```java
import java.util.ArrayList;
import java.util.Arrays;
public class _1st_ArrayList_2 {
    public static void main(String[] args) {
        String[] fruits = {"apple", "strawberry", "grape", "watermelon"};
        //Arrays.asList()를 통해 배열 등을 ArrayList로 변형 할 수 있다.
        ArrayList<String> fruitsList = new ArrayList<>(Arrays.asList(fruits));

        System.out.println("<<fruitsList[]>>");
        for(int i = 0; i < 4 ; i++) {
            System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
            // ArrayList에 있는 값을 가져올 때는 get(index)를 사용.
            // 여기서 index는  | data1| ->| data2 | -> | data8|
            //                    0         1           2
        }
        // fruitsList의 내용을 clone 변수에 복제함.
        // 배열을 복제했으니까 이건 객체(원본)을 복사했다 라고 오해 할 수 있는데
        // clone은 그 값들을 복사하는 것이니 주의
        // ArrayList에 구현되어 있는 clone은 객체를 복제할 수 있게 support
        ArrayList<String> clone = (ArrayList<String>) fruitsList.clone();
        System.out.println("<<clone>>");
        for(int i = 0; i < 4 ; i++) {
            System.out.printf("get(%d) = %s\n", i, clone.get(i));
        }
```

```
"C:\Program Files\Java
<<fruitsList[]>>
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
<<clone>>
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
```

2021.05.28 Java

〈ArrayList의 method〉

- Arrays.asList()

배열 등을 ArrayList로 변형할 수 있다.
이미 배열로 만들어 놓은 fruits[]를
fruitsList[]라는 ArrayList로 변형

- .get

ArrayList에 있는 값을 가져올 때 사용

- .clone

how to use:
(ArrayList<type>)ArrayList이름.clone
ArrayList의 '값'을 복사
clone이라는 변수에 fruitsList[]의 값들을
복사.
〈〈객체원본 복사 아님〉〉

```java
// 복사한 대상인 fruitsList[]에서 grape를 빼도
// 다시 clone의 값들을 확인해보면 grape가 빠지지 않은 것을 확인.
fruitsList.remove( o: "grape"); // .remove
System.out.println("<<fruitsList[]에서 grape를 삭제 후 clone[]의 값들을 확인>>");
for(int i = 0; i < 4 ; i++) {
    System.out.printf("get(%d) = %s\n", i, clone.get(i));
}

System.out.println("<<fruitsList[]에서 garpe가 빠졌는지 확인>>");
for(int i = 0; i < 3 ; i++) {
    System.out.printf("get(%d) = %s\n", i, fruitsList.get(i));
}
```

```
<<fruitsList[]에서 grape를 삭제 후 clone[]의 값들을 확인>>
get(0) = apple
get(1) = strawberry
get(2) = grape
get(3) = watermelon
<<fruitsList[]에서 garpe가 빠졌는지 확인>>
get(0) = apple
get(1) = strawberry
get(2) = watermelon
```

- .remove

값을 삭제할 때 쓰임

fruitsList[]에서 grape를 삭제 후

clone[]과 fruitsList[]를 비교.

clone[]은 .clone으로 복사한 값이기 때문에

grape가 그대로 들어가 있다.

fruitsList는 index[2]에 있던 grape가 삭제되고

index[3]에 있던 watermelon이 index[2]의 자리를 채움.

```java
fruitsList.clear(); // .clear
System.out.println("<< clear로 배열 값들 삭제 >>");
System.out.println("after clear: " + fruitsList);
// clear는 내부의 '값'만 clear시킴. 즉 객체는 살아있음 = 객체는 data 위에 있음.

// indexOf // fruitsList clear로 밀어버려서 clone[]을 응용
System.out.println("grape는 어느 index에? " + clone.indexOf("grape"));
System.out.println("strawberry는 어느 index에? " + clone.indexOf("strawberry"));
System.out.println("없는 값을 indexOf하면? " + clone.indexOf("낑깡"));
// ArrayList에 존재하지 않는 것은 -1
// -1은 오류를 뜻 하지만 숫자로도 사용가능.
// indexOf를 사용할 수 있는 예를 들면
// 당첨자 명단[] / 시상식 참석자 명단[]
/* for( 참가판단[] : 시상식 참석자 명단[])      >> for-each 이런 느낌으로 쓸 수 있다~ 정도로 인지.
    if(indexOf("당첨자") == -1){ continue}
    else{ ~~~ }
*/
//System.out.println("ArrayList로 만들지 않은 fruits 배열은 error 발생 " + fruits.indexOf("낑깡"));

// contains  >> 포함되어 있는가?를 의미 true 또는 false로 결과가 나온다.
System.out.println("grape가 contain되어 있는가? " + clone.contains("grape"));
System.out.println("낑깡이 contain되어 있는가? " + clone.contains("낑깡"));
```

- .clear

내부의 '값'을 clear 시킨다.

즉. 객체는 살아있다. = data상에 있다.

상자는 그대로 있고 상자내부만 비우는 것.

- .indexOf()

어느 index에 위치하는지 알 수 있다.

존재하지 않을 때에는 '-1'로 나타내고

'-1'은 오류를 뜻 하지만 숫자로도 사용가능.

- .contains()

배열에 포함되어 있는가를 알 수 있다.

포함되어 있으면 true

없으면 false

```
<< clear로 배열 값들 삭제 >>
after clear: []
grape는 어느 index에? 2
strawberry는 어느 index에? 1
없는 값을 indexOf하면? -1
grape가 contain되어 있는가? true
낑깡이 contain되어 있는가? false
```

```java
System.out.println("ArrayList로 만들지 않은 fruits 배열은 error 발생 " + fruits.indexOf("낑깡"));
```

ArrayList의 method들이기 때문에 ArrayList가 아닌 배열에 사용하면 당연히 error.

```java
import java.util.Arrays;
class Test{
    int[] arr;

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

    public int[] clone(){
        int[] cloneTestArr = new int[arr.length];
        for (int i = 0; i < arr.length; i++){
            cloneTestArr[i] = arr[i];
        }
        return cloneTestArr;
    }

    public void changeArr() { arr[1] = 100000; }

    @Override
    public String toString() {
        return "Test{" +
                "arr=" + Arrays.toString(arr) +
                '}';
    }
}
```

```java
public class _1st_ArrayCloneExample {
    public static void main(String[] args) {
        //...
        Test t = new Test();
        System.out.println(t);
        System.out.println("--------");

        int[] cloneArr = t.clone();

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

        t.changeArr();
        System.out.println("t.changeArr() method를 통해서 arr[1]의 값을 변경 ");
        System.out.println(t); // arr[] 값 print
        System.out.println("--------");
        for(int i = 0; i < cloneArr.length; i++){
            System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
        }

        System.out.println("--------");
        cloneArr[1] = 777777; // cloneArr[1]의 값을 변경
        System.out.println("cloneArr[1]의 값을 777777로 변경");
        for(int i = 0; i < cloneArr.length; i++){
            System.out.printf("cloneArr[%d] = %d\n", i, cloneArr[i]);
        }
        System.out.println("--------");
        System.out.println("arr[]의 값을 print해서 확인");
        System.out.println(t);
    }
}
```

```
Test{arr=[6, 6, 4]}
--------
cloneArr[0] = 6
cloneArr[1] = 6
cloneArr[2] = 4
--------
t.changeArr() method를 통해서 arr[1]의 값을 변경
Test{arr=[6, 100000, 4]}
--------
cloneArr[0] = 6
cloneArr[1] = 6
cloneArr[2] = 4
--------
cloneArr[1]의 값을 777777로 변경
cloneArr[0] = 6
cloneArr[1] = 777777
cloneArr[2] = 4
--------
arr[]의 값을 print해서 확인
Test{arr=[6, 100000, 4]}
```

객체/값/clone에 대한 예제

간단하게 독해하고 이해하고 넘어가면 될 듯.

```java
import java.util.ArrayList;
public class _1st_Quiz49 {
    public static void main(String[] args) {
        // ArrayList에 중복을 허용하여 랜덤 숫자 10개 생성.
        // 숫자의 범위는 10~12
        // 여기서 각각의 숫자들이 몇 개씩 중복되었는지 count.

        // ArrayList에 data type 적을 때는 class type으로 적어야됨. int > Integer
        ArrayList<Integer> number = new ArrayList<Integer>();

        for(int i = 0; i < 10; i++){
            //number.set(i, (int)(Math.random() * 3 + 10)); 이거는 왜 오류가 나지..?
            number.add((int)(Math.random() * 3 + 10));
        }
        for(int numberprint : number){
            System.out.println("잘 들어가나 확인용: " + numberprint);
        }
        System.out.println(number.indexOf(10));
    }
}
```

〈Quiz.49〉

수업 중에 풀어보다가 실패한 코드인데

여기서 알고 가야할 개념

ArrayList의 data type을 적을 때에는 class type으로 적어야된다.

int (x) 〉〉〉〉 Integer (O)

error가 나는 이유는

.set은 '수정'하는 method이기 때문.

값도 안 정해져있는데 수정하라니까 당연히 error.

〈Quiz.49〉는 강사님 풀이 이용해서 복습

〈Quiz.49 강사님 풀이〉 이번 풀이는 일일이 정리/복습 하는 것 보다 스스로 한 번 더 쭉 독해 하는 것이 더 유익하고 재밌다.

```java
import java.util.ArrayList;
class Array_DuplicateCnt{

    public ArrayList<Integer> intLists;
    public ArrayList<Integer> duplicateLists;

    final int DATA_LENGTH = 10;
    final int RANGE = 3;
    final int START = 10;

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

    final int FIRST_IDX = FIRST_VALUE - START;
    final int SECOND_IDX = SECOND_VALUE - START;
    final int THIRD_IDX = THIRD_VALUE - START;
```

```java
public Array_DuplicateCnt(){ //생성자
    // 10~12의 숫자를 중복 허용된 상태로 10개 배치시킬 배열
    intLists = new ArrayList<Integer>();
    //각각의 숫자가 몇 개씩 중복되었는지 체킹에 활용할 배열
    duplicateLists = new ArrayList<Integer>();

    // intLists에 10개의 값 setting
    for(int i = 0; i < DATA_LENGTH; i++){
        intLists.add((int)(Math.random() * RANGE + START));
    }

    // 10, 11, 12는 총 3개로 랜덤하게 10개 생성
    // 10은 index[0] / 11은 index[1] / 12는 index[2] 로 취급하겠다는 전략.
    for(int i = 0; i < RANGE; i++){
        duplicateLists.add(0);
    }
```

```java
public void cntDupicate(){
    //for-each 사용
    for(int num : intLists){
        if(num == FIRST_VALUE){ // 값이 10일 때
            //[0, 0, 0]  | 그 다음 10이 나왔을 때는 [1, 0 ,0]
            duplicateLists.add(FIRST_IDX, element: duplicateLists.get(FIRST_IDX) + 1);
            //[1, 0, 0, 0] | [2, 0, 0, 0]
            duplicateLists.remove( index: 1);
            // [1, 0, 0] | [2, 0, 0,]

            //아래의 개념을 위의 if문에 적용해서 생각해보자.
            //ArrayList.add(x)는 x를 배열 가장 마지막에 추가
            // ex) 0, 1, 4, 8
            // ArrayList.add(77) >> 0, 1, 4, 8, 77
            // ---------------------------------------------
            // ArrayList.add(n, x)는 x를 index[n]에 위치하도록 함.
            // 기존에 있던 정보들은 뒤로 한 칸씩 밀림
            // ex) 0, 1, 4, 8
            // ArrayList.add(2, 77) >> ex) 0, 1, 77, 4, 8
        } else if (num == SECOND_VALUE){ // 값이 11일 때
            duplicateLists.add(SECOND_IDX, element: duplicateLists.get(SECOND_IDX) + 1);
            duplicateLists.remove( index: 2);
        } else if (num == THIRD_VALUE){ // 값이 12일 때
            duplicateLists.add(THIRD_IDX, element: duplicateLists.get(THIRD_IDX) + 1);
            duplicateLists.remove( index: 3);
        }
    }
```

```java
    @Override
    public String toString() {
        return "Array_DuplicateCnt{" +
                "intLists=" + intLists +
                ", duplicateLists=" + duplicateLists +
                '}';
    }
}
public class _2nd_Quiz49TeacherSolution {
    public static void main(String[] args) {

        Array_DuplicateCnt dc = new Array_DuplicateCnt();
        System.out.println(dc);
        dc.cntDupicate();
        System.out.println(dc);
    }
}
```

```
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C:\Users\Samuel\AppData\Local\JetBrains\T
Array_DuplicateCnt{intLists=[11, 12, 10, 11, 11, 11, 11, 11, 10, 12], duplicateLists=[0, 0, 0]}
Array_DuplicateCnt{intLists=[11, 12, 10, 11, 11, 11, 11, 11, 10, 12], duplicateLists=[2, 6, 2]}
```

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

class Shop {
    final String[] FOR_SALE = {"apple", "banana", "melon", "sugar", "salt", "oil"}; // 판매리스트
    int money = 13000; // 구매자 보유자금
    Scanner scan;
    public ArrayList<String> buy_list;

    public Shop() {
        buy_list = new ArrayList<String>();
    }
    public void order() {
//        boolean buy_more = true;
//        while (buy_more) {
            System.out.print("무엇을 구매하시겠습니까?: ");
            scan = new Scanner(System.in);
            String what_to_buy = scan.nextLine();
            buy_list.add(what_to_buy);

//            System.out.println("추가로 구매하시겠습니까? yes/no");
//            String yes_or_no = scan.nextLine();
//            if (yes_or_no == "yes") {
//                buy_more = true;
//            } else if (yes_or_no == "no") {
//                // buy_more = false; // 이 code도 안 되고
//                // break;             // break; 해도 계속 무한루프..
```

〈Quiz.51〉
문제대로 coding하기 실패
강사님 풀이 듣고 다시 작성 후
pdf 업데이트

Q. 추가로 구매할지 여부를 while을 통해서 coding하고 싶었는데 "no"를 입력해도 계속 무한루프가 걸립니다..

```
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C:
가게 판매리스트 = [apple, banana, melon, sugar, salt, oil]
무엇을 구매하시겠습니까?: apple
추가로 구매하시겠습니까? yes/no
no
무엇을 구매하시겠습니까?: banana
추가로 구매하시겠습니까? yes/no
no
무엇을 구매하시겠습니까?:
```

```java
    public void calculation(){
        if(buy_list.contains("apple")){
            money -= 1500;
        }
        if(buy_list.contains("banana")){
            money -= 3500;
        }
        if(buy_list.contains("melon")){
            money -= 5000;
        }
        if(buy_list.contains("sugar")){
            money -= 2000;
        }
        if(buy_list.contains("salt")){
            money -= 1500;
        }
        if(buy_list.contains("oil")){
            money -= 3500;
        }
```

```java
    @Override
    public String toString() {return "가게 판매리스트 = " + Arrays.toString(FOR_SALE);}
    public int getMoney() {
        System.out.print("남은 돈: ");
        return money;
    }
}
public class _99th_Quiz51 {
    public static void main(String[] args) {
//        ArrayList를 활용하여 상점을 프로그래밍.
//                '구매', '판매', '목록보기', '소지금', '물건의 구매가' 등을 지정해서
//                적정 버튼을 누르면 처리되도록 할 것.
//                초기에 구매리스트에는 아무것도 없다.
//                초기에는 상점 주인이 파는 판매리스트만 존재.
//                물건을 구매하면 구매한 물품이 구매리스트에 보인다.
//                '목록 보기'는 단순히 현재 소지한 물건 리스트만 보여준다.

        // 판매리스트 작성 / OK
        // 판매리스트마다 가격 갖는 method
        // 초기 구매자 보유금 setting / OK
        // 구매리스트 - ArrayList
        // 구매 - Switch Case
        // 목록보기 - for + .get() 사용
        Shop shopping = new Shop();
        System.out.println(shopping);
        shopping.order();
        shopping.calculation();
        System.out.println(shopping.getMoney());
```

```
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C
가게 판매리스트 = [apple, banana, melon, sugar, salt, oil]
무엇을 구매하시겠습니까?: banana
남은 돈: 9500
```