

# Assignments for week 9

Anders Degn Lapiki, Jacob Kjærulff Furberg, Jonas Ishøj Nielsen

29/10/2020

## Contents

<b>1</b>	<b>9.1</b>	<b>2</b>
1.1	9.1.1 - green . . . . .	2
1.2	9.1.2 - green . . . . .	2
1.3	9.1.3 - yellow . . . . .	2
<b>2</b>	<b>9.2</b>	<b>2</b>
2.1	9.2.1 - green . . . . .	2
2.2	9.2.2 - green . . . . .	2
2.3	9.2.3 - yellow . . . . .	3
2.4	9.2.4 - red . . . . .	3
<b>3</b>	<b>9.3</b>	<b>3</b>
3.1	9.3.1 - green . . . . .	3
3.2	9.3.2 - green . . . . .	3
3.3	9.3.3 - yellow . . . . .	3
<b>4</b>	<b>9.4</b>	<b>4</b>
4.1	9.4.1 - green . . . . .	4
4.2	9.4.2 - yellow . . . . .	4
<b>5</b>	<b>9.5</b>	<b>4</b>
5.1	9.5.1 - green . . . . .	4
5.2	9.5.2 - yellow . . . . .	4
5.3	9.5.3 - red . . . . .	4

## 1 9.1

### 1.1 9.1.1 - green

Changed from:

```
TimeUnit.Seconds.sleep(1);
```

To:

```
TimeUnit.MILLISECONDS.sleep(100);
```

Also updated updateTime function to take an int indicating how much faster a second is now. updateTime now also displays deciseconds. We also added a minor change to allzero to have the stopwatch start with an extra zero.

### 1.2 9.1.2 - green

Done. Moved original code from Stopwatch from main into a static method which Stopwatch2 calls.

Each UI element in stopwatchUI are now stored in array of size n. When creating a stopwatchUI n is given as a parameter and the arrays are instantiated. For each index  $i < n$  we do exactly the same as the original constructor, but with each element shifted  $i*100$  to the right. The method updateTime now also takes an index, so each stopwatch works independently.

### 1.3 9.1.3 - yellow

Instead of  $n=2$ , we use  $n=14$ .



## 2 9.2

### 2.1 9.2.1 - green

Done

### 2.2 9.2.2 - green

Ui now have an extra constructor taking the buffer. If the buffer is given it will continue to listen in a new thread for messages. The messages contain parameters for updateTime, so instead of stopWatch calling the function, the function is called when receiving a message.

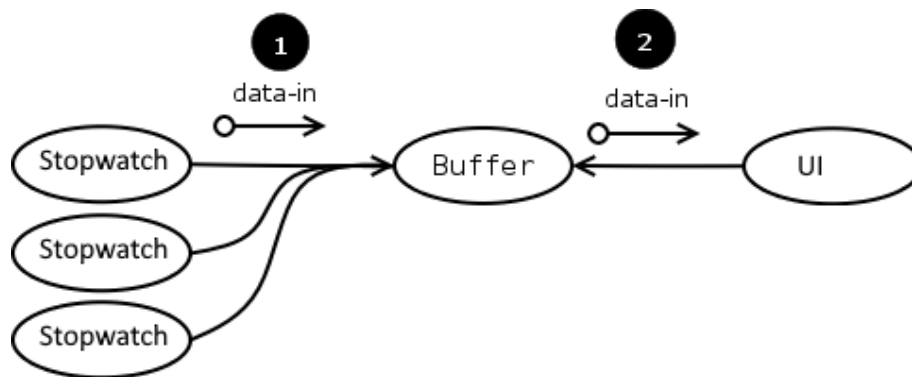
One class is responsible for building all the stopwatches, and each message indicates which stopwatch to update. With this done, yellow and red for 9.2 was finished also.

### 2.3 9.2.3 - yellow

Done

### 2.4 9.2.4 - red

Done



## 3 9.3

### 3.1 9.3.1 - green

Prints: "Hello World!"

### 3.2 9.3.2 - green

Prints: "Hello World"

### 3.3 9.3.3 - yellow

Prints at this time:

```
Processing Thread pool-1-thread-1
Processing Thread pool-3-thread-1
Receiver Thread pool-1-thread-1, Item length 1
Processing Thread pool-4-thread-1
Receiver Thread pool-4-thread-1, Item length 3
Receiver Thread pool-3-thread-1, Item length 2
```

Steps:

One observable emitting "A", "AB", "ABC".

Messages are flat mapped to observables that onNext sleep a random amount, and then print "processing thread "+ threadName and emitting the original message .length(), meaning the new emitted messages are 1, 2, 3 + the side effects.

The emitter is assigned one of 3 threads from a threadPool.

An observer is assigned/subscribed on an observable that calls the onNext, then gets the emitted length and prints observer thread-Name + the emitted length.

## **4 9.4**

### **4.1 9.4.1 - green**

Done

### **4.2 9.4.2 - yellow**

Done

## **5 9.5**

### **5.1 9.5.1 - green**

Done

### **5.2 9.5.2 - yellow**

Done

### **5.3 9.5.3 - red**

Done