

# 02-06 [Java]

 소유자	 종수 김
 태그	

## Class와 Data.

Java는 Class, 객체로 이루어져 있다.

학생을 표현하기 위해 필요한 변수가 name, age, grade(Data) 라면 객체가 없을 때 학생을 표현하기 위해서는 name1, age1, grade1, name2, age2, grade2 .... 로 변수를 늘려 나갈 수 밖에 없으며 '배열'을 이용한다고 하더라도 같은 학생을 정렬하기 위한 문제등이 생긴다.

즉, 학생을 표현하는 '데이터'는 있지만 '학생'이라는 개념은 없는 것.

이를 위해 Java에서는 현실세계를 더 자세하게 표현하기 위한 Class라는 개념이 존재한다.

## Class 도입.

### ▼ Class

클래스에 정의한 변수들을 멤버 변수, 또는 필드라고 한다.

멤버 변수 = 필드

```
public class Student {  
    String name;  
    int age;  
    int grade;  
}
```

- 클래스는 관례상 대문자로 시작하며, 낙타 표기법 사용.

클래스와 사용자정의 타입

- 타입은 데이터의 종류나 형태를 나타낸다.
- int라고 하면 정수 타입, String이라면 문자열 타입.

⇒

학생이라는 형태를 나타내기 위한 타입을 직접 만들자.

- 사용자가 직접 만드는 '사용자 정의 타입'을 만드려면 '설계도'가 필요하다.  
⇒ 이 설계도가 바로 Class
- 설계도인 클래스를 사용해서 '실제 메모리에 만들어진 실체를 [객체] or [인스턴스]라고 한다.'

\* 즉, 해당 실체(학생, 자동차, etc...)를 위한 멤버 변수 및 함수를 정의한 것이 클래스.  
이를 실체화(new를 통한 메모리[힙] 할당)한 것이 객체 or 인스턴스.

- 클래스를 통해 사용자가 원하는 종류의 데이터 타입을 정의할 수 있다.
- 이렇게 만들어진 객체(인스턴스)는 메모리에 할당되며, 이 객체에 접근할 수 있는 메모리 주소(참조 값)을 반환한다.

해당 참조값(레퍼런스)으로 생성된 인스턴스에 접근할 수 있는 것.

참조값을 변수에 보관해야 하는 이유

객체를 생성하는 new Student()코드 자체에는 아무런 이름이 없음.

이 코드는 단순히 Student 클래스를 기반으로 메모리에 실제 객체를 만드는 것.

이런 이유로 객체를 생성할 때 반환되는 참조값을 어딘가에 보관해두어야 하는 것.

앞서 Student student1 변수에 참조값을 저장해두었으므로 저장한 참조값을 통해 실제 메모리에 존재하는 객체에 접근하는 것.

## Class(클래스), Object(객체), Instance(인스턴스)

- Class(클래스)
  - 객체를 생성하기 위한 '틀', '설계도'
    - \* 자동차 설계도는 자동차가 아님.
    - 설계도는 실제 존재하는 것이 아닌 개념으로만 있는 것
  - 객체가 가져야 할 속성(변수), 기능(메서드)를 정의
- Object(객체)
  - 클래스에서 정의한 속성과 기능을 가진 실체
    - \* new 키워드를 통해 생성(힙 메모리에 할당)된 Class
  - 객체는 서로 독립적인 상태를 가짐.  
student1은 학생1의 속성을 가지는 객체이고, student2는 학생2의 속성을 가지는 객체.  
같은 Student Class에서 만들어졌지만 서로 다른 객체
- Instance(인스턴스)

- 특정 클래스로부터 생성된 객체
- 객체 = 인스턴스
- 주로 객체가 어떤 클래스에 속해 있는지 강조할 때 사용  
student1은 Student Class의 Instance.
- Object vs Instance
  - 둘 다 클래스에서 나온 실체라는 의미에서 비슷하게 사용됨.
  - But, 용어상 Instance는 객체보다 좀 더 관계에 초점을 맞춘 단어.  
모든 Instance는 Object이지만, Student는 Class로부터 생성된다는 점을 명확히.
  - 사실, 따로 구분해서 사용하지 않으나 풍기는 뉘앙스가 조금 다르다는 점.

근본적으로 같다.

## 객체 사용

클래스를 통해 생성한 객체를 사용하려면 먼저 메모리에 존재하는 객체에 접근해야 한다.  
객체에 접근하려면 .(점, dot)을 사용.

객체가 가지고 있는 멤버 변수(name, age, grade)에 값을 대입하려면 먼저 객체에 접근해야 함.

객체의 멤버변수에 접근하기 위한 키워드가 바로 .(점, dot)

student1. ⇒ student1 변수가 가지고 있는 참조값을 통해 실제 객체에 접근하며,  
student1은 참조값을 가지고 있으므로 해당 메모리 주소에 있는 Student객체에 접근.

- 배열 생성 시 Student [] students = new Student[2];  
해당 선언은 말 그대로, 변수 공간을 '생성만 하는 것'으로, 참조하려하면 NPE가 발생.
- 배열 생성 시 생성된 변수 공간을 student1 = new Student(); 라는 new 키워드를 통해 생성된 변수 공간에 참조값에 Student객체를 할당.
- **\*\* Java에서 대입은 항상 변수에 들어 있는 값을 복사. [매우 중요]**  
변수에는 인스턴스 자체가 들어있는 것이 아님. 인스턴스의 위치를 가리키는 참조값 만 이 들어있는 것.  
즉, =을 통해 대입하는 것은 인스턴스 자체를 복사하는 것이 아닌, 인스턴스의 위치를 가리키는 참조값만 복사된다.
-