**TP : Java – Processus légers**

**La classe Cpt.**

* 1. Le nombre maximum de processus légers Co-existants au cours de l’exécution du Cpt est : 3 (main, thr1 et thr2).
  2. Le nombre d’incrémentation de la variable global \_cpt, par chaque processus léger est 100000.
  3. A la fin de l’exécution du code, la valeur affichée est 0.
  4. Deux threads thr1 et thr2 qui utilisent une variable commune \_cpt.  
     Les deux threads incrémente \_cpt.

**La classe Cptb.**

a. Thr1.run () et Thr2.start () :

Thr1.run () : Ne crée pas de nouveau thread, elle elle donne le travail au thread au courant d’exécution « main ».

Thr2.start () : crée un nouveau thread, dans lequel ; elle exécute la méthode « run () ».

b. Le nombre maximum de processus légers Co-existants au cours de l’exécution du Cptb est : 2 (main et thr2).

c. Dans la classe Cptb, on n’a pas de variable partagée entre Thr1 et Thr2.

d. Le nombre d’incrémentation de la variable global \_cpt, par chaque processus léger est 100000.

e. La valeur affichée à la fin de l’exécution du code est 200000.

f. On a deux threads qui s’exécutent séparément. Résultat fixe = 200001.

**La classe Train**

Dans cette classe, on a deux threads qui s’exécutent en parallèle.   
Ce programme termine avant la fin des threads créés, et imprime « Fin de main ».

Dans cette classe, il y a appel à la méthode " sleep ()", qui fait attendre le thread en courant un nombre precise de millisecondes. Alors il y a une synchronisation entre les deux threads.