

# Implémentation d'un compilateur simplifié

Federico Pfeiffer

10 juin 2017

## Résumé

Ce rapport décrit l'implémentation d'un compilateur du langage Hepial, défini les spécifications en annexe.

## Table des matières

<b>1</b>	<b>Présentation</b>	<b>1</b>
1.1	État général de l'implémentation . . . . .	1
1.2	Utilisation . . . . .	1
1.3	Vue d'ensemble . . . . .	1

# 1 Présentation

## 1.1 État général de l'implémentation

Le compilateur a pu être implémenté selon le cahier des charges demandé : il analyse syntaxiquement un code source en créant un arbre abstrait et une table des symboles, puis recherche les erreurs sémantiques possibles au sein de l'arbre abstrait et la table des symboles. Si aucune erreur a été détectée, le compilateur produit un code source en jasmin, qui est ensuite compilé en byte code `.class`.

La notion de récursivité de fonctions, et de portée des variables a également pu être implémentée. De même, des expressions complexes telles que celles-ci ont également pu être implémentées. *if*( $a < b \parallel 2*6+4 > 0$ ). Enfin, des imbrications de boucles (telles que *while* et *for*) sont également possible.

Bien que les tableaux aient été ajoutés, ceux-ci n'ont été que très peu testés. Plusieurs dimensions sont en théories possibles, mais cela n'a pas été testé non plus. La gestion des tableaux a été définie de la sorte : `array[5 .. 10]` crée un tableau de 5 cases, dont l'index va de 0 à 5.

Enfin, les expressions *non* et *innégal* s'écrivent respectivement *!* et *!=* au lieu de *non* et *<>*. Par ailleurs, la déclaration de fonction au sein d'une fonction est interdite.

## 1.2 Utilisation

L'utilisation du compilateur se fait de la manière suivante :

1. Compilez le compilateur avec la commande *make*. Cela va générer un dossier */bin/* contenant le byte code du compilateur. L'exécutable nommé *hepiaCompile* sera généré dans le même dossier où se trouve le fichier *make*. (Pas besoin de changer de dossier avec *cd*).
2. Compilez ensuite le fichier *hepial* avec la commande *bash hepiaCompile* suivi du nom du fichier *Hepial* à compiler. Cela va générer un dossier *compiledBin* contenant le byteCode du programme *Hepial*, ainsi que ses sources en jasmin. L'exécutable du programme sera généré dans le même dossier où se trouve le fichier *make*.

### 1 – Utilisation du compilateur

```
# compiler le compilateur
$ make

# compiler le fichier hepial
$ bash hepiaCompile <fileName> # fileName = input.txt par défaut

# lancer l exécutable compilé
$ bash <programName>
```

### 1.3 Vue d'ensemble

La compilation s'exécute dans cet ordre :

1. Génération de l'arbre abstrait, de la table des symboles pendant le parsing syntaxique de Cup.
2. Vérification sémantique dans l'arbre abstrait et dans la table des symboles.
3. Production du code : fichiers jasmin et fichier .class.

Les diverses étapes de la compilation sont exécutées depuis le fichier *HepialCompilateur.java*, à l'intérieur de la fonction *main*.