**Classe programme**

Méthodes :

* *Syntaxe* : vérifier si on a un let ou un print dans le premier mot de la ligne

Si on a aucun des deux => erreur

**LET** : créer un objet de type variable avec la chaine avant ‘=’

- Tester si c’est une variable : si faux => erreur

- Sinon continuer l’évaluation :

Vérifier si la chaine après ‘=’ est une expression (Créer un objet de type expression :

Si exp => ok sinon => Erreur

- Si ok on ajoute la variable dans la table

**PRINT** :

Classes héritées :

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**~~Classe abstraite Symboles~~** ~~(pour les fonctions et les variables)~~

~~Variables :~~

*~~Protected~~* ~~table de symboles de type map~~

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Classe Abstraite Elément**

*Protected* table de symboles de type map

Méthodes :

Abstract IsItElement

Print

Classes héritées : *Nombre variable expression fonction*

**\*\* Expression**

**\*\*Variable**

Méthodes :

* *IsItVariable* : pour s’assurer si c’est une variable (aa ou a1)
* *IsFound* : s’assurer si la variable existe dans la table des symboles
* Implémentation PRINT : récupérer la valeur dans la table si elle existe sinon **ERREUR**

Variables :

Var : nom de la variable

**\*\*Fonction**

Méthodes :

* IsItFonction :

Pour s’assurer si c’est une fonction (Nom + parenthèse ouvrante + paramètres (élément ou expression déjà définis) + parenthèse fermante)

\*Si une des conditions manque => message d’erreur avec précision de l’erreur

*Dans le constructeur : si true => this.fct = fct*

* Implémentation PRINT

Variables :

Paramètre et nom de fonction (si c’est une fonction prédéfinie

Variable résultat (contient un résultat si isitfunction est true)

\*\*\*\*chercher comment transformer une chaine de caractères en fonction

**\*\*Nombre**

Méthodes :

* *IsItNumber* : Pour s’assurer si c’est un nombre

Si c’est un nombre : constructeur this.nb = nb (variable de classe)

* Implémentation PRINT :

Si true écrire nb sinon Erreur : ce n’est pas un nombre

Variables :

Nb : nombre

**\*\*Expression**

Variables : t1 t2

Constrcuteur Expression : t1 t2

Constrcuteur2 Expression : t1=t1 t2=0

Methodes :

Addition : t1 + t2

Soustraction : t1 – t2

**Enumération Fonctions standards**

SIN, COS, TAN, LOG, ABS, SQRT

**Classe Facteur**

Variables:

El1, el2 : Objet type element

Constructeur Facteur : el1 el2

Constrcucteur 2 Facteur : el1=el1 el2=1

Méthodes :

* Puissance : el1 ^ el2
* IsItFacteur : si c’est un facteur

Si e1 élément et e2 element

Classe Terme

Variables : f1 f2 : objet de type facteur

Constrcucteur Terme : f1 f2

Constrcucteur2 Terme : f1=f1 f2=1

Methodes :

Multiplication : f1 \* f2

Division : f1 / f2 f2=/0 sinon Erreur

IsItTerme : si c’est un terme

Si