

Informations complémentaires

Nous avons choisi d'implémenter le **Switch** et le **if**.

Pour une vision d'ensemble des capacités de ce projet, le fichier `input - 1.1ea` récapitule plus ou moins tout ce qui est implémenté.

Analyse de type

Gestion des types sur les opérateurs

La gestion des types dans ce langage s'inspire du Java. Voici deux tableaux récapitulatifs de ce qui est possible.

opérateurs	int	char	float	string	boolean	
++	allowed	type error	type error	type error	type error	
--	allowed	type error	type error	type error	type error	
opérateurs	int/int	char/char	float/float	string/string	boolean/boolean	output
+=	allowed	allowed	allowed	allowed	type error	//
-=	allowed	allowed	allowed	type error	type error	//
*=	allowed	allowed	allowed	type error	type error	//
/=	allowed	allowed	allowed	type error	type error	//
%=	allowed	allowed	allowed	type error	type error	//
+	allowed	allowed	allowed	allowed	type error	//
-	allowed	allowed	allowed	type error	type error	//
*	allowed	allowed	allowed	type error	type error	//
/	allowed	allowed	allowed	type error	type error	//
%	allowed	allowed	allowed	type error	type error	//
&=	allowed	allowed	allowed	type error	type error	integer
=	allowed	allowed	allowed	type error	type error	integer
<<	allowed	allowed	allowed	type error	type error	//
>>	allowed	allowed	allowed	type error	type error	//
<<=	allowed	allowed	allowed	type error	type error	//
>>=	allowed	allowed	allowed	type error	type error	//
&&	type error	type error	type error	type error	allowed	boolean
	type error	type error	type error	type error	allowed	boolean
&&=	type error	type error	type error	type error	allowed	boolean
=	type error	type error	type error	type error	allowed	boolean
&	voir notes	voir notes	voir notes	voir notes	voir notes	voir notes
	voir notes	voir notes	voir notes	voir notes	voir notes	voir notes
!	type error	type error	type error	type error	allowed	boolean
<	allowed	allowed	allowed	allowed	type error	//
<=	allowed	allowed	allowed	allowed	type error	boolean

opérateurs	int/int	char/char	float/float	string/string	boolean/boolean	output
>	allowed	allowed	allowed	allowed	type error	boolean
==	allowed	allowed	allowed	allowed	type error	boolean
!=	allowed	allowed	allowed	allowed	type error	boolean
>=	allowed	allowed	allowed	allowed	type error	boolean

// = Même type qu'en entrée(s)

Notes et spécificités

- Pour les opérateurs & et |, la syntaxe les considère comme des && et || booléens. Ainsi, il n'y a pas de différence possible dans les Stree, il faut donc utiliser &= et |= pour réaliser une opération binaire.
- L'incréméntation et décrémentation (var++, var--) s'effectuent uniquement sur le type INTEGER
- Le CJUMP dans le StreeIF vérifie que la condition soit égale à 1, c'est-à-dire true
- Pour le type FLOAT, la valeur est stockée sous forme binaire puisque CONST ne peut prendre qu'un entier
- Il est possible que l'affichage de l'arbre ne respecte pas l'ordre gauche et droit des noeuds, ce qui peut prêter à confusion. Pour remédier au problème, on peut rajouter au début du .dot les commandes suivantes:

```
constraint=false;
rank=same;
```

Répartition des tâches

Pour ce projet, nous avons décidé de répartir tous les Stree de façon aléatoire entre les membres du groupe. Pour StreeIF et StreeSWITCH en revanche, ceux-ci ont été fait ensemble puisque plus difficiles. Il y a eu quelques Stree qui n'avaient aucun liens avec If et Switch et ceux-ci ont été supprimés des listes (barrés).

Voici la liste :

Clément Albanel

[X] StreeTIMESAFF.java
[X] StreeSUCC.java
[X] StreeRETURN.java
~~[X] StreeTABLE.java~~
[X] StreeGT.java
[X] StreeBORAFF.java
[X] StreeMINUS.java
[X] StreeLE.java
[X] StreeDIVAFF.java
[X] StreeAND.java

Gilles Souton

[X] StreeDIV.java
[X] StreeGE.java
[X] StreeORAFF.java
[X] StreeRSHIFT.java
[X] StreeMINUSAFF.java
[X] StreeNE.java
[X] StreeDEFAULT.java
[X] StreeLSHIFTAFF.java
[X] StreeBANDAFF.java
~~[X] StreeBREAK.java~~
[X] StreeSEQ.java

Iantsa Provost

[X] StreeSLOT.java
[X] StreeDEC.java
~~[] StreeIDENTIFIER.java~~
[X] StreeEQ.java
[X] StreeCASE.java
[X] StreeTIMES.java
[X] StreeANDAFF.java
~~[] StreeCALL.java~~
~~[] StreeDO.java~~
[X] StreePRODUCT.java
~~[] StreeARGS.java~~
[X] StreeNULL.java

Pierre Boisselier

[X] StreeFLOAT.java
~~[] StreeRANGE.java~~
[X] StreeTHENELSE.java
[X] StreeOR.java
[X] StreeLT.java
[X] StreePLUSAFF.java
[X] StreeSTRING.java
[X] StreeNOT.java
[X] StreeLSHIFT.java
[X] StreeRSHIFTAFF.java
[X] StreeCHAR.java

Effort commun

[X] StreeSWITCH.java
[X] StreeIF.java