

이상한 정렬

여러 복잡한 문제에 대해 동시에 처리하는 것은 유의미한 전략입니다. 프로그램 관점에서는 Thread로 구현할 수 있습니다.

이 문제에서는 실제로 복잡하지는 않지만 시간을 많이 소모하는 작업을 Thread.sleep() 함수를 이용해 모사합니다. 이 작업은 Worker 클래스에 이미 구현되어 있으며, Worker.run() 함수는 Worker.value의 값(millisecond)만큼 유휴 시간을 가진 후 value를 출력합니다.

value가 크면 클수록 긴 유휴 시간을 가지므로, 모든 value에 대해 Worker를 동시에 실행하면 출력은 value에 대해 오름차순으로 정렬됩니다. 이를 구현하는 것이 목적입니다.

- 문제

- (1) 주어진 Worker 클래스를 Thread로 실행할 수 있도록 변환하시오.
 - Thread 클래스를 상속받든 Runnable 인터페이스를 구현하든 무방합니다.
- (2) (1)에서 만든 클래스를 멀티스레드로 실행하시오.
- (3) NumberFormatException 클래스를 정의하시오.
- (4) input() 함수에 대해 NumberFormatException에 대한 처리를 구현하시오.

- 출력

8 800 600 700 400 500 200 300 100 100 200 300 400 500 600 700 800	4 1002 4000 60 8 NumberFormatException
----------------------------------------------------------------------------------------------	----------------------------------------------

- 주어진 코드

```
import java.util.Scanner;

// TODO 1 : make Thread or Runnable out of this class
// Hint : Thread class or Runnable interface
class Worker {

    private int value;

    public Worker(int value) {
        this.value = value;
    }

    public void run() {
        try {
            Thread.sleep(value);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println(value);
    }
}
```

```

}

// TODO 3 : Define NumberBoundException

public class Sandbox {

    public static void main(String[] args) {

        int[] arr = null;
        // TODO 4 : Catch NumberBoundException at input() function.
        // If exception occurred, Print 'NumberBoundException' and finish program
        // Hint : 'return' will finish main() function
        arr = input();

        for(Integer n : arr) {
            // TODO 2 : run Worker class
        }

    }

    public static int[] input() throws NumberBoundException {
        Scanner s = new Scanner(System.in);
        int N = s.nextInt();
        int[] arr = new int[N];
        for(int cnt = 0; cnt < N; cnt++) {
            arr[cnt] = s.nextInt();
            if(arr[cnt] > 1000) throw new NumberBoundException();
        }
        s.close();

        return arr;
    }

}

```