INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT074-3-2
CONCURRENT PROGRAMMING

APD2F2109CS(DA) / APU2F2109SE / APU2F2109CS(DA) /
APD2F2109CS / APD2F2109SE / APU2F2109CS

HAND OUT DATE: 11 MARCH 2022

HAND IN DATE: 22 APRIL 2022

WEIGHTAGE: 20%

INSTRUCTIONS TO CANDIDATES:

1. **Submit your assignment at the administrative counter.**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**
3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**
4. **Cases of plagiarism will be penalized.**
5. **The assignment should be bound in an appropriate style (comb bound or stapled).**
6. **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**
7. **You must obtain 50% overall to pass this module.**
8. **This report is Part 1 of 2 for the assignment.**

Table of Contents

1.0 Assumption

- There is only 1 runway for the whole airport, 2 Gates for planes to dock at, and no available waiting space for plane between landing and docking.
- There is only 1 refuelling truck, meaning refuelling is an exclusive to one plane at a time.
- The runway can only be used one at a time, so if currently a plane is landing, no plane can take off/land at the same time.
- The landing system should be first come first serve basis, so if two planes have arrived on the airspace on the same time, the first one arrived is give the permission to land first, while the latter should wait at the airspace until the runway is empty.
- If there is no plane currently at the gates, a plane can land, immediately go to one of the gates to dock, do all the procedures to make sure the plane can take off again, then it can move to the runway again for take-off.
- If there is one plane currently at the gates, a plane can land, go to the remaining gate to dock, do all the procedures during docking, but if the other plane is in the process of refuelling, the plane must wait for the refuelling process done before it can start its own refuelling process.
- If there are already two planes at the gates, and a new plane come and land, that would make a deadlock situation where no plane can leave the gate as the runway is occupied, no plane can enter the gate as it is full, and no new plane can arrive as the runway is full.
- So, if all gates are occupied, any coming planes should wait above the airport’s airspace, wait until of the plane in the gate finished and took off before it can land and go to the empty gate.
- All passengers are ready and accounted for, so no problem will occur during the passengers disembark/embark process.

2.0 Flow of Activities

2.1 General Overview

During the whole process in the airport, every plane should do all this process in the exact order:

1. Permission to Land
2. Landing
3. Coasting to the Gates
4. Connecting to the Gates
5. Passengers Disembark/Cleaning and Refilling Supplies/Mechanical Check
6. Passengers Embark/Refilling Fuel/Mechanical Check
7. Plane Coasting to the Runway
8. Plane Taking Off

Based on the criteria that should be fulfilled, the plane will be a class with a couple of attributes. The runway and gates will be another class, while the refuelling truck will be implemented into the gates class.

2.2 Plane Landing

When the plane first arrived at the airspace, it will request a permission to land. If the runway is empty and one or more gates are available, then the plane will get permission to land. The landing process should take around 3 seconds to complete. The plane will be inserted into the airspace Linked List when it entered the airspace/created. If the plane gets the permission to land, it will be taken out from the list. The runway will use lock method, so if the runway is being occupied, no other plane can access the runway.

2.3 Plane Docking at the Gate

After the plane has arrived, it will take some time before arriving at the gate. It will be assigned with the available gate beforehand, where it will rest for a while (wait function), until all process such as disembark, cleaning, mechanical check, and refilling supplies and fuel done.

2.4 Passengers Disembark/Cleaning and Refilling Supplies/Mechanical Check

While it is docked at the gate, passengers disembark, cleaning and refilling supplies, and mechanical check could be done at the same time. After all the process has been done, it could advance into the next one.

2.5 Passengers Embark/Refilling Fuel/Mechanical Check

After the previous procedure has been done, the next step is preparing for the next step. Passengers will enter the airplane, while the refuelling process will be done if the truck is available. If the mechanical check process has not yet been done, it can be continued during this step, where the notify function will be used to wake up the plane thread.

2.6 Plane Take Off

After the plane is ready to depart, the ATC will check if the runway is available. If it is available, then the plane can enter the runway where it will start the departure process. After the plane has taken off and exited the airspace, the connection between ATC and the mentioned plane will be disconnected.

3.0 Concurrency Concept

Java Concurrency refers to having two executions happening inside the same application. The executor of the application process is referred as Threads. One or more threads that working together is called multithreading, where it can better utilize the usage of a computer's CPU power, at the cost of application design becoming more complex.

There are multiple problems that occurred regularly in concurrent programming. The first one is interleaving, where multiple threads having access to the same data, causing a process to be skipped in the middle of each thread execution. (DataCadamia, n.d.) Interleaving is also known with the name race condition. The other main problem in concurrent programming is deadlock situation, where two or more threads are stuck and blocked from progressing or being released due to waiting for each other. (DataCadamia, n.d.)

The main concurrency concepts and methods that will be applied towards the problem are Locks and Synchronizations. Locks is an API in java.util.concurrent package where it is a more flexible and sophisticated version of synchronization compared to the synchronized block method.

Lock has two main functions to work, lock() and unlock() function. lock() will make a thread acquired the lock if it is available. If the lock is acquired, the thread can do its processes without any interruption from any other threads. After it is done, unlock() function will make the thread released the lock, allowing it to be grabbed by other threads. If a lock can is not acquired, the thread will be blocked. (Baeldung, Guide to java.util.concurrent.Locks, 2021)

Synchronizations refers to the synchronized block keyword, where it will mark a method as synchronized. This will cause the marked method to be accessible by only 1 thread at a time. This is one of the simplest ways to prevent race conditions from happening. Synchronizations itself have many forms of application, based about the problem. (Jenkov, 2020)

References

Baeldung. (22 December, 2021). *Guide to java.util.concurrent.Locks*. Retrieved from Baeldung: https://www.baeldung.com/java-concurrent-locks

Baeldung. (22 December, 2021). *Guide to the Synchronized Keyword in Java*. Retrieved from Baeldung: https://www.baeldung.com/java-synchronized

DataCadamia. (n.d.). *Concurrency - Deadlock*. Retrieved from DataCadamia: https://datacadamia.com/data/concurrency/deadlock

DataCadamia. (n.d.). *Concurrency - Thread Interference (Interleave on shared data)*. Retrieved from DataCadamia: https://datacadamia.com/data/concurrency/interference

Jenkov, J. (12 August, 2020). *Java Synchronized Blocks*. Retrieved from Jenkov: https://jenkov.com/tutorials/java-concurrency/synchronized.html