



Langages de programmation 2 : Projet Java

Yves Roggeman *

INFO-F202 — Année académique 2016-2017

Résumé

Ceci est l'un des deux énoncés de l'épreuve de première session qui se déroule en janvier. Il sert de base à l'épreuve orale ; l'évaluation finale porte sur l'acquisition des concepts démontrée à cette occasion. Le but de cet exercice de programmation est donc de démontrer une connaissance approfondie et un usage adéquat des constructions du langage de programmation Java, en particulier de l'usage des constructions standard de parallélisation *via* l'usage de « *threads* » et de classes génériques.

1 Problème

Le problème posé consiste à définir une structure de données générique de vecteur — que nous appellerons ici « `class Vect0` » — constituée d'un simple « *array* » de valeurs d'un type de base qui est un paramètre du type. Outre ses méthodes standard (construction, etc.), ce type implante deux méthodes : l'une réalisant la somme vectorielle (et plaçant le résultat dans le premier argument, comme un « += »), l'autre calculant le produit scalaire. Il va de soi que ces opérations ne sont possibles que sur des vecteurs de même type de base et de même nombre d'éléments.

La particularité de cette classe est que ces deux opérations sont effectuées en parallèle, du moins partiellement. En effet, selon un paramètre global de la classe indiquant le nombre de threads que l'on peut lancer en parallèle, ces méthodes répartissent leurs opérations. Chaque thread lancé par elle se charge de l'opération sur une partie des composantes du vecteur : un bloc d'éléments successifs. Tous ces blocs sont de même taille, sauf éventuellement le dernier si le nombre de threads ne divise pas exactement la taille du vecteur.

Il est demandé de réaliser ce parallélisme exclusivement à l'aide des constructions de base du langage présentes dans « `java.lang` ».

2 Réalisation

Il vous est demandé d'écrire en Java la définition de cette classe et de toutes les méthodes nécessaires, ainsi qu'un programme principal servant de test de plusieurs instanciations de tels objets, pour différentes valeurs des paramètres et types de base.

L'évaluation portera essentiellement sur la pertinence des choix effectués dans l'écriture : la codification, la présentation et l'optimisation du programme justifiées par la maîtrise des mécanismes mis en œuvre lors de la compilation et l'exécution du code. D'une manière générale, le respect strict des directives, la concision, la précision, la lisibilité (clarté du texte source), l'efficacité (pas d'opérations inutiles ou inadéquates) et le juste choix des syntaxes typiques de Java seront des critères essentiels d'appréciation. De brefs commentaires dans le code source sont souhaités pour éclairer les choix de codification.

Votre travail doit être réalisé pour le vendredi 22 décembre 2017 à 10 heures au plus tard. Vous remettrez tout votre travail empaqueté en un seul fichier compacté (« `.zip` » ou autre) *via* le site du cours

*Université libre de Bruxelles (ULB) <yves.roggeman@ulb.ac.be>



(INFO-F202) sur l'Université Virtuelle (<https://uv.ulb.ac.be/>). Ceux-ci devront contenir en commentaire vos matricule, nom et prénom. Le jour de l'examen, vous viendrez avec une version imprimée — un *listing* — de ces divers fichiers. Une impression du résultat d'une exécution du programme est également demandée.