## Declaration on Plagiarism

*This form must be filled in and completed by the student(s) submitting an assignment*

|  |  |
| --- | --- |
| **Name:** |  |
| **Student Number:** |  |
| **Programme:** | MCM |
| **Module Code:** | CA670 |
| **Assignment Title:** | Concurrent Programming |
| **Submission Date:** |  |
| **Module Coordinator:** | Dr. David Sinclair |

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found at <http://www.dcu.ie/info/regulations/plagiarism.shtml>, <https://www4.dcu.ie/students/az/plagiarism> and/or recommended in the assignment guidelines

Name:\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_ March 2020\_\_\_\_\_\_\_\_\_\_\_\_

## Project Description

1. Project overview

A small barber shop has a set of M barbers who spends all their lives serving customers, one at a time. Each barber has chair in which the customer sits when they are getting their hair cut. When there are no customers in the shop waiting for their hair to be cut, a barber sleeps in his chair. Customer arrive at random intervals. If a customer arrives and finds the a barber asleep, he awakens the barber, sits in the barber’s chair and sleeps while his hair is being cut. If a customer arrives and all the barbers are busy cutting hair, the customer goes asleep in waiting chairs. When the barber finishes cutting a customer’s hair, he awakens the customer and holds the exit door open for him. If there are any waiting customers, he awakens one and waits for the customer to sit in the barber's chair, otherwise he goes to sleep.

1. The Purpose of the Project

Implement the technology of thread concurrent operation. Using java technology to avoid deadlock, starvation and other non-ideal situations.

1. Design ideas

在单理发师问题中，理发师要先检查等待队列，当无客户在队列中时则出于等待状态（睡觉）。当用户进入店后唤醒理发师并理发。如果没有空闲的理发师，客户则进入等待队列。如果等待队列已满则客户离开，释放线程并计数。 多个睡眠理发师问题具有在等待客户中协调多个理发师的额外复杂性。 此时需要将客户与理发师进行绑定。每个理发师共享客户等待队列，但每个客户只能选择自己的理发师进行理发，避免出现多个理发师同时被一个客户唤醒的情况。为此，需要为客户和理发师额外添加id属性，并通过get(),set()来将他们之间绑定。理发师数量，客户数量，等待椅子数手动输, 用户的进入间隔和理发师的工作时间由random（）函数随机分配。主要有以下一些函数来实现整个问题的实现过程：

1. 用BarThread thread 控制理发师的行为。
2. 用CusThread thread 控制客户的行为。
3. visitShop()实现客户唤醒理发师并与之ID绑定。
4. leaveShop()实现客户理发并离开。
5. helloCustomer()实现理发师给客户理发，无客户则睡眠。
6. byeCustomer()实现理发师完成理发释放当前客户。

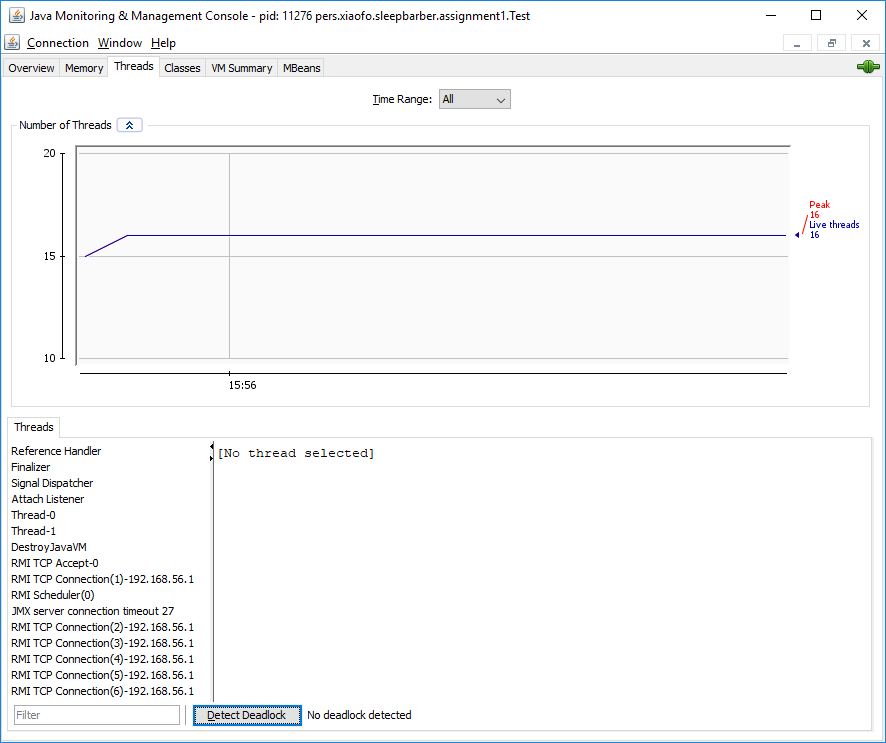
## Correctness and Fairness

1. Safety properties

创建理发师和客户队列，分别包含barId 和CustomerId属性。当客户唤醒理发师时都会获取当前被唤醒的理发师的Id, 同时这个理发师也对获得当前客户的Id. 避免客户唤醒多个理发师及一个理发师同时为多个客户理发的情况。 其次，将主线程的Lock在try{}catch{}块中进行，以保证线程无法运行时的异常处理机制。

1. Deadlock check

使用jdk自带的jconsole 工具检测程序死锁



线程0 和线程1 均无死锁被发现。

## Problem solution

在这次实践中，我的方案是使用java中lock的condition方法进行进程间的通信解决睡眠的理发师问题。客户进入理发店后先检查等待队列是否已满，如果满则线程死亡。如果不满则检查是否有空闲的理发师。如果没有空闲的理发师，则客户调用customer.Condition.await()并进入等待队列（椅子）等待。如果有空闲的理发师，则唤醒并开始理发。理发师在没有顾客的时候，调用自己的barber.Condition.await()。客户来的时候调用Barber.Condition.singalAll()唤醒barber并调用自己的customer.Condition.await()理发，同时，barber调用自己的barber.Condition.await()为客户理发;理发结束后，理发师通知客户理发完成并调用客户的customer.Condition.signalAll()唤醒客户并调用自己的barber.Condition.await()陷入等待。这时客户离开，调用理发师的barber.Condition.singalAll() 结束理发师的工作状态。这时检查等待队列，如果有客户在等待队列中，则理发师调用客户的customer.condition.signalAll()唤醒客户并开始理发，如果等待队列中没有客户则理发师调用自己的barber.condition.await()开始睡觉。