Tên: Phan Ngọc Hạnh Nhi

MSSV: 2131209002

**Practice Lab 4\_2**

For each of the following problems,

1. Identify the correctness constraints of the problem

2. Specify the conditions that each method must wait for

3. Write down the shared state that you will use to check these conditions

## **Assignment 1**

1. **Tạo một class để tạo hợp chất nước (H2O).**

* Chương trình có hai loại thread: **hydrogen (H)** và **oxygen (O)**.
* Khi có **2 nguyên tử hydro và 1 nguyên tử oxy**, chúng sẽ kết hợp lại để tạo thành **H₂O**.
* Nếu chưa đủ số lượng cần thiết, các thread phải **chờ**.

1. **Cơ chế hoạt động của hydrogen()**

* Nếu **số lượng hydrogen >= 1 và oxygen >= 1**, nghĩa là:
  + **1 hydro đã có sẵn + 1 hydro vừa gọi + 1 oxy đã có** → **Đủ điều kiện tạo thành H₂O**.
  + Khi đó, **giảm số H và O đi, sau đó tạo H₂O và đánh thức các thread khác**.
* Nếu chưa đủ điều kiện, **tăng số lượng hydrogen và cho thread H chờ**.

1. **Cơ chế hoạt động của oxygen()**

* Nếu **số lượng hydrogen >= 2**, nghĩa là:
  + **2 hydro đã có sẵn + 1 oxy vừa gọi** → **Đủ điều kiện tạo thành H₂O**.
  + Khi đó, **giảm số H và O đi, sau đó tạo H₂O và đánh thức các thread khác**.
* Nếu chưa đủ điều kiện, **tăng số lượng oxygen và cho thread O chờ**.

1. **Sử dụng số lượng hydrogen và oxygen để kiểm tra điều kiện chờ.**

* Khi một thread H hoặc O gọi phương thức, nó sẽ **kiểm tra số lượng H và O trước khi tiếp tục chạy**.
* Nếu đủ nguyên tử, **tạo H₂O và đánh thức các thread chờ**.
* Nếu chưa đủ, **thread sẽ đợi (await) cho đến khi có đủ số lượng**.

Lab 4.2: Thread Synchronization 2

**Assignment 1 (25 points) - Building H2O**

**1. Identify the correctness constraints of the problem**

The problem requires a synchronization mechanism that ensures:

* Correct molecular composition: A water molecule (H₂O) consists of exactly two hydrogen (H) and one oxygen (O) before any thread can proceed.
* Thread blocking until formation:
* Hydrogen threads must wait until two H and one O are available to proceed.
* Oxygen threads must wait until two H are present before proceeding.
* Fairness (optional): If fairness is enabled (fairness = true), thread execution should follow a first-come, first-served (FIFO) order, ensuring that no hydrogen or oxygen thread is unfairly delayed.
* Deadlock and starvation prevention:
* No thread should be left waiting indefinitely (deadlock).
* All waiting threads should eventually form H₂O and return (no starvation).

**2. Specify the conditions that each method must wait for**

Each method must wait for specific conditions before continuing:

* hydrogen() method (H) must wait until:
* There are fewer than two hydrogen atoms already waiting. If two hydrogen atoms are already present, any additional hydrogen must wait.
* There must be at least one oxygen present before two hydrogen atoms can proceed together.
* oxygen() method (O) must wait until:
* There are at least two hydrogen atoms available before it proceeds.
* It should not proceed alone—it must form a complete H₂O molecule before returning.

Once both conditions are met in hydrogen() and oxygen(), the threads should proceed to form a molecule and then reset their counts to allow new threads to start forming another H₂O.

**3. Write down the shared state that you will use to check these conditions**

The shared state consists of:

* Counters to track available atoms:
* hydroCount: The number of hydrogen threads waiting.
* oxyCount: The number of oxygen threads waiting.
* Locks and Conditions for Synchronization:
* A ReentrantLock (lock) to ensure thread-safe access to shared resources.
* A Condition (hydroCond) for hydrogen threads to wait when necessary.
* A Condition (oxyCond) for oxygen threads to wait when necessary.
* Fairness control:
* The fairness boolean flag ensures that threads are processed in FIFO order if enabled (new ReentrantLock(fairness)).