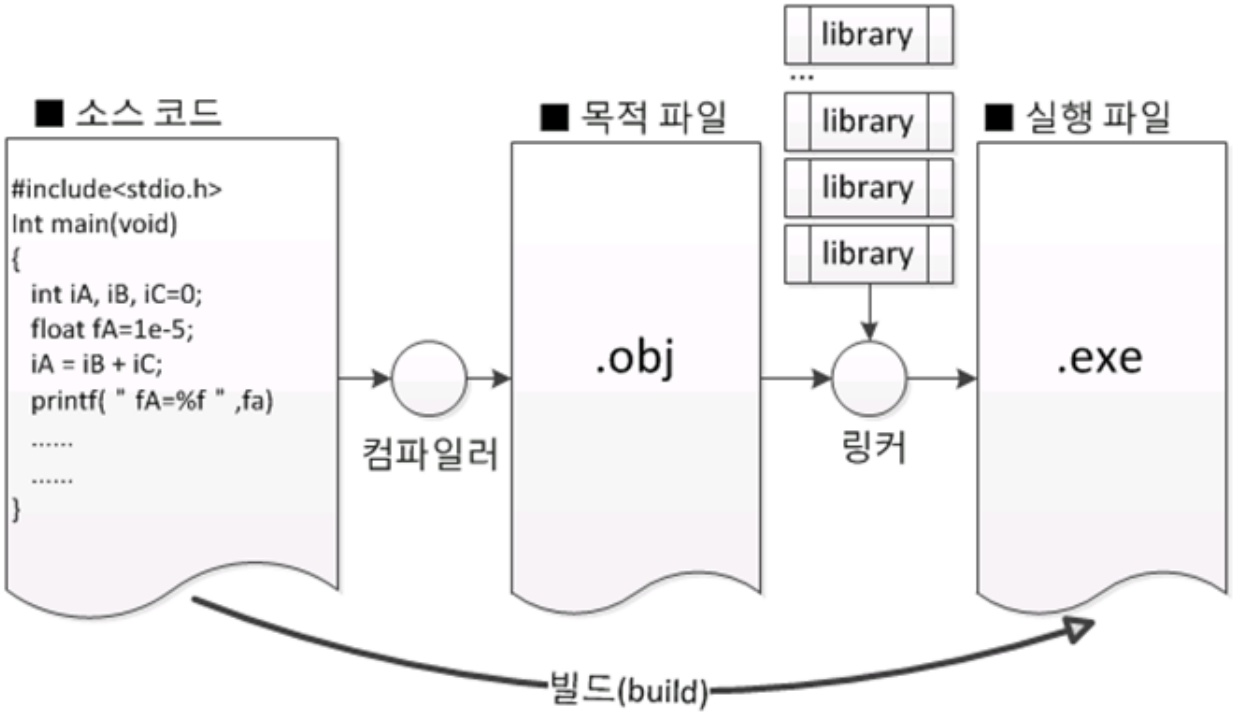
- **C# 프로그램 실행 과정**
  - 컴파일 과정

    csc : C# compiler
  - 실행 시스템

    CLR - Common Language Runtime



- C 프로그램 실행 과정



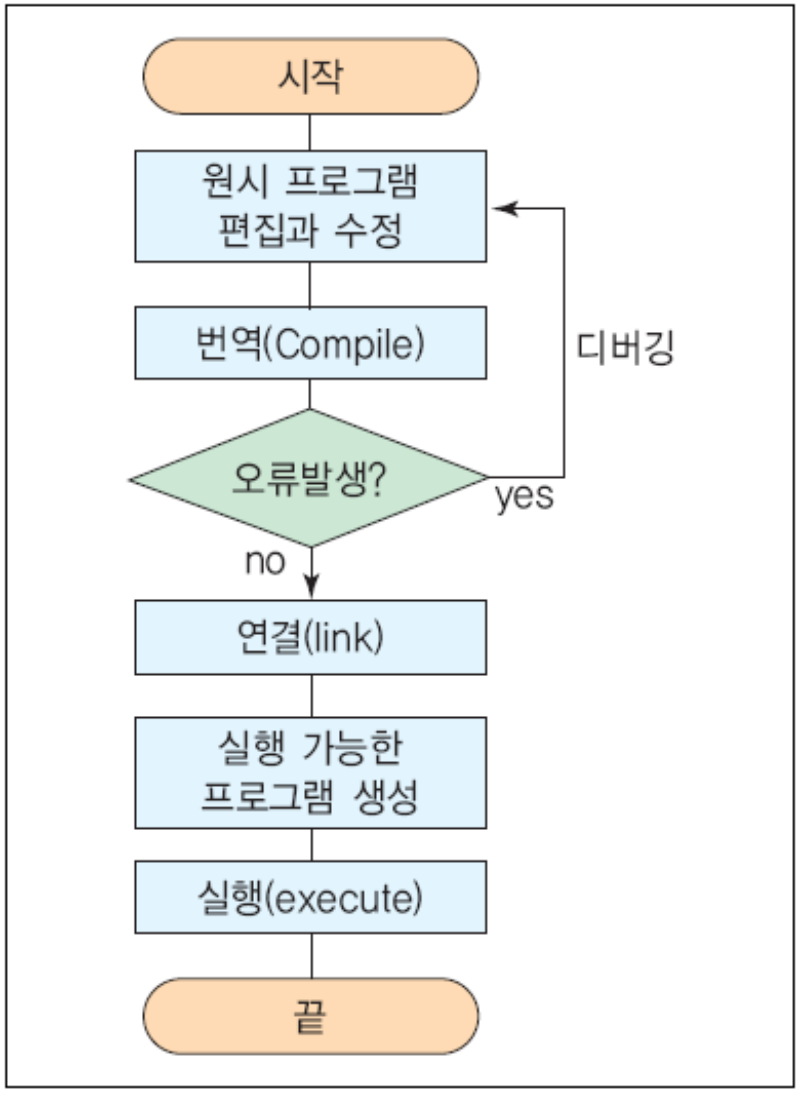
실행 가능한 프로그램이 만들어지기까지의 과정

- **원시 프로그램은**

  컴파일 과정을 거쳐야만 실행 가능한 프로그램으로 만들어진다.

  문법오류 : 프로그램이 약속된 문법을 지키지 않거나 약속된 명령을 사용하지 않아 컴파일러가 번역할 수 없는 오류

  실행오류 (Runtime Error)



[그림 1-6] 실행프로그램의 생성 과정

오류 확인

- **오류 확인 방법**

코드 1-3 오류가 발생하는 코드 /1장/ErrorCheck

```
static void Main(string[] args)
{
    Console.WriteRine("Hello World");
}
```

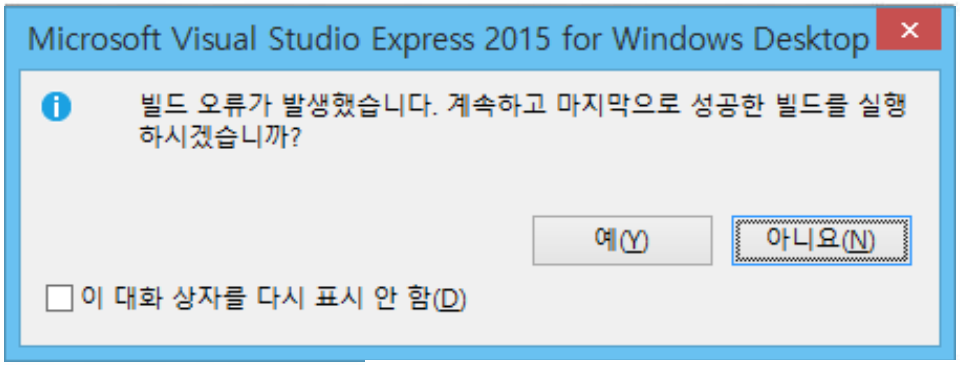


그림 1-22 오류 발생

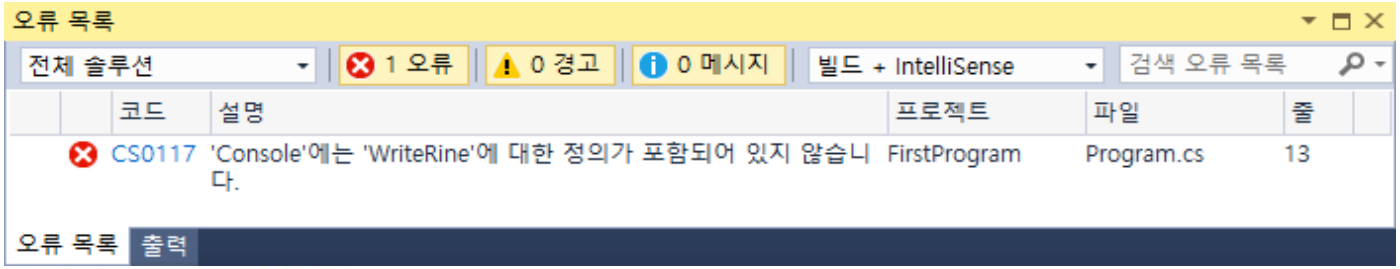


그림 1-23 오류 확인

1) 각각 어떤 결과를 얻는지 확인하세요.

```
Console.WriteLine(“Hello, world”);  // OKed
Console.WriteLine(“Hello, world”)   // 컴파일오류 ;
Console.WriteRine(“Hello, world”);  // 오류
Console.Writeline(“Hello, world”);  // 오류 -- 대소문자구별
Console.Write(“Hello, world”);
```

2) 주석을 달아보세요.

```
// ...
/* ... */
```

- **자동 완성 기능(인텔리센스Intellisense)과 보조 기능**
  - 코드 입력 시 Ctrl + Space 단축키 누르면 [그림 1-24]처럼 자동 완성 기능이 실행
  - [그림 1-25]와 같이 현재 위치에서 사용할 수 있는 코드가 뜨고 메서드를 사용할 때는 해당 메서드와 관련된 설명이 뜸

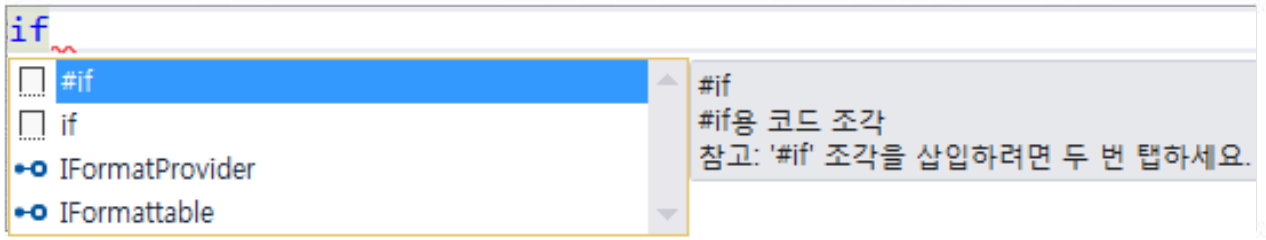


그림 1-24 자동 완성 기능

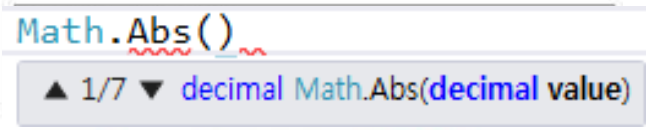


그림 1-25 메서드 설명과 사용 방법

요약정리

1. C# 개발 환경을 구축하였다
2. C# 코드를 작성하여 실행하여 보았다

과제 출력

C#으로 만든 모든 프로그램에는 main() 함수가 반드시 존재해야 한다

모든 문장은 마지막에 반드시 세미콜론(;)을 사용하여 문장 끝을 표시해야 한다

C#는 대소문자를 구분하여 사용한다

- **콘솔창에 위의 문장을 출력하도록 코드를 보완하세요**