



# 2 학기

# JAVA Class

## ‘이것이 자바다’



**2025.09**

**Cover Description**

made by Lewis





# 복습

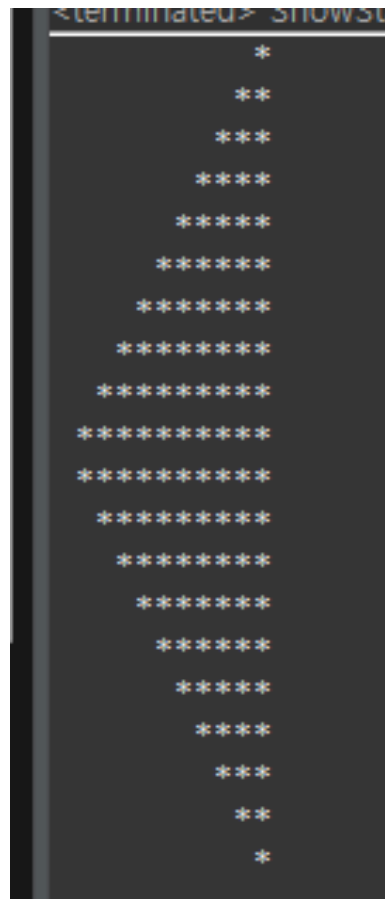
## ■ 복습 1

1. 1 ~ 20 까지의 랜덤 숫자를 발생시키고  
숫자를 입력 받아 맞추는 프로그램을 작성 하시오.
2. 입력한 숫자가 랜덤 숫자보다 작으면 Up  
입력한 숫자가 랜덤 숫자보다 크면 Down  
맞으면 Bingo 표시
3. 몇번만에 맞추는지 표시 하시오.

```
Input a number between 1 and 20.  
18  
Down!  
Input a number between 1 and 20.  
15  
Down!  
Input a number between 1 and 20.  
10  
Bingo!  
You passed the Quize, 3 th try..
```



1. 아래와 같이 표시되도록 코딩 하시오



## 01 데이터 타입

데이터 타입 분류 – 기본/참조 타입

## 02 메모리 영역

JVM 구조

## 03 참조 타입의 비교 연산과 초기화

== , != , null

## 04 문자열 타입

비교, 추출, 길이, 대체, 잘라내기, 찾기, 분리

## 05 배열 타입

생성, 길이, 다차원 배열, 배열 참조

컴퓨터가 이해하는 코드는  
누구라도 작성할 수 있습니다.  
뛰어난 프로그래머는 사람이  
이해하는 코드를 작성합니다.



## ■ 데이터 타입

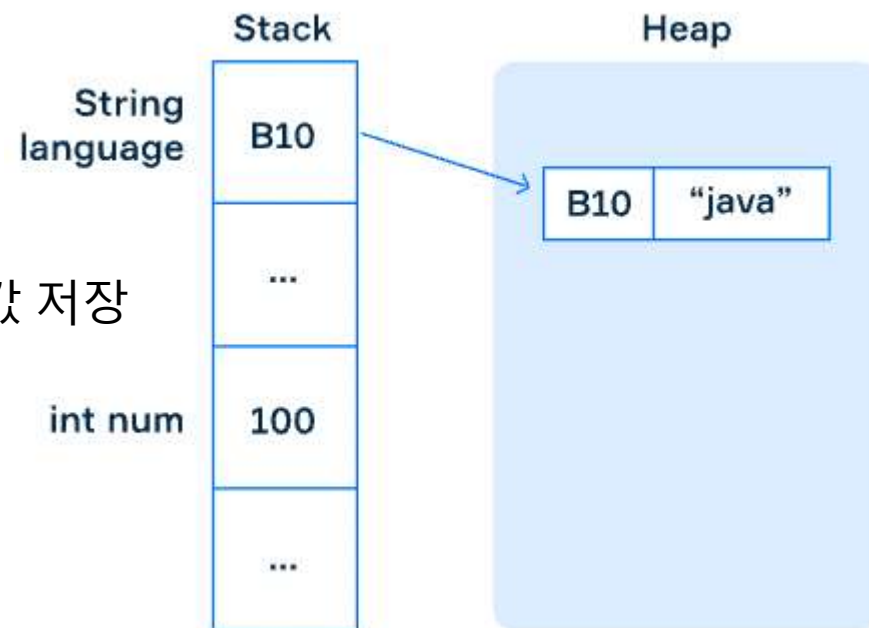
[ 데이터 타입 분류 ]

\* 기본 타입 - 값 자체를 저장

- 정수 타입 : byte, char, short, int, long
- 실수 타입 : float, double
- 논리 타입 : Boolean

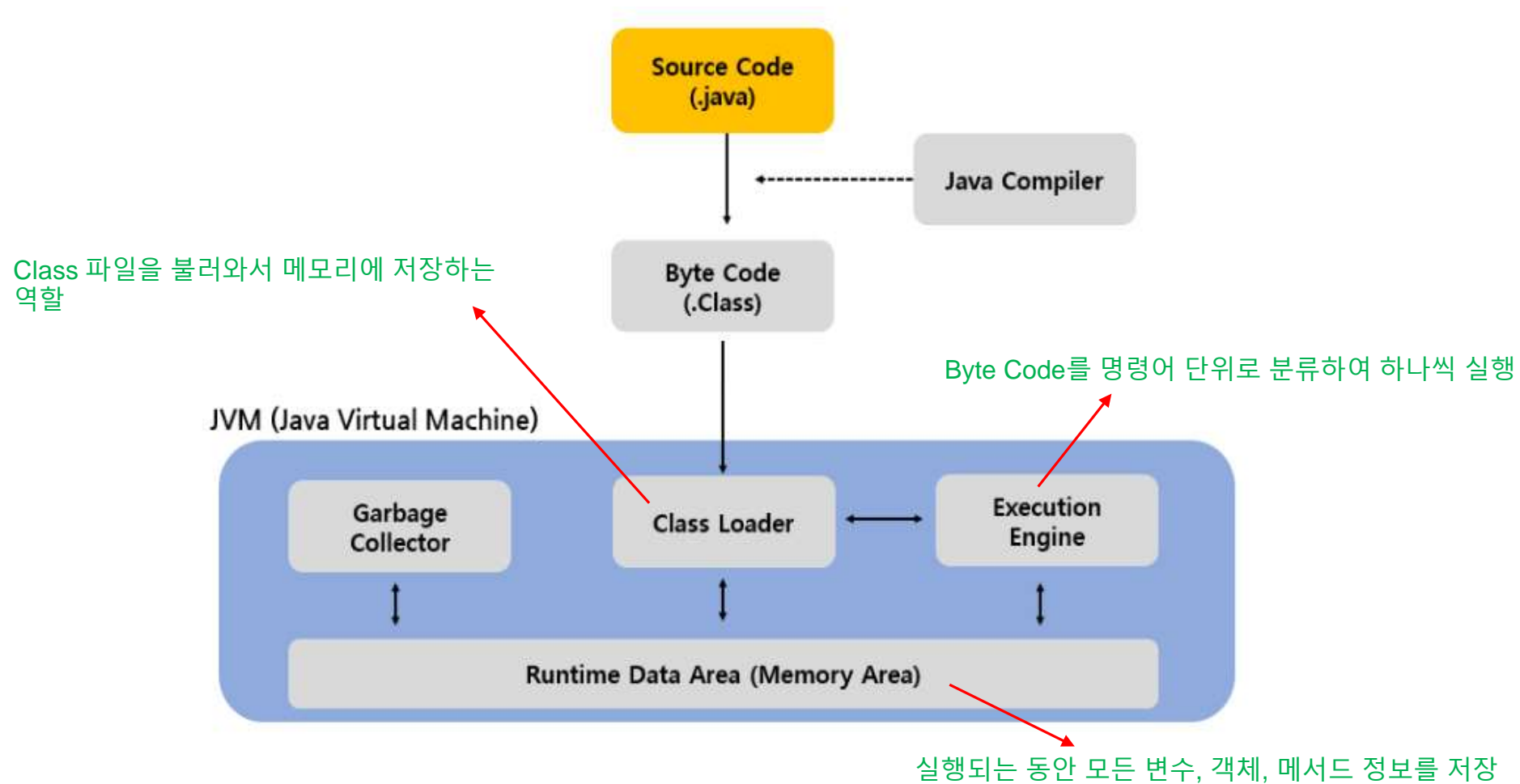
\* 참조 타입 - 변수 객체가 생성되는 메모리 주소 값 저장

- 배열 타입
- 열거 타입
- 클래스
- 인터페이스



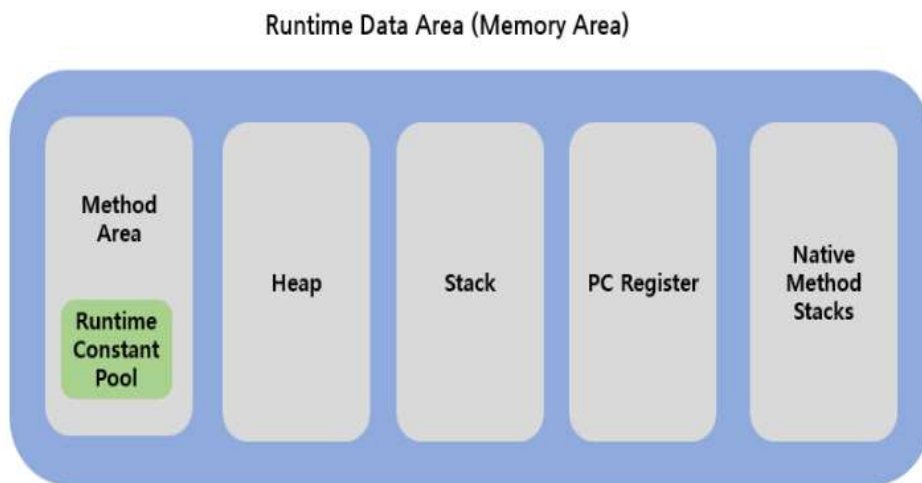


## Java 프로그램이 실행되는 구조





# ■ 메모리 영역



## 1. Method Area

- Class, 전역변수, Static 변수 및 상수 정보가 저장되는 공간
- 모든 스레드에서 정보가 공유

## 2. Heap

- 동적으로 생성된 데이터가 저장되는 공간(new, Array 등)
- 참조타입(Reference Type)의 데이터가 저장되는 공간

## 3. Stack

- 잠시 사용되고 필요가 없어지는 데이터가 저장되는 공간
- 참조타입(Reference Type)일 경우에는 Heap에 저장된 데이터의 주소 값을 Stack에 저장해서 사용

## 4. PC Register

- 스레드가 생성되면서 생기는 공간

## 5. Native Method Stack

- Java가 아닌 다른 언어(C, C++)로 구성된 메소드를 실행이 필요할 때 사용되는 공간

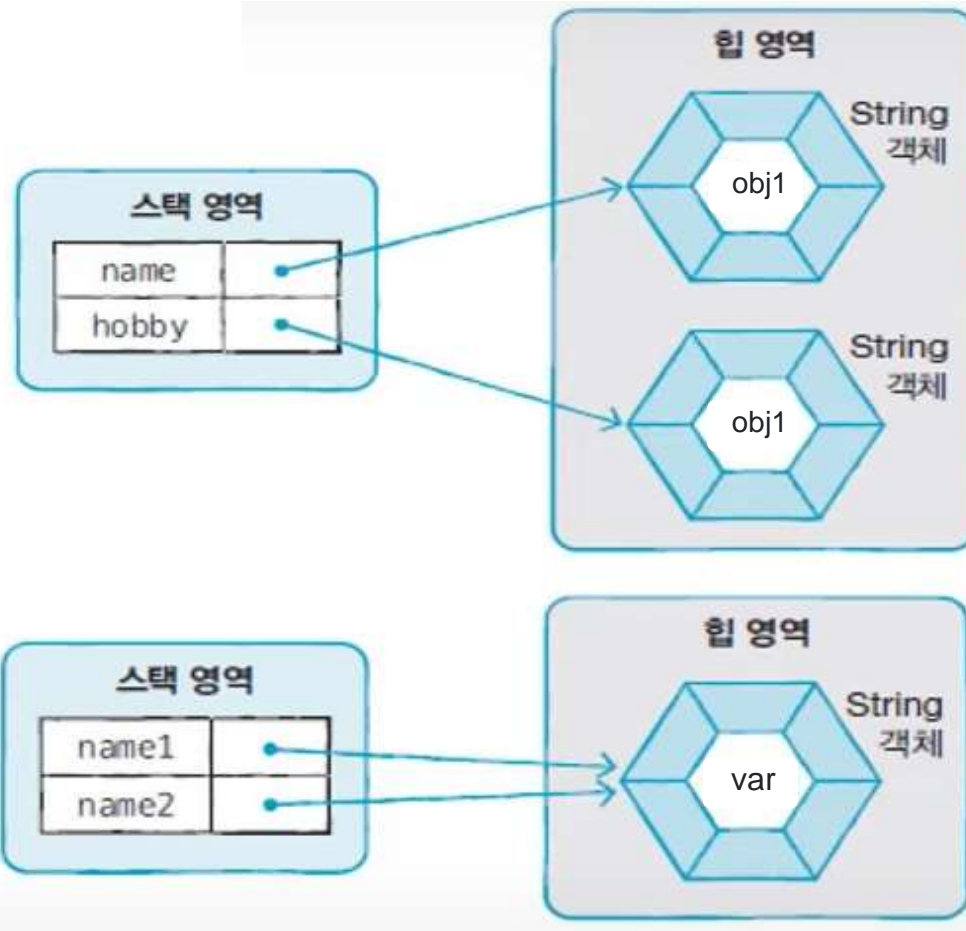


## ■ 메모리 영역 – Heap, Stack 비교

1. 힙 메모리는 애플리케이션의 모든 부분에서 사용되며, 스택 메모리는 하나의 스레드가 실행될 때 사용.
2. 힙과 메서드 공간에 저장된 객체는 어디서든지 접근이 가능하지만, 스택 메모리는 다른 스레드가 접근할 수 없다.
3. 언제든지 객체가 생성되면 항상 힙 공간에 저장되며, 스택 메모리는 힙 공간에 있는 객체를 참조만 한다.  
→ 스택 메모리는 primitive 타입의 지역변수와 힙 공간에 있는 객체 참조 변수만 갖고 있다.
4. 스택 메모리의 생명주기는 매우 짧으며, 힙 메모리는 애플리케이션의 시작부터 끝까지 살아남는다.
5. 스택 메모리가 가득차면 자바에서는 `java.lang.StackOverflowError`를 발생.  
힙 메모리가 가득차면 `java.lang.OutOfMemoryError : Java Heap Space` 에러를 발생
6. 스택 메모리 사이즈는 힙 메모리와 비교했을 때 매우 적다.
7. 스택 메모리는 간단한 메모리 할당 방법(LIFO)를 사용하므로 힙 메모리보다 빠르다



## 참조 타입의 비교 연산



참조타입 변수에서 `==`, `!=` 연산자는  
메모리 주소를 비교

```
String name = new String("obj1");  
String hobby = new String("obj1");
```

```
System.out.println(name == hobby);  
System.out.println(name.equals(hobby));
```

```
String name1 = "var";  
String name2 = "var";
```

```
System.out.println(name1 == name2);  
System.out.println(name1.equals(name2));
```



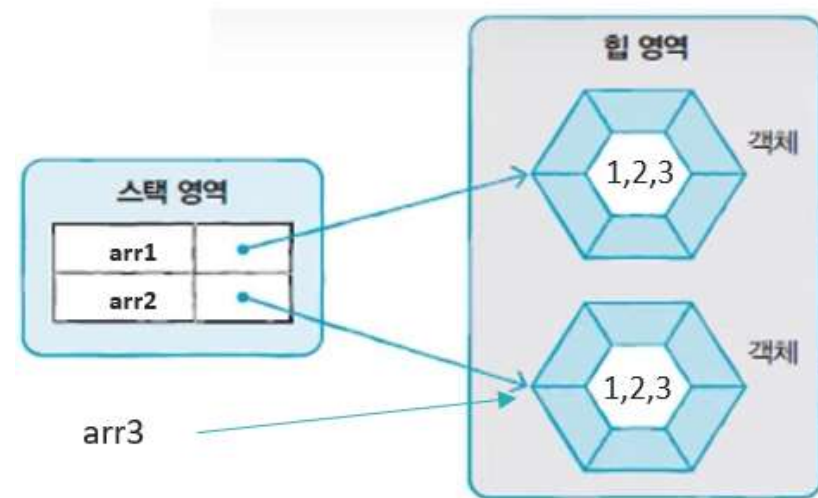
## 참조 타입의 비교 연산

ReferenceVariableCompareExample.java

```
1 package ch05.sec03;
2
3 public class ReferenceVariableCompareExample {
4     public static void main(String[] args) {
5         int[] arr1; //배열 변수 arr1 선언
6         int[] arr2; //배열 변수 arr2 선언
7         int[] arr3; //배열 변수 arr3 선언
8
9         arr1 = new int[] { 1, 2, 3 }; //배열 { 1, 2, 3 }을 생성하고 arr1 변수에 대입
10        arr2 = new int[] { 1, 2, 3 }; //배열 { 1, 2, 3 }을 생성하고 arr2 변수에 대입
11        arr3 = arr2; //배열 변수 arr2의 값을 배열 변수 arr3에 대입
12
13        System.out.println(arr1 == arr2); //arr1과 arr2 변수 값이 동일한 변수인지 검사
14        System.out.println(arr2 == arr3); //arr2와 arr3 변수가 같은 배열을 참조하는지 검사
15    }
16 }
17
18
```

Console Problems Debug Shell

```
<terminated> ReferenceVariableCompareExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse.just
false
true
```





## 참조 타입의 초기화

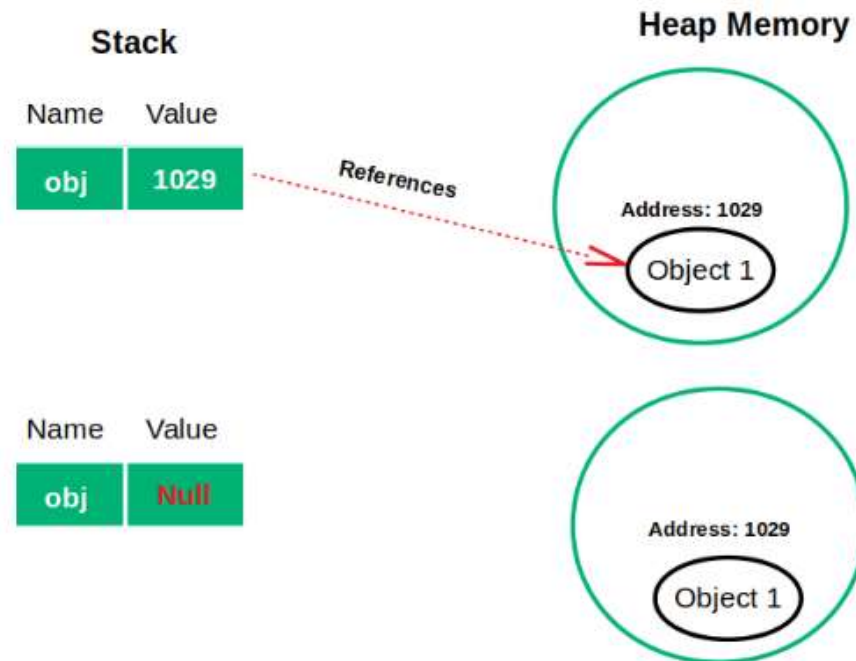
```
1 package ch05.sec04;
2
3 public class NullPointerExceptionExample {
4     public static void main(String[] args) {
5         int[] intArray = null;
6         //intArray[0] = 10; //NullPointerException
7
8         String str1 = null;
9         //System.out.println("총 문자수: " + str1.length() ); //NullPointerException
10        //String str2;
11        String str3 = null;
12
13        //System.out.println("str1 == str2 : " + (str1 == str2)); // Error !!
14        System.out.println("str1 == str2 : " + (str1 == str3));
15    }
16 }
17
```

\*\* 데이터의 값이 null 인 상태에서 사용하는 경우  
→ NullPointerException 발생

```
String refVar1 = "Java";
String refVar2 = null;
```

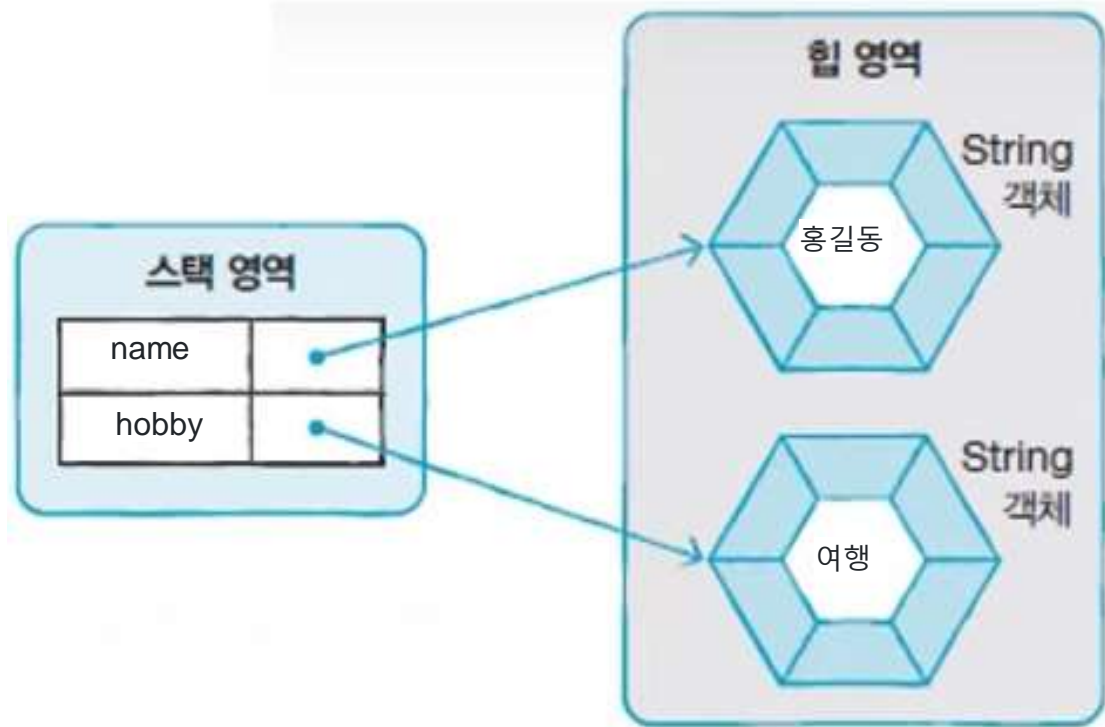
```
String refVal3;
String refVal4 = null;
System.out.println(refVal3 == refVal4); ???
```

```
String refVal5 = null;
String refVal6 = null;
System.out.println(refVal5 == refVal6); ???
```



\* null 은 주소값이 없다는 것을 표현하는 키워드

## ■ 문자열 타입



```
String name = "홍길동";  
String hobby = "여행";
```

```
String name1 = "홍길동";  
String name2 = "홍길동";  
Name1 == name2 ?? True
```

```
String name1 = new String("홍길동");  
String name2 = new String("홍길동");  
Name1 == name2 ?? False
```



## 문자열 타입 비교

`equals()` : String 객체의 내부 문자열만  
비교하는 경우

```
EqualsExample.java ×
1 package ch05.sec05;
2
3 public class EqualsExample {
4     public static void main(String[] args) {
5         String strVar1 = "홍길동";
6         String strVar2 = "홍길동";
7
8         if(strVar1 == strVar2) {
9             System.out.println("strVar1과 strVar2는 참조가 같음");
10        } else {
11            System.out.println("strVar1과 strVar2는 참조가 다름");
12        }
13
14        if(strVar1.equals(strVar2)) {
15            System.out.println("strVar1과 strVar2는 문자열이 같음");
16        }
17
18        String strVar3 = new String("홍길동");
19        String strVar4 = new String("홍길동");
20
21        if(strVar3 == strVar4) {
22            System.out.println("strVar3과 strVar4는 참조가 같음");
23        } else {
24            System.out.println("strVar3과 strVar4는 참조가 다름");
25        }
26
27        if(strVar3.equals(strVar4)) {
28            System.out.println("strVar3과 strVar4는 문자열이 같음");
29        }
30    }
31 }
32
```

Console × Problems Debug Shell

<terminated> EqualsExample [Java Application] C:\Users\LewisW.p2\pool\plugins\org.eclipse

strVar1과 strVar2는 참조가 같음  
strVar1과 strVar2는 문자열이 같음  
strVar3과 strVar4는 참조가 다름  
strVar3과 strVar4는 문자열이 같음



## 문자열 타입 비교

```
EmptyStringExample.java ×
1 package ch05.sec05;
2
3 public class EmptyStringExample {
4     public static void main(String[] args) {
5         String hobby = "";
6         if(hobby.equals("")) {
7             System.out.println("hobby 변수가 참조는 String 객체는 빈문자열");
8         }
9     }
10 }
11
```

Console × Problems Debug Shell

<terminated> EmptyStringExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse

hobby 변수가 참조는 String 객체는 빈문자열

1. String 변수에 "" 빈 문자열인 경우에도 비교
2. null != ""

```
String sVal1 = null;
String sVal2 = "";
System.out.println("1.Check: " + (sVal1 == sVal2));|
```



## ■ 문자열 타입 비교 테스트

임의의 난수(1 ~ 10)를 생성하고 숫자를 입력하여 맞추는 프로그램을 작성 하시오.  
단 기회는 3번 까지로 지정 하시오.

→ 숫자 비교가 아닌 string 비교를 사용





## 문자열 추출

**charAt** : 특정 위치의 문자열 추출  
\* 문자열 인덱스는 0 부터 시작

**String substring(int start)** : 현재 문자열 객체  
start위치부터 끝까지 문자열 발취

**String substrnig(int start, int end)** : 현재 문자열  
객체에서 start 부터 end 직전까지  
문자열 발취

**int indexOf(int ch)** : 현재 문자열 객체에서  
ch문자가 첫번째로 발견된 위치를  
반환, 없으면 -1반환

**int indexOf(String str)** : 현재 문자열 객체에서  
str를 찾아서 존재하면 첫째문자  
위치를 반환하고 없으면 -1반환

```
CharAtExample.java x
1 package ch05.sec05;
2
3 public class CharAtExample {
4     public static void main(String[] args) {
5         String ssn = "0106241230123";
6         char sex = ssn.charAt(6);
7         switch (sex) {
8             case '1':
9             case '3':
10             System.out.println("남자 입니다.");
11             break;
12             case '2':
13             case '4':
14             System.out.println("여자 입니다.");
15             break;
16         }
17
18         //=====
19         String str = "CharAtExample_Test1";
20         String s1 = str.substring(12); // 시작값만 주어진다면 그 위치부터 끝까지 추출
21         System.out.printf("s1 : %s\n", s1);
22         String s2 = str.substring(12,15); // 시작값위치부터 끝값-1 위치까지 추출(끝값위치의 문자는 포함하지않음)
23         System.out.printf("s2 : %s\n", s2);
24
25         //=====
26         String str2 = "CharAtExample_Test2";
27         int a1 = str2.indexOf('E'); // 맨 처음 'E' 값의 위치를 찾을
28         System.out.printf("a1 : %d\n", a1);
29     }
30 }
31
```

Console x Problems Debug Shell

```
<terminated> CharAtExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22
남자 입니다.
s1 : e_Test1
s2 : e_T
a1 : 6
```





## 문자열 길이 및 대체

**length()** : 특정 문자열에서 문자 개수 추출

String sSubject = "Java Progaming";

int iLength = sSubject.length(); => 14

**replace()** : 문자열에서 특정 문자열을 다른 문자열로 교체

String sOldStr = "Java Prog";

String sNewStr = sOldStr.replace("Java", "JAVA");

→ JAVA Prog

```
1 package ch05.sec05;
2
3 public class LengthReplaceExample {
4     public static void main(String[] args) {
5         //-----
6         String ssn = "7306241230123";
7         int length = ssn.length();
8         if(length == 13) {
9             System.out.println("주민번호 자리수가 맞습니다.");
10        } else {
11            System.out.println("주민번호 자리수가 틀립니다.");
12        }
13
14        //-----
15        String oldStr = "Java 문자열은 불변입니다. Java 문자열은 String입니다.";
16        String newStr = oldStr.replace("Java", "JAVA");
17
18        System.out.println(oldStr);
19        System.out.println(newStr);
20    }
21 }
22
```

Console × Problems Debug Shell

<terminated> LengthReplaceExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse

주민번호 자리수가 맞습니다.

Java 문자열은 불변입니다. Java 문자열은 String입니다.

JAVA 문자열은 불변입니다. JAVA 문자열은 String입니다.



## 문자열 분리

**split** : 문자열을 특정 구분자를 사용하여 따로 분리하여 문자열 추출

```
String sClass = "Java,C,Python,Go";
```

```
String[] arr = sClass.split(",");
```

```
arr[0] == "Java";
```

```
arr[1] == "C";
```

```
arr[2] == "Python";
```

```
arr[3] == "Go";
```

arr 에 있는 총 개수 : arr.length → 4개

```
1 package ch05.sec05;
2
3 public class SplitExample {
4     public static void main(String[] args) {
5         String board = "1,자바학습,참조타입 String을 학습합니다.,홍길동";
6
7         //문자열 분리
8         String[] tokens = board.split(",");
9
10        //인덱스별로 읽기
11        System.out.println("번호: " + tokens[0]);
12        System.out.println("제목: " + tokens[1]);
13        System.out.println("내용: " + tokens[2]);
14        System.out.println("쓴이: " + tokens[3]);
15        System.out.println();
16
17        //for문을 이용한 읽기
18        for(int i=0; i<tokens.length; i++) {
19            System.out.println(tokens[i]);
20        }
21    }
22 }
23
```

Console × Problems Debug Shell

<terminated> SplitExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse

번호: 1  
제목: 자바학습  
내용: 참조타입 String을 학습합니다.  
쓴이: 홍길동

1  
자바학습  
참조타입 String을 학습합니다.  
홍길동



## ■ 문자열 연습문제 1

String sTot = "10Point, 23Point, 33Point, 45Point, 55Point, 67Point";  
이라는 문자열에서 모든 점수 총 합과 평균 점수는 몇 점인가?  
아래와 같이 표현되도록 코딩하시오

```
<terminated> ScoreCheck [Java  
1.TotCnt : 6  
2.SumValue : 233  
3.Average : 38.833332  
4.Average :      38.83
```



## ■ 문자열 연습문제 2

math, calc, science 라는 과목이 있고

```
String sStudent1 = " math:85, calc:93, science:71 " ;  
String sStudent2 = " calc:71, math:88, science:92 " ;
```

위와같이 2 학생이 해당 과목에 점수를 받았다면  
각각의 총 합과 평균 점수는 몇 점인가?

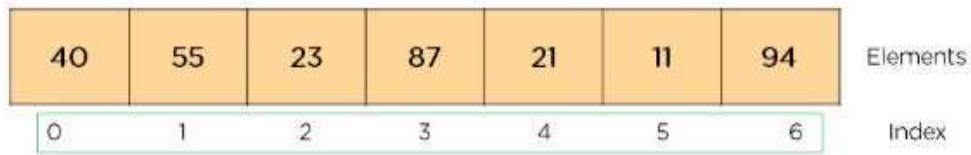
아래와 같이 표현되도록 코딩하시오.

```
<terminated> E:\source\check java Application  
Tot Math : 173    Average : 86.5  
Tot Calc : 164    Average : 82.0  
Tot Science : 163    Average : 81.5
```



## 배열 타입

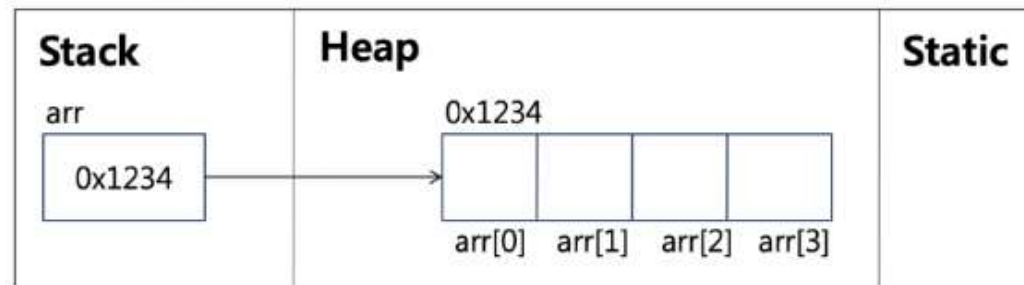
- 배열이란 동일한 데이터 타입을 가지는 데이터 구조  
→ 하나의 변수로 여러 인스턴스를 표현하는 것



Elements[0] == 40;

Elements[5] == 11;

`int[] arr = new int[4];`



- 배열의 장점
  - 요소에 대한 효율적인 액세스
  - 빠른 데이터 검색 - 연속된 메모리 위치에 저장(메모리 효율성)
  - 구현의 용이함



## ■ 배열 변수 선언

- 배열 선언 방법 : 변수타입[] 변수명;  
                          변수타입 변수[];

- 배열 선언 및 초기화  
    int[] iArray = null;  
    String[] sArray = null;  
    double[] dArray = null;

-> 배열 값이 null 인 상태에서 사용하면 NullPointerException 이 발생



## ■ 값 목록으로 배열 생성

1. 변수타입[] 변수명 = { 값1, 값2, 값3..... };
2. 변수타입[] 변수명;  
   변수명 = { 값1, 값2, 값3..... };  
   ➔ 컴파일 에러 : 배열 변수 선언과  
                      동시에 사용해야함  
   or 변수명 = new 타입[] { 값1, 값2, 값3..... };

```
1 package ch05.sec06;
2
3 public class ArrayCreateByVaueListExample1 {
4     public static void main(String[] args) {
5         //배열 변수 선언과 배열 생성
6         String[] season = { "Spring", "Summer", "Fall", "Winter" };
7
8         //배열의 항목값 읽기
9         System.out.println("season[0] : " + season[0]);
10        System.out.println("season[1] : " + season[1]);
11        System.out.println("season[2] : " + season[2]);
12        System.out.println("season[3] : " + season[3]);
13
14        //인덱스 1 항목의 값 변경
15        season[1] = "여름";
16        System.out.println("season[1] : " + season[1]);
17        System.out.println();
18
19        //배열 변수 선언과 배열 생성
20        int[] scores = { 83, 90, 87 };
21
22        //총합과 평균 구하기
23        int sum = 0;
24        for(int i=0; i<3; i++) {
25            sum += scores[i];
26        }
27        System.out.println("총합 : " + sum);
28        double avg = (double) sum / 3;
29        System.out.println("평균 : " + avg);
30    }
31 }
```

Console × Problems Debug Shell

<terminated> ArrayCreateByVaueListExample1 [Java Application] C:\Users\LewisW.p2\pool\plugins\

```
season[0] : Spring
season[1] : Summer
season[2] : Fall
season[3] : Winter
season[1] : 여름

총합 : 260
평균 : 86.66666666666667
```



## ■ new 연산자로 배열 생성

1. 값의 목록은 없지만 생성될 값을 위한 객체 생성
2. 변수타입[] 변수명 = new 변수타입[길이];
3. 변수타입[] 변수명 = null;  
변수명 = new 변수타입[길이];

타입	타입	타입	타입	타입	타입	타입	
기본값	기본값	기본값	기본값	기본값	기본값	기본값	Elements
0	1	2	3	4	5	6	Index

```
<terminated> ArrayCreateByNewExample [Java Application] C:\W
arr1[0] : 0, arr1[1] : 0, arr1[2] : 0,
arr1[0] : 10, arr1[1] : 20, arr1[2] : 30,

arr2[0] : 0.0, arr2[1] : 0.0, arr2[2] : 0.0,
arr2[0] : 0.1, arr2[1] : 0.2, arr2[2] : 0.3,

arr3[0] : null, arr3[1] : null, arr3[2] : null,
arr3[0] : 1월, arr3[1] : 2월, arr3[2] : 3월,
```

```
1 package ch05.sec06;
2
3 public class ArrayCreateByNewExample {
4     public static void main(String[] args) {
5         //배열 변수 선언과 배열 생성
6         int[] arr1 = new int[3];
7         //배열 항목의 초기값 출력
8         for(int i=0; i<3; i++) {
9             System.out.print("arr1[" + i + "] : " + arr1[i] + ", ");
10        }
11        System.out.println();
12        //배열 항목의 값 변경
13        arr1[0] = 10;
14        arr1[1] = 20;
15        arr1[2] = 30;
16        //배열 항목의 변경값 출력
17        for(int i=0; i<3; i++) {
18            System.out.print("arr1[" + i + "] : " + arr1[i] + ", ");
19        }
20        System.out.println("\n");
21
22        //배열 변수 선언과 배열 생성
23        double[] arr2 = new double[3];
24        //배열 항목의 초기값 출력
25        for(int i=0; i<3; i++) {
26            System.out.print("arr2[" + i + "] : " + arr2[i] + ", ");
27        }
28        System.out.println();
29        //배열 항목의 값 변경
30        arr2[0] = 0.1;
31        arr2[1] = 0.2;
32        arr2[2] = 0.3;
33        //배열 항목의 변경값 출력
34        for(int i=0; i<3; i++) {
35            System.out.print("arr2[" + i + "] : " + arr2[i] + ", ");
36        }
37        System.out.println("\n");
38
39        //배열 변수 선언과 배열 생성
40        String[] arr3 = new String[3];
41        //배열 항목의 초기값 출력
42        for(int i=0; i<3; i++) {
43            System.out.print("arr3[" + i + "] : " + arr3[i] + ", ");
44        }
45        System.out.println();
46        //배열 항목의 값 변경
47        arr3[0] = "1월";
48        arr3[1] = "2월";
49        arr3[2] = "3월";
50        //배열 항목의 변경값 출력
51        for(int i=0; i<3; i++) {
52            System.out.print("arr3[" + i + "] : " + arr3[i] + ", ");
53        }
54    }
55 }
56
```





## 배열 길이

1. 배열에 저장할 수 있는 항목 수
2. 변수타입[] 변수명 = { 값1, 값2, 값3};  
변수명.length;  
→ 3
3. 변수명.length = 5; → 컴파일 오류

```
1 package ch05.sec06;
2
3 public class ArrayLengthExample {
4     public static void main(String[] args) {
5         //배열 변수 선언과 배열 대입
6         int[] scores = { 84, 90, 96 };
7
8         //배열 항목의 총합 구하기
9         int sum = 0;
10        for(int i=0; i<scores.length; i++) {
11            sum += scores[i];
12        }
13        System.out.println("총합 : " + sum);
14
15        //배열 항목의 평균 구하기
16        double avg = (double) sum / scores.length;
17        System.out.println("평균 : " + avg);
18    }
19 }
20
```

Console × Problems Debug Shell

<terminated> ArrayLengthExample [Java Application] C:\Users\#Lewis#\p2\#

총합 : 270

평균 : 90.0



## 다차원 배열

\* 2차원 배열

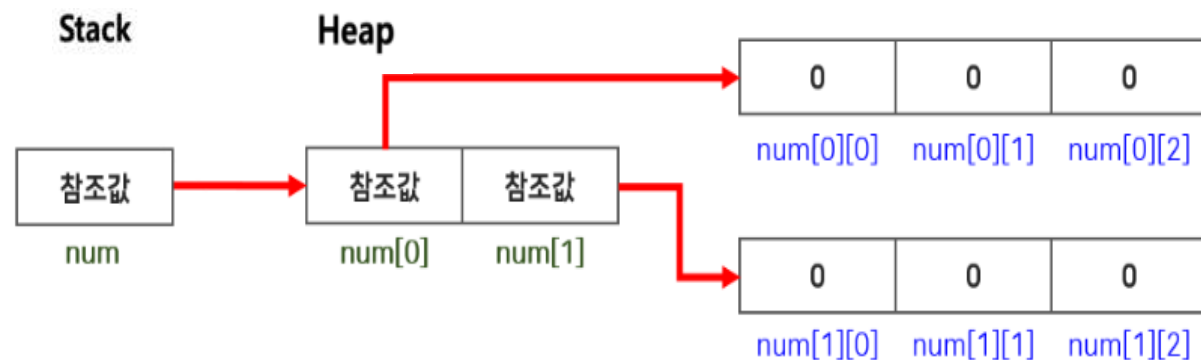
1. 자료형이 같은 1차원 배열들의 묶음
2. 할당 된 공간 마다 인덱스 번호 두개 부여  
( 행, 열 )

3. 값 목록으로 배열 생성

```
타입[][] 변수 = { {}, {}..... };  
int[][] iScores = { { 80, 90, 70 }  
                    , { 76, 86, 96 } };  
iScores[0][1] ➔ 90;  
iScores[1][2] ➔ 96;
```

4. new 연산자로 배열 생성

```
타입[][] 변수 = new 타입[1차원수][2차원수];  
int[][] iScores = new int[2][3];
```





## 2차원 배열

### 3. 값 목록으로 배열 생성

타입[][] 변수 = { {}, {}..... };

int[][] iScores = { { 80, 90, 70 },  
                          { 76, 86, 96 } };

iScores[0][1] ➔ 90;

iScores[1][2] ➔ 96;

```
1 package ch05.sec07;
2
3 public class MultidimensionalArrayByValueListExample {
4     public static void main(String[] args) {
5         //2차원 배열 생성
6         int[][] scores = {
7             { 80, 90, 96 },
8             { 76, 88 }
9         };
10
11         //배열의 길이
12         System.out.println("1차원 배열 길이(반의 수): " + scores.length);
13         System.out.println("2차원 배열 길이(첫번째 반의 학생 수): " + scores[0].length);
14         System.out.println("2차원 배열 길이(두번째 반의 학생 수): " + scores[1].length);
15
16         //첫번째 반의 세번째 학생의 점수 읽기
17         System.out.println("scores[0][2]: " + scores[0][2]);
18
19         //두번째 반의 두번째 학생의 점수 읽기
20         System.out.println("scores[1][1]: " + scores[1][1]);
21
22         //첫번째 반의 평균 점수 구하기
23         int class1Sum = 0;
24         for(int i=0; i<scores[0].length; i++) {
25             class1Sum += scores[0][i];
26         }
27         double class1Avg = (double) class1Sum / scores[0].length;
28         System.out.println("첫번째 반의 평균 점수: " + class1Avg);
29
30         //첫번째 반의 평균 점수 구하기
31         int class2Sum = 0;
32         for(int i=0; i<scores[1].length; i++) {
33             class2Sum += scores[1][i];
34         }
35         double class2Avg = (double) class2Sum / scores[1].length;
36         System.out.println("두번째 반의 평균 점수: " + class2Avg);
37
38         //전체 학생의 평균 점수 구하기
39         int totalStudent = 0;
40         int totalSum = 0;
41         for(int i=0; i<scores.length; i++) { //반의 수만큼 반복
42             totalStudent += scores[i].length; //반의 학생수 합산
43             for(int k=0; k<scores[i].length; k++) { //해당 반의 학생 수만큼 반복
44                 totalSum += scores[i][k]; //학생 점수 합산
45             }
46         }
47         double totalAvg = (double) totalSum / totalStudent;
48         System.out.println("전체 학생의 평균 점수: " + totalAvg);
49     }
50 }
51
```



## 2차원 배열

4. new 연산자로 배열 생성  
타입[][] 변수 = new  
타입[1차원수][2차원수];  
int[][] iScores = new int[2][3];

```
1 package ch05.sec07;
2
3 public class MultidimensionalArrayByNewExample {
4     public static void main(String[] args) {
5         // 각 반의 학생 수가 3명으로 동일할 경우 점수 저장을 위한 2차원 배열 생성
6         int[][] mathScores = new int[2][3];
7         // 배열 항목 초기값 출력
8         for (int i = 0; i < mathScores.length; i++) {
9             for (int k = 0; k < mathScores[i].length; k++) {
10                 System.out.println("mathScores[" + i + "][" + k + "]: " + mathScores[i][k]);
11             }
12         }
13         System.out.println();
14         // 배열 항목 값 변경
15         mathScores[0][0] = 80;
16         mathScores[0][1] = 83;
17         mathScores[0][2] = 85;
18         mathScores[1][0] = 86;
19         mathScores[1][1] = 90;
20         mathScores[1][2] = 92;
21         // 전체 학생의 수학 평균 구하기
22         int totalStudent = 0;
23         int totalMathSum = 0;
24         for (int i = 0; i < mathScores.length; i++) {
25             totalStudent += mathScores[i].length;
26             for (int k = 0; k < mathScores[i].length; k++) {
27                 totalMathSum += mathScores[i][k];
28             }
29         }
30         double totalMathAvg = (double) totalMathSum / totalStudent;
31         System.out.println("전체 학생의 수학 평균 점수: " + totalMathAvg);
32         System.out.println();
33     }
34 }
```

```
33
34 // 각 반의 학생 수가 다를 경우 점수 저장을 위한 2차원 배열 생성
35 int[][] englishScores = new int[2][];
36 englishScores[0] = new int[2];
37 englishScores[1] = new int[3];
38 // 배열 항목 초기값 출력
39 for (int i = 0; i < englishScores.length; i++) {
40     for (int k = 0; k < englishScores[i].length; k++) {
41         System.out.println("englishScores[" + i + "][" + k + "]: " + englishScores[i][k]);
42     }
43 }
44 System.out.println();
45 // 배열 항목 값 변경
46 englishScores[0][0] = 90;
47 englishScores[0][1] = 91;
48 englishScores[1][0] = 92;
49 englishScores[1][1] = 93;
50 englishScores[1][2] = 94;
51 // 전체 학생의 영어 평균 구하기
52 totalStudent = 0;
53 int totalEnglishSum = 0;
54 for (int i = 0; i < englishScores.length; i++) {
55     totalStudent += englishScores[i].length;
56     for (int k = 0; k < englishScores[i].length; k++) {
57         totalEnglishSum += englishScores[i][k];
58     }
59 }
60 double totalEnglishAvg = (double) totalEnglishSum / totalStudent;
61 System.out.println("전체 학생의 영어 평균 점수: " + totalEnglishAvg);
62 }
63 }
64 }
```



## 배열 테스트 1

길이가 10인 배열을 선언하고 임의의 수( 1부터 10까지)를  
생성 하여 배열에 등록 하시오.  
그 중 7이 몇 번째의 값인지 출력하는 소스를 구현 하시오.

```
<terminated> ArrayTest [Java Appl  
5 5 5 8 5 7 9 9 7 1  
7 is 6 th Number.
```

```
<terminated> ArrayTest [Java Ap  
10 7 6 10 7 6 7 10 1 10  
7 is 2 th Number.  
7 is 5 th Number.  
7 is 7 th Number.
```



## ■ 배열 테스트 2

학생 3명이 있습니다

1번 학생의 국어, 영어, 수학 점수는 90, 80, 72

2번 학생의 국어, 영어, 수학 점수는 94, 85, 92

3번 학생의 국어, 영어, 수학 점수는 87, 91, 78

점 입니다.

학생들의 과목별 총점 및 평균, 개인별 총점 및 평균을  
구하는  
프로그램을 코딩 하시오.

```
1 Student KOR : 90 MATH : 80 ENG : 72 [ SumVal : 242, Avr : 80.67 ]
2 Student KOR : 94 MATH : 85 ENG : 92 [ SumVal : 271, Avr : 90.33 ]
3 Student KOR : 87 MATH : 91 ENG : 78 [ SumVal : 256, Avr : 85.33 ]
KOR SUM : 271, Avr : 90.33
MATH SUM : 256, Avr : 85.33
ENG SUM : 242, Avr : 80.67
```

## ■ 배열 복사

1. 배열은 한 번 생성하면 길이 변경 불가
2. 다른 배열을 생성하여 값을 복사
3. 얇은 복사 - Shallow Copy  
=> 변경 시 원본 배열에 바로 적용
4. 깊은 복사 - Deep Copy  
=> 독립된 형태로 생성됨
5. 2차원 이상의 경우 DeepCopy를 하기 위해선 for 문으로 값을 넣어야 한다.





## 얕은 복사

\* Shallow Copy

```
<terminated> ShallowCop
a Array : 1 8 3
b Array : 1 8 3
-----
a Array : 10 8 3
b Array : 10 8 3
```

```
1 package ch05.sec09;
2
3 public class ShallowCopy {
4
5     public static void main(String[] args) {
6         int a[] = {1, 8, 3};
7         int b[] = a;
8
9         System.out.print("a Array : ");
10        for (int i = 0; i < a.length; i++)
11            System.out.print(a[i] + " ");
12
13        System.out.print("\nb Array : ");
14        for (int i = 0; i < b.length; i++)
15            System.out.print(b[i] + " ");
16
17        System.out.print("\n-----\n");
18        b[0] = 10;
19        System.out.print("a Array : ");
20        for (int i = 0; i < a.length; i++)
21            System.out.print(a[i] + " ");
22
23        System.out.print("\nb Array : ");
24        for (int i = 0; i < b.length; i++)
25            System.out.print(b[i] + " ");
26    }
27
28 }
29
```





## ■ 깊은 복사 1

- Deep Copy

b = a.clone();

```
<terminated> DeepCopy [Java Application]
```

```
a Array : 1 8 3
```

```
b Array : 1 8 3
```

```
-----
```

```
a Array : 1 8 3
```

```
b Array : 10 8 3
```

```
1 package ch05.sec09;
2
3 public class DeepCopy {
4
5     public static void main(String[] args) {
6         int a[] = {1, 8, 3};
7         int b[] = a.clone();
8
9         System.out.print("a Array : ");
10        for (int i = 0; i < a.length; i++)
11            System.out.print(a[i] + " ");
12
13        System.out.print("\nb Array : ");
14        for (int i = 0; i < b.length; i++)
15            System.out.print(b[i] + " ");
16
17        System.out.print("\n-----\n");
18        b[0] = 10;
19        System.out.print("a Array : ");
20        for (int i = 0; i < a.length; i++)
21            System.out.print(a[i] + " ");
22
23        System.out.print("\nb Array : ");
24        for (int i = 0; i < b.length; i++)
25            System.out.print(b[i] + " ");
26    }
27 }
28
```



## ■ 깊은 복사 2

- 기존 배열의 일부만 덮어쓰우기
- Clone 보다 빠르게
- 배열의 길이 확인 필수!

**System.arraycopy( A, sourcePos, B, destPos, len );**

- Object A : 복사할 대상
- int sourcePos : 복사할 배열(A)의 시작 index
- Object B : 복사될 곳
- int destPos : 복사될 곳(B)의 시작 index
- int len : 복사할 길이(원소 개수)

```
<terminated> DeepCopy [Java Application] C:\Users+  
a Array : 10 20 30 40 50 60 70 80 90 100  
b Array : 0 1 2 3 4 40 50 60 70 9  
-----  
a Array : 10 20 30 40 50 60 70 80 90 100  
b Array : 10 1 2 3 4 40 50 60 70 9
```

```
1 package ch05.sec09;  
2  
3 public class DeepCopy {  
4  
5     public static void main(String[] args) {  
6         int a[] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100};  
7         int b[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
8  
9         int sourcePos = 3;  
10        int destPos = 5;  
11        int len = 4;  
12  
13        System.arraycopy(a, sourcePos, b, destPos, len);  
14  
15        System.out.print("a Array : ");  
16        for (int i = 0; i < a.length; i++)  
17            System.out.print(a[i] + " ");  
18  
19        System.out.print("\nb Array : ");  
20        for (int i = 0; i < b.length; i++)  
21            System.out.print(b[i] + " ");  
22  
23        System.out.print("\n-----\n");  
24        b[0] = 10;  
25        System.out.print("a Array : ");  
26        for (int i = 0; i < a.length; i++)  
27            System.out.print(a[i] + " ");  
28  
29        System.out.print("\nb Array : ");  
30        for (int i = 0; i < b.length; i++)  
31            System.out.print(b[i] + " ");  
32    }  
33 }  
34
```



## ■ 깊은 복사 3

### 1. 원하는 길이 설정

**B = Arrays.copyOf( A, newLength );**

- int[] A : 복사할 배열
- int newLength : 새로 만들 배열(b)의 길이

<terminated> DeepCopy [Java Applicatio

```
a Array : 1 2 3
b Array : 1 2 3 0 0
-----
a Array : 1 2 3
b Array : 10 2 3 0 0
```

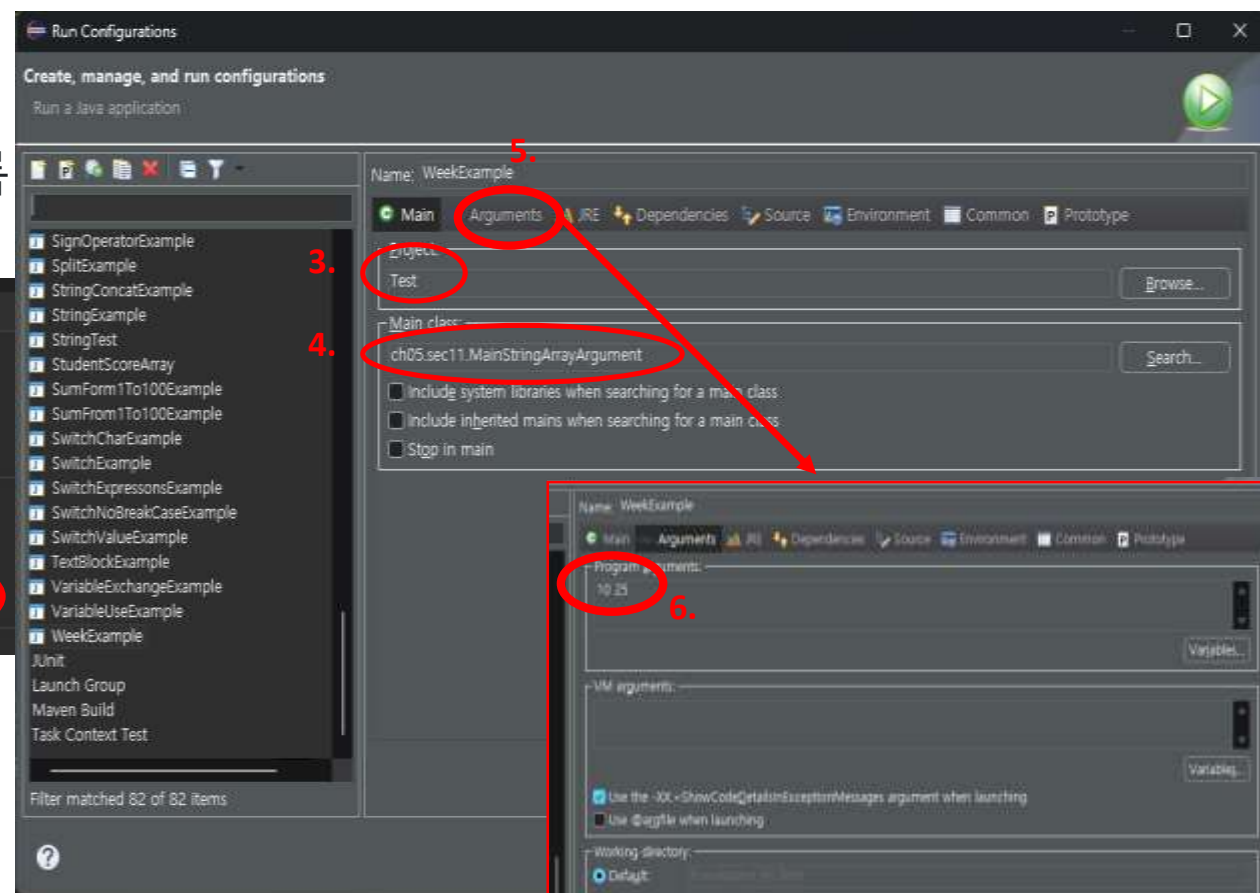
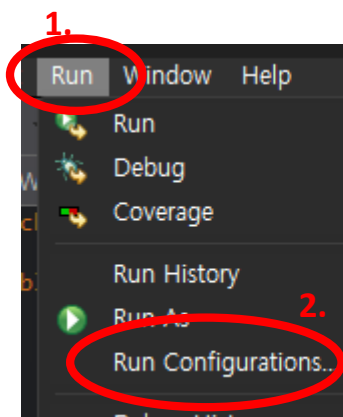
```
1 package ch05.sec09;
2 import java.util.Arrays;
3 public class DeepCopy {
4
5     public static void main(String[] args) {
6         int[] a = new int[] {1, 2, 3};
7         int[] b = Arrays.copyOf(a, 5);
8
9         System.out.print("a Array : ");
10        for (int i = 0; i < a.length; i++)
11            System.out.print(a[i] + " ");
12
13        System.out.print("\nb Array : ");
14        for (int i = 0; i < b.length; i++)
15            System.out.print(b[i] + " ");
16
17        System.out.print("\n-----\n");
18
19        b[0] = 10;
20        System.out.print("a Array : ");
21        for (int i = 0; i < a.length; i++)
22            System.out.print(a[i] + " ");
23
24        System.out.print("\nb Array : ");
25        for (int i = 0; i < b.length; i++)
26            System.out.print(b[i] + " ");
27    }
28 }
29
```



## ■ main() args 사용

- 원도우 – cmd 창에서 실행 시  
→ 최초 실행 시 사용할 인자 값 등록

Java Sum 10 20  
-> args[0] → 10;  
args[1] → 20;





## ■ main() args 실습

\* 입력 받은 2개의 값 덧셈 프로그램

1. 입력 값 1개만 입력한 경우
2. 입력 값 2개 입력한 경우

```
1 package ch05.sec11;
2
3 public class MainStringArrayArgument {
4     public static void main(String[] args) {
5         if(args.length != 2) {
6             System.out.println("프로그램 입력 값이 부족");
7             System.exit(0);
8         }
9
10        String strNum1 = args[0];
11        String strNum2 = args[1];
12
13        int num1 = Integer.parseInt(strNum1);
14        int num2 = Integer.parseInt(strNum2);
15
16        int result = num1 + num2;
17        System.out.println(num1 + " + " + num2 + " = " + result);
18    }
19 }
```

Console × Problems Debug Shell Coverage

<terminated> WeekExample [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse.ju

10 + 32 = 42



## ■ 열거(Enum) 타입

1. 한정된 값(요일이나 계절 등)의 고정된 데이터의 집합을 표현
2. 코드를 더 간결하고 의미 있는 형태로 작성

```
1 package ch05.sec12;
2
3 enum Season {
4     SPRING,
5     SUMMER,
6     AUTUMN,
7     WINTER
8 }
9
10 public class SeasonExe {
11     public String sAbbr;
12     public Season sFavoriteSeason;
13
14     public static void main(String[] args) {
15         SeasonExe season = new SeasonExe();
16         season.sAbbr = "Win";
17         season.sFavoriteSeason = Season.WINTER;
18
19         System.out.println("ABBR : " + season.sAbbr);
20         System.out.println("Full Name: " + season.sFavoriteSeason);
21     }
22 }
23
24
```

Console × Problems Debug Shell Coverage

<terminated> SeasonExe [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse.justj  
ABBR : Win  
Full Name: WINTER



## 열거(Enum) 타입

1. Week 열거형 선언
2. Calendar 사용하기 위한 선언 추가
3. import java.util.Calendar;
4. Calendar.DAY\_OF\_WEEK  
: 1 ~ 7 요일 표시
5. Calendar.YEAR, MONTH, DATE 등

p.197 Week.java

```
1 package ch05.sec12;
2
3 public enum Week {
4     MONDAY,
5     TUESDAY,
6     WEDNESDAY,
7     THURSDAY,
8     FRIDAY,
9     SATURDAY,
10    SUNDAY
11 }
12
```

```
1 package ch05.sec12;
2
3 import java.util.Calendar;
4
5 public class WeekExample {
6     public static void main(String[] args) {
7         //Week 열거 타입 변수 선언
8         Week today = null;
9
10        //Calendar 열기
11        Calendar cal = Calendar.getInstance();
12
13        //오늘의 요일 열기(1~7)
14        int week = cal.get(Calendar.DAY_OF_WEEK);
15
16        //숫자를 열거 상수로 변환해서 변수에 대입
17        switch(week) {
18            case 1: today = Week.SUNDAY; break;
19            case 2: today = Week.MONDAY; break;
20            case 3: today = Week.TUESDAY; break;
21            case 4: today = Week.WEDNESDAY; break;
22            case 5: today = Week.THURSDAY; break;
23            case 6: today = Week.FRIDAY; break;
24            case 7: today = Week.SATURDAY; break;
25        }
26
27        //열거 타입 변수를 사용
28        if(today == Week.SUNDAY) {
29            System.out.println("일요일에는 축구를 합니다." + Week.SUNDAY);
30        } else {
31            System.out.println("열심히 자바 공부합니다." + Week.SUNDAY + "--" + today);
32        }
33    }
34 }
```

Console × Problems Debug Shell Coverage

<terminated> WeekExample (1) [Java Application] C:\Users\Lewis\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe -Djava.library.path=C:\Users\Lewis\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe 열심히 자바 공부합니다.SUNDAY--THURSDAY



## ■ 열거(Enum) 타입 테스트

1. enum Operator { PLUS, MINUS, MULTIPLY, DIVIDE }
2. 두 정수와 연산자(PLUS, MINUS, MULTIPLY, DIVIDE)를 입력받아 해당 연산 결과를 출력하는 프로그램을 작성하시오.
3. 나눗셈 시 0으로 나누면 "Cannot divide by zero." 출력 하시오.

```
1. First Val : 4
2. Second Val : 3
Insert Operator(PLUS, MINUS, MULTIPLY, DIVIDE) : minus
*Result : 1
```