

Programmation avancée — Java

TP7 – Générateur multi-threadé

1 Introduction

Les threads sont couramment utilisés dans le développement. Ils servent à paralléliser les traitements pour gagner en vitesse de d'exécution et rendre plus réactifs les programmes. C'est donc une notion fondamentale à maîtriser pour tout développeur.

Ce TP consistera à créer un générateur automatique et multi-threadé de données. Nous allons illustrer cela avec des étudiants. Ces étudiant vont posséder plusieurs champs : identifiant national étudiant, nom, prénom, date de naissance, lieu de naissance et un petit descriptif de leur parcours.

La seule documentation autorisée est la documentation officielle Java <http://docs.oracle.com/javase/8/docs/api/> (pour le reste **vous me demandez**).

2 Travail à effectuer

1. Créez la classe Etudiant avec ses différents champs
2. Créez un générateur retournant un étudiant créé avec des données aléatoires.
3. Créez un programme séquentiel qui génère une liste de 150000 étudiants.
4. Mesurez le temps d'exécution de votre programme
5. Modifiez votre programme pour qu'il crée 3 Thread qui créent chacun 50000 étudiants. N'oubliez pas que vous voulez une liste de 150000 étudiants et pas 3 listes de 50000.
6. Mesurez le temps d'exécution. Combien avez-vous gagné? Si vous ne gagnez pas en performances (ou pas assez à votre goût), analysez votre gestion de la liste d'étudiants entre les Threads et voyez comment améliorer cela.

Bonus Sauvegardez les 150000 étudiant sur le disque le plus rapidement possible. Mesurez l'impact sur les performances.

Note : pour mesurer les performances d'un programme, il faut généralement le lancer à de multiples reprises puis faire une moyenne du temps d'exécution

3 Livraison

Ce TP doit être rendu terminé avant le **29 novembre 2021 à 14h00**.

Pour le rendre, utilisez le dépôt GitHub pour lequel vous avez reçu une invitation. Si vous n'avez pas d'invitations, merci de me l'indiquer en début de TP. Ce dépôt privé doit être utilisé tout au long du développement. Un bonus de 0.1 point est donné pour chaque commit dans le dépôt dans la limite de 2 points. Un Readme clair et mis en forme permet de gagner 1 point supplémentaire. Un TAG doit être réalisé pour indiquer la version à tester. Le nommer clairement, par exemple REL_1.0. Le dépôt ne doit plus être utilisé à partir de la date limite : seuls les commit réalisés avant l'heure seront pris en compte.