

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

Array (배열) : stack & Heap 비교

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/AllocHeapArray.java>

< 기본 개념 >

stack : 정적 배열

stack에 할당한다는 것은 지역변수로 처리함을 의미한다.
매서드나 클래스에서 stack 할당 시,
해당 매서드나 클래스 내부에서만 배열이 활성화 된다.
new를 사용하지 않는 정적 배열이며, 미리 준비를 할 수 있다.

제약 : 개수가 할당된 값으로 고정된다는 제약이 있다.

배열 만드는 방법 : **stack**(지역 변수) 에 할당하는 방법

데이터 타입 변수명 [] =
int arr [] 로 만든다.

int arr[] = { 1, 2, 3, 4, 5 }; **// 할당된 값이 있음**
위 데이터는 아래와 같은 형식으로 저장된다.

```
-----  
arr  | 1 | 2 | 3 | 4 | 5 |  
-----
```

인덱스(방) [0] [1] [2] [3] [4]

배열의 인덱스(방) 번호는 0번부터 시작한다.

arr[0] = 1, arr[1] = 2, arr[2] = 3, arr[3] = 4, arr[4] = 5

< 기본 개념 >

Heap : 동적 배열

동적할당 되는 데이터를 관리하는 메모리를 말하며,
new 로 만든 데이터는 전부 Heap에서 관리한다.
Heap은 공간을 필요한만큼 할당할 수 있어 유연한 접근이
가능하다
new를 사용하는 동적 배열이며, 프로그램이 실행되는 도중에
만들어져서 미리 준비를 할 수 없고, 그에 stack보다 느리다.

제약 : stack보다 느리다.

새로운 개념 new를 통해 공간을 만드는 방법

데이터 타입 변수명[] = new 데이터타입 [개수];
(만약 데이터타입이 class라면 소괄호를 열고 필요하다면 인자 설정)

```
int studentArr[] = new int [studentNum];
```

Heap 공간에 int형 배열을 studentNum 개수만큼 만들겠습니다.
그리고 studentArr는 Heap에 생성된 공간을 제어한다.

Array Length : 배열 길이 구하기

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/ArrayLengthTest.java>

```
1 public class Test {
2     public static void main(String[] args) {
3         ①
4         int arr[] = { 2, 3, 4, 5, 6, 7 };
5
6         System.out.println("arr의 길이 = " + arr.length);
7
8         ②
9         int dynamicArr[] = new int [30];
10
11         System.out.println("dynamicArr의 길이 = " + dynamicArr.length);
12     }
13 }
14
```

```
Run: Day7_Quiz1 x ArrayLengthTest x
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"
arr의 길이 = 6
dynamicArr의 길이 = 30
Process finished with exit code 0
```

Length : 길이를 구하는 것이다.

① Stack 형 :

int arr[] = {2, 3, 4, 5, 6, 7}; 에 대해서,
Stack은 정적 배열로 이미 할당된 값이 준비되어 있다.
2, 3, 4, 5, 6, 7이 그 값이며 length는 6.
따라서 System.out.println("arr의 길이 = " + arr.length); 은
arr의 길이 = 6이 출력된다.

② Heap 형 :


데이터 타입 변수명[] = new 데이터타입 [개수]; 의 형태이다.
int dynamicArr[] = new int [30];
정수 dynamicArr이름의 변수는 heap형이며 30개의 정수 데이터를
가지고 있다.

System.out.println("dynamicArr의 길이 = " + dynamicArr.length); 은
dynamicArr의 길이 = 30이 출력된다.
(30개니깐 0 ~ 29라는 것을 주의해야한다.)

dynamicArr	1 2 3 4 5 ... 28 29 30
인덱스(방)	[0] [1] [2] [3] [4] ... [27] [28] [28]

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/ClassTest.java>

```
1 class Person {
2     int age;
3     String name;
4
5 }
6
7 public class Test {
8     public static void main(String[] args) {
9
10        Person person = new Person();
11
12        person.age = 21;
13        person.name = "안녀영";
14
15        System.out.println("나이는 ? " + person.age);
16        System.out.println("이름은 ? " + person.name);
17    }
18 }
```



Run: Test x

```

C:\Program Files\Java\jdk-15.0.2\b
나이는 ? 21
이름은 ? 안녕영
Process finished with exit code 0

```

class :

클래스는 사용자가 직접 만들 수 있는 커스텀 가능의 데이터 타입이다.

클래스를 추가하기 위해서는 상단에 class 클래스이름 { }을 생성한다.
(클래스 이름은 대문자로 시작한다.)

이는 Person 란 이름을 갖고, 형태를 가진 공간이 생긴 것이다.

class로 변수 만들기 :

위에 만들어진 클래스를 통해 변수를 만들어 보자.

```
Person person = new Person();
```

변수 이름이 person인 Person 형태의 텅 빈 공간이 만들어진다.

데이터 타입이 클래스일 경우 소괄호를 사용한다.

person.age를 통해 클래스 Person의 int age를 부르고, 그 값은 21로 설정한다.

person.name를 통해 클래스 Person의 String name을 부르고, 그 값은 안녕으로 설정한다.

(‘.’은 공간 내부에 접근하겠다는 의미이다.)

class Method : (Getter / Setter) - 1

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/ClassMethodTest.java>

```
1  class Teacher {
2      int age;
3      String name;
4      String major;
5
6      int getAge() {
7          return age;
8      }
9
10     void setAge(int age) {
11         this.age = age;
12     } this.age는 class Teacher의 int age를 의미한다.
13     age는 입력으로 들어온 (int age)를 의미한다.
14     String getName() {
15         return name;
16     }
17     void setName(String name) {
18         this.name = name;
19     }
20
21     String getMajor() {
22         return major;
23     }
24     void setMajor(String major) {
25         this.major = major;
26     }
27 }
28
```

< Getter / Setter 만들기 >

Getter : 값을 얻는 목적으로 사용하는 것.

Setter : 값을 설정하는 목적으로 사용하는 것.

a) Getter / Setter 자동으로 만들기 :

class 내부에 마우스 커서를 가져다 놓고 Alt + insert를 누른다.
Getter & Setter 를 선택하고 전부 ok하면 자동으로 만들어진다.

b) Getter / Setter 수동으로 만들기 :

매서드를 만드는 방법

1. 리턴(return) 타입을 작성한다.
2. 매서드의 이름을 작성한다. (용도에 맞게 작성한다.)
Getter은 값을 얻고자 할 때, (return의 용도)
Setter는 값을 설정하고자 할 때 사용한다.
(그 외, 커스텀 매서드의 경우 자동 완성 기능으로 만들 수 없기 때문에 매서드 작성법에 대해 알 필요가 있다.)
3. 소괄호 내부에 인자로 입력 받을 매개변수를 설정한다.
4. 중괄호 내부에 해당 매서드(기능)이 수행할 업무를 작성한다.

class Method : (Getter / Setter) - 2

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/ClassMethodTest.java>

```
29 public class Test {
30     public static void main(String[] args) {
31         ① Teacher t = new Teacher();
32
33         ② t.setAge(40);
34         t.setMajor("Physics");
35         t.setName("GAORICK");
36
37         ③ System.out.printf("%s 는 %s를 전공하였고 %d 세다.\n",
38             t.getName(), t.getMajor(), t.getAge());
39     }
40 }
41 }
```

```
Run: Test x
"C:\Program Files\Java\jdk-15.0.2\bin\
GAORICK 는 Physics를 전공하였고 40 세다.
Process finished with exit code 0
```

Getter : 값을 얻는 목적으로 사용하는 것.
Setter : 값을 설정하는 목적으로 사용하는 것.

- ① 위에 만들어진 클래스를 통해 변수를 만들어 보자.
Teacher t = new Teacher();
변수이름이 t인 Teacher 형태의 텅 빈 공간이 만들어 진다.
데이터 타입이 클래스일 경우 소괄호를 사용한다.
- ② 클래스 내부의 값을 Setter로 설정하고 값을 입력한다.
ex_ t.setAge(40);
t라는 클래스를 통해 만들어진 변수에서 age값을 설정한다.
setAge는 상단 void setAge(int age) {
 this.age = age; }
으로 Teacher 클래스 내부의 ing age값은 setAge로
입력된 값이 된다. 즉, 40이 된다.
- ③ 설정된 값을 Getter를 통해 얻는다.
ex_ System.out.printf("%s 는 %s를 전공하였고 %d 세다.\n",
 t.getName(), t.getMajor(), t.getAge());

(System.out.printf 사용시, %d는 int 출력, %s는 String 출력)

25번 : 복습 문제

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day6/src/Day6_Quiz1.java

```
1 public class Day6_Quiz1 {
2     public static void main(String[] args) {
3         // 25번
4         int sum = 0;
5
6         for (int i = 1; i <= 100; i++) {
7             if (i % 11 == 0 && i % 5 == 0 && i % 2 == 0) {
8                 ;
9                 // 110의 배수일때 아무것도 실행하지 않고,
10                // 110의 배수가 아니라면 아래를 실행한다.
11            } else if (i % 11 == 0 && i % 5 == 0) {
12                ;
13                // 55의 배수일때 아무것도 실행하지 않고,
14                // 55의 배수가 아니라면 (55의배수 제외) 아래를 실행한다.
15                // 55의 배수 : 55 제외
16            } else if (i % 11 == 0 && i % 2 == 0) {
17                ;
18                // 22의 배수일때 아무것도 실행하지 않고,
19                // 22의 배수가 아니라면 (22의배수 제외) 아래를 실행한다.
20                // 22의 배수 : 22 , 44 , 66 , 88 제외
21            } else if (i % 5 == 0 && i % 2 == 0) {
22                ;
23                // 10의 배수일때 아무것도 실행하지 않고,
24                // 10의 배수가 아니라면 (10의배수 제외) 아래를 실행한다.
25                // 10의 배수 : 10 , 20 , 30 , 40, 50, 60, 70, 80, 90, 100 제외
26
27                // 위 결과값으로 제외 된 숫자 : 10 , 20 , 22 , 30 , 40, 44, 50,
28                // 55, 60, 66, 70, 80, 88, 90, 100 이다.
29            }
```

```
30        } else if (i % 11 == 0) {
31            sum += i;
32            // 위에서 제외된 숫자 빼고 11의 배수일때 더한다.
33            sum = sum + i
34            sum = 0 + ( 11 + 33 + 77 + 99 )
35            sum = 220
36        } else if (i % 5 == 0) {
37            sum -= i;
38            // 위에서 제외된 숫자 빼고 5의 배수일때 뺀다.
39            sum = sum - i
40            sum = 220 - (5 + 15 + 25+ 35 + 45 + 65 + 75 + 85 + 95 )
41            sum = 220 - 445
42            sum = -225
43        } else if (i % 2 == 0) {
44            sum += i;
45            // 위에서 제외된 숫자 빼고 2의 배수일때 더한다.
46            sum = sum + i
47            sum = -225 + ( 2 + 4 + 6 + 8 + 12 + 14 + 16 + 18 + 24 + 26 + 28
48                + 32 + 34 + 36 + 38 + 42 + 46 + 48 + 52 + 54 + 56
49                + 58 + 62 + 64 + 68 + 72 + 74 + 76 + 78 + 82 + 84
50                + 86 + 92 + 94 + 96 + 98)
51            sum = -225 + 1780
52            sum = 1555
53        }
54    }
55
56    System.out.println("최종 결과 = " + sum);
57 }
58
59 }
```


27번 : 피보나치 수열 복습 문제

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day6/src/Day6_Quiz2.java

```
4 public void test() { int num = 3, res = 2; }
7 public static void main(String[] args) {
8     // 피보나치수열
9     // 사용자가 15를 입력하면 15번째 값을, 8을 입력하면 8번째 값을 구하도록 프로그래밍
10    System.out.println("사용자로부터 n을 입력받아 n 번째 피보나치 수열의 항을 구합니다.");
11
12    Scanner scan = new Scanner(System.in);
13    System.out.print("n 값을 입력하십시오: ");
14
15    int num = scan.nextInt();
16    int res = 0;
17
18    // 예외 처리(음수와 0)
19    if (num <= 0) {
20        System.out.println("0번째 항 혹은 음수 항은 존재하지 않습니다.");
21    } else if (num < 3) { // 0이 아니며 음수가 아니고 3미만이라면 1번째와 2번째뿐
22        // 1번째와 2번째값은 무조건적으로 1에 해당함
23        System.out.println("당신이 찾는 값은 1입니다.");
24    } else {
25        // 위의 조건이 모두 만족되지 않는다는 것은 결론적으로 숫자 3보다 크다는 것을 의미함
26        int first = 1, second = 1;
27        // -2를 했던 이유는 시작할 때 first 값과 second 값 2개를 알고 시작했기 때문
28        ① for (int i = 2; i < num; i++) {
29            //for (int i = 0; i < num - 2; i++) {
30            res = first + second;
31            first = second;
32            second = res;
33        }
34
35        System.out.println("결과는 = " + res);
36        System.out.println("first = " + first);
37    }
```

27번 문제 내용 :

아래와 같은 형태의 숫자들이 있다.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... 피보나치수열
n을 입력하여 n번째 값을 구하도록 프로그래밍 해보자.

① for (int i = 2; i < num ; i++)

초기 first값은 1, second 값도 1이다.

그러나 for문 루프로 값이 계속해서 바뀌게 될 것이다.

res = first + second

결과값에 first + second값을 대입한다.

3번째 피보나치 수열의 값은 1+1 =2이다.

4번째 피보나치 수열의 값은

first 자리에 second값이 대입되고, second자리에 res값이
대입된다. 따라서 res = second + (3번째 피보나치 수열의 값)으로
res = 1 + 2 = 3

5번째 피보나치 수열의 값은

first 자리에 (3번째 피보나치 수열의 값) 값이 대입되고,
second자리에 (4번째 피보나치 수열의 값) 값이 대입된다.
따라서 res = (3번째 피보나치 수열의 값) + (4번째 피보나치 수열의
값) 으로 res = 2 + 3 = 5(반복)

28번 : Array 복습 문제

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/Day7_Quiz28.java

```
1 import java.util.Scanner;
2
3 public class Day7_Quiz28 {
4     public static void main(String[] args) {
5         System.out.println("1, 2, 4, 8, ... 1024, ...");
6
7         Scanner scan = new Scanner(System.in);
8
9         System.out.print("몇 번째 항을 구할까요 ? ");
10        int num = scan.nextInt();
11
12        if (num <= 0) {
13            System.out.println("잘못된 값을 입력하였습니다.");
14        } else if (num < 2) {
15            System.out.println("당신이 찾는 값은 1입니다.");
16        } else {
17            int numArr[] = new int[num];
18
19            numArr[0] = 1;
20
21            for (int i = 1; i < numArr.length; i++) {
22                numArr[i] = numArr[i - 1] * 2;
23                // Math.pow(A, B)는 A^B(A의 B승)을 계산한다.
24                // Math.pow는 double을 결과로 내놓기 때문에 강제로 int 타입으로 변형하였음
25                numArr[i] = (int) Math.pow(2, i);
26            }
27
28            System.out.printf("%d 번째 항은 = %d\n", num, numArr[num - 1]);
29        }
30    }
31 }
```

28번 문제 내용 :

아래와 같은 형태의 숫자들이 있다.

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, ...

n을 입력하여 n번째 값을 구하도록 프로그래밍 해보자.

for(초기화 코드 ; 조건식 코드 ; 증감식 코드)

for (int i = 1 ; i < numArr.length; i++)의 조건식 코드에서
왜 i < numArr.length일까요?

(length인 이유가 이해가 가지않습니다.)

32번 : 멍멍이 class 만들기 (Setter / Getter)

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/Day7_Quiz32.java

```
1 class Dog {
2     String name;
3     int age;
4     String gender;
5     int weight;
6
7     String getName() {
8         return name;
9     }
10    void setName(String name) {
11        this.name = name;
12    }
13    int getAge() {
14        return age;
15    }
16    void setAge(int age) {
17        this.age = age;
18    }
19    String getGender() {
20        return gender;
21    }
22    void setGender(String gender) {
23        this.gender = gender;
24    }
25    int getWeight() {
26        return weight;
27    }
28    void setWeight(int weight) {
29        this.weight = weight;
30    }
31 }
```

32번 문제 내용 :

앞서 풀어봤던 예제인 class 문제를 참고하여 멍멍이 클래스를 만들어 보자.

```
33 public class Day7_Quiz32 {
34     public static void main(String[] args) {
35
36         Dog dog = new Dog();
37
38         dog.setName("강아지");
39         dog.setAge(3);
40         dog.setGender("female");
41         dog.setWeight(15);
42
43         System.out.printf("%s는 %d살이고, 성별은 %s이다. 또한 몸무게는 %d Kg이다.\n",
44             dog.getName(), dog.getAge(), dog.getGender(), dog.getWeight());
45
46     }
47 }
48
49 }
```

Run: Day7_Quiz1 x Day7_Quiz32 x

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent
강아지는 3살이고, 성별은 female이다. 또한 몸무게는 15 Kg이다.

Process finished with exit code 0
```

33번 : 고양이 class 만들기 (Setter / Getter)

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day7/src/Day7_Quiz33.java

```
1 class Cat {
2     String name;
3     int age;
4     String gender;
5     String eyecolor;
6
7     String getName() {
8         return name;
9     }
10    void setName(String name) {
11        this.name = name;
12    }
13    int getAge() {
14        return age;
15    }
16    void setAge(int age) {
17        this.age = age;
18    }
19    String getGender() {
20        return gender;
21    }
22    void setGender(String gender) {
23        this.gender = gender;
24    }
25    String getEyecolor() {
26        return eyecolor;
27    }
28    void setEyecolor(String eyecolor){
29        this.eyecolor = eyecolor;
30    }
31 }
32 }
```

33번 문제 내용 :

앞서 풀어봤던 예제인 class 문제를 참고하여 고양이 클래스를 만들어 보자.

```
33 public class Day7_Quiz33 {
34     public static void main(String[] args) {
35         Cat cat = new Cat();
36
37         cat.setName("고양이");
38         cat.setAge(10);
39         cat.setGender("male");
40         cat.setEyecolor("green");
41
42         System.out.printf("%s는 %d살이고, 성별은 %s이다. 또한 %s색의 눈을 갖고있다.\n",
43             cat.getName(), cat.getAge(), cat.getGender(), cat.getEyecolor());
44     }
45 }
46 }
```

Run: Day7_Quiz1 x Day7_Quiz33 x

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:
고양이는 10살이고, 성별은 male이다. 또한 green색의 눈을 갖고있다.

Process finished with exit code 0
```