

컴프2 10주차 : 경마 프로그램

과제 목표 : 스레드의 기본적인 사용 방법을 알고 코딩할 수 있다.

학번 : 201902722

이름 : 유형곤

큰 과제 : 스레드를 사용해 말이 서로 경주하는 것을 구현한다.

해결 과정의 흐름 : 과제에서 명시한 대로, 말을 표현하는 클래스를 구현한 뒤, 메인메서드에서 말을 선택하고 주행 결과를 출력하면 됩니다.

*내용이 길어서 다음 페이지에서 다시 작성하겠습니다.

작은 과제 : 말을 표현하는 클래스를 구현한다.

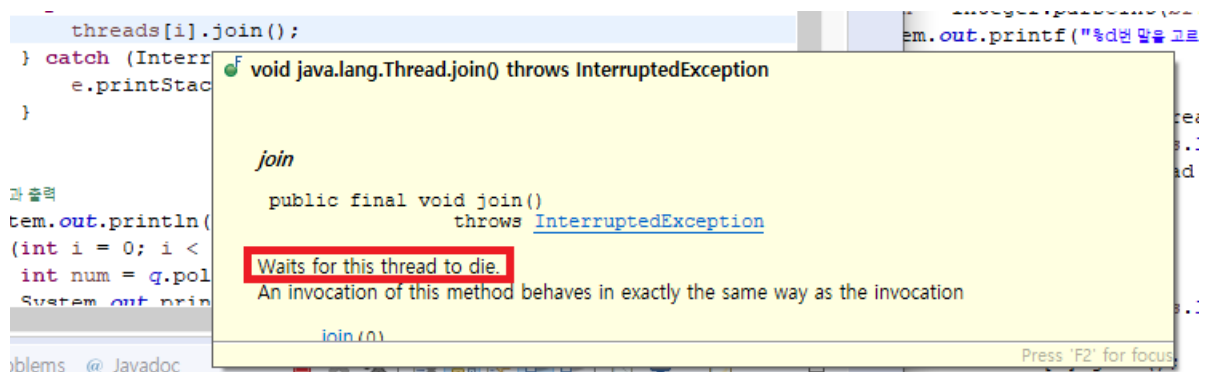
```
7
8 class HorseRunnable implements Runnable {
9     static int completed = 0;
10    int num;
11    int dist;
12    public HorseRunnable(int num) {
13        this.num = num;
14        this.dist = 0;
15    }
16
17    @Override
18    public void run() {
19        while(dist <= 100) { //100m를 넘으면 끝임 -> 100m를 초과해야 끝임, 그 전까지는 계속 달림
20            try {
21                int randInt = new Random().nextInt(11);
22                dist += (randInt);
23                int sleepTime = 1000;
24                if(new Random().nextInt(10) == 1) { //10% 확률
25                    sleepTime += 100;
26                }
27                System.out.printf("%d번 말은 지금 %dm에 있습니다.\n", num, dist);
28                Thread.sleep(sleepTime);
29            } catch (InterruptedException e) {
30                e.printStackTrace();
31            }
32        }
33        System.out.printf("%d번 말이 %d등으로 들어왔습니다.\n", num, completed+1);
34        Racing.q.add(num);
35        completed++;
36    }
```

우선 말을 표현하려면 말의 번호와 달린 거리를 변수로 저장해야 합니다. 그래서 각각의 변수와 생성자를 선언해주었습니다. 말은 1초마다 이동하는데, 10%의 확률로 0.1초를 더 쉬므로, 이동 거리와 이번에 쉬어야 하는지를 랜덤한 숫자를 뽑아서 정했습니다. 그리고 주행거리가 100m가 넘으면 Racing 클래스의 전역변수인 q (Queue<Integer> q)에 말의 번호를 저장합니다. 또한 HorseRunnable 클래스는 스레드로 동작해야 하므로 Runnable 인터페이스를 구현했습니다.

작은 과제 : 메인 메서드에서 말 선택하고, 결과 출력하기

```
39 public class Racing {
40     static Queue<Integer> q = new LinkedList<Integer>();
41     public static void main(String[] args) throws IOException {
42         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
43         System.out.println("5마리의 말이 100m 경주를 합니다.");
44         System.out.print("말 번호를 고르세요. (1~5): ");
45         int n = Integer.parseInt(br.readLine());
46         System.out.printf("%d번 말을 고르셨습니다.\n", n);
47
48         //1~5번 스레드 초기화 및 실행
49         Thread[] threads = new Thread[5];
50         for(int i = 0; i < threads.length; i++) {
51             threads[i] = new Thread(new HorseRunnable(i+1));
52             threads[i].start();
53         }
54
55         //모든 스레드가 끝날 때까지 대기한다.
56         for(int i = 0; i < threads.length; i++) {
57             try {
58                 threads[i].join();
59             } catch (InterruptedException e) {
60                 e.printStackTrace();
61             }
62         }
63     }
```

Racing 클래스에는 Queue<Integer> q (말이 완주한 순서를 저장)라는 정적변수와 메인메서드가 존재합니다. 메인메서드의 초반부에서는 말의 번호를 고르고, 말을 표현하는 스레드를 저장하기 위해서 스레드 배열을 선언하고 , 초기화한 후에 각각의 스레드를 실행합니다. 그 후 Thread 클래스의 join() 메서드를 활용하여 모든 스레드가 멈출 때까지 대기합니다. join() 메서드를 사용하는 근거는 다음 사진에 나와있습니다.



```

64         //결과 출력
65         System.out.println("### 경기 결과 ###");
66         for(int i = 0; i < threads.length; i++) {
67             int num = q.poll();
68             System.out.printf("[%d등] : %d번말", i+1, num);
69             if(num == n) {
70                 System.out.print("          <--- 당신이 고른 말");
71             }
72             System.out.println();
73         }
74     }
75 }
76

```

join() 메서드로 모든 스레드가 종료할 때 까지 대기했으니, 이제 결과를 출력하면 됩니다. 결과를 출력할 때에는 Queue<Integer> q에서 말 번호를 꺼내서 등수와 말 번호를 출력합니다. 이때 말 번호가 사용자가 선택한 말 번호라면 "< --- 당신이 고른 말" 이라는 문자열을 같이 출력합니다.

***실행 결과는 다음 페이지에 있습니다**

#실행결과

```
Console Problems Javadoc
Racing (1) [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (2019. 11. 12. 오후 4:13:45)
5마리의 말이 100m 경주를 합니다.
말 번호를 고르세요.(1~5): 3
3번 말을 고르셨습니다.
1번 말은 지금 7m에 있습니다.
2번 말은 지금 1m에 있습니다.
3번 말은 지금 6m에 있습니다.
4번 말은 지금 8m에 있습니다.
5번 말은 지금 4m에 있습니다.
1번 말은 지금 17m에 있습니다.
5번 말은 지금 5m에 있습니다.
3번 말은 지금 6m에 있습니다.
2번 말은 지금 6m에 있습니다.
4번 말은 지금 14m에 있습니다.
5번 말은 지금 10m에 있습니다.
4번 말은 지금 24m에 있습니다.
5번 말이 2등으로 들어왔습니다.
1번 말이 3등으로 들어왔습니다.
2번 말은 지금 86m에 있습니다.
3번 말은 지금 87m에 있습니다.
3번 말은 지금 95m에 있습니다.
2번 말은 지금 86m에 있습니다.
3번 말은 지금 97m에 있습니다.
2번 말은 지금 94m에 있습니다.
2번 말은 지금 103m에 있습니다.
3번 말은 지금 106m에 있습니다.
3번 말이 4등으로 들어왔습니다.
2번 말이 4등으로 들어왔습니다.
### 경기 결과 ###
[1등] : 4번 말
[2등] : 5번 말
[3등] : 1번 말
[4등] : 3번 말 <--- 당신이 고른 말
[5등] : 2번 말
```