

Examen de Langage Java

Durée : 0h30

Aucun document autorisé

Mobiles interdits *Note 1 : le sujet de l'exercice qui suit est tiré du site de M. S. Perennes (www-sop.inria.fr). Note 2 : la qualité des commentaires, avec notamment la présence d'affirmations significatives, ainsi que les noms donnés aux variables, et la bonne indentation rentreront pour une part importante dans l'appréciation du travail.*

-
- 1. Écrivez une application Java qui, dans sa méthode `main`, crée deux threads. Le premier incrémente de 1 dix millions de fois un compteur (une variable entière) initialisée à 0. Le deuxième thread décrémente de façon concurrente de 1 dix millions de fois ce même compteur.

Une fois l'exécution des deux threads achevée, vous afficherez la valeur du compteur qui normalement doit être égale à 0.

```
/** @author: V. Granet (Vincent.Granet@univ-cotedazur.fr) */
public class Compteur {
    //ressource critique
    private int cpt = 0;
    //Rôle : incrémente le compteur de 1.
    public synchronized void incr() { this.cpt++; }
    //Rôle : décrémente le compteur de 1
    public synchronized void décr() { this.cpt--; }
    //
    public String toString() {return String.valueOf(this.cpt);}
    //
    public static void main(String [] args) {
        int max = 10000000; //dix millions
        Compteur cpt = new Compteur();
        //créer les 2 threads d'incrémementation et de décrémentation
        Thread incr = new Thread(() -> {for (int i=0; i<max; i++) cpt.incr();});
        Thread décr = new Thread(() -> {for (int i=0; i<max; i++) cpt.décr();});
        //démarrer les threads
        incr.start(); décr.start();
        //attendre la fin des threads
        try {
            incr.join();
            décr.join();
        }
        catch (InterruptedException e) { System.exit(1); }
        //afficher le compteur => 0
        System.out.println(cpt);
    }
}
```
