TD: threads

Exercice 1. Compteurs

Un compteur a un nom (une chaîne de caractère) et compte de 0 à un entier n en marquant une pause aléatoire (Math.random() renvoit un flottant entre 0 et 1) entre chaque itération.

a. Écrire la classe compteur qui affiche chaque nombre compté, et un message final sous la forme `Le compteur Toto a fini de compter jusqu'à 12`. Écrire aussi un programme lançant une instance illustrant l'utilisation de plusieurs compteurs simultanément.

b. Modifier cette classe pour que chaque compteur affiche son ordre d'arrivée au bout du décompte. `Le compteur CompteurX a fini de compter jusqu'à 12 en position 3`.

c. Est-ce que vous pouvez garantir que votre programme s'exécutera proprement ? Donner un exemple de scénario où le comportement est non conforme.

d. Proposer une modification de votre code permettant d'éviter ce problème en utilisant les attributs `synchronized`

Exercice 2. Recherche de nombres premiers

On souhaite écrire un programme recherchant un nombre premier ayant au moins un certain nombre b de bits. Pour cela, on va énumérer les nombres entiers impairs plus grands que 2^b et tester leur primalité en testant s'il sont divisible par tous les entiers inférieurs à leur racine carré (algorithme peu efficace).

On souhaite par ailleurs exploiter le parallélisme de la machine hôte en utilisant des threads : on lancera ainsi un nombre fixe de threads testant chacun un intervalle de nombre.

Enfin, il faut que dès qu'un thread a trouvé un nombre premier, il le signale aux autres et les fasse s'arrêter.

Écrire ce programme, en le structurant en classes utilisant les Threads Java.