

Intégration des nombres complexes et des Unums en Java avec COJAC

PV de la séance du 21 juin 2020 (17h00 - 17h35) via Teams

Présents: Cédric Tâche (Etudiant), Baptiste Wicht (Expert)

Ordre du jour

Les points suivants ont été abordés durant la séance:

1. Cahier des charges v1.2
2. Gestion de projet
3. Rapport v0.4
4. Démonstration

COJAC

1. COJAC permet de détecter des comportements arithmétiques limites (overflow, ...)
2. COJAC permet aussi d'ajouter de nouvelles capacités aux programmes cibles (calcul par intervalle, précision arbitraire, ...)
3. M. Wicht a déjà travaillé sur une des premières versions de COJAC et a aussi accompagné un projet plus récent.

Objectifs

1. L'objectif principal est l'intégration des nombres complexes et des unums avec COJAC. Il faudra aussi créer des démonstrations pour montrer l'utilité de ces fonctionnalités.
2. Comme COJAC est un projet ancien et que de nombreuses personnes y ont travaillées, il y a beaucoup d'aspects qui peuvent aussi être améliorés: documentation, architecture, logs, CI, version des librairies, ...

Gestion de projet

1. Il y a une réunion par semaine entre M. Bapst et M. Tâche.
2. Les communications sont essentiellement réalisés par Teams et parfois par email.
3. Il y a un dépôt GitLab qui contient tout le code et la documentation (PV compris).
4. Seul M. Tâche travaille sur ce dépôt GitLab.
5. Une branche *dev* existe et est majoritairement utilisée. Elle sera ajoutée au *master* lorsque la partie des nombres complexes sera terminée.
6. Un CI qui crée le JAR de Maven (et qui exécute les tests) a été ajouté sur le GitLab.

Démonstration

1. La démonstration permet de montrer l'utilité des nombres complexes conformément aux spécifications.
2. Une deuxième démonstration est en cours de réalisation pour montrer les méthodes magiques.

Etat du projet

1. L'implémentation est presque terminée.
2. Des tests unitaires ont déjà été réalisés.
3. La démonstration fonctionne comme prévu.

Problèmes

1. La méthode *Double.isNaN* appelle la méthode du wrapper, mais provoque un cast en double également. Ce qui bloque le fonctionnement d'un mode de comparaison stricte.

Améliorations possibles

1. Les avantages et inconvénients des wrappers et behaviours peuvent être détaillés conformément à un des objectifs secondaires.
2. Dire explicitement que la librairie analysée a été utilisée pour gérer les nombres complexes.
3. Les références peuvent être citées que la première fois.
4. Lorsqu'on référence les figures, on l'écrit normalement avec une majuscule.
5. Le cahier des charges n'a pas besoin d'être en annexe. Il faut le voir avec M. Bapst.
6. Les tests de performance pourraient aussi être exécutés automatiquement par le CI (objectif secondaire).
7. Pour les unums, au lieu d'utiliser JNI, il pourrait être avantageux d'utiliser JNA.

Planification

1. Le projet est en accord avec la planification.
2. Le problème au début pour la compilation de COJAC a retardé le projet d'environ 1 à 2 jours.
3. Deux objectifs secondaires ont déjà été planifiés. La mise à jour des librairies est très probable. Les tests de performance sont un bon objectif secondaire, mais un autre objectif pourrait être réalisé à la place.

Prochaine visite: Le lundi 12 juillet 2021 à 17h00