



3장 자바 프로그래밍의 기본-배열

남 광 우

배열 (Arrays)

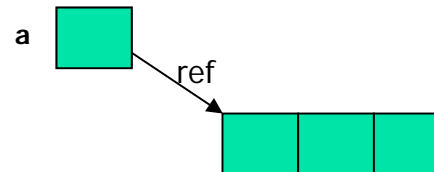
- 자바에서의 배열은 1급 클래스 객체
 - First Class Objects,
 - 즉, 데이터 타입이라기 보다 클래스로서 동작

□ 배열의 사용 예

C
int a[3];



JAVA
int[] a=new int[3];



□ 배열의 경쟁자들

- Vector 클래스
- Collection 클래스
- Map 클래스
- ArrayList 클래스



배열 클래스의 선언

□ 기본 데이터 타입을 이용한 배열 클래스

- byte[]
- int[]
- long[]
- float[]
- double[]
- char[]

Examples

```
int[]    studentPoints;  
double[] balances;  
char[]   grades;
```

□ 클래스를 이용한 배열 클래스

- String[]
- Image[]

Examples

```
String[]  usernames;  
Integer[] studentPoints;
```

배열의 생성

□ 배열의 생성 방법

- new 를 사용하여 배열의 크기 등을 설정하여 생성

```
int[] arrayOfInt = new int[100];
```

```
for(int i=0; i<100; i++)  
{  
    arrayOfInt[i] = i;  
}
```

- new를 통하여 생성되지 않은 배열은 사용할 수 없음

```
int[] arrayOfInt = null;
```

```
arrayOfInt[3] = 100; (X)
```



배열 생성시 초기화

□ 배열의 값 할당 방법

- 배열을 생성하는 동시에 초기화
- 배열을 생성한 후 배열 내의 변수에 하나씩 값 할당

□ 배열의 선언과 초기화를 동시에 하는 방법

```
// 크기 5의 arrayOfInt1을 생성하고 초기화  
int[] arrayOfInt1 = new int[] {1,2,3,4,5}
```

```
// 크기 5의 arrayOfInt1을 생성하고 초기화  
int[] arrayOfInt2 = {6,7,8,9,10}
```

```
// 크기 3의 String 배열의 초기화  
String[] titles = {"이름", "주소", "전화번호"}
```

배열 선언의 좀 더 많은 예

□ 임의 배열(anonymous array)

- 변수 이름 없이 배열을 선언하여 사용하는 경우
- 특히, 메소드에 배열을 파라미터로 전달할 필요가 있으나 지역 변수를 만들고 싶지 않을 때 사용

```
String[] titles = {"이름", "주소", "전화번호"}  
printTitles( titles );
```

```
// 위의 배열 사용을 다음과 같이 한 줄로 쓸수 있음  
printTitles( new String[] {"이름", "주소", "전화번호"} )
```

□ 배열의 길이

- 배열이름.length 의 형태로 쉽게 배열길이 파악 가능

```
for(int i=0; i< titles.length; i++)  
    System.out.println(titles[i]);
```

배열의 복사

□ 배열값의 복사

- System.arraycopy 메소드를 사용

// System.arraycopy의 형식

`System.arraycopy(from[], fromIndex, to[], toIndex, count)`

□ 배열값 복사의 예

```
int[] smallPrimes={2,3,5,7,11,13};  
int[] numbers={100, 101, 102, 103, 104, 105, 106}
```

`System.arraycopy(smallPrimes, 2, numbers, 3, 3);`

배열값의 출력

```
for(int i= 0; i<smallPrimes.length;i++)  
    System.out.print( smallPrimes[i] + " ");
```

결과

```
smallPrimes => {2, 3, 5, 7, 11, 13}  
numbers => {100, 101, 102, 5, 7, 11, 106}
```



배열의 활용(Arrays)

□ 배열값의 복사

- Arrays.copyOf() 메소드를 사용

```
int[] smallPrimes={2,3,5,7,11,13};  
int[] newCopy = Arrays.copyOf( smallPrimes, 5);
```

□ 배열값의 정렬

```
int[] numbers={110, 105, 107, 103, 104, 109, 106}
```

```
Arrays.sort( numbers );  
// numbers = {103, 104, 105, 106, 107, 109, 110}
```

□ 배열값에서 이진탐색

```
int pos1 = Arrays.binarySearch( numbers, 103);  
int pos2 = Arrays.binarySearch( numbers, 108);
```

```
System.out.println(" Search Result1 = "+pos1 + " Result2:" + pos2);
```


배열의 참조값 사용

□ 동일 배열에 대한 참조변수

- 한 배열에 대하여 두 개의 이상의 배열 변수로 참조하는 경우
- 배열의 참조값 할당에 의해 사용

□ 동일 배열의 참조 변수 사용 예

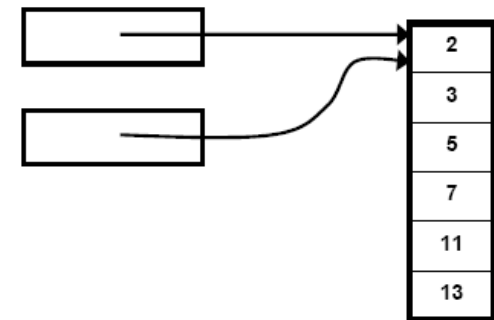
```
int[] smallPrimes={2,3,5,7,11,13};  
int[] smallPrimesRef;
```

```
smallPrimesRef = smallPrimes;  
smallPrimesRef[3] = 100;  
smallPrimesRef[4] = 101;
```

결과

```
smallPrimes => {2, 3, 5, 100, 101, 13}  
smallPrimesRef => {2, 3, 5, 100, 101, 13}
```

**smallPrimes와 smallPrimesRef
두 변수는 동일 배열을 가르키
고 있음!!**



배열의 Clone 생성

□ clone의 사전적 의미

- 네이버 백과사전 : 단일세포 또는 개체로부터 무성적(無性的)인 증식에 의하여 생긴 **유전적으로 동일한 세포군 또는 개체군**
- 배열의 clone : 동일 크기, 동일 값을 갖는 복제 배열을 생성
 - System.arraycopy : 배열 값을 다른 배열에 복사하는 것

□ clone의 사용 예

```
int[] smallPrimes={2,3,5,7,11,13};  
int[] smallPrimesRef;
```

```
smallPrimesRef = (int[])smallPrimes.clone();  
smallPrimesRef[3] = 100;  
smallPrimesRef[4] = 101;
```

**smallPrimes와 smallPrimesRef
두 변수는 서로 다른 배열!**

결과

```
smallPrimes => {2, 3, 5, 7, 11, 13}  
smallPrimesRef => {2, 3, 5, 100, 101, 13}
```

메소드 매개변수로서의 배열

□ 배열 매개변수의 사용

- 배열의 변수 명을 넘기는 것으로 배열을 다른 메소드로 전달
- 배열의 값만 넘기고 싶을 경우에는 clone()을 사용

□ 배열 매개변수의 사용 예

```
public class ArrayMethodTest
{
    public static void print( int[] printTarget )
    {
        for(int i=0; i<printTarget.length; i++)
            System.out.print( printTarget[i] + " ");
        System.out.println()
    }

    public static void main(String[] args)
    {
        int[] randomNumbers = new int[10];
        for( int i = 0; i<randomNumbers.length ; i++)
            randomNumbers[i] = (int)(Math.random()*100);
        print( randomNumbers ); // print( a.clone() ) 을 사용한다면?
    }
}
```

메소드 반환값으로서의 배열

□ 메소드 반환값으로 배열 반환

- 메소드의 반환값으로 배열 타입을 선언

```
public static int[] getLottoValues(int high, int number)
```

□ 배열을 반환하는 메소드의 예

// 로또 번호 추출기(high:최대값(예:45), number:갯수(예:6))

```
public static int[] getLottoValues(int high, int number)
{
    int i;
    int[] numbers = new int[high];
    int[] luckyNumbers = new int[number];
    // numbers 배열을 1, 2, 3, ..., high 값으로 채움
    for (i=1; i<= high; i++) numbers[i] = i;
    // luckyNumbers 추출
    for (i=0; i < number; i++)
    {
        int j = (int)((Math.random() *high):
        luckyNumbers[i] = numbers[j];
        numbers[j] = numbers[high];
        high--;
    }
    return luckyNumbers;
}
```

```
public static main(... )
{
    ....
    int[] luckyValues;

    luckyValues=getLottoValues(45, 6);
    print(luckyValues);
}
```



실습 예제

- 실습 프로그램 : 로또 번호 추출기
- 입력
 - 로또 번호 최대 값 : 37 ??
 - 추출할 로또 번호 갯수 : 5???
- 출력
 - 추출한 로또 번호
- 추가 요구사항 : 옵션
 - 추출한 로또 번호를 정렬해서 출력
 - JOptionPane.showMessageDialog를 사용해서 출력!
 - 예 : JOptionPane.showMessageDialog(null, “테스트 “);