

Rapport du Laboratoire 7 : Calculatrice en notation polonaise inverse

Nathan Fülleemann – Mathéo Lopez

1. Introduction

Ce laboratoire consiste à implémenter une calculatrice graphique et en mode console en utilisant la **notation polonaise inverse (RPN)**, où les opérateurs suivent leurs opérandes. La calculatrice repose sur un modèle **Modèle-Vue-Contrôleur (MVC)** pour séparer la logique métier de l'interface utilisateur. Les fonctionnalités principales incluent :

- La gestion d'une pile pour stocker les résultats intermédiaires.
- Les opérations arithmétiques usuelles, y compris des fonctions avancées comme la racine carrée et l'inverse.
- La gestion de la mémoire (MS, MR).
- La possibilité de revenir sur une entrée avec l'opérateur **Backspace**.

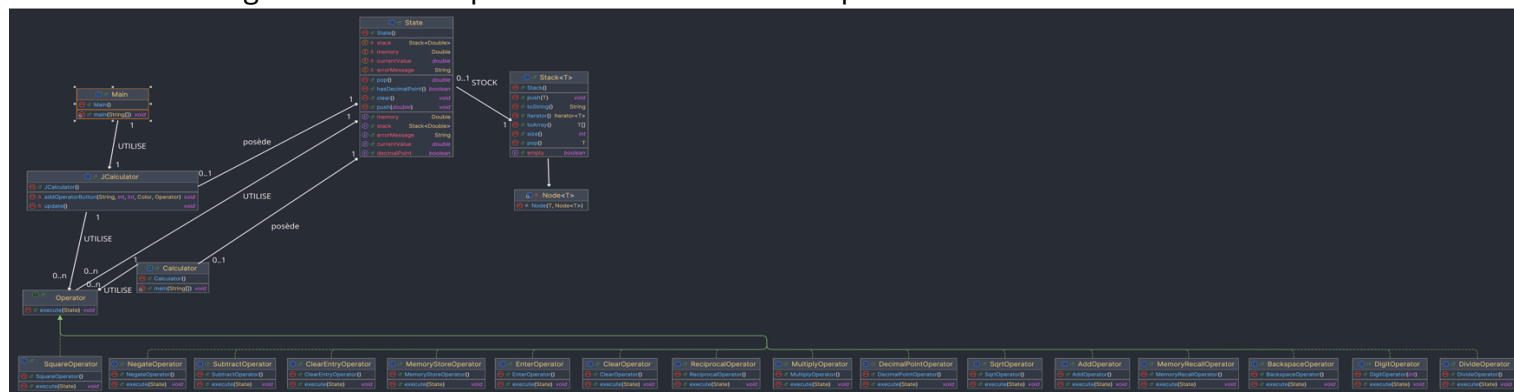
2. Exemple d'utilisation

Voici un exemple d'utilisation en mode graphique, où l'on évalue l'expression $(3.5+4)/(2.5^2+1)(3.5+4)/(2.5^2+1)(3.5+4)/(2.5^2+1)$:

1. Appuyez sur 3, . et 5, puis sur Ent. La valeur 3.5 est placée sur la pile.
2. Appuyez sur 4, puis sur +. La valeur 7.5 est calculée.
3. Appuyez sur 2, . et 5. La valeur 2.5 est placée sur la pile.
4. Appuyez sur x^2 . La valeur intermédiaire 6.25 est calculée.
5. Appuyez sur 1, puis sur +. La valeur intermédiaire 7.25 est calculée.
6. Appuyez sur /. Le résultat final 1.0344827586206897 est affiché.

3. Diagramme des classes

Le diagramme suivant présente une structure simplifiée de la calculatrice :



- **Stack** : Classe générique représentant une pile avec des méthodes pour empiler, dépiler, et parcourir ses éléments.

4. Implémentation

a. Classe Stack (dans le package util)

Une pile générique avec les fonctionnalités suivantes :

- **push(T value)** : Empile un élément sur le sommet.
- **pop()** : Dépile et retourne l'élément au sommet.
- **toString()** : Renvoie une représentation de la pile sous forme de chaîne.
- **toArray()** : Retourne un tableau d'objets représentant la pile.
- **iterator()** : Retourne un itérateur pour parcourir la pile.

b. Modèle (Classe State)

- **Attributs principaux** :
 - double currentValue : Valeur courante.
 - Stack<Double> stack : Pile pour stocker les résultats intermédiaires.
 - String errorMessage : Message d'erreur éventuel.
 - boolean hasDecimalPoint : Indique si un point décimal a été ajouté.
- **Méthodes clés** :
 - pushValue(double value) : Ajoute une valeur à la pile.
 - popValue() : Retire une valeur de la pile.
 - setCurrentValue(double value) : Met à jour la valeur courante.

c. Contrôleurs (Sous-classes d'Operator)

Chaque opérateur implémente l'interface Operator et définit la méthode execute(State state). Exemples :

- **AddOperator** : Effectue l'addition des deux dernières valeurs de la pile.
- **SqrtOperator** : Calcule la racine carrée de la valeur au sommet.
- **BackspaceOperator** : Supprime le dernier caractère d'une valeur en cours d'introduction.
- **MemoryStoreOperator et MemoryRecallOperator** : Gèrent le stockage et la récupération de la mémoire.

d. Interface graphique (JCalculator)

- Implémentée en héritant de JFrame.
- Les boutons (JButton) sont associés à des instances d'Operator via la méthode addOperatorButton.
- Méthode update() :
 - Met à jour l'affichage de la valeur courante.
 - Réactualise le contenu de la pile.

5. Extension : Mode console

La classe Calculator offre une version console de la calculatrice.

Exemple d'exécution :

```
plaintext
> 1 <enter>
1.0
> 2 <enter>
```

```
2.0 1.0
```

```
> +
```

```
3.0
```

```
> sqrt
```

```
1.7320508075688772
```

```
> exit
```

Cette classe réutilise State et les sous-classes d'Operator.

6. Liste des tests

Grille de tests :

Fonctionnalité	Cas de test	Résultat attendu
Addition	3 + 5	8
Division par zéro	10 / 0	Message "Division par zéro"
Backspace	123, puis <-	12
Mémoire (MR/MS)	MS, puis MR	La mémoire est correctement restituée
Calcul complexe	$(3.5 + 4) / (2.5^2 + 1)$	1.0344827586206897

7. Liste des problèmes rencontré

Nous avons dû faire face à un problème où, dès qu'un nouveau chiffre était ajouté, il était à la fois considéré comme un entier et un nombre à virgule flottante (double). Cela entraînait son ajout après la virgule, même si le bouton de la virgule n'avait pas encore été pressé. Pour résoudre ce problème, nous avons implémenté une logique qui traite les chiffres comme des entiers tant que le bouton de la virgule n'a pas été pressé. Une fois le bouton virgule activé, les chiffres suivants sont considérés comme des décimaux (double).

8. Conclusion

Ce projet met en œuvre une calculatrice graphique et console en notation polonaise inverse, en respectant le design pattern MVC. Les défis incluent la gestion d'une pile, la manipulation de l'état interne et la synchronisation avec l'interface graphique. Ce laboratoire met en avant la modularité et la réutilisabilité des classes.