

ArrayListTest > main



ArrayListAllowDuplicate



Git:



Test.java × ArrayListTest.java × ArrayListAllowDuplicate.java ×

```
4 public class ArrayListTest {
5     public static void main(String[] args) {
6         String[] location = {"거제도", "제주도", "오사카", "부산"};
7
8         ArrayList<String> locationsList = new ArrayList<>(Arrays.asList(location));
9         //1 ArrayList를 이용하려면 "ArrayList<String> list = new ArrayList<String>()" 형식으로 한다.
10        //2일반배열을 ArrayList로 바꾸려면 "Array.asList(배열이름)"을 이용한다.
11
12        for (int i = 0; i < 4; i++) {
13            System.out.printf("get(%d) = %s\n", i, locationsList.get(i));
14            //3 ArrayList에서 값을 얻어서 출력하려면 "get(배열인덱스값)"을 이용한다.
15        }
16        System.out.println("복제 이후!");
17
18        ArrayList<String> clone = (ArrayList<String>) locationsList.clone();
19        //4 ArrayList에서 값을 복제하려면 "clone()"을 이용한다.
20
21        for (int i = 0; i < 4; i++) {
22            System.out.printf("get(%d) = %s\n", i, clone.get(i));
23        }
24        locationsList.remove(0, "거제도");
25        //5 ArrayList에서 값을 지우려면 "remove()"을 이용한다.
26
27        System.out.println("locationsList의 거제도 삭제 이후 clone 다시 보기");
28        for (int i = 0; i < 4; i++) {
29            System.out.printf("get(%d) = %s\n", i, clone.get(i));
30        }
31    }
32 }
```

"Kotlin" plugin update

Update

Plugin Setting

tTest > main



ArrayListAllowDuplicate ▾



Git:

Test.java × ArrayListTest.java × ArrayListAllowDuplicate.java ×

```
locationsList.clear();
```

```
//6 ArrayList에서 값을 모두 지우려면 "clear()"을 이용한다.
```

```
//Q. 혹시 clear()로 모든값을 삭제한후에, 다시 값을 복원할 수 있는 메서드가 있나요?
```

```
System.out.println("after clear: " + locationsList);
```

```
System.out.println("거제도 어딴니 ? " + clone.indexOf("거제도"));
```

```
//7 ArrayList에서 인덱스값을 반환하려면 "indexOf()"을 이용한다.
```

```
// cf)lastIndexOf()는 반대순서로 반환한다.
```

```
System.out.println("제주도는 ? " + clone.indexOf("제주도"));
```

```
//8 찾는 값이 존재하지 않을때는 -1이 나온다.
```

```
System.out.println("오사카는 ? " + clone.indexOf("오사카"));
```

```
System.out.println("거제도 어딴니 ? " + clone.contains("거제도"));
```

```
//9 ArrayList에서 값이 존재하는지 구하려면 "contains()"을 이용한다. 출력은 boolean값으로 나온다.
```

```
System.out.println("제주도는 ? " + clone.contains("제주도"));
```

```
System.out.println("오사카는 ? " + clone.contains("오사카"));
```

}



"Kotlin" plugin up


```
3 class ArrayListCount {
4     ArrayList<Integer> intLists;
5     //1 ArrayList에서 문자열들이 값일때는 <String>, 숫자들이 값으로 존재할때는 <Integer>로 선언한다.
6     ArrayList<Integer> duplicateLists;
7
8     final int RAND_RANGE = 3;
9     final int RAND_START_OFFSET = 10;
10    final int DATA_LENGTH = 10;
11    final int FIRST_VALUE = 10;
12    final int SECOND_VALUE = 11;
13    final int THIRD_VALUE = 12;
14    final int FIRST_IDX = FIRST_VALUE - RAND_START_OFFSET;
15    final int SECOND_IDX = SECOND_VALUE - RAND_START_OFFSET;
16    final int THIRD_IDX = THIRD_VALUE - RAND_START_OFFSET;
17
18    public ArrayListCount () {
19        intLists = new ArrayList<Integer>();
20        duplicateLists = new ArrayList<Integer>();
21        //2 선언은 위에서 1번했지만, 초기화를 할때 객체는 여러번 쓸수 있는걸 기억한다.
22
23        // 10개 만들라 했으니 10개 만듬
24        for (int i = 0; i < DATA_LENGTH; i++) {
25            intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));
26            //intLists.add(10);
```

```
29 // 10, 11, 12는 총 3개로 랜덤 범위만큼 생성
30 // 10은 인덱스 0번
31 // 11은 인덱스 1번
32 // 12는 인덱스 2번
33 // 위와 같이 취급하겠다는 전략
34 for (int i = 0; i < RAND_RANGE; i++) {
35     duplicateLists.add(0);
36     //3 duplicateLists.add(0)는 "0을 마지막에 추가한다"라는 의미이다.
37     //4 duplicateLists.add(인덱스값, 0)는 "0을 인덱스값에 추가한다"라는 의미이다.
38 }
39 }
40
41 public void cntDuplicate () {
42
43     for (int num : intLists) {
44
45         if (num == FIRST_VALUE) {
46             // ArrayList.add(x): x를 현재 리스트에서 가장 마지막에 추가함
47             // ex) 0, 1, 4, 8
48             // ArrayList.add(77)
49             // ex) 0, 1, 4, 8, 77
50
51             // ArrayList.add(idx, x): x를 idx 인덱스 위치에 저장하고
52             // 기존에 있던 정보를 뒤로 한칸씩 밀어버림
```

```
41 public void cntDuplicate () {  
42  
43     for (int num : intLists) {  
44  
45         if (num == FIRST_VALUE) {  
46             // ArrayList.add(x): x를 현재 리스트에서 가장 마지막에 추가함  
47             // ex) 0, 1, 4, 8  
48             // ArrayList.add(77)  
49             // ex) 0, 1, 4, 8, 77  
50  
51             // ArrayList.add(idx, x): x를 idx 인덱스 위치에 저장하고  
52             // 기존에 있던 정보를 뒤로 한칸씩 밀어버림  
53             // ex) 0, 1, 4, 8  
54             // ArrayList.add(2, 77)  
55             // ex) 0, 1, 77, 4, 8  
56  
57             // 처음 시작시 [0, 0, 0]  
58             duplicateLists.add(FIRST_IDX, element: duplicateLists.get(FIRST_IDX) + 1);  
59             // 이후 [1, 0, 0, 0]  
60             // 그 다음은 [2, 1, 0, 0]  
61  
62             duplicateLists.remove(index: 1);  
63             // 이후 [1, 0, 0]  
64             // 그 다음은 [2, 0, 0]  
65             // 뺀 후 값이 1이 되면
```