

컴퓨터정보과 C# 프로그래밍

1주차 환경 구축 및 C# 소개

강의 순서

1. C# 환경설치 / C# 기본 구조
2. 클래스 기본(필드) + 변수, 자료형
3. 클래스 기본(메소드) + 연산자, 수식
4. 클래스 기본(메소드) + 제어문 빠르게 보기
5. 배열/리스트/딕셔너리
6. 클래스 기본: 접근제한자 (한정자)
7. 클래스 심화: 상속
8. 클래스 심화: 인터페이스/추상 클래스
9. 프로퍼티/예외처리/일반화
10. 파일처리
11. UI (Winform or WPF)
12. LINQ/Delegate/Lambda/...

[개발환경 구축] Visual Studio 2022 설치 & 프로젝트 생성 및 컴파일

(2024.03 기준)

기본 수업 환경

- Windows 10/11
- Visual Studio 2022

폴더설정 (WIN 10)

폴더 옵션

일반 보기 검색

폴더 보기

이 유형의 모든 폴더에 현재 보기(예: 자세히 또는 아이콘)를 적용할 수 있습니다.

모든 폴더에 적용(L) 모든 폴더를 원래대로(R)

고급 설정:

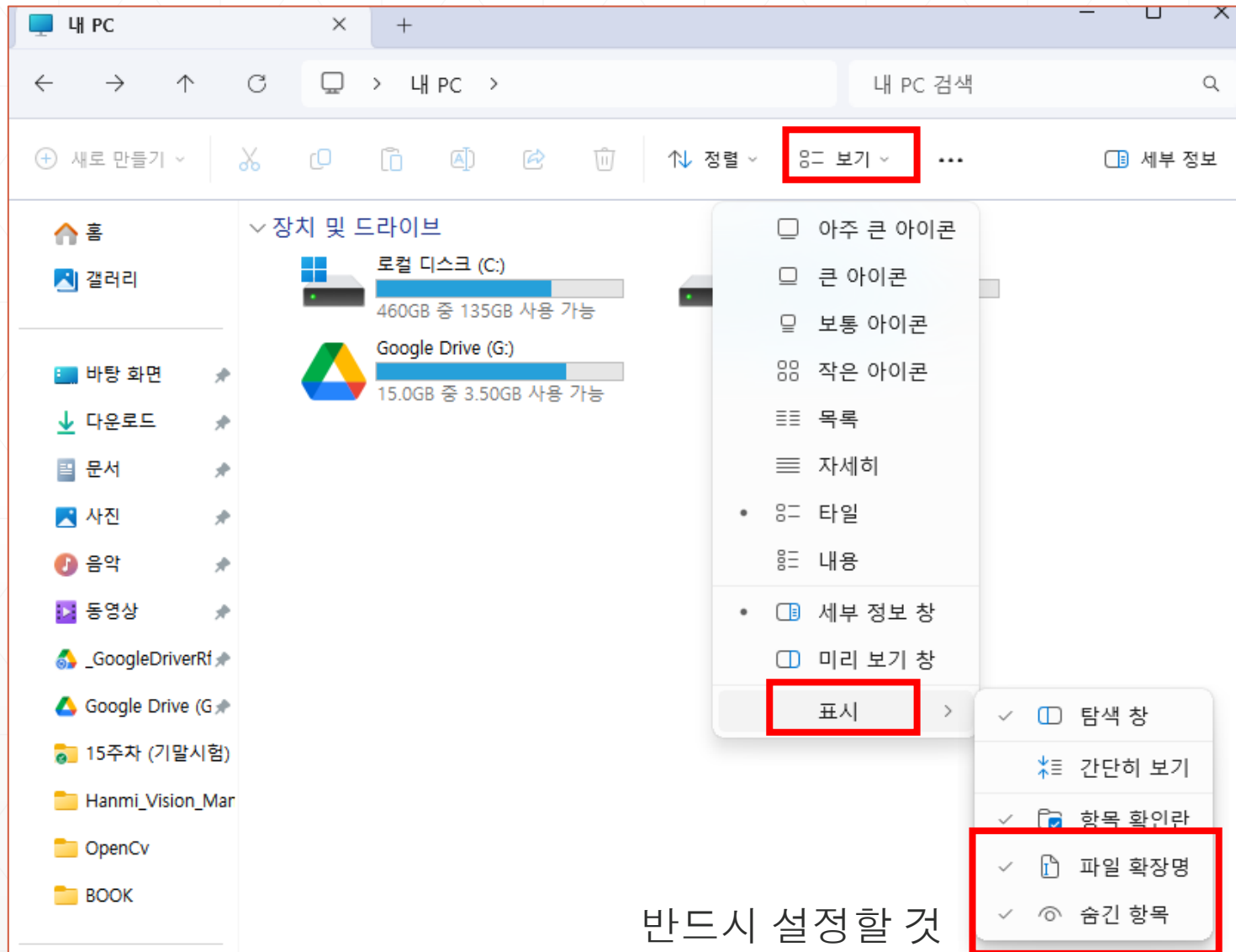
- ☒ 상태 표시줄 표시
- ☒ 숨김 파일 및 폴더
 - ☐ 숨김 파일, 폴더 또는 드라이브 표시 안 함
 - ☒ 숨김 파일, 폴더 및 드라이브 표시
- ☐ 아이콘은 항상 표시하고 미리 보기는 표시하지 않음
- ☐ 알려진 파일 형식의 파일 확장명 숨기기
- ☒ 암호화되거나 압축된 NTFS 파일을 컬러로 표시
- ☐ 제목 표시줄에 전체 경로 표시
- ☒ 폴더 및 바탕 화면 항목에 팝업 설명 표시
- ☒ 폴더 병합 충돌 숨기기
- ☒ 폴더 팁에 파일 크기 정보 표시
- ☒ 확인란을 사용하여 항목 선택

기본값 복원(D)

확인 취소 적용(A)

반드시 해제할 것

폴더설정 (WIN 11)



반드시 설정할 것

다운로드



Visual Studio 2022 |

웹, 클라우드, 데스크톱, 모바일 앱, 서비스, 게임을 빌드하는 Windows의 .NET 및 C++ 개발자를 위한 가장 포괄적인 IDE입니다.

Preview

아직 주 릴리스에 없는 최신 기능에 대한 초기 액세스

[자세한 정보 →](#)

커뮤니티

학생, 오픈 소스 참여자 및 개인에게 무료로 제공되는 강력한 IDE

[무료 다운로드](#)

Professional

소규모 팀에 가장 적합한 Professional IDE

[시작 평가판](#)

Enterprise

모든 규모의 팀을 위한 확장성 있는 엔드 투 엔드 솔루션

[시작](#)

[릴리스 정보 →](#)

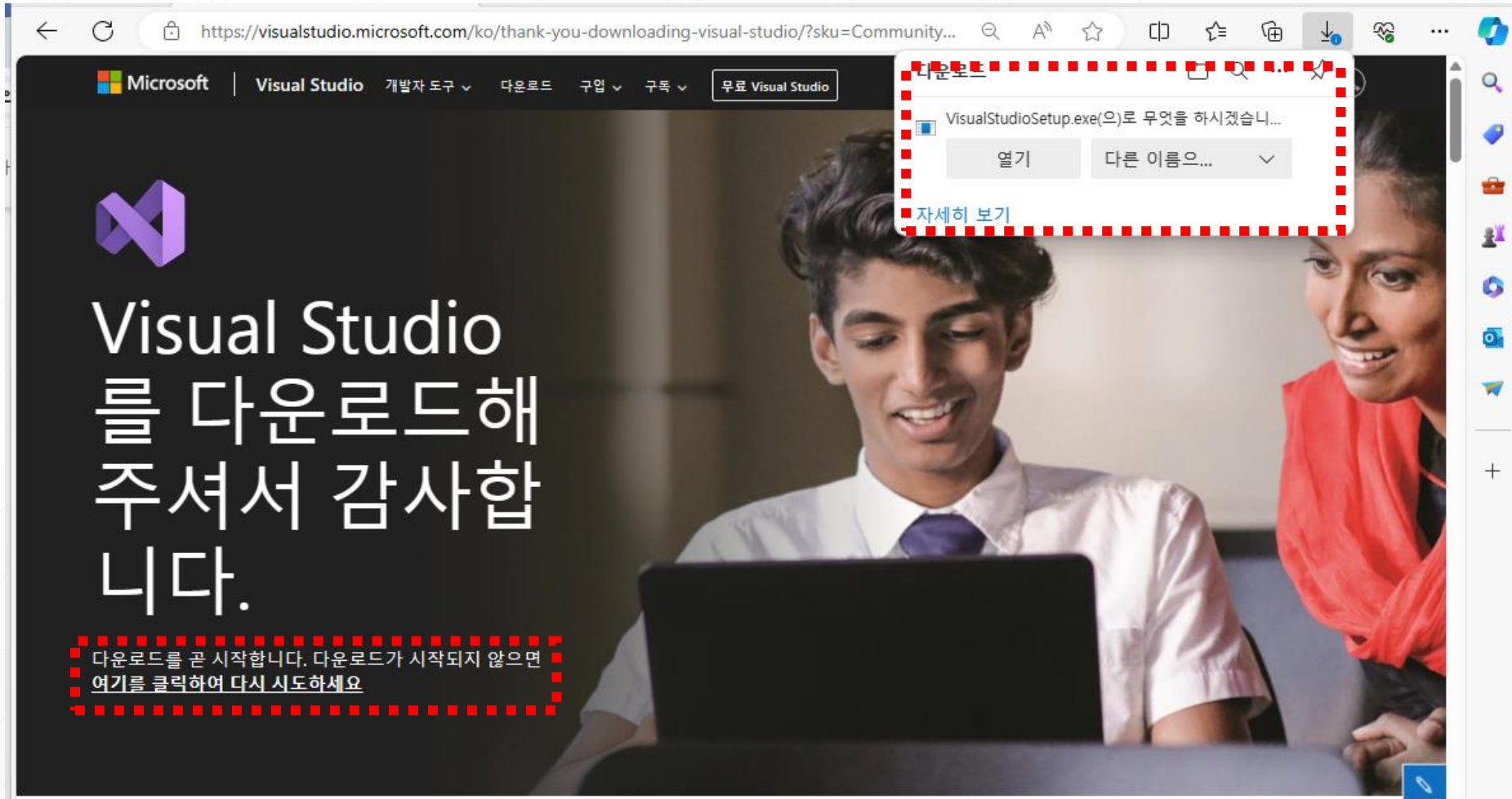
[→ 클릭](#)

[오프라인 설치 방법 →](#)

[라이선스 사용 조건 →](#)

다운로드

 VisualStudioSetup.exe



The screenshot shows the Visual Studio download page on the Microsoft website. The page features the Visual Studio logo and the text "Visual Studio 를 다운로드해 주셔서 감사합니다." (Thank you for downloading Visual Studio). A red dashed box highlights a message at the bottom: "다운로드를 곧 시작합니다. 다운로드가 시작되지 않으면 여기를 클릭하여 다시 시도하세요" (Download will start soon. If the download does not start, click here to try again). Another red dashed box highlights a download dialog box that appears when the download button is clicked, showing the file name "VisualStudioSetup.exe(으)로 무엇을 하시겠습니까?" (What do you want to do with VisualStudioSetup.exe?). The dialog box has buttons for "열기" (Open), "다른 이름으로..." (Save with this name...), and a link for "자세히 보기" (View details).

Microsoft | Visual Studio 개발자 도구 ▾ 다운로드 ▾ 구입 ▾ 구독 ▾ 무료 Visual Studio

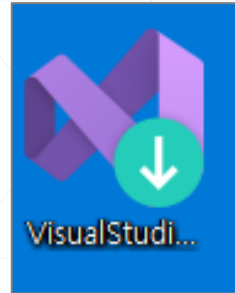
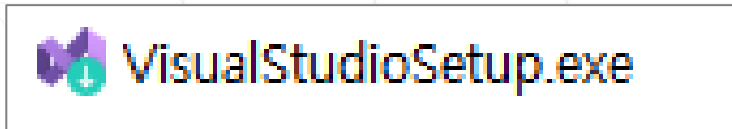
Visual Studio
를 다운로드해
주셔서 감사합
니다.

다운로드를 곧 시작합니다. 다운로드가 시작되지 않으면
여기를 클릭하여 다시 시도하세요

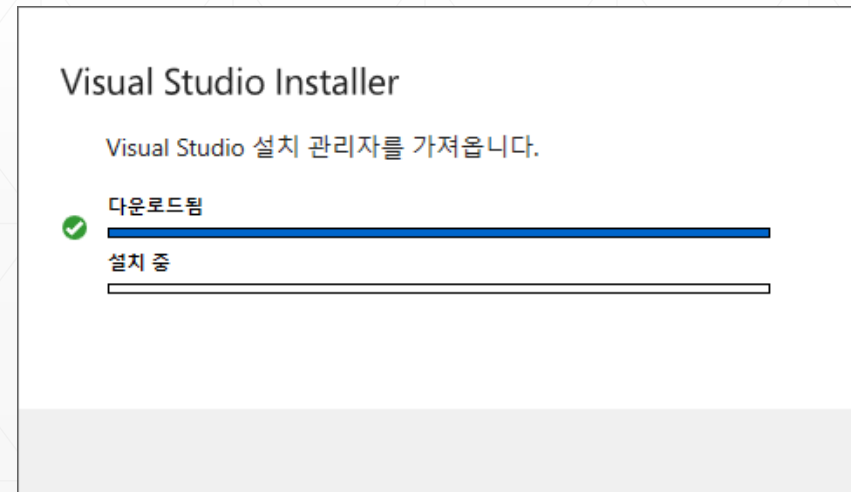
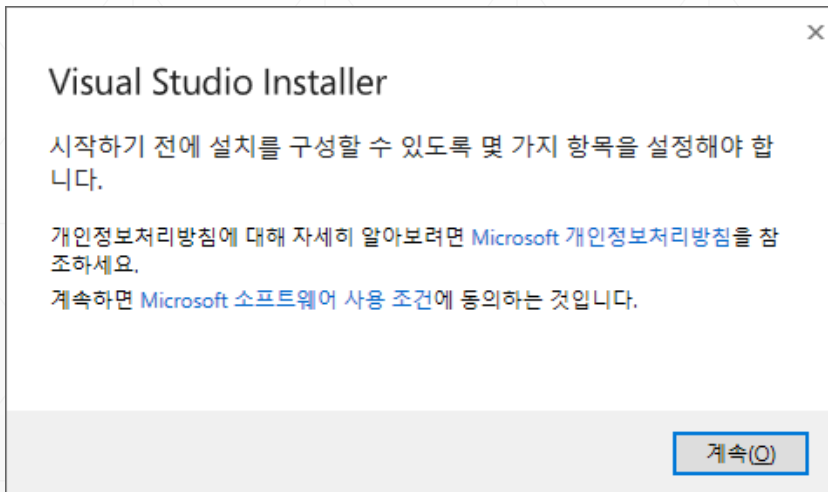
다운로드
VisualStudioSetup.exe(으)로 무엇을 하시겠습니까?
열기 다른 이름으로... ▾
자세히 보기

설치 (1) - 시작

- 다운로드 폴더나 본인이 지정한 위치 열면 설치 파일 있음



- 파일을 실행한다. (마우스 우클릭 → 관리자 권한으로 실행)
 - Visual Studio 설치 전 Visual Studio Installer가 먼저 설치할 수 있기 때문에 ,상황에 따라 다른 화면이 나올 수 있음 (보통, 계속이나 다음 누르면 됨)



설치 (2) – C#기준 설정


Visual Studio Installer


설치 — Visual Studio Community 2022 — 17.5.1

워크로드 개별 구성 요소 언어 팩 설치 위치


설치할 항목을 선택하는 데 도움이 필요하신가요? [추가 정보](#)


데스크톱 및 모바일 (5)

 **.NET Multi-Platform App UI 개발**
.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서 Android, iOS, Windows 및 Mac용 앱을 빌드합니다.

 **C++를 사용한 데스크톱 개발**
MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.

 **C++를 사용한 모바일 개발**
C++를 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다.

 **.NET 데스크톱 개발**
.NET 및 .NET Framework와 함께 C#, Visual Basic 및 F#을 사용하여 WPF, Windows Forms 및 콘솔 애플리케이션을...

 **유니버설 Windows 플랫폼 개발**
C#, VB 또는 C++(선택 사항)를 사용하여 유니버설 Windows 플랫폼용 애플리케이션을 만듭니다.

설치 세부 정보

Visual Studio 핵심 편집기

C++를 사용한 데스크톱 개발

.NET 데스크톱 개발

포함됨

- ✓ .NET 데스크톱 개발 도구
- ✓ .NET Framework 4.7.2 개발 도구
- ✓ C# 및 Visual Basic

선택 사항

- ✓ .NET용 개발 도구
- ✓ .NET Framework 4.8 개발 도구
- ✓ Entity Framework 6 도구
- ✓ .NET 프로파일링 도구
- ✓ IntelliCode
- ✓ Just-In-Time 디버거
- ✓ Live Share
- ✓ ML.NET Model Builder
- ✓ Blend for Visual Studio
- ☐ F# 데스크톱 언어 지원

지원되지 않는 구성 요소 제거(R)

위치

C:\Program Files\Microsoft Visual Studio\2022\Community [변경...](#)

계속하면 선택한 Visual Studio 버전에 대한 [라이선스](#)에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 부여됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

필요한 전체 공간 14.53GB


다운로드하는 동안 설치

설치

설치 (3) - 진행중

Visual Studio Installer

[설치됨](#) [사용 가능](#)

 **Visual Studio Community 2022**

다운로드 및 확인 중(14MB/3.98GB)

0%

(3MB/초)

패키지 설치 중(4/736개)

0%

Microsoft.VisualStudio.MinShell.Auto.Resources

☒ 설치 후 시작

[릴리스 정보](#)

일시 중지

개발자 뉴스

[GitHub Copilot in Visual Studio 2022](#)
GitHub Copilot uses OpenAI Codex to suggest co...
2023년 3월 1일 수요일

[Try Visual Studio 2022 v17.6 Preview 1](#)
The first Preview for Visual Studio 2022 17.6 is no...
2023년 2월 28일 화요일

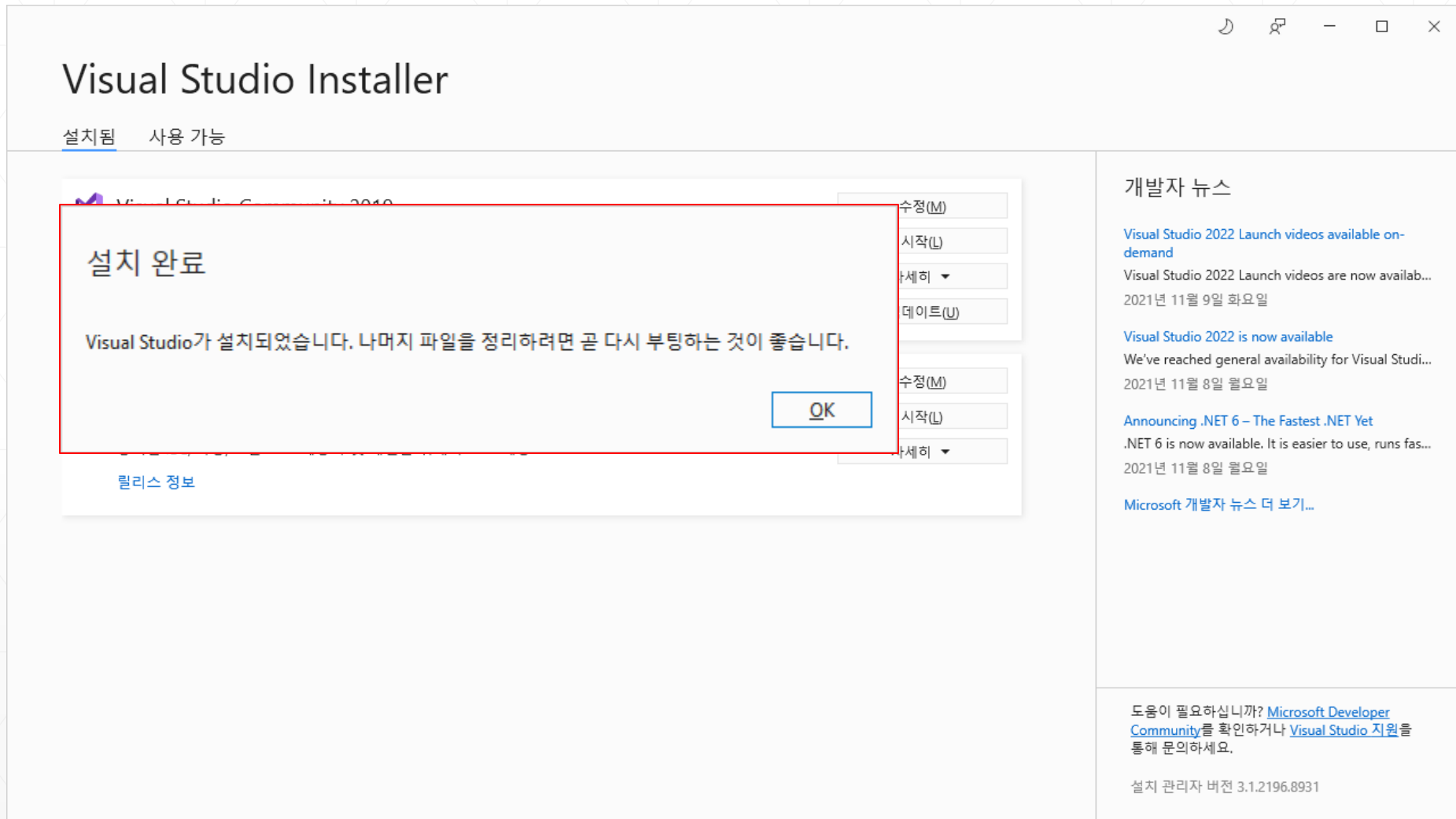
[The fastest way to add files and folders](#)
Your Visual Studio team is excited to announce th...
2023년 2월 27일 월요일

[Microsoft 개발자 뉴스 더 보기...](#)

도움이 필요하신가요? [Microsoft Developer Community](#)를 확인하거나 [Visual Studio 지원](#)을 통해 문의하세요.

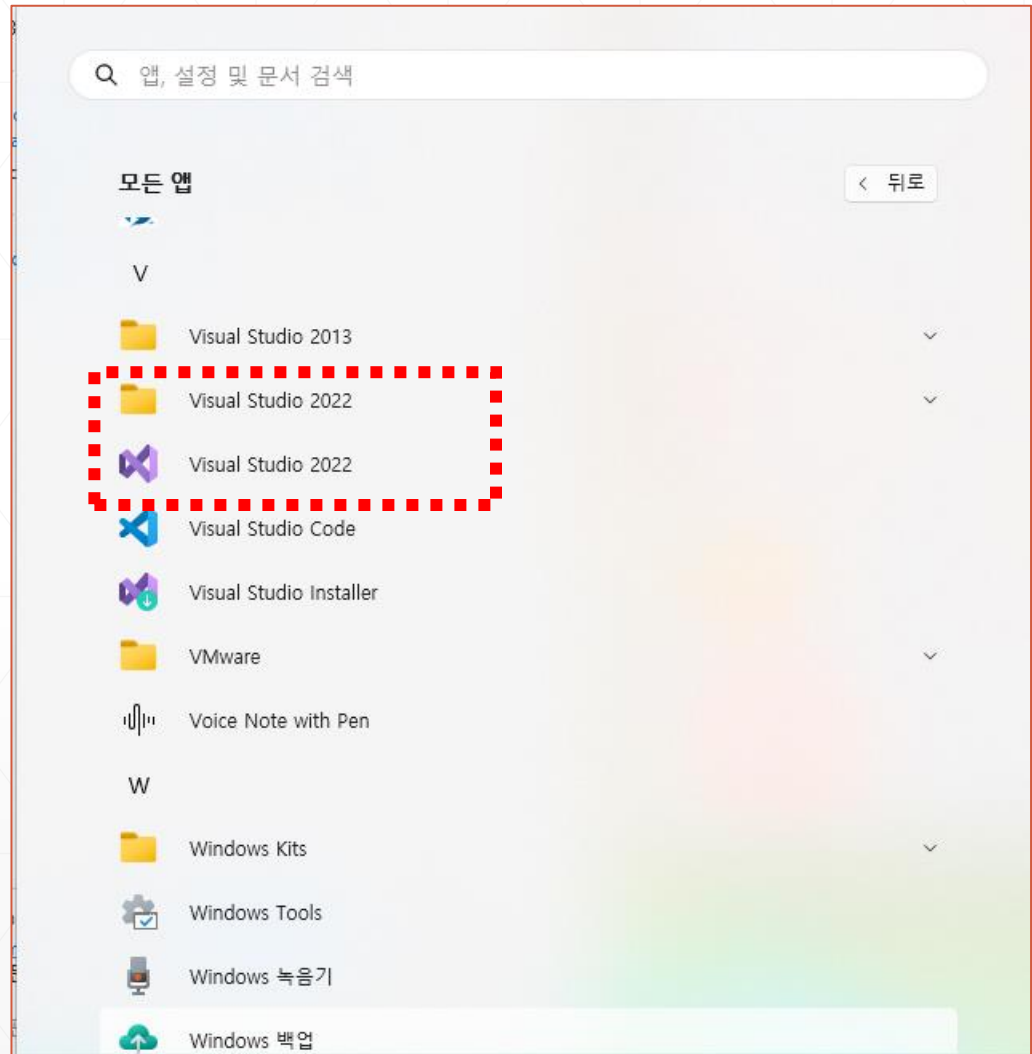
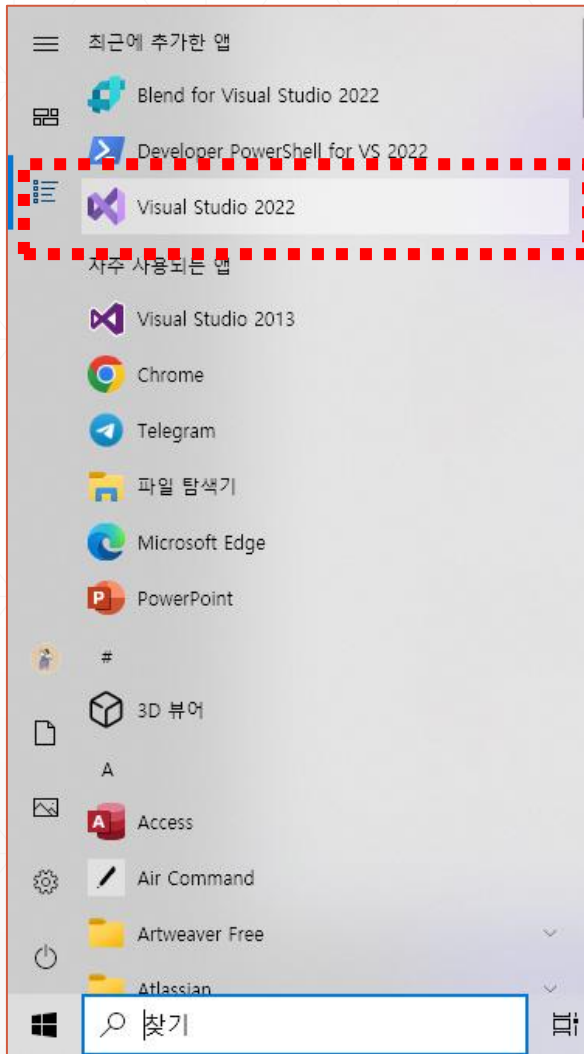
설치 관리자 버전 3.5.2145.59678

설치 (4) - 종료



설치 진행이 끝나면 되면, 다시 시작하기 화면이 나올 수 있음

설치 (5) – 종료



C# 소개

일반적인 사항을 포함

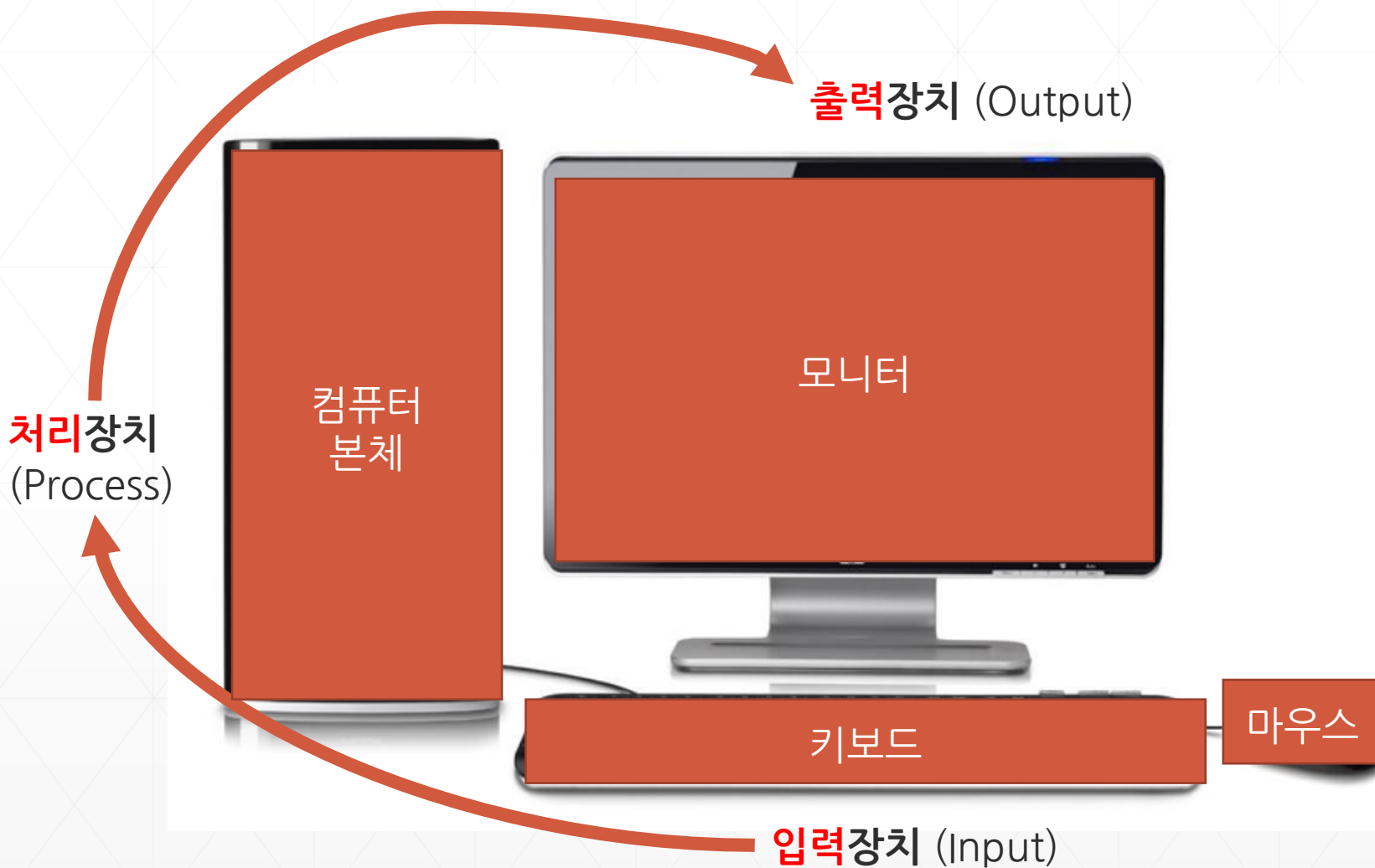
프로그래밍 개념

참고: 교재 p.24

- 컴퓨터 = 하드웨어 + **소프트웨어**(프로그램)
 - 하드웨어
 - 중앙처리장치 \leftrightarrow 기억장치 \leftrightarrow 저장장치
 - 소프트웨어(프로그램) : 컴퓨터를 **범용적**으로 만드는 것
 - 컴퓨터는 프로그램만 바꾸어 주면 다양한 작업을 할 수 있다.



프로그램 실행 구조 (1)



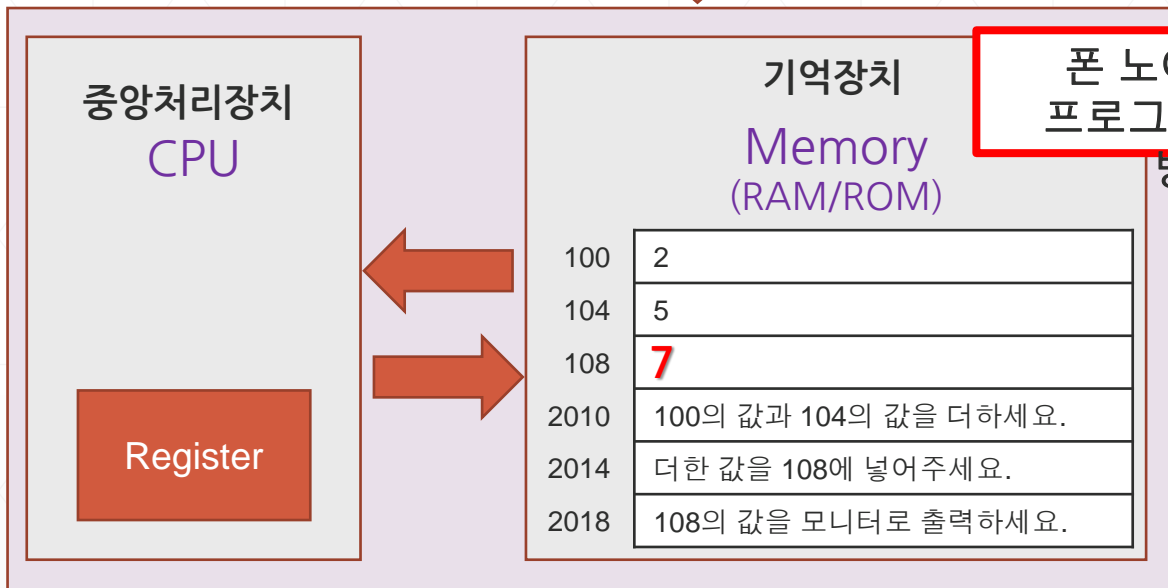
프로그램 실행 구조 (2)

입력장치

KEYBOARD

```
int a = 2;
int b = 5;
int c = a + b;
print("%d", c);
```

처리장치



폰 노이만구조
프로그래밍 내장

방식

file 단위로 저장

보조기억장치
Storage
(HDD/SSD)

출력장치

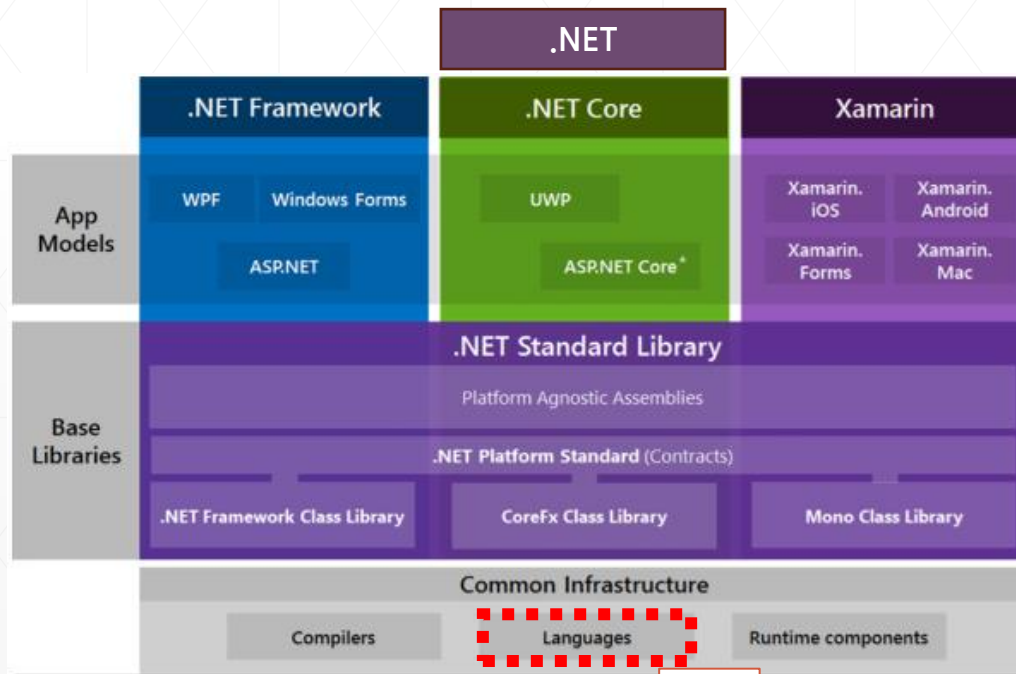
MONITOR

프로그램 개발 순서



.NET 구성

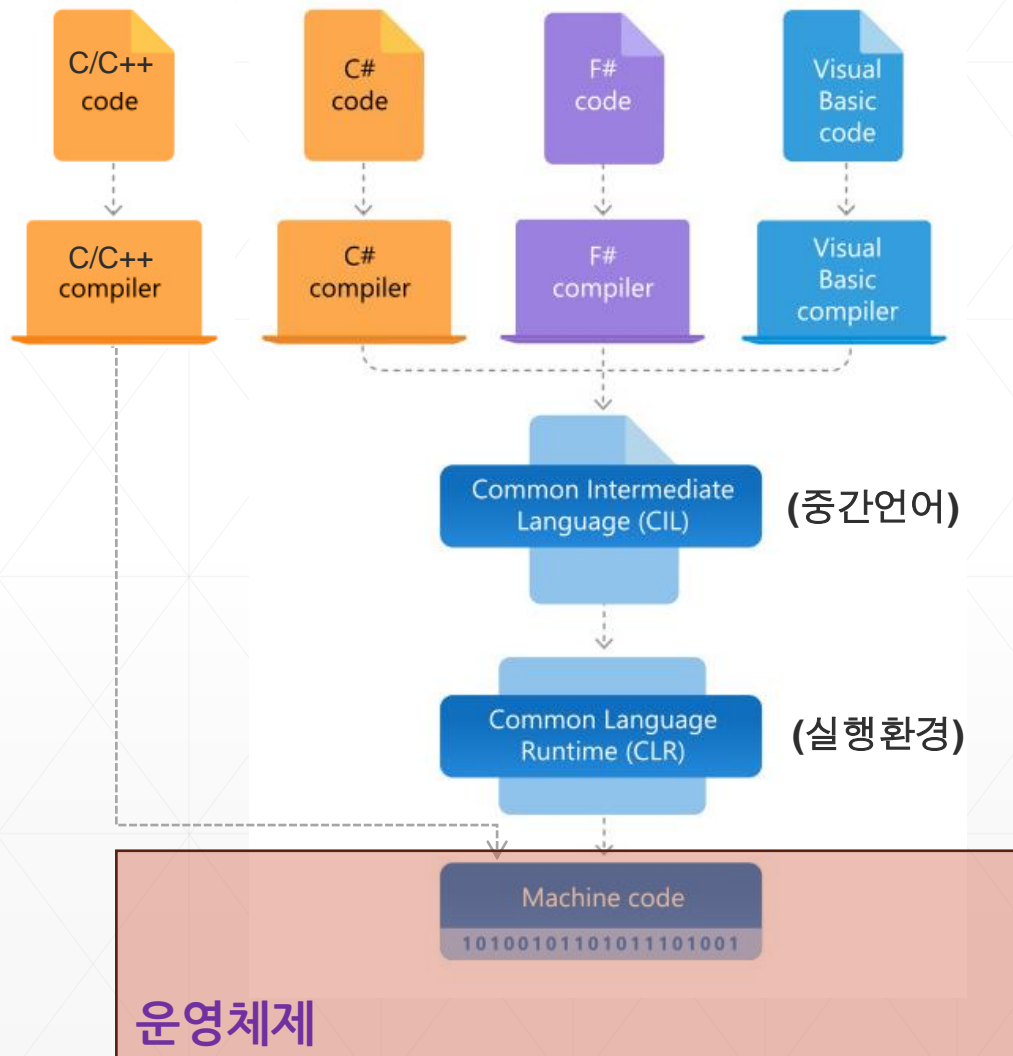
- **.NET Framework** - 원래(최초) .NET입니다. Windows 및 Windows Server의 광범위한 기능에 액세스할 수 있습니다. 유지 관리 면에서는 적극적으로 지원됩니다.
- **Xamarin(Mono)** - 원래 커뮤니티 및 오픈 소스 .NET입니다. .NET Framework의 플랫폼 간 구현입니다. Android, iOS 및 WebAssembly에 대해 적극적으로 지원됩니다.
- **.NET(Core)** - 최신 .NET. 클라우드 시대에 맞게 재구성된 플랫폼 간 및 오픈 소스 구현으로, .NET Framework와 상당한 호환성을 유지하면서 .NET을 구현합니다. Linux, macOS 및 Windows에 대해 적극적으로 지원됩니다.



C#

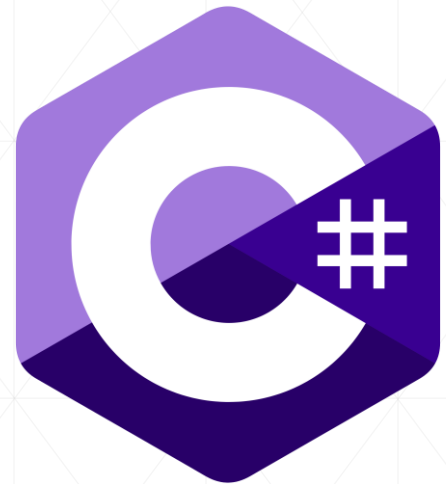
.NET Framework Architecture

- **공용 언어 런타임(CLR)**은 실행 중인 애플리케이션을 처리하는 실행 엔진으로서 스레드 관리, **가비지 수집**, 형식 안전성, 예외 처리 등의 서비스를 제공합니다.
- **클래스 라이브러리**는 일반적인 기능을 위한 API 및 형식 집합을 제공합니다. 문자열, 날짜, 숫자 등에 대한 형식을 제공합니다. 클래스 라이브러리에는 파일 읽기 및 쓰기, 데이터베이스에 연결, 그리기 등을 위한 API가 포함되어 있습니다.
- **Runtime** : 프로그램을 실행 할 수 있도록 수행해주는 환경
- **가비지 컬렉션**(쓰레기 수집, GC, Garbage Collection) : 동적 할당 메모리를 자동으로 해제 (C/C++ 불가능)



C#

- C++(C언어 + OOP) 계승
 - 마이크로소프트사에서 개발
 - 닷넷(.NET) 프로젝트와 함께 2000년 7월 PDC에서 발표
 - C++ 언어 기반 + Java 언어의 장점
- C# 언어 특징
 - 개체지향 언어 (Object-Oriented Programming)
 - 공식표현, 보통 객체지향 언어라고 표기
 - 배우고 활용하기 쉽다.
 - C++에서 사용하는 포인터 사용 제거/Garbage Collection 제공 (메모리 관리 편의성)
 - 엄격한 문법 제공 (프로그래머의 실수 방지)
 - 객체지향 언어 제공 및 다양한 응용 프로그램 작성
 - Desktop, web, game, db, ...
 - 다양한 플랫폼 지원



C# Version

C# Ver.	Platform		Date	Visual Studio
	.NET Framework(CLR)	.NET (Core)		
1.0	.NET Framework 1.0		2002	.NET 2002
1.1	.NET Framework 1.1		2003	.NET 2003
2.0	.NET Framework 2.0		2005	Visual Studio 2005
3.0	.NET Framework ~3.5		2006/2007	Visual Studio 2008
4.0	.NET Framework 4.0		2010	Visual Studio 2010
5.0	.NET Framework 4.5 .NET Framework 4.5.1 .NET Framework 4.5.2		2012 2013 2014	Visual Studio 2012 <u>Visual Studio 2013</u>
6.0	.NET Framework 4.6 .NET Framework 4.6.1		2015	Visual Studio 2015
7.0	.NET Framework 4.6.2		2017	Visual Studio 2017
7.1	.NET Framework 4.7	.NET Core 2.0	2017	Visual Studio 2017
7.2	.NET Framework 4.7.1		2017	Visual Studio 2017
7.3	.NET Framework 4.7.2	.NET Core 2.1/2.2	2018	Visual Studio 2017
8.0	.NET Framework 4.8 .NET Framework 4.8.1	.NET Core 3.0/3.1	2019 2022	Visual Studio 2019 Visual Studio 2022
9.0		.NET 5.0	2020	Visual Studio 2019
10.0		.NET 6	2021	Visual Studio 2022
11.0		.NET 7	2022	Visual Studio 2022
12.0		.NET 8 .NET 9 (Preview)	2023 2024	Visual Studio 2022 Visual Studio 2022

C# 기본구조

- 소스 파일 확장자 : cs
- 예제 프로그램 (vs. C언어)

C# (test.cs)

```
using System;

namespace FirstApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("안녕, C#");
        }
    }
}
```

C (test.c)

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    printf("안녕, C\n");
    return 0;
}
```

도형으로 보는 기본 구조

namespace (네임스페이스, 이름공간)

class (클래스)

field(필드, variable, 변수)

method (메소드)

local variable (지역변수)

제어문

class

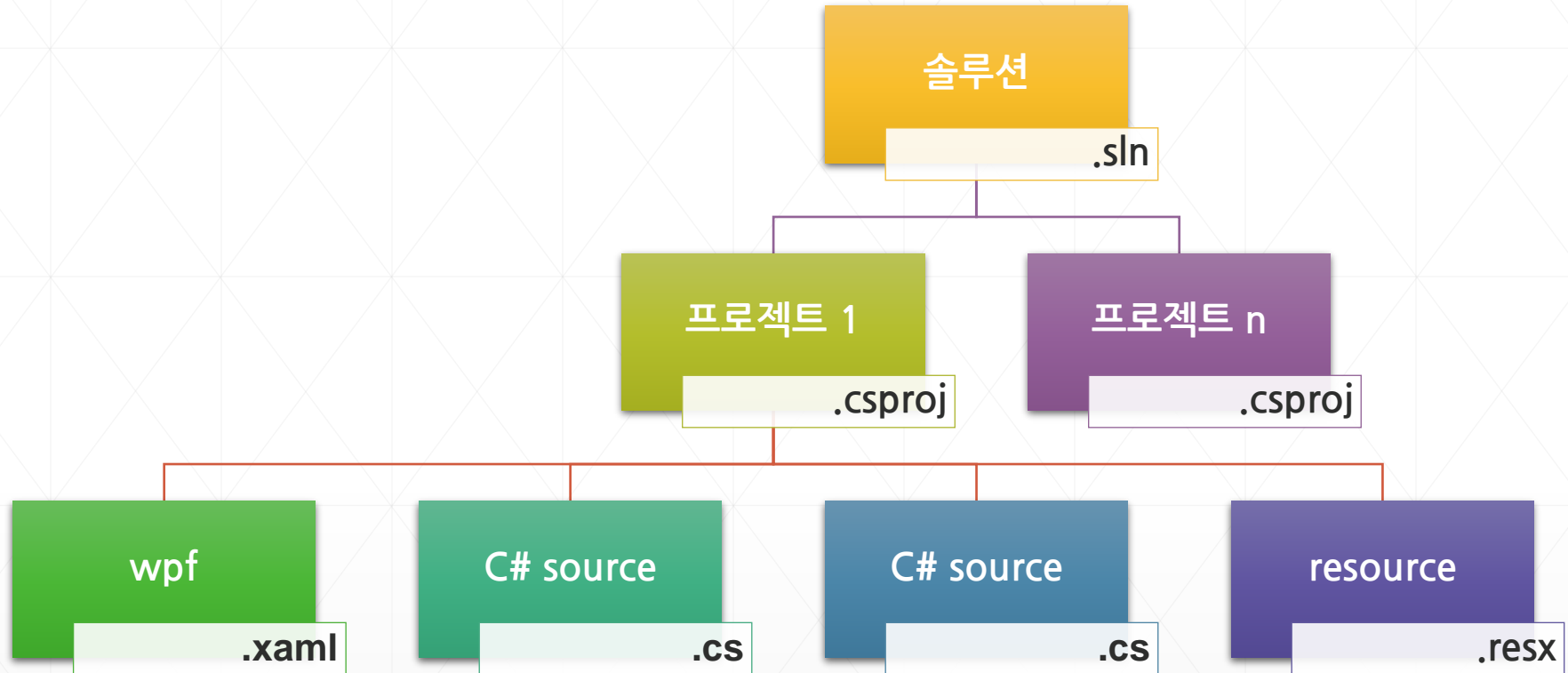
프로그램 > namespace > **class** > field / method

프로그램은 1개 이상의 class로 이루어져 있다.

Visual Studio 사용하기

version 2022

Visual Studio 프로젝트 관리

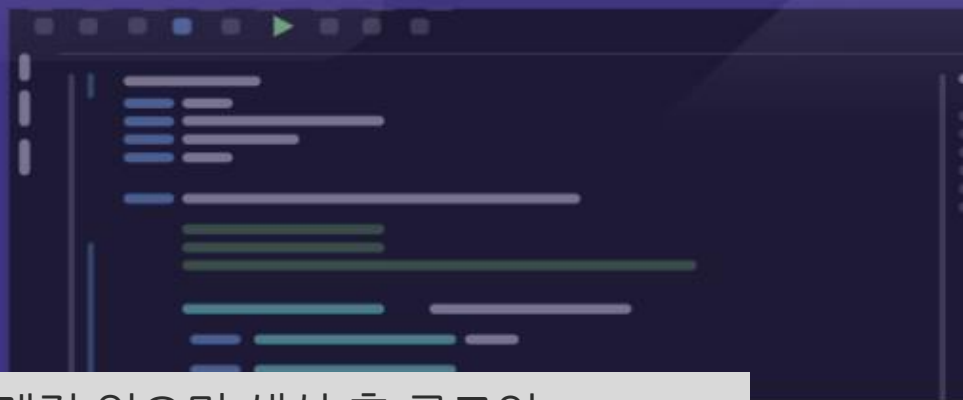




2022

Visual Studio에 로그인

디바이스 간에 설정을 동기화하고, 실시간으로 공동 작업하고, Azure 서비스와 원활하게 통합하세요.



로그인

계정 만들기

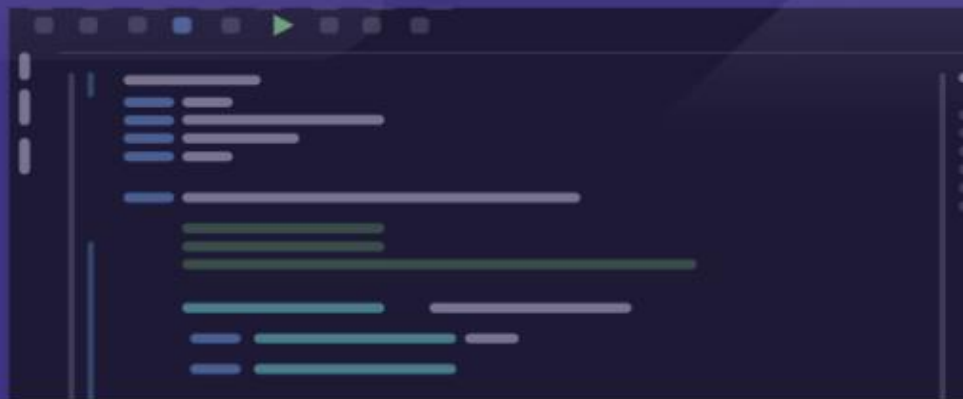
지금은 이 항목을 건너뛰니다.

계정 없으면 생성 후 로그인
(몇 달 뒤면, 로그인 없이 사용 불가)

Visual Studio 환경 개인 설정

워크플로의 레이아웃 및 바로 가기 키를 최적화합니다. 스타일에 맞는 색 테마를 선택하세요.

나중에 언제든지 이 설정을 변경할 수 있습니다.

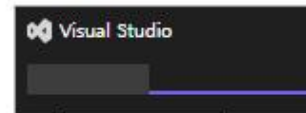


개발 설정

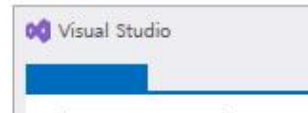
Visual C#

색 테마 선택

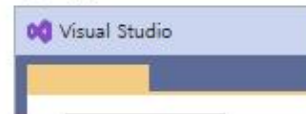
☐ 어둡게



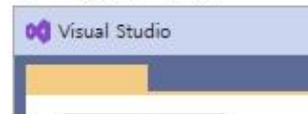
☒ 광원



☐ 파랑



☐ 파랑(추가 대비)



Visual Studio 시작

Visual Studio 2022

최근 파일 열기(R)

Visual Studio를 사용할 때 여는 프로젝트, 폴더 또는 파일은 빠른 액세스를 위해 여기에 표시됩니다.

항상 목록의 맨 위에 표시되도록 자주 여는 항목을 고정할 수 있습니다.

시작



리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는 .sln 파일 열기



로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

코드를 사용하지 않고 계속(W) →

.NET Framework

Console 프로젝트

실행-프로젝트 만들기



실행-프로젝트 만들기

— □ ×

새 프로젝트 구성

콘솔 앱(.NET Framework) C# Windows 콘솔

프로젝트 이름(I)

위치(L)
 ...

솔루션 이름(M) ⓘ

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

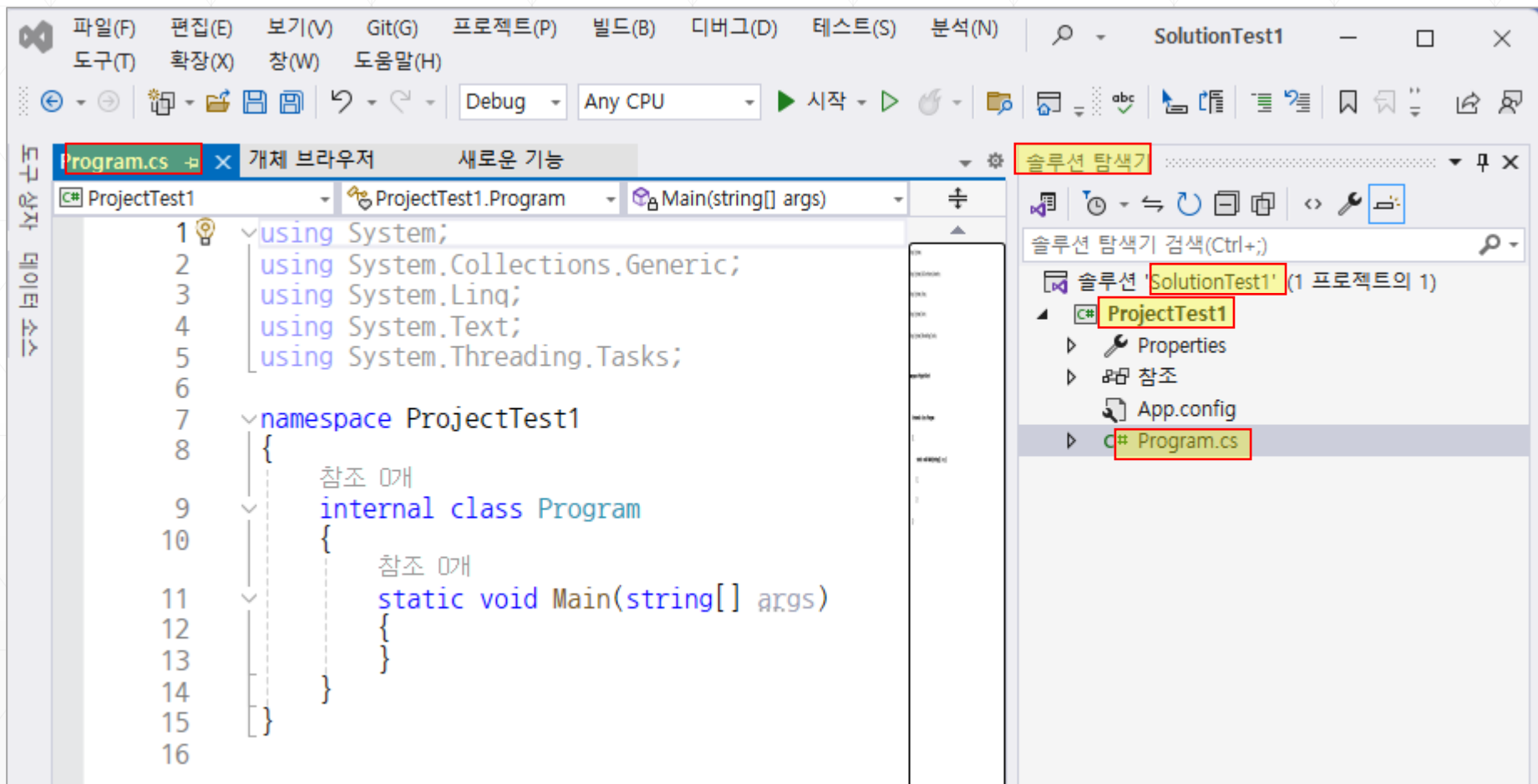
프레임워크(F)

"C:\W2024_csharp\W\SolutionTest1\W\ProjectTest1\W"에 프로젝트이(가) 만들어집니다.

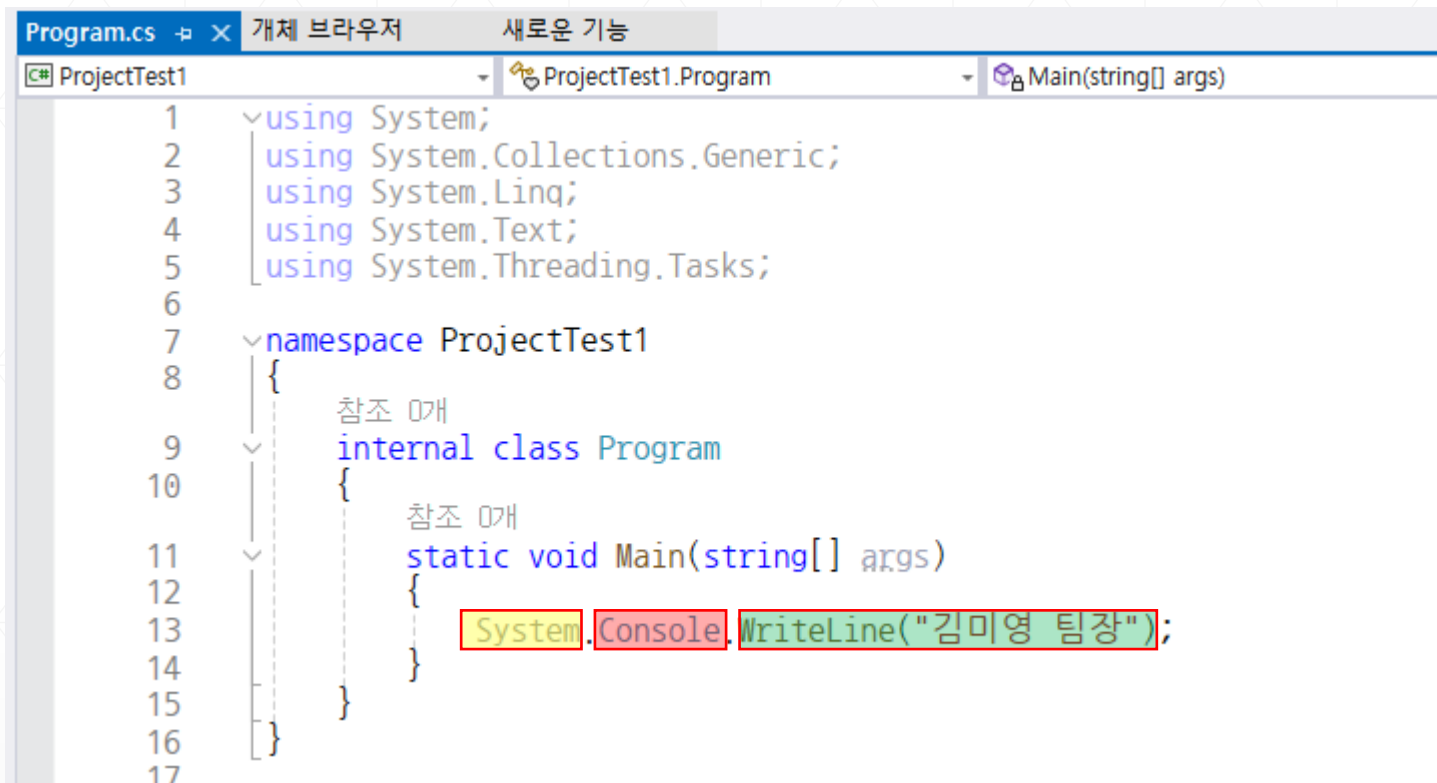
뒤로(B)

만들기(C)

실행-프로젝트 만들기



실행-코드 작성



```
Program.cs  x  개체 브라우저  새로운 기능
C# ProjectTest1  ProjectTest1.Program  Main(string[] args)
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ProjectTest1
8  {
9      참조 0개
10     internal class Program
11     {
12         참조 0개
13         static void Main(string[] args)
14         {
15             System.Console.WriteLine("김미영 팀장");
16         }
17     }
18 }
```

- System :
- Console :
- WriteLine() :
 - “김미영 팀장”:

실행-빌드 (실행파일 만드는 과정)

The screenshot shows the Visual Studio IDE with the 'Build' menu open. The 'Build' menu item in the top bar is highlighted in yellow. The 'Build' menu is also highlighted in yellow, and its options are listed below. The 'Build' menu options are:

- 솔루션 빌드(B) Ctrl+Shift+B
- 솔루션 다시 빌드(R)
- 솔루션 정리(C)
- 솔루션에서 코드 분석 실행(Y) Alt+F11
- ProjectTest1 작성(U...) Ctrl+B
- ProjectTest1 다시 빌드(E)
- ProjectTest1 정리(N)
- ProjectTest1 게시(H)
- ProjectTest1에 대해 코드 분석 실행(A)
- 일괄 빌드(T)...
- 구성 관리자(O)...

The code editor shows the following C# code:

```

9      internal class Program
10     {
11         참조 0개
12         static void Main(string[] args)
13         {
14             Console.WriteLine("김미영 팀장");
15         }
16     }
    
```

The status bar at the bottom indicates '문제가 검색되지 않음' (No problems found).

The Output window at the bottom shows the build process:

출력 보기 선택(S): 빌드

오후 2:29에 빌드를 시작한

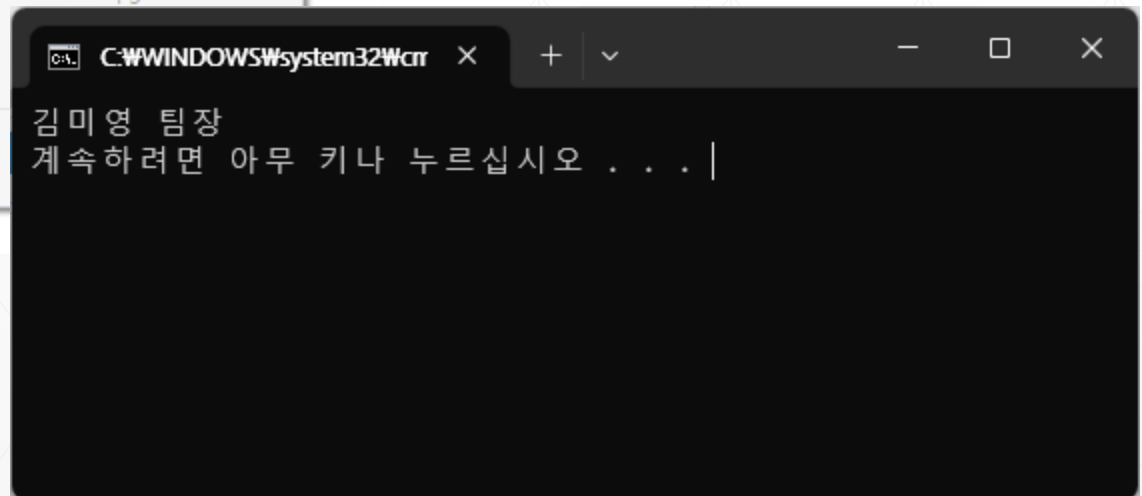
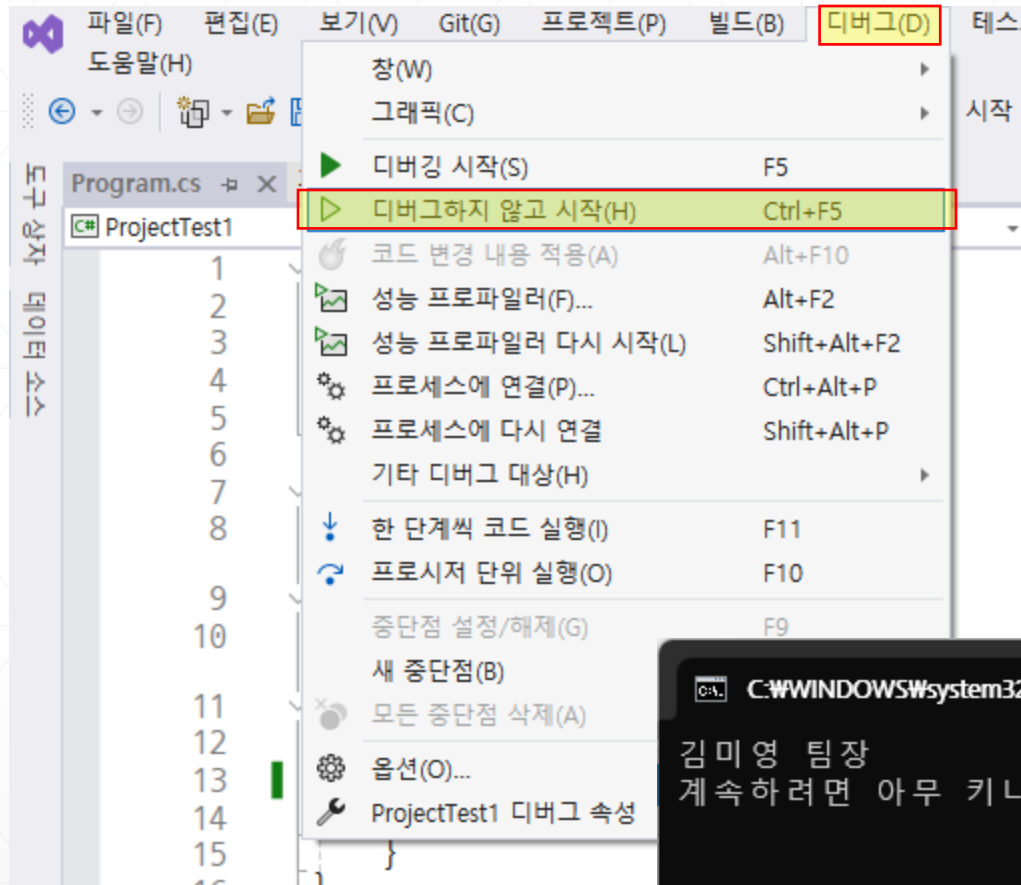
1>----- 빌드 시작: 프로젝트: ProjectTest1, 구성: Debug Any CPU -----

1> ProjectTest1 -> C:\2024_csharp\SolutionTest1\ProjectTest1\bin\Debug\ProjectTest1.exe

===== 빌드 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====

===== 빌드이(가) 오후 2:29에 완료되었으며, 00.150 초이(가) 걸림 =====

실행-디버그하지 않고 시작 (Ctrl+F5)



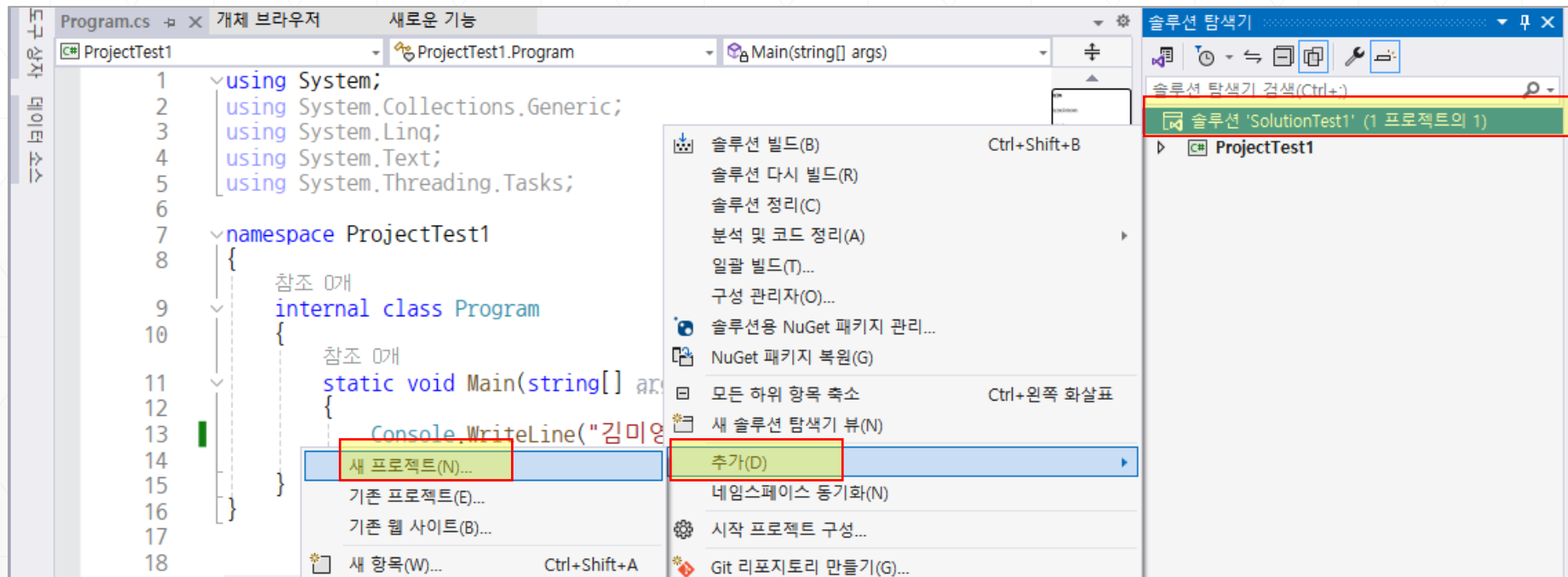
솔루션에 프로젝트 추가

■ Main() 메소드

- C언어와 동일하게 시작점(진입점, entry point) 역할을 수행함.
 - 프로젝트 당 한 개만 존재해야 함.

■ 시작 프로젝트

- 솔루션의 여러 프로젝트 중 시작할 프로젝트를 한 개 선택해야 함.



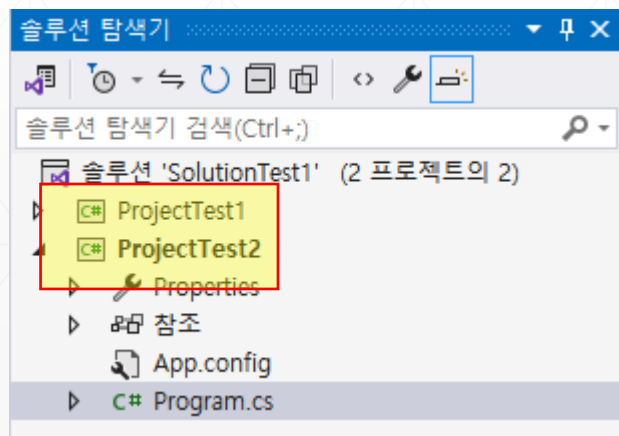
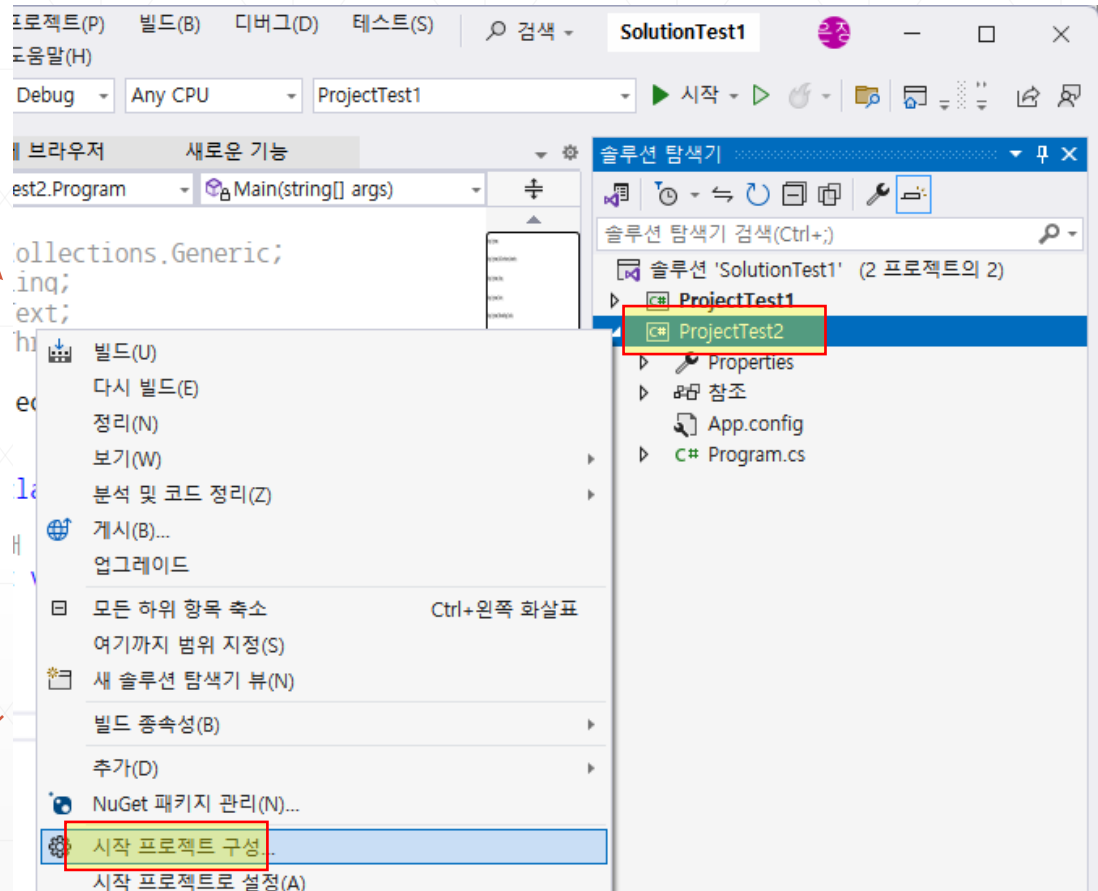
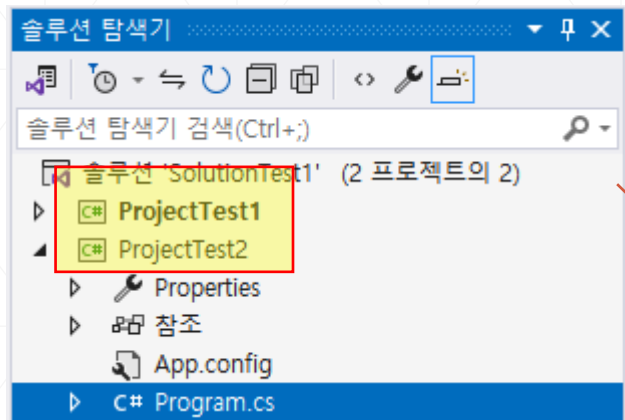
솔루션에 프로젝트 추가



솔루션 탐색기 (Solution Explorer) 화면:

- 제목: 솔루션 탐색기 (Solution Explorer)
- 검색창: 솔루션 탐색기 검색(Ctrl+;) (Solution Explorer Search)
- 솔루션: 솔루션 'SolutionTest1' (2 프로젝트의 2) (Solution 'SolutionTest1' (2 of 2 projects))
- 프로젝트 목록:
 - ▶ C# ProjectTest1
 - ▲ C# ProjectTest2
 - ▶ Properties
 - ▶ 참조 (References)
 - ▶ App.config
 - ▶ C# Program.cs

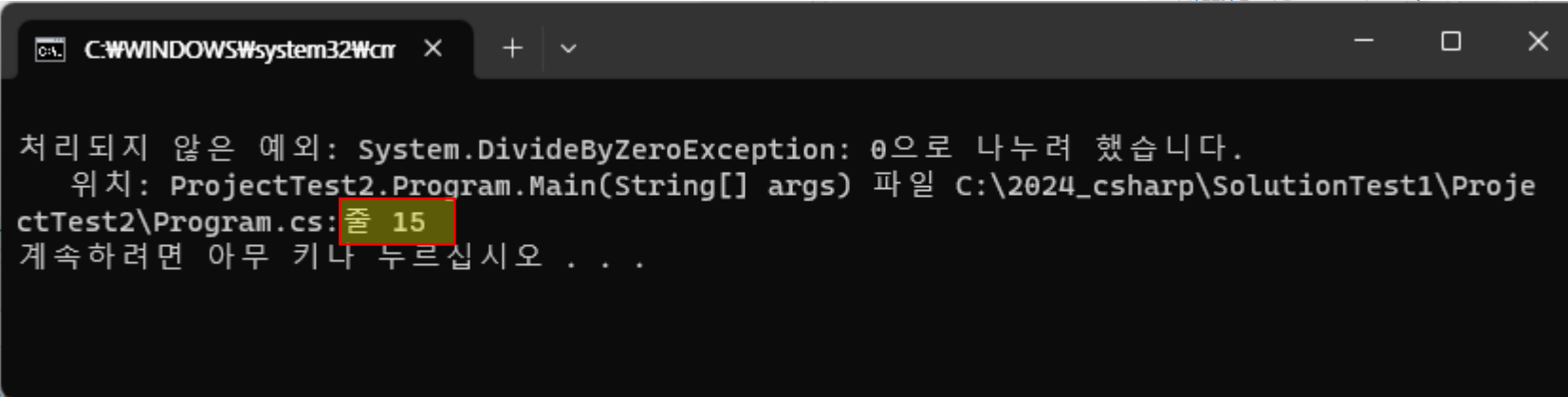
솔루션에 프로젝트 추가



디버깅 (Debugging)

- 프로그래밍 과정의 실수(bug)를 분석
- 디버깅 도구를 이용
- 프로그램 실행과정에서 변수의 변화된 값을 추적

```
static void Main(string[] args)
{
    int a = 10;
    int b = 2 / 3;
    int c = a / b;
    Console.WriteLine(a);
    Console.WriteLine(b);
    Console.WriteLine(c);
}
```

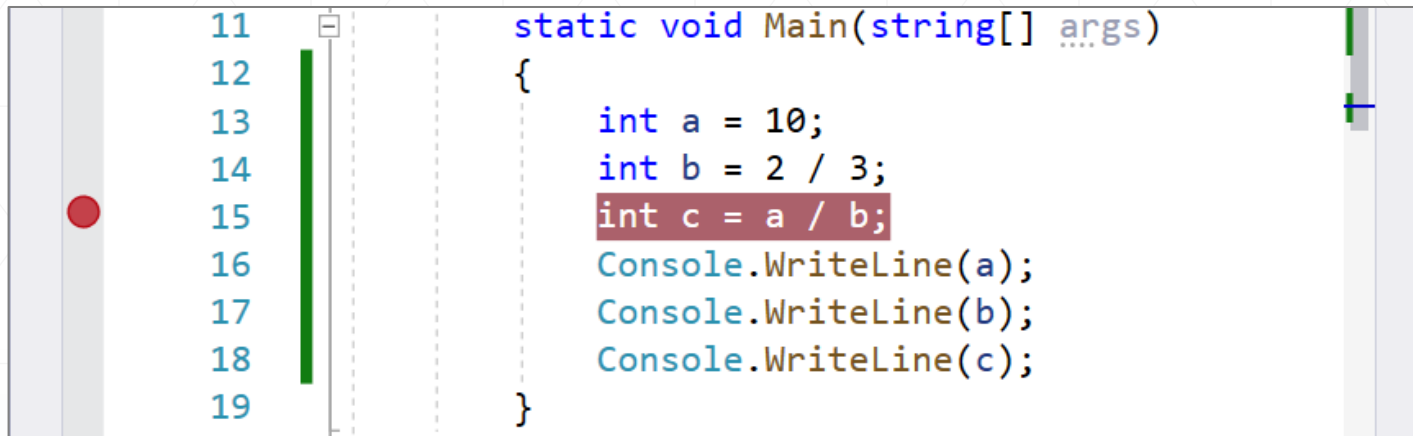


C:\WINDOWS\system32\cmd.exe

처리되지 않은 예외: System.DivideByZeroException: 0으로 나누려 했습니다.
위치: ProjectTest2.Program.Main(String[] args) 파일 C:\2024_csharp\SolutionTest1\ProjectTest2\Program.cs: 줄 15
계속하려면 아무 키나 누르십시오 . . .

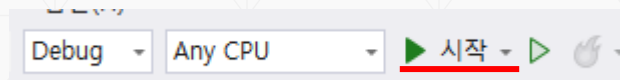
디버깅 (1)

- 중단점 설정
 - 15번 라인에서 'F9' 실행



```
11 static void Main(string[] args)
12 {
13     int a = 10;
14     int b = 2 / 3;
15     int c = a / b;
16     Console.WriteLine(a);
17     Console.WriteLine(b);
18     Console.WriteLine(c);
19 }
```

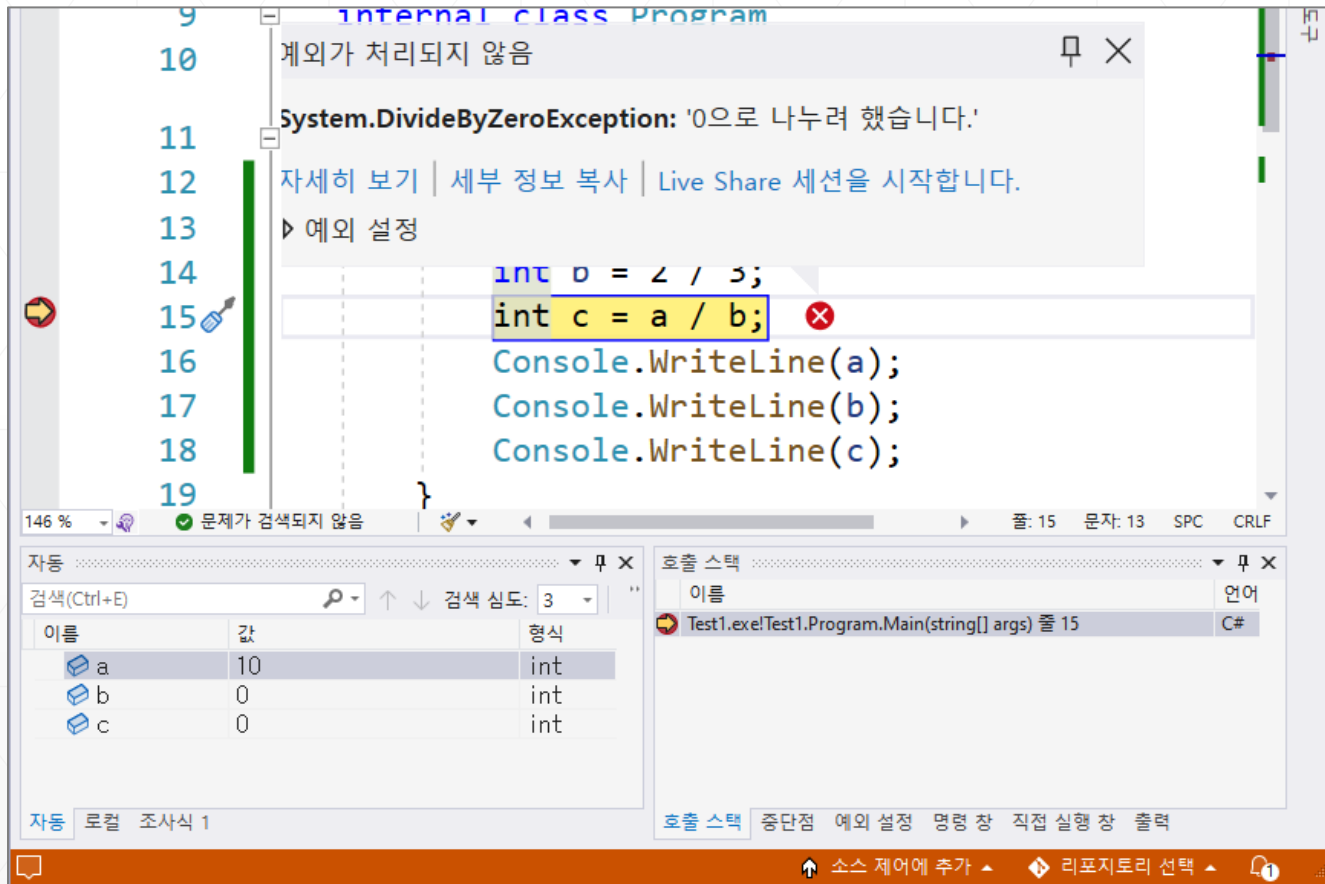
- 디버깅 모드 실행
 - 'F5' 실행



-

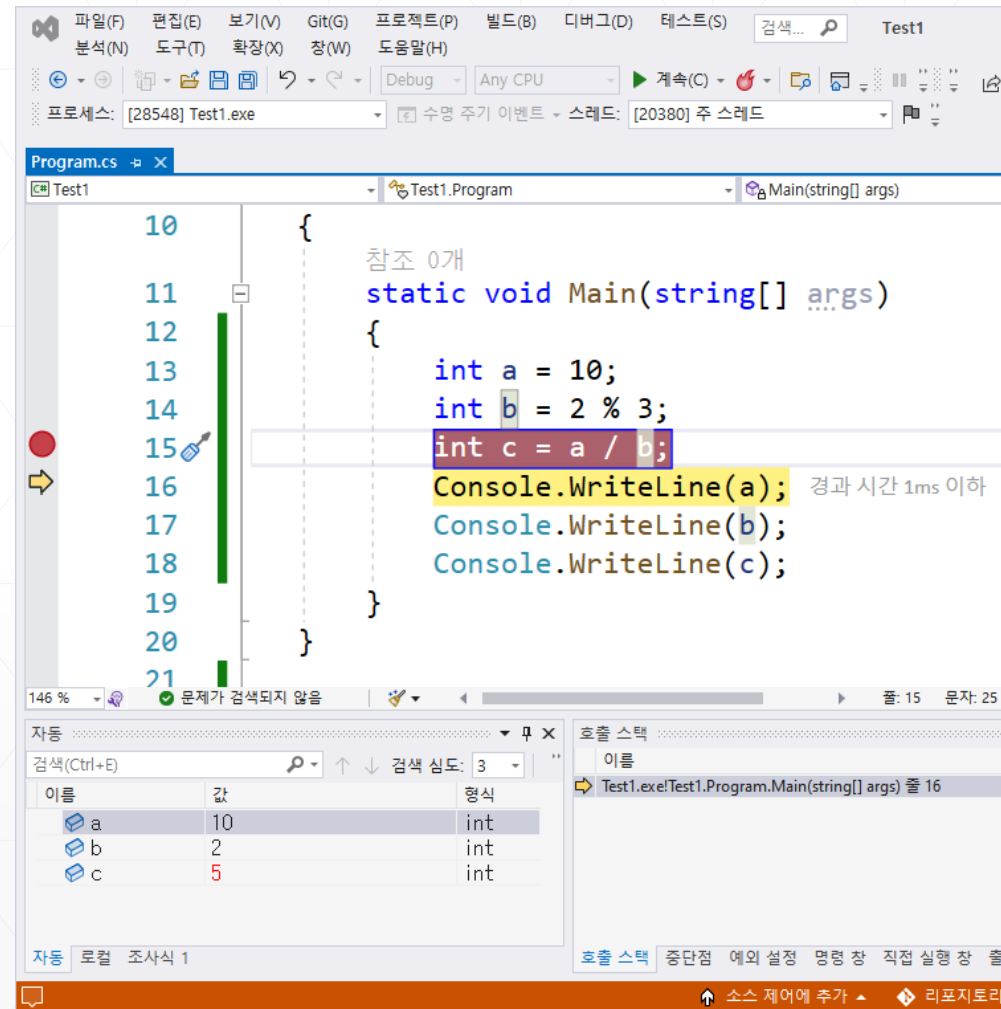
디버깅 (3)

- 다음 단계 실행 : F10/F11



디버깅 (4)

- 디버깅을 중단한 후, 14번 라인을 수정하고 다시 디버깅
- 다음 단계 실행 : F10/F11
 - 'F11'을 실행한 결과 화면이다.
 - 15번 라인을 수행하고 16번 라인에서 중지한 상태이고, c의 계산 결과 5를 확인할 수 있다.
- 디버깅 모드 프로시저 단위 실행
 - F10
- 디버깅 모드 한 단계씩 코드 실행
 - F11



디버깅 (5)

- 디버깅 나가기(중지) : Shift + F5
 - 디버깅을 중지한다. (실행이 완료되지 않는다)
- 단계별 진행없이, 계속 진행 : F5
 - 다른 중단점이 없으면, 실행을 끝까지 수행
 - 다른 중단점이 있으면, 그 곳에서 실행을 중지하고 대기

인텔리코드

참조 0개

`internal class Program`

{

참조 0개

`static void Main(string[] args)`

{

`int a = 10;`

탭 수락하려면

`Console.WriteLine(a + b);`

}

}

`static void Main(string[] args)`

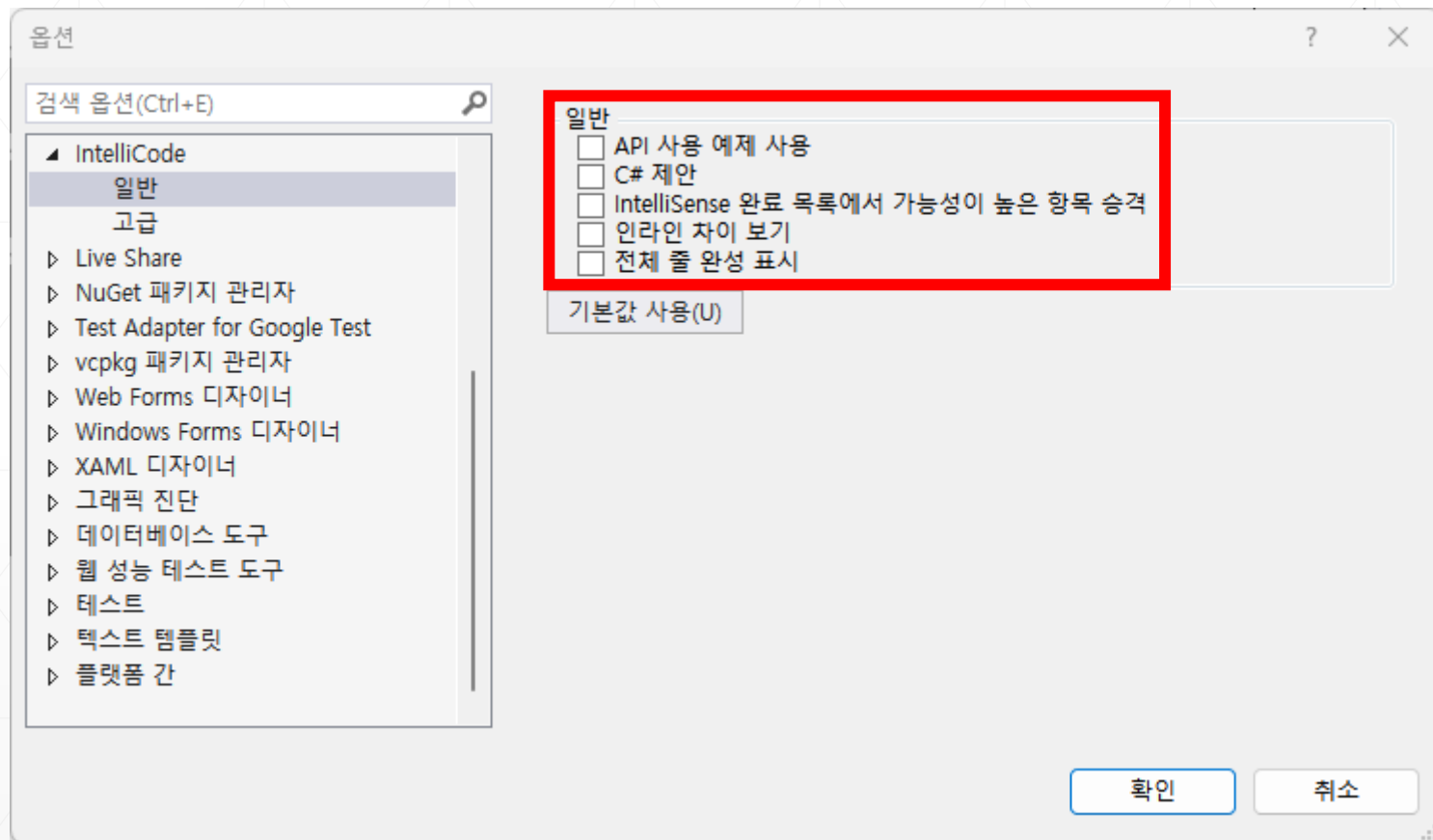
{

`int a = 10;``int b = 20;``Console.WriteLine(a + b);`

}

인텔리센스

도구 → 옵션 → IntelliCode → 일반 → 모두 해제



.NET Framework

Winform 프로젝트

Visual Studio 2022

최근 파일 열기(R)

Visual Studio를 사용할 때 여는 프로젝트, 폴더 또는 파일은 빠른 액세스를 위해 여기에 표시됩니다.

항상 목록의 맨 위에 표시되도록 자주 여는 항목을 고정할 수 있습니다.

시작



리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는 .sln 파일 열기



로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집

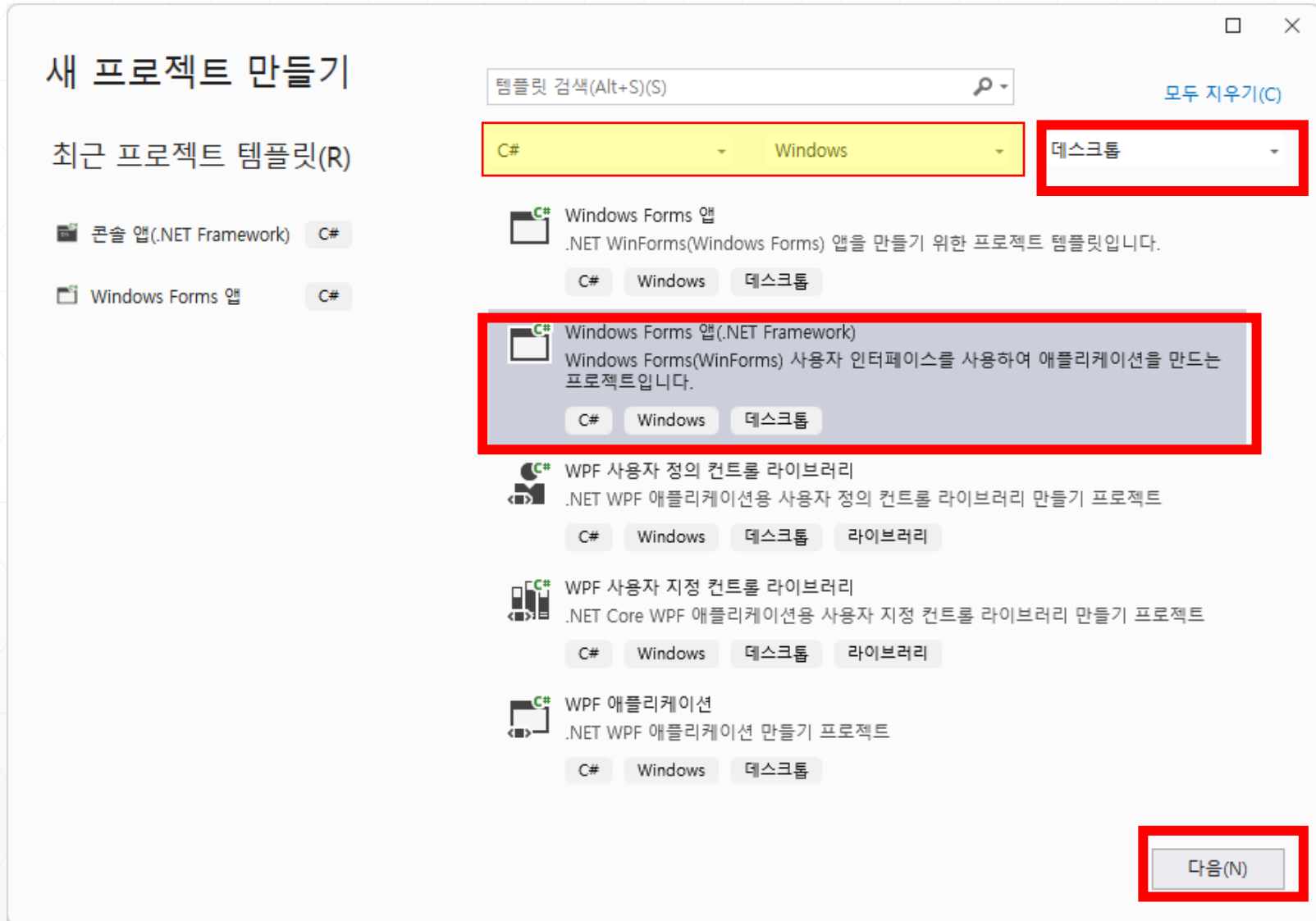


새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

코드를 사용하지 않고 계속(W) →

WinForm 프로젝트 만들기 (1)



WinForm 프로젝트 만들기 (2)

새 프로젝트 구성

Windows Forms 앱(.NET Framework) C# Windows 데스크톱

프로젝트 이름(I)
ProjectTest1

위치(L)
C:\2024_csharp ...

솔루션(S)
새 솔루션 만들기

솔루션 이름(M) ⓘ
SolutionTest1_2

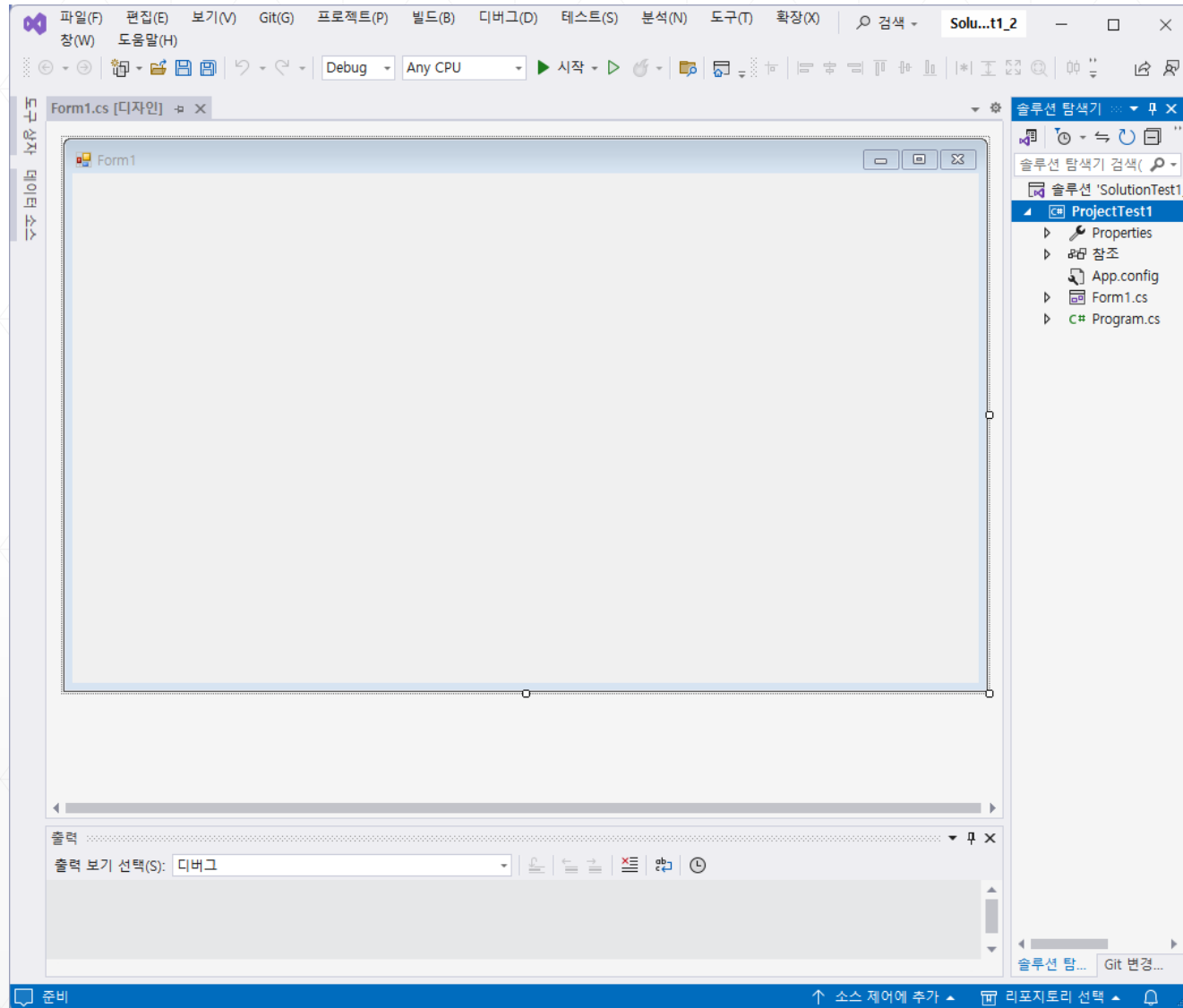
☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

프레임워크(F)
.NET Framework 4.7.2

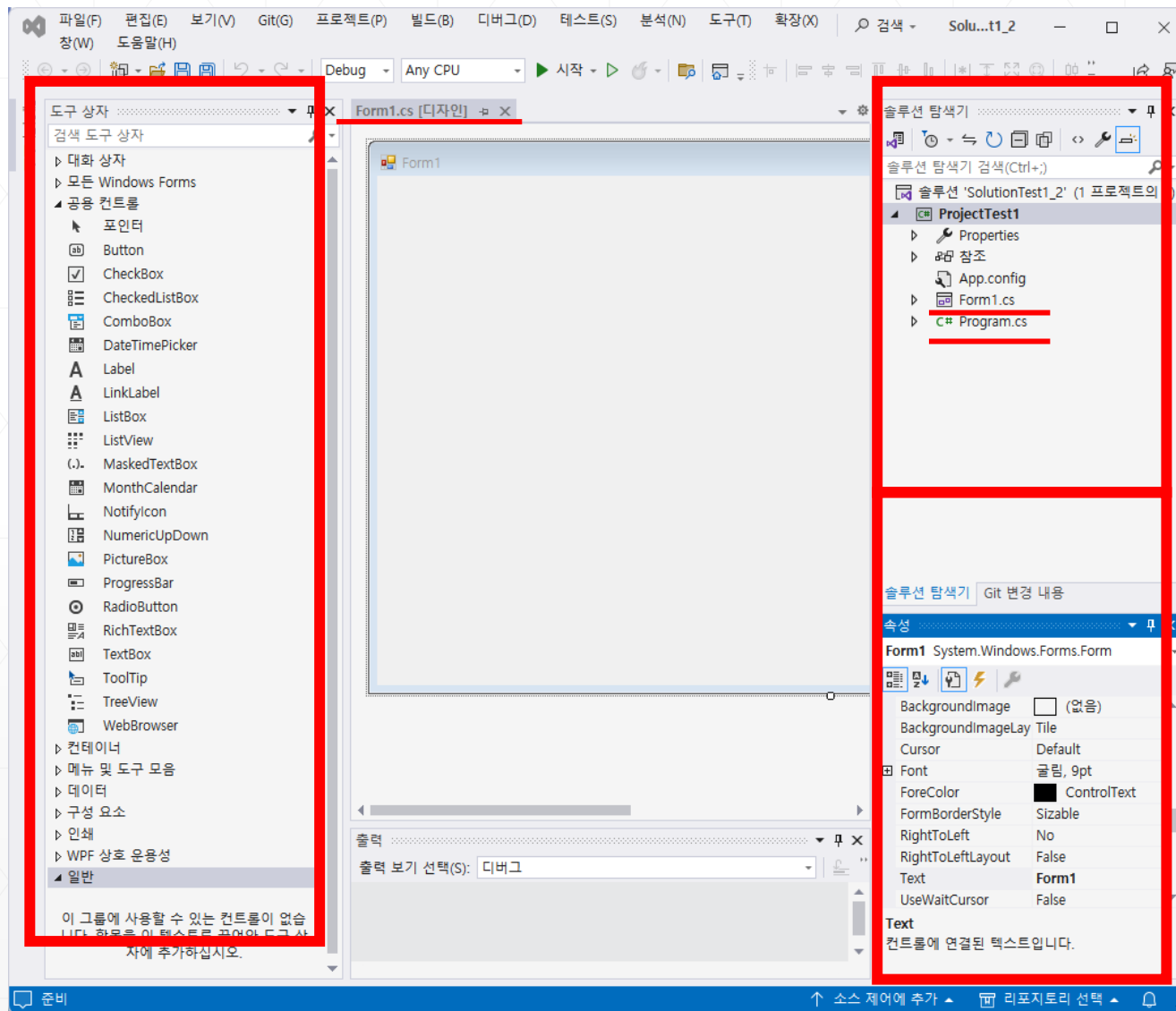
"C:\2024_csharp\SolutionTest1_2\ProjectTest1\ProjectTest1.csproj"에 프로젝트이(가) 만들어집니다.

뒤로(B) 만들기(C)

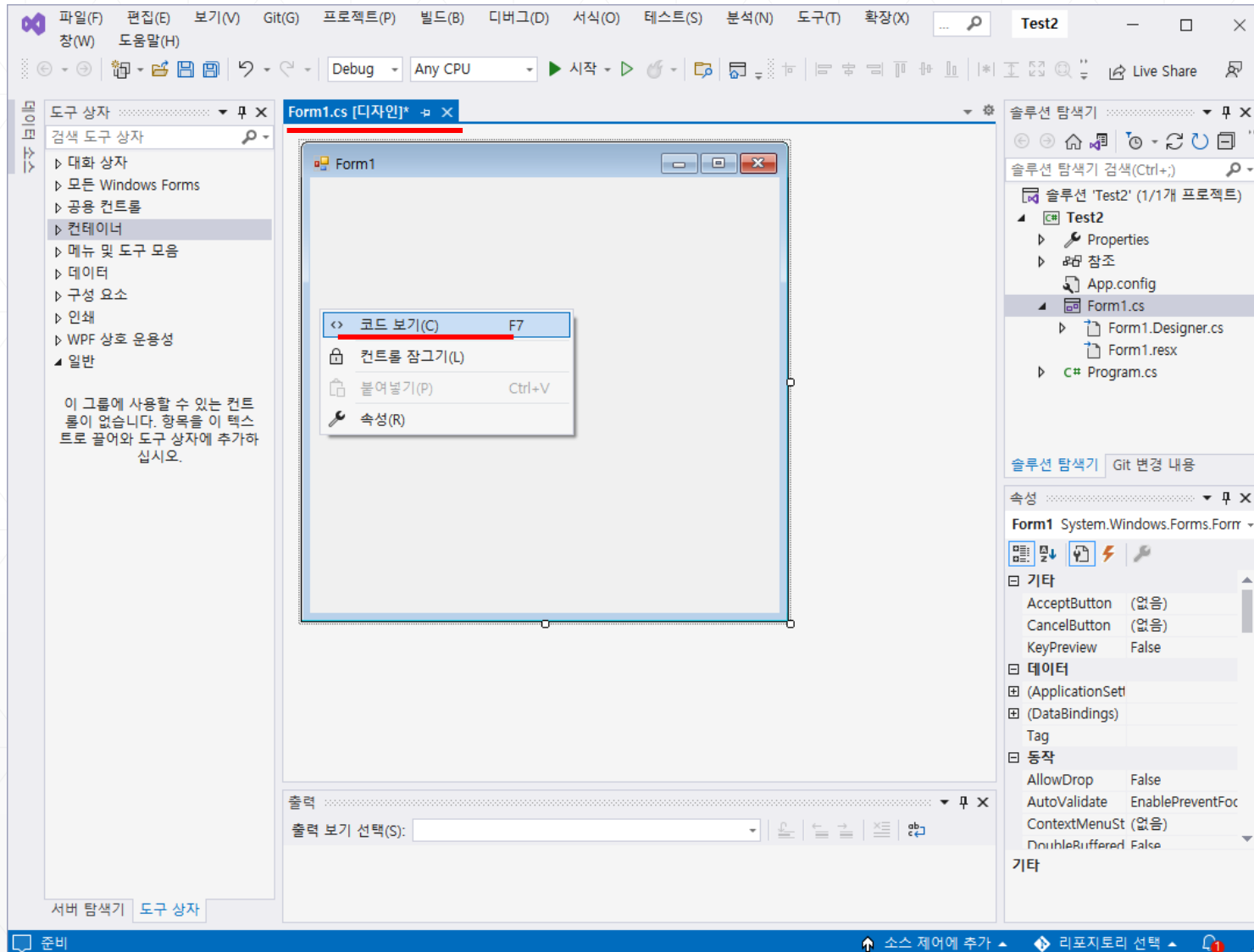
WinForm 프로젝트 만들기 (3)



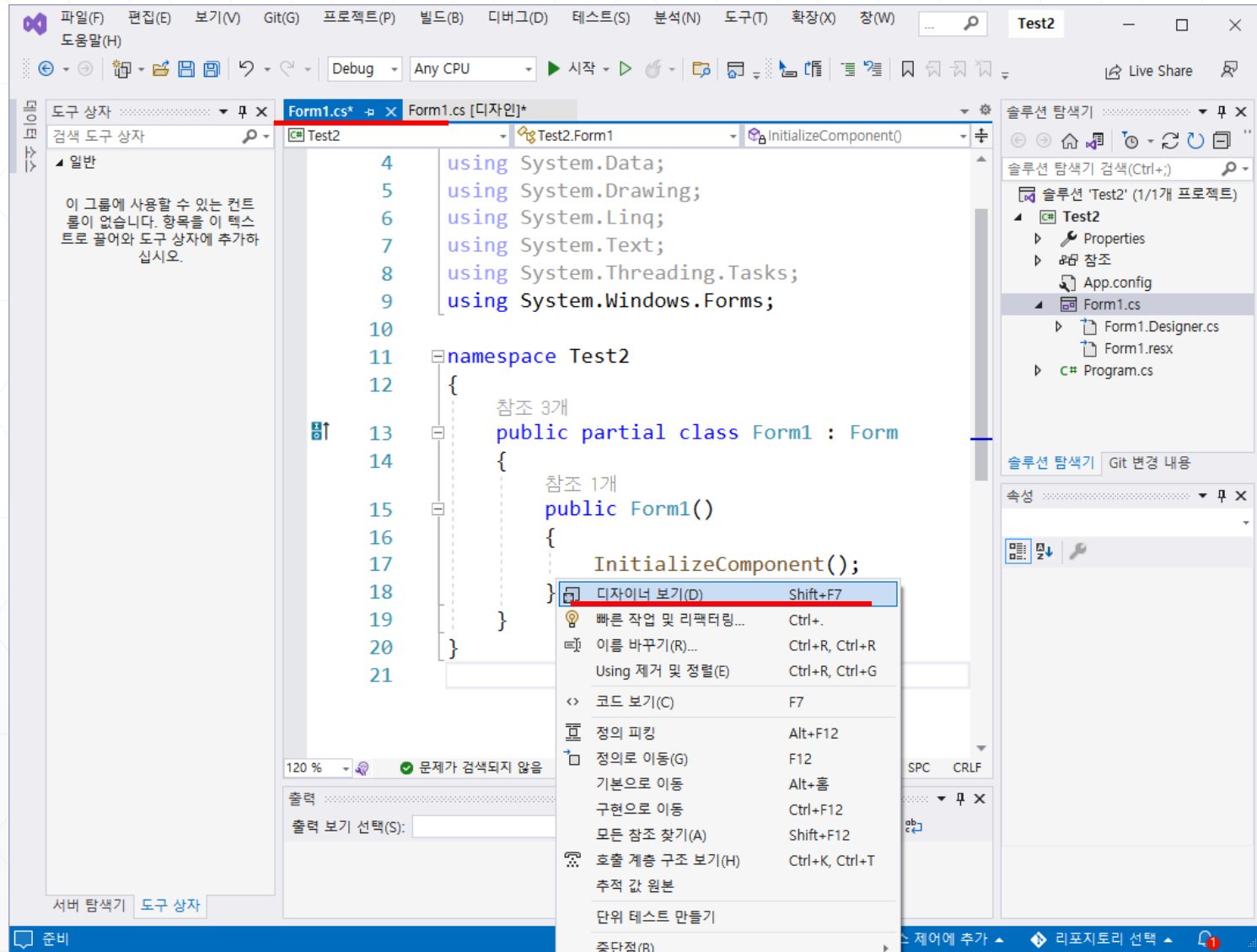
WinForm 프로젝트 만들기 (4)



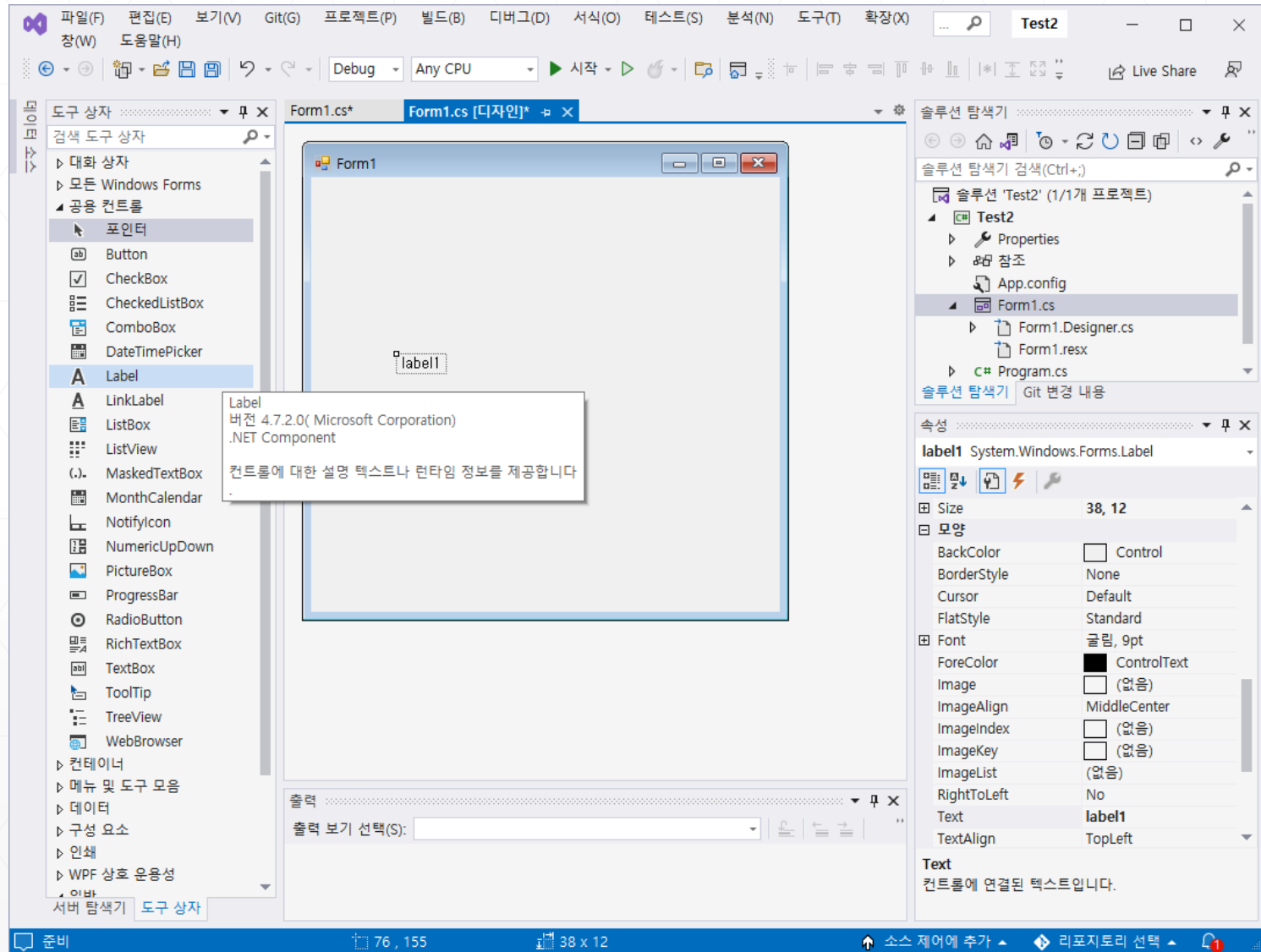
WinForm 프로젝트 만들기 (5)



WinForm 프로젝트 만들기 (6)



WinForm 프로젝트 만들기 (7)



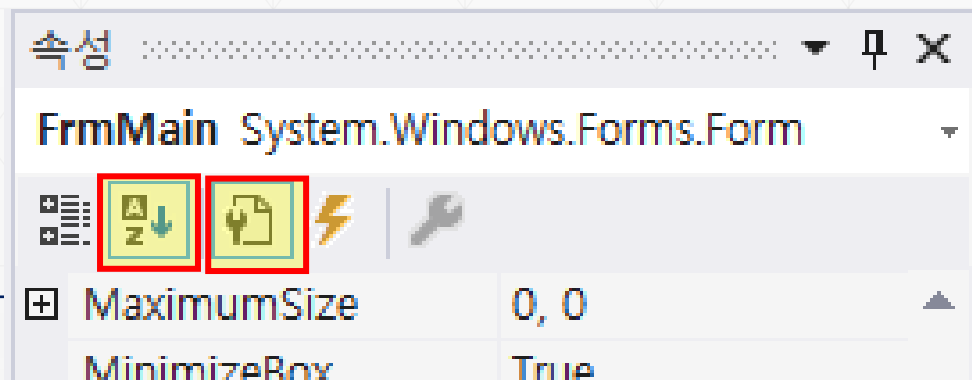
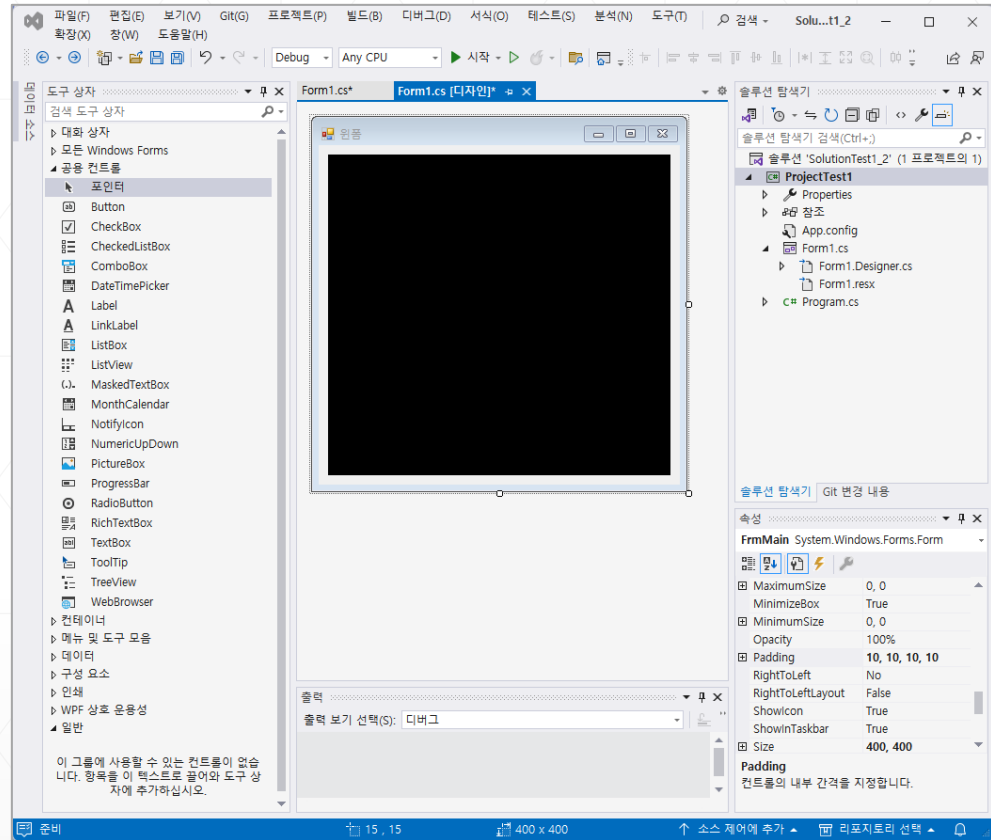
WinForm 프로젝트 만들기 (8)

Form 속성설정

- (Name) : FrmMain
- Padding : 10, 10, 10, 10
- Size : 400, 400
- Text : 원폼

Label 속성설정

- (Name) : lblContent
- AutoSize: False
- BackColor : Black
- Dock : Fill
- ForeColor : Gold
- Padding : 5, 5, 5, 5
- Text :

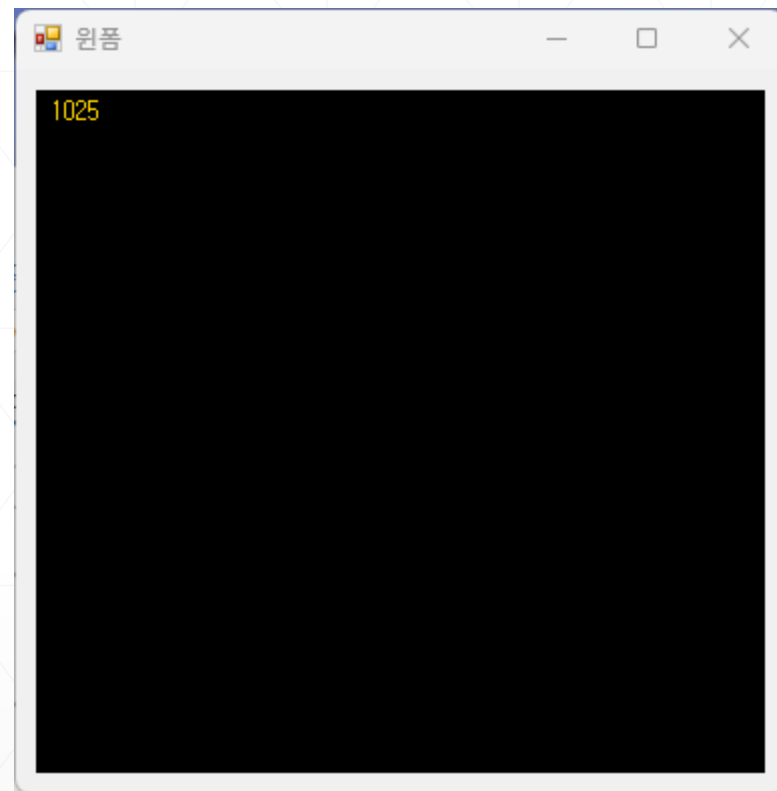


WinForm 프로젝트 만들기 (9)

```
namespace ProjectTest1
{
    참조 3개
    public partial class FrmMain : Form
    {
        참조 1개
        public FrmMain()
        {
            InitializeComponent();

            int a = 10;
            int b = 2 % 3;
            int c = a / b;

            lblContent.Text = a.ToString();
            lblContent.Text += b.ToString();
            lblContent.Text += c.ToString();
        }
    }
}
```



결과에 만족하는가?

환경설정

- 메뉴- 도구 > 옵션
 - 줄번호
 - 텍스트편집기 → C# → “줄 번호” 체크
- 글꼴
 - 환경 → 글꼴 및 색 → 글꼴 변경

추천글꼴

1. Consolas
2. Courier
3. D2Coding (추가설치 필요)

