Grenoble INP - Ensimag

Projet Génie Logiciel

MANUEL UTILISATEUR

Guide d'utilisation et Diagnostic

ÉQUIPE GL56

Ayoub – Hamza – Mouad – Chaima – Abdellah

Janvier 2026 – Version 1.0

Table des matières

1 INTRODUCTION

Ce manuel décrit l'utilisation du compilateur **Decac** développé par l'équipe GL56. Il permet de transformer des programmes écrits en langage Deca vers l'assembleur de la machine abstraite IMA.

Ce document détaille les options de compilation, la signification des messages d'erreur retournés (compilation et exécution) ainsi que les spécificités de l'implémentation (Extensions et Limitations).

2 UTILISATION STANDARD

Le compilateur **Decac** s'utilise via l'interface en ligne de commande. Il analyse des fichiers sources Deca (`.deca`) et produit le code assembleur IMA (`.ass`) dans le même répertoire que le fichier source.

2.1 Syntaxe de la commande

La compilation peut être lancée via le script de test rapide `deca_test` (si configuré à la racine) ou directement via l'exécutable principal du projet.

Syntaxe Générale

Via le script de test :

```
$ ./deca_test [options] <fichier.deca>
```

Via le chemin complet (Standard) :

```
$ ./src/main/bin/decac [options] <fichier.deca>
```

Exemples d'exécution concrète : Voici quelques commandes types pour tester différentes fonctionnalités :

— **Compilation avec contrainte de registres (3 registres max) :**

```
./deca_test -r 3 test_class.deca
```

Génère `test_class.ass` en utilisant uniquement R2, R3 et R4.

— **Vérification syntaxique seule (Parseur) :**

```
./deca_test -p test_class.deca
```

Affiche le code décompilé sans générer de fichier assembleur.

— **Appel complet depuis la racine du projet :**

```
./src/main/bin/decac -p src/test/deca/syntax/valid/created/test_class.deca
```

2.2 Options Disponibles

Le comportement du compilateur peut être altéré grâce aux drapeaux suivants :

Option	Description et Effet
`-p`	**Parseur seul.** Arrête le processus après l'analyse syntaxique. Affiche le code source décompilé sur la sortie standard. Utile pour vérifier comment le code est "compris" par le compilateur.
`-v`	**Vérification seule.** Arrête le processus après l'analyse contextuelle (Passe 2). Ne génère aucun fichier `.ass`. Permet de vérifier la validité des types rapidement.
`-n`	**No Check.** Désactive la génération des tests de sécurité à l'exécution (débordement, division par zéro, null check). Le code généré est plus léger mais potentiellement instable.
`-r X`	**Registres.** Limite le nombre de registres banalisés à X (avec $4 \leq X \leq 16$). Utile pour simuler une architecture contrainte et forcer le mécanisme de *spilling*.
`-d`	**Debug.** Active les traces de débogage internes (états du lexer, table des symboles).

3 DIAGNOSTIC ET MESSAGES D'ERREUR

Le compilateur a été conçu pour guider l'utilisateur. Voici la liste des erreurs possibles classées par étape.

3.1 Erreurs Système (Fatal)

Ces erreurs bloquent le démarrage immédiat du compilateur.

— `Erreur fatale : <fichier> (No such file...)` : Le fichier source n'existe pas.

— `Les options -p et -v sont incompatibles` : Usage incorrect de la ligne de commande.

3.2 Erreurs Syntaxiques (Partie A)

Le code ne respecte pas la grammaire du langage Deca.

— `mismatched input ';' expecting ...` : Souvent une parenthèse ou un point-virgule oublié.

— `token recognition error` : Utilisation d'un caractère interdit (ex : accents dans un identifiant).

— `no viable alternative` : Structure de phrase inconnue.

3.3 Erreurs de Syntaxe Contextuelle (Partie B)

Cette phase vérifie que le programme respecte les règles de typage et de portée du langage Deca. Les erreurs sont détectées après la construction de l'arbre (AST). Elles sont toujours accompagnées de la position précise (Ligne :Colonne) dans le fichier source.

3.3.1 1. Déclarations et Hiérarchie (Passes 1 & 2)

Ces erreurs concernent la structure des classes, l'héritage et la signature des méthodes.

Message d'erreur	**Configuration provocatrice**
`Class <Name> is already defined`	Déclaration de deux classes avec le même nom dans le même fichier.
`Superclass <Name> is undefined` *ou* `Circular inheritance detected`	Une classe tente d'hériter d'une classe inexistante ou crée un cycle d'héritage (ABA).
`Method <Name> signature does not match superclass`	Redéfinition invalide d'une méthode dans une sous-classe (changement du nombre ou du type des paramètres). Le langage Deca n'autorise pas la surcharge.
`Return type mismatch in override`	Une méthode redéfinie tente de renvoyer un type qui n'est pas un sous-type du type original (non-respect de la covariance).

3.3.2 2. Typage et Expressions (Passe 3)

Ces erreurs surviennent lors de la vérification du corps des méthodes et du bloc principal.

Message d'erreur	**Configuration provocatrice**
`Target of selection must be an object`	Tentative d'accès à un champ (ex : `a.x`) sur une variable de type primitif (`int`, `float`, `boolean`).
`Incompatible types for assignment`	Affectation d'un type incompatible (ex : mettre un `float` dans un `int`). *Note : la promotion inverse int → float est autorisée.*
`Invalid cast between types`	Tentative de *cast* entre deux classes n'appartenant pas à la même lignée d'héritage (ni parent, ni enfant).
`Condition must be a boolean`	Utilisation d'un entier ou d'un objet comme condition dans un `if` ou un `while`.
`Binary operation not supported`	Opération invalide pour les types fournis (ex : additionner un `boolean` avec un `int`, ou `&&` sur des flottants).

3.3.3 3. Flux de Contrôle et Instructions (Passe 3)

Vérifications liées à la structure logique des blocs de code et aux identificateurs.

Message d'erreur	**Configuration provocatrice**
`Undefined variable <Name>`	Utilisation d'un identificateur (variable ou paramètre) qui n'a été déclaré ni localement, ni comme champ de la classe.
`printx does not support type string`	Tentative d'affichage hexadécimal (`printx`) sur une chaîne de caractères. Seuls `int` et `float` sont supportés.

3.4 Erreurs d'Exécution (Partie C / Runtime)

Ces erreurs surviennent lors de l'exécution du programme `.ass` (sauf si l'option `-n` est activée).

Message Affiché	**Cause et Diagnostic**
`Erreur : Debordement de la pile`	**Stack Overflow.** La pile mémoire est pleine. Souvent causé par une récursion infinie ou de très grosses variables locales.
`Erreur : tas plein`	**Heap Overflow.** Le tas est saturé. Trop d'objets créés avec `new`. Augmentez la mémoire du simulateur IMA.
`Erreur : dereferencement de null`	**Null Pointer Exception.** Tentative d'accès à `x.champ` alors que `x` vaut `null`.
`Erreur : Division par zero`	Opération mathématique interdite (entière ou flottante).
`Erreur : Saisie invalide`	L'utilisateur a entré du texte lors d'un appel à `readInt()`.
`Erreur : sortie sans return`	Une méthode typée (non-void) a terminé son exécution sans renvoyer de valeur.

4 EXTENSIONS ET LIMITATIONS

Le compilateur `decac` intègre des fonctionnalités avancées dépassant la spécification standard, notamment une bibliothèque mathématique et le support de l'assembleur inline.

4.1 Bibliothèque Mathématique (Math.decah)

Le compilateur propose une implémentation native des fonctions trigonométriques (`sin`, `cos`, `atan`, `asin`, `acos`) basée sur l'algorithme CORDIC.

Documentation Détaillée

Note importante : Afin de préserver la concision de ce manuel utilisateur, les détails techniques approfondis concernant cette extension sont disponibles dans le document séparé intitulé ***Extension***.
Vous y trouverez notamment :

- L'analyse théorique de l'algorithme CORDIC.
- Les mesures précises de performance (marges d'erreur, ULP, temps de cycle).
- Les preuves de convergence et les domaines de définition exacts.

4.2 Limitation Connue : Gestion de la Pile

Comme évoqué précédemment, la gestion de la mémoire (instruction `TSTO`) est réalisée de manière statique et conservatrice. Cela garantit la sécurité de l'exécution mais peut limiter la profondeur de récursion acceptée par rapport à une gestion dynamique.

Pour une analyse exhaustive des limites architecturales imposées par la machine IMA (entiers 32 bits, flottants), veuillez également vous référer au ***Extension***.