

C# 언어의 이해

C# 프로그래밍

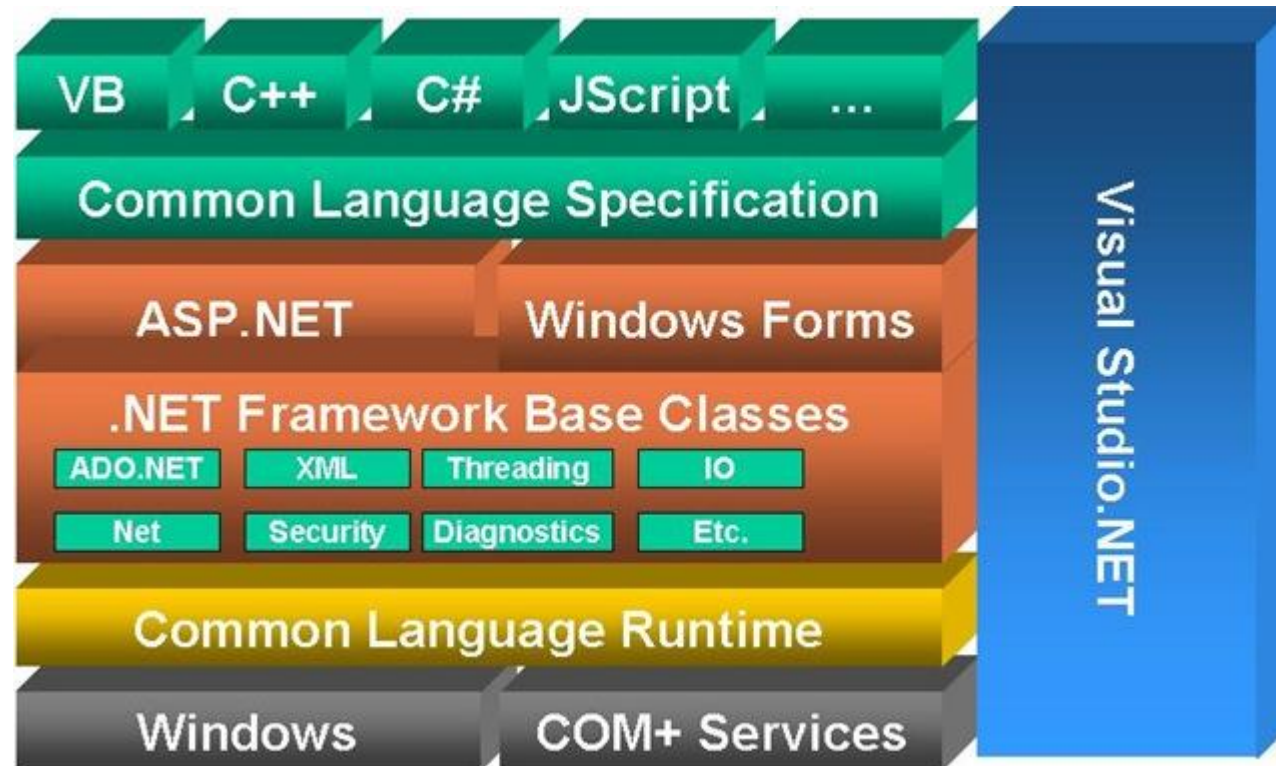
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace foreachEx
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

.NET 프레임워크

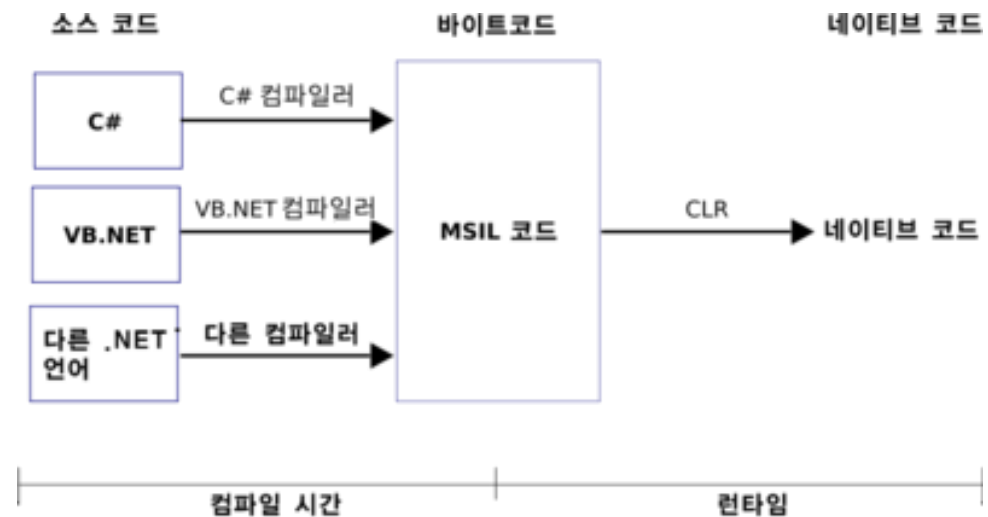
- 마이크로소프트에서 개발한 윈도우 프로그램 개발 및 실행 환경
- 네트워크 작업, 인터페이스 등의 많은 작업을 캡슐화
- 공통 언어 런타임(Common Language Runtime)(CLR)이라는 이름의 가상 머신 위에서 작동

.NET 프레임워크의 구조



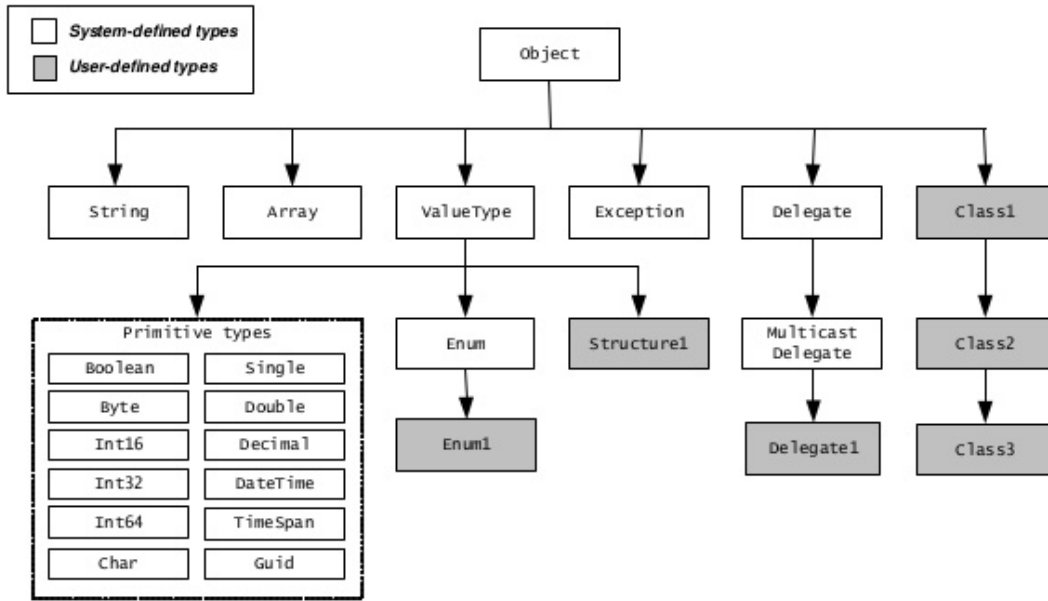
공통 언어 런타임(CLR)

- 마이크로소프트 .NET 프레임워크가 동작하는 가상 머신
- 프로그래머들이 CPU의 자세한 부분에까지 신경 쓰지 않아도 되게 해 준다.
 - 메모리 관리
 - 스레드 관리
 - 예외 처리
 - 쓰레기 수집
 - 보안



공통 형식 시스템(Common Type System)

Common Type System (CTS)



CTS	크기	부호	C#	C	C++
System.Byte	1	없음	byte	unsigned char	unsigned char
System.Sbyte	1	있음	sbyte	char	signed char
System.Int16	2	있음	short	short	short
System.UInt16	2	없음	ushort	unsigned char	unsigned short
System.Int32	4	있음	int	int	int 또는 long
System.UInt32	4	없음	uint	unsigned char	unsigned(long)
System.Int64	8	있음	long	long	_int64
System.UInt64	8	없음	ulong	unsigned long	unsigned_int64
System.Char	2	없음	char		wchar_1
System.Single	4	있음	float	float	float
System.Double	8	있음	double	double	double
System.Decimal	16	있음	decimal		Decimal
System.Boolean	1		bool		bool
System.String	가변		string		string
System.Object	가변		object		Object*

정수 형식

CTS	크기	부호	C#	C	C++
System.Byte	1	없음	byte	unsigned char	unsigned char
System.Sbyte	1	있음	sbyte	char	signed char
System.Int16	2	있음	short	short	short
System.UInt16	2	없음	ushort	unsigned char	unsigned short
System.Int32	4	있음	int	int	int 또는 long
System.UInt32	4	없음	uint	unsigned char	unsigned(long)
System.Int64	8	있음	long	long	_int64
System.UInt64	8	없음	ulong	unsigned long	unsigned_int64
System.Char	2	없음	char		wchar_1

Type	Range	Size
sbyte	-128 to 127	Signed 8-bit integer
byte	0 to 255	Unsigned 8-bit integer
char	U+0000 to U+ffff	Unicode 16-bit character
short	-32,768 to 32,767	Signed 16-bit integer
ushort	0 to 65,535	Unsigned 16-bit integer
int	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
uint	0 to 4,294,967,295	Unsigned 32-bit integer
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer
ulong	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer

부동 소수점 형식

System.Single	4	있음	float	float	float
System.Double	8	있음	double	double	double

Type	Approximate range	Precision
float	$\pm 1.5e-45$ to $\pm 3.4e38$	7 digits
double	$\pm 5.0e-324$ to $\pm 1.7e308$	15-16 digits

Decimal 형식

- 128비트 데이터 형식
- 부동 소수점 형식에 비해 전체 자릿수는 크고 범위는 작다.
- 재무 및 통화 계산에 적합

형식	근사 범위	전체 자릿수 (Precision)	.NET Framework 형식
decimal	$(-7.9 \times 10^{28} - 7.9 \times 10^{28}) / (10^0 - 28)$	28-29개의 유효 자릿수	Decimal

문자열 형식

```
string message1;  
string message2 = null;  
string message3 = System.String.Empty;  
string oldPath = "c:\\Program Files\\Microsoft Visual Studio 8.0";  
string newPath = @"c:\Program Files\Microsoft Visual Studio 9.0";  
System.String greeting = "Hello World!";  
const string message4 = "You can't get rid of me!";  
  
char[] letters = { 'A', 'B', 'C' };  
string alphabet = new string(letters);
```

논리 형식

- true 및 false를 저장하는 형식

CTS	크기	부호	C#	C	C++
System.Boolean	1		bool		bool

형식 접미사

- 형식 접미사를 꼭 적어주어야 함

값 형식	형식 접미사
bool	
byte	
char	
decimal	M 또는 m
double	D 또는 d
enum	
float	F 또는 f
int	
long	L 또는 l
sbyte	
short	
struct	
uint	U 또는 u
ulong	UL 또는 ul
ushort	

형변환

- 형변환(Type Casting)
 - Giraffe g2 = (Giraffe) a;
- 문자열을 숫자로

```
int number = Int32.Parse(value);
```

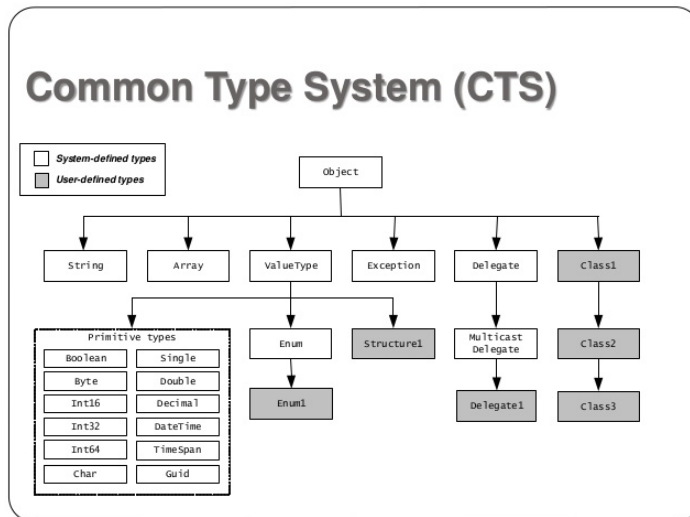
- 숫자에서 문자로

```
string myString = myInt.ToString();
```

```
double x = 1234.7;  
  
int a;  
  
a = (int)x;
```

object 형식

- 모든 데이터 형식은 object 를 상속 받음.
- C++ 의 레퍼런스 역할을 할 수 있다.
- 데이터는 힙 메모리에 저장



```
class ObjectTest
{
    public int i = 10;
}

class MainClass2
{
    static void Main()
    {
        object a;
        a = 1;    // an example of boxing
        Console.WriteLine(a);
        Console.WriteLine(a.GetType());
        Console.WriteLine(a.ToString());

        a = new ObjectTest();
        ObjectTest classRef;
        classRef = (ObjectTest)a;
        Console.WriteLine(classRef.i);
    }
}
```

var 형식

- 암시적으로 형식화된 지역 변수
- 할당된 값을 보고 컴파일러가 알아서 형식을 지정
- 클래스의 멤버 변수로 사용할 수 없음
- 반드시 값을 할당해야 함

```
var i = 5;  
var s = "Hello";  
var a = new[] { 0, 1, 2 };  
var anon = new { Name = "Terry", Age = 34 };
```

Nullable 형식

- Null 값을 가질 수 있는 형식

type? 변수명;

```
int? num = null;

if (num.HasValue)
{
    Console.WriteLine("num = " + num.Value);
}
else
{
    System.Console.WriteLine("num = Null");
}
```


배열

- 배열

type[] 변수명;

```
int[] array1 = new int[5];  
int[] array2 = new int[] { 1, 3, 5, 7, 9 };  
int[] array3 = { 1, 2, 3, 4, 5, 6 };  
int[,] multiDimensionalArray1 = new int[2, 3];  
int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };  
int[][] jaggedArray = new int[6][];  
jaggedArray[0] = new int[4] { 1, 2, 3, 4 };
```

분기

- If, else, elseif

```
bool condition = true;
if (condition)
{
    Console.WriteLine("The variable is set to true.");
}
else
{
    Console.WriteLine("The variable is set to false.");
}
```

분기

- switch

```
int caseSwitch = 1;
switch (caseSwitch)
{
    case 1:
        Console.WriteLine("Case 1");
        break;
    case 2:
        Console.WriteLine("Case 2");
        break;
    default:
        Console.WriteLine("Default case");
        break;
}
```

반복

- while
- do while
- For
- foreach

```
int[] a = { 1, 2 };  
  
foreach (int value in a)  
{  
    Console.WriteLine(value);  
}
```

C#

```
int a[2] = { 1, 2 };  
for each(int value in a)  
{  
    std::cout << value;  
}
```

C++

점프

- break
- continue
- goto

C# Study

- C# 프로그래밍 가이드 참고(<https://msdn.microsoft.com/ko-kr/library/67ef8sbd.aspx>)

The screenshot shows the Microsoft Developer Network (MSDN) website. The top navigation bar includes the Microsoft logo, "Developer Network", and links for "기술" (Technology), "다운로드" (Downloads), "프로그램" (Programs), "커뮤니티" (Community), "설명서 작성" (Writing Documentation), and "샘플" (Samples). On the right, there are links for "로그인" (Login), "MSDN Subscription", and "도구" (Tools). A "모두 축소" (Collapse All) button is also present.

The left sidebar contains a navigation menu with the following items:

- MSDN Library
- 개발 도구 및 언어
- Visual Studio 2013 시작
- Visual Basic 및 Visual C#
- Visual C#
- C# 프로그래밍 가이드
 - C# 프로그램 내부
 - 배열(C# 프로그래밍 가이드)
 - 클래스 및 구조체(C# 프로그래밍 가이드)
 - 대리자(C# 프로그래밍 가이드)
 - 열거형 형식(C# 프로그래밍 가이드)
 - 이벤트(C# 프로그래밍 가이드)
 - 예외 및 예외 처리(C# 프로그래밍 가이드)

The main content area displays the "C# 프로그래밍 가이드" (C# Programming Guide) for "Visual Studio 2013". It includes a "다른 버전" (Other Versions) link. The text states: "이 단원에서는 C# 언어의 주요 기능 및 C#에서 .NET Framework를 통해 액세스할 수 있는 기능에 대한 자세한 정보를 제공합니다." (This unit provides detailed information about the major features of the C# language and the features accessible through the .NET Framework in C#). It also mentions: "이 단원에서는 독자가 C# 및 일반적인 프로그래밍 개념에 대해 잘 알고 있다고 가정합니다. 프로그래밍이나 C#에 처음 입문하는 사용자는 C# 개발자 센터에 도움이 되는 많은 자습서, 샘플 및 비디오를 찾을 수 있습니다." (This unit assumes that the reader is familiar with C# and general programming concepts. Users new to programming or C# can find many helpful tutorials, samples, and videos in the C# Developer Center). A note at the bottom says: "특정 키워드, 연산자 및 전처리기 지시문에 대한 자세한 내용은 C# 참조를 참조하십시오. C# 언어 사양에 대한 자세한 내용은 C# 언어 사양을 참조하십시오." (For more information about specific keywords, operators, and preprocessor directives, see the C# Reference. For more information about the C# language specification, see the C# Language Specification).

Below the text, there is a section titled "언어 단원" (Language Units) with a list of links: "C# 프로그램 내부", "배열", "클래스 및 구조체", "대리자", "열거형 형식", "이벤트", and "예외 및 예외 처리".