Bouyiatiotis - Jaquet

POO1 Labo08

Calculator

Table des matières

[Objectif 2](#_Toc531761507)

[Mise en œuvre : 2](#_Toc531761508)

[Exemple d’utilisation : 2](#_Toc531761509)

Labo 08 Calculator

# Objectif

Réaliser, en utilisant JCalculator.java (fichier fournis), une calculatrice qui permet de faire les opérations en utilisant la notation polonaise inverse.

Un diagramme de classe partiel est fourni. Il faudra le complété en spécifiant ce que nous allons utiliser pour State et Operator.

Il nous faudra compléter JCalculator.java ainsi que créer les classes State, stockant l’état de la machine non graphique de manière à pouvoir le réutilisé pour le mode console, et Operator permettant de réaliser les opérations de la calculatrice.

Il faudra aussi rendre possible l’utilisation de State et Operator pour le mode console.

## Mise en œuvre :

* Définir une hiérarchie de classes avec pour racine une classe Operator en factorisant au maximum le code. Cette classe devra posséder une méthode public void execute(), qui est automatiquement invoquée par l’interface à l’appui d’un bouton (initialisation effectuée dans la méthode JCalculator.addOperatorButton).
* Représenter l’état interne (non graphique) de la calculatrice (valeur courante, pile, erreur, etc).
* Remplacer les valeurs null des appels à addOperateurButton par des instances ad hoc de la classe Operator.
* Définir la méthode JCalculator.update(), invoquée après chaque opération, depuis addOperateurButton (qui ne devra pas être modifiée) pour réactualiser l’interface (valeur courante, état de la pile). Le rapport comprendra un diagramme des classes complet ainsi qu’une description des choix de

## Exemple d’utilisation :

Soit à évaluer l’expression (3.5 + 4) / (2.52 + 1). Avec cette calculatrice, les opérations seront:

* Appui des touches 3, . et 5, et Ent. La valeur 3.5 est placée sur la pile.
* Appui de la touche 4.
* Appui de la touche +: évaluation du résultat, 7.5.
* Appui des touches 2 (la valeur précédemment calculée, 7.5, est placée sur la pile), . et 5.
* Appui de la touche x^2: évaluation du résultat intermédiaire 6.25.
* Appui de la touche 1 (la valeur 6.25 est placée sur la pile).
* Appui de la touche +: évaluation du résultat intermédiaire 7.25.
* Appui de la touche /: évaluation du résultat final 1.0344827586206897.

# Diagramme des classes

## Description

Comme indiqué par le professeur les classes JCalculator, JButton et Stack ne vont pas être détaillé autant que les classes State et Operator.

### JCalculator :

La classe permet de gérer l’affichage sur l’interface graphique via :

* JTextField jNumber : Qui est l’affichage sur la ligne d’entrée.
* JList jStack : Qui affiche les éléments dans la stack.
* State state : Permet de garder en mémoire ce qu’il y a à afficher. Les attribues de JCalculator utiliseront les valeurs contenue dans state pour l’affichage.

Elle construit aussi l’affichage des boutons de commande sur la calculatrice et d’assigner les opérations avec : private void addOperatorButton(String name, int x, int y, Color color, final Operator operator) selon le bouton.

Permet avec la méthode update() de mettre à jour l’affichage après une opération.

### State :

C’est la classe qui permet de garder en mémoire les informations entrée par l’utilisateur. Pour cela la classe dispose :

* Stack<Double> values : Qui seront les valeurs contenue dans la stack.
* String currentDisplay : Qui gardera en mémoire les valeurs entrée sur la calculette mais qui ne sont pas encore stocké dans la stack.
* Double memory : Qui sera la valeur gardé en mémoire pour une utilisation future.

La classe Operator manipulera les informations à l’intérieurs de State alors que JCalculator ne fera que les affichers. State représente l’état mémoire de la machine.

### Operator :

La classe Operator gère toutes les opérations sur les boutons via une méthode execute() qui sera abstract car chaque opération sera découpé et donc demandera une redéfinition d’execute.

Operator est coupée en 3 grandes catégories :

* Arithmetic : Représente toutes les opérations arithmétiques de la calculette. Addition, soustraction, division etc... Toutes ses opérations manipulent directement la partie dit mémoire (la stack).
* Memory : Ce sera toutes les opérations qui enregistrent et/ou supprime les éléments de la stack. Comme le enter, backspace etc…
* Number : Sont les opérations qui manipule uniquement l’entrée du nombre. Comme les touches 1, 2, 3 etc… ou encore le « . » ou bien le changement de signe.