



5월 28일 복습&퀴즈

이태양



```
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));

        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 | --->
            //                      0           1           2
            System.out.printf("get(%d) = %s\n", i, clone.get(i));
        }
    }
}
```

스트링배열안에 값을 넣어서 선언해주고

어레이리스트에 똑같은 내용을 저장해준다

어레이리스트를 보기위해 리스트명.get()사용

Clone어레이리스트에 값을 복제하여 넣는다

내용확인



```
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));
}

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

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

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"));
// 존재하지 않는 것은 -1 이 나옴(오류)
System.out.println("드립을 치고 싶은데 ? " + clone.indexOf("드립"));

// 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.
}
```

프루트리스트에 grape를 삭제한다

클론을 출력하고

프루트리스트 확인

프루트리스트를 다 비운다

출력

검색할 땐 indexOf를 사용

Contains또한 유사하게 사용하나

Contains는 참 거짓으로 결과값이 나옴



```
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
after clear: []
포도 어딴니 ? 2
딸기는 ? 1
드립을 치고 싶은데 ? -1
포도 어딴니 ? true
딸기는 ? true
드립을 치고 싶은데 ? false
```

복제하면 복제했을 때의 리스트의 내용이
그대로 들어가고 복제해왔던 리스트에
변동이 있어도 클론은 그대로이다



```
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 만듦
        for (int i = 0; i < DATA_LENGTH; i++) {
            intLists.add((int)(Math.random() * RAND_RANGE + RAND_START_OFFSET));
            //intLists.add(10);
        }
    }
}
```

필요한 리스트들과 변수들을 선언

어레이리스트 객체 생성하고

10~12사이의 난수들을 저장



```
// 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) {
        // 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
        }
    }
}
```

중복체크를 할 리스트의 각각의 자리를 0으로 설정한다

중복을 체크하는법

값을 넣어서 원래 있던 값이 뒤로빠지면 그 뒤에 있는 값을 지워주는 방식으로 사용



```
    } 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);  
    }  
}
```

toString을 이용한 출력

객체 생성 후 함수들 호출





```
class Shopping {
    ArrayList<String> buyLists;
    ArrayList<String> obLists;
    ArrayList<Integer> choicelists;
    ArrayList<Integer> pricelists;
    Scanner sc;
    Boolean isTrue = true;
    final int LENGTH = 4;

    public Shopping() {
        String[] sell = {"제육덮밥", "김치볶음밥", "라면", "돈까스"};
        Integer[] price = {5000, 6000, 3000, 7000};

        priceLists = new ArrayList<Integer>(Arrays.asList(price)) ;
        obLists = new ArrayList<String>(Arrays.asList(sell));

        buyLists = new ArrayList<String>();
        choiceLists = new ArrayList<Integer>();
    }

    public void showList() {
        System.out.println("목록을 출력합니다 !");

        for (int i = 0; i < LENGTH; i++) {
            System.out.println(i + 1 + "번" + " " + obLists.get(i)+ " 가격 : "+priceLists.get(i));
        }
        System.out.println("더구매하시려면 7번을, 구매한 목록을 보고싶으시면 8번, 결제는 9번을 입력해주세요 : ");
    }
}
```

가격과 음식 네개만 정해서
배열에 선언해주고

어레이리스트에 같은값을 넣어주었다

첫화면이랄까.. 가격과 음식이 같이 나오게
끔해주었다



```
public void buyItem(){

    System.out.println("구매를 진행합니다 !");
    sc = new Scanner(System.in);
    System.out.printf("숫자를 입력해주세요 :");
    int num = sc.nextInt();

    if(num == 1){
        System.out.println("제육덮밥을 구매하셨습니다!");
        buyLists.add("제육덮밥");
        buyItem();
    }else if(num == 2){
        System.out.println("김치볶음밥을 구매하셨습니다!");
        buyLists.add("김치볶음밥");
        buyItem();
    }else if(num == 3){
        System.out.println("라면을 구매하셨습니다!");
        buyLists.add("라면");
        buyItem();
    }else if(num == 4){
        System.out.println("돈까스를 구매하셨습니다!");
        buyLists.add("돈까스");
        buyItem();
    }else if(num == 7) {
        System.out.println("구매한 목록 ----- ");
        System.out.println(buyLists);
        buyItem();
    }else if(num == 8) {
        System.out.println("구매를 더 진행합니다. ");
        buyItem();

        else if(num == 9){
            printPrice();
        }
    }

}
```

숫자를 입력받아서

구매리스트에 추가해주게끔 해봤다,,

리스트보여주는것도 있고

구매를 더 진행한다곤하는데,,,그냥 다시 이 함수를 돌게되었다 음식이 있는 번호를 넣어도 리스트에 추가하고 다시 함수가 돌게끔되어있다



```
public void calcMenu() {  
    int cnt1=0;  
    int cnt2=0;  
    int cnt3=0;  
    int cnt4=0;  
  
    for(int i = 0; i < buyLists.size(); i++){  
  
        if(buyLists.get(i)=="제육덮밥"){  
            cnt1++;  
        }else if(buyLists.get(i)=="김치볶음밥"){  
            cnt2++;  
        }else if(buyLists.get(i)=="라면"){  
            cnt3++;  
        }else if(buyLists.get(i)=="돈까스"){  
            cnt4++;  
        }  
    }  
    choiceLists.add(cnt1);  
    choiceLists.add(cnt2);  
    choiceLists.add(cnt3);  
    choiceLists.add(cnt4);  
    System.out.println(choiceLists);  
}
```

뭔가 고민을 많이해봤는데 개수를 카운트해서 넣고싶은데
이방법밖에 떠오르지않았습니다 π.π



```
public void printPrice() {
    int sum = (priceLists.get(0)*choiceLists.get(0))+
              (priceLists.get(1)*choiceLists.get(1))+
              (priceLists.get(2)*choiceLists.get(2))+
              (priceLists.get(3)*choiceLists.get(3));
    System.out.println("제육덮밥 : "+choiceLists.get(0)+ "개 " + (priceLists.get(0)*choiceLists.get(0)));
    System.out.println("김치볶음밥 : "+choiceLists.get(1)+ "개 " + (priceLists.get(1)*choiceLists.get(1)));
    System.out.println("라면 : "+choiceLists.get(2)+ "개 " + (priceLists.get(2)*choiceLists.get(2)));
    System.out.println("돈까스 : "+choiceLists.get(3)+ "개 " + (priceLists.get(3)*choiceLists.get(3)));
    System.out.println("총 계산할 금액은 : "+ sum);
}
```

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

        Shopping sp = new Shopping();

        sp.showList();
        sp.buyItem();
        sp.calcMenu();

    }
}
```

진짜 그냥 노가다로 푼거같아서...
조금 찝찝합니다

몇개인지와 그에대한 값을 계산해서 출력할 수 있게 했는데 두 개의 어레이리스트에서 나오는 값을 받아서 같은 위치에있는 것 끼리 곱할 수 있을까?라는 생각은 했는데 구현은 못했습니다
(배열이였으면 할수있었을 것 같기도하고),,