

1. Architecture

Gestionnaires de génération de code

ErrorCatcher

Cette classe génère les gestionnaires d'erreur lors de la phase codegen. Cela comprend pour le moment :

- io_error (erreur d'entrée/sortie)
- so_error (débordement de pile)
- ov_error (débordement d'un résultat lors d'une opération)

Elle permet de créer les Label d'erreur (cf ci-dessus) et de générer le code assembleur gérant l'erreur (affichage d'un message d'erreur et arrêt du programme).

LabelManager

Il s'agit d'un dictionnaire associant le nom d'une étiquette (string) à un objet Label. Elle gère la création et la récupération de label pour des sauts, conditions et boucles.

StackManager

Cette classe gère l'initialisation de la pile. Elle permet :

- de compter le nombre de variables utilisées lors de la génération du code des déclarations et des évaluations d'expression
- d'allouer la taille nécessaire en début de code

RegisterManager

Cette classe stocke le nombre maximum de registre utilisable pour la génération de code. Par défaut ce nombre est 16. Il est modifiable via l'option -r de decac.

Ces gestionnaires sont en attributs dans la classe DecacCompiler et se situent dans le répertoire src/main/java/fr/ensimag/deca/codegen/.

2. Spécification sur le code du compilateur

Pas de spécification autres que celles du cahier des charges.

3. Algorithmes et structures de données

Pas d'algorithmes ou de structures de données autres que ceux du cahier des charges.