

01/21 [24회차] 수업 질문

12/20 개강 SW개발자 양성과정 [손현지]

2021. 01. 24

```
Bank6ProbIntegration.java x ThreadCalculation.java x
1 public class Bank6ProbIntegration {
2     public static void main(String[] args) throws InterruptedException {
3
4         //ThreadManager를 tm1로 객체화한다. (매개변수로 Rectangle Thread를 출력할 수 있게 해줌)
5         //realTotal = 2000 부터 sum = 0.00000000 까지 모두 객체화된 스레드 매니저를 통해 출력한다.
6         ThreadManager tm1 = new ThreadManager( threadNum: 0, parameter1: 2, parameter2: 2, ThreadCalculation.SQUARE);
7         //ThreadCalculation의 전역변수 SQUARE의 값을
8         // 입력하겠다는 뜻.
9
10        System.out.println("계산된 값 = " + tm1.calcArea());
11
12        // ThreadManager tm2 = new ThreadManager(4, 0, 3, ThreadCalculation.QUADRATIC);
13        // System.out.println("계산된 값 = " + tm2.calcArea());
14    }
15 }
```

<main>

```

7 private int threadNum;
8 private int serviceCode;
9
10 //스레드 매니저의 인수가 매개변수를 인식할 수 있도록 한다.
11 // 4, 2, 2, ThreadCalculation.SQUARE
12 public ThreadManager (int threadNum, int parameter1,
13                        int parameter2, int serviceCode) {
14
15     this.threadNum = threadNum;
16     this.serviceCode = serviceCode;
17
18     checkService(threadNum); //this threadNum으로 인식된 숫자를 체크 서비스 매개변수로 입력하여 실행
19     serviceAlloc(parameter1, parameter2); //parameter1, 2 값을 매개변수로 serviceAlloc을 실행
20 }
21
22 //ThreadManager내에서 확인된 thread Num 매개변수를 인수로 받으며 작동
23 public void checkService (int threadNum) {
24     switch (serviceCode) {
25         case ThreadCalculation.SQUARE: //ThreadCalculation의 전역변수 SQUARE의 값 = 0;
26             tRect = new ThreadRectangle[threadNum]; //배열 선언(threadNum값수만큼 생성됨)
27             break;
28
29         case ThreadCalculation.QUADRATIC:
30             tqe = new ThreadQuadraticEquation[threadNum]; //배열 선언(threadNum값수만큼 생성됨)
31             break;
32     }
33 }
```

<ThreadManager>

질문 1.

threadNum과 serviceCode는 this.으로 메서드
내에 값이 인식시키는데

threadNum과 마찬가지로 메서드에 매개변수로
인식시키는 parameter1, 2의 경우에는 왜 this.
를 해주지 않아도 괜찮은 건가요??

this.를 사용한 변수들의 경우는
ThreadManager에서만 인자로만 사용하는 것
뿐 아니라, ThreadManager class 내의 다른
메서드들에서도 사용해야 하기 때문에 이렇게
표시해주는 건가요?

```

114 public float calcArea () throws InterruptedException {
115     eachThreadStartWork(); //메서드 실행
116     eachThreadWaitFinish(); //메서드 실행
117     return sumEachThreadResult(); //sumEachThreadResults()메서드의 sum값을
118         //계산된 값 = 으로 return 함
119 }
120 }

```

```

60 public void eachThreadStartWork () {
61     switch (serviceCode) {
62         case ThreadCalculation.SQUARE:
63             for (int i = 0; i < threadNum; i++) {
64                 tRect[i].start(); //스레드 스타트
65             }
66             break;
67
68         case ThreadCalculation.QUADRATIC:
69             for (int i = 0; i < threadNum; i++) {
70                 tqe[i].start();
71             }
72             break;
73     }
74 }
75
76 public void eachThreadWaitFinish () throws InterruptedException {
77     switch (serviceCode) {
78         case ThreadCalculation.SQUARE:
79             for (int i = 0; i < threadNum; i++) {
80                 tRect[i].join(); // 메서드 종료 후에도 스레드 값이 사라지지 않도록 대기시킴
81             }
82             break;
83
84         case ThreadCalculation.QUADRATIC:
85             for (int i = 0; i < threadNum; i++) {
86                 tqe[i].join();
87             }
88             break;
89     }
90 }

```

<ThreadManager>

질문2.

eachThreadStartWork();와
eachThreadWaitFinish();는 tRect[i] 배열을 만들고
대기하는 과정인가요? tRect[i]에 getSum이라는값이
언제 들어간 건지 잘 모르겠습니다.

```

ThreadManager.java
92 public float sumEachThreadResult () {
93     float sum = 0;
94
95     switch (serviceCode) {
96         case ThreadCalculation.SQUARE:
97             for (int i = 0; i < threadNum; i++) {
98                 sum += tRect[i].getSum(); // tRect[i] 주소값마다 저장된
99             }
100             break;
101
102         case ThreadCalculation.QUADRATIC:
103             for (int i = 0; i < threadNum; i++) {
104                 sum += tqe[i].getSum();
105             }
106             break;
107     }
108
109     return sum;
110 }
111
112 public float calcArea () throws InterruptedException {
113     eachThreadStartWork(); //메서드 실행
114     eachThreadWaitFinish(); //메서드 실행
115     return sumEachThreadResult(); //sumEachThreadResults()메서드의 sum값을
116         //계산된 값 = 으로 return 함
117 }
118 }

```

그리고 ThreadCommon에서
getSum이 return하는 sum 값이 0 아닌가요?
어디에서 sum값이 변한건지...잘 모르겠어요.

문제은행 [7] 2번 문제

12/20 개강 SW개발자 양성과정 [손현지]

2021. 01. 25

```

28 //2. 3초마다 test라는 변수의 값을 1씩 증가시키도록 한다.
29 // 또한 /homework2 URL을 요청하면
30 // 현재의 test 값을 화면에 출력되게 만들어보자!
31
32 @RequestMapping("/homework2")
33 @Controller
34 public String homework2(Model model) throws InterruptedException {
35     logger.info("client entered / homework2");
36
37     int counter = 0;
38
39     while(true){
40         model.addAttribute("attributeName: ", counter);
41
42         //3초마다 +1씩 counter의 값 증가
43         Thread.sleep(millis: 3000);
44         counter++;
45         System.out.println(counter);
46
47         return "7th/homework2";
48     }
49 }

```

어차피 무한반복으로 3초마다 돌아가야 하는 거니까 while(true)를 사용해보려고 했음.
그렇지만 화면에 숫자는 0만 나오고 갱신이 되지 않음.

이 상태로는 반복문이 원래 작동되지 않는 건가요?
매서드화를 시켜야 했던 것인지...

```

homework2.html
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8
9     <h1> Counter Test</h1>
10    <p th:text = "${counter}>counter: ${counter}</p>
11
12 </body>
13 </html>

```

```

//timer.schedule()을 사용.
//schedule(TimerTask task, Date firstTime, long period)
//지정한 시간(firstTime) 부터 일정 간격(period)으로 지정한 작업(task)을 수행한다.
//0, 3초, counter++
/*

```

일정 간격으로 지정한 작업을 수행하도록 하는
방법으로
timer.schedule()이라는 것을 찾았지만 사용법을 몰라
응용은 하지 못했습니다.