Middle East Technical University | Department of Computer Engineering

CENG 443

Introduction to Object-Oriented Programming Languages and Systems

Spring 2022-2023

Lab 2 Quiz

(Ver 1.0)

Due date: May 17, 2023, Wednesday, 20:00

1 Introduction

Keywords: *Concurrency*

Your task is to implement a solution to Lab 2 Preliminary Question with some additions. The text will only explain the differences from the preliminary question. You can refer to the preliminary text for the rest.

2 Overview

There is a necessary item we omitted before: books where records are written. In addition to pens and ink bottles, your program also needs to manage books with concurrency primitives. So to write, a scribe needs **all three** items: a pen, an ink bottle and a book. After writing, a scribe should put all three items back until next time.

Number of scribes, pens, ink bottles and books will be given to your program as arguments:

```
java YourProgramName 5 2 4 1
```

will run your program with **5** scribes with **2** pens, **4** ink bottles and **1** book.

Your code should run correctly with any combination of inputs, be it with a single ink bottle or abundant number of books.

3 Example Output

First few lines of an output of the program with 5 scribes 2 pens, 4 bottles and 1 book can be seen below. Note that your app should run indefinitely until terminated:

```
Scribe 1 takes a pen
Scribe 1 takes a bottle
Scribe 1 takes a book
Scribe 5 takes a pen
```

```
Scribe 5 takes a bottle
Scribe 3 takes a bottle
Scribe 1 writes a record
Scribe 4 takes a bottle
Scribe 1 puts the book back
Scribe 5 puts the bottle back
Scribe 5 puts the pen back
Scribe 5 takes a pen
Scribe 5 takes a bottle
Scribe 5 takes a book
Scribe 1 puts the bottle back
Scribe 2 takes a bottle
Scribe 1 puts the pen back
Scribe 1 takes a pen
Scribe 5 writes a record
Scribe 3 puts the bottle back
Scribe 3 takes a bottle
Scribe 5 puts the book back
Scribe 4 puts the bottle back
Scribe 4 takes a bottle
Scribe 4 takes a book
Scribe 2 puts the bottle back
Scribe 2 takes a bottle
Scribe 5 puts the bottle back
Scribe 1 puts the pen back
Scribe 1 takes a pen
Scribe 1 takes a bottle
```

...

4 Some Remarks

- Your code should utilize concurrency
- Your output code should be correct. In other words
  - It should not be garbled.
  - It should not invalidate the specifications e.g. a scribe should not take 4th pen where there are only 3 pens.
  - Output format should match exactly with the example output e.g. if third scribe writes a record, your program should print "Scribe 3 writes a record" exactly.
- Every Scribe Class should be a separate thread and act independent of each other.
- Resources (pen, ink bottle and book) should be managed using locks and not atomics.
- At a single time step, a scribe can do single action: this action can be either taking an item, putting an item or writing a record. So something like taking all necessary items at once is not a valid solution.
- You may want to introduce a small random delay between operations for robustness using `Thread.sleep((int)(Math.random() * 10000))`

- You should make sure that no deadlock, no busy-wait and no starvation (like some scribes not being able to manage to write in a reasonable time) will occur in runtime.
- Your solution will be graded as a whole: If your code only solves the preliminary question, but not the quiz question, you will get 0.
- Use Java 8 or a newer version