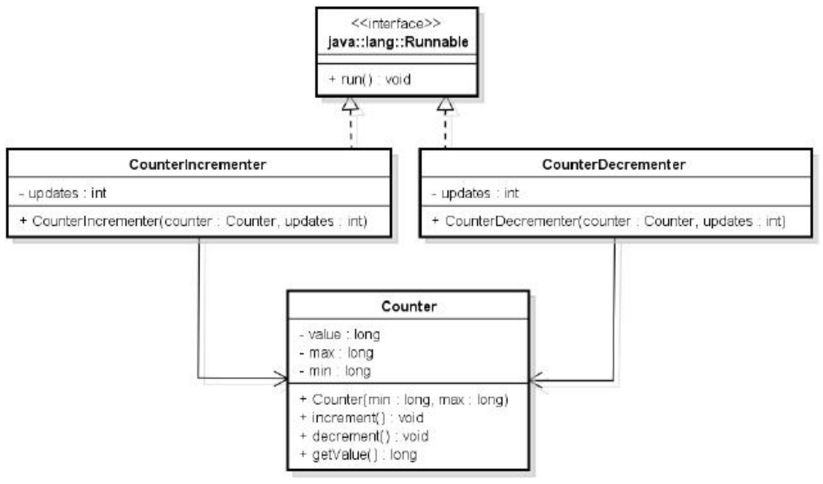
Exercises Threads2 SDJ2

Exercise: Counter increment/decrement

Implement the following system (see below)



A class `Counter` as a **Monitor** class (with private instance variables and all methods synchronized):

- A constructor setting value to 0 and min and max to whatever the values of the two arguments
- A method increment() incrementing the value by 1 (and let the calling thread wait if counter >= max)
- A method decrement() decrementing the value by 1 (and let the calling thread wait if counter <= min)
- A method getValue() returning the value

A class `CounterIncrementer` implementing `Runnable`. In the `run` method create a loop with `updates` loop cycles and call the `Counter` method `increment()` in the loop body. After the loop, print out the value of the counter.

A class `CounterDecrementer` which is almost the same as `CounterIncrementer`, except that this one calls `decrement()`.

Implement a class with a `main` method in which you create a `Counter` object, pass this to 2 `CounterIncrementer` objects and 2 `CounterDecrementer` objects (all with the second argument set to 200, i.e. 200 updates), create 4 threads with each of the 4 Runnable objects and start up the 4 threads.

…insert a few print-statements in class `Counter` to see when it is being updated (and by which thread), e.g. insert something similar to the following when `value` is updated and when a thread is blocked:

```
System.out.println(value + ": " + Thread.currentThread().getName());
```

Run the program a few times and inspect the output.