



DOSSIER INTERPRÉTEUR DU LANGAGE LIR

PROJET PROPOSÉ PAR FRÉDÉRIQUE BARRIOS

Nicolas CAMINADE, Sylvan COURTIOL,
Pierre DEBAS, Heïa DEXTER,
Lucàs VABRE

Sommaire

I	Plan projet	2
II	Spécifications détaillées	3
III	Conception	4
IV	Codage	5
V	Tests	6
1	Tests du paquetage interpreteurlir.donnees.litteraux	8
1.1	Litteral	8
1.2	Chaine	8
1.3	Entier	8
1.4	Booleen	9
VI	Conclusion	10
VII	CD	11
VIII	Manuel utilisateur	12

Première partie

Plan projet

Deuxième partie

Spécifications détaillées

Troisième partie

Conception

Quatrième partie

Codage

Cinquième partie

Tests

Démarche globale

Afin de développer l'Interpréteur LIR selon un modèle de cycle de vie itératif, nous avons privilégié la méthode du TDD, Test Driven Development ou développement dirigé par les tests en français.

Ainsi, la majorité des composants de ce logiciel ont été développés selon cette méthode à l'aide des outils de tests écrits lors des TD de Programmation Orientée Objet du semestre 2. Par conséquent, nous n'avons pas utilisé le framework de test JUnit.

Chapitre 1

Tests du paquetage `interpreteurlir.donnees.litteraux`

1.1 Litteral

Lors des itérations 1, la classe `Litteral` a été développée comme une classe non-abstract, en effet nous n'avions pas encore abordé cette notion en cours. Ainsi cette classe a été développée en TDD et nous avons par conséquent effectué les tests unitaires de cette classe.

Cependant à la fin de l'itération 3, dans une optique d'amélioration des codes sources, nous avons passé cette classe en abstract, ainsi les tests unitaires menés n'avaient plus lieu d'être et ont donc été tout de même conservés en commentaire.

1.2 Chaîne

Les jeux de tests de la classe `Chaîne` prennent en compte les cas de chaînes vide, la taille maximale des chaînes, leur syntaxe (avec le contenu de la chaîne entre "). Aussi l'opération de concaténation a été testée. Tous les tests menés ont été concluants.

1.3 Entier

La classe `Entier` est très proche de la classe `Integer` existant déjà dans le JDK, ainsi son développement a été rapide. À l'instar des tests menés pour la classe `Chaîne`, tous les tests de la classe `Entier` notamment des opérations arithmétiques ont été concluants.

Notons, que pour les opérations arithmétiques telles que la division et le reste de la division, le cas particulier de la division par zéro a été testé à part.

1.4 Booleen

La classe Booleen n'a posé aucun problème particulier.

Chapitre 2

Tests du paquetage interpreteurlir.donnees

2.1 Identificateur

La classe Identificateur a été développée en TDD lors de l'itération 1 cependant elle a été passée en abstract lors de l'itération 3, comme pour la classe Litteral, les tests unitaires menés lors de l'itération 1 n'avaient plus lieu d'être et ont été conservés en commentaire. La méthode d'instance compareTo() testée avant le passage de la classe en abstrait et vaut pour les identificateurs d'entier et de chaîne.

2.2 IdentificateurChaine et IdentificateurEntier

Lors des tests unitaires des deux classes, la syntaxe des identificateurs a été testées. Les tests ont été concluants.

2.3 Variable

La classe Variable a été développée lors de l'itération 1 et a donc été testée avec les identificateurs d'entier et de chaîne et seulement avec des valeurs de type Chaine, en effet, la classe Entier ne faisait pas partie de la conception de l'itération 1.

Sixième partie

Conclusion

Septième partie

CD

Huitième partie

Manuel utilisateur