

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

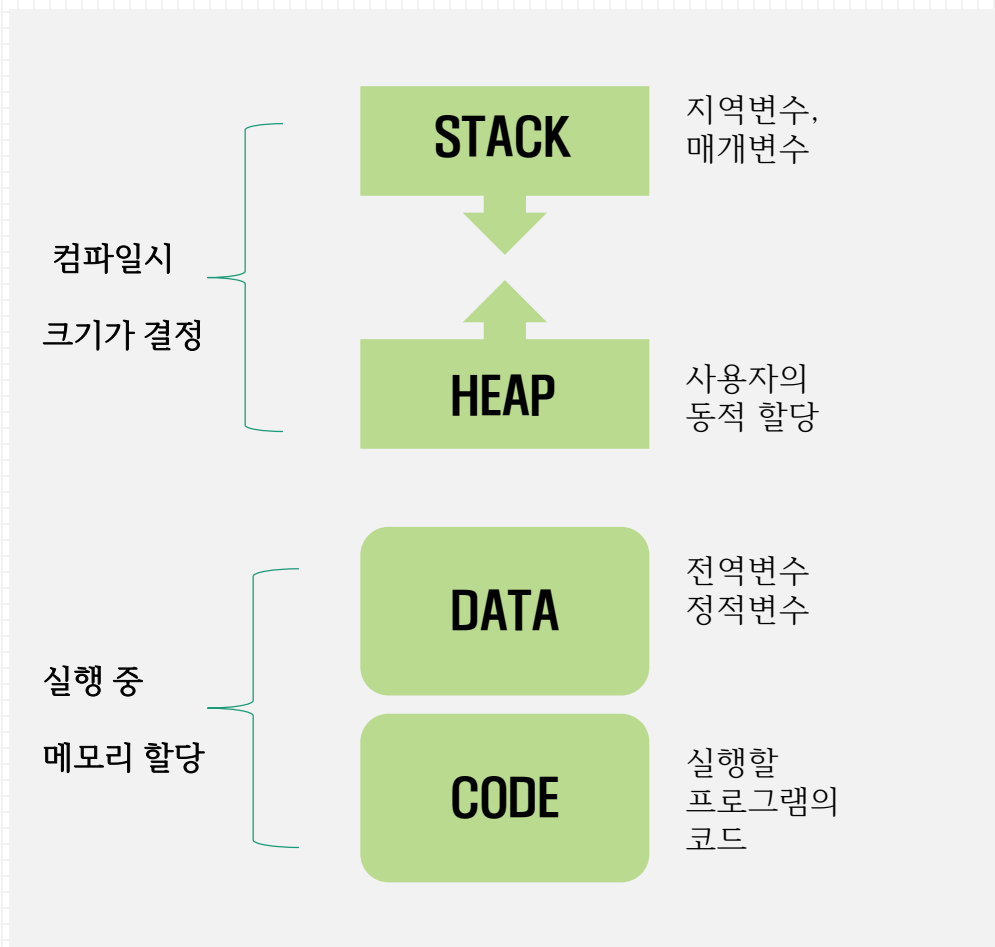
강사 : 이상훈

학생 : 임초롱

Static

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

< 메모리 구조 >



변수는 유효범위에 따라 전역변수와 지역변수로 구분할 수 있다.

- 전역변수는 함수 외부에서 선언된 변수로, 프로그램 전체에서 접근할 수 있는 변수이다.
- 지역변수는 함수 내부에서 선언된 변수로, 함수가 실행되면 만들어지고 함수가 종료되면 소멸하는 변수이다. 함수 외부에서는 접근할 수 없다.

Static :

- 정적 변수로 메모리 레이아웃에서 DATA 섹션이다.
- DATA 섹션에서 잡히는 것은 전역변수로서 역할을 수행한다.
- 특정한 값을 공유해야 하는 경우 static 사용 시 메모리의 이점을 얻을 수 있다.
- new가 없어도 사용이 가능하다.
- 대표적으로 main 매서드가 new 없이 static이 붙어있다.

Static

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

```
56 ▶ public class StaticMethodTest {
57
58     static final Scanner scan = new Scanner(System.in);
59
60     public static String inputName() {
61         System.out.println("학생 이름을 입력하세요: ");
62         return scan.nextLine();
63     }
64     public static int inputYear() {
65         Boolean isTrue = false;
66         int num;
67
68         do{
69             System.out.println("몇 년치를 계산할까요?");
70             num = scan.nextInt();
71
72             if (num <= 0) {
73                 System.out.println("0보다 큰 숫자를 입력해주세요!");
74                 isTrue = true;
75             }
76         } while (!isTrue);
77
78         return num;
79     }
80
81 @ public static float[] calcEmployeeMean(Employee[] emp){
82     int[] pay;
83     int length = emp[0].getPay().length;
84     float mean[] = new float[length];
85
86     for(int i = 0 ; i < length; i++){
87         float sum = 0;
88
89         for(int j = 0; j < emp.length; j++) {
90             sum += emp[j].getYearPay(i);
91         }
92         mean[i] = sum / (float)emp.length;
93     }
94     return mean;
95 }
```

Static변수 (정적변수) :

- 메모리에 고정적으로 할당되어, 프로그램이 종료될 때 해제되는 변수이다.
- Static변수는 클래스 변수이다.
- 객체를 생성하지 않고도 Static 자원에 접근이 가능하다.

Static 매서드 (정적 매서드) :

- 객체의 생성 없이 호출이 가능하며, 객체에서는 호출이 불가능하다.
- Static 매서드 안에서는 인스턴스 변수 접근이 불가능하다.

Static 매서드의 대표적인 예 :

- `int num = (int)Math.pow(2,2);`
특정한 클래스에 소속될 필요없이 필요하면 항상 사용하는 경우.
이런 경우를 유틸리티 매서드라고 하는데,
유틸리티 매서드들은 static 매서드로 만들어서 사용하곤 한다.
필요하다면 클래스에 static 매서드를 만든다.

Ex_ `Math.exp()` ➔ 오일러 상수 e^x 표현 시

Ex_ `Math.pow()` ➔ x^y 표현

Math 라는 것이 수학 라이브러리 임을 알려주고,
뒤에 `exp`, `pow`, `sqrt` 등이 어떤 매서드인지 알려준다.

do ~ while 반복문

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

```
64 public static int inputYear() {  
65     Boolean isTrue = false;  
66     int num;  
67  
68     do{  
69         System.out.println("몇 년치를 계산할까요?");  
70         num = scan.nextInt();  
71  
72         if (num <= 0) {  
73             System.out.println("0보다 큰 숫자를 입력해주세요!");  
74             isTrue = true;  
75         }  
76     } while (isTrue);  
77  
78     return num;  
79 }  
80
```

do while 반복문 :

while문은 조건식이 참일 때 코드가 실행되는 반면,
do while 반복문은 무조건 한 번 이상 실행된다.
do를 실행하고 while이 true일때 반복된다.

Do가 실행되어,
System.out.println("몇 년치를 계산할까요?");
Num = scan.nextInt();
몇 년치를 계산할까요? 를 출력하고 키보드 값을 받는다.

입력값이 0 또는 음수일때,
System.out.println("0보다 큰 숫자를 입력해주세요!"); 출력되며,
isTrue = true;가 되어 while 반복문이 true가 된다.
다시 do코드로 진입한다.

입력값이 0 또는 음수가 아닐때,
While (isTrue), isTrue가 false 이기때문에 반복문에서 벗어난다.

return num;

43번 : 클래스 배열 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

```
1 // 클래스 배열을 사용하므로
2 // 1. 직원 정보에는 이름과 자신의 연봉이 기록됨
3 // 여기서 연봉은 배열로 작성하여 연별값을 얻을 수 있어야함
4 // 2. 초봉 설정은 2400 ~ 3500만원
5 // 3. 매년 인상은 변동하여 1 ~ 20% (데이터가 궁금할 수 있으므로 배열로 관리)
6 // 4. 각각의 연봉 정보(클래스 배열들)에 대한 합산 결과가 요구됨 = 연별 평균
7 // 새로운 개념인 유틸리티 매서드 (static 매서드)를 사용
8 // 5. Scanner 사용이 필요하므로 학생 이름이 키보드로 들어옴
9 // 6. 몇 명을 입력할 것인지에 대한 정보가 추가로 필요함
10
11 import java.util.Scanner;
12
13 class Employee{
14     final float PERCENT = 0.01f;
15
16     String name;
17     int[] pay; //연봉
18     float[] incRate; //연봉인상률
19
20     int curYear; //현재 년차
21
22     public Employee(String name, int year) {
23         this.name = name;
24         curYear = 0; //신입일때(초봉)
25
26         pay = new int[year];
27         int pay[] = new int[year]
28         int year만큼 pay 인덱스가 생긴다.
```

```
31 incRate = new float[year];
32 in incRate[] = new int[year]
33 int year만큼 incRate 인덱스가 생긴다.
34
35 pay[0] = (int)(Math.random() * 1101 + 2400);
36 // 초봉 랜덤값 (연봉상승률은 포함되지 않기에 pay[0] 값을 초기설정한다)
37 }
38
39 public void calcYearPay() {
40     for (int i = curYear + 1; i < pay.length; i++) {
41         curYear = 0;
42         pay[0] 으로 초기설정 되어 있어 i + 1 해준다.
43         pay[i] = pay[i - 1] + (int)(pay[i - 1] * getRandomIncRate() * PERCENT);
44         pay[1] = pay[0] + (int)(pay[0] * getRandomIncRate() * 0.01f)
45     }
46 }
47
48 // Convention(규약)
49 // 나중에 팀 프로젝트 할 때도 팀간에 컨벤션 설정이 중요
50 // 전부 다 가져오는 것은 표기를 어떻게 하자
51 // 날개로 가져오는 것은 표기를 어떻게 하자
52 // 위와 같은 규약에 따라 그냥 지금 생각나는대로 작성한 부분
53 // calc_종류_변수 : 변수의 속성에 해당하는 종류 전체값을 전부 처리함
54 // calcYearPay : pay 변수의 속성 year에 해당하는 전체값을 모두 처리
55 // 이 규약은 만들기 나름이고 팀 프로젝트를 하는 경우엔 반드시 필요함
56 // (왜냐하면 내가 만든 매서드를 다른 사람이 사용할 수도 있기 때문)
57
58 // calcPay는 규약의 예시, 현재 코드에서는 사용되고 있지 않다.
59 public void calcPay(){
60     pay[curYear + 1] = pay[curYear] + (int)(pay[curYear] * getRandomIncRate() * PERCENT);
61     curYear++;
```

43번 : 클래스 배열 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

```
65 public void printPay() {
66     for(int i = 0 ; i < pay.length; i++){
67         // pay.length는 int year의 year.length값과 동일하다.
68         System.out.printf("%s님의 %d년차 연봉은 %d입니다.\n", name, i+1, pay[i]);
69     }
70 }
71
72 public float getRandomIncRate() {
73     return (float)(Math.random() * 20 + 1);
74 }
75
76 public int[] getPay() {
77     return pay;
78 }
79
80 public int getYearPay(int year){
81     return pay[year];
82 }
83 }
84
85
86 public class StaticMethodTest {
87     //static 매서드
88     static final Scanner scan = new Scanner(System.in);
89
90     public static String inputName() {
91         System.out.print("학생 이름을 입력하세요: ");
92         return scan.nextLine();
93     }
```

```
96 // public static void inputName() {
97 //     System.out.print("학생 이름을 입력하세요: ");
98 //     scan.nextLine();
99 // }
100
101 // void가 없었기때문에 return을 사용할 수 있었다.
102 // 만약 return이 아닌 scan.nextLine();으로 사용하고 void를 추가한다면
103 // 똑같이 잘 작동할까 ?
104 // 하단 부 144번에서 inputName() 오류 발생
105 // get이 쓰여져있지는 않지만, get의 역할을 하고있는 듯 하다.
106
107
108 public static int inputYear() {
109     Boolean isTrue = false;
110     int num;
111
112     do{
113         // while이 참일때 do가 반복된다.
114         System.out.print("몇 년치를 계산할까요?");
115         num = scan.nextInt();
116
117         // num값이 0 또는 음수일때, while이 참이되며 do가 반복된다.
118         if (num <= 0) {
119             System.out.println("0보다 큰 숫자를 입력해주세요!");
120             isTrue = true;
121         }
122         // num값이 0 또는 음수가 아닐때, while의 isTrue는 false이기 때문에
123         // 반복문은 끝난다.
124     } while (isTrue);
125 }
```

43번 : 클래스 배열 문제

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/StaticMethodTest.java>

```
126 // 또한 상단과 같이 void가 없기때문에 return을 받을 수 있으며,  
127 // 입력한 키보드값을 리턴한다.  
128 // 리턴 : 매서드 내에서 사용한 return은 return뒤에 따라오는 값을  
129 // 매서드의 결과로 반환한다.  
130 return num;  
131 }  
132  
133 @ public static float[] calcEmployeeMean(Employee[] emp){  
134 // calcEmployeeMean()가 float[] 형태이기 때문에 똑같은 형태의  
135 // 값을 받되, 그 값은 Employee[] emp이다.  
136 int[] pay;  
137 int length = emp[0].getPay().length;  
138 // emp는 메인코드에 작성되어 있다.  
139 // emp는 Employee 클래스 배열(studentNum.length만큼)  
140 // emp[0].getPay().length;  
141 // 0번째 인덱스 직원의 ~10년 월급의 길이 = 10  
142  
143 float mean[] = new float[length];  
144  
145 // 몇 년 간  
146 for(int i = 0 ; i < length; i++){  
147 float sum = 0;  
148  
149 // 사람 수  
150 for(int j = 0; j < emp.length; j++) {  
151 // j번째 사람의 i년차 연봉  
152 sum += emp[j].getYearPay(i);  
153 }  
154 mean[i] = sum / (float)emp.length;  
155 }  
156 return mean;  
157 }  
158 }
```

```
159 public static void main(String[] args) {  
160 final int YEAR = 10;  
161  
162 System.out.print("몇 명의 학생 정보를 입력할까요 ? ");  
163 int studentNum = scan.nextInt();  
164 scan.nextLine(); //오류 방지용  
165  
166 Employee[] emp = new Employee(studentNum);  
167 // Employee 클래스 배열 (studentNum.length만큼)  
168  
169 for(int i = 0; i < studentNum; i++) {  
170 emp[i] = new Employee(inputName(), YEAR);  
171  
172 emp[i].calcYearPay();  
173 emp[i].printPay();  
174 }  
175  
176 float[] mean = calcEmployeeMean(emp);  
177  
178 for(int i = 0 ; i < mean.length; i++) {  
179 System.out.printf("%d년차 평균 연봉은 %f\n", i+1, mean[i]);  
180 }  
181  
182 }  
183 }
```

```
몇 명의 학생 정보를 입력할까요 ? 3  
학생 이름을 입력하세요: 가가  
가가님의 1년차 연봉은 3364입니다.  
가가님의 2년차 연봉은 3760입니다.  
가가님의 3년차 연봉은 4442입니다.  
가가님의 4년차 연봉은 5200입니다.  
가가님의 5년차 연봉은 5344입니다.  
가가님의 6년차 연봉은 6309입니다.  
가가님의 7년차 연봉은 7485입니다.  
가가님의 8년차 연봉은 8535입니다.  
가가님의 9년차 연봉은 10001입니다.  
가가님의 10년차 연봉은 11447입니다.  
학생 이름을 입력하세요: |
```

```
다다님의 8년차 연봉은 7076입니다.  
다다님의 9년차 연봉은 7297입니다.  
다다님의 10년차 연봉은 8669입니다.  
1년차 평균 연봉은 3268.333252  
2년차 평균 연봉은 3744.333252  
3년차 평균 연봉은 4320.333496  
4년차 평균 연봉은 4713.000000  
5년차 평균 연봉은 5327.666504  
6년차 평균 연봉은 6117.333496  
7년차 평균 연봉은 6887.000000  
8년차 평균 연봉은 7339.333496  
9년차 평균 연봉은 8226.000000  
10년차 평균 연봉은 9463.333008
```


45번 : 종합 문제 (챌린지)

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/Quiz45.java>

```
1 // 1. 나와 상대방 수중에는 1000 만원이 있다.
2 // 2. 베팅 비율은 Scanner를 통해 입력받을 수 있다.(비율)
3 // 3. 2번의 주사위를 사용하는 게임이다
4 // 주사위 값의 합이 큰 사람이 이긴다.
5 // 4. 첫번째 주사위에서 짝수가 나왔을 때만 두번째 주사위를 굴릴 수 있다.
6 // 5. 두번째 주사위에서 1이 나오면 +3
7 // 3이 나오면 상대방 주사위 값 -2
8 // 4가 나오면 내 주사위 값이 0
9 // 6이 나오면 주사위 값의 합 *2
10 // 6. 경기가 질때마다 수중에 있는 돈은 베팅했던 만큼 줄어든다.
11 // 7. 상대방이 파산하면 이긴다.
12 // 승자와 패자가 있는 게임을 프로그래밍 해보자.
13
14 // 게임에서 이겼을때 베팅한 돈은 어떻게 되는건지?
15 // 컴퓨터의 베팅은?
16
17
18 import java.util.Scanner;
19
20 // 한 게이머에게 쓸 코드
21 class OneGamer{
22     // 한 게이머의 돈, 주사위, 베팅비율
23     String name;
24     int wallet; //수중 금액
25     int[] dice; // 주사위
26     int betRate; //베팅비율
27
28     public OneGamer(String name){
29         this.name = name;
30         wallet = 1000;
31         dice[0] = (int)(Math.random() * 6 + 1);
32     }
```

```
35 public void inputBetRate(int rateScan){
36     betRate = 1000 * rateScan;
37     // rateScan = 입력값으로 받은 베팅비율
38 }
39
40 public void clacDice() {
41     for (int i = 0; i < 2; i++) {
42         if (dice[0] % 2 != 0) {
43             System.out.printf("주사위의 값은 %d입니다. 홀수임으로 주사위를 더 던질 수 없습니다.", dice[0]);
44             break;
45         } else if (dice[0] % 2 == 0) {
46             System.out.printf("주사위 값은 %d입니다. 짝수임으로 주사위를 한번 더 던집니다.", dice[0]);
47         }
48
49         System.out.printf("첫번째 주사위 값은 %d, 두번째 주사위 값은 %d입니다.", dice[0], dice[1]);
50     }
51
52     //
53     if (dice[1] == 1) {
54         System.out.println("두번째 주사위 값이 1이므로, + 3 = " + (dice[0] + dice[1] + 3));
55     } else if (dice[1] == 4) {
56         System.out.println("두번째 주사위 값이 4이므로, 모든 주사위 값이 0이 되었습니다.");
57     } else if (dice[1] == 6) {
58         System.out.println("두번째 주사위 값이 6이므로, * 2 ! = " + ((dice[0] + dice[1]) * 2));
59     }
60
61     // 두번째 주사위 값이 3일때 상대방 주사위 값을 -2 한다는 것을 적을 수 없다.
62     // 위 계산은 여기서 진행되어야 하는것이 아닌 것 같다.
63 }
```


45번 : 종합 문제 (챌린지)

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day12/src/Quiz45.java>

```
65 public int getBetRate() {
66     return betRate;
67 }
68
69 public int getWallet() {
70     return wallet;
71 }
72
73 public int[] getDice() {
74     return dice;
75 }
76 }
77
78 // 두 게이머에게 모두 쓸 코드
79 public class Quiz45 {
80
81     static final Scanner scan = new Scanner(System.in);
82
83     public static int inputBetRate() {
84         System.out.println("베팅 비율을 적어주세요 : ");
85         return scan.nextInt();
86     }
87
88     public static void calcDiceGame() {
89         // 사용자의 베팅 비율 받기
90         // 컴퓨터의 주사위 던지기 + 특수 주사위 실행
91         // 사용자의 주사위 던지기 + 특수 주사위 실행
92         // 승자와 패자
93         // wallet이 0이 되는 게이머가 나타날때까지 반복
94         ;
95
96     }
```

```
99 // 메인 진행코드
100 public static void main(String[] args) {
101     String name[] = {"사용자", "컴퓨터"};
102     int len = name.length;
103     OneGamer[] od = new OneGamer[len];
104
105     for(int i = 0; i < len; i++) {
106         od[i] = new OneGamer(name[i]);
107     }
108
109
110     System.out.println("사용자와 컴퓨터의 주사위 게임을 시작합니다.");
111
112
113
114 }
115 }
```

45번 문제는 풀이하지 못했습니다.
막히면 막히는 대로 우선 제가 작성해봤던 코드들 올립니다.

오늘 풀이해주시는 것 듣고 좀 더 복습해서 풀이하도록 하겠습니다.