

객체지향 프로그램 - Java

(5일 배열)

kimth4110@gmail.com

김 태 환

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 for 문
- 1차원 배열
- 다차원 배열
- Arrays 클래스와 System 클래스를 이용한 배열의 사용



배열의 개요

■ 배열의 필요성

- 학생이 10명이 있고 이들의 평균 성적을 계산

개별변수를
사용하는 방법은
학생수가 많아지면
번거로워집니다.

방법 #1 : 개별 변수 사용

```
int s0;  
int s1;  
...  
int s9;
```

방법 #1 : 배열 사용

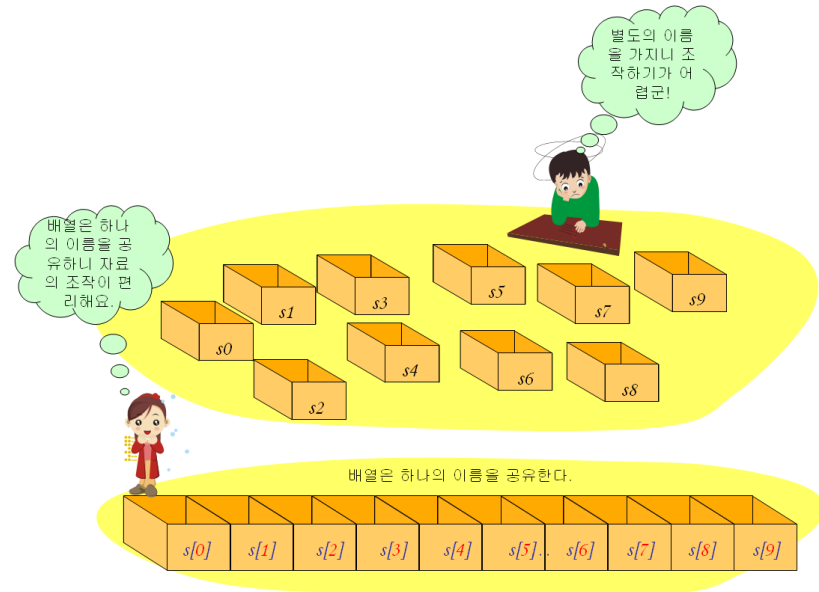
```
int s[10];
```



배열의 개요

■ 배열이란?

- 동일한 타입의 데이터들을 저장하는 장소
- 정수로 되어 있는
번호(첨자)에 의해 접근
- 여러 개의 값을 하나의
이름으로 처리 가능



배열의 개요

- 배열은 기본 자료형이 아니라 참조 자료형
 - 배열 각각의 요소는 기본 자료형, 참조 자료형 모두 가능

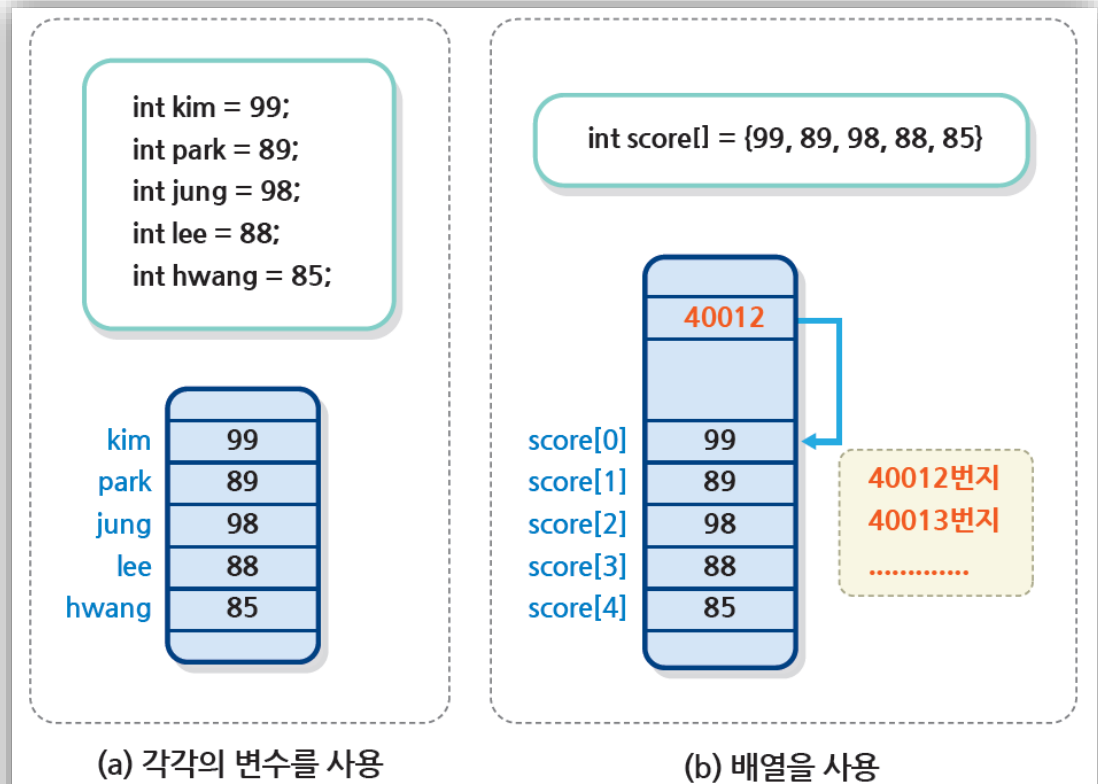


그림 6-2 변수를 사용하는 경우와 배열을 사용하는 경우의 메모리 구조

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 for 문
- 1차원 배열
- 다차원 배열
- Arrays 클래스와 System 클래스를 이용한 배열의 사용



배열의 선언과 생성

■ 배열을 사용하기 위한 선언 및 생성 과정

배열의 선언

`type name[];` ← 1차원 배열 name 선언. []를 형이나 이름에 붙인다.
`type[] name;` ←
`type[][] name;` ← 2차원 배열 name 선언. []를 하나씩 나누어 붙일 수 있다.
`type name[][];` ←
`type[] name[];` ←

배열의 생성

`name = new type[size];` ← size 크기의 1차원 배열 생성
`name = new type[size][size];` ← size 크기의 2차원 배열 생성

배열의 선언과 생성 : 한 문장으로 선언과 생성 가능

```
type[] name = new type[size];
type name[] = new type[size];
type[][] name = new type[size][size];
type name[][] = new type[size][size];
type[] name[] = new type[size][size];
```

배열의 선언과 생성

배열의 선언과 생성의 예

```
int[] id;
```

```
id = new int[3];
```

 ← 3개의 int 요소를 가진 배열 생성

또는

```
int[] id = new int[3];
```

```
String[] student_name ;
```

```
student_name = new String[3];
```

 ← 3개의 문자열 요소를 가진 배열 생성

또는

```
String[] student_name = new String[3];
```

```
int[][] id_and_score = new int[5][2];
```

 ← 5행과 2열을 가진 정수 2차원 배열 생성

```
String add_and_name[][] = new String[10][10];
```

 ← 10행과 10열을 가진 문자열 배열 생성

배열 요소의 사용 : 배열 이름과 첨자를 이용하여 접근

```
int[] id = new int[3];
```

 ← 3개의 요소를 가진 int 배열

```
int sum = int[0] + int[1];
```

 ← 첨자는 0부터 시작

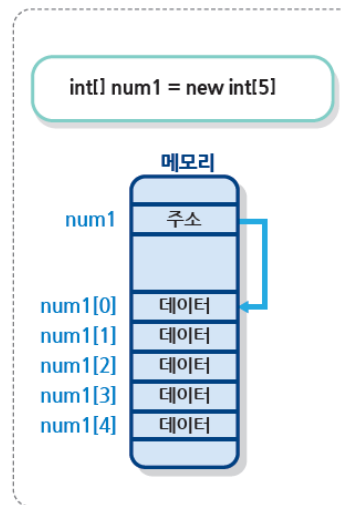
```
int[][] stnum = new int[3][3];
```

```
stnum[0][2] = stnum[0][0]+stnum[0][1];
```

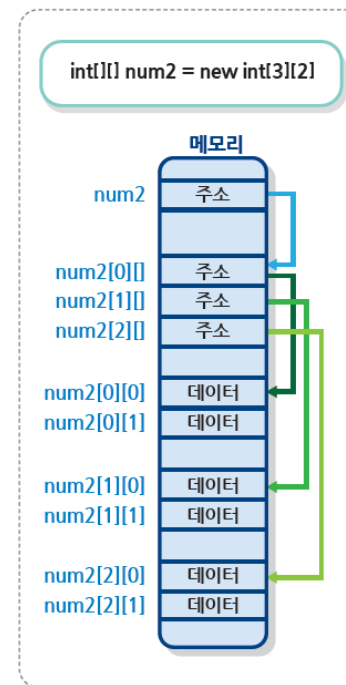
 ← 2차원 배열 첨자 사용

배열의 선언과 생성

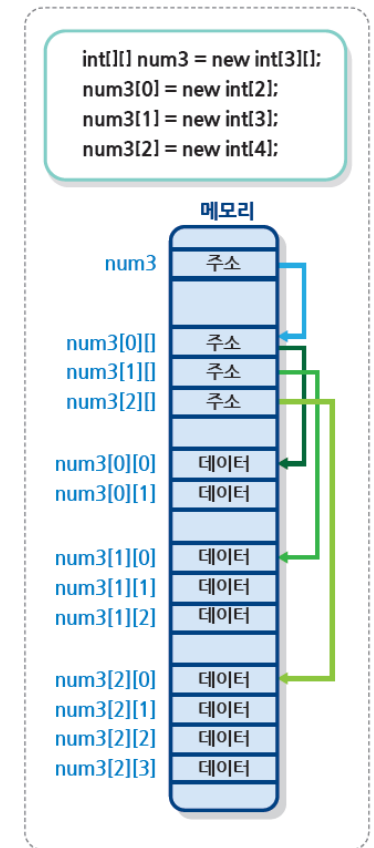
■ 배열의 첨자는 0부터 시작



(a) 1차원 배열과 메모리



(b) 2차원 배열과 메모리



(c) 배열의 요소 수가 다른
2차원 배열과 메모리

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 for 문
- 1차원 배열
- 다차원 배열
- Arrays 클래스와 System 클래스를 이용한 배열의 사용



배열의 초기화 및 확장 *for*문

■ 배열의 초기화와 요소의 수 (length)

■ 배열의 초기화

- 처음 생성된 배열에 데이터를 저장하는 과정

■ 배열은 초기화 하지 않아도 사용 가능

- 초기화 되지 않는 경우 묵시적인 값으로 자동 설정

배열의 초기화 및 확장 for문

■ 배열의 초기화와 요소의 수 (length)

배열의 생성과 초기화

```
int[] id = new int[5];  
id[0] = 201195041;  
id[1] = 201195042;  
id[2] = 201195043;  
id[3] = 201195044;  
id[4] = 201195045;
```

또는

```
int id[] = {201195041, 201195042, 201195043, 201195044, 201195045}  
String[][] name_addr = new String[3][2]  
name_addr[0][0] = "kim";  
name_addr[0][1] = "seoul";  
name_addr[1][0] = "park";  
name_addr[1][1] = "busan";  
name_addr[2][0] = "lee";  
name_addr[2][1] = "incheon";
```

또는

```
String[][] name_addr = {{ "kim", "seoul"}, { "park", "busan"}, { "lee", "incheon" }};
```

또는

```
int num[];  
num = {1,2,3,4,5};  
num = new int[] {1,2,3,4,5};
```

1차원 배열 선언과 생성

초기화. 각 요소에 값을 하나씩 저장하여 초기화

배열의 선언과 생성, 초기화 과정을
한 문장으로 작성

2차원 배열 선언과 생성

초기화. 각 요소에 값을 하나씩 저장하여 초기화

배열의 선언과 생성, 초기화 과정을
한 문장으로 작성

오류 발생 []으로 초기화를 하는 경우 선언과 동시에 초기화

new를 이용한 생성과 초기화(동시)는 가능

배열의 초기화 및 확장 for문

- 배열의 초기화와 요소의 수 (length)
 - 초기화 되지 않을 경우 가지는 묵시적인 값

형	묵시적 값
byte, short, int, long	0
float, double	0.0
char	공백 문자
boolean	거짓(false)
참조 자료형	null

배열의 초기화 및 확장 for문

- 배열의 초기화와 요소의 수 (length)
 - 배열의 길이를 나타내는 length 속성

```
int a[] = { 10, 20, 30, 40, 50 };
```

```
System.out.println(a.length);
```

← 5를 출력

```
int b[][] = {{10,20,30},{40,50,60,70}};
```

```
System.out.println(b.length);
```

← 배열 b의 행의 길이 2를 출력

```
System.out.println(b[0].length);
```

← 배열 b 첫 번째 행의 길이 3을 출력

```
System.out.println(b[1].length);
```

← 배열 b 두 번째 행의 길이 4를 출력

배열의 초기화 및 확장 for문

예제 6.1

ArraysTest1.java

```
01: import java.util.Scanner;
02: public class ArraysTest1 {
03:     public static void main(String args[])
04:     {
05:         Scanner stdin = new Scanner(System.in);
06:         int i;
07:         double sum=0.0, avg;;
08:         double dnum[] = new double[5]; ← 1차원 double 배열 선언
09:         System.out.println("dnum 배열의 길이 : " + dnum.length); ← length를 이용하여 배열의 길이 출력
10:         System.out.print("초기화 하지 않은 dnum[]의 값: ");
11:         for (i=0; i < dnum.length ; i++) ← for문의 조건으로 length 속성 사용.
12:             System.out.print(dnum[i]+" "); ← 초기화하지 않은 배열값 출력
13:         System.out.println();
14:
15:         for (i=0; i < dnum.length ; i++){ ← 반복문을 이용하여 배열 초기화
16:             System.out.print("dnum["+i+"] 번째 데이터 입력 : ");
17:             dnum[i] = stdin.nextDouble();
18:         }
19:         for (i=0; i < dnum.length ; i++) ← 배열 요소의 합을 구한다.
20:             sum = sum + dnum[i];
21:         System.out.println("배열의 합은 " + sum + "입니다");
22:         avg = sum/dnum.length;
23:         System.out.println("배열 값의 평균은 " + avg + "입니다");
24:
```

배열의 초기화 및 확장 for문

■ 배열 처리를 위한 확장된 for문 제공

- 배열의 요소를 순차적으로 처리하는 간결한 구문 제공

확장된 for문의 형식

```
for ( type 변수명 : 배열 이름 )
```

→ 처리할 배열의 이름

→ 배열의 값을 저장할 변수명

→ 배열값을 저장할 변수의 형

배열 이름으로 지정된 배열의 첫 번째 요소부터 마지막 요소까지를 변수명에 배정하여 반복 처리를 수행.

```
sum=0;
```

```
int[] inum = { 10, 20, 30, 40, 50 };
```

```
for ( int x : inum ) ← inum 배열의 첫 번째 요소부터 마지막 요소까지를 차례로 변수 x에 배정하여 반복을 처리한다.
```

```
    sum = sum + x;
```

```
System.out.println(sum);
```


배열의 초기화 및 확장 for문

■ 배열 처리를 위한 확장된 for문 제공

예제 6.2

ArraysTest2.java

```
01: public class ArraysTest2 {
02:     public static void main(String args[])
03:     {
04:         int score[] = {88,97,53,62,92,68,82};
05:         int max=score[0];
06:         for (int i : score)
07:             // 기존 for문 : for (int i=1; i < score.length ; i=i+1)
08:             {
09:                 if ( i > max )
10:                     // 기존의 for 문을 사용할 때의 if문 : if ( score[i] > max )
11:                     max = i;
12:                     // max = score[i];
13:             }
14:         System.out.println("배열 요소의 최대값은 " + max + "입니다");
15:     }
16: }
```

확장된 for문 사용. 변수 x에는 인덱스가 아닌 배열 요소의 값이 순차적으로 배정

기존의 for문 형태. 변수 i는 배열의 인덱스

i 값을 max와 비교

배열에서 인덱스 i번째의 값과 max 비교

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 **for** 문
- **1차원 배열**
- 다차원 배열
- **Arrays** 클래스와 **System** 클래스를 이용한 배열의 사용



1차원 배열

```

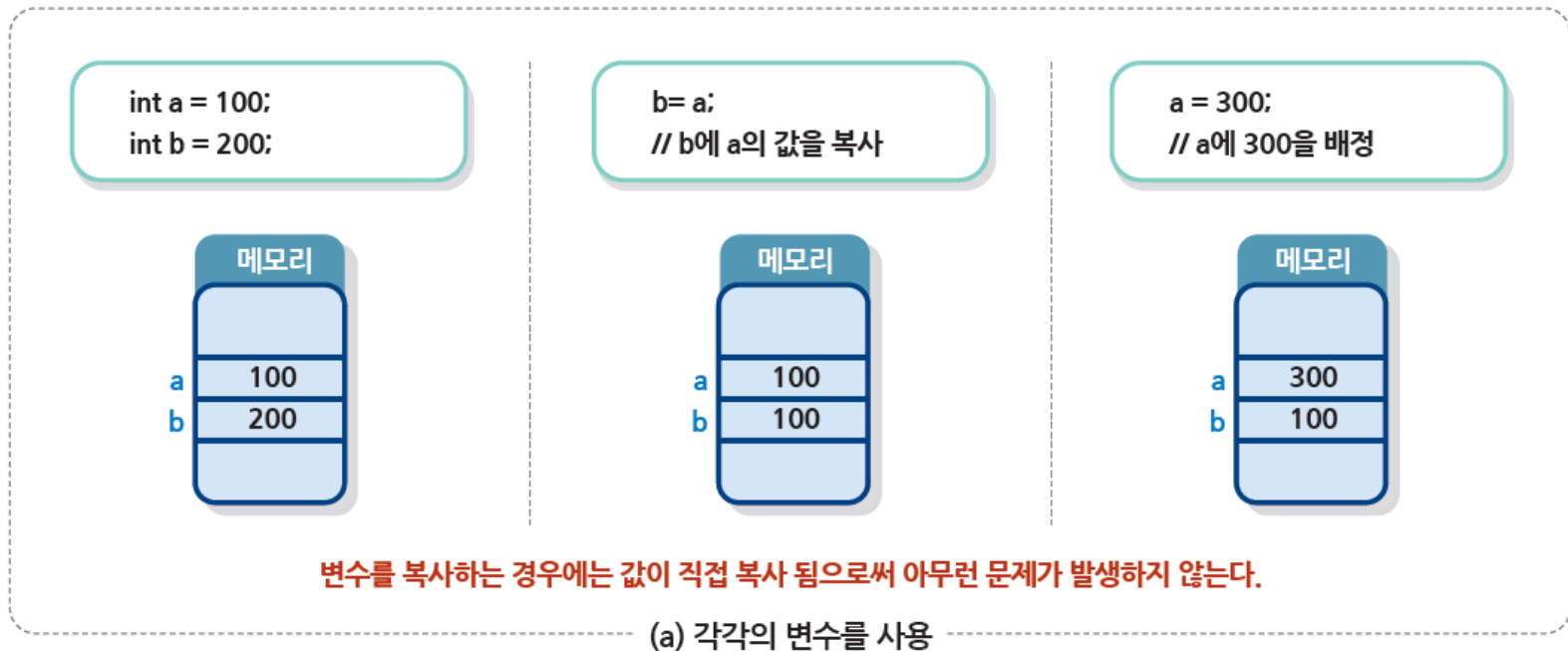
01: import java.util.Scanner;
02: public class OneArraysTest1 {
03:     public static void main(String args[])
04:     {
05:         int inum[] = {8,7,3,6,9,6,8,7,0,4,1,2};
06:         Scanner stdin = new Scanner(System.in);
07:         System.out.print("찾고 싶은 숫자 입력 : ");
08:         int key = stdin.nextInt();
09:         int count = 0;
10:         for (int i = 0 ; i < inum.length ; i++)
11:         {
12:             if ( inum[i] == key ) {
13:                 count++;
14:                 System.out.println((i+1) + "번째 데이터와 일치");
15:             }
16:         }
17:         if (count == 0)
18:             System.out.println(key + "값은 배열에 없습니다");
19:         else
20:             System.out.println(key+ "값은 배열에 "+count+"개 있습니다");
21:     }
22: }

```

09: int count = 0; ← 회수를 저장하기 위한 변수 선언
 10: for (int i = 0 ; i < inum.length ; i++) ← 위치를 알아내야 하기 때문에 확장된 for문 사용 불가
 12: if (inum[i] == key) { ← 값이 지정된 값과 일치하면 count 값을 증가하고 출력
 13: count++;
 14: System.out.println((i+1) + "번째 데이터와 일치");
 15: }
 17: if (count == 0) ← 값이 없을 경우 값이 없음을 출력
 18: System.out.println(key + "값은 배열에 없습니다");
 19: else ← 값이 있는 경우 회수를 출력
 20: System.out.println(key+ "값은 배열에 "+count+"개 있습니다");

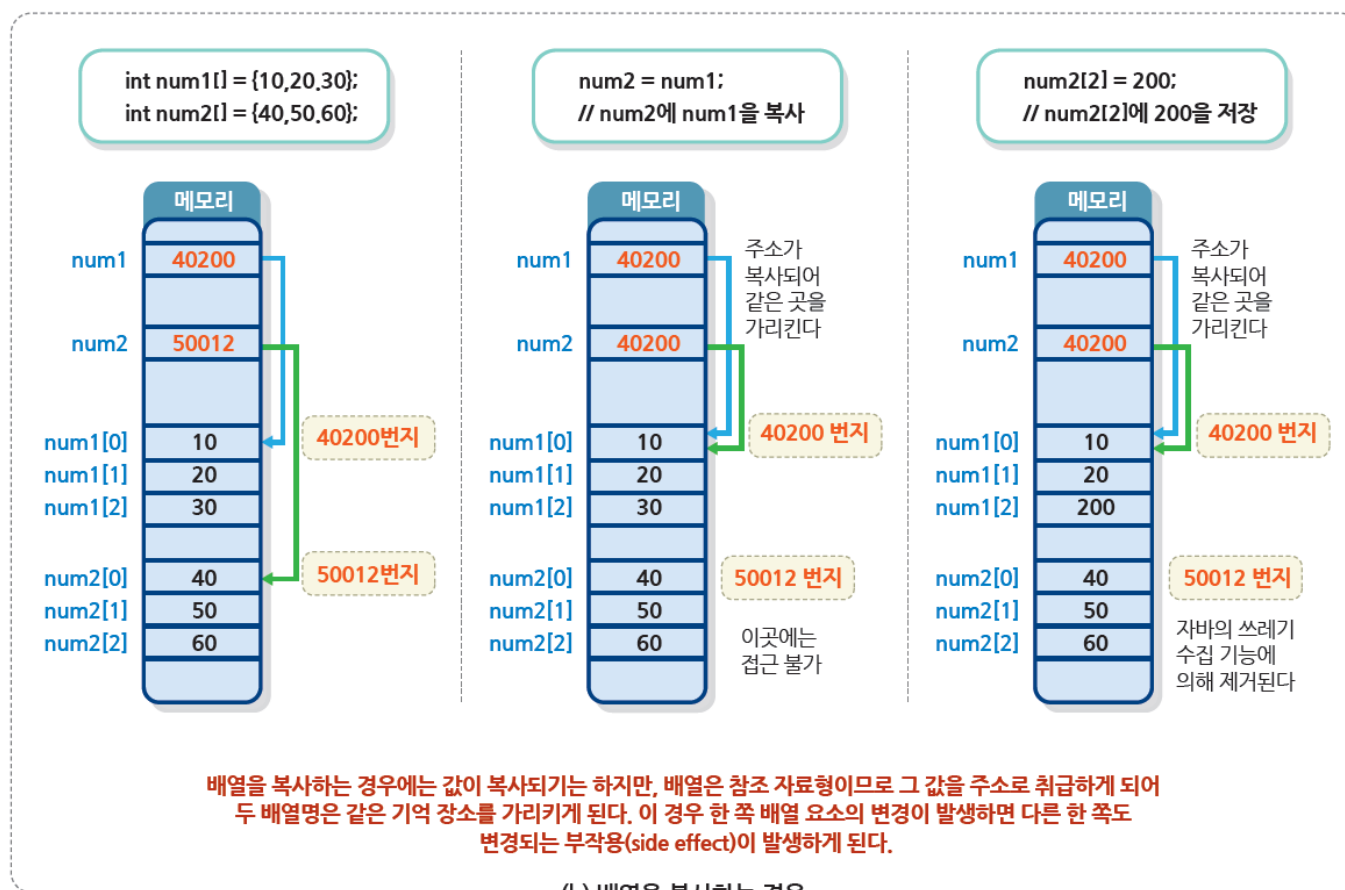
1차원 배열

■ 변수의 복사와 배열의 복사



1차원 배열

■ 변수의 복사와 배열의 복사



(b) 배열을 복사하는 경우

1차원 배열

■ 변수의 복사와 배열의 복사

예제 6.4

CopyArraysTest1.java

```
01: public class CopyArraysTest1 {
02:     public static void main(String args[])
03:     {
04:         int num1[] = {10,20,30};
05:         int num2[] = {40,50,60};
06:         num2 = num1; ← 배열명을 이용하여 복사
07:         num2[2] = 200; ← num2[2] 번째 요소의 값을 200으로 변경
08:         System.out.print("num1 배열의 값 : ");
09:         for (int i : num1) ← num1의 값을 출력. num1[2] 값이
10:             System.out.print(i+" "); ← 200으로 출력
11:         System.out.print("\nnum2 배열의 값 : ");
12:         for (int i : num2) ← num2의 값을 출력
13:             System.out.print(i+" ");
14:         int num3[] = {100,200,300}; ← num3, num4를 생성
15:         int num4[] = {400,500,600};
16:         for (int i = 0 ; i < num3.length ; i ++ ) ← 배열 요소의 값들을 각각 복사
17:             num4[i] = num3[i];
18:         num4[2]=999; ← num4[2] 요소를 999로 변경
19:         System.out.print("\nnum3 배열의 값 : ");
20:         for (int i : num3) ← num3의 값을 출력. num3[2]의 값은 그대로 300 출력
21:             System.out.print(i+" ");
22:         System.out.print("\nnum4 배열의 값 : ");
23:         for (int i : num4) ← num4의 값 출력. num4[2]는 999 출력
24:             System.out.print(i+" ");
25:     }
26: }
```

1차원 배열

예제 6.5

OneArraysTest2.java

```
01: public class OneArraysTest2 {
02:     public static void main(String args[])
03:     {
04:         String slist[] = {"seoul", "daejeon", "daegu", "kwangju", "inchon", "jeju", "busan"};
05:         System.out.print("원래의 배열 : ");
06:         for (String s : slist)
07:             System.out.print(s + " ");
08:         System.out.println();
09:         for (int i = 0 ; i < slist.length / 2 ; i++)
10:         {
11:             String temp = slist[i];
12:             slist[i] = slist[slist.length - i - 1];
13:             slist[slist.length - i - 1] = temp;
14:         }
15:         System.out.print("역순으로 재배치된 배열 : ");
16:         for (String s : slist)
17:             System.out.print(s + " ");
18:     }
19: }
```

원래의 배열을 출력

배열 길이의 반에 해당되는
횟수만큼 교환 개수가 홀수
인 경우는 가운데는 바뀌지
않는다.

역순으로 재배치된 배열 출력

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 **for** 문
- 1차원 배열
- 다차원 배열
- **Arrays** 클래스와 **System** 클래스를 이용한 배열의 사용



다차원 배열

■ 자바에서 2차원 배열은 1차원 배열의 배열

예제 6.6

MultiArraysTest1.java

```
01: import java.util.Random;
02: public class MultiArraysTest1 {
03:     public static void main(String args[])
04:     {
05:         Random ran = new Random();
06:         int[][] score = new int[4][4];
07:         for(int i=0 ; i< 4 ; i++)
08:             for(int j=0; j<4 ; j++)
09:                 score[i][j] = ran.nextInt(10);
10:
11:         for (int k=0 ; k < 4 ; k++) {
12:             int sum=0;
13:             for (int value : score[k]) {
14:                 System.out.print(value + " ");
15:                 sum = sum + value;
16:             }
17:             System.out.println("의 합계는 "+sum);
18:         }
19:     }
20: }
```

Random 클래스를 import 시킨다.

Random 객체를 생성

2차원 배열 score 생성

score 배열을 0-9 사이의 정수
난수로 초기화

각 행의 합계를 구하기 위해
sum 변수 0으로 설정

각 행을 출력

각 행의 합계를 출력

다차원 배열

■ 자바에서 3차원 배열은 2차원 배열의 배열

예제 6.8

MultiArraysTest3.java

```
01: public class MultiArraysTest3 {
02:     public static void main(String args[])
03:     {
04:         int[][][] threeD = new int[3][4][5]; ← 3차원 배열을 선언
05:         int i, j, k, count=11;
06:         for( i=0 ; i < threeD.length ; i++) ←
07:             for( j=0; j < threeD[i].length ; j++)
08:                 for( k=0; k < threeD[i][j].length ; k++)
09:                     {
10:                         threeD[i][j][k] = count;
11:                         count++;
12:                     } ← 3차원 배열의 초기화
13:         for(i=0; i<threeD.length; i++) ←
14:         {
15:             System.out.println((i+1) + "번째 2 차원 배열 ");
16:             for(j=0; j<threeD[i].length; j++)
17:             {
18:                 for(int val : threeD[i][j])
19:                     System.out.print(val + " ");
20:                 System.out.println();
21:             }
22:             System.out.println(); ← 3차원 배열의 출력
23:         }
24:     }
25: }
```

수업 목표

- 배열의 개요
- 배열의 선언과 생성
- 배열의 초기화 및 확장 **for** 문
- 1차원 배열
- 다차원 배열
- **Arrays** 클래스와 **System** 클래스를 이용한 배열의 사용



Arrays 클래스와 System 클래스 스를 이용한 배열의 사용

■ Arrays 클래스 와 System 클래스

■ 배열을 복사하기 위한 메소드

메소드	설명
<code>static int binarySearch(int[] a, int key)</code>	배열 a에서 key로 지정된 값을 찾아 반환. boolean을 제외한 7가지 기본 자료형과 참조 자료형 사용 가능. 이 메소드는 배열 요소들이 정렬된 상태에서 사용되어야 한다. 배열에서 key로 지정된 값의 위치를 반환한다.
<code>static boolean equals(int[] a, int[] b)</code>	배열 a와 b가 같은지를 비교하여 결과를 반환. 8개의 기본 자료형과 참조 자료형에서도 사용 가능
<code>static void fill(int[] a, int value)</code>	배열 a의 모든 요소를 value값으로 설정한다.
<code>static void fill (int[] a, int from, int to, int value)</code>	배열 a의 from부터 to-1까지를 value값으로 설정한다(인덱스 값 기준). 8개의 기본 자료형과 참조 자료형에서도 사용 가능
<code>static void sort(int[] a)</code>	배열 a의 요소들을 정렬. boolean을 제외한 7가지 기본 자료형과 참조 자료형에서도 사용 가능.
<code>static void sort(int[] a, int from, int to)</code>	배열 a의 from부터 to까지를 정렬. boolean을 제외한 7가지 기본 자료형과 참조 자료형에서도 사용 가능.
<code>static String toString(int[] a)</code>	배열 a의 요소들을 문자열로 반환한다. 이 메소드는 모든 자료형에 적용 가능.

Arrays 클래스와 System 클래스를 이용한 배열의 사용

■ Arrays 클래스의 사용 예

```
int[] a = new int[10];  
Arrays.fill(a,1); ←----- 배열 a의 모든 요소를 1로 채운다.  
Arrays.fill(a,1,5,10); ←----- a[1]부터 a[4]까지의 값을 10으로 채운다.  
System.out.println(Arrays.toString(a)); ←----- 배열 a를 문자열로 출력  
  
int[] b = { 3, 7, 1, 0, 8, 9 };  
Arrays.sort(b); ←----- 배열 b의 요소들을 오름차순으로 정렬  
System.out.println(Arrays.toString(b)); ←----- 정렬된 결과 출력  
System.out.println(Arrays.binarySearch(b, 7)); ←----- 7의 인덱스 값 3 출력  
  
int[] c = {1, 2, 3};  
int[] d = {1, 2, 3};  
int[] e = {4, 5, 6};  
System.out.println(Arrays.equals(c,d)); ←----- true 출력  
System.out.println(Arrays.equals(c,e)); ←----- false 출력
```

Arrays 클래스와 System 클래스를 이용한 배열의 사용

■ System 클래스의 arraycopy() 메소드 사용 예

메소드	설명
<code>static void arraycopy(int[] a, int s_index, int[] b, int t_index, int num)</code>	배열 a에서 s_index로 지정된 요소부터 배열 b의 t_index로 지정된 위치로 num 개의 요소를 복사한다.

```
int[] a = {1, 2, 3};
int[] b = {4, 5, 6};
int[] c = new int[3];
System.arraycopy(a,0,b,0,3); ←----- 배열 a의 처음부터 3개의 요소를 배열 b에 복사
System.out.println(Arrays.toString(b)); ←----- [1, 2, 3] 출력
System.arraycopy(a,0,c,0,3); ←----- 배열 a의 처음부터 3개의 요소를 배열 c에 복사
System.out.println(Arrays.toString(c)); ←----- [1, 2, 3] 출력
```

Arrays 클래스와 System 클래스를 이용한 배열의 사용

예제 6.9

ArraysCMethodTest1.java

```
01: import java.util.Arrays;
02: public class ArraysCMethodTest1 {
03:     public static void main(String[] args) {
04:         int[] int1 = {9,1,7,3,5,4,6,2,8,0};
05:         System.out.println("초기배열 : " + Arrays.toString(int1));
06:         Arrays.fill(int1, 3, 5, 33);
07:         System.out.println("fill() 수행 후 : " + Arrays.toString(int1));
08:         Arrays.sort(int1);
09:         System.out.println("sort() 수행 후 : " + Arrays.toString(int1));
10:         System.out.println("33은 배열의 " + Arrays.binarySearch(int1,33) +
    "번째 요소");
11:         int[] int2 = {5,4,3,2,1};
12:         System.out.println("두 번째 배열 : " + Arrays.toString(int2));
13:         System.out.println("두 개의 배열이 같은가? " + Arrays.equals(int1,
    int2));
14:         int[] int3 = new int[5];
15:         System.arraycopy(int2, 0, int3, 0, 5);
16:         System.out.println("복사된 배열 : " + Arrays.toString(int3));
17:     }
18: }
```

배열의 요소를 출력

int1의 int[3]과 int[4]를 33으로 바꾼다.

int1 배열을 오름차순으로 정렬

정렬된 int1 배열에서 33의 위치를 찾는다. 2진 탐색.

int1과 int2가 같은가? false 출력

5개의 요소를 가진 빈 배열을 선언

배열을 복사

Arrays 클래스와 System 클래스를 이용한 배열의 사용

예제 6.10

ArraysCMethodTest2.java

```
01: import java.util.Arrays;
02: public class ArraysCMethodTest2 {
03:     public static void main(String args[]) {
04:         String[] array1 = {"IMF", "제주도", "자바도사", "한글나라", "Computer",
            "모카", "JAVA", "인터넷탐색", "초롱초롱", "come", "바람", "스크립터",
            "군고구마", "도서", "their", "country" }; ← 16개의 문자열 요소를 가진 배열 생성
05:         System.out.println("===== 정렬 전 데이터 =====");
06:         System.out.println(Arrays.toString(array1)); ← array1 배열 출력
07:         Arrays.sort(array1); ← array1 배열을 오름차순으로 정렬
08:         System.out.println("===== 정렬 후 데이터 =====");
09:         System.out.println(Arrays.toString(array1)); ← 정렬된 array1 배열 요소 출력
10:         System.out.println("군고구마는 배열의 " + Arrays.binarySearch(array1, "
            군고구마") + "번째 요소"); ← "군고구마"가 배열의 몇 번째 요소인지 출력
11:         String[] array2 = array1; ← array2에 array1을 대입(같은 배열을 가리킨다)
12:         System.out.println("array1과 array2가 같은가? : "
            +Arrays.equals(array1,array2)); ← 두 배열이 같은가? true 출력
13:         String[] array3 = new String[20]; ← 20개의 요소를 가진 문자열 배열 생성
14:         System.arraycopy(array2, 0, array3, 0, array2.length); ←
15:         System.out.println("array3 배열 : " + Arrays.toString(array3));
16:     }
17: }
```

array2를 array3에 복사



Q & A

- Thank you for your attention