

(디지털커버전스)스마트 콘텐츠와 웹 융합응용SW개발자 양성과정

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - Joongyeon Kim(김종연)

jjjr69@naver.com

2021년 5월 28일 지문노트

[김종연]

```

import java.util.Arrays;
// 객체전달과 값 전달의 차이 다시 알아보기
class Test {
    int[] arr;

    public Test () {    // 생성자 -> 초기화 역할
        arr = new int[3];

        for (int i = 0; i < 3; i++) {
            arr[i] = (int)(Math.random() * 6 + 1);
        }
    }

    public int[] clone () {    // arr 배열의 인덱스값을 testArr[i]의 인덱스에 대입
        int[] testArr = new int[arr.length];

        for (int i = 0; i < arr.length; i++) {
            testArr[i] = arr[i];
        }

        return testArr;
    }

    public void changeArr () { arr[1] = 10000000; }

    public void vectorAdd (int[] vec) {    //
        for (int i = 0; i < vec.length; i++) {
            arr[i] += vec[i];
        }
    }

    System.out.println("-----절취선-----");
    cloneArr[1] = 7777777;    // 또다시 값을 인덱스에 대입한다

    for (int i = 0; i < cloneArr.length; i++) {
        System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);    // 변화된 값이 출력됨
    }

    t.vectorAdd(cloneArr);

    System.out.println(t);

```

```

@Override
    public String toString() {    // arr의 값을 나타냄    // toString 객체가 가지고 있는 값을 문자열로 나타냄
        return "Test{" +
            "arr=" + Arrays.toString(arr) +
            '}';
    }

    public class CanWeClone {
        public static void main(String[] args) {
            Test t = new Test();

            System.out.println(t);    // 랜덤값이 그대로나옴
            System.out.println("-----절취선-----");

            int[] cloneArr = t.clone();

            for (int i = 0; i < cloneArr.length; i++) {    // cloneArr의 배열을 불러온다(객체 불러옴)
                System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
            }
            System.out.println("-----절취선-----");

            t.changeArr();    // 기존 arr 배열을 복제하고 다른 값을 출력함

            System.out.println(t);
            System.out.println("-----절취선-----");

            for (int i = 0; i < cloneArr.length; i++) {
                System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);    // changeArr은 복제된 것이기에 기존 객체의 값은 변하지 않음
            }

```

```

import java.util.ArrayList;

//ArrayList에 중복을 허용하여 랜덤 숫자를 10개 만든다.
//랜덤 숫자의 범위는 10 ~ 12 다.
//여기서 각각의 숫자들이 몇 개씩 중복 되었는지 카운트해보자!
class ArrayListCount {
    ArrayList<Integer> intLists;
    ArrayList<Integer> duplicateLists;

    final int RAND_RANGE = 3; // 랜덤값 설정할 때 랜덤값의 범위를 3으로 맞추기위해 만든 변수
    final int RAND_START_OFFSET = 10; // 배열의 인덱스값을 맞추기위해 -10을 하기위한 변수
    final int DATA_LENGTH = 10; // 랜덤 숫자를 10개 만들기위해 만든 변수

    final int FIRST_VALUE = 10;
    final int SECOND_VALUE = 11; //10~12 값을 설정
    final int THIRD_VALUE = 12;

    final int FIRST_IDX = FIRST_VALUE - RAND_START_OFFSET;
    final int SECOND_IDX = SECOND_VALUE - RAND_START_OFFSET; //배열의 인덱스값에 맞추기 위해 모두 -10을 해준다
    final int THIRD_IDX = THIRD_VALUE - RAND_START_OFFSET;

    // 생성자 - 초기화
    public ArrayListCount () {
        // 10 ~ 12의 숫자가 중복 허용된 상태로 10개 배치됨
        intLists = new ArrayList<Integer>();
        // 각각의 숫자가 몇 개씩 중복되었는지 체크에 활용함
        duplicateLists = new ArrayList<Integer>();

        // 10개 만들라 했으니 10개 만듬
        for (int i = 0; i < DATA_LENGTH; i++) {
            intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));

```

```

        //intLists.add(10);
    }

    // 10, 11, 12는 총 3개로 랜덤 범위만큼 생성
    // 10은 인덱스 0번
    // 11은 인덱스 1번
    // 12는 인덱스 2번
    // 위와 같이 취급하겠다는 전략
    for (int i = 0; i < RAND_RANGE; i++) {
        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,
    element: duplicateLists.get(FIRST_IDX) + 1); // +1을 하는 이유 : 10이 나오면 중복값에 카운트하기 위해서 (+10을 하면 한번 나올때마다 중복값이 10씩 카운트된다)
// 이후 [1, 0, 0, 0]
// 그 다음은 [2, 1, 0, 0]
duplicateLists.remove(index: 1);
// 이후 [1, 0, 0]
// 그 다음은 [2, 0, 0]
// 빼온 값이 11이면
} else if (num == SECOND_VALUE) {
    duplicateLists.add(SECOND_IDX,
        element: duplicateLists.get(SECOND_IDX) + 1);
    duplicateLists.remove(index: 2);
    // 빼온 값이 12라면
} else if (num == THIRD_VALUE) {
    duplicateLists.add(THIRD_IDX,
        element: duplicateLists.get(THIRD_IDX) + 1);
    duplicateLists.remove(index: 3);
}
}
}

```

```

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

```

```

public class Prob49 {
    public static void main(String[] args) {
        ArrayListCount alc = new ArrayListCount();

        System.out.println(alc);

        alc.cntDuplicate();

        System.out.println(alc);
    }
}

```

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

```
class ArrayListCnt {
    ArrayList<Integer> intLists;
    ArrayList<Integer> duplicateLists;

    public ArrayListCnt(){
        intLists = new ArrayList<Integer>();
        duplicateLists = new ArrayList<Integer>();

        for (int i=0; i<10; i++){
            intLists.add((int)(Math.random()*3+10));
        }

        for (int i=0; i<3; i++){
            duplicateLists.add(0);
        }
    }

    public void cntDuplicat(){

        for (int num : intLists){

            if (num == 10){
                duplicateLists.add(index: 0, element: duplicateLists.get(10)+1);
                duplicateLists.remove(index: 1);
            } else if (num== 11){
                duplicateLists.add(index: 1, element: duplicateLists.get(11)+1);
                duplicateLists.remove(index: 2);
            }
        }
    }
}
```

```
        } else if(num==12){
            duplicateLists.add(index: 2, element: duplicateLists.get(12)+1);
            duplicateLists.remove(index: 3);
        }
    }

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

public class SelfTest49 {
    public static void main(String[] args) {
        ArrayListCnt alc = new ArrayListCnt();

        System.out.println(alc);

        alc.cntDuplicat();

        System.out.println(alc);
    }
}
```

```

import java.util.ArrayList;
import java.util.Arrays;

//ArrayList에 관하여
public class ArrayListTest {
    public static void main(String[] args) {
        String[] fruits = {"apple", "strawberry", "grape", "watermelon"};
        // asList()를 통해 배열등을 ArrayList로 변형할 수 있음
        ArrayList<String> fruitsList = new ArrayList<>(Arrays.asList(fruits)); //ArrayList배열 만드는 방법

        for (int i = 0; i < 4; i++) {
            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
        }

        System.out.println("복제 이후!");

        // fruitsList의 내용을 clone 변수에 복제함
        // ArrayList에 구현되어 있는 clone은 객체를 복제할 수 있게 서포트한다.
        ArrayList<String> clone = (ArrayList<String>) fruitsList.clone();
        for (int i = 0; i < 4; i++) {
            // ArrayList에 있는 내용을 가져올때 get(index)를 사용합니다.
            // 여기서 index는 | 데이터1 | ---> | 데이터2 | ---> | 데이터8 | ---> // 질문 데이터4가 아니라 8인건가요??
            //           0           1           2
            System.out.printf("get(%d) = %s\n", i, clone.get(i));
        }

        fruitsList.remove(0: "grape");
        System.out.println("fruitsList의 grape 삭제 이후 clone 다시 보기");

        for (int i = 0; i < 4; i++) {
            System.out.printf("get(%d) = %s\n", i, clone.get(i)); //clone에서는 지워지지않았다 (fruitList를 복제한 것이기에)
        }
    }
}

```

```
System.out.println("fruitsList는 지워졌을까 ?");
```

```
for (int i = 0; i < 3; i++) {  
    System.out.printf("get(%d) = %s\n", i, fruitsList.get(i)); //fruitList에서는 지워졌다  
}
```

```
// 완전 통채로 밀림
```

```
fruitsList.clear();
```

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

```
// 검색 - indexOf
```

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

```
System.out.println("딸기는 ? " + clone.indexOf("strawberry"));
```

```
System.out.println("사과는 ? " + clone.indexOf("apple"));
```

```
// 존재하지 않는 것은 -1 이 나옴(오류)
```

```
System.out.println("드립을 치고 싶은데 ? " + clone.indexOf("드립"));
```

```
// 당첨자 명단 ~~~~
```

```
// 공모자 명단 ~~~~
```

```
// 시상식 참석자 명단 ~~~
```

```
// for (참석자 명단 ~~~) {
```

```
//     if (indexOf("당첨자 명단") == -1) {
```

```
//         continue;
```

```
//     } else {
```

```
//         cnt++
```

```
// }  
// }  
// 위와 같은 형태로 시상식에 참석하면 수상자 숫자를 집계할 수 있다.  
  
// contains 또한 IndexOf와 유사하게 사용이 가능하다.  
System.out.println("포도 어딴니 ? " + clone.contains("grape"));  
System.out.println("딸기는 ? " + clone.contains("strawberry"));  
System.out.println("드립을 치고 싶은데 ? " + clone.contains("드립"));  
  
// contains()는 참 혹은 거짓으로 결과가 나오는 것을 확인할 수 있다.  
}
```



```

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;

// 1. 파는 상품 목록 (표시)
// 2. 메뉴 숫자를 눌러서 구매
// 3. 물건 구매시 판매리스트가 보임
// 4. 현재 소지한 물건리스트 표시
// 5. 구매가 판매가 지정하기
// 6. 상인 메소드, 손님 메소드, 물건 메소드, 상점 메소드 만들기

class Store {
    ArrayList<String> buyList;
    ArrayList<String> sellList;
    Scanner scan;

    String[] sellMenu = {"피자", "치킨", "콜라", "호빵"};

    public Store(){
        scan = new Scanner(System.in);
        buyList = new ArrayList<>();
        sellList = new ArrayList<>(Arrays.asList(sellMenu));
    }

    public void showMenu(){
        sellmenu = new Item[5];
        menu[0] = Item("피자", 1000, 10);
        menu[1] = Item("치킨", 2000, 10);
        menu[2] = Item("콜라", 800, 20);
        menu[3] = Item("호빵", 500, 10);
    }
}

public class Prob51 {
    public static void main(String[] args) {

    }
}

```