

컴퓨터정보과 C# 프로그래밍

3주차 클래스와 연산자/수식 (메소드)
- 메소드, 변수 추가 자료

메소드 기본

method

- method : 프로그램이 실질적으로 실행하는 구역
 - 정의 : 생성
 - 호출 : 사용
- 기본 형태

```
[접근제한자] [반환형] 메소드이름 ([매개변수1, ... , 매개변수n])
{
    [실행 코드]
}
```

method의 다양한 정의 형태 (1)

```
void Test()  
{  
    return;  
}
```

정의

```
Test();  
Test(3);  
Test(3,2);  
int a = Test();  
int a = Test(3,2);
```

호출

반환타입 : void
매개 변수 : 없음

method의 다양한 정의 형태 (2)

```
int Test()  
{  
    return 3;  
}
```

정의

```
Test();  
Test(3);  
Test(3,2);  
int a = Test();  
int a = Test(3,2);
```

호출

반환타입 : int
매개 변수 : 없음

method의 다양한 정의 형태 (3)

```
void Test(int a)
{
    Console.WriteLine(a * 2);
    return;
}
```

정의

```
Test();  
Test(3);  
Test(3,2);  
int a = Test();  
int a = Test(3,2);
```

호출

반환타입 : void
매개 변수 : int

method의 다양한 정의 형태 (4)

```
int Test(int a)
{
    return a * 2;
}
```

정의

```
Test();  
Test(3);  
Test(3,2);  
int a = Test();  
int a = Test(3,2);
```

호출

반환타입 : int
매개 변수 : int

method의 다양한 정의 형태 (5)

```
int Test(int a, int b)
{
    return a * b;
}
```

정의

```
Test();  
Test(3);  
Test(3,2);  
int a = Test();  
int a = Test(3,2);
```

호출

반환타입 : int
매개 변수 : int, int

변수 심화

객체 생성 (값 생성)

- `int test1 = 1;` //리터럴 1
- `string test2 = " 1 " ;` //리터럴 " 1 "
- `Score s = ?; //리터럴 ?`
 `Score s = new Score();`
 new 명령어
 `Score()` 생성자

변수

- 변수는 선언 위치에 따라서 의미와 생존시간(생성 시점과 종료 시점)이 다르다
 - class/struct 안에 선언한 변수
 - 필드 (field)
 - 인스턴스 필드 (Instance field)
 - 정적 필드 (static field) : static
 - 상수 필드 (constant field) : const
 - 읽기 전용 필드 (readonly field) : readonly
 - ...
 - method 안에 선언한 변수
 - 지역변수 (local variable)
 - 매개변수 (parameter variable)

Employee (1)

```
class Score
```

```
{  
    public static double MidtermRate = 0.35;  
    public static double FinalRate = 0.35;  
    public static double AttendRate = 0.15;  
    public static double HomeworkRate = 0.15;  
  
    public double Midterm;  
    public double Final;  
    public double Attend;  
    public double Homework;  
  
    public double Sum()  
    {  
        var summary = Midterm + Final + Attend + Homework;  
        return summary;  
    }  
}
```

정적 필드 - 클래스 당 하나만 생성하는 변수

(프로그램 실행시 생성, 프로그램 종료시 사라지는 변수)

외부에서 접근하고자 할 때, 인스턴스가 아닌 클래스로 접근

인스턴스 필드 - 클래스의 인스턴스 별로 생성하는 변수

(인스턴스 생성 시 생성되며, 인스턴스 소멸시 사라지는 변수)

외부에서 접근하고자 할 때, 인스턴스로 접근

지역변수 - 메소드 실행시에만 생성하고, 메소드 실행이 종료되면 사라지는 변수

외부에서 접근할 방법 없음

예외) ref, out 사용시 ...

Employee (2)

```
class Score
```

```
{  
    public const double MidtermRate = 0.35;  
    public const double FinalRate = 0.35;  
    public const double AttendRate = 0.15;  
    public const double HomeworkRate = 0.15;  
  
    public double Midterm;  
    public double Final;  
    public double Attend;  
    public double Homework;  
  
    public double Sum()  
    {  
        var summary = Midterm + Final + Attend + Homework;  
        return summary;  
    }  
}
```

상수 필드 (+정적 필드의 의미도 같이 있음)

추후. 읽기전용 필드와의 관계와 비교해야 함.

Program

```
class Program
{
    static void Main(string[] args)
    {
        var rate1 = $"중간:{Score.MidtermRate * 100}%";
        var rate2 = $"기말:{Score.FinalRate * 100}%";
        var rate3 = $"출석:{Score.AttendRate * 100}%";
        var rate4 = $"과제:{Score.HomeworkRate * 100}%";

        Console.WriteLine(rate1);
        Console.WriteLine(rate2);
        Console.WriteLine(rate3);
        Console.WriteLine(rate4);

        Score score1 = new Score();
        Score score2 = new Score();
```

```
        Score score1 = new Score();
        Score score2 = new Score();

        score1.Midterm = 100;
        score1.Final = 90;
        score1.Attend = 90;
        score1.Homework = 100;
        Console.WriteLine(score1.Sum());

        score2.Midterm = 90;
        score2.Final = 80;
        score2.Attend = 80;
        score2.Homework = 100;
        Console.WriteLine(score2.Sum());
    }
}
```