

Projet Génie Logiciel

Compilateur Decac

Manuel de conception

Équipe GL2

Étudiants : Élina Houdé, Julien Pinto Da Fonseca,

Léa Solo Kwan, Yin Xu

Sommaire

Introduction	3
1. Architecture	4
1.1. Gestion des erreurs	4
1.2. Codegen	4
2. Spécifications sur le code	5
3. Algorithmes et structures de données	6

Introduction

Ce manuel de conception s'adresse à tout développeur qui souhaiterait maintenir et/ou faire évoluer le compilateur. Celui-ci est essentiellement composé de la description de l'organisation générale de l'implémentation.

1. Architecture

1.1. Gestion des erreurs

Afin de simplifier la gestion des messages d'erreurs et la maintenabilité du code, il a été décidé de créer une classe dont le rôle est la centralisation des différents messages d'erreur pouvant être retournés à l'utilisateur. De ce fait, lorsqu'une erreur doit être déclenchée au sein du code, le message d'erreur approprié provenant de la classe statique **ErrorMessages** est passé en paramètre de l'erreur.

Chemin depuis la racine du projet :

src/main/java/fr/ensimag/deca/ErrorMessage.java

1.2. Codegen

Afin de mener à bien l'implémentation des différentes méthodes entrant dans le cadre de l'étape de génération de code, plusieurs classes personnalisées ont dû être créées. On retrouvera notamment les classes suivantes :

- **RegisterManager** : permet de gérer tout ce qui touche aux registres ;
- **StackManager** : permet de gérer tout ce qui touche à la pile ;
- **LabelManager** : permet de gérer la génération des labels ;
- **LabelType** : énumération composée des différents types de label.

Chemin depuis la racine du projet :

src/main/java/fr/ensimag/deca/codegen/

2. Spécifications sur le code

-- Disponible plus tard --

3. Algorithmes et structures de données

-- Disponible plus tard --