Class

1. class는 **"객체를 생성하기 위한 틀"**이다.
2. **class** = **속성**(field, 전역변수) + **기능**(method, 함수)
3. class 형식

```
접근제어자 class ClassName {

   - 변수선언(전역변수(instance변수), class변수);

   - method영역

     * Constructor(생성자)
     * method()

}
```

   - **variable**

```
형식) 접근제어자 DataType 변수명;

예) public int age;
    private boolean flag;
    char c;
    String name;(Object..)
```

   - **method** : class의 기능(일처리, logic).

```
형식)
접근제어자 RetrunType methodName([args]) {
    기능 구현; ---> 변수(지역변수), 제어문, 반복문, ....
    [return resultValue;]
}

-- ReturnType : return 하는 결과값의 type (int, boolean,..., 참조형(class, 배열))
                return할 값이 없을 경우 : void
```

   - **생성자(constructor)**

     1. 클래스의 이름과 똑같은 method의 일종.
     2. 객체 생성시 **최초로 호출**됨.
     3. 생성자는 **객체의 초기화**작업. (전역변수 값설정..)

```
형식) ReturnType이 없다. (void XXXX)
접근제어자 ClassName([args]) {
    객체의 초기화 작업.
}
```

4. method **overloading** : 똑같은 이름의 메소드를 여러개 정의 하는것.
   단, 인자값의 갯수나 인자값의 type이 달라야 한다.

```
public void test() {

}

int test() { // error : return type이나 접근제어자는 상관 X
  return 1;
}

void test(int x) {

}

void test(int y) { // error : 매개변수의 이름과 상관 X

}

void test(int x, String s) {

}

void test(String s, int x) {

}
```

5. 사용자 정의 **생성자가 없을 경우 default 생성자가 생략**되어 있다.

```
public ClassName() {}
```

   단, 사용자가 생성자를 하나이상 만들경우 default생성자는 사라진다.

6. **this** : 자기 클래스 참조변수
   super : 상위 클래스 참조변수

   //아래의 2가지는 생성자의 첫줄에서만 호출 가능.
   this() : 자기 클래스 생성자 호출
   super() : 상위 클래스 생성자 호출

Class(Reference Data Type)의 사용

1. **선언**

```
접근제어자 ClassName 객체명;
```

```
ex)
public String name;
Car car1;
```

2. **생성** : class를 memory에 올리는 작업.

```
객체명 = new 생성자([args]);
```

```
ex)
name = new String("안효인");
car1 = new Car();
```

```
1, 2 동시 : 접근제어자 ClassName 객체명 = new 생성자();
Car car = new Car();
```

3. **사용**

```
객체명.변수;
객체명.변수 = 값;
객체명.method();
```

```
ex)
System.out.println("차이름 : " + car1.carName);
car1.speed = 100;
car1.stop();
```

```
위 개념을 이용하여 클래스를 만들어보자.
자동차를 주제로 클래스를 만들고자한다.
자동차의 속성과 기능을 생각해보고 클래스를 작성해 본다.
속성의 type을 고민해 보고 method의 return type과 arguments를 생각해보자.
```

이름	자동차	Car
속성	차이름, 색상, 제조사, 속도	carName, carColor, maker, speed
기능	가속, 감속, 정지	speedUp(), speedDown(), stop()