

Manuel utilisateur

UTILISATION

Usage : `decac [[-p | -v] [-n] [-r X] [-d]* [-P] [-w] <fichier Deca>...] | [-b]`

La compilation se fait en utilisant la commande *decac* suivie d'éventuelles options et d'un ou de plusieurs noms de fichiers *.deca*. La liste des options disponibles peut être affichée en utilisant la commande *decac* seule.

Options :

-b	(banner)	affiche une bannière avec les noms des membres de l'équipe
-p	(parse)	arrête le programme après la construction de l'arbre, et en montre la décompilation
-v	(verification)	arrête decac après l'étape de vérification (sans sortie s'il n'y a pas d'erreur)
-n	(no check)	ne vérifie pas le points 11.1 et 11.3 de la sémantique de Deca (erreurs à l'exécution)
-r X	(registers)	limite les registres disponibles à R0, ..., R{X-1}, avec $4 \leq X \leq 16$
-d	(debug)	active les traces de debug (-d pour info ; -d -d pour debug ; -d -d -d pour traces)
-P	(parallel)	s'il y a plusieurs fichiers sources, lance la compilation des fichiers en parallèle
-byte	(bytecode)	génère du bytecode Java au lieu de code assembleur ima

N.B.

Aucun affichage n'est produit si lorsque la compilation réussit, sauf lors de l'utilisation des options -b ou -p.

Les options -p et -v sont incompatibles.

L'option -b s'utilise sans autre option et sans fichier source.

MESSAGES D'ERREUR

Les messages d'erreur ont le format suivant:

<nom de fichier.deca>:<ligne>:<colonne>: <description informelle du problème>

Voici la liste des messages d'erreurs pouvant s'afficher et leur signification:

Erreurs à l'exécution:

Message d'erreur	Signification
ERROR: Input error	Erreur lors de la lecture de l'entrée utilisateur
ERROR: Stack overflow	Débordement de pile
ERROR: Dereferencing null	Tentative de déréférencer null
ERROR: Dividing (or modulo) by zero	Tentative de diviser (ou prendre le modulo) par zéro.
ERROR: Float overflow	Les calculs avec des nombres à virgule flottante ont dépassé la capacité maximale représentable.
ERROR: No return during execution of method	Une méthode n'a pas d'instruction de retour alors qu'elle est censée en avoir une d'après sa signature.
ERROR: Cast error	Tentative de conversion de type inappropriée

ERROR: Heap overflow	Débordement de tas
----------------------	--------------------

Erreurs contextuelles:

Message d'erreur	Signification
Incompatible cast	Tentative de conversion incompatible d'un type à un autre.
Wrong types on a modulo operation	Types incorrects dans une opération modulo. L'opération modulo (%) doit être utilisée uniquement avec des types numériques (entiers ou flottants).
Type error in a boolean operation	Utilisation incorrecte des types dans une opération booléenne.
Type error in a inequality comparison	Comparaison d'inégalité effectuée entre des types incompatibles.
Type error in a unary minus operation	Erreur de type dans une opération de moins unaire. L'opération de moins unaire (par exemple, -x) doit être appliquée à des types numériques.
Unknown identifier <nom>	L'identificateur spécifié n'est pas déclaré. Utilisation d'une variable, d'une classe, ou d'une fonction non définie.
Unknown type identifier	L'identificateur de type spécifié n'est pas défini. Utilisation d'un type personnalisé non défini.
The method name already exist in the super class as a non-method object	Tentative de définir une méthode avec un nom qui est déjà utilisé dans la classe mère, mais pour un objet non lié à une méthode (par

	exemple un attribut).
Overridden method hasn't the same signature as the initial one	Tentative de redéfinir une méthode avec une signature différente de celle de la méthode initiale.
The return type isn't a subType of the type returned in the inherited method	Le type de retour n'est pas un sous-type du type retourné dans la méthode héritée.
Type error in an exact comparison	Erreur de type dans une opération de comparaison exacte.
Parameter cannot be void in a method call	Tentative de passer un paramètre de type "void" lors de déclaration ou appel d'une méthode.
Incompatible assignment	Tentative d'assigner une valeur d'un type incompatible à une variable.
Wrong return type	La méthode retourne un type qui ne correspond pas à celui spécifié dans la déclaration.
Non boolean expression in a condition statement	Utilisation d'une expression qui ne produit pas une valeur booléenne dans une condition.
Trying to call a method on a non class object	Utilisation incorrecte d'une méthode sur un objet qui n'est pas une instance de classe.
Non-existent method identifier	Utilisation d'un identifiant de méthode qui n'a pas été préalablement déclaré.
<nom> is not a method identifier	L'identificateur spécifié n'est pas une méthode.
incorrect number of parameters in a method call	Nombre incorrect de paramètres fournis dans un appel de méthode.

incorrect parameter type in a method call	Utilisation d'un type incorrect pour au moins un paramètre dans un appel de méthode.
Trying to use New method on a non class object	Utilisation incorrecte de la méthode "New" sur un objet qui n'est pas une classe
Redefinition of an already existing class	Tentative de redéfinition d'une classe qui existe déjà.
Non-existent super class	La classe spécifiée comme classe mère n'existe pas.
Super class identifier must be a class type	L'identificateur spécifié comme super classe n'est pas une classe.
Error: type of expression in print is neither int nor float nor string!	L'expression dans l'instruction "print" doit être de type int, float ou string.
Using a Selection on a non class Object	Une sélection est effectuée sur un objet qui n'est pas de type classe.
Using a Selection method on a Class Object which is not a sub class of the current class	Appel d'une méthode de sélection sur un objet de classe qui n'est pas une sous-classe de la classe courante.
Using a Selection method in a class which is not a sub class of the Field class	Appel d'une méthode de sélection dans une classe qui n'est pas une sous-classe de la classe de champ.
[fieldIdentifier] used in a selection is not defined	Utilisation d'un identifiant de champ qui n'a pas été déclaré.
[fieldIdentifier] is used in a selection but it's not a field identifier	Utilisation d'un identifiant qui n'est pas un champ dans une opération de sélection.
The type of field [fieldName] cannot be void	Déclaration d'un champ avec le type "void".
The identifier [identifierName] in [className] is already defined in	L'identificateur est déjà défini dans la super classe, mais il n'est pas

the super class but his type is not field	de type champ.
Type error in an arithmetic operation	Erreur de type dans une opération arithmétique.
Variable type cannot be void	Déclaration d'une variable avec le type "void".
Variable already defined	Tentative de redéfinir une variable qui a déjà été déclarée.
Type error in a not operation	Erreur de type dans une opération logique "not".
Two symbols have the same name Symbol : [symbol] is declared several times	Déclaration de deux éléments (variables, classes, etc.) avec le même nom

Erreurs au lancement:

Message d'erreur	Signification
Failed to open output file	Impossible d'ouvrir le fichier de sortie spécifié pour l'écriture.
Failed to open input file	Impossible d'ouvrir le fichier d'entrée spécifié pour la lecture.
Error during compilation with decac	Une erreur s'est produite pendant la compilation avec decac
-r option should be followed by an integer between 4 and 16, skipping option	L'entier suivant l'option -r n'est pas compris entre 4 et 16. La compilation aura lieu sans cette option
-r option should be followed by an integer, skipping option	L'option -r n'est pas suivi d'un entier. La compilation aura lieu sans cette option
File doesn't end with .deca, skipping file	Le fichier source n'est pas un fichier .deca. Le fichier est ignoré par le compilateur.

Cannot find file <nom_fichier> , skipping file	Le fichier source est introuvable.
---	------------------------------------

LIMITATIONS ET PARTICULARITÉS

Notre compilateur dispose de quelques limitations.

Premièrement, l'usage des strings est limité, et la compilation des programmes en contenant n'est pas garantie. Dans des utilisations simples sans objet, ces strings peuvent néanmoins fonctionner correctement.

Deuxièmement, lorsque, dans une méthode, une nouvelle variable est déclarée avec un nom déjà utilisé pour un attribut, il est impossible de renvoyer cette nouvelle variable (c'est l'attribut qui sera renvoyé).

Troisièmement, tester "null instanceof A", où A est une classe, causera une génération de code assembleur incorrect.

GÉNÉRATION DE BYTECODE

Cette implémentation de decac permet de compiler un programme Deca en bytecode Java. Pour ce faire, il suffit de se conformer à la syntaxe : "decac -byte <fichier Deca>".

Cette étape nécessite l'exécution de javac, qui doit donc être accessible depuis le PATH.

Toutes les classes définies dans le programme Deca seront compilées dans des fichiers à part, en plus du main ; par exemple, si le programme contient les classes A et B, trois fichiers seront générés : A.class, B.class et Main.class. Ces fichiers se trouveront dans le même répertoire que le fichier Deca.

Bien que la compilation de plusieurs fichiers Deca soit possible par la syntaxe "decac -byte <fichier Deca 1> <fichier Deca 2> ...", cette utilisation entraînera des conflits (les fichiers des classes ou du main du dernier fichier Deca compilé pouvant remplacer ceux générés précédemment), et est fortement déconseillée.

Le bytecode peut ensuite être exécuté normalement.

GÉNÉRATION DE BYTECODE : LIMITATIONS

Lors de la génération du bytecode, il est nécessaire d'ajouter plusieurs attributs et méthodes aux classes Deca. Un conflit peut donc apparaître si une classe Deca contient un attribut ou une méthode dont le nom est préfixé par la chaîne “__reff58e4f6se4rs54gr86eruugrioutilsoeksfiougq87546qsd5dqzeg__” (les chances de conflit sont toutefois réduites)

De plus, certains programmes Deca corrects ne peuvent pas être compilés, car ils déclenchent des erreurs de compilation de javac lorsqu'ils sont traduits en Java. Cela concerne certains cas de unreachable code : tandis que des instructions telles que “while(false)” ou “if (0 != 0)” ne posent pas de problème, l'ajout d'instructions dans une méthode après un “return” dans une méthode empêche le programme de compiler correctement. Il faut donc s'abstenir de rajouter des lignes inatteignables, et, en particulier, après un return dans une méthode.