

(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-15일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - Kyeonghwan Lee(이경환)
airtrade7@naver.com

```

3 class Test {
4     int[] arr;
5
6     public Test () {
7         arr = new int[3];
8
9         for (int i = 0; i < 3; i++) {
10             arr[i] = (int)(Math.random() * 6 + 1);
11         }
12     }
13
14     public int[] clone () {
15         int[] testArr = new int[arr.length];
16
17         for (int i = 0; i < arr.length; i++) {
18             testArr[i] = arr[i];
19         }
20
21         return testArr;
22     }
23
24     public void changeArr () {
25         arr[1] = 10000000;
26     }
27
28     public void vectorAdd (int[] vec) {
29         for (int i = 0; i < vec.length; i++) {
30             arr[i] += vec[i];
31         }
32     }
33
34     @Override
35     public String toString() {
36         return "Test{" +
37             "arr=" + Arrays.toString(arr) +
38             '}';
39     }

```

```

42 public class CanWeClone {
43     public static void main(String[] args) {
44         Test t = new Test();
45
46         System.out.println(t);
47         System.out.println("-----절취선-----");
48
49         int[] cloneArr = t.clone();
50
51         for (int i = 0; i < cloneArr.length; i++) {
52             System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
53         }
54         System.out.println("-----절취선-----");
55
56         t.changeArr();
57
58         System.out.println(t);
59         System.out.println("-----절취선-----");
60
61         for (int i = 0; i < cloneArr.length; i++) {
62             System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
63         }
64
65         System.out.println("-----절취선-----");
66         cloneArr[1] = 77777777;
67
68         for (int i = 0; i < cloneArr.length; i++) {
69             System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
70         }
71
72         t.vectorAdd(cloneArr);
73
74         System.out.println(t);
75     }
76 }

```

```

Test{arr=[2, 4, 2]}
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 4
cloneArr[2] = 2
-----절취선-----
Test{arr=[2, 10000000, 2]}
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 4
cloneArr[2] = 2
-----절취선-----
cloneArr[0] = 2
cloneArr[1] = 77777777
cloneArr[2] = 2
Test{arr=[4, 87777777, 4]}
Process finished with exit code 0

```

원본 인덱스 1번 데이터를 수정하였으나 클론 데이터는 변경 되지 않았다.

클론 데이터 수정도 가능하다.

ArrayListTest

```

4 public class ArrayListTest {
5     public static void main(String[] args) {
6         String[] fruits = {"apple", "strawberry", "grape", "watermelon"};
7         // asList()를 통해 배열등을 ArrayList로 변형할 수 있음
8         ArrayList<String> fruitsList = new ArrayList<>(Arrays.asList(fruits));
9
10        for (int i = 0; i < 4; i++) {
11            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
12        }
13
14        System.out.println("복제 이후!");
15
16        // fruitsList의 내용을 clone 변수에 복제함
17        // ArrayList에 구현되어 있는 clone은 객체를 복제할 수 있게 서포트한다.
18        ArrayList<String> clone = (ArrayList<String>) fruitsList.clone();
19        for (int i = 0; i < 4; i++) {
20            // ArrayList에 있는 내용을 가져올때 get(index)를 사용합니다.
21            // 여기서 index는 | 데이터1 | ---> | 데이터2 | ---> | 데이터8 | --->
22            //                0         1         2
23            System.out.printf("get(%d) = %s\n", i, clone.get(i));
24        }
25
26        fruitsList.remove("grape");
27        System.out.println("fruitsList의 grape 삭제 이후 clone 다시 보기");
28
29        for (int i = 0; i < 4; i++) {
30            System.out.printf("get(%d) = %s\n", i, clone.get(i));
31        }
32
33        System.out.println("fruitsList는 지워졌을까 ?");
34
35        for (int i = 0; i < 3; i++) {
36            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
37        }

```

원본 수정 시 Clone데이터(값)는 반영이 되지 않는다.

원본데이터는 정상 반영 되어있다

ArrayList 장점

데이터가 삭제되어도 인덱스의 빈 공간이 생기지 않고 바로 다음 저장 데이터로 넘어간다. 즉 데이터가 빈번히 변경하는 경우에 사용하면 좋다.

asList()
- 배열을 ArrayList로 변형 함

```

get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
복제 이후!
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
fruitsList의 grape 삭제 이후 clone 다시 보기
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
fruitsList는 지워졌을까 ?
get(0) = apple
get(1) = strawberry
get(2) = watermelon

```


ArrayListTest

```

39 // 완전 통채로 밀림
40 fruitsList.clear();
41
42 System.out.println("after clear: " + fruitsList);
43
44 // 검색 - indexOf
45 System.out.println("포도 어딴니 ? " + clone.indexOf("grape"));
46 System.out.println("딸기는 ? " + clone.indexOf("strawberry"));
47 // 존재하지 않는 것은 -1 이 나옴(오류)
48 System.out.println("드립을 치고 싶은데 ? " + clone.indexOf("드립"));
49
50 // 당첨자 명단 ~~~~
51 // 공모자 명단 ~~~~
52 // 시상식 참석자 명단 ~~~~
53 // for (참석자 명단 ~~~~) {
54 //     if (indexOf("당첨자 명단") == -1) {
55 //         continue;
56 //     } else {
57 //         cnt++
58 //     }
59 // }
60 // 위와 같은 형태로 시상식에 참석하면 수상자 숫자를 집계할 수 있다.
61
62 // contains 또한 IndexOf와 유사하게 사용이 가능하다.
63 System.out.println("포도 어딴니 ? " + clone.contains("grape"));
64 System.out.println("딸기는 ? " + clone.contains("strawberry"));
65 System.out.println("드립을 치고 싶은데 ? " + clone.contains("드립"));
66
67 // contains()는 참 혹은 거짓으로 결과가 나오는 것을 확인할 수 있다.
68 }
69 }

```

indexOf 활용 예시

clear()

- 배열에 들어가있는 내용 초기화

indexOf()

- 검색한 해당 데이터의 저장 위치(인덱스)를 확인 가능하다.
- 존재하지 않는 데이터의 경우 -1로 출력된다

contains()

- indexOf처럼 검색 데이터의 위치를 확인 하지는 못하나 데이터가 있는 경우에 true 없는 경우 false로 출력된다.
- indexOf와 유사하게 활용이 가능하다.

fruitsList는 지워졌을까 ?

get(0) = apple

get(1) = strawberry

get(2) = watermelon

after clear: []

포도 어딴니 ? 2

딸기는 ? 1

드립을 치고 싶은데 ? -1

포도 어딴니 ? true

딸기는 ? true

드립을 치고 싶은데 ? false

Process finished with exit code 0

```

6  class ArrayListCount {
7      ArrayList<Integer> intLists;
8      ArrayList<Integer> duplicateLists;
9
10     final int RAND_RANGE = 3;
11     final int RAND_START_OFFSET = 10;
12     final int DATA_LENGTH = 10;
13
14     final int FIRST_VALUE = 10;
15     final int SECOND_VALUE = 11;
16     final int THIRD_VALUE = 12;
17
18     final int FIRST_IDX = FIRST_VALUE - RAND_START_OFFSET;
19     final int SECOND_IDX = SECOND_VALUE - RAND_START_OFFSET;
20     final int THIRD_IDX = THIRD_VALUE - RAND_START_OFFSET;
21
22     // 생성자 - 초기화
23     public ArrayListCount () {
24         // 10 ~ 12의 숫자가 중복 허용된 상태로 10개 배치됨
25         intLists = new ArrayList<Integer>();
26         // 각각의 숫자가 몇 개씩 중복되었는지 체크에 활용함
27         duplicateLists = new ArrayList<Integer>();
28
29         // 10개 만들라 했으니 10개 만들
30         for (int i = 0; i < DATA_LENGTH; i++) {
31             intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));
32             //intLists.add(10);
33         }
34
35         // 10, 11, 12는 총 3개로 랜덤 범위만큼 생성
36         // 10은 인덱스 0번
37         // 11은 인덱스 1번
38         // 12는 인덱스 2번
39         // 위와 같이 취급하겠다는 전략
40         for (int i = 0; i < RAND_RANGE; i++) {
41             duplicateLists.add(0);

```

```

public void cntDuplicate () {
    // foreach <<<
    for (int num : intLists) {
        // 빼온 값이 10이면
        if (num == FIRST_VALUE) {
            // ArrayList.add(x): x를 현재 리스트에서 가장 마지막에 추가함
            // ex) 0, 1, 4, 8
            // ArrayList.add(77)
            // ex) 0, 1, 4, 8, 77

            // ArrayList.add(idx, x): x를 idx 인덱스 위치에 저장하고
            // 기존에 있던 정보를 뒤로 한칸씩 밀어버림
            // ex) 0, 1, 4, 8
            // ArrayList.add(2, 77)
            // ex) 0, 1, 77, 4, 8

```

```

// 처음 시작시 [0, 0, 0]
duplicateLists.add(FIRST_IDX,
    duplicateLists.get(FIRST_IDX) + 1);
// 이후 [1, 0, 0, 0]
// 그 다음은 [2, 1, 0, 0]
duplicateLists.remove(1);
// 이후 [1, 0, 0]
// 그 다음은 [2, 0, 0]
// 빼온 값이 11이면
} else if (num == SECOND_VALUE) {
    duplicateLists.add(SECOND_IDX,
        duplicateLists.get(SECOND_IDX) + 1);
    duplicateLists.remove(2);
// 빼온 값이 12라면
} else if (num == THIRD_VALUE) {
    duplicateLists.add(THIRD_IDX,
        duplicateLists.get(THIRD_IDX) + 1);
    duplicateLists.remove(3);

```

ArrayList.add(x)
- x를 배열의 마지막에 추가됨

ArrayList.add(idx, x)
- x를 idx 인덱스 위치에 저장하고
기존에 있던 정보를 뒤로 한칸씩
밀어버림

```

@Override
public String toString() {
    return "ArrayListCount{" +
        "intLists=" + intLists +
        ", duplicateLists=" + duplicateLists +
        '}';
}

public class ArrayListAllowDuplicate {
    public static void main(String[] args) {
        ArrayListCount alc = new ArrayListCount();

        System.out.println(alc);

        alc.cntDuplicate();

        System.out.println(alc);
    }
}

```

```

ArrayListCount{intLists=[10, 12, 12, 11, 10, 11, 10, 10, 10, 11], duplicateLists=[0, 0, 0]}
ArrayListCount{intLists=[10, 12, 12, 11, 10, 11, 10, 10, 10, 11], duplicateLists=[5, 3, 2]}

```

Process finished with exit code 0

▣Prob50

깃주소:

<https://github.com/LeeKyeongHwan/LectureContents/blob/main/java/KyeonghwanLee/FifteenthLecture/src/Prob50.java>

=====상품 목록=====

- 1.푸라면: 150
- 2.삼겹살: 300
- 3.무울: 50
- 4.파인애플: 100
- 5.초콜릿: 30
- 6.캐비어: 1000

구매 할 상품의 번호를 선택하세요.(장바구니 확인:7 종료:8): 1

푸라면 1개를 장바구니에 담았습니다.

구매 할 상품의 번호를 선택하세요.(장바구니 확인:7 종료:8): 3

무울 1개를 장바구니에 담았습니다.

구매 할 상품의 번호를 선택하세요.(장바구니 확인:7 종료:8): 5

초콜릿 1개를 장바구니에 담았습니다.

구매 할 상품의 번호를 선택하세요.(장바구니 확인:7 종료:8): 7

=====장바구니 목록=====

[푸라면, 무울, 초콜릿]

구매 할 상품의 번호를 선택하세요.(장바구니 확인:7 종료:8): 8

구매를 종료합니다.

Process finished with exit code 0

기본적인 세팅까지는 설정 하였으나, 소지금 연동 및 장바구니 구매 수량 설정, 구매 물품 판매 설정은 하지 못하였습니다. 이후 풀이해주시면 복습 진행 하도록 하겠습니다.

출력값