

INSSET | UPJV

15 Janvier 2023

Projet : Calculatrice

Cours : Algorithmie

Section : MN

Nom : Emile

Prenom : Steevenson

Sommaire

- 1 - Introduction
- 2 - Objectifs
- 3 - Diagramme de classe
- 4 - Conception
 - 4.1 - module de moteur de calcul
 - 4.2 - module d'interface utilisateur
- 5 - Capture d'écran
- 6 - conclusion
- 7 - perspectives

Introduction

Dans le cadre du cours d'algorithmique, nous devons réaliser un programme de calculatrice. L'objectif de ce projet consiste en la réalisation d'une calculatrice qui sait calculer les expressions mathématiques contenant des parenthèses. Ainsi, Il a été demandé d'utiliser les arbres binaires et la notation polonaise pour sa réalisation.

Objectifs

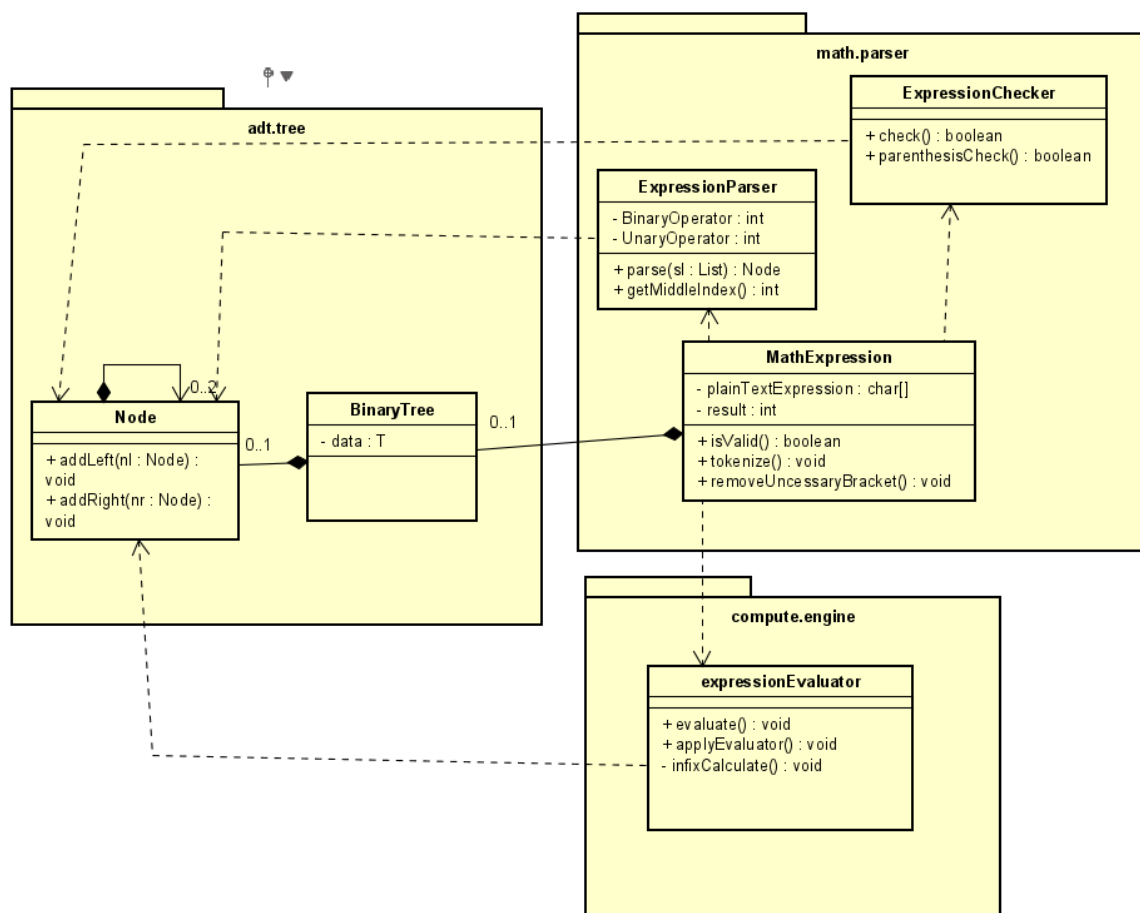
La calculatrice doit avoir un interface utilisateur

La calculatrice doit gérer les parenthèses

La calculatrice reconnaît les nombres décimaux.

Nous devons pouvoir ajouter de nouvelle operations a la calculatrice

Diagramme de classe



Conception

Module moteur de calcul : engine

Une calculatrice est composée de deux modules, un module d'interface et un module de moteur de calcul. Le module d'interface permet l'interaction entre l'utilisateur et le moteur de calcul. Le moteur de calcul permet de traduire l'expression arithmétique fourni par l'interface en expression calculable ou non pour enfin les calculer si possible et les mettre à disposition de l'interface en question.

Dans le module moteur de calcul pour favoriser la scalabilité d'une module moteur de calcul, c'est-à-dire ajouter des calcul d'expression plus complexes et d'algorithme plus performant, nous avons identifié les objets suivant :

- une Expression Mathématique : (**MathExpression**)
- un Parseur : (**ExpressionParser**),
- un vérificateur d'expression : (**ExpressionChecker**)
- un évaluateur d'expression : (**ExpressionEvaluator**).

Enfin deux classes Utilitaires **BinaryTree** et **Node**

Classes Node et Tree

Un Arbre Binaire est composé d'un Noeud racine. chaque Noeud est composé d'un Noeud Gauche et d'un Noeud droit ou pas.

Classe Expression Mathématique - MathExpression

Une classe d'expression mathématique est composée d'une liste d'opérateurs et valeurs. S'il est correct, elle est composée d'un arbre équivalent. Pour s'évaluer, Une Expression mathématique sait vérifier s'il est une expression correcte ou pas en utilisant un vérificateur d'expression. Il sait Ensuite trouver son arbre équivalent en utilisant Un Parseur d'expression, et enfin s'évaluer avec un Évaluateur d'expression

Classe Parseur - (ExpressionParser)

Un parseur d'expression est une entité qui a des stratégies de traduction d'expression, il est disponible à tout expression arithmétique. donc n'est pas nécessaire d'être appartenu a une expression en particulier.

Classe Vérificateur d'expression - (ExpressionChecker)

Un vérificateur d'expression est une entité qui a des stratégies de vérification de validité d'expression. qui peut être utilisé par un parseur ou par une expression mathématique.

Classe Évaluateur d'expression - ExpressionEvaluator

Un Évaluateur d'expression a des stratégies de calcul d'expression. Il sait choisir quelle stratégie de calcul utiliser pour le calcul d'une expression dépendamment par exemple de la profondeur ou la largeur de l'arbre équivalent de l'expression.

Il est possible d'ajouter de nouvelles stratégies de calcul, la reconnaissance de nouvelles opérations au parseur, vérification et évaluateur d'expression. par exemple si vous voulez ajouter l'opération racine, il suffit de :

- 1) Choisir un symbole de reconnaissance et le fournir au vérificateur, parseur, et évaluateur.
- 2) Ecrire l'algorithme de calcul de cette opération dans l'évaluateur.

En résumé, le module moteur de calcul de notre calculatrice est composé d'une série d'objets qui travaillent ensemble pour traduire les expressions arithmétiques en expressions calculables, vérifier leur validité, trouver leur arbre équivalent et finalement les évaluer. En utilisant les arbres binaires et la notation polonaise, nous avons pu créer un moteur de calcul efficace et extensible, qui peut gérer des expressions mathématiques complexes avec précision.

Conception de l'algorithme

ExpressionParser.parse(LinkedList<String> expressionList)

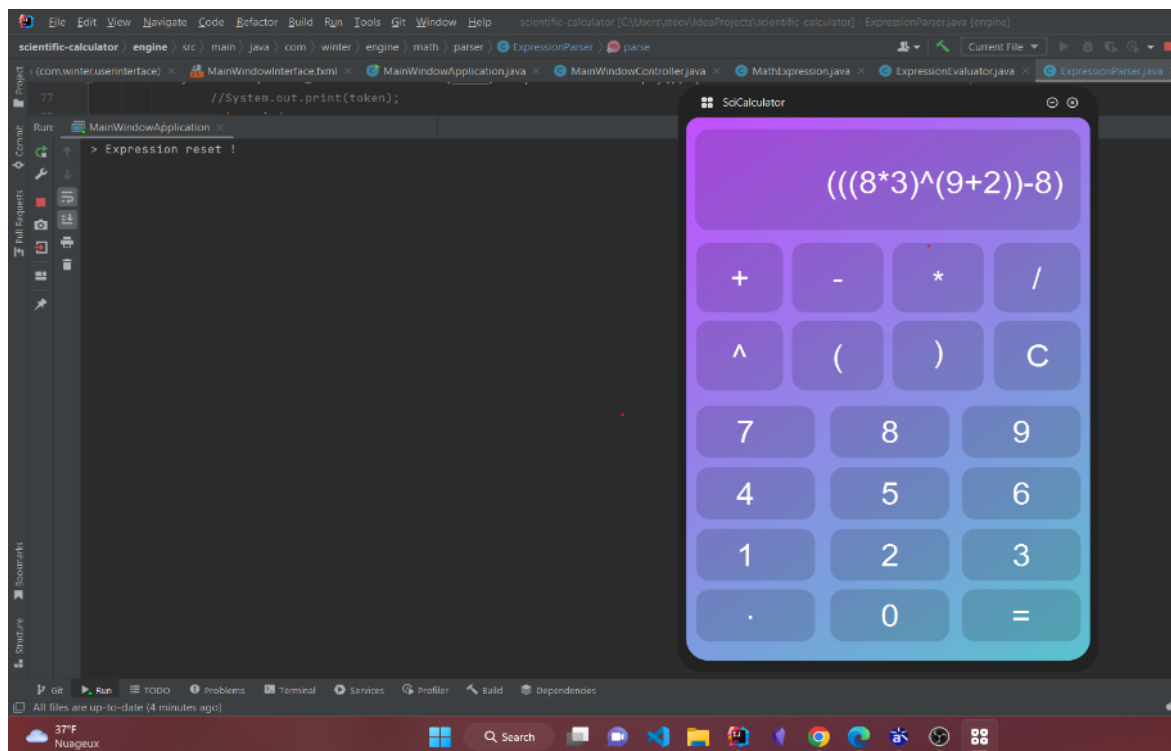
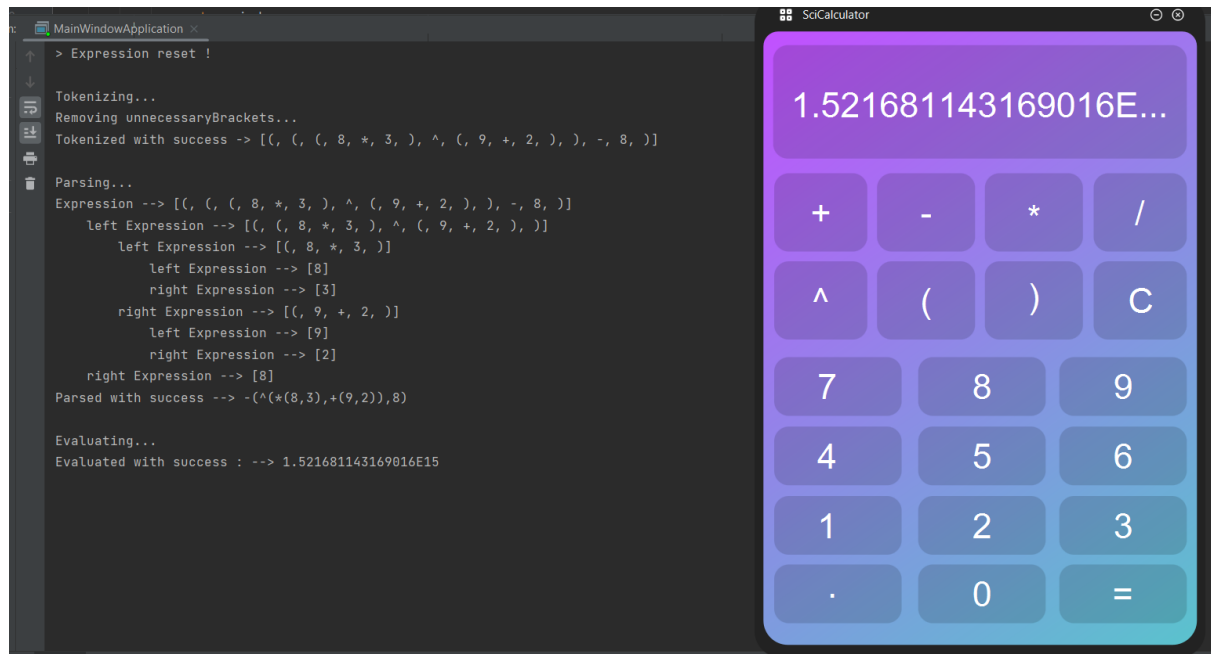
Le Parseur de cette version de la calculatrice ne supporte que les opérations binaires et les nombres positifs. De façon récurrente, il cherche l'opération au milieu et la met à la racine d'un arbre, il fait de même pour la liste de l'opérande de gauche et pour la liste de l'opérande de droite, et retourne un arbre.

Module interface Utilisateur :userinterface

Le design pattern MVC a été choisi afin d'ajouter plus de souplesse à la gestion de nouvelle fonctionnalité dans la calculatrice.

La calculatrice est à une version basique, donc tout peut se faire dans le contrôleur de la fenêtre `MainWindowController` y compris le modèle qui est réduit à quelques lignes de code.. et les vues sont décrites dans un langage de description de composants d'interface `FXML` qui est une variante de XML.

Capture d'écran



Conclusion

Nous avons réalisé une calculatrice qui peut faire des opérations binaire avec des parenthèses. Elle sait faire des opérations d'addition, de division, de soustraction, de multiplication et de puissance une. L'application est composée de deux modules : Interface utilisateur et moteur de calcul. Le module interface utilisateur est composé d'un seul package: le package interface qui contient pour l'instant le Contrôleur et le Vue pour l'interface utilisateur. Le module Moteur de calcul comprend trois packages, un package utilitaire `adt.tree`, Un package qui regroupe les classe d'expression et les classe qui interviennent dans la transformation d'expression et enfin le package moteur `compute.engine` qui comprend la classe évaluateur d'expression.

Perspectives

Comme perspective, vu l'architecture employée nous pouvons facilement intégrer de nouvelles opérations dans la calculatrice comme le calcul des intégrales. et les racines carrés. Il suffit d'ajouter une nouvelle méthode dans le parseur d'expression et dans l'évaluateur.