Principle of Operating Systems (CSCI4340)
Synchronization in Java

Project Description

In this project, you will develop a multithreaded program for calculating the summation of elements in an array. To use multi-threaded programing, the array of integers is divided into two smaller sublists. Two separate threads add the elements in the first half and second half of the array to the summation. To demonstrate the benefits of synchronization for multi-threaded program, the program repeats the procedure for 100 times where the results should always be the same.

This program will require the two threads share the same variable of summation. Summing the elements of the sublists to a **local** variable can be concurrent. But adding the results to the summation should be synchronized.

Project learning Objectives

1. To expose you to the concepts of synchronization in Java
2. To enhance your skills of analyzing problems and designing multithreaded programs.

Program Structure

The program consists of the following Classes:

1. **`SynchronizationDemo:`**
   `data field: sumTotal`
   `main` method performs the following tasks:
   1) Generate an integer array whose 100 elements are 1,2, 3,... 100
   2) Repeat the following 100 times:
      i. Create two threads for the class **`Sum`** and use them to calculate the summation of the 100 elements. The first thread adds the first 50 elements to **sumTotal**, and the second thread adds the second 50 elements to **sumTotal**.
      ii. Display **sumTotal**.
2. **`Sum:`** an **inner class** that implements Runnable interface. `run` method performs the following tasks:
   1) Calculate the total of the elements in the array.
   2) Call `Thread.sleep(5)` to pause the thread
   3) Add the total to **sumTotal**. (this part must be synchronized)

The following are sample runs of the program:

With synchronization

```
Loop = 0    sum = 5050
Loop = 1    sum = 5050
Loop = 2    sum = 5050
Loop = 3    sum = 5050
```

Loop = 4 sum = 5050
Loop = 5 sum = 5050
Loop = 6 sum = 5050
Loop = 7 sum = 5050
Loop = 8 sum = 5050
Loop = 9 sum = 5050
…

Loop = 94 sum = 5050
Loop = 95 sum = 5050
Loop = 96 sum = 5050
Loop = 97 sum = 5050
Loop = 98 sum = 5050
Loop = 99 sum = 5050

Without synchronization

Loop = 0 sum = 1326
Loop = 1 sum = 5050
Loop = 2 sum = 1326
Loop = 3 sum = 5050
Loop = 4 sum = 5050
Loop = 5 sum = 5050
Loop = 6 sum = 3724
Loop = 7 sum = 5050
Loop = 8 sum = 5050
…

Loop = 94 sum = 5050
Loop = 95 sum = 5050
Loop = 96 sum = 5050
Loop = 97 sum = 5050
Loop = 98 sum = 5050
Loop = 99 sum = 5050

Program Documentation

The program should have appropriate documentation.

Deliverable

program source code .java file