

# **Revisit Java Programming (Array)**

**Fall, 2020**

# Table of Contents

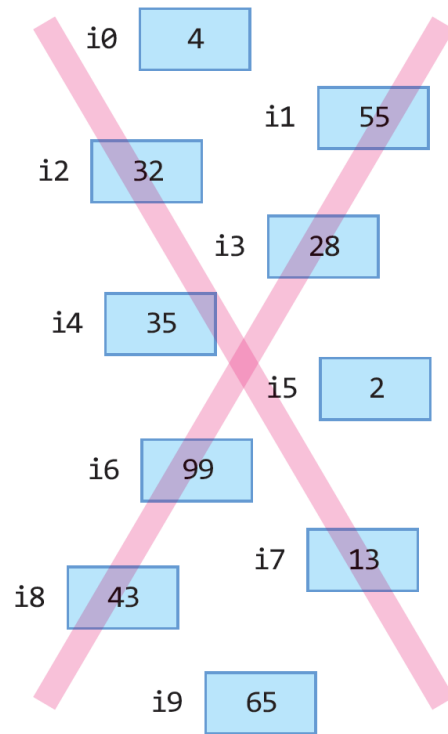
- Array
  - 1 dimensional array
  - Iteration
  - 2 dimensional array
  - Practice

# Array

- Array
  - A data structure of homogeneous values accessed by index

(1) 10개의 정수형 변수를 사용하는 경우

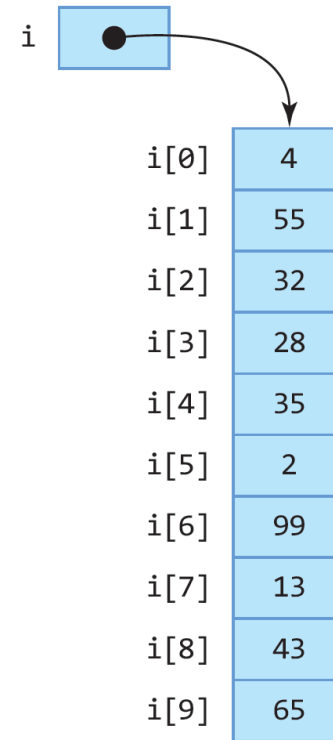
```
int i0, i1, i2, i3, i4, i5, i6, i7, i8, i9;
```



```
sum = i0+i1+i2+i3+i4+i5+i6+i7+i8+i9;
```

(2) 10개의 정수로 구성된 배열을 사용하는 경우

```
int i[] = new int[10];
```



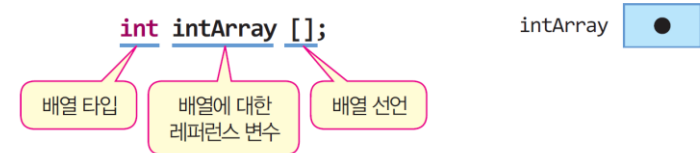
```
for(sum=0, n=0; n<10; n++)  
    sum += i[n];
```

# One dimensional array

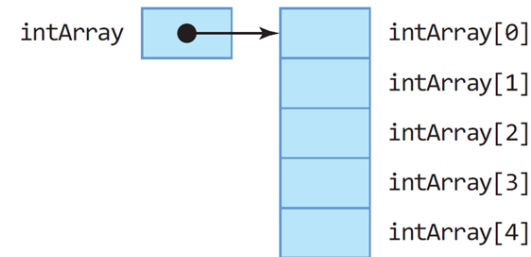
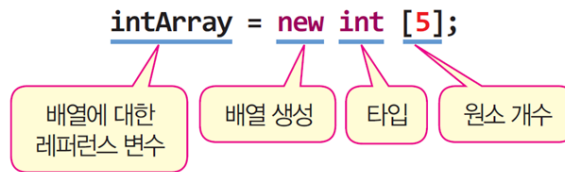
- Initialization

- `int intArray[] = new int[10];`
- `char charArray[] = new char[20];`
- `int intArray[] = {0,1,2,3,4,5,6,7,8,9};`
- `char charArray[] = {'a','b','c','d'};`

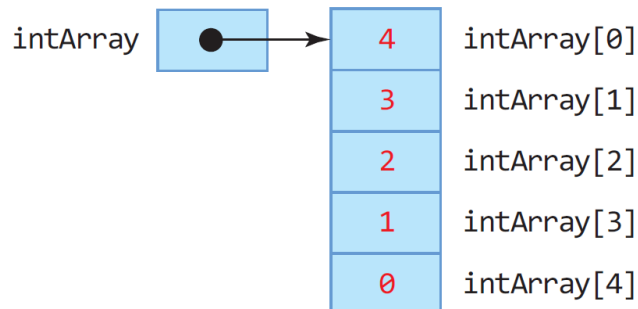
(1) 배열에 대한 레퍼런스 변수 `intArray` 선언



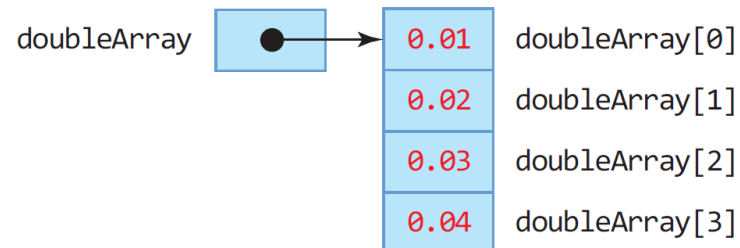
(2) 배열 생성



`int intArray[] = {4, 3, 2, 1, 0};`



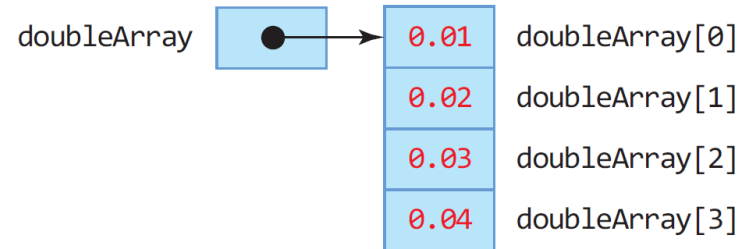
`double doubleArray[] = {0.01, 0.02, 0.03, 0.04};`



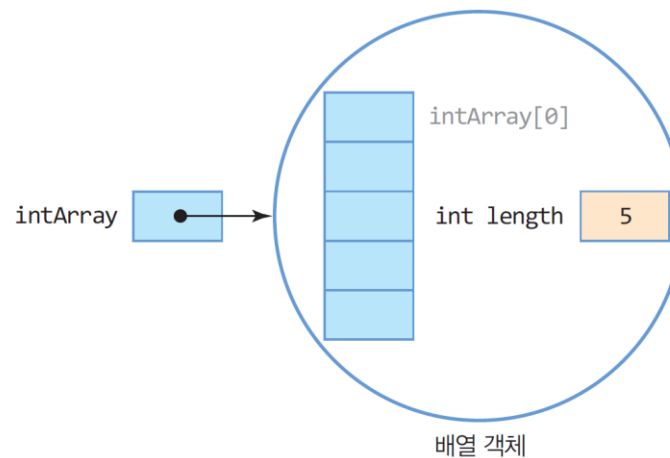
# One dimensional array

- Array Index
  - From 0 to Length -1 (e.g., 0~3)
- SET value
  - `doubleArray[1] = 0.05`
- GET value
  - `System.out.println(doubleArray[1]);`
- Get length using special length field
  - `doubleArray.length`

```
double doubleArray[] = {0.01, 0.02, 0.03, 0.04};
```

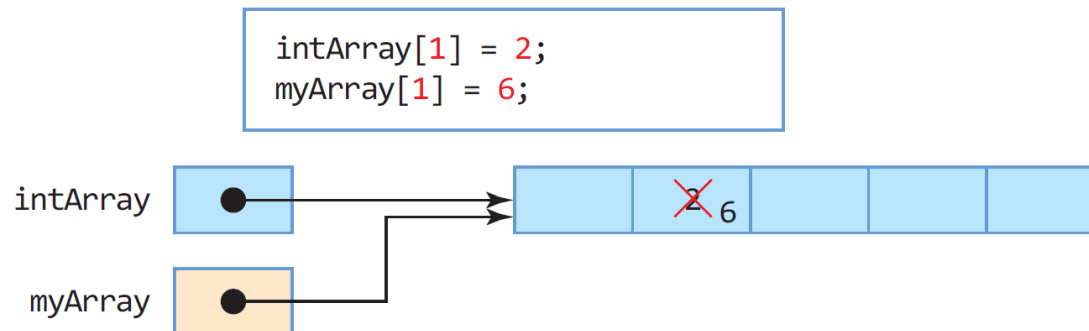
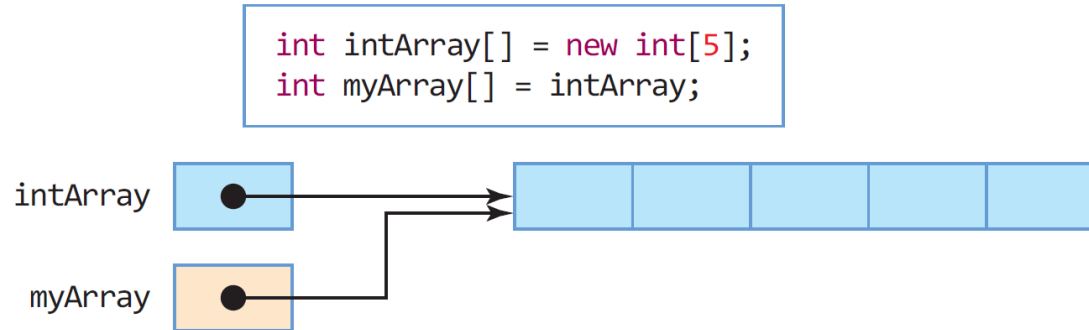


```
int intArray[];  
intArray = new int[5];  
  
int size = intArray.length;  
// size = 5
```



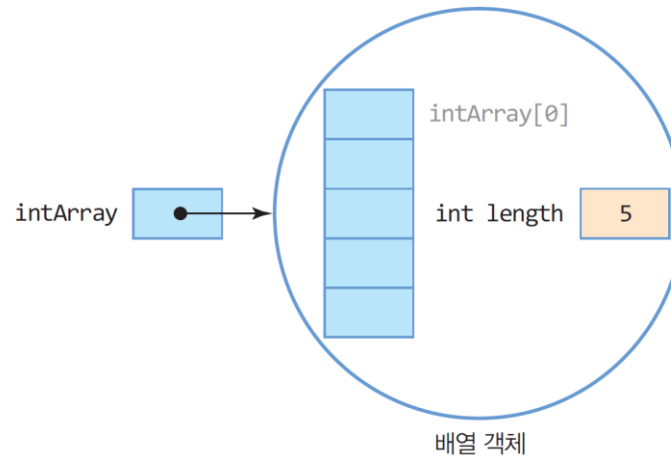
# About Reference Type

- Revisit Java Data Type
  - Primitive types
  - Reference types (e.g., Array)
- An array variable indicates a location for the array



# Array Iteration (Index)

```
int intArray[];  
intArray = new int[5];  
  
int size = intArray.length;  
// size = 5
```



- `for( int i = 0 ; i < intArray.length ; i++){`
  - `int currentValue = intArray[i];`
- `}`

## Practice 1: 예제 3-7 : 배열에 입력받은 수 중 제일큰수 찾기

양수 5개를 입력 받아 배열에 저장하고, 제일 큰 수를 출력하는 프로그램을 작성하라.

양수 5개를 입력하세요.

1

39

78

100

99

가장 큰 수는 100입니다.

Algorithm to find a maximum value

- Prepare one variable with a minimum value
- Iterate values and update the variable if a value exceeds the temporary maximum value



# Practice 1: 예제 3-7 : 배열에 입력받은 수 중 제일큰수 찾기

양수 5개를 입력 받아 배열에 저장하고, 제일 큰 수를 출력하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class ArrayAccess {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int intArray[] = new int[5]; // 배열 생성

        int max=0;    // 현재 가장 큰 수
        System.out.println("양수 5개를 입력하세요.");
        for(int i=0; i<5; i++) {
            intArray[i] = scanner.nextInt(); // 입력받은 정수를 배열에 저장
            if(intArray[i] > max) // intArray[i]가 현재 가장 큰 수보다 크면
                max = intArray[i]; // intArray[i]를 max로 변경
        }
        System.out.print("가장 큰 수는 " + max + "입니다.");

        scanner.close();
    }
}
```

양수 5개를 입력하세요.

1

39

78

100

99

가장 큰 수는 100입니다.

## Practice 2: 예제 3-8 : 배열 원소의 평균 구하기

배열의 length 필드를 이용하여 배열 크기만큼 정수를 입력 받고 평균을 구하는 프로그램을 작성하라.

5개의 정수를 입력하세요>> 2 3 4 5 9  
평균은 4.6

## Practice 2: 예제 3-8 : 배열 원소의 평균 구하기

배열의 length 필드를 이용하여 배열 크기만큼 정수를 입력 받고 평균을 구하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class ArrayLength {
    public static void main(String[] args) {
        int intArray[] = new int[5]; // 배열의 선언과 생성
        int sum=0;

        Scanner scanner = new Scanner(System.in);
        System.out.print(intArray.length + "개의 정수를 입력하세요>>");
        for(int i=0; i<intArray.length; i++)
            intArray[i] = scanner.nextInt(); // 키보드에서 입력받은 정수 저장

        for(int i=0; i<intArray.length; i++)
            sum += intArray[i]; // 배열에 저장된 정수 값을 더하기

        System.out.print("평균은 " + (double)sum/intArray.length);
        scanner.close();
    }
}
```

5개의 정수를 입력하세요>> 2 3 4 5 9  
평균은 4.6

# Array Iteration (for-each)

- for-each
  - Unnecessary to deal with index
  - Less flexible

```
int[] num = { 1,2,3,4,5 };  
int sum = 0;  
for (int k : num) // 반복될 때마다 k는 num[0], num[1], ..., num[4] 값으로 설정  
    sum += k;  
System.out.println("합은 " + sum);
```

합은 15

```
String names[] = { "사과", "배", "바나나", "체리", "딸기", "포도" };  
for (String s : names) // 반복할 때마다 s는 names[0], names[1], ..., names[5] 로 설정  
    System.out.print(s + " ");
```

사과 배 바나나 체리 딸기 포도

```
enum Week { 월, 화, 수, 목, 금, 토, 일 }  
for (Week day : Week.values()) // 반복될 때마다 day는 월, 화, 수, 목, 금, 토, 일로 설정  
    System.out.print(day + "요일 ");
```

월요일 화요일 수요일 목요일 금요일 토요일 일요일

## Practice 3: 예제 3-9 : for-each 문 활용

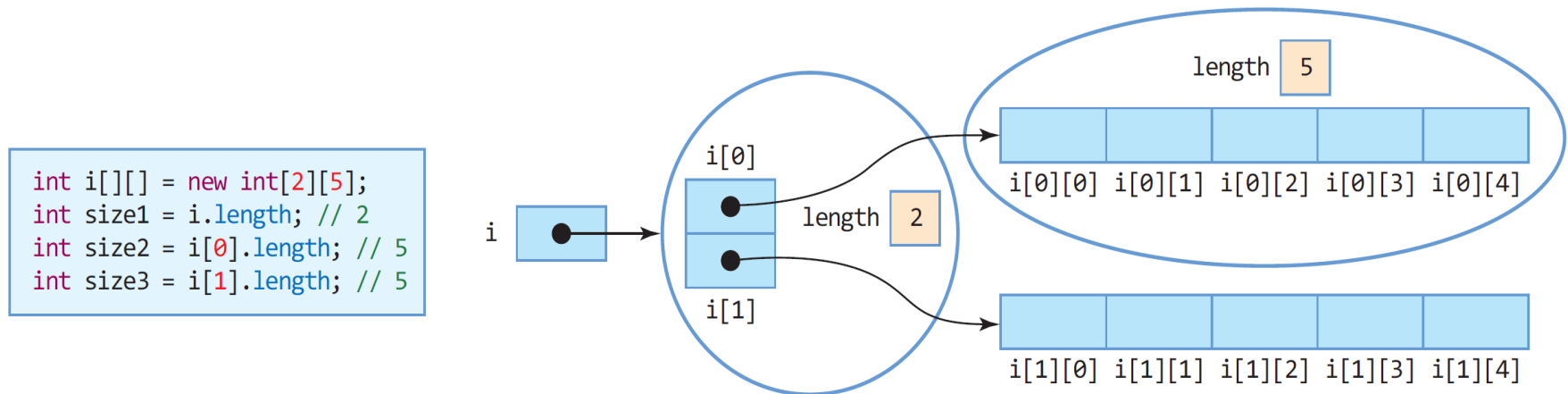
for-each 문을  
활용하는  
사례를 보자.

```
public class foreachEx {  
    enum Week { 월, 화, 수, 목, 금, 토, 일 }  
  
    public static void main(String[] args) {  
        int [] n = { 1,2,3,4,5 };  
        String names[] = { "사과", "배", "바나나", "체리", "딸기", "포도" };  
  
        int sum = 0;  
        // 아래 for-each에서 k는 n[0], n[1], ..., n[4]로 반복  
        for (int k : n) {  
            System.out.print(k + " "); // 반복되는 k 값 출력  
            sum += k;  
        }  
        System.out.println("합은" + sum);  
  
        // 아래 for-each에서 s는 names[0], names[1], ..., names[5]로 반복  
        for (String s : names)  
            System.out.print(s + " ");  
        System.out.println();  
  
        // 아래 for-each에서 day는 월, 화, 수, 목, 금, 토, 일 값으로 반복  
        for (Week day : Week.values())  
            System.out.print(day + "요일 ");  
        System.out.println();  
    }  
}
```

1 2 3 4 5 합은 15  
사과 배 바나나 체리 딸기 포도  
월요일 화요일 수요일 목요일 금요일 토요일 일요일

## 2 dimensional array

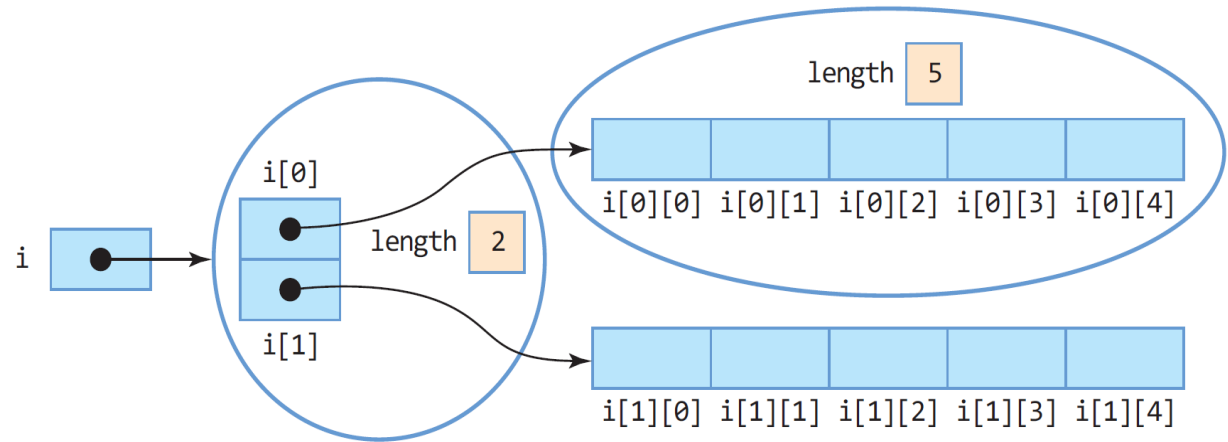
- Initialization
  - `int intArray[][] = new int[2][5];`
  - `char charArray[][] = new char[5][5];`
  - `double doubleArray[][] = new double[5][2];`
- `int intArray[][] = {{0,1,2},{3,4,5},{6,7,8}};`
- `char charArray[][] = {{'a','b','c'}, {'d','e','f'}};`
- `double doubleArray[][] = {{0.01,0.02},{0.03,0.04}};`



## 2 dimensional array (how it looks like)

- Example

```
int i[][] = new int[2][5];  
int size1 = i.length; // 2  
int size2 = i[0].length; // 5  
int size3 = i[1].length; // 5
```



- length

- `i.length` -> 2차원 배열의 행의 개수로서 2
- `i[n].length`는 n번째 행의 열의 개수
  - `i[0].length` -> 0번째 행의 열의 개수로서 5
  - `i[1].length` -> 1번째 행의 열의 개수로서 5

## Practice 4

- Construct 9 x 9 multiplication table with 2 dimensional array
- Print out 9 x 9 multiplication table
- $\text{array}[i][j] \rightarrow (i+1)*(j+1)$

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
		9						
			16					
				25				
					36			
						49		
							64	
								81



# Array

- Character Array can represent String
  - Practice the followings and feel its difficulty
    - Make two arrays of “hello” “world”
      - `a[] = {'h','e','l','l','o'};`
      - `b[] = {'w','o','r','l','d'};`
    - Make another word by concatenating a[] and b[]
      - `c[] = {'h','e','l','l','o',' ','w','o','r','l','d'};`
    - Remove white space
      - `c[] = {'h','e','l','l','o','w','o','r','l','d',' '};`
    - Print out the string

# Summary

- Array
  - 1 dimensional array
  - Iteration
  - 2 dimensional array
  - Practice