

Rapport de projet de POO

I) Installation

Il faut d'abord désarchiver le projet : « `tar -xf ProjetPOO_Thomas_Kuntz.tar.gz` ». Veuillez être sûr d'avoir mis le tar.gz dans un répertoire dédié au test du programme au préalable car la commande va tout désarchiver dans le répertoire courant.

Si vous vous trouvez sur un environnement Linux, pour installer, il suffit de taper « `make` » dans le terminal (si vous êtes dans le répertoire du projet). Tous les .class seront créés et placés dans le répertoire « `bin` ».

Si vous n'êtes pas sous Linux, les .class créés chez moi sont dans « `bin` ».

La classe à lancer pour exécuter le programme est « `ProjetPOO` ».

Pour lancer le programme, taper simplement « `java ProjetPOO` » tout en étant dans le répertoire « `bin` » dans le terminal si vous êtes sous Linux.

II) Diagramme des classes

Voir le fichier `DiagrammeClasse.jpg` dans le répertoire du projet.

III) Syntaxe est grammaire autorisé par le programme

Le programme n'a pas besoin qu'on lui donne les expressions arithmétiques en notation polonaise inversée : le programme arrive à analyser une expression écrite avec les règles standards (autrement dit, comme on les écrits tous les jours). Il suffit simplement de respecter la grammaire des symboles des opérations.

Fonction binaires :

Les fonctions binaires ont besoin de deux opérandes : `arg1` et `arg2`, qui peuvent être n'importe quelle autre expression arithmétique (avec ou sans parenthèses selon le besoin de priorité).

Opération	Symbole	Grammaire	Exemple
Addition	+	<code>arg1 + arg2</code>	<code>a+b</code>
Soustraction	-	<code>arg1 - arg2</code>	<code>a-b</code>

Multiplication	*	arg1*arg2	a*b
Division	/	arg1/arg2	a/b

Fonction Unaire :

Les fonctions unaires n'ont besoin que d'un seul argument 'arg', ainsi que pour certaines d'une constante 'n' (fonction puissance et exponentielle).

Opération	Symbole	Grammaire	Exemple
Cosinus/Sinus/Tangente	cos/sin/tan	cos(arg)/sin(arg)/ tan(arg)	cos(1+x)/sin(x) tan(2*x)
Exponentielle naturel	e^	e^arg e^(arg)	e^(2*t) e^t
Exponentielle de base a	n^	n^(arg) n^arg	2^x 2^(4+y)
Logarithme naturel	ln	ln(arg)	ln(1+x)
Opposé	-	-arg -(arg)	-X -(a+b)
Puissance	^n	arg^n	(a+b)^2 x^3
Racine carré	sqrt	sqrt(arg)	sqrt(x) sqrt(a*b)

Et pour finir les constantes et inconnues/variables. Tout nombre sera considéré comme une constante, et toutes chaînes de caractère de répondant à aucun motif montré ci-dessus (incluant des symboles des opérations). Sera donc pris comme nom d'inconnu « 2x » car ce n'est ni un nombre, ni ne contient une opération : Pour correctement donner l'expression au programme, il faut taper « 2*x ».

IV) Règles de dérivations et simplification utilisées.

Dérivation :

Addition	(arg1+arg2)'	arg1'+arg2'
-----------------	---------------------	--------------------

Soustraction	$(\arg 1 - \arg 2)'$	$\arg 1' - \arg 2'$
Multiplication	$(U * V)'$	$U' * V + U * V'$
Division	$(U/V)'$	$(U'V - UV')/V^2$
Cosinus Sinus Tangente	$\cos(x)'$ $\sin(x)'$ $\tan(x)'$	$-\sin(x)$ $\cos(x)$ $\tan^2(x) + 1$
Exponentielle naturel	$(e^x)'$	e^x
Exponentielle de base a	$(a^x)'$	$\ln(a) * a^x$
Logarithme naturel	$\ln(x)'$	$1/x$
Opposé	$-x'$	$-(x')$
Puissance	$(X^n)'$	$n * X^{(n-1)}$
Racine Carré	$\text{Sqrt}(x)'$	$1/(2 * \text{sqrt}(x))$
Constante	K	0
Inconnue	X	1 si on dérive selon X 0 sinon

A cela on ajoute que pour toute fonction composée $f(g(x))'$, on associe la dérivé $g'(x) * f'(g(x))$.

Dans le programme, quand on dérive, après avoir appliqué les règles de dérivation, on applique une simplification sur le nouveau arbre d'expression créé.

Simplification :

Addition	$X + 0$ ou $0 + X = X$ $X + X = 2 * X$ $X + (-X) = 0$ $n * X + X / X + n * X = (n+1) * X$ $n * X + m * X = (n+m) * X$ sinon $X + Y = X + Y$
Soustraction	Les mêmes que l'addition $X - X = 0$
Multiplication	$0 * X$ ou $X * 0 = 0$ $1 * X$ ou $X * 1 = X$

	$X * X = X^2$ Constante * Constante = Constante Constante * Oppose(x) = Oppose(x * Constante) $(a^X) * a = a^{(X+1)}$ $(a^X) * (a^Y) = a^{(X+Y)}$ $(X^n) * X = X^{(n+1)}$ $(X^m) * (X^n) = X^{(n+m)}$ sinon $X * Y = X * Y$
Division	$0/X = 0$ $X/X = 1$ Constante/Constante = Constante $X^n/X = X^{(n-1)}$ $X^n/X^m = X^{(n-m)}$ $(a^X)/a = a^{(X-1)}$ $(a^X)/(a^Y) = a^{(X-Y)}$ $(a/b)/(c/d) = (a*d)/(b*c)$ sinon $X/Y = X/Y$
Cosinus Sinus Tangente	Si l'argument est une constante, on simplifie en changeant l'opération par son évaluation : $\cos(\pi) = -1$
Exponentielle naturel	$e^{(\ln(X))} = X$ sinon $e^X = e^X$
Exponentielle de base a	$a^X = a^X$
Logarithme naturel	$\ln(e^X) = X$ $\ln(a^X) = X * \ln(a)$ $\ln(a*b) = \ln(a) + \ln(b)$ $\ln(a/b) = \ln(a) - \ln(b)$ sinon $\ln(x) = \ln(x)$
Opposé	$-(-X) = X$ $-(\text{Constante}) = -\text{Constante}$ (le négatif de la constante). sinon $-X = -X$
Puissance	$(X^n)^m = X^{(n*m)}$ $X^1 = X$ $X^{0.5} = \text{sqrt}(X)$ $X^0 = 1$ $X^{(-n)} = 1/(X^n)$

	sinon $X^n = X^n$
Racine Carré	Même que puissance
Constante	Constante
Inconnue	Inconnue

Dans le programme, la simplification passe par plusieurs étapes :

- 1) D'abord on simplifie localement : on simplifie les arguments, puis on renvoie une expression en fonction des arguments simplifiés et de l'opération. Cette première étapes permet de réduire un peu l'expression, mais ne la simplifie pas totalement.**
- 2) On développe l'expression**
- 3) On factorise le résultat du développement**
- 4) On re-simplifie localement**
- 5) Après ça l'expression est le plus simplifié possible**

V) Exemples

Les exemples dans le sujet fonctionnent parfaitement.

Mes exemples :

Simplification :

$$(a+b)^2 \rightarrow a^2+b*a+a*b+b^2$$

$$(2*b+b^2+5)*(b+a) \rightarrow 5.0*b+2.0*b^2+b^3+5.0*a+2.0*b*a+b^2*a$$

$$e^{(2.0*x+1.0)}/(e^x) \rightarrow e^{(1.0+x)}$$

$$(a*1.0+0.0*x)*x+a*x*1.0 \rightarrow 2.0*a*x$$

Dérivation :

$$x^2+y \text{ par } x = 2.0*x$$

$$a^4*b+b^2+5+c \text{ par } b = a^4.0+2.0*b$$

$$3*2^{(x+1)} \text{ par } x = 3.0*\ln(2.0)*2.0^{(x+1.0)}$$

$$\text{Sqrt}(2*x+3) \text{ par } x = 1.0/(\sqrt{(3.0+2.0*x)})$$

Kuntz Thomas

Webank Antoine

$$(a+3)*(4+a^2) \text{ par } a = 4.0+3.0*a^2+6.0*a$$

$$t-e^{(2*t)}+1/x \text{ par } t = 1.0-2.0*e^{(2.0*t)}$$

Evaluation :

$$x^2+y = 21 \text{ pour } x=4 \text{ et } y=5$$

$$(a+b)^2 = 25 \text{ pour } a=3 \text{ et } b=2$$

$$(2*b+b^2+5)*(b+a) = 65 \text{ pour } b=2 \text{ et } a=3$$

$$3*2^{(x+1)}=12.0 \text{ pour } x=1$$

$$(2*x^2+3*x+4)*(y+1) = 4 \text{ pour } x=0 \text{ et } y=0$$