02-17 [Java]

소유자	종수 김
태그	

기본형과 참조형

변수의 데이터 타입은 크게

1. 기본형
    a. primitve type : int, long, double, boolean과 같은 사용 값을 직접 넣는 기본형
    b. 초기값이 존재
2. 참조형
    a. Student, LocalDate, 배열과 같은 데이터에 접근하기 위한 참조(주소)를 저장하는 데이터 타입.
    b. 초기값이 존재하지 않음. null

기본형 vs 참조형 : 계산

- 기본형은 들어있는 값을 그대로 계산에 사용 가능.
- 참조형은 참조값을 그대로 사용할 수 없으므로 주소에 접근하여 실체에서 계산을 진행해야함.

기본형을 제외한 나머지는 모두 참조형 / 소문자로 시작
기본형은 개발자가 새로 정의하는 것이 불가능.

- String은 클래스. 참조형이지만 문자값을 바로 대입할 수 있음.
  이는 자바에서 특별하게 기본형처럼 사용할 수 있게 만들어준것.

기본형 vs 참조형 : 변수대입

- 자바는 항상 변수의 값을 복사해서 대입한다.
  자바에서 변수에 값을 대입하는 것은 변수에 들어 있는 값을 복사해서 대입.
  기본형 : 들어있는 기본 값을 복사해서 대입.
  참조형 : 참조 주소를 복사해서 대입.
- 기본형 : 변수에 들어 있는 실제 사용하는 값을 복사해서 대입

```
int a = 10;
int b = a;
//값만을 복사해서 대입했으므로 서로 완전 다른 것
```

- 참조형 : 변수에 들어있는 참조 값을 복사해서 대입.

```
Student s1 = new Student();
Student s2 = s1;
//실제 사용하는 객체가 아니라 객체의 위치를 가르키는 참조값이 복사.
//참조값이 복사되었으므로 s1,s2어떤 주소에 접근해서 실체를 변경하더라도 둘
//= 같은 곳을 바라보고 있으므로.
```

기본형 vs 참조형 : 메서드 호출

자바는 항상 변수의 값을 복사해서 대입한다.
이는 메서드 호출도 마찬가지.
사용하는 매개변수(파라미터)도 결국은 변수이므로 메서드 호출할 때 매개변수에 값을 전달하는 것도 마찬가지로 값을 복사해서 전달.

기본형 : 메서드로 기본형 데이터를 전달하면, 해당 값이 복사되어 전달.
이 경우 메서드 내부에서 파라미터 값을 변경해도 호출자의 변수 값에는 영향이 없다.

```
public static void main(String[] args) {
        int a = 10;
        System.out.println("메서드 호출 전 a : "+a); // 10
        change(a);
        System.out.println("메서드 호출 후 a : "+a); // 10

    }
    public static void change(int a) {
        a = 20;
    }
```

참조형 : 메서드로 참조형 데이터를 전달하면, 참조값이 복사되어 전달된다.
이 경우 메서드 내부에서 파라미터로 전달된 객체의 멤버 변수를 변경하면, 호출자의 객체도 변경.

```
public static void main(String[] args) {
        Data d = new Data();
        d.value = 10;
        System.out.println("메서드 호출 전 : "+d.value); // 10
        change(d);
        System.out.println("메서드 호출 후 : "+d.value); // 20
    }
    public static void change(Data d){
        d.value = 20;
    }
```

메서드의 매개변수 또한 파라미터 이므로, 기본형은 값을 / 참조형은 주소를 복사해서 보낸 것.
기본형을 매개변수로 사용하느냐, 참조형을 매개변수로 사용하느냐에 따라 동작이 달라짐.

참조형과 메서드 호출 : 활용

```
public static void main(String[] args) {
        Student student1 = new Student();
        initStudent(student1,"학생1",16,90);
        Student student2 = new Student();
        initStudent(student2,"학생2",17,75);

        printStudent(student1);
        printStudent(student2);
    }

    public static void initStudent(Student student, String na
        student.name = name;
        student.age = age;
        student.grade = grade;
    }
    public static void printStudent(Student student) {
        System.out.println("이름 : "+student.name + "나이 : "+s
    }
```

initStudent : Student객체의 참조 값을 복사해서 전달했으므로, 내부에서 값을 변경한 것이 실 호출자의 데이터 또한 변경할 수 있는 것.

= 생성자.

변수와 초기화

변수의 종류

- 멤버(필드) 변수 : 클래스에 선언
- 지역 변수 : 메서드(메인을 포함하는)에 선언, 매개변수도 지역 변수의 일종.

변수의 초기화

- 멤버 변수 : 자동 초기화.
  - 인스턴스의 멤버 변수는 인스턴스를 생성할 때 자동으로 초기화.
  - 숫자 = 0 / boolean = false / 참조형 = null
- 지역 변수 : 수동 초기화
  - 지역 변수는 항상 직접 초기화해야함.

```
public static void main(String[] args){
    int data;
    System.out.println(data);
    //compile error!
}
```

NULL

참조형 변수에는 항상 객체가 있는 위치를 가리키는 참조 값이 들어간다.

아직 가리키는 대상이 없는 상태.
값이 존재하지 않는다.(자바에서는 참조형에서만 NULL 사용 가능)

- 아무도 참조하지 않는 인스턴스는 GC에 의해 수거.

```
public static void main(String[] args) {
        Data data1 = null;
        System.out.println("data1 = " + data1);
        data1 = new Data(); // 해당 데이터는 GC에 의해 수거.
        System.out.println("data1 = " + data1);
        data1 = null;
```

```
        System.out.println("data1 = " + data1);
    }
```

NullPointerException(NPE)

참조주소가 없는 객체를 사용하려고 하면? = NullPointerException!

객체를 참조할 때는 .(dot)을 사용하는데 참조할 곳이 없는 상태에 .을 찍으면 바로 NPE