

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

변수 / 클래스 / 메소드

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day14/src/OOP.java>

```
21 ▶ public class OOP {
22 ▶     public static void main(String[] args) {
23         // 객체는 사용하려고 메모리에 올리는 것으로 아래와 같이 만들며,
24         // new는 객체를 생성할 때 사용하는 키워드이다.
25         // 이는 Animal 클래스의 인스턴스인 cat, 즉 Animal의 객체가 만들어진다.
26         Animal cat = new Animal();
27
28         // 객체와 인스턴스의 차이 ?
29         // 위 예제에서 cat은 객체이고, cat이라는 객체는 Animal의 인스턴스이다.
30         // 인스턴스는 특정 객체(cat)이 어떤 클래스(Animal)의 객체인지를 관계 위주로 설명할 때 사용된다.
31         // 즉, cat은 Animal의 인스턴스이다.
32
33
34         // 1. 변수란 ?
35         // 아래와 같은 예에서는 a,b를 변수라고 한다.
36         int a; // int(정수) 자료값을 담을 수 있는 변수 a
37         String b; //String(문자열) 자료값을 담을 수 있는 변수 b
38
39         // 변수의 이름은 마음대로 지을 수 있지만, 숫자, 특수문자, 자바의 키워드는 변수명으로 사용될 수 없다.
40         // ex_ int 2nd (X) , int c@ (X), int boolean (X) 등
41
42
43     }
44 }
```

```
1 class Animal{
2     // 2. 클래스란 ?
3     // 클래스에서 선언된 데이터들은 객체가 사용할 데이터이다.
4     // 클래스는 객체를 만드는 기능을 가진다.
5
6     String name;
7     // name이라는 String 변수,
8     // 클래스에 의해 생성되는 것 = 객체,
9     // 클래스에서 선언된 변수는 객체변수
10
11     public void setName(String name) {
12         this.name = name;
13     }
14     // 3. 메소드란 ?
15     // 메소드는 클래스 내에 구현된 함수
16     // setName 메소드는 다음과 같다.
17     // 입력으로 name이라는 문자열을 받고 출력이 없는 형태의 메소드다.
18     // 메소드가 하는 일은 입력을 가지고 어떤 일을 수행한 다음 결과물을 내놓는 것.
19 }
20 }
```

객체 전달과 값 전달의 차이점

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day14/src/MemoryCloneTest.java>

```
50 ▶ public class MemoryCloneTest {
51 ▶     public static void main(String[] args) {
52         CloneMemory cm = new CloneMemory();
53
54         // 원본
55         System.out.println(cm);
56
57         // 게터는 모두 클론(복사본) 인데?
58         // 리턴받아 저장했던 배열을 출력했을때 위와 동일함
59         // 결과가 같으면 객체가 전달된 것이다.
60         // 같은 결과가 출력됐으니, 원본이 아닌가?
61         // 게터인데 원본인가?
62         int[] save = cm.getCloneArr();
63         System.out.printf("save[0] = %d, save[1] = %d, save[2] = %d\n",
64             save[0], save[1], save[2]);
65         // (게터에서도 객체를 리턴하는 것은 메모리, 즉 원본이다.)
66         // 게터에서도 객체를 리턴하는 경우가 위의 경우인 걸까?
67         //     public int[] getCloneArr () {
68         //         return arr; }
69         // 게터에서 arr을 리턴했기 때문에, 배열 객체 전체, 즉 원본이었나?
70
71
72         // int[] arr에서 arr은 배열 객체 전체
73         // arr[5] 와 같은 특정 인덱스 지정은 변수와 동일하게 값을 전달함.
74
75
76         cm.reRandArr();
77         //     public void reRandArr () {
78         //         for (int i = 0; i < 3; i++) {
79         //             arr[i] = (int)(Math.random() * 6 + 1);
80         //         }
81         //     }
82         // 각각 a[0], a[1], a[2]
83
```

객체 전달과 값 전달을 잘 이해가 가지않아서,
각 부분에서 생각했던 의문을 적어봤습니다...
둘의 차이점 한번만 더 설명 부탁드립니다.

```
84         System.out.println("객체에 접근해 출력");
85         System.out.println(cm);
86         // 객체를 요청하면 원본을 주는 것이 아닌가?
87         // 원본이라면 위와 동일하게 출력되어야하는데
88         // 다르게 출력이 된다.
89         // cm.reRandArr();로 원본의 값이 바뀔건가?
90         // 값이 다르지만 객체를 요청한 위의 경우는 원본일까 복사본일까?
91
92         System.out.println("사전 저장 정보 출력");
93         System.out.printf("save[0] = %d, save[1] = %d, save[2] = %d\n",
94             save[0], save[1], save[2]);
95
96
97         // 결론: 자바에서 객체에 대한 접근은 모두 메모리를 제어하는 방식이 된다.
```

```
▶ ↑ "C:\Program Files\Java\jdk-15.0.2\bin\
↓ arr[0] = 1, arr[1] = 1, arr[2] = 5
arr[0] = 1, save[1] = 1, save[2] = 5
객체에 접근해 출력
arr[0] = 4, arr[1] = 5, arr[2] = 1
사전 저장 정보 출력
save[0] = 4, save[1] = 5, save[2] = 1
객체내 변수값 획득: 3
변경 후 사전 획득한 정보 재출력: 3
변경 정보 파악: 7
```


링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day14/src/MemoryCloneTest.java>

```

99      int num = cm.getCloneVariable();
100      //      public int getCloneVariable () {
101          //          return num; }
102      //      리턴받은 값이니 복사본인가 ?
103
104      System.out.println("객체 내 변수값 획득: " + num);
105      //      3 출력
106
107      cm.reNum();
108      //      위를 통해 num = 7;
109
110      System.out.println("변경 후 사전 획득한 정보 재출력: " + num);
111      //      여기서 num = cm.getCloneVariable()
112      //      값을 변경했으니 7이 나와야 할 것같은데 3이 나온다.
113
114
115      // 결론: 앞서서도 확인했지만 값에 대해서는 복제가 이루어짐을 확인할 수 있다.
116      System.out.println("변경 정보 파악: " + cm.getCloneVariable());
117      //      int num = cm.getCloneVariable(); 으로 부르고,
118      //      cm.getCloneVariable(); 으로 부르고 방식의 차이인지 위는 7이 출력된다.
119      }
120  }

```

변경 후 사전 획득한 정보 재출력: 3
변경 정보 파악: 7

```

125 // 배열 객체는 결국 메모리고 배열의 이름은 이 배열 객체의 대표로 전체 메모리를 전달하게 된다.
126 // int[] arr에서 arr은 배열 객체 전체
127 // arr[5] 와 같은 특정 인덱스 지정은 변수와 동일하게 값을 전달함
128
129
130 // 배열의 이름은 배열의 대표로 객체가 전달된다( 원본 형태로 전달)
131 // 배열 전달시 인덱스 지정하여 전달하면 값이 전달된다.
132 // 클래스를 메모리에 올린 객체를 전달하면 말 그대로 객체가 전달된다.
133 // 나머지는 전부 값이라 보면 되겠다.
134
135 // 원리 관점에서 바라보면 결국 메모리를 바라보는지,
136 // 메모리 내부에 박혀 있는 값을 보는것인지
137
138 // 객체(메모리)를 요청하면 원본을 줌
139 // 객체내에 들어있는 객체를 요청하면 원본을 줌 <<<<<<<<<----- 제일 주의해야하는 부분
140 // 객체내에 들어있는 값을 요청하면 이 값을 복사해서 줌
141 // 게터에서도 값을 리턴하는 것은 전부 복제인간이다.
142 // 게터에서도 객체를 리턴하는 것은 메모리 즉, 원본이다.
143 // 인간 vs 복제 인간
144 // 둘은 같은 인간일까요 ? 다른 인간일까요 ?
145 // DNA 정보는 같지만 나는 나, 너는 너 관점이라고 보면 되겠다

```

ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day14/src/ArrayListTest.java>

```
22 ▶ public class ArrayListTest {
23 ▶     public static void main(String[] args) {
24         // 용도: 일종의 배열임
25         // 배열의 사이즈를 지정하고 사용해야 하지만
26         // 이 녀석은 넣고 싶은대로 아무때나 막 넣어도 된다.
27         // (참고로 이 녀석도 Heap을 이용한 동적할당을 수행함)
28
29         // 사용법: ArrayList<내부에저장할데이터타입> 변수명 = new ArrayList<내부에저장할데이터타입>();
30
31         // 일반 배열과의 차이점은 ?
32         // 배열은 메모리가 연속적으로 배치된다.
33         // 이 녀석은 불연속 배치다.
34         // 어떻게 ?
35         // | 데이터1 | 다음링크 | ---> | 데이터2 | 다음링크 | ---> | 데이터3 | 다음링크 | ---> ....
36         // 배열은 ?
37         // | 데이터1 | 데이터2 | 데이터3 | 데이터4 | 데이터5 | 데이터6 | 데이터7 | ...
38         ArrayList<String> lists = new ArrayList<String>();
39
40         lists.add("빵");
41         lists.add("버터");
42         lists.add("우유");
43         lists.add("계란");
44         lists.add("쥬스");
45         lists.add("베이컨");
46         lists.add("파스타");
47         lists.add("비프샐러드");
48         lists.add("피자");
49
50         for (String list : lists) {
51             System.out.println("현재 항목은 = " + list);
52         }
53     }
54 }
```

Run: ArrayListTest ×

↑ "C:\Program Files\Java

현재 항목은 = 빵

현재 항목은 = 버터

현재 항목은 = 우유

현재 항목은 = 계란

현재 항목은 = 쥬스

현재 항목은 = 베이컨

현재 항목은 = 파스타

현재 항목은 = 비프샐러드

현재 항목은 = 피자

Q. 물건이 일렬로 있을 때와 아무 곳에나 있을 때 중에서
물건이 찾기 쉬운 경우는?

A. 일렬로 있을 때 찾기가 빠르다.(배열)

배열 :

- 메모리가 연속적으로 배치된다.
- 그러나 반드시 인덱스 값을 지정해줘야 한다.

ArrayList :

- 인덱스 값 지정이 안보이면서, 유연성이 극대화되었다.
- 그러나 메모리가 불연속적 = 속도가 느려진다.

EX_) 회원 가입 중, 배열 인덱스에 10을 설정하면,
10명만 회원가입이 가능하다.
그러나 회원이 몇 명 들어올 줄 모른다.
= 이럴 때 쓰기엔 ArrayList 이 좋다.

하지만, 선착순 10명만 받을 때라면, ArrayList 사용하면
좋지 않다.

- 내가 사용할 배열의 갯수가 명확하다 = 배열
- 내가 몇개를 사용하게 될지 모른다 = ArrayList

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day14/src/Quiz48.java>

```
112 //      당첨번호 3명출력
113 public void printSuccessArr () {
114     for (int i = 0; i < success.length; i++) {
115         System.out.printf("success[%d] = %d\n", i, success[i]);
116         System.out.printf("당첨되었습니다. %s 축하합니다 ^^ \n", tmpArr[success[i]]);
117     }
118 }
119
120 //      당첨자 인덱스에 들어있는 번호를 가진 사람을
121 //      shuffle 인덱스에서 찾아내어 출력하기
122
123 @Override
124 public String toString() {
125     return "Roulette{" +
126         "tmpIdx=" + Arrays.toString(tmpIdx) +
127         '}';
128 }
```

Quiz48 x

C:\Program Files\Java\jdk-15.0.2\bin\java.exe -javaagent:C:\Users\user\AppData\Local\Jet...
Roulette{tmpIdx=[0, 0]
Roulette{tmpIdx=[5, 22, 17, 10, 23, 24, 11, 4, 8, 2, 16, 18, 27, 26, 7, 12, 19, 1, 20, 15,
success[0] = 5
당첨되었습니다. 강병화 축하합니다 ^^
success[1] = 9
당첨되었습니다. 전승리 축하합니다 ^^
success[2] = 17
당첨되었습니다. 황정아 축하합니다 ^^

Process finished with exit code 0

기존 수업코드 복습하고 고민하다가 사람 이름은 출력했습니다.

근데 중복이 되는 걸 보니 랜덤 숫자에 대해서는
중복이 적용됐지만, 당첨번호 3개에 대해서는 중복방지 코드가
적용되지 않은 것 같아서 코드를 살펴보고 있었습니다..

복습 + 현재 코드까지 올리고 수업에서 풀이 듣도록 하겠습니다.

```
Quiz48
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"
Roulette{tmpIdx=[0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
Roulette{tmpIdx=[13, 11, 4, 9, 22, 23, 2, 26, 2
success[0] = 14
당첨되었습니다. 이태양 축하합니다 ^^
success[1] = 9
당첨되었습니다. 전승리 축하합니다 ^^
success[2] = 14
당첨되었습니다. 이태양 축하합니다 ^^

Process finished with exit code 0
```