



# 5월 27일 복습&퀴즈

이태양



```
// 최종 결론
// 게터고 나발이고 다 떠나서
// 객체는 메모리 자체를 전달하며
// 값은 메모리가 아닌 값을 전달한다.
```

```
class CloneMemory {
    int[] arr;
    int num;

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

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

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

    public void reNum () {
        num = 7;
    }
}
```

```
// 게터는 모두 클론임을 어떻게 증명할까 ?
// 일단 게터로 배열값을 리턴받아 어딘가에 저장한다.
// 다음에 매서드를 사용해서 객체의 배열의 값을 변경한다.
// 리턴받아 저장했던 배열을 출력했을때 결과가 같으면 객체가 전달된
// 만약 다른 결과가 도출된다면 복제되었음을 알 수 있다.
// 결론: 결국 객체에 대한 리턴이므로 메모리 정보가 전달됨
```

```
public int[] getCloneArr () {
    return arr;
}

public int getCloneVariable () {
    return num;
}

public String toString () {
    return "arr[0] = " + arr[0] +
        ", arr[1] = " + arr[1] +
        ", arr[2] = " + arr[2];
}
```

```
public class MemoryCloneTest {
```

```
public static void main(String[] args) {
    CloneMemory cm = new CloneMemory();

    System.out.println(cm);

    int[] save = cm.getCloneArr();

    System.out.printf("save[0] = %d, save[1] = %d, save[2] = %d\n",
        save[0], save[1], save[2]);

    cm.reRandArr();

    System.out.println("객체에 접근해 출력");
    System.out.println(cm);

    System.out.println("사전 저장 정보 출력");
    System.out.printf("save[0] = %d, save[1] = %d, save[2] = %d\n",
        save[0], save[1], save[2]);

    // 결론: 자바에서 객체에 대한 접근은 모두 메모리를 제어하는 방식이 된다.

    int num = cm.getCloneVariable();

    System.out.println("객체내 변수값 획득: " + num);

    cm.reNum();

    System.out.println("변경 후 사전 획득한 정보 재출력: " + num);

    // 결론: 앞서서도 확인했지만 값에 대해서는 복제가 이루어짐을 확인할 수 있다.
    System.out.println("변경 정보 파악: " + cm.getCloneVariable());
}
```



chapter

```
arr[0] = 4, arr[1] = 3, arr[2] = 1
```

```
save[0] = 4, save[1] = 3, save[2] = 1
```

객체에 접근해 출력

```
arr[0] = 1, arr[1] = 3, arr[2] = 3
```

사전 저장 정보 출력

```
save[0] = 1, save[1] = 3, save[2] = 3
```

객체내 변수값 획득: 3

변경 후 사전 획득한 정보 재출력: 3

변경 정보 파악: 7

← 값의 복제를 확인



```
import java.util.ArrayList;

public class ArrayListTest {
    public static void main(String[] args) {
        // 일종의 배열
        // 배열의 사이즈를 지정하고 사용해야 하지만
        // 넣고 싶은대로 막 넣어도된다
        // 힙메모리에 동적할당을 수행한다
        ArrayList<String> lists = new ArrayList<String>();
        // 배열과의 차이점은 메모리가 불연속 배치된다는것

        lists.add("빵");
        lists.add("버터");
        lists.add("우유");
        lists.add("계란");
        lists.add("주스");
        lists.add("베이컨");
        lists.add("파스타");
        lists.add("피자");

        for(String list : lists){
            System.out.println("현재 항목은 =" + list);
        }
    }
}
```

## ArrayList를 이용하는 방법연습



```
class Shop {
    ArrayList<String> lists;
    Scanner scan;

    public Shop () {
        lists = new ArrayList<String>();
        scan = new Scanner(System.in);
    }

    public void deliveryCome () {
        System.out.print("필요한 물품을 말씀하세요: ");
        lists.add(scan.nextLine());
    }

    public void cancelOrder () {
        System.out.print("취소할 물품을 말씀하세요: ");
        lists.remove(scan.nextLine());
    }

    // toString 으로 자동 완성 가능
    // 객체 정보 출력에 사용합니다.
    // 아직 인터페이스 배우지 않았으므로 설명은 향후 진행
```

```
@Override
public String toString() {
    return "Shop{" +
        "lists=" + lists +
        '}';
}

public class ShopTest {
    public static void main(String[] args) {
        Shop s = new Shop();

        for(int i = 0; i < 3; i++) {
            s.deliveryCome();
        }

        s.cancelOrder();

        // 아래와 같이 객체를 전달하면 toString이 호출됨
        System.out.println(s);
    }
}

// ArrayList는 Queue 혹은 Stack 역할을 할 수 있는데 기본이 Queue(큐) 역할
// 맨 처음 넣은 정보가 가장 앞에 배치되고
// 두 번째 넣은 정보가 두 번째에 배치되고 ...
```

toString함수를 이  
용

어레이리스트에  
추가하고 삭제하는  
방법 연습



```
class Roulette {  
    ArrayList<String> nameLists;  
    String[] tmpArr;  
    int[] tmpIdx;  
  
    int[] success;  
  
    int nameLength;  
    Boolean isRedundant;  
  
    public Roulette (String[] names) {  
        nameLength = names.length;  
        isRedundant = true;  
  
        nameLists = new ArrayList<String>();  
        tmpArr = new String[nameLength];  
        tmpIdx = new int[nameLength];  
  
        success = new int[3];  
  
        int i = 0;  
  
        for (String name : names) {  
            tmpArr[i++] = name;  
        }  
    }  
}
```

이름을 받아넣을 배열  
번호를 받아넣을 배열  
랜덤으로 당첨번호 3개를 넣을 배열

Name배열의 길이를 받아쓰기위해 변수선언  
반복문사용을 위해 불린 변수선언

new로 각각의 객체를 생성해줬다

3개의 당첨번호를 받기위해 3으로 지정

Names에 있는 이름배열을 tmpArr에 옮겨 저장



```
private Boolean checkDuplicate (int idx) {
    for (int i = 0; i < idx; i++) {
        if (tmpIdx[i] == tmpIdx[idx]) {
            return true;
        }
    }

    return false;
}

public void shuffle () {
    int i = 0;

    isRedundant = true;

    do {
        tmpIdx[i] = (int)(Math.random() * nameLength);

        if (checkDuplicate(i)) {
            continue;
        }

        i++;

        if (i == nameLength) {
            isRedundant = false;
        }
    } while (isRedundant);
}
```

중복을 배제하기 위한 함수

난수를 발생시켜서 숫자를 섞기 위한 함수



```
public void checkSuccess () {
    int i = 0;

    isRedundant = true;

    do {
        success[i] = (int)(Math.random() * nameLength);

        if (checkDuplicate(i)) {
            continue;
        }

        i++;

        if (i == 3) {
            isRedundant = false;
        }
    } while (isRedundant);
}

public void printSuccessArr () {
    for (int i = 0; i < 3; i++) {
        System.out.printf("success[%d] = %d\n", i, success[i]);
    }
}

public void checkName() {
    for(int i = 0; i < tmpIdx.length; i++){
        for(int j = 0; j < 3; j++){
            if(success[j] == tmpIdx[i]){
                System.out.println("축하합니다 !! 당첨자는 : " + tmpArr[i] + "입니다!");
            }
        }
    }
}
```

당첨번호 3개를 뽑기위한 함수(중복제거 포함)

출력함수

checkName()이라는 함수를 만들어 Success의 3개의 인덱스와 tmpIdx의 값을 비교해서 같으면 그와 같은 위치에 있는 tmpArr[](이름이 들어 있는배열)에서 이름을 출력한다.





```
String[] name = { 박세진, 김창욱, 김민규, 김종연, 문성호, 강병화, 최승현, 유종현, 한상우, 전승리, 이경환, 최준환, 김원석, 여인준, 이태양, 김윤영, 정도영, 황정아, 임초롱, 김남교, 이주형, 김도연, 최혜주, 김도혜, 고재권, 임익환, 안보미, 이상훈 };
```

```
Roulette{tmpIdx=[11, 6, 27, 3, 4, 19, 16, 0, 23, 2, 22, 24, 9, 10, 14, 12, 5, 25, 18, 21, 26, 17, 15, 8, 1, 7, 13, 20]}  
success[0] = 3  
success[1] = 22  
success[2] = 25  
축하합니다 !! 당첨자는 : 김종연입니다!  
축하합니다 !! 당첨자는 : 이경환입니다!  
축하합니다 !! 당첨자는 : 황정아입니다!
```

Success의 값이 오름차순으로 나오면 이름도 순서대로 나오지만 값이 오름차순이 아니라면 당첨자의 이름 순서가 맞지않는다 둘 다 맞출려면 어떻게 해야할까요??