

[과목] C#

[챕터] Chapter 2. 처음 만드는 C# 프로그램

[소제] 2.2 첫 번째 프로그램 뜯어보기

[내용]

```
using System;
using static System.Console;

namespace Hello
{
    class MainApp{
        static void Main(string[] args){
            if(args.Length == 0){
                WriteLine("사용법 : Hello.exe<이름>");
                return;
            }
            WriteLine("Hello, {0}!", args[0]);
        }
    }
}
```

### 2.2.1 using System;

using 은 C#의 키워드 중 하나. 키워드는 C# 언어의 규격에 미리 정의되어 있는 특별한 단어이다.

System 은 C# 코드가 기본적으로 필요로 하는 클래스를 담고 있는 네임스페이스이다.

using System;은 System 네임스페이스 안에 있는 클래스를 사용하겠다고 컴파일러에 알리는 역할이다.

### 2.2.2 using static System.Console;

using 키워드만 사용하면 네임스페이스 전체를 사용한다는 의미지만 using static 은 어떤 데이터 형식의 정적 멤버를 데이터 형식의 이름을 명시하지 않고 참조하겠다고 선언하는 기능이다.

### 2.2.3 namespace Hello{ }

네임스페이스는 성격이나 하는 일이 비슷한 클래스, 구조체, 인터페이스, 대리자, 열거 형식 등을 하나의 이름 아래 묶는 일을 한다.

네임스페이스를 만들 때는 다음과 같이 `namespace` 키워드를 이용하며, 네임스페이스\_이름 뒤에 따라오는 괄호 {와} 사이에는 이 네임스페이스에 소속되는 클래스 등이 들어간다.

```
namespace 네임스페이스_이름
{
    //클래스
    //구조체
    //인터페이스 등...
}
```

### 2.2.4 class MainApp{ }

`class MainApp` 구문은 `MainApp` 이라는 이름의 클래스를 만든다.

클래스는 C# 프로그램(C# 언어로 만드는 프로그램)을 구성하는 기본 단위로서 데이터와 데이터를 처리하는 기능(메소드 `Method`)으로 이루어진다.

### 2.2.5 static void Main(string[] args) { }

`static void Main(string[] args){ }`은 메소드이다. 프로그램의 진입점으로서 프로그램을 시작하면 실행되고, 이 메소드가 종료되면 프로그램도 종료된다.

모든 프로그램은 반드시 `Main` 이라는 이름을 가진 메소드를 하나 가지고 있어야 한다.

`static`은 한정자로서 메소드나 변수 등을 수식한다. C# 프로그램의 각 요소는 코드가 실행되는 시점에 비로소 메모리에 할당되는 반면, `static` 키워드로 수식되는 코드는 프로그램이 처음 구동될 때부터 진작에 메모리에 할당된다.

프로그램이 실행되면 CLR(Common Language Runtime)은 프로그램을 메모리에 올린 후 프로그램의 진입점을 찾는데, 이때 `Main()` 메소드가 `static` 키워드로 수식되어 있지

않다면 CLR 은 진입점을 찾지 못했다는 에러 메시지를 남기고 프로그램을 종료시킬 것이다.

void 는 메소드의 반환 형식이다. 메소드가 어떤 결과도 돌려주지 않을 것이다.

Main 은 메소드의 이름이며, 메소드 이름 뒤에 있는 괄호와 그 사이에 있는 코드는 메소드에 입력되는 매개변수이다.

#### 2.2.6 if(args.Length == 0){}

if 문은 조건을 평가해서 프로그램의 흐름을 결정하는 코드인데, 매개변수 목록의 길이가 0(args.Length == 0)일 때 안내문을 출력하고 프로그램을 종료한다.