



개발 스터디 자바기초 #3



자바 스터디 운영을 위해 작성하는 문서입니다.

2022.06.18 작성완료

작성자 : daria ([git](#)) ([velog](#))

▼ 짧은 질문시간!

- 배열이란?
- if문과 switch&case의 다른 점은?
- break의 용도는?
- continue의 용도는?
- for문의 용도는?
- for문의 필수요소는?

▼ 숙제 풀이

2022.06.12 실습 예제 답

▼ List



```
ArrayList<String> list = new ArrayList<>();
```

- 지난주에 배운 배열(Array)의 단점을 보완하기 위해 나온 자료구조
- 배열의 크기를 지정하지 않고 선언이 가능하다.
- 삽입, 삭제 등의 기능을 제공한다.

▼ List vs Array

	List(ArrayList, LinkedList)	Array
삽입	add(), addAll()	X
삭제	remove(), removeAll()	X
크기	size()	length
데이터	class	기본자료형

List는

ArrayList 와 LinkedList로 나뉘지만 해당 과정은 심화과정임으로생략!

나중에 같이 공부해요 :) → 신입 기술면접 거의 필수 예제!

▼ 단점

당연히 조건없이 그냥 삽입이 가능할 리가 없다.

준비한 배열의 사이즈를 초과하면 배열을 더 큰 사이즈의 배열을 새로 만든다.

사이즈를 초과하는 add()는 많은 시간과 비용이 걸린다.

배열의 사이즈를 안다면, 처음에 초기 선언해주는 것이 더 효과적이다.

▼ 주요함수

▼ get()

```
list.get(1);  
//해당 하는 index로 데이터를 가져올 수 있다.
```

▼ add()

```
list.add("add");  
  
List<String>  
List<Long>  
  
List<String> list = new ArrayList<>();  
list.add("123");  
list.add("234");  
list.add("345");  
  
for(int i = 0; i<list.size(); i++){  
    System.out.println(list.get(i));  
}
```

▼ remove()

```
list.remove(0) //해당하는 인덱스의 값을 제거하고 값을("remove") 반환한다.  
list.remove("remove") //해당하는 값을 제거하고 true, false를 반환한다.
```

▼ contains()

```
list.contains("contains"); //해당하는 값이 있는지 true, false를 반환한다.
```

▼ sort()

```
list.sort(Comparator.naturalOrder()); list를 오름차순으로 정렬한다.  
list.sort(Comparator.reverseOrder()); list를 내림차순으로 정렬한다.
```

▼ String.join()

```
String.join(" ", list);
//각 요소에 "구분자"를 삽입하여 하나의 문자열로 만들 수 있다.
//"get, add, remove, contains, sort" 반환
```

▼ 예제

▼ ListTest1.java

```
package thirdday;

import java.util.ArrayList;
import java.util.Comparator;

public class ListTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>(5);
        for(int i = 0; i<5; i++){
            list.add(String.valueOf(i));
        }
        list.remove(0);
        list.remove("3");
        int index = 0;
        for(int i = 0; i<5; i++){
            if(list.contains(String.valueOf(i))){
                System.out.println("list 속에 "+ index+"번째에 들어있는 값:" + list.get(index++));
            }
        }
        System.out.println(list.size());

        list.sort(Comparator.reverseOrder());
        System.out.println("거꾸로 하면 이렇게: " + String.join(" ", list));
    }
}
```

```
/Users/user/Library/Java/JavaVirtualMachines/corretto-1.8.0_332/Contents/Home/bin/java ...
list 속에 들어있는 값:1
list 속에 들어있는 값:2
list 속에 들어있는 값:4
거꾸로 하면 이렇게: 4, 2, 1

Process finished with exit code 0
```

예제에서 알 수 있듯이 remove를 하면 index가 삭제한 인덱스로 채워진답니다.??

▼ Set



```
Set<String> set1 = new Set<>();
Set<String> set2 = new Set<>(Arrays.asList("H", "e", "l", "l", "o"));
```

- 순서와 중복이 없는 자료구조
- 삽입, 삭제 등의 기능을 제공한다.

▼ 주요함수

▼ add()

```
set1.add("add");
set1.addAll(new Set<>(Array.asList("addAll", "addAll2"));
```

▼ remove()

```
set2.remove("H");//순서가 없기 때문에 값으로만 삭제가 가능하다.
```

▼ retainAll()

```
Set<String> set1 = new Set<>(Arrays.asList("H", "e", "l", "l", "o");
Set<String> set2 = new Set<>(Arrays.asList("e", "l", "l", "o");
set2.retainAll(set2);// 교집합 수행 boolean 반환
```

▼ size()

```
Set<String> set1 = new Set<>(Arrays.asList("H", "e", "l", "l", "o");
set1.size();// 4반환, 중복을 허용하지 않기 때문
```

▼ 예제

▼ SetTest1.java

```
package thirdday;

import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class SetTest1 {
    public static void main(String[] args) {
        Set<Integer> set1 = new HashSet<>();
        set1.add(1);
        set1.add(2);
        set1.add(3);
        set1.add(4);
        set1.add(5);

        Set<Integer> set2 = new HashSet<>(Arrays.asList(3, 4, 5, 6));
        System.out.println(set2.remove(5));
        System.out.println(set2.remove(1));
    }
}
```

```
/Users/user/Library/Java/JavaVirtualMachines/corretto-1.8.0_332/Contents/Home/
set1는 set2을 모두 포함하고 있습니다.
set2에 set1에 포함되지 않은 값이 있습니다.
```

▼ Map



```
HashMap<Integer, String> map = new HashMap<>();
```

- key, value로 이루어져있는 자료구조
- 순서가 없다.
- 삽입, 삭제 등의 기능을 제공한다.
- key로 value을 가져올 수 있다.

▼ 주요함수

▼ put()

```
map.put(1, "1입니다.");  
map.put(2, "2입니다.");  
map.put(3, "3이라고 할 줄 알았지?");
```

▼ get()

```
map.get(1); //"1입니다."  
map.get(2); //"2입니다."  
map.get(3); //"3이라고 할 줄 알았지?"
```

▼ containsKey()

```
//key에 해당하는 값이 있는지 true, false 반환  
map.containsKey(1) //true  
map.containsKey(6) //false
```

▼ remove()

```
//key값에 해당하는 아이템(key, value)을 삭제한 후 그 value 값을 리턴한다.  
map.remove(3); // 3이라고 할 줄 알았지?
```

▼ keyset()

```
map.keySet();//[1, 2]  
//key값으로 이루어진 set 반환
```

▼ values()

```
map.values(); //"1입니다.", "2입니다."  
//value값으로 이루어진 set 반환
```

```
map.values(); //"1입니다.", "2입니다."  
//value값으로 이루어진 set 반환
```

▼ 예제

▼ MapTest1.java

```
package thirdday;

import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class MapTest1 {
    public static void main(String[] args) {
        Map<Integer, String> map= new HashMap<>(5);
        for(int i=0;i<5; i++){
            map.put(i+1, i+1+"입니다.");
        }
        System.out.println(map.remove(3));
        System.out.println(map.remove(7)); //없는 key이기 때문에 null값 반환
        Set<Integer> keyset = map.keySet();
        for(int i=0;i<keyset.size();i++){
            if(map.containsKey(i+1)){
                System.out.println(map.get(i+1));
            }else{
                System.out.println("삭제된 값입니다.");
            }
        }
        System.out.println(String.join(" : ", map.values()));
    }
}
```

```
/Users/user/Library/Java/JavaVirtualMachines/corretto-1.8.0_332/Contents/Home/bin/java ...
3입니다.
null
1입니다.
2입니다.
삭제된 값입니다.
4입니다.
1입니다. : 2입니다. : 4입니다. : 5입니다.

Process finished with exit code 0
```

▼ 객체 for(forEach)

```
for (T t : this) {
    action.accept(t);
}
```

- List나 Map 등을 순회할 때 사용

▼ 예제

```
package thirdday;

import java.util.ArrayList;
import java.util.List;
```

```
public class ForEachTest {  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<>(5);  
        for(int i = 1; i<6; i++){  
            list.add(String.valueOf(i));  
        }  
        for(String i : list){  
            System.out.println(i);  
        }  
    }  
}
```

```
/Users/user/Library/Java/JavaVirtualMachines/corretto-1.8.0_332/Contents/Home/bin/java ...  
1  
2  
3  
4  
5  
|
```

▼ 실습 예제

2022.06.19 실습 예제