

Person.java

```
1 package test;
2
3 public class Person {
4     String name;
5
6     Person(String name) {
7         this.name = name;
8     }
9
10    void setName(String name) {
11        this.name = name;
12    }
13
14    String getName() {
15        return name;
16    }
17 }
18
```

Test1.java

```
1 package test;
2
3 public class Test1 {
4     public static void main(String[] args) {
5         Person seohyun = new Person("서현");
6         change(seohyun);
7         print(seohyun);
8     }
9
10    public static void change(Person person) {
11        person = new Person("김태희"); 케이스 1
12        // person.setName("김태희");
13    }
14
15    public static void print(Person person) {
16        System.out.println(person.getName());
17    }
18 }
19
```

Console

<terminated> Test1 [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\javaw.exe (2019. 6. 4. 오후 12:20:50)

서현

Person.java

```
1 package test;
2
3 public class Person {
4     String name;
5
6     Person(String name) {
7         this.name = name;
8     }
9
10    void setName(String name) {
11        this.name = name;
12    }
13
14    String getName() {
15        return name;
16    }
17 }
18
```

Test1.java

```
1 package test;
2
3 public class Test1 {
4     public static void main(String[] args) {
5         Person seohyun = new Person("서현");
6         change(seohyun);
7         print(seohyun);
8     }
9
10    public static void change(Person person) {
11        // person = new Person("김태희");
12        person.setName("김태희"); 케이스 2
13    }
14
15    public static void print(Person person) {
16        System.out.println(person.getName());
17    }
18 }
19
```

Console

<terminated> Test1 [Java Application] C:\Program Files\Java\jdk1.8.0_201\bin\javaw.exe (2019. 6. 4. 오후 12:22:56)

김태희

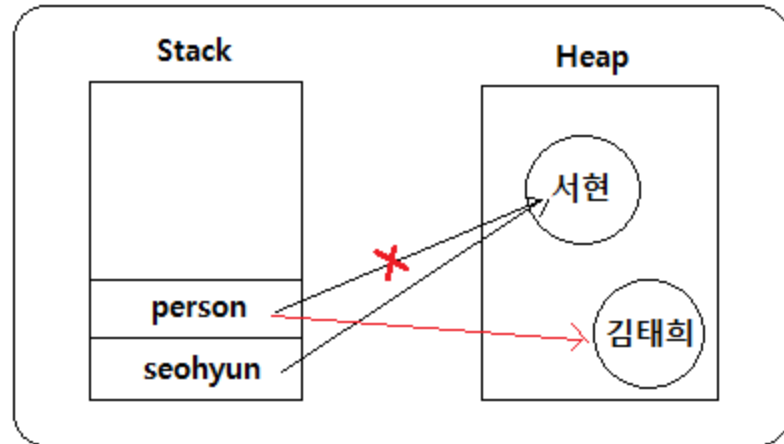
케이스 1이 자바가 Call by value인 이유라고 OKKY에 올라왔던 코드다.
자바가 Call by value인 점은 이슈라고 하나 이후에 더 공부하도록 하고,
여기선 케이스 1과 케이스 2의 결과가 왜 다른지에 대해 나름대로 기록해보고자 한다.

처음에 케이스 1의 결과가 김태희로 출력이 되지 않은 걸 보고 의문이 들었다.
매개변수로 참조형 변수를 넣었는데 왜 적용이 안 되지?
변수의 Scope 문제인가?

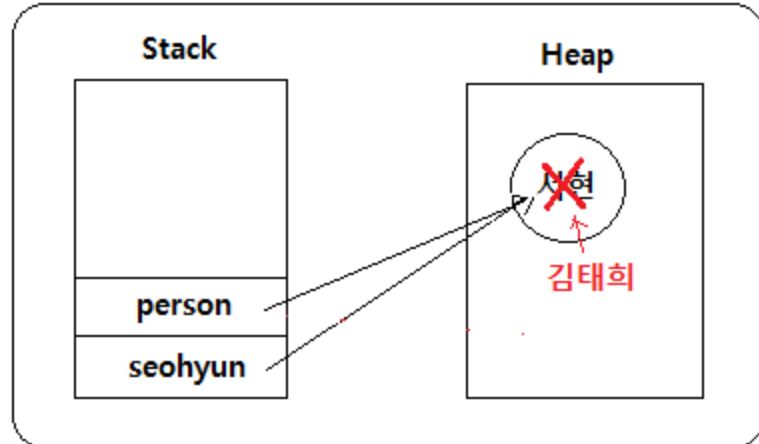
먼저, Scope 문제는 아니었다.
그렇게 치면 흔히 Call by Reference와 Call by value를 비교할 때 예로 드는
swap 메소드의 경우도 바뀌면 안 되었다.

두 케이스의 차이는 아래와 같다.

케이스 1



케이스 2



즉, 객체 자체의 필드를 변화시켜주느냐에 그 차이가 있다.