

## 1 But

De nos jours la quasi-totalité des langages de programmation actuels, sont en anglais. L'idée de notre projet est de pouvoir écrire du code entièrement français, sans opérateurs mathématiques. Grâce à ceci les enfants francophones pourront dès leur plus jeune âge s'élancer dans la joie de la programmation. La sortie finale de notre compilation sera un fichier « .java » qui devra être compilé par l'utilisateur.

## 2 Spécification

Dans Frava les accents sont interdits et ne sont pas gérés. Quant aux noms des variables, ils sont limités aux lettres (a-z) et sont sensibles à la casse. La notation en CamelCase est fortement conseillée.

### 2.1 Dépendances

Langage source : **Frava**

Langage intermédiaire : **Python 3.3.2** avec

- pydot 1.0.3
- PLY 3.4 (avec les modules lex et yacc)
- Graphviz (utilitaire)

Langage de destination : **Java**

### 2.2 Types

Frava	Java
<b>b entier;</b>	int b = 0;
<b>b reel;</b>	double b = 0.0;
<b>b texte;</b>	String b = "";
<b>b booleen;</b>	boolean b = false;

### 2.3 Opérations

Frava	Java
<b>b entier egal 2;</b>	int b =2 ;
<b>b egal 3;</b>	b = 3;
<b>b egal 3,14;</b>	b = 3.14;
<b>b egal "bonjour tout le monde";</b>	b = "bonjour tout le monde";
<b>b egal vrai; //ou faux</b>	b = true; // or false
<b>plus</b>	+
<b>moins (binaire ou unaire)</b>	-
<b>fois</b>	*
<b>divise par</b>	/
<b>plus petit que</b>	<
<b>plus petit ou egal que</b>	<=
<b>plus grand que</b>	>
<b>plus grand ou egal que</b>	>=
<b>est egal a</b>	==

## 2.4 Opérations acceptées selon le type

Type	Opérations acceptées	
entier	entier	plus petit que
reel (attention : comparaison)	plus	plus petit ou égal que
	moins	plus grand que
	fois	plus grand ou égal que
	divise par	est égal a
booléen	égal	est égal a
texte	égal	est égal a
	plus	

## 2.5 Fonctions de bases

Frava	Java
afficher variable ;	System.out.println(variable);

## 2.6 Conditions et boucles

Frava	Java
SI( cond ) { expr };	if( cond ) { expr }
SI( cond ) { expr }	if( cond ) { expr }
SINON { expr };	else { expr }
TANT QUE( cond ) { expr };	while( cond ) { expr }
POUR( I de A a B par pas de C ) { expr } ;	for(int I = A ; I <= B ; I = I + C) { expr }

## 3 Mots réservés

Voici la liste des mots réservés indépendamment de la casse.

si	entier	plus
sinon	reel	fois
pour	texte	vrai
afficher	booléen	faux
de	égal	
a	moins	

## 4 Prise en main

Notre compilateur est composé de quatre modules distincts :

- Analyseur lexical (lex\_1.py)
- Analyseur syntaxique (parser\_2.py)
- Analyseur sémantique (semantic\_3.py)
- Génération du code en java (generator\_4.py)

Le fichier « compiler.py » effectue chacune des quatre étapes ci-dessus et vérifie les erreurs après chaque étape et s'arrête si nécessaire.

Pour compiler du code Frava les dossiers suivants sont nécessaires :

Dossier	Utilité
<b>to_compile</b>	Contient les fichiers à compiler
<b>pdf</b>	Emplacement des pdf représentant les arbres syntaxiques
<b>outputs</b>	Contient les fichiers de sorties en java ainsi que les fichiers logs
<b>generated</b>	Nécessaires pour l'analyse syntaxique

## 5 Planification effective

		Idée de projet	Analyse lexicale	Analyse Syntaxique	Analyse Sémantique	Partie Arrière	Finalisation et tests	Rapport final
1	13 novembre 2013							
2	20 novembre 2013							
3	27 novembre 2013							
4	4 décembre 2013							
5	11 décembre 2013							
6	18 décembre 2013							
7	8 janvier 2014							