src/Calculator.java

```java
/**
 * Représente une calculatrice interactive fonctionnant en console.
 * Elle permet d'exécuter des opérations arithmétiques et des manipulations d'état
 * en saisissant des commandes textuelles.
 *
 * @Author : Maxime Lestiboudois
 * @Author : Nathan Parisod
 * @date : 27/11/2024
 */
import calculator.*;
import java.util.Scanner;
public class Calculator {
/** Représente l'état interne de la calculatrice, incluant la pile et la mémoire.*/
    private final State state;

/** Permet de lire les entrées de l'utilisateur via la console.*/
    private final Scanner scanner;

    /**
     * Constructeur par défaut qui initialise la calculatrice.
     * Crée un état vierge et configure le scanner pour les entrées utilisateur.
     */
    public Calculator() {
        this.state = new State();
        this.scanner = new Scanner(System.in);
    }

    /**
     * Démarre la boucle principale de la calculatrice.
     * Permet à l'utilisateur de saisir des commandes ou des opérandes.
     */
    public void start(){
        boolean stay = true;
        while(stay){
            System.out.print("> ");
            String input = scanner.nextLine().trim();

            if(input.equals("exit")){
                break;
            }

            stay = processInput(input);

            displayState();
        }
    }

    /**
     * Traite une entrée utilisateur.
     * Si l'entrée est un nombre, elle est ajoutée à la pile.
     * Si l'entrée correspond à un opérateur, celui-ci est exécuté.
     *
     * @param input La chaîne saisie par l'utilisateur.
     * @return {@code true} si la calculatrice doit continuer à fonctionner,
     *         {@code false} sinon (lorsqu'un opérateur "exit" est saisi ou une erreur survient).
     */
    private boolean processInput(String input){
        if(isNumber(input)){

            for(char c : input.toCharArray()){
                new NumberOperator(c - '0', state).execute();
            }
            new EnterOperator(state).execute();
```

```
    }
    else {
        // Si l'entrée est un opérateur, trouver et exécuter l'opération correspondante
        switch (input) {
            case "+":
                new OperandOperator(new Addition(),state).execute();
                break;
            case "-":
                new OperandOperator(new Substraction(), state).execute();
                break;
            case "*":
                new OperandOperator(new Multiplication(), state).execute();
                break;
            case "/":
                new OperandOperator(new Division(), state).execute();
                break;
            case "x^2":
                new SquareOperator(state).execute();
                break;
            case "sqrt":
                new SqrtOperator(state).execute();
                break;
            case "1/x":
                new FractionnalOperator(state).execute();
                break;
            case "c":
                new COperator(state).execute();
                break;
            case "exit":
                return false;
            default:
                System.out.println("Opérateur inconnu");
                return false;
        }
    }
    return true;
}

/**
 * Affiche l'état actuel de la calculatrice.
 * Si une erreur est présente, affiche "Error". Sinon, affiche la pile actuelle.
 */
private void displayState(){
    if
    (state.hasError()){
        System.out.println("Error");
    }
    else{
        System.out.println(state.getStack().toString());
    }
}
/**
 * Vérifie si une chaîne correspond à un nombre entier valide.
 *
 * @param input La chaîne à vérifier.
 * @return {@code true} si la chaîne est un nombre entier, {@code false} sinon.
 */
private boolean isNumber(String input){
    try {
        Double.parseDouble(input);
        return true;
    } catch (NumberFormatException e) {
        return false;
    }
}
```

```java
    /**
     * Point d'entrée de l'application.
     * Initialise et démarre une nouvelle instance de la calculatrice.
     *
     * @param args Les arguments de ligne de commande (non utilisés).
     */
    public static void main(String[] args){
        new Calculator().start();
    }









}
```