

C# 소개와 Visual Studio

2018.00.00 HyoJoon Han



프로그래밍 언어

C# 이란?

Visual Studio Community 설치하기

Visual Studio Community 개발 환경 소개

작업 효율을 높여주는 환경 설정과 유용한 기능

- 프로그래밍 언어와 컴파일러
 - 프로그래밍 언어
 - 인간과 컴퓨터가 상호 작용을 할 때 사용하는 언어
 - 인간이 문제 해결을 위해 명령어를 정해진 규칙에 맞추고 코드를 작성
 - 작성된 코드를 컴퓨터가 해석하여 수행
 - 고급 언어
 - 인간이 사용하는 언어와 유사
 - C , JAVA, C#, Python ...
 - 저급 언어
 - 기계가 이해하고 처리할 수 있는 언어
 - 기계어, 어셈블리어

그림 1-1 프로그래밍 언어의 의미



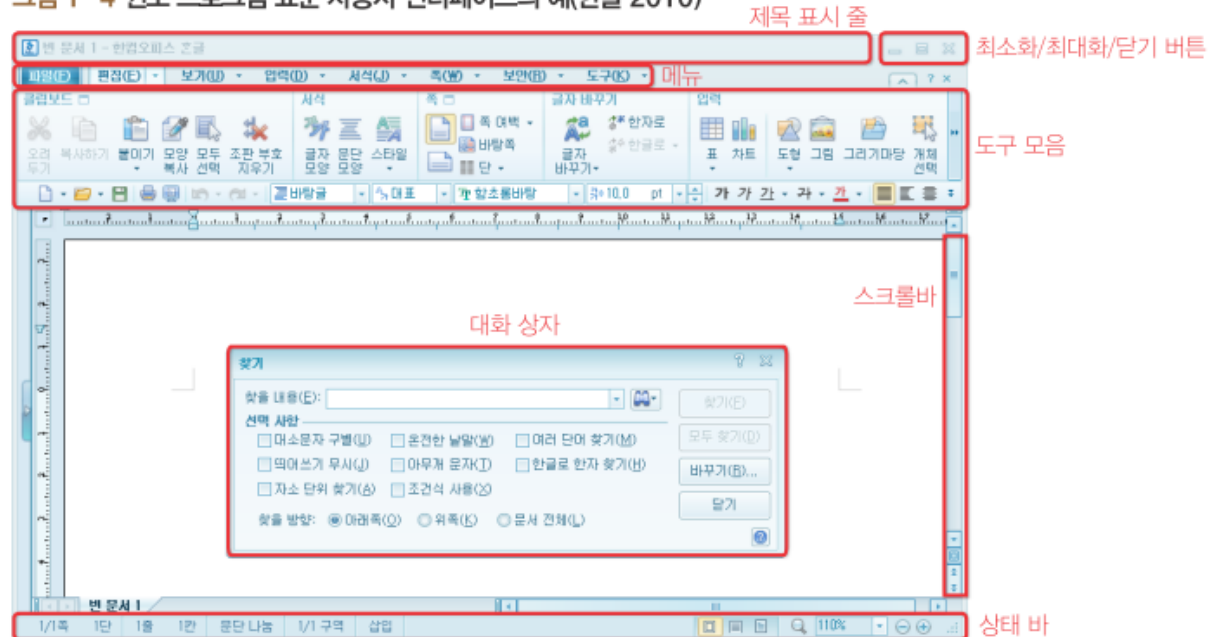
- 프로그래밍 언어와 컴파일러
 - 기계어나 어셈블리어는 인간이 이해하기 어려워서 인간이 사용하는 언어와 유사한 고급 언어가 등장
 - 컴파일러
 - 고급 언어를 컴퓨터가 이해할 수 있는 기계어로 바꿔줌
 - 일반적으로 컴파일러 언어로 작성된 프로그램은 하나의 문서 파일 형태(소스 파일)로 저장되며, 번역기 프로그램인 컴파일러가 이 문서 파일을 컴파일하여 이해하고 실행 가능한 형태인 EXE 파일(실행 파일)을 생성한다.
 - 편집기
 - 프로그램을 작성하는 툴
 - 디버깅
 - 문법에 맞게 프로그램이 작성되지 않아 컴파일 과정에서 발생한 오류를 수정하는 것

그림 1-3 실행 파일 작성 과정



- 윈도우 프로그램
 - 윈도우 운영체제에서 동작하는 프로그램
 - 공통된 그래픽 사용자 인터페이스 (Graphic User Interface, GUI)를 사용한다.
 - 예 - MS 워드, 한글, 계산기 ...
 - C#, VB.net으로 이와 같은 윈도우 프로그램을 개발할 수 있다.

그림 1-4 윈도 프로그램 표준 사용자 인터페이스의 예(한글 2010)



- 마이크로소프트에서 개발한 객체 지향 프로그래밍 언어
- 기존 프로그래밍 언어의 생산성을 개선하여 성능이 굉장히 높음
- 현재 윈도우, 맥, 리눅스, 안드로이드, 아이폰 등의 다양한 운영체제나 플랫폼에서 동작
- 윈도우에서 동작하는 .net(닷넷) 플랫폼과 대부분의 운영체제에서 동작하는 모노 플랫폼에서 작동하는 프로그램을 만들 수 있음

- .Net
 - .Net은 마이크로소프트에서 Java 에 대응하여 나온 기술
 - .Net Visual Studio, .Net Framework, C#, ASP.NET 등으로 구성
 - 언어의 통합성을 제공
 - 다른 언어로 작성된 부분들을 통합하여 하나의 큰 프로그램을 완성할 수 있음

- 플랫폼과 소프트웨어 플랫폼
 - 플랫폼(Platform) : 소프트웨어 응용 프로그램의 실행에 사용되는 하드웨어와 소프트웨어의 집합

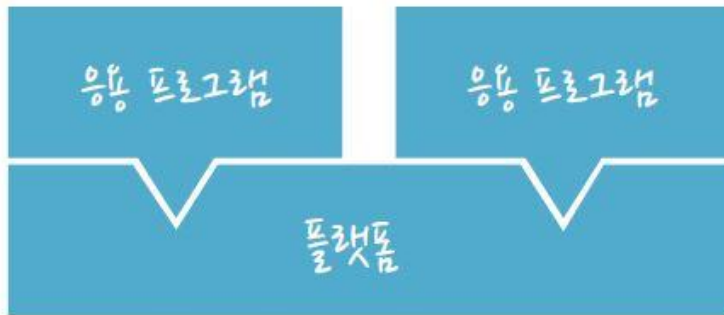
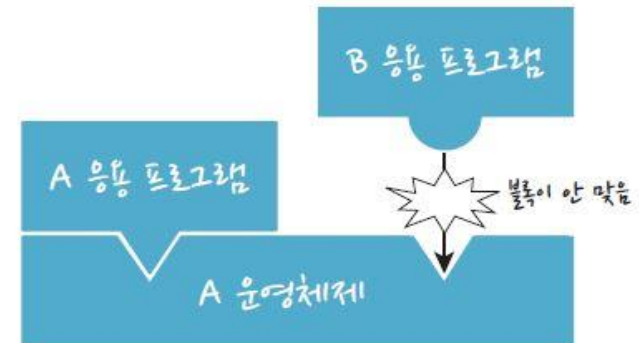
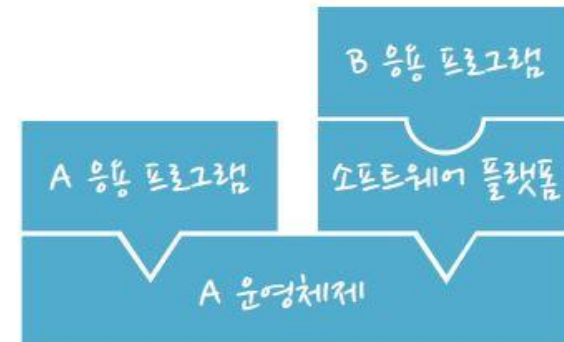


그림 1-1 플랫폼과 응용 프로그램

- 소프트웨어 플랫폼
 - [그림 1-2]와 같이 중간 레고 블록 역할을 함
 - 예 : 자바 가상 버신, 액션스크립트, 닷넷 플랫폼



(a) 소프트웨어 플랫폼이 없는 경우: B 운영체제용으로 만들어진 B 응용 프로그램을 A 운영체제에서 실행 불가



(b) 소프트웨어 플랫폼이 있는 경우: B 운영체제용으로 만들어진 B 응용 프로그램을 A 운영체제에서 실행 가능

그림 1-2 소프트웨어 플랫폼의 역할

- 닷넷 플랫폼
 - 마이크로소프트 사가 만든 중간 레고 블록, 플랫폼의 기본적인 발전 형태
 - 초기에는 윈도우에서만 동작했지만 현재는 마이크로소프트 사가 활용할 수 있는 모든 프로그래밍 언어(20개 이상)를 연결 가능
 - C#은 모노 플랫폼이라는 소프트웨어 플랫폼 위에서도 동작



그림 1-3 닷넷 플랫폼



그림 1-4 모노 플랫폼이 지원하는 운영체제와 사용할 수 있는 프레임워크(또는 엔진)

- C#의 기타 활용
 - 게임 프레임워크(게임 엔진) : 유니티
 - 모바일 응용 프로그램 프레임워크(안드로이드와 아이폰) : Xamarin

- C#으로 할 수 있는 일
 - GUI 개발
 - 윈도우에서 동작하는 GUI(Graphic User Interface) 프로그램 개발을 위한 윈도우 폼과 WPF 제공
 - 윈도우 폼(Windows Form)
 - C#의 가장 기초 프레임워크로 , C++를 사용한 윈도우 개발(Win32 API 또는 MFC)을 C#으로 옮겨 놓은 형태
 - 개발자가 폼 디자이너를 이용해 도구 상자에서 버튼이나 콤보 박스 등의 컨트롤을 윈도우에 배치할 때마다 폼 디자이너가 프로그램의 UI를 표시하면서 C# 코드를 자동으로 만들어 줌
 - WPF(Windows Presentation Foundation)
 - 현대적인 개발 패턴 중에 MVC 패턴과 MVVM 패턴을 적용해 개발 생산성을 향상시킨 프레임워크
 - DirectX 등의 기능도 추가로 내장하여 3D 그래픽까지 자체적으로 처리 가능

- 웹 개발
 - C#이 가장 대표적으로 사용되는 부분
 - 마이크로소프트 사는 2가지 프레임워크 지원(ASP.NET 프레임워크와 ASP.NET MVC)

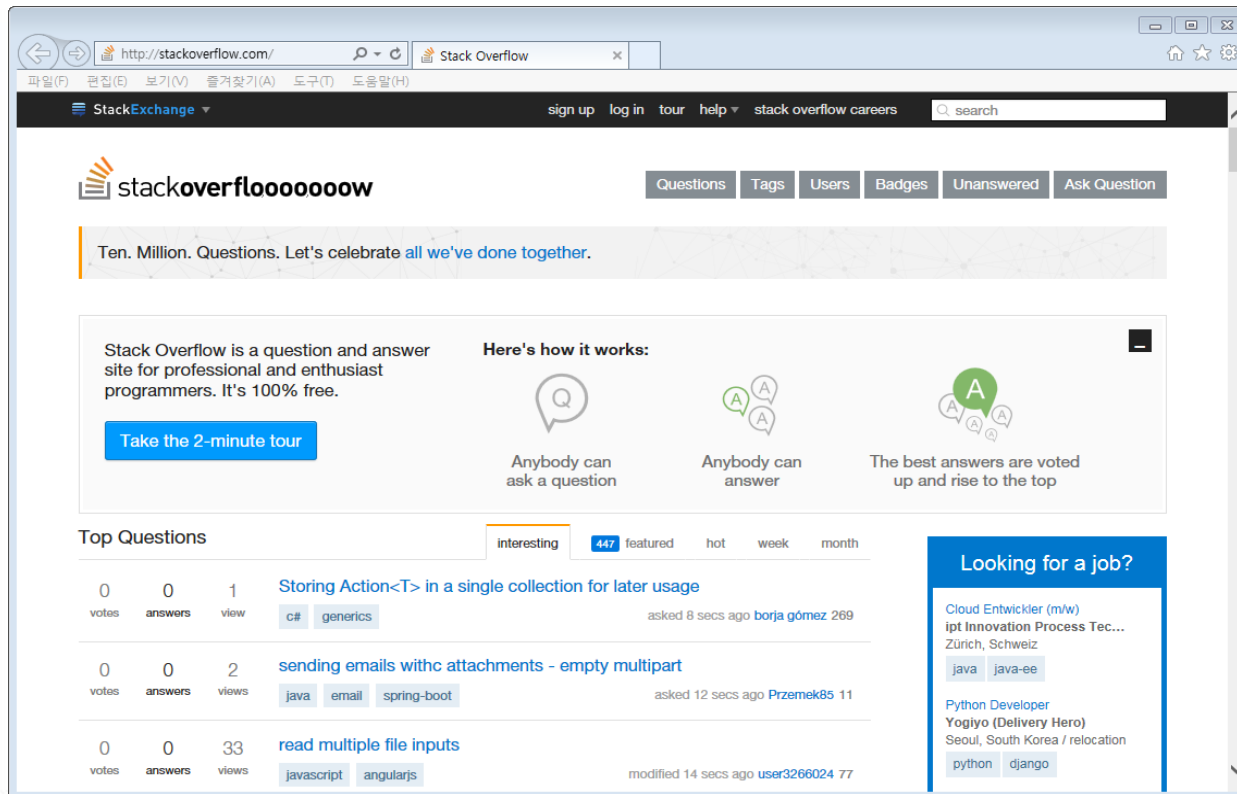


그림 1-10 ASP.NET MVC로 개발된 스택오버플로

- 게임 개발

- C#을 이용해 게임 클라이언트와 게임 서버 개발 가능
- 게임 클라이언트 개발
 - 유니티 엔진 개발로 C#으로 게임 클라이언트 개발 활성화
 - 유니티는 모노 플랫폼을 사용해 다양한 플랫폼에서 작동하는 게임 개발 가능








iOS							
	WebGL	PS3	PS4	PSVITA	XBOX ONE	XBOX 360	WiiU
			Gear VR				

그림 1-12 유니티가 지원하는 플랫폼

- 게임 서버 개발
 - 간단한 모바일 게임 서버는 웹 서버처럼 개발
 - 대규모 MMOPRG 게임 서버도 개발

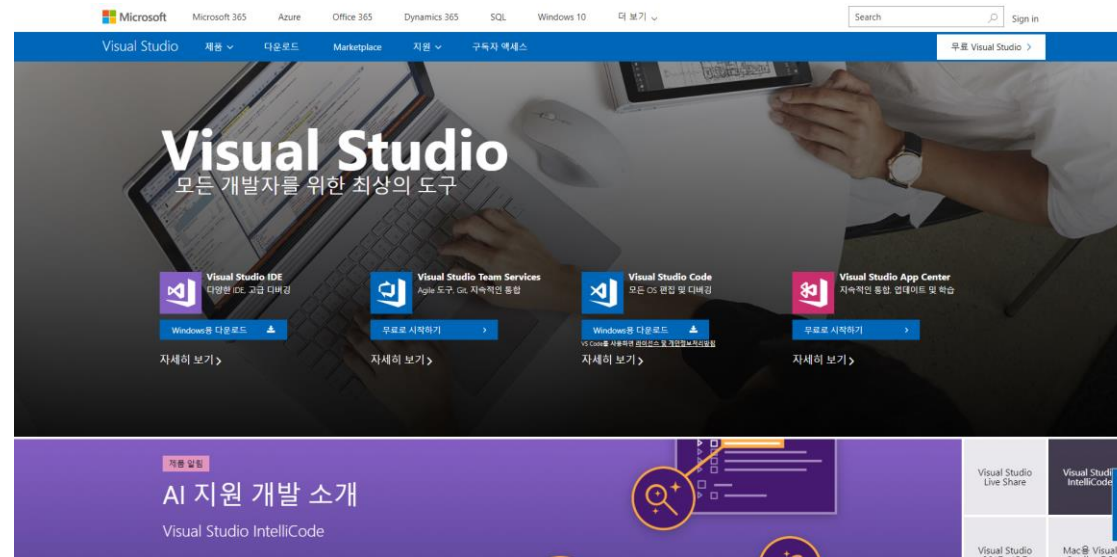
- IoT(사물 인터넷, Internet of Things) 개발
 - C#을 사용해 다양한 IoT 개발 가능



그림 1-13 윈도 10 IoT 코어

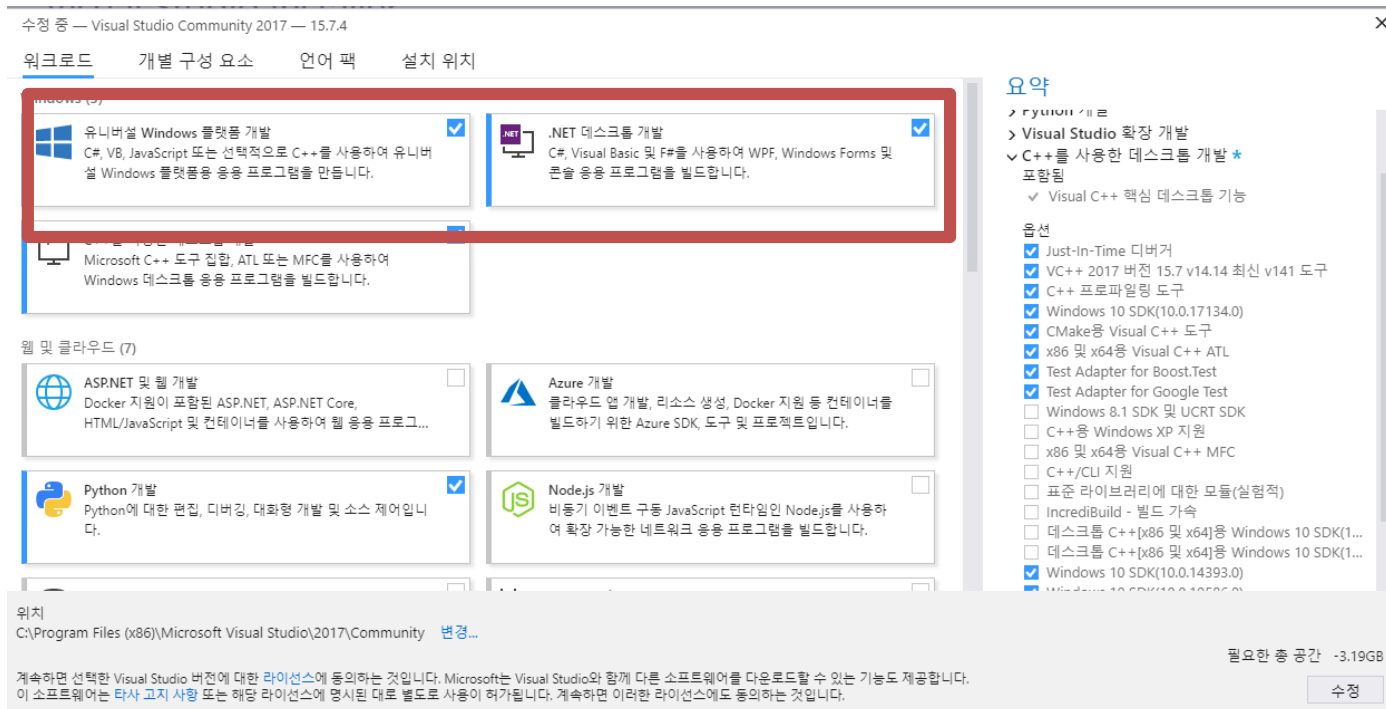
Visual Studio Community 설치하기

- 개발 환경 설치
 - Visual Studio 사용 (C++, C#, Python 등 다양한 언어를 사용 가능한 에디터)
 - 마이크로소프트 사의 비주얼 스튜디오 사이트에 들어가서 설치 파일 다운
 - <https://visualstudio.microsoft.com/ko>
 - [다운로드] – [Visual Studio Community] – [무료 다운로드]
 - 설치 가이드
 - <https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio>



Visual Studio Community 설치하기

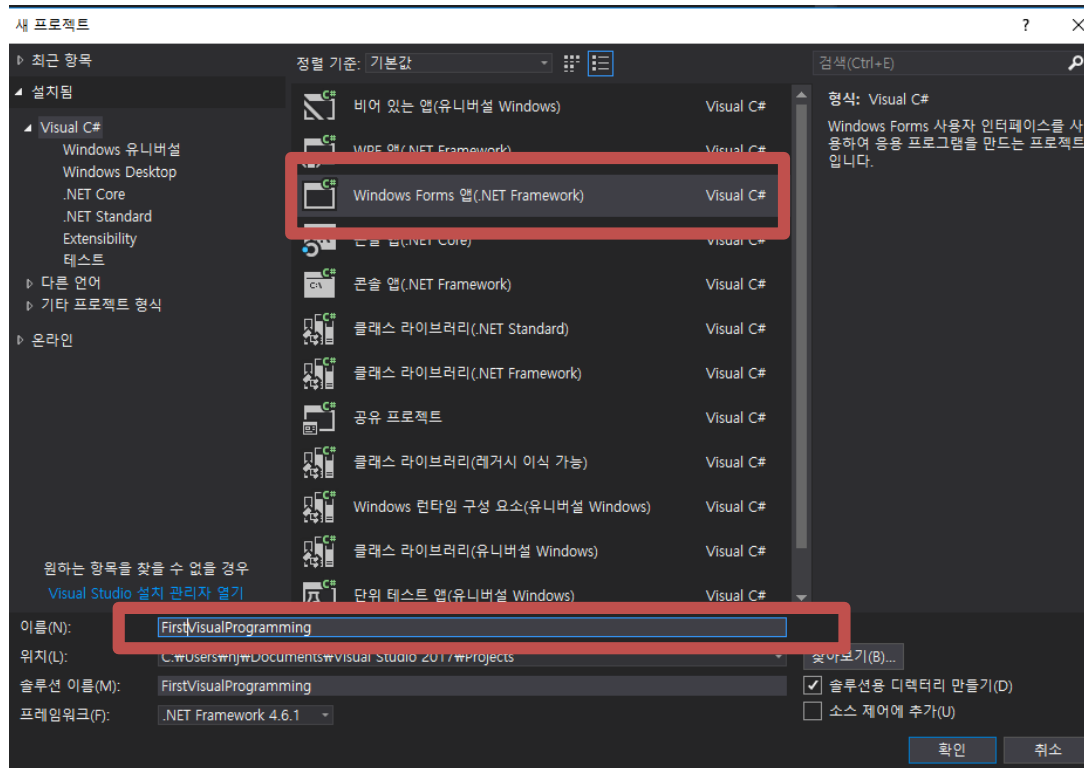
- Visual Studio 설치 관리자
 - 필수 선택 항목
 - 유니버설 Windows 플랫폼 개발
 - .NET 데스크톱 개발



Visual Studio Community 설치하기

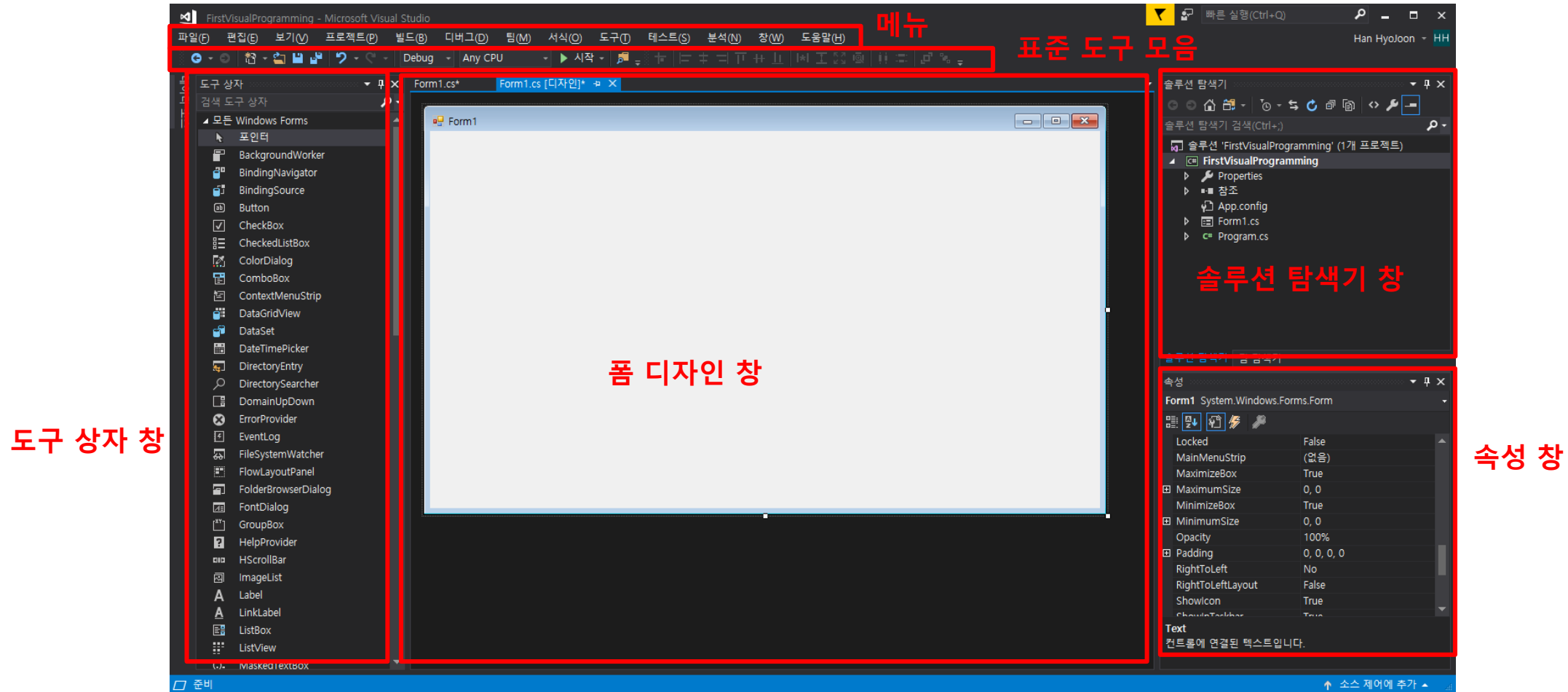
- 프로젝트 생성

- ① [파일] – [새로 만들기] – [프로젝트] 메뉴 선택 (or Ctrl+Shift+N 키 입력)
- ② [Visual C#] – [Windows Forms 앱(.NET Framework)] 선택
- ③ 프로젝트 이름 입력 후 [확인]을 누름

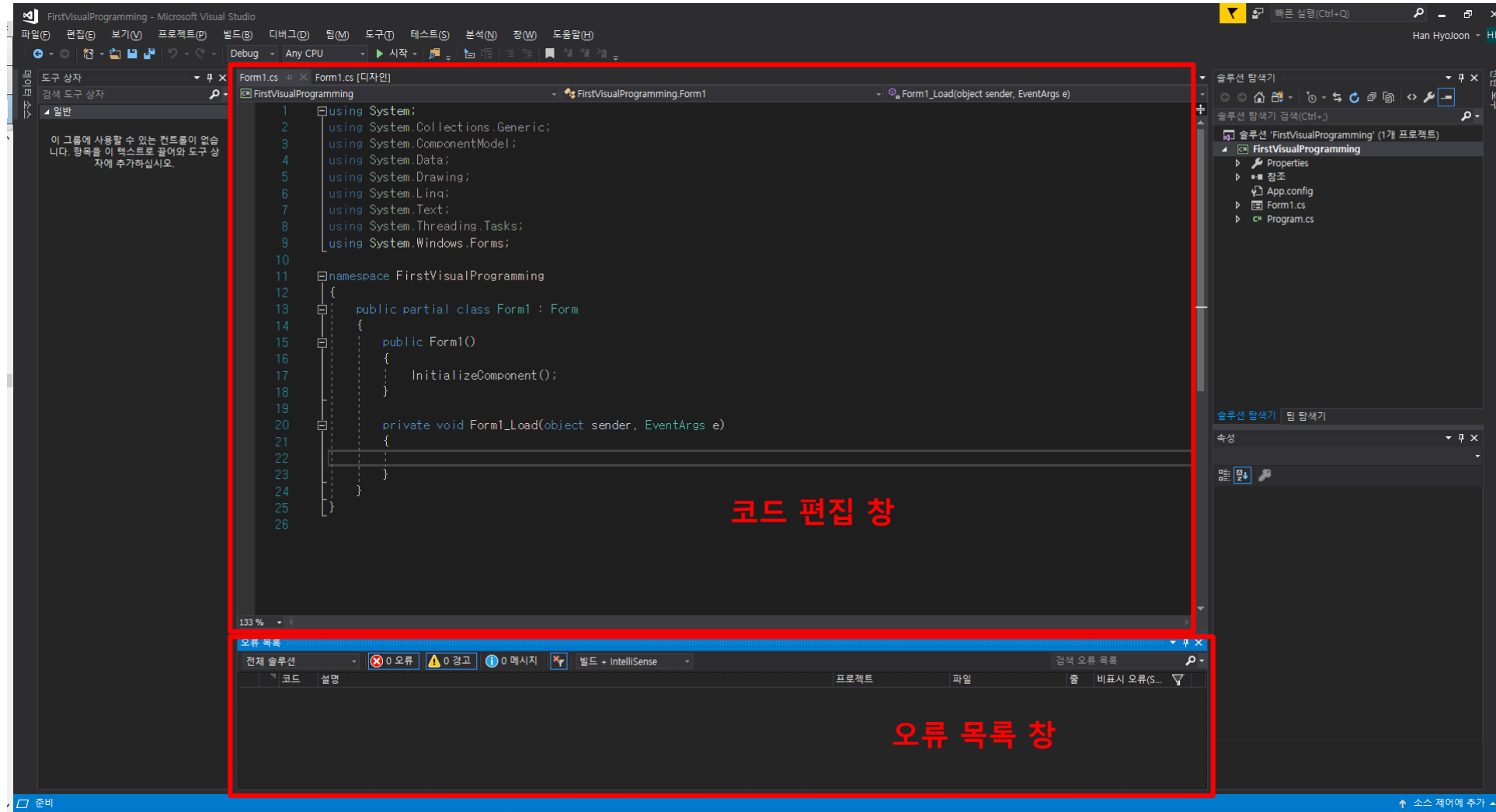


Visual Studio Community 개발 환경 소개

- 프로젝트 화면 구성



Visual Studio Community 개발 환경 소개



- Visual Studio Community 화면 구성
 - 메뉴
 - 모든 작업 명령을 풀다운 메뉴 방식으로 제공한다.
 - 표준 도구 모음
 - 자주 사용하는 메뉴 항목을 아이콘을 만들어 모아 놓은 곳이다. 아이콘에 커서를 놓으면 그 기능을 풍선 도움말로 보여준다.
 - 도구 상자 창
 - 프로그램을 작성할 때 사용하는 컨트롤을 모아 놓은 곳이다. 컨트롤은 텍스트 박스, 커맨드 버튼, 체크 박스 등 화면 디자인을 할 때 필요한 기본적인 구성 요소를 시각화 해둔 것을 말한다.
 - 폼 디자인 창
 - 컨트롤을 배치하여 실행 시 사용자에게 보일 화면을 디자인하는 곳이다. 컨트롤을 정렬된 형태로 쉽게 그릴 수 있도록 모눈을 표시할 수도 있다.
 - 솔루션 탐색기 창
 - 프로젝트를 구성하는 폼과 모듈, 클래스와 리소스, 문서 파일 구조 등을 보여준다. 폼과 모듈의 추가나 삭제, 폼 창과 코드 편집 창의 활성화 등을 편리하게 할 수 있다.

– 속성 창

- 각 컨트롤의 속성을 보여주는 창으로, 속성 값을 직접 편집하고 그 결과를 즉시 확인할 수 있다. 코드 편집 창에서 속성 값을 지정할 수도 있지만 모두 일대일로 대응하지는 않는다.

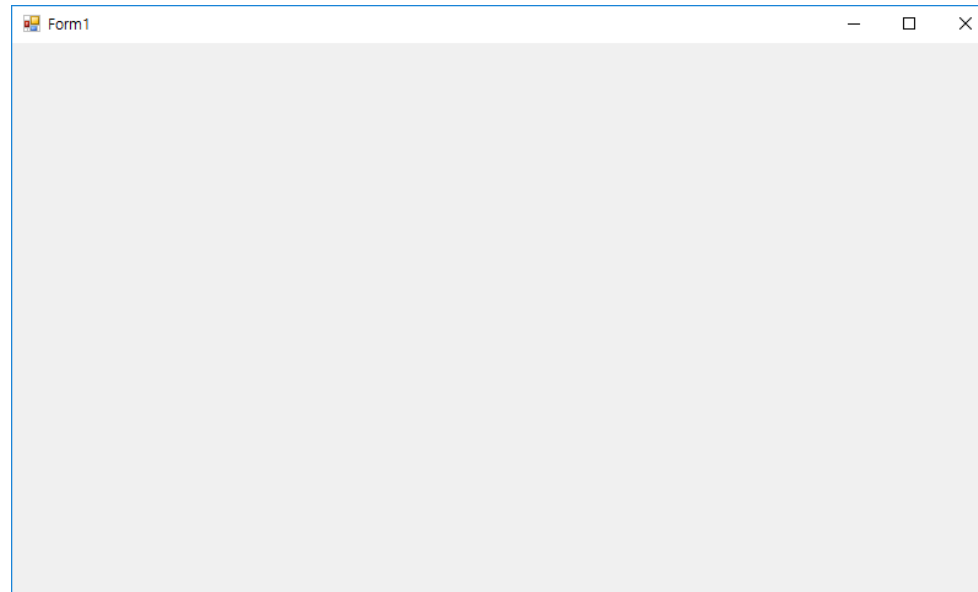
– 오류 목록 창

- 코드 작성 시에 구문 오류나 컴파일 시에 컴파일 오류가 있는 경우에 오류 내용을 보여주는 창이다.
- 프로그램 실행 시 활성화 되며, 비활성화 시 [보기] – [오류 목록] 을 통해 활성화 가능

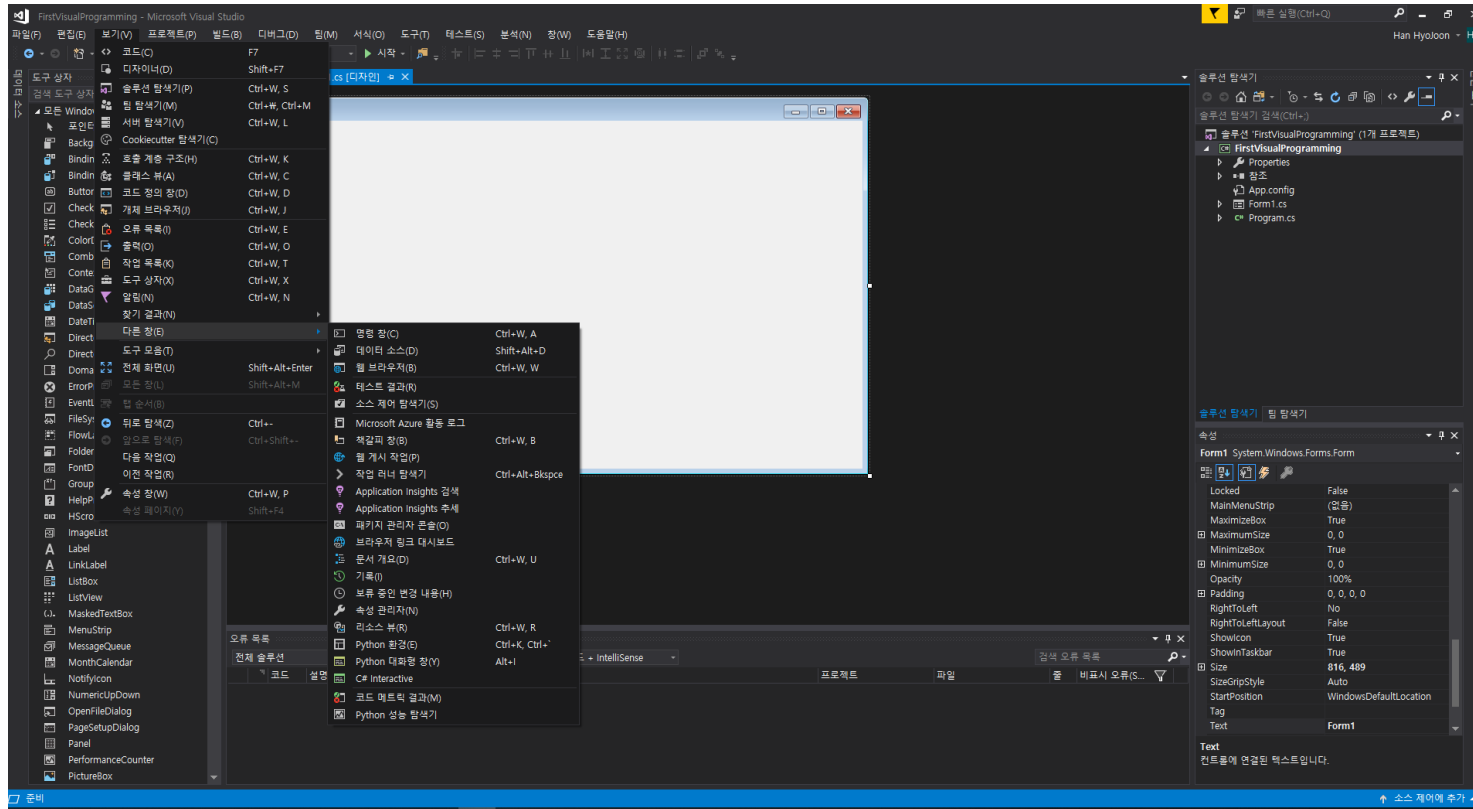
– 코드 편집 창

- 프로그램 코드를 직접 입력하여 편집하는 곳인 코드 편집 창은 처음부터 나타나지는 않으며, 다음과 같이 세 가지 방법으로 열 수 있다.
- 폼 창에서 코드를 작성할 컨트롤을 더블클릭한다.
- 컨트롤에서 마우스 오른쪽 버튼을 클릭한 다음 [코드 보기]를 선택한다.
- 메인 메뉴에서 [보기] – [코드] 를 선택한다.

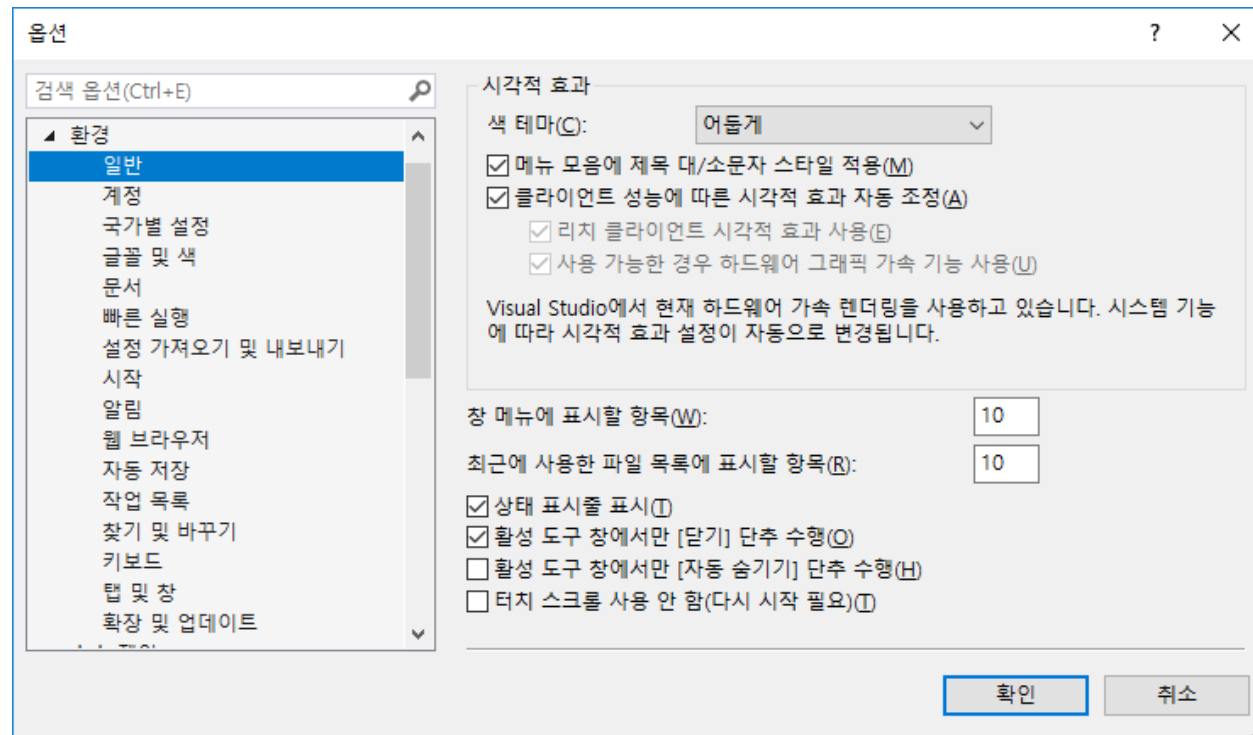
- 프로젝트 실행
 - 방법 1 : [시작] 또는 [디버그] - [디버깅 시작] 메뉴 또는 단축키(F5) 누름
 - 비어있는 폼 화면이 실행
 - 디버그 모드 실행(오류 등을 확인하기 위해 사용되는 모드)
 - 방법 2 : [디버그] - [디버그하지 않고 시작] 메뉴 또는 Ctrl + F5 키를 누름



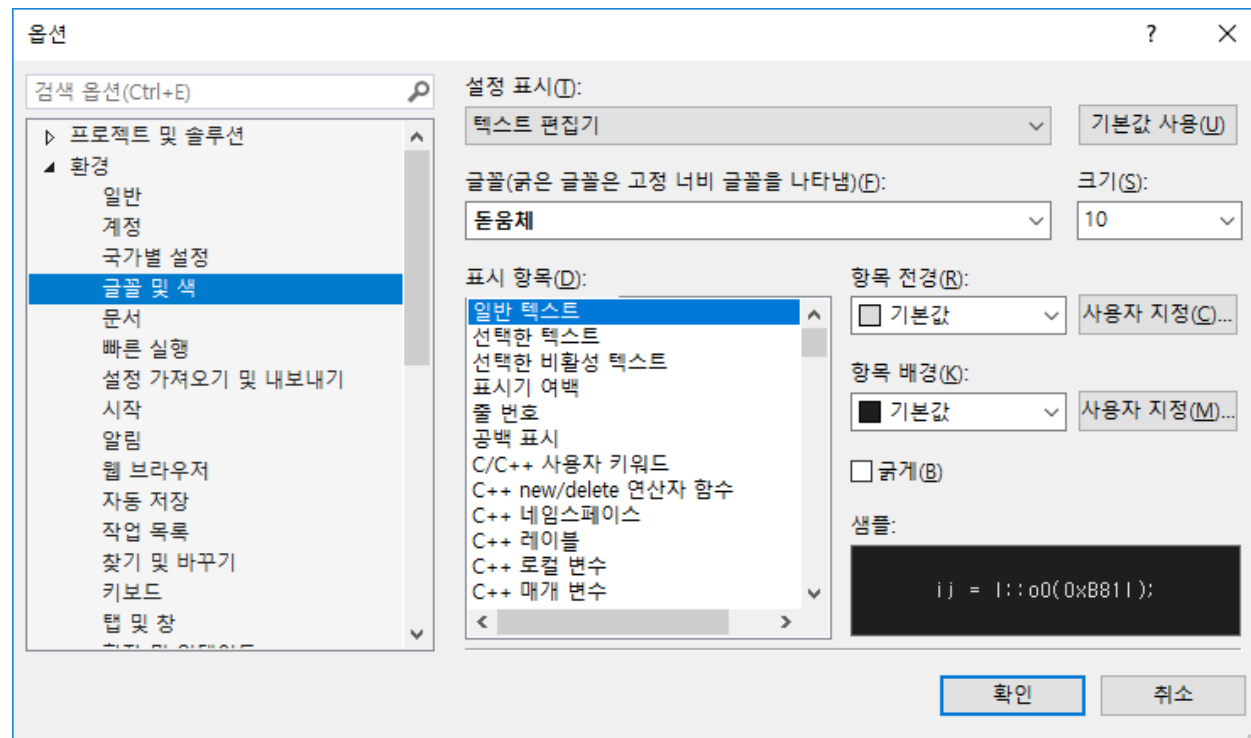
- 작업 창 화면에 표시하기
 - [보기] 메뉴를 이용하여 화면에서 사라진 솔루션 탐색기 창, 속성 창, 도구 상자 창 등이 다시 화면에 나타나게 할 수 있다.



- 작업 환경 설정하기
 - 효율적인 개발 환경 설정하기
 - [도구] - [옵션] 을 선택하면 옵션 대화 상자가 나타나는데 여기서 개발에 필요한 다양한 환경을 설정할 수 있다.



- 작업 환경 설정하기
 - 코드 창의 텍스트 크기 변경하기
 - [환경] - [글꼴 및 색] 을 선택하면 글꼴, 색, 글꼴 크기 등을 변경할 수 있다.



- 유용한 단축키

표 1-2 도구 상자 단축키

단축키	기능
[Enter]	선택한 컨트롤을 폼에 배치하기
[Tab]	도구 상자에 있는 다음 컨트롤 선택하기
[End]	도구 상자에 있는 마지막 컨트롤 선택하기
[Home]	도구 상자에 있는 처음 컨트롤 선택하기
[Alt] + [F4]	도구 상자 닫기

표 1-3 코드 편집 창 단축키

단축키	기능	단축키	기능
[Ctrl] + [P]	인쇄하기	[F2]	개체 찾아보기
[Ctrl] + [H]	바꾸기	[F3]	다음 찾기
[Ctrl] + [C]	복사하기	[F4]	속성 창 사용하기
[Ctrl] + [V]	붙여 넣기	[F5]	실행하기
[Ctrl] + [F]	찾기	[F8]	코드를 한 단계씩 실행하거나 프로시저 단위로 실행하기

- 유용한 단축키

Ctrl + Z	실행 취소하기	Ctrl + →	한 단어만큼 오른쪽으로 이동하기
Ctrl + J	속성 및 메서드 목록 보기	Ctrl + ←	한 단어만큼 왼쪽으로 이동하기
Ctrl + I	요약 정보 보기	Ctrl + ↑	이전 프로시저의 첫 행으로 이동하기
Ctrl + G	실행 창 직접 사용하기	Ctrl + ↓	다음 프로시저의 첫 행으로 이동하기
Ctrl + R	프로젝트 탐색기 사용하기		

표 1-4 품 창 단축키

단축키	기능	단축키	기능
F7	코드 편집 창으로 이동하기	Ctrl + Z	방금 삭제한 개체 되살리기
Ctrl + C	복사하기	Ctrl + E	메뉴 편집기 사용하기
Ctrl + V	붙여 넣기	Ctrl + J	맨 앞으로 가져오기
Ctrl + X	오려 붙이기	Ctrl + K	맨 뒤로 보내기
Delete	선택한 개체 삭제하기		