



*Power Java*

## 제10장 배열



© 2009 인피니티북스 All rights reserved



## 이번 장에서 학습할 내용

- 배열의 선언
- 배열의 사용
- 배열과 메소드
- 객체들의 배열
- 2차원 배열
- 정렬과 탐색

자바에서  
배열은 객체로  
취급됩니다.



© 2009 인피니티북스 All rights reserved



## 배열의 개념

- 배열(array): 같은 타입의 변수들의 모임

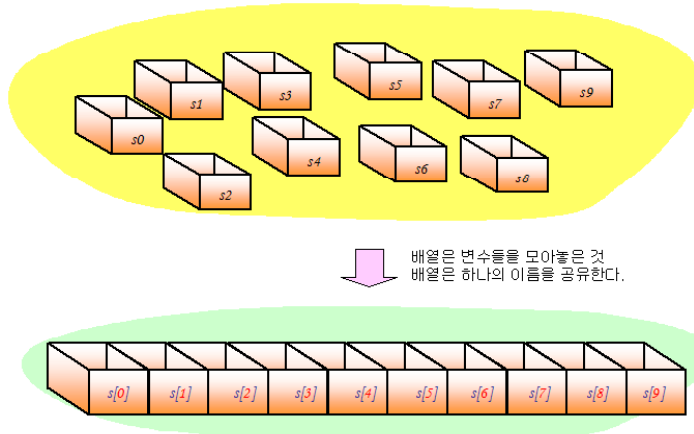


그림 10.1 배열은 변수들의 모임이다.

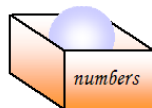
© 2009 인피니티박스 All rights reserved



## 배열을 만드는 절차

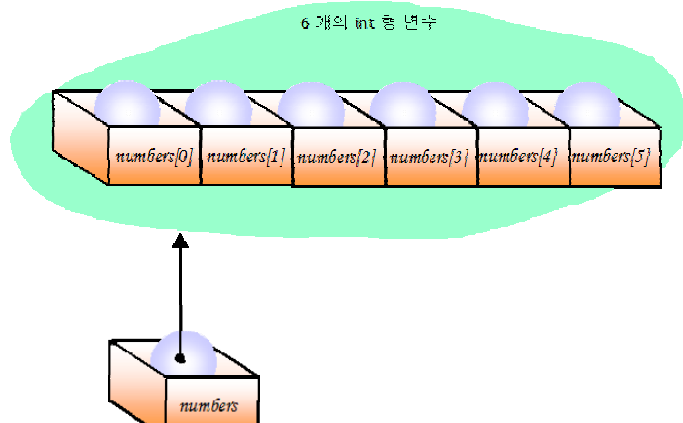
- 먼저 배열 참조 변수부터 선언

`int[] numbers;` // 배열 참조 변수 선언



- 배열을 new 연산자를 사용하여 생성

`numbers = new int[6];` // 배열 객체 생성



© 2009 인피니티박스 All rights reserved



## 배열 예제

ArrayTest1.java

```
import java.util.Scanner;

public class ArrayTest1 {
    public static void main(String[] args) {
        int[] salary = new int[2];
        Scanner scan = new Scanner(System.in);
        System.out.print("직원1의 월급을 입력하시오: ");
        salary[0] = scan.nextInt();
        System.out.print("직원2의 월급을 입력하시오: ");
        salary[1] = scan.nextInt();
        System.out.println("직원1의 월급은 " + salary[0]);
        System.out.println("직원2의 월급은 " + salary[1]);
    }
}
```

© 2009 인피니티박스 All rights reserved



## 또 다른 배열 선언 방법

- `int[] values;` // ① 자바 방식
- `int values[];` // ② C언어 유사 방식



© 2009 인피니티박스 All rights reserved



## 중간 점검 문제

1. `int`형의 100개의 원소를 가지고 `array`로 참조되는 배열을 생성하는 문장을 써라.
2. 10개의 원소를 가지는 배열에서 올바른 인덱스 값의 범위는?
3. 만약 배열의 인덱스가 올바르지 않으면 어떻게 되는가?
4. { 1.2, 3.1, 6.7 }의 값으로 초기화되는 `double`형 배열을 생성하는 문장을 작성하시오.

© 2009 인피니티박스 All rights reserved



## 배열의 사용

```
import java.util.Scanner;

public class ArrayTest4 {
    public static void main(String[] args) {
        final int STUDENTS = 5;
        int total = 0;
        Scanner scan = new Scanner(System.in);
        int[] scores = new int[STUDENTS];
        for (int i = 0; i < STUDENTS; i++) {
            System.out.print("성적을 입력하시오:");
            scores[i] = scan.nextInt();
        }
        for (int i = 0; i < STUDENTS; i++)
            total += scores[i];
        System.out.println("평균 성적은" + total / STUDENTS + "입니다");
    }
}
```

© 2009 인피니티박스 All rights reserved



## 배열의 사용

실행결과

```
성적을 입력하시오:10
성적을 입력하시오:20
성적을 입력하시오:30
성적을 입력하시오:40
성적을 입력하시오:50
평균 성적은30입니다
```

© 2009 인피니티박스 All rights reserved



## 배열의 크기

```
for(int i=0; i < numbers.length; i++)
    numbers[i] = (int)(Math.random()*1000);
```

배열의 크기는  
length라는  
필드로 알 수  
있습니다.



© 2009 인피니티박스 All rights reserved



## for-each 루프

```

for (자료형 변수 : 배열이름)
{
    //반복 문장들
}

```

Numbers.java

```

public class Numbers {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        for (int i = 0; i < numbers.length; i++)
            numbers[i] = (int) (Math.random()*1000);
        for (int value : numbers)
            System.out.println(value);
    }
}

```

실행결과

```

688
773
94
691
349

```

© 2009 인피니티박스 All rights reserved



## 예제

Strings2.java

```

public class Strings2 {
    public static void main(String[] args) {
        String[] strings = { "Java", "C", "C++" };
        for (String s : strings)
            System.out.println(s);
    }
}

```

실행결과

```

Java
C
C++
null
null

```

© 2009 인피니티박스 All rights reserved



## 사용자가 배열의 크기를 지정

```
import java.util.Scanner;

public class ScoreTest {
    public static void main(String[] args) {
        int total = 0;
        int size;
        Scanner scan = new Scanner(System.in);
        System.out.print("배열의 크기를 입력하시오:");
        size = scan.nextInt();
        int[] scores = new int[size];
        for (int i = 0; i < scores.length; i++) {
            System.out.print("성적을 입력하시오:");
            scores[i] = scan.nextInt();
        }
        for (int i = 0; i < scores.length; i++)
            total += scores[i];
        System.out.println("평균 성적은" + total / scores.length + "입니다");
    }
}
```

© 2009 인피니티박스 All rights reserved



## 중간 점검 문제

1. int형 정수를 저장하고 있는 배열 array의 모든 원소의 값을 두배로 만드는 반복 루프를 작성하라.
2. 사용자에게 배열의 크기를 받아서 double형 배열을 생성하는 문장을 작성하라.
3. for-each와 전통적인 for 루프를 비교하라.
4. 하나의 배열을 다른 배열로 복사하는 반복 루프를 작성하라.

© 2009 인피니티박스 All rights reserved



## 배열을 메소드의 매개 변수로 전달

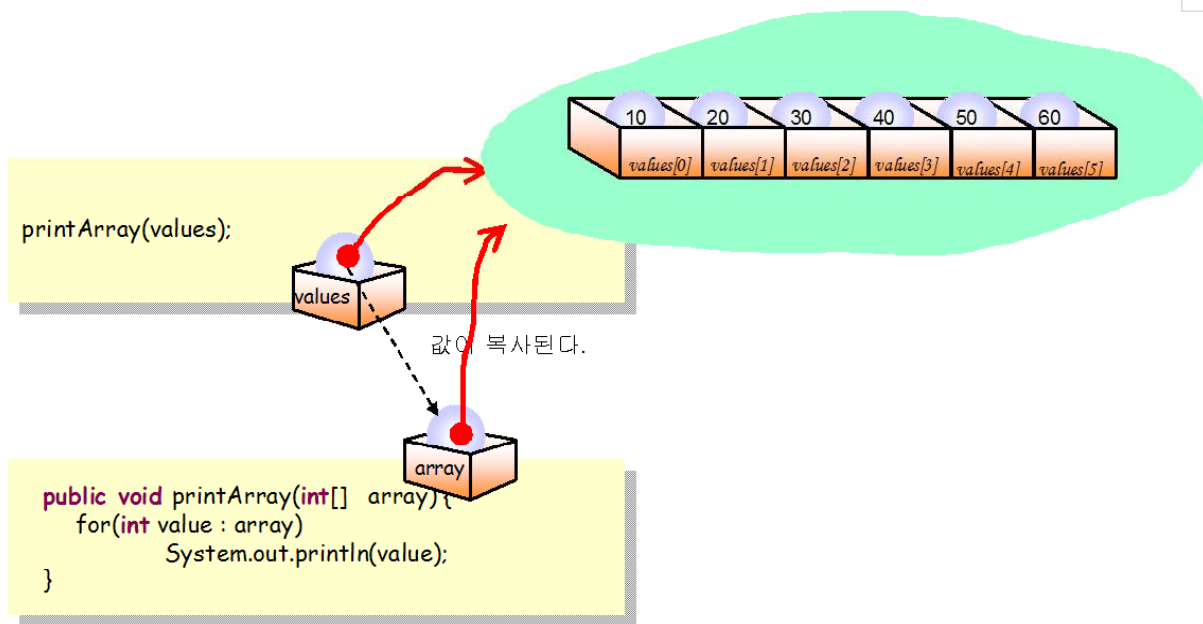


그림 10.2 메소드와 배열



## 예제

```
import java.util.Scanner;

public class ScoreTest1 {
    final static int STUDENTS = 5;

    public static void main(String[] args) {
        int[] scores = new int[STUDENTS];
        getValues(scores);
        getAverage(scores);
    }

    private static void getValues(int[] array) {
        Scanner scan = new Scanner(System.in);
        for (int i = 0; i < array.length; i++) {
            System.out.print("성적을 입력하시오:");
            array[i] = scan.nextInt();
        }
    }
}
```





## 예제

```

private static void getAverage(int[] array) {
    int total = 0;
    for (int i = 0; i < array.length; i++)
        total += array[i];
    System.out.println("평균 성적은 " + total / array.length + "입니다");
}

```

### 실행결과

```

성적을 입력하시오:10
성적을 입력하시오:20
성적을 입력하시오:30
성적을 입력하시오:40
성적을 입력하시오:50
평균 성적은 30입니다

```

© 2009 인피니티박스 All rights reserved



## 메소드의 반환값으로 배열 반환

```

import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        int[] array;
        array = getData();
        printData(array);
    }

    private static int[] getData() {
        int[] testData = { 10, 20, 30, 40, 50 };
        return testData;
    }

    private static void printData(int[] array) {
        for (int i = 0; i < array.length; i++)
            System.out.println(array[i]);
    }
}

```



## 중간 점검 문제

1. 배열이 메소드로 전달되면 어떤 값이 전달되는 것인가?
2. 배열 원소가 전달되는 경우와 배열 전체가 전달되는 경우를 비교하여 차이점을 설명하라.
3. int형 배열을 전달받아서 배열의 각 원소를 하나 증가시키는 메소드를 작성하시오.

© 2009 인피니티박스 All rights reserved



## 객체들의 배열

- 객체들의 배열에서는 객체에 대한 참조값만을 저장  
`Car[] cars = new Car[5];`

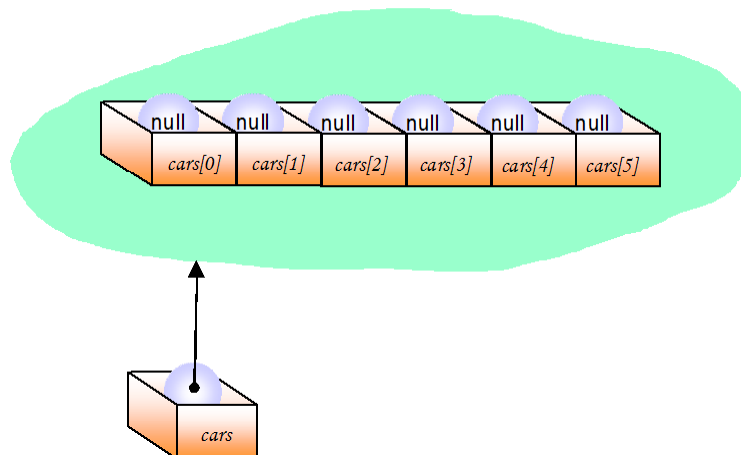


그림 10.3 객체들의 배열

© 2009 인피니티박스 All rights reserved



## 객체들의 배열

- 각 원소에 들어가는 객체는 따로 생성하여야 한다.

```
cars[0] = new Cars();
```

```
cars[1] = new Cars();
```

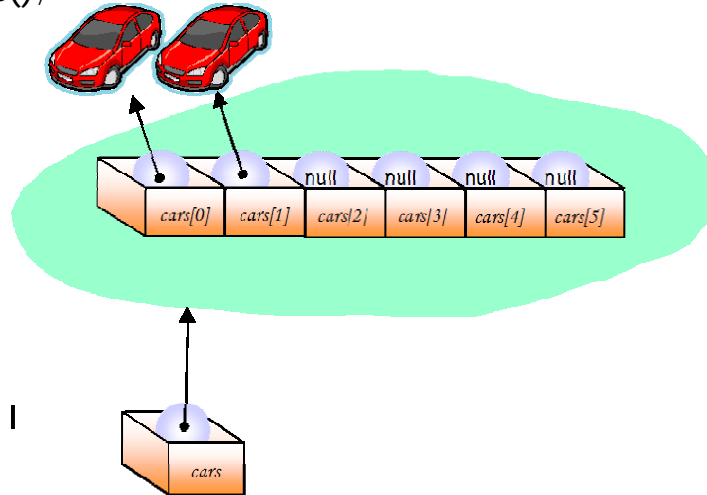


그림 10.4 객체들의 배열은 사실 참조값만을 저장한다.



## 예제

CarTest.java

```
import java.util.Scanner;
class Car {
    public int speed // 속도
    public int mileage // 주행거리
    public String color // 색상

    public Car() {
        speed = mileage = 0;
        color = "red";
    }
    public void speedUp() { // 속도 증가 메소드
        speed += 10;
    }
    public String toString() { // 객체의 상태를 문자열로 반환하는 메소드
        return "속도: " + speed + " 주행거리: " + mileage + " 색상: " + color;
    }
}
```



## 예제

```

public class CarTest {
    public static void main(String[] args) {
        final int NUM_CARS = 5;
        Car[] cars = new Car[NUM_CARS];
        for (int i = 0; i < cars.length; i++)
            cars[i] = new Car();
        for (int i = 0; i < cars.length; i++)
            cars[i].speedUp();
        for (int i = 0; i < cars.length; i++)
            System.out.println(cars[i]);
    }
}

```

실행결과

```

속도: 10 주행거리: 0 색상: red
속도: 10 주행거리: 0 색상: red
속도: 10 주행거리: 0 색상: red
속도: 10 주행거리: 0 색상: red
속도: 10 주행거리: 0 색상: red

```

© 2009 인피니티박스 All rights reserved



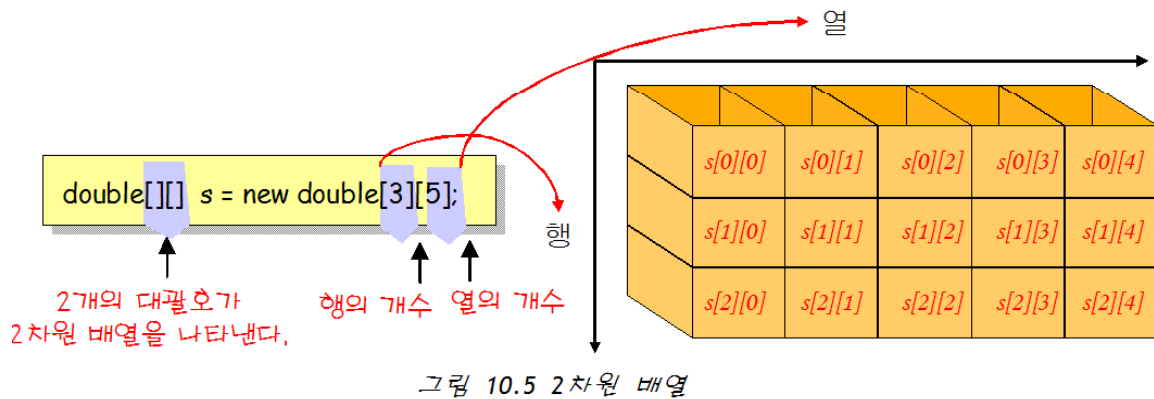
## 중간 점검 문제

1. 은행 계좌를 나타내는 클래스 `BankAccount`를 가정하자. `BankAccount`는 잔액만을 가지고 있다. 3개의 원소를 가지는 `BankAccount`의 배열을 생성하고 각 배열 원소를 `BankAccount` 객체로 초기화하라.
2. 객체들의 배열을 메소드로 전달하면 무엇이 전달되는가?

© 2009 인피니티박스 All rights reserved



## 2차원 배열



© 2009 인피니티박스 All rights reserved



### Rainfall.java

```
import java.util.Scanner;

public class Rainfall {
    public static void main(String[] args) {
        final int YEARS = 3;
        final int QUARTERS = 4;
        double[][] rain = new double[YEARS][QUARTERS];
        Scanner scan = new Scanner(System.in);

        for (int y = 0; y < YEARS; y++){
            for (int q = 0; q < QUARTERS; q++) {
                System.out.print(y + "차년도" + q + "분기 강수량: ");
                rain[y][q] = scan.nextDouble();
            }
        }

        for (int y = 0; y < YEARS; y++) {
            double total = 0.0;
            for (int q = 0; q < QUARTERS; q++) {
                total += rain[y][q];
            }
            System.out.println(y + "차년도 강수량은 " + total);
        }
    }
}
```



## 2차원 배열의 초기화와 크기

ArrayTest.java

```
import java.util.Scanner;

public class ArrayTest {
    public static void main(String[] args) {
        int[][] array = { { 10, 20, 30, 40 }, { 50, 60, 70, 80 },
                          { 90, 100, 110, 120 } };

        for (int r = 0; r < array.length; r++) {
            for (int c = 0; c < array[r].length; c++) {
                System.out.println(r + "행" + c + "열:" + array[r][c]);
            }
        }
    }
}
```

실행결과

```
0행0열:10
0행1열:20
...
2행2열:110
2행3열:120
```



## 2차원 배열 매개 변수

ArrayTest.java

```
import java.util.Scanner;

public class ArrayTest {
    public static void main(String[] args) {
        int[][] array = { { 10, 20, 30, 40 }, { 50, 60, 70, 80 },
                          { 90, 100, 110, 120 } };
        printArray(array);
    }

    public static void printArray(int[][] array) {
        for (int r = 0; r < array.length; r++) {
            for (int c = 0; c < array[r].length; c++) {
                System.out.println(r + "행" + c + "열:" + array[r][c]);
            }
        }
    }
}
```



## 배열의 응용: 정렬

- 선택 정렬: 최소값을 정렬되지 않은 첫번째 원소와 교환



그림 10.8 선택 정렬의 과정

© 2009 인피니티박스 All rights reserved



## 선택 정렬 코드

```
public static void selectionSort(int[] list) {
    int temp, least;

    for (int i = 0; i < list.length - 1; i++) {
        least = i;
        for (int j = i + 1; j < list.length; j++)
            // 최소값 탐색
            if (list[j] < list[least])
                least = j;
        // i번째 원소와 least 위치의 원소를 교환
        temp = list[i];
        list[i] = list[least];
        list[least] = temp;
    }
}
```

© 2009 인피니티박스 All rights reserved



## 배열의 응용: 이진 탐색

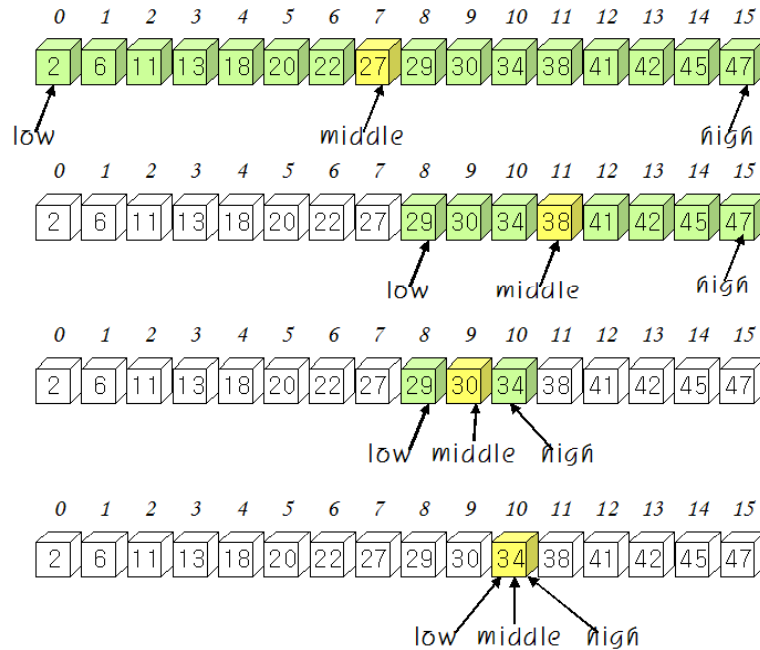


그림 10.11 이진탐색의 구체적인 예: 34를 탐색한다.

© 2009 인피니티박스 All rights reserved



## 이진 탐색

BinarySearch.java

```
import java.util.Scanner;

public class BinarySearch {
    public static void main(String[] args) {
        int[] data = { 10, 20, 30, 40, 50, 60, 70, 80 };
        selectionSort(data);
        int retValue = binarySearch(data, 60);
        if (retValue != -1)
            System.out.println("위치 " + retValue + " 에서 발견");
        else
            System.out.println("탐색 실패");
    }

    public static void selectionSort(int list[]) {
        ...//앞의 코드 참조
    }
}
```

© 2009 인피니티박스 All rights reserved





## 이진 탐색

```
public static int binarySearch(int[] list, int key) {  
    int low, high, middle;  
  
    low = 0;  
    high = list.length - 1;  
  
    while (low <= high) { // 아직 숫자들이 남아있으면  
        middle = (low + high) / 2; // 중간 요소 결정  
        if (key == list[middle]) // 일치하면 탐색 성공  
            return middle;  
        else if (key > list[middle]) // 중간 원소보다 크다면  
            low = middle + 1; // 새로운 값으로 low 설정  
        else  
            high = middle - 1; // 새로운 값으로 high 설정  
    }  
    return -1;  
}
```

© 2009 인피니티박스 All rights reserved



## 중간 점검 문제

1. 선택 정렬에서 가장 작은 값은 몇 번이나 교환되는가?
2. 1024개의 원소가 있는 배열을 이진 탐색할 때 최대 몇 번의 비교가 필요한가?
3. 순차 탐색과 이진 탐색을 비교하라. 순차 탐색이란 배열의 첫 번째 원소부터 순차적으로 비교하여 탐색하는 알고리즘이다.

© 2009 인피니티박스 All rights reserved



## Q & A

