Due Date: 02.05.2021, 23:55

CENG 112 - Data Structures

Assignment 3: Operating System Process Management Simulation

In this homework, you are expected to implement a basic "Operating System Process Management Simulation" application using Java. This homework will cover the topics given below:

1. Linked Lists
2. List ADT (Abstract Data Type)
3. Priority Queue ADT (Abstract Data Type)

Assume that you are designing a process management system for an operating system. In this system, different processes with various priorities "request" computing time from the operating system. Only one process can be computed at a given time. In other words, there is no parallel computing.

There are three types of process in the system:

1. High → Priority: 1
2. Normal → Priority: 2
3. Low → Priority: 3

At any given time system can have different number of processes. The systems should keep all the processes in a list.

The Process class should implement the following IProcess interface:

```
public interface IProcess {

  public String getType();

  public int getPriority();

  public String toString();

  ...

}
```

You can assume that all processes request computation at the same time. The computations should be placed in a queue-based structure with **a FIFO approach with priorities.**

In this system, each "*computation*" has an occupation time that denotes how many nanoseconds the operating system will be occupied. Your application should keep statistics about the estimated waiting time for each *process*, and this waiting time is equal to the sum of occupation time of all **prior** *computations*.

An **outline** of the Computation class is given below:

```
public Computation {

 private int id;                        // unique computation id in [1,1000]

 private IProcess process;              // the process that makes computation request

 private int occupation;                // needed time for the computation

 public String toString();

 ... // Constructors, getter setter and
 other helper methods

 }
```

You are expected to create 3 simulations. Each simulation should have 3, 5, and 10 random processes respectively. Furthermore, for each simulation there should be a corresponding priority computation queue. The simulations should be connected to each other in a linked list fashion.

Your application should print some statistics at the end of each simulation which are:

- Simulation Number
- Representation of the Computation Queue
- Total number of computations for the simulation
- Total and average waiting times
- Total number of computations for each specific type of process
- Total and average waiting time for each specific type of process

Summary of the Requirements

1. Create 3 simulations (which are linked to each other) with 3,5, and 10 random processes.
2. Place processes in a list for each simulation.
3. Create a "computation queue" for each simulation in which each computation should have a random occupation time (between 1-10ns).
4. For each simulation, print statistics.

The output of your program for the first simulation should look something like this:

```
Simulation Number: 1
Computation Queue: P2,High,10ns ← P1,High,2ns  ← P3,Low,7ns
Total numbers of computations : 3

Total waiting time: 12
Average waiting time: 4

Total number of computations for High: 2
Total number of computations for Normal: 0
Total number of computations for Low: 1

Total waiting time for High: 10
Average waiting time for High: 5

Total waiting time for Normal: 0
Average waiting time for Normal: 0

Total waiting time for Low: 12
Average waiting time for Low: 12

....
```

You should print this output for each simulation.

Assignment Rules

- This is a group assignment (2 students). However, inter-group collaboration is not allowed!
- All assignments are subject to plagiarism detection and the suspected violations (the solutions derived from or inspired by the solution of other groups) cause to be graded as zero.
- It's not allowed to use Java Collections Framework.
- Your code should be easy to read and test:
  - Keep your code clean. Avoid duplication and redundancy. 🔗

- Follow Java Naming Conventions.
- Use relative paths instead of absolute ones.

Submission Rules

All submissions must:

- be performed via **Microsoft Teams** by only one of the group members,
- be performed before the deadline,
- be exported as an Eclipse Project and saved in ZIP format,
- include all necessary data files (TXT, CSV, JSON, etc.) in the right directory,
- follow a specific naming convention such that CENG112_HW3_*groupID*.

Eclipse Project: CENG112_HW3_*G5*
Exported Archive File: CENG112_HW3_*G5*.zip

Submissions that do not comply with the rules above are penalized.

NOTE: Those who want to change groups can send their requests on Microsoft Teams.