

Manuel utilisateur

Equipe 46 : -Ait Taleb Aymane
-Boussemid Youssef
-Hassou Karim
- Bendou Safouane
-Jaoudar Reda

January 2022

1 Comment utiliser le compilateur :

1.1 Le potentiel du compilateur :

Le compilateur que nous avons conçu est complet, qui permet par conséquent à l'utilisateur de faire des calculs, des affichages et des lectures. En outre, notre compilateur est capable de gérer les classes et les objets.

1.2 Syntaxe d'utilisation et options du compilateur :

On désigne le fichier d'entrée par les chemins de la forme **répertoires / nom.deca**, et le résultat est également situé dans le même répertoire que le fichier source, dans un fichier **répertoires/nom.ass**

Notre commande de compilation est **decac**, cette commande possède les options suivantes :

- b : affiche une bannière indiquant le nom de l'équipe .
- p : arrête decac après l'étape de construction de l'arbre, et affiche la décompilation de ce dernier (i.e. s'il n'y a qu'un fichier source à compiler, la sortie doit être un programme deca syntaxiquement correct).
- v : arrête decac après l'étape de vérifications (ne produit aucune sortie en l'absence d'erreur) .
- n : supprime des tests spécifiés à l'exécution.
- r X : limite les registres banalisés disponibles à R0 ... RX-1, avec $4 \leq X \leq 16$
- d : active les traces de debug. Répéter l'option plusieurs fois pour avoir plus de traces.

2 Les limitations du compilateur :

Le compilateur ne supporte pas les opérations de transtypage (cast). L'option decac qui gère l'exécution en parallèle de plusieurs fichiers deca n'est pas gérée par notre compilateur.

3 Les messages d'erreur et les configurations qui les provoquent :

Le tableau suivant résume les messages d'erreur qui peuvent être retournés à l'utilisateur, ainsi que les configurations qui provoquent ces messages. Le message d'erreur qui pourra s'afficher lors de la manipulation de l'utilisateur est sous la forme suivante : NomFichier:ligne d'erreur:colonne d'erreur: message d'erreur

Nom fichier:ligne d'erreur:colonne d'erreur:les messages d'erreur	les configurations provoquantes
Assign incorrect: types différents	Affecter une valeur d'un certain type à une variable d'un type différent
This condition is not a bool	l'expression de condition ne retourne pas une valeur de type bool
Void not supported for arithmetic operation	Faire des opérations arithmétiques avec des variables de type void
Boolean not supported for arithmetic operation	Faire des opérations arithmétiques avec des variables de type bool
Class not supported for arithmetic operation	Faire des opérations arithmétiques avec des classes
String not supported for arithmetic operation	Faire des opérations arithmétiques avec des variables de type string
Void not supported for boolean operation	Faire des opérations booléennes avec des variables de type void
Float not supported for boolean operation	Faire des opérations booléennes avec des variables de type float
Class not supported for boolean operation	Faire des opérations booléennes avec des classes
String not supported for boolean operation	Faire des opérations booléennes avec des variables de type String
Int not supported for boolean operation	Faire des opérations booléennes avec des variables de type Int
Variable 'Nom Variable' already declared	Initialiser deux fois la meme variable
'Nom Variable' used but not declared	oubli d'initialisation d'une variable
Inexisting type	attribuer à une variable un type autre que : int, float, bool, void, string
Void cannot be declared as a variable	attribuer le type void à une variable
String cannot be declared as a variable	déclarer une variable de type String
Type not found	déclarer une variable avec un type inexistant
Incorrect initialisation, incompatible types(' Nom Type1 ' and ' Nom Type2 ')	initialiser une variable avec une valeur de type incompatible avec son propre type
Only integers are allowed for modulo operation	utiliser l'opération modulo pour des variables qui ne sont pas de type int
Only booleans are supported for the Not expression	Utiliser l'expression NOT pour des variables qui ne sont pas de type bool
' operandtype ' not supported for UnaryMinus	utiliser le signe Minus pour des types autre que float ou int
Arret de l'exécution: division par zéro impossible	diviser une variable par 0
Cannot be initiated with a nullPointer	Assigner une valeur nulle à un type autre qu'une class
Void not supported for comparison operation	comparer un type void

String not supported for inequality operations	comparer un type string
Booleans not supported for inequality operations	comparer un type boolean
Class "nom.class" already declared	declarer une deuxième classe de même nom qu'une déjà déclarée
Superclass "nom.super.class" undefined.	hériter d'une classe non déclarée avant l'opération d'héritage
"nom.super.class" is not a class.	hériter d'un type différent de class
Void cannot be declared as a Class field.	déclarer un type void dans le champ d'une classe
String cannot be declared as a Class field.	déclarer un type string dans le champ d'une classe
method name already declared	declarer deux méthodes de même nom
parametre "nom.paramètre" already mentionned.	écrire deux paramètres de même nom en une déclaration de méthode
Methods can only be used on objects	appel de méthode
"Nom Méthode" is not a method	appel d'une méthode inexsistente
Problem in the number of parameters of the function	appel d'une fonction sans respecter le nombre de ses paramètres
Method "Nom Méthode" expecting "type parametre expected" received "type paramètre "	appel de méthode sans respecter les types de ses paramètres
Type de retour incompatible	Méthode retournant un type autre que classe en expectant une classe au retour
Le type de retour n'est pas une sous classe de celui attendu	La méthode retourne une classe ou un type qui n'est pas une sous classe de la classe expectée au retour
Type de retour different de celui attendu	retourner un type incompatible avec celui expecté
Return can not be used in the main	utiliser instruction return en main
Fields can only be used on objects	appel d'un attribut en utilisant un type autre que classe
"Nom Ident" isn't a field of the class in question	appel d'un attribut inexistant dans la classe
"Nom Ident" is a protected field	appel d'un attribut protected hors d'une sous classe de la classe contenant l'attribut
This cannot be used in the main section of the program	utiliser this en main

4 L'Extension TRIGO de la bibliothèque standard :

Il s'agit de l'extension TRIGO, permettant d'enregistrer comme ressources pour le langage DECA la bibliothèque Math, ayant en disposition toutes les fonctions trigonométriques qui concernent les flottants, ainsi que leurs fonctions inverses. La fonction ULP prend place au sein de cette bibliothèque. En effet, celle-ci permet de calculer la distance entre le flottant passé en paramètre de fonction et le flottant qui lui est le plus proche. Elle permet ainsi de calculer le degré de précision, pour gérer cette dernière vis-à-vis aux fonctions trigonométriques.

4.1 Mode opératoire de l'utilisation de l'Extension TRIGO :

Il s'agit d'évoquer les méthodes d'utilisation de la bibliothèque Math.decah. Celle-ci figure dans le fichier Math.deca, évoqué par le biais de l'instruction *include"src/main/resources/include/Math.decah"*.

Ensuite, il s'agit de montrer la méthode d'appel des fonctions trigonométriques issues de cette bibliothèque. D'abord, il sied de faire appel à la classe Math en l'initialisant : *Math m = new Math()*. Ceci crée une instance de la classe Math qui permet de faire appel à toutes les fonctions. Un exemple de test est fourni dans le répertoire de la bibliothèque, faisant un calcul simple, en utilisant l'instruction *m.print(cos(pi))* par exemple.

D'ailleurs, dans le repertoire figure le fichier .java traitant toutes les fonctions de la bibliothèques maths de façon détaillée, faisant usage des de méthodes efficaces.

4.2 Les limitations de l'Extension TRIGO :

Quant aux limitations de notre extensions, l'algorithme Crdic fait usage des tableaux. Or, ces derniers ne sont guère présents au sein du compilateur Decah. Dès lors, l'initialisation de plusieurs variables était nécessaire. L'extension nous permet de calculer les fonctions trigo ; la seule limitation étant l'impossibilité de passer en paramètre un entier dans nos fonctions. En effet, le type dynamique doit nécessairement être un flottant.