

TP 4 : Exercices d'optimisation

1 Objectifs

L'objectif de ce TP est d'étudier et d'optimiser quelques exemples simples de code Java.

2 Matériels et logiciels nécessaires

Ces travaux pratiques se déroulent sur système GNU/Linux. Si vous travaillez sur votre machine personnelle, il est de votre responsabilité d'y installer et configurer tous les logiciels nécessaires. Les machines de l'université fonctionnent sous la distribution ubuntu. Vous pouvez travailler sur d'autres distribution si vous le souhaitez. La version du JDK que nous utilisons est la version 8.

3 Évaluation

L'évaluation de chaque TP s'effectue via compte-rendu écrit, seul ou en binôme, des travaux effectués durant la séance de TP. Ce compte-rendu doit être fourni au format PDF pour chacun des TP. Vous pouvez utiliser tous les outils que vous souhaitez pour mettre en forme votre compte-rendu : Word, Latex, LibreOffice... Il est conseillé de commencer à préparer le compte-rendu au fur et à mesure que vous avancez sur le TP. Quelques points rentrant en compte pour la notation :

- Maîtrise des outils abordés en cours
- Analyses personnelles des travaux réalisés
- Présence de screenshots, de résultats des outils etc
- Qualité de la mise en page (Latex non obligatoire, mais une mise en page correcte n'est)
- Pas de plagiat
- Présence en cours/TP
- Respect des délais et des consignes pour rendre le compte-rendu

L'objectif du compte rendu est de montrer que vous avez compris ce que vous faisiez, et que vous êtes capables d'avoir un regard critique. Prendre du recul sur les différents outils en allant plus loin que la simple description de ceux-ci est très important.

Concernant les délais, le compte-rendu doit être rendu au plus tard une semaine après la dernière séance du TP sur la plateforme Celene dédiée à ce cours. Le mot de passe vous sera fourni lors des séances.

Exemple de date limite de rendu du compte-rendu : Ce sujet de TP s'étale sur 1 séance. Si la dernière séance est un mardi, vous avez jusqu'au mardi de la semaine suivante 23h59 pour envoyer votre rapport.

Bien entendu, je serai présent lors des TP, n'hésitez donc pas à me poser des questions si vous en avez besoin. Vous pouvez aussi me contacter par mail entre deux séances en cas de problème.

Il n'existe pas une seule et unique correction possible pour ce TP. De ce fait, aucune correction ne vous sera fournie. Cependant, si vous souhaitez avoir des informations/précisions sur votre compte rendu, n'hésitez pas à me contacter soit par mail soit directement pendant les séances suivantes pour que nous puissions en discuter.

TP 4 : Exercices d'optimisation

1.1 Généralités

Au cours de ce TP, vous allez devoir travailler sur un ensemble d'exercices écrits en Java. J'ai écrit ces exercices en me servant de problématiques que j'ai déjà rencontré dans mon travail. Votre mission, si vous l'acceptez (et même si vous ne l'acceptez pas), est d'étudier ces différents exercices un par un, de comprendre ce qu'ils font, de les optimiser et enfin de faire un bilan sur vos modifications pour chacun des exercices.

Afin de vous aider, vous avez le droit d'utiliser tous les outils qui vous paraissent utiles afin d'analyser l'exercice et de trouver ce qui ne fonctionne pas ou mal. Nous avons vu bon nombre d'outils dans les TP précédents, cela ne devrait pas poser problème. La mise en place de benchmarks est fortement recommandée.

Attention, les phases d'analyse et d'études comparatives avant/après optimisations ne sont pas facultatives, bien au contraire. C'est la partie la plus importante et intéressante de votre travail. Toute modification de code que vous effectuez devra être expliquée dans votre rapport final. Toute modification de code sans explication ni analyse ne sera pas prise en compte.

Chacun des exercices possède une petite documentation qui explique ce que doit faire le programme que vous avez à optimiser. Vous êtes libres d'effectuer toutes les modifications sur le programme que vous souhaitez dans la mesure où vous suivez toujours scrupuleusement ce qui est décrit dans la documentation, même si cela peut vous paraître idiot. Par exemple, si la documentation précise qu'il faut un certain nombre de threads, vous devez conserver ce nombre de threads travaillant en parallèle, même si le programme pourrait être optimisé en augmentant ou réduisant ce nombre. Vous avez cependant le droit de mentionner ces optimisations dans votre rapport final.

Afin de **limiter** la possibilité d'avoir un bug suite à vos modifications, des tests unitaires sont mis en place. Vous pouvez en rajouter si vous le souhaitez, du moment que ceux-ci sont toujours conformes à la description de chacun des exercices.

Bien évidemment, il vous est demandé de me fournir vos sources en même temps que le compte-rendu.

1.2 Description du projet

Le projet Java qui vous est fourni est sous la forme d'un projet maven standard qui intègre directement les tests unitaires. Vous pouvez builder le projet et exécuter les tests depuis n'importe quel IDE ou directement en ligne de commande.

Chaque exercice fait l'objet d'une classe à part entière et ils sont tous indépendants. Ils ne sont pas triés par ordre de difficultés.