

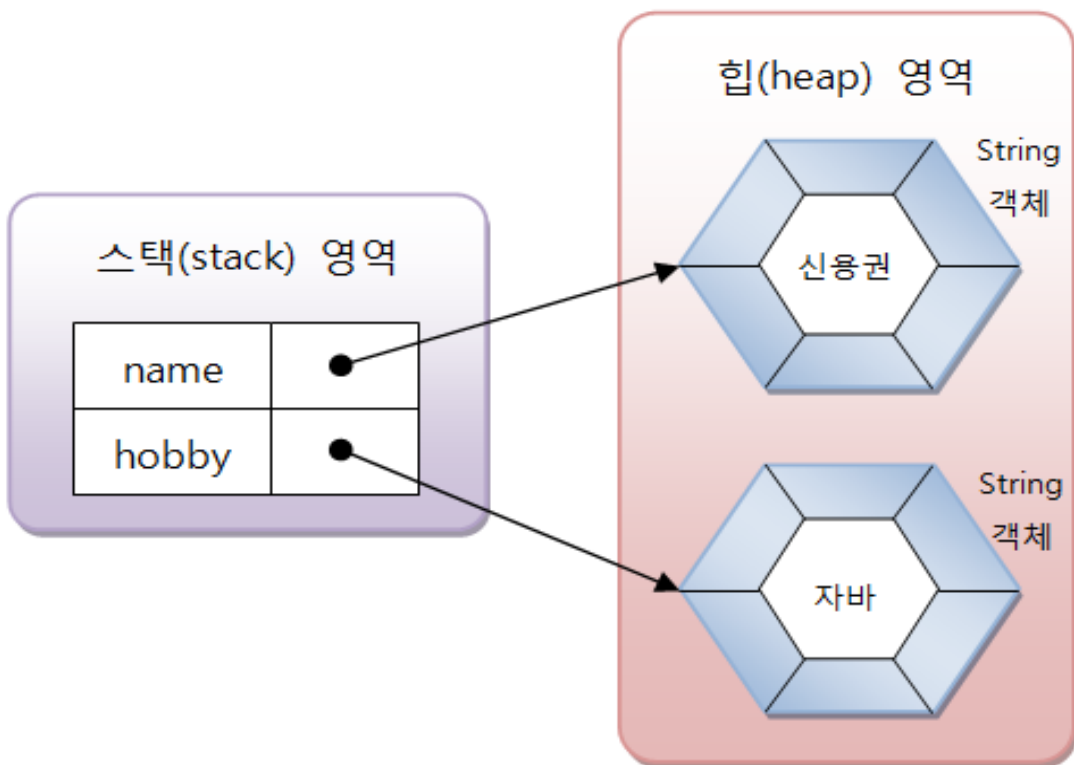
String 타입

String 타입

❖ String 타입

- 문자열을 저장하는 클래스 타입

```
String name;  
name = "신용권";  
String hobby = "자바";
```

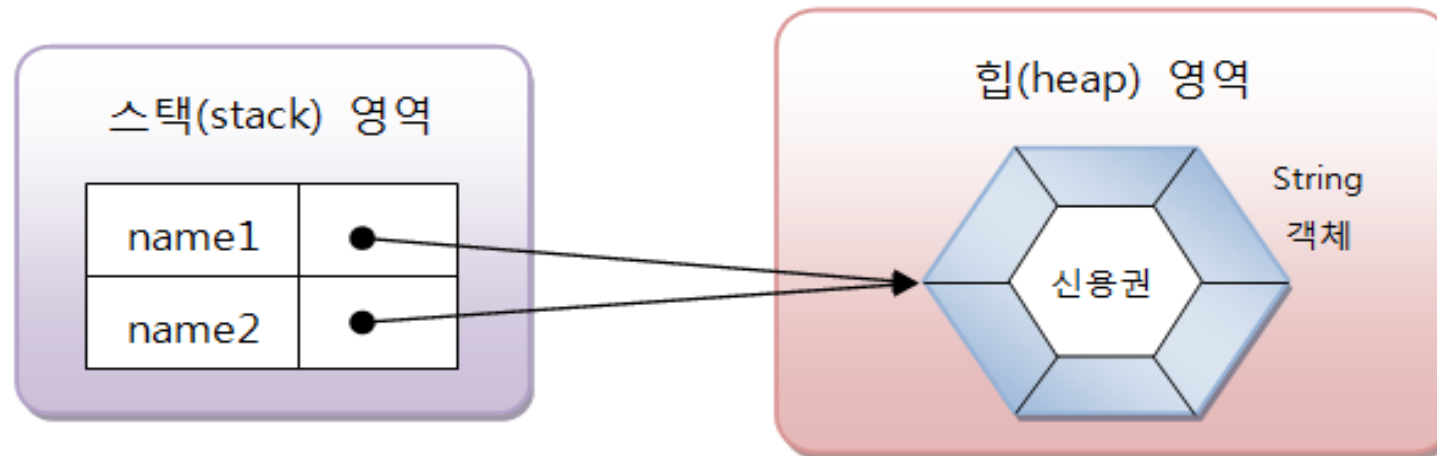


String 타입

❖ String 타입

- 문자열 리터럴 동일하다면 String 객체 공유

```
String name1 = "신용권";  
String name2 = "신용권";
```

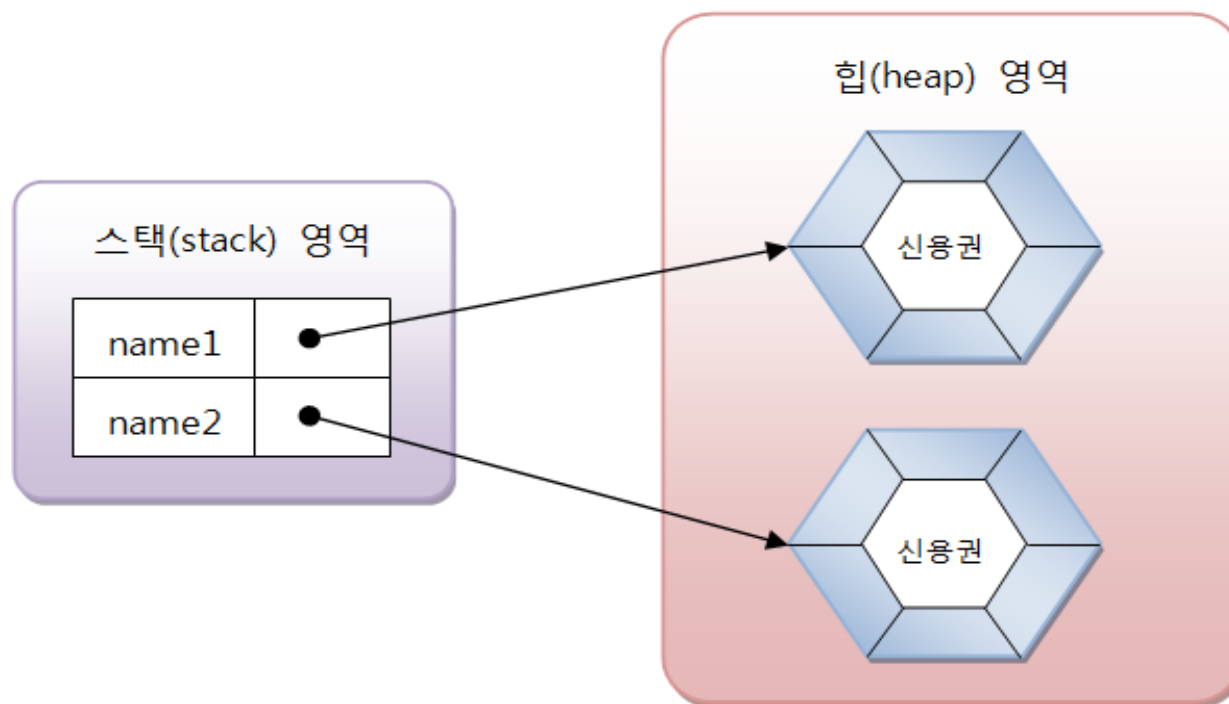


String 타입

❖ String 타입

- new 연산자를 이용한 String 객체 생성
 - 힙 영역에 새로운 String 객체 생성
 - String 객체를 생성한 후 번지 리턴

```
String name1 = new String("신용권");  
String name2 = new String("신용권");
```



String 타입

❖ 문자열 비교: StringEqualsExample.java

```
public class StringEqualsExample {  
    public static void main(String[] args) {  
        String strVar1 = "신민철";  
        String strVar2 = "신민철";  
  
        if (strVar1 == strVar2) {  
            System.out.println("strVar1과 strVar2는 참조가 같음");  
        } else {  
            System.out.println("strVar1과 strVar2는 참조가 다름");  
        }  
  
        if (strVar1.equals(strVar2)) {  
            System.out.println("strVar1과 strVar2는 문자열이 같음");  
        }  
    }  
}
```

String 타입

❖ 문자열 비교: StringEqualsExample.java

```
String strVar3 = new String("신민철");
String strVar4 = new String("신민철");

if (strVar3 == strVar4) {
    System.out.println("strVar3과 strVar4는 참조가 같음");
} else {
    System.out.println("strVar3과 strVar4는 참조가 다름");
}

if (strVar3.equals(strVar4)) {
    System.out.println("strVar3과 strVar4는 문자열이 같음");
}
}
```