Laure VINCENT-CUAZ
Groupe TP2

TP2

Exercice 2

- Que fait le programme, que fait chaque thread ?

Le programme lance plusieurs threads dont le nombre est spécifié en paramètre et chacun effectue une incrémentation correspondant au numéro du thread jusqu'à atteindre une valeur maximale spécifiée en paramètre. De plus, avant chaque incrémentation, le thread effectue une pause de 0.2 seconde.

- L'affichage sur la sortie standard est-il déterministe ? Ou le semble-t-il seulement ?

L'affichage semble déterministe car lors du lancement de plusieurs threads, la valeur finale attendue n'est pas la même que celle spécifiée en paramètre.

- Quel est la principale ressource utilisée concurremment par les threads ?

La principale ressource utilisée est l'attribut de la classe nommé « max_value ».

Exercice 3

- Que fait le programme, que fait chaque thread ?

Ce programme lance deux threads : l'un incrémentant la valeur en attribut Compteur qui vaut 0 avec un pas de 1, l'autre décrémentant la valeur « notre_entier » avec un pas de 1.

- Quelle devrait être la valeur stockée par l'ObjetEntier nommé Compteur à la fin du calcul ?

Comme le programme incrémente autant la variable Compteur qu'il la décrémente, celle-ci devrait revenir à sa valeur initiale soit 0.

- Expliquer ce qui arrive, où se trouve la concurrence ?

Les deux threads lancés s'exécutent en même temps et non l'un après l'autre ce qui entraîne une concurrence.

- Comment pourrait-on faire pour que le calcul retourne 0 et cela de façon certaine ?

On pourrait synchroniser la méthode d'incrémentation de la classe ObjetEntier.

- Le programme principal peut-il détecter la terminaison des threads ?

Comme un message est affiché à la fin de l'exécution de la boucle d'incrémentation sur la variable Compteur, on peut détecter la terminaison de chacun des threads.

Exercice 4

- Quelle méthode pose problème dans Petit_Job ? La rendre synchrone.
- Observer le résultat ? Pourquoi le thread principal n'affiche-t-il pas 0 ?
- Ajouter une longue boucle (:math:`10^9 ` tours) juste avant la fin du programme principal et afficher à nouveau le compteur, que concluez-vous ?

Exercice 5

- Quel est le principal problème de cette implémentation ?

Exercice 6

- Reprendre encore une fois le code de l'exemple Petit_job.java, et demander au thread principal d'attendre que les threads t1, t2 aient terminé avant d'afficher la valeur du compteur. Le résultat est-il enfin fiable ?

Exercice 7

- Que se passerait-il si on ajoutait un thread TS avec D<C et A<D ?