Travaux Dirigés n°7

Java Avancé

—M1 Apprentissage—

Threads

Threads

`https://classroom.github.com/a/6GYqyuSE`

Rappel, dans Eclipse, pour passer d'un processus à un autre tournant en même temps, cliquer sur l'avant dernière flèche au dessus de la console (voir la capture d'écran). Pour stopper un processus, cliquer sur le carré rouge. Comme il reste au premier plan, cliquer sur la croix à côté du carré pour avoir la main sur les autres processus s'ils existent.

▸ Exercice 1. On commence doucement

1. Créer une méthode prenant en argument un entier id et retournant un `Runnable` (sous forme de classe anonyme) dont vous redéfinirez la méthode run afin qu'elle fasse une boucle infinie affichant "Thread id - i", où i est le nombre de fois où la boucle a été effectuée.
2. Créer 2 threads (prenant en argument du constructeur le Runnable créé par votre méthode précédente) avec id 1 et 2 et les lancer. Que remarquez-vous sur la valeur des compteurs de chaque thread ?

▸ Exercice 2. On continue doucement

On modifie le code de l'exercice précédent.

1. Modifiez votre code pour que le nombre de threads à lancer simultanément soit donné en argument (args) de votre programme.
2. On souhaite que l'utilisateur puisse interrompre un thread en particulier en entrant son ID au clavier (utiliser un `Scanner`). Modifier le code de vos threads pour que les affichages ne se fassent plus à chaque tour de boucle mais uniquement lorsqu'on sort de la boucle. Modifier la condition de la boucle pour qu'elle ne soit plus infinie mais lorsque le thread courrant est interrompu (visible par `Thread.currentThread().isInterrupted()`).
3. Votre programme doit s'arrêter lorsque tous les threads ont été interrompus.

▸ **Exercice 3. Variable commune**

Dans le code suivant, deux threads changent la même variable.

```
public class Oups {
  static int nb;
  public static void main(String[] args) {
    for(int i=0;i<2;i++) {
      final int id=i;
      new Thread(new Runnable() {
        public void run() {
          while(true) {
            nb = id;
            if(nb!=id)
              System.out.println("oups");
          }
        }
      }).start();
    }
  }
}
```

1. Exécuter le code précédent. Est-ce que l'affichage est "normal" ?
2. Pourquoi l'affichage n'évolue plus au bout d'un certain temps ?
3. Déclarer la variable nb en tant que `volatile`. Qu'est-ce que cela entraine ?