

Arrays 클래스

Arrays 클래스

❖ Arrays

- 배열 조작 기능을 가지고 있는 클래스 - 배열 복사, 항목 정렬, 항목 검색
- 제공하는 정적 메소드

리턴타입	메소드 이름	설명
int	binarySearch(배열, 찾는값)	전체 배열 항목에서 찾는값이 있는 인덱스 리턴
타겟배열	copyOf(원본배열, 복사할길이)	원본배열의 0 번 인덱스에서 복사할 길이만큼 복사한 배열 리턴, 복사할 길이는 원본배열의 길이보다 크도 되며, 타겟배열의 길이가 된다.
타겟배열	copyOfRange(원본배열, 시작인덱스, 끝인덱스)	원본배열의 시작인덱스에서 끝인덱스까지 복사한 배열 리턴
boolean	deepEquals(배열, 배열)	두 배열의 깊은 비교(중첩 배열의 항목까지 비교)
boolean	equals(배열, 배열)	얕은 비교(중첩 배열의 항목은 비교하지 않음)
void	fill(배열, 값)	전체 배열 항목에 동일한 값을 저장
void	fill(배열, 시작인덱스, 끝인덱스, 값)	시작인덱스부터 끝인덱스까지의 항목에만 동일한 값을 저장
void	sort(배열)	배열의 전체 항목을 올림차순으로 정렬

Arrays 클래스

❖ 배열 복사

- `Arrays.copyOf(원본배열, 복사할 길이)`
 - `0 ~ (복사할 길이-1)`까지 항목 복사
 - 복사할 길이는 원본 배열의 길이보다 커도 되며 타겟 배열의 길이

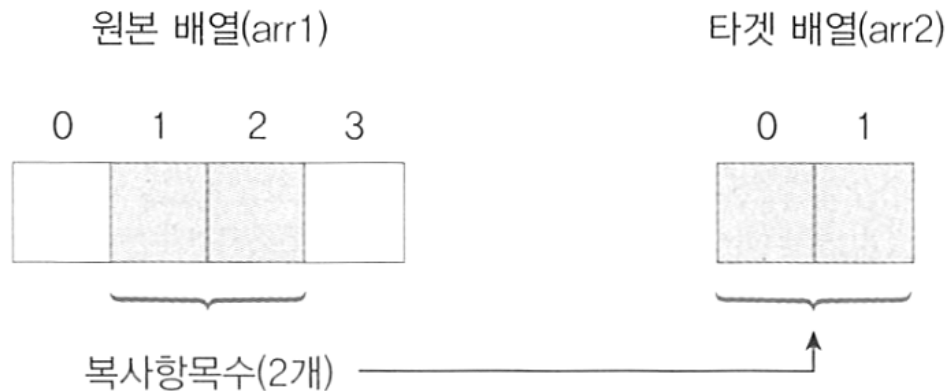
```
char[] arr1 = {'J', 'A', 'V', 'A'};  
char[] arr2 = Arrays.copyOf(arr1, arr1.length);
```

Arrays 클래스

❖ 배열 복사

- copyOfRange(원본 배열, 시작 인덱스, 끝 인덱스)
 - 시작인덱스 ~ (끝 인덱스-1)까지 항목 복사

```
char[] arr1 = {'J', 'A', 'V', 'A'};  
char[] arr2 = Arrays.copyOfRange(arr1, 1, 3);
```



Arrays 클래스

❖ 배열 복사

- `System.arraycopy()`

<code>System.arraycopy(Object src,</code>	<code>int srcPos,</code>	<code>Object dest,</code>	<code>int destPos,</code>	<code>int length)</code>
원본배열	원본시작인덱스	타겟배열	타겟시작인덱스	복사개수

```
System.arraycopy(arr1, 1, arr2, 3, 2);
```



Arrays 클래스

❖ 배열복사 : ArrayCopyExample.java

```
import java.util.Arrays;

public class ArrayCopyExample {
    public static void main(String[] args) {
        char[] arr1 = { 'J', 'A', 'V', 'A' };

        // 방법1
        char[] arr2 = Arrays.copyOf(arr1, arr1.length);
        System.out.println(Arrays.toString(arr2));

        // 방법2
        char[] arr3 = Arrays.copyOfRange(arr1, 1, 3);
        System.out.println(Arrays.toString(arr3));

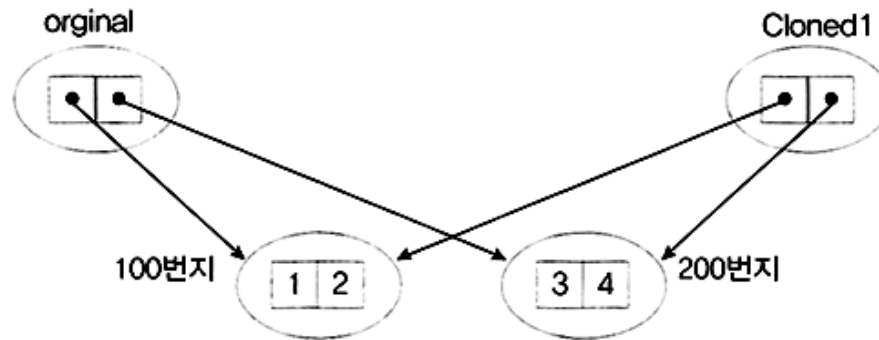
        // 방법3
        char[] arr4 = new char[arr1.length];
        System.arraycopy(arr1, 0, arr4, 0, arr1.length);
        for (int i = 0; i < arr4.length; i++) {
            System.out.println("arr4[" + i + "]= " + arr4[i]);
        }
    }
}
```

Arrays 클래스

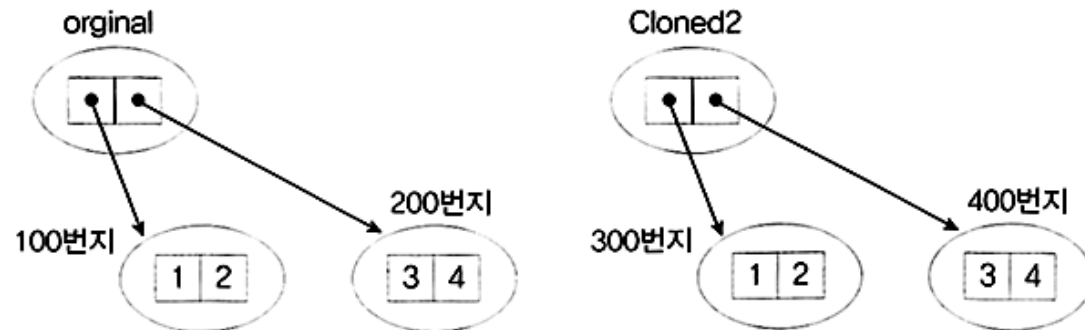
❖ 배열 항목 비교

- `Arrays.equals(배열, 배열)` - 1차 항목의 값만 비교
- `Arrays.deepEquals(배열, 배열)` - 중첩된 배열의 항목까지 비교

[얕은 복사]



[깊은 복사]



Arrays 클래스

❖ EqualsExample.java

```
import java.util.Arrays;

public class EqualsExample {
    public static void main(String[] args) {
        int[][] original = { { 1, 2 }, { 3, 4 } };

        // 얕은 복사후 비교
        System.out.println("[얕은 복제후 비교]");
        int[][] cloned1 = Arrays.copyOf(original, original.length);

        System.out.println("배열 번지 비교: " + original.equals(cloned1));
        System.out.println("1차 배열 항목값 비교: " +
            Arrays.equals(original, cloned1));
        System.out.println("중첩 배열 항목값 비교: " +
            Arrays.deepEquals(original, cloned1));
    }
}
```


Arrays 클래스

❖ EqualsExample.java

```
// 깊은 복사후 비교
System.out.println("\n[깊은 복제후 비교]");
int[][] cloned2 = Arrays.copyOf(original, original.length);

cloned2[0] = Arrays.copyOf(original[0], original[0].length);
cloned2[1] = Arrays.copyOf(original[1], original[1].length);

System.out.println("배열 번지 비교: " + original.equals(cloned2));
System.out.println("1차 배열 항목값 비교: " +
    Arrays.equals(original, cloned2));
System.out.println("중첩 배열 항목값 비교: " +
    Arrays.deepEquals(original, cloned2));
    }
}
```

Arrays 클래스

❖ 배열 항목 정렬

- `Arrays.sort(배열)` - 항목 오름차 순으로 정렬
 - 기본 타입이거나 `String` 배열 자동 정렬
- 사용자 정의 클래스 배열은 `Comparable` 인터페이스를 구현해야만 정렬

Arrays 클래스

❖ Comparable 구현 클래스 : Member.java

```
public class Member implements Comparable<Member> {  
    String name;  
  
    Member(String name) {  
        this.name = name;  
    }  
  
    @Override  
    public int compareTo(Member o) {  
        return name.compareTo(o.name);  
    }  
}
```

Arrays 클래스

❖ 배열 정렬 : SortExample.java

```
public class SortExample {  
    public static void main(String[] args) {  
        int[] scores = { 99, 97, 98 };  
  
        Arrays.sort(scores);  
  
        for (int i = 0; i < scores.length; i++) {  
            System.out.println("scores[" + i + "]=" + scores[i]);  
        }  
        System.out.println();  
  
        String[] names = { "홍길동", "박동수", "김민수" };  
        Arrays.sort(names);  
  
        for (int i = 0; i < names.length; i++) {  
            System.out.println("names[" + i + "]=" + names[i]);  
        }  
        System.out.println();  
    }  
}
```

Arrays 클래스

❖ 배열 정렬 : SortExample.java

```
Member m1 = new Member("홍길동");  
Member m2 = new Member("박동수");  
Member m3 = new Member("김민수");
```

```
Member[] members = { m1, m2, m3 };  
Arrays.sort(members);
```

```
for (int i = 0; i < members.length; i++) {  
    System.out.println("members[" + i + "].name=" + members[i].name);  
}  
}
```

Arrays 클래스

❖ 배열 항목 검색

- 특정 값 위치한 인덱스 얻는 것
- `Arrays.sort(배열)`로 먼저 정렬
- `Arrays.binarySearch(배열, 찾는 값)` 메소드로 항목을 찾아야

Arrays 클래스

❖ 배열 검색 : SearchExample.java

```
import java.util.Arrays;

public class SearchExample {
    public static void main(String[] args) {
        // 기본 타입값 검색
        int[] scores = { 99, 97, 98 };
        Arrays.sort(scores);
        int index = Arrays.binarySearch(scores, 99);
        System.out.println("찾은 인덱스: " + index);

        // 문자열 검색
        String[] names = { "홍길동", "박동수", "김민수" };
        Arrays.sort(names);
        index = Arrays.binarySearch(names, "홍길동");
        System.out.println("찾은 인덱스: " + index);
    }
}
```

Arrays 클래스

❖ 배열 검색 : SearchExample.java

```
// 객체 검색
Member m1 = new Member("홍길동");
Member m2 = new Member("박동수");
Member m3 = new Member("김민수");
Member[] members = { m1, m2, m3 };

Arrays.sort(members);
index = Arrays.binarySearch(members, m1);

System.out.println("찾은 인덱스: " + index);
    }
}
```