동국대학교
dongguk university

클래스 기본

HyoJoon Han
동국대학교
han6343@dongguk.edu

목차

클래스 개요

- 사용자 정의 자료형
  - 클래스 : 객체 지향 언어, 원하는 새로운 자료형 정의
  - 예

```
class Car
{
    int carNumber;
    int inTime;
    int outTime;
}

static void Main(string[] args)
{
    Car[] cars = new Car[10];
}
```

클래스 개요

- 메소드 사용한 코드

```
class Car
{
    int carNumber;
    DateTime inTime;
    DateTime outTime;

    public void SetInTime()
    {
        this.inTime = DateTime.Now;
    }

    public void SetOutTime()
    {
        this.outTime = DateTime.Now;
    }
}
```

```
static void Main(string[] args)
{
    Car car = new Car();
    car.SetInTime();
    car.SetOutTime();
}
```

클래스 개요

- 클래스와 인스턴트

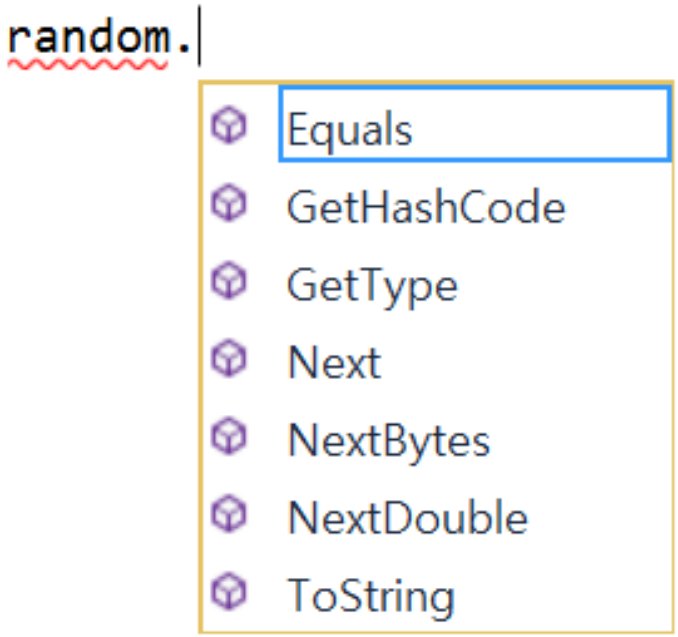
```
Car car = new Car()
```

클래스 인스턴스 new 키워드 생성자

  - 클래스 : 사용자 정의 자료형
  - 인스턴스(객체) : 클래스 자료형을 변수로 선언한 것
  - 생성자 : 클래스 이름과 같은 메소드(클래스 이름 뒤에 괄호가 붙은 것)
  - 클래스 이름 대문자로 시작(관례)

클래스 사용

- Random 클래스
  - 임의의 숫자 생성시 사용
  - 인스턴스 생성 방법 : Random random = new Random()



클래스 사용

- List 클래스
  - 배열과 유사
  - 제네릭(Generic) : 클래스 선언 시 어떤 자료형인지 알려주는 것
  - 예

코드 5-5 List 클래스의 인스턴스 생성 /5장/ClassUses

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    List<int> list = new List<int>();
}
```

클래스 사용

List 클래스 참조 추가

```
// 변수를 선언합니다.
List<int> list = new List<int>();
```

'List' 형식 또는 네임스페이스 이름을 찾을 수 없습니다. using 지시문 또는 어셈블리 참조가 있는지 확인하십시오.

마우스 가져다 대기

```
// 변수를 선언합니다.
List<int> list = new List<int>();
```

```
// 변수를 선언합니다.
List<int> list = new List<int>();
```

using System.Collections.Generic;

System.Collections.Generic.List

'List'에 대한 클래스 생성(C)

새 형식 생성(T)...

Ctrl + . 단축키를 누르거나
[그림 5-10] 파란 글상자 선택

클래스 사용

- List 인스턴스 생성과 동시에 요소 추가

코드 5-7 List 인스턴스 생성과 동시에 요소 추가

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    List<int> list = new List<int>() { 52, 273, 32, 64 };

    // 반복을 수행합니다.
    foreach (var item in list)
    {
        Console.WriteLine("Count: " + list.Count + "\titem: " + item);
    }
}
```

클래스 사용

- Math 클래스
  - 수학과 관련된 변수 또는 메소드 제공
  - 인스턴스를 만들지 않고 사용

```
Math.
```

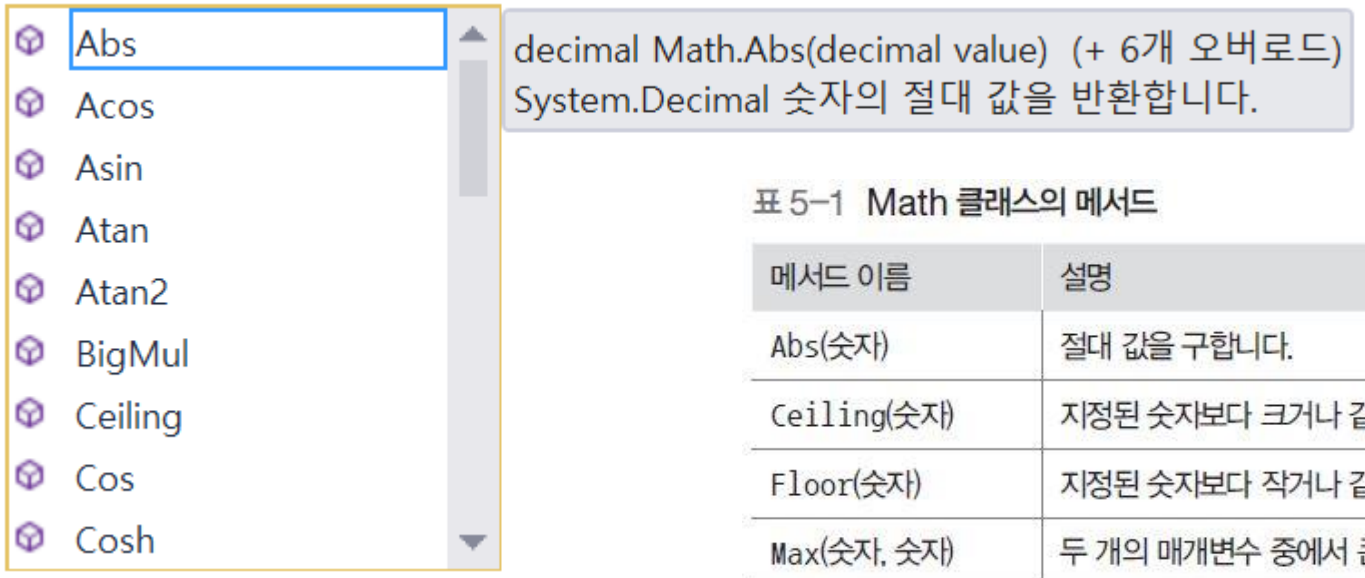


표 5-1 Math 클래스의 메서드

메서드 이름	설명
Abs(숫자)	절대 값을 구합니다.
Ceiling(숫자)	지정된 숫자보다 크거나 같은 최소 정수를 구합니다.
Floor(숫자)	지정된 숫자보다 작거나 같은 최대 정수를 구합니다.
Max(숫자, 숫자)	두 개의 매개변수 중에서 큰 값을 구합니다.
Min(숫자, 숫자)	두 개의 매개변수 중에 작은 값을 구합니다.
Round(숫자)	반올림합니다.

클래스 생성

- 클래스 생성 예

```
class [클래스 이름]
{

}
```

클래스 생성

- 하나의 파일에 여러 개의 클래스 생성
  - 클래스를 생성 하기 가장 쉬운 방법 : 파일 하나에 여러 개의 클래스를 생성하는 것

```
using System;

namespace ClassBasic
{
    class FirstClass
    {
    }

    class SecondClass
    {
    }

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

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

FieldAccessException
FileStyleUriParser
finally
FirstClass
fixed
MissingFieldException
NotFiniteNumberException
RuntimeFieldHandle

클래스 생성

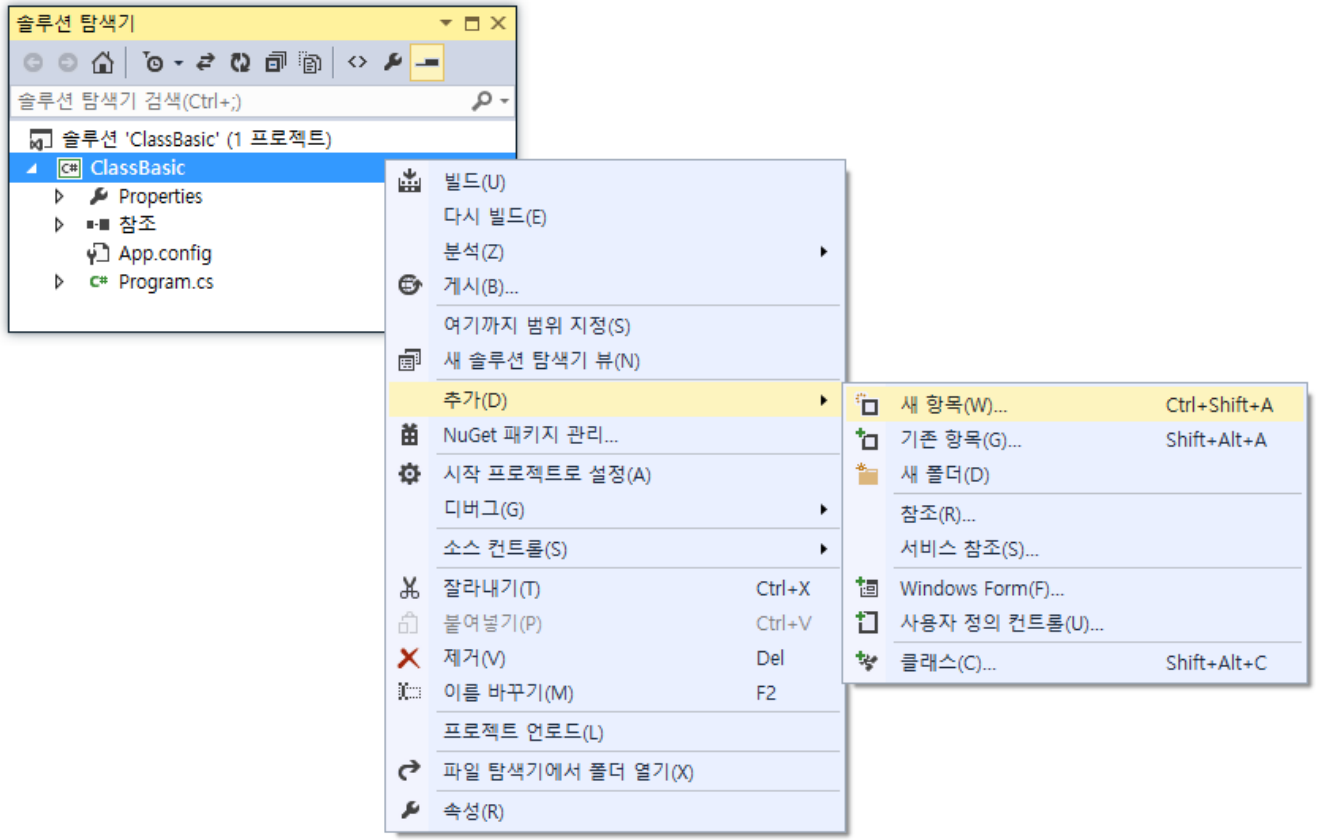
- 클래스 내부에 클래스 생성

```
class Program
{
    class FirstClass
    {
    }

    class SecondClass
    {
    }

    static void Main(string[] args)
    {
    }
}
```

클래스 생성

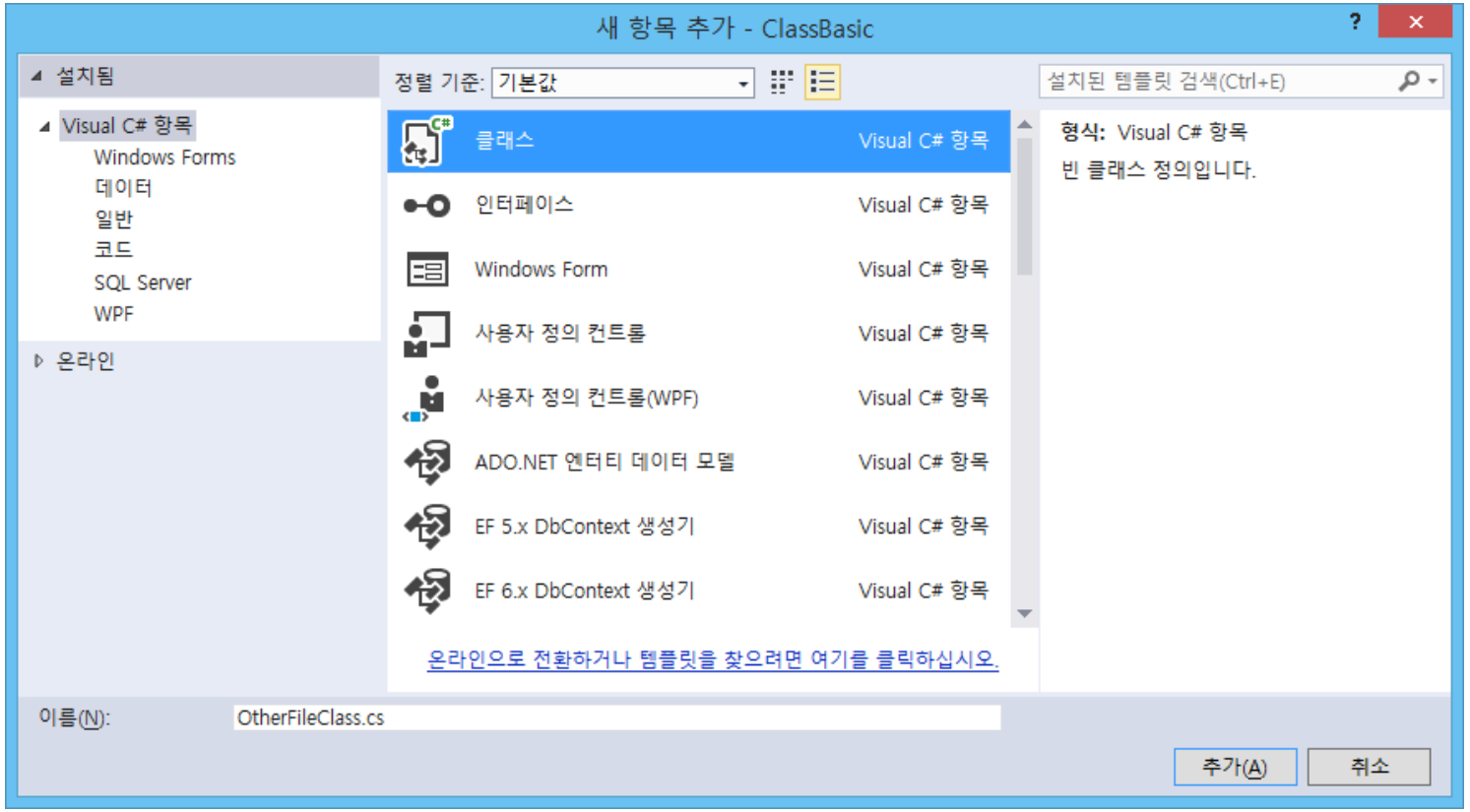
- 서로 다른 파일에 클래스 생성
  - 파일 하나에 클래스 하나를 넣고, 파일의 이름과 맞추어 만드는 것이 일반적
  - 파일의 이름과 클래스 이름이 달라도 상관없음

① 마우스 오른쪽 클릭 [추가] – [새 항목] 또는 [클래스] 클릭



클래스 생성

② 새 파일 대화상자 실행, 클래스 이름 입력, [추가] 버튼 클릭



클래스 생성

- 클래스 이름 충돌

코드 5-10 클래스 이름 충돌 /5장/NameCollision

```
class Program
{
    class Math { }

    static void Main(string[] args)
    {
        Console.WriteLine(Math.Abs(-10));
    }
}
```

```
class Program
{
    class Math { }

    static void Main(string[] args)
    {
        Console.WriteLine(Math.Abs(-10));
    }
}
```

'ITCookbook.Program.Math'에 'Abs'에 대한 정의가 없습니다.

  - 클래스 이름 지정 시 특별한 목적이 없는 한 기존 클래스 이름과 다르게 선언