

(디지털 컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

```

class Test {
    int[] arr;

    public Test () {
        arr = new int[3];

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

    public int[] clone () {
        int[] testArr = new int[arr.length];

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

        return testArr;
    }
}

```

원본 배열과 복제된 배열

Test{arr=[2, 2, 5]}

-----절취선-----

cloneArr[0] = 2

cloneArr[1] = 2

cloneArr[2] = 5



Test{arr=[2, 10000000, 5]}

-----절취선-----

cloneArr[0] = 2

cloneArr[1] = 2

cloneArr[2] = 5

원본의 처음 숫자와 같은 배열이 복제된다.

이후에 원본의 배열을 바꾸면 복제된 배열에는 적용되지 않는다.
반대로 사본에서 바뀌도 원본에 영향이 없다.

```

public class ArrayListTest {
    public static void main(String[] args) {
        String[] fruits = {"Apple", "Strawberry", "Grape", "WaterMelon"};
        ArrayList<String> fruitsList = new ArrayList<>(Arrays.asList(fruits));
        // asList()를 통해 배열등을 ArrayList로 변형할 수 있음
        for (int i = 0; i < 4; i++) {
            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
        }

        System.out.println("복제 이후 ");

        ArrayList<String> clone = (ArrayList<String>) fruitsList.clone();

        for (int i = 0; i < 4; i++) {
            System.out.printf("get(%d) = %s\n", i, clone.get(i));
        }
        System.out.println("제거 이후 ");
        fruitsList.remove(0: "Grape");

        // 원본
        for (int i = 0; i < 3; i++) {
            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
        }
        // 사본
        for (int i = 0; i < 4; i++) {
            System.out.printf("get(%d) = %s\n", i, clone.get(i));
        }
    }
}

```

ArrayList에서 배열을 복사하는 기능 : `Array.asList()`

마찬가지로 복제된 배열이기때문에 복제 이후에는 원본을 수정해도 복제된 배열에 반영되지않는다.

복제된 배열에서 값을 수정해도 원본에 반영되지않음.

변경 전

```

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

```

변경 후

```

제거 이후
get(0) = Apple
get(1) = Strawberry
get(2) = WaterMelon
복제 이후
get(0) = Apple
get(1) = Strawberry
get(2) = Grape
get(3) = WaterMelon

```

indexOf

```
// 검색 - indexOf
for (int i = 0; i < 4; i++) {
    System.out.printf("get(%d) = %s\n", i, clone.get(i));
}
System.out.println("포도 " + clone.indexOf("Grape"));
System.out.println("딸기 " + clone.indexOf("Strawberry"));
```

검색하면 배열의 인덱스 번호가 출력되는 indexOf
존재하지않는 경우에는 -1로 나온다.

```
get(0) = Apple
get(1) = Strawberry
get(2) = Grape
get(3) = WaterMelon
```

포도 2

딸기 1

contains

```
//contains 또한 IndexOf와 유사하게 사용이 가능하다 true or false 값을 반환
System.out.println("포도 " + clone.contains("Grape"));
System.out.println("딸기 " + clone.contains("Strawberry"));

System.out.println("드립 " + clone.contains("드립"));
```

contains 또한 유사하게 사용 가능하면 참 혹은 거짓 값

```
get(0) = Apple
get(1) = Strawberry
get(2) = Grape
get(3) = WaterMelon
```

포도 true

딸기 true

드립을 치고 싶은데 false

ArrayList를 통해 중복값 제거

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;
    final int RAND_START_OFFSET = 10;
    final int DATA_LENGTH = 10;

    final int FIRST_VALUE = 10;
    final int SECOND_VALUE = 11;
    final int THIRD_VALUE = 12;

    final int FIRST_IDX = FIRST_VALUE - RAND_START_OFFSET;
    final int SECOND_IDX = SECOND_VALUE - RAND_START_OFFSET;
    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 만들
        for (int i = 0; i < DATA_LENGTH; i++) {
            intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));
            //intLists.add(10);
        }

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

0~2번째 배열 저장

숫자 10개 생성

숫자 범위 :10~12

```
public void cntDuplicate () {
    for (int num : intLists) {
        // 빠른 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);
            // 이후 [1, 0, 0, 0]
            // 그 다음은 [2, 1, 0, 0]
            duplicateLists.remove( index: 1);
            // 이후 [1, 0, 0]
            // 그 다음은 [2, 0, 0]

            // 빠른 값이 11이면
        } else if (num == SECOND_VALUE) {
            duplicateLists.add(SECOND_IDX,
                element: duplicateLists.get(SECOND_IDX) + 1);
            duplicateLists.remove( index: 2);
            // 빠른 값이 12라면
        } else if (num == THIRD_VALUE) {
            duplicateLists.add(THIRD_IDX,
                element: duplicateLists.get(THIRD_IDX) + 1);
            duplicateLists.remove( index: 3);
        }
    }
}
```

주의:
값이 add되면 인덱스위치에 저장되고 기존 정보를 한칸씩 뒤로 밀어낸다.
밀려난 인덱스를 강제로 삭제해야한다.

ArrayList - 상점 생성

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

class Shop {
    ArrayList<Merchandise> list;
    Customer me;
    Scanner sc;

    final int LIST_LENGTH = 3;

    public Shop() {
        list = new ArrayList<Merchandise>();
        me = new Customer();
        sc = new Scanner(System.in);
    }

    public void linker() {
        listSetting();
        listPrint();
        buy();
    }

    public void buy() {
        System.out.print("스피커 : 0 , 등 : 1, 모니터 : 2");
        int i = sc.nextInt();
        me.addList(list.get(i));
        me.addStock(list.get(i));
    }
}
```

중간부터 어떻게 해야될지 몰라서 완성하지 못했습니다.

```
public void listSetting() {
    Merchandise speaker = new Merchandise( name: "스피커", price: 1000, stock: 5);
    Merchandise lamp = new Merchandise( name: "등", price: 2000, stock: 5);
    Merchandise monitor = new Merchandise( name: "모니터", price: 3000, stock: 10);
    list.add(speaker);
    list.add(lamp);
    list.add(monitor);
}

public void listPrint() {
    System.out.println("=====상품목록=====");
    for (int i = 0; i < LIST_LENGTH; i++) {
        System.out.println(list.get(i).toString());
    }
}
```

```
class Merchandise { //상품설정
    String name;
    int price;
    int stock;

    public Merchandise(String name, int price, int stock) {
        this.name = name;
        this.price = price;
        this.stock = stock;
    }

    public String toString() { return "상품 : [" + name + "] 가격 : [" + price + "] 재고 : [" + stock + "]"; }
}
```

```

class Customer {
    ArrayList<Merchandise> cart_M;
    ArrayList<Integer> cart_Stock;
    ArrayList<String> cart;
    int money = 10000;
    Boolean isTrue;

    public Customer() {
        cart = new ArrayList<String>();
        cart_M = new ArrayList<Merchandise>();
        cart_Stock = new ArrayList<Integer>();
    }

    public void addList(Merchandise merchandise) {
        cart.add(merchandise.name);
    }

    public void addStock(Merchandise merchandise) {
        cart_Stock.add(merchandise.stock);
    }

    public void myList() {
        System.out.print("장 바 구 니 : \n");
        for (String data : cart) {
            System.out.println(data);
        }
    }

    public String toString() { return "상품 : " + cart + " 수량 : " + cart_Stock; }
}

```

```

public class ArrayListShop {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Shop sh = new Shop();
        Customer ct = new Customer();
        int num = 1;

        while (num != 0) {
            System.out.print("0 : 종료, 1: 상품목록 2: 내 장바구니 3: 총 합계 4: 지갑")
            num = sc.nextInt();
            switch (num) {
                case 1:
                    sh.linker();
                    continue;
                case 2:
                    sh.me.myList();
                    continue;
                case 3:
                    continue;
                case 4:
                    System.out.println(sh.me.money);
                    continue;
                default:
                    System.out.println("다시 입력해주세요");
                    break;
            }
        }
    }
}

```

구매 했을 때 각 스피커, 모니터 등의 Merchandise 객체를 Customer 클래스에서 받고 상품 재고를 어떻게 수정해야 될지 모르겠습니다.