[Java] 문자열 비교하기 == , equals() 의 차이점

자바에서 일반적인 데이터 타입의 비교는 ==이라는 연산자를 사용하여 비교합니다. 하지만 String 문자열의 값을 비교할때에는 ==이 아닌 equals()라는 메소드를 사용하여 비교를 합니다. equals와 == 은 어떤 차이점이 있을까요.

**String 변수 생성시 주소할당**

String변수를 생성할때는 두가지 방법이 있습니다.

**1.**리터럴을 이용한 방식   
**2.** new 연산자를 이용한 방식 

위의 두 가지 방식에는 큰 차이점이 있습니다. 리터럴을 사용하게 되면 string constant pool이라는 영역에 존재하게 되고 new를 통해 String을 생성하면 Heap 영역에 존재하게 됩니다. String을 리터럴로 선언할 경우 내부적으로 String의 intern() 메서드가 호출되게 됩니다. intern() 메서드는 주어진 문자열이 string constant pool에 존재하는지 검색하고 있다면 그 주소값을 반환하고 없다면 string constant pool에 넣고 새로운 주소값을 반환합니다.

**주소값 비교(==)와 값 비교(equals)**

==연산자와 String클래스의 equals()메소드의 가장 큰 차이점은 == 연산자는 비교하고자 하는 두개의 대상의 주소값을 비교하는데 반해 String클래스의 equals 메소드는 비교하고자 하는 두개의 대상의 값 자체를 비교합니다. 일반적인 타입들 int형, char형등은 Call by Value 형태로 기본적으로 대상에 주소값을 가지지 않는 형태로 사용됩니다. 하지만 String은 일반적인 타입이 아니라 클래스입니다. 클래스는 기본적으로 Call by Reference형태로 생성 시 주소값이 부여됩니다. 그렇기에 String타입을 선언했을때는 같은 값을 부여하더라도 서로간의 주소값이 다를 수가 있습니다.

**문자열 비교 (==연산자)**

public class compare {

public static void main(String[] args) {

String s1 = "abcd";

String s2 = new String("abcd");

if(s1 == s2) {

System.out.println("두개의 값이 같습니다.");

}else {

System.out.println("두개의 값이 같지 않습니다.");

}

}

}



== 연산자의 경우 객체의 주소값을 비교하기 때문에 일반 객체처럼 Heap 영역에 생성된 String 객체와 리터럴을 이용해 string constant pool에 저장된 String 객체의 주소값은 다를 수밖에 없습니다. 그러므로 두개의 값은 서로 다르다는 결론이 나오게 됩니다. 이러한 경우가 발생할 수 있기에 자바에서 문자열을 비교하려면 equals라는 메서드를 활용하여 두개의 값을 비교해주어야 합니다.

**문자열 비교 (equals메서드)**

public class compare {

public static void main(String[] args) {

String s1 = "abcd";

String s2 = new String("abcd");

if(s1.equals(s2)) {

System.out.println("두개의 값이 같습니다.");

}else {

System.out.println("두개의 값이 같지 않습니다.");

}

}

}



String 클래스안에 있는 equals라는 메서드를 사용하면 두 비교대상의 주소 값이 아닌 데이터값을 비교하기 때문에 어떻게 String을 생성하느냐에 따라 결과가 달라지지 않고 정확한 비교를 할 수 있습니다.