Stack-Based Calculator

By: Yashika

|  |  |  |
| --- | --- | --- |
| **S. No** | **Table of Content** | **Page**  **No.** |
| **1.** | **Project Title** | 1 |
| **2.** | **Description of Project**  **2.1 Problem Statement** | 3 |
| **3.** | **Problem analysis**   * 1. **Hardware Requirements**   2. **Software Requirements** | 4 |
| **4.** | **Design**   * 1. **UML Class Diagram**   2. **Flow Diagram** | 5 |
| **5.** | **Output** | 7 |
| **6.** | **Conclusion and Future Scope** | 10 |

# Project Description

### Arithmetic expressions come in various forms, each with its own advantages and use cases. The challenge lies in creating a calculator that can seamlessly convert between different expression notations (infix, postfix, and prefix) and accurately evaluate the results. The need for such a tool arises from the complexity of mathematical expressions and the diverse requirements of users who may prefer or encounter different expression formats.

The Comprehensive Expression Evaluator and Converter is a feature-rich calculator designed to handle diverse arithmetic expressions. The program supports the conversion between infix, prefix, and postfix notations, along with the evaluation of expressions in each format. Developed using a stack- based approach, the calculator provides a versatile tool for mathematicians, programmers, and students alike. It supports basic mathematical operators such as Addition (+) , Subtraction (-) , Multiplication (\*) and division (/).

* 1. Problem Statement

### Create a calculator that uses a stack data structure to evaluate arithmetic expressions, including infix to postfix conversion.

*Definition of Done:*

* + - User can input arithmetic expressions with basic operators (+, -, \*, /).
    - The calculator can convert infix expressions to postfix notation.
    - The calculator can evaluate postfix expressions.
    - The application handles invalid expressions and provides meaningful error message

# Problem Analysis

## Hardware Requirements

### Processor:

* + A modern processor with sufficient processing power to handle graph traversal algorithms efficiently.

### Memory (RAM):

* + Adequate RAM to store the metro network graph and support the execution of the shortest path algorithm. The size of the graph and the algorithm's memory requirements will influence the RAM specifications.

### Storage:

* + Disk space is not a significant concern for this type of application, as the program's data is relatively small. However, ensure there is enough storage for the software and any potential future updates.

## Software Requirements

### Java Development Kit (JDK):

* + Install a compatible JDK to compile and run Java code. The project appears to be written in Java, so having a suitable JDK version is essential.

### Integrated Development Environment (IDE):

* + Choose an IDE for Java development, such as IntelliJ IDEA, Eclipse, or NetBeans. An IDE provides a convenient environment for coding, debugging, and managing the project.

### Operating System:

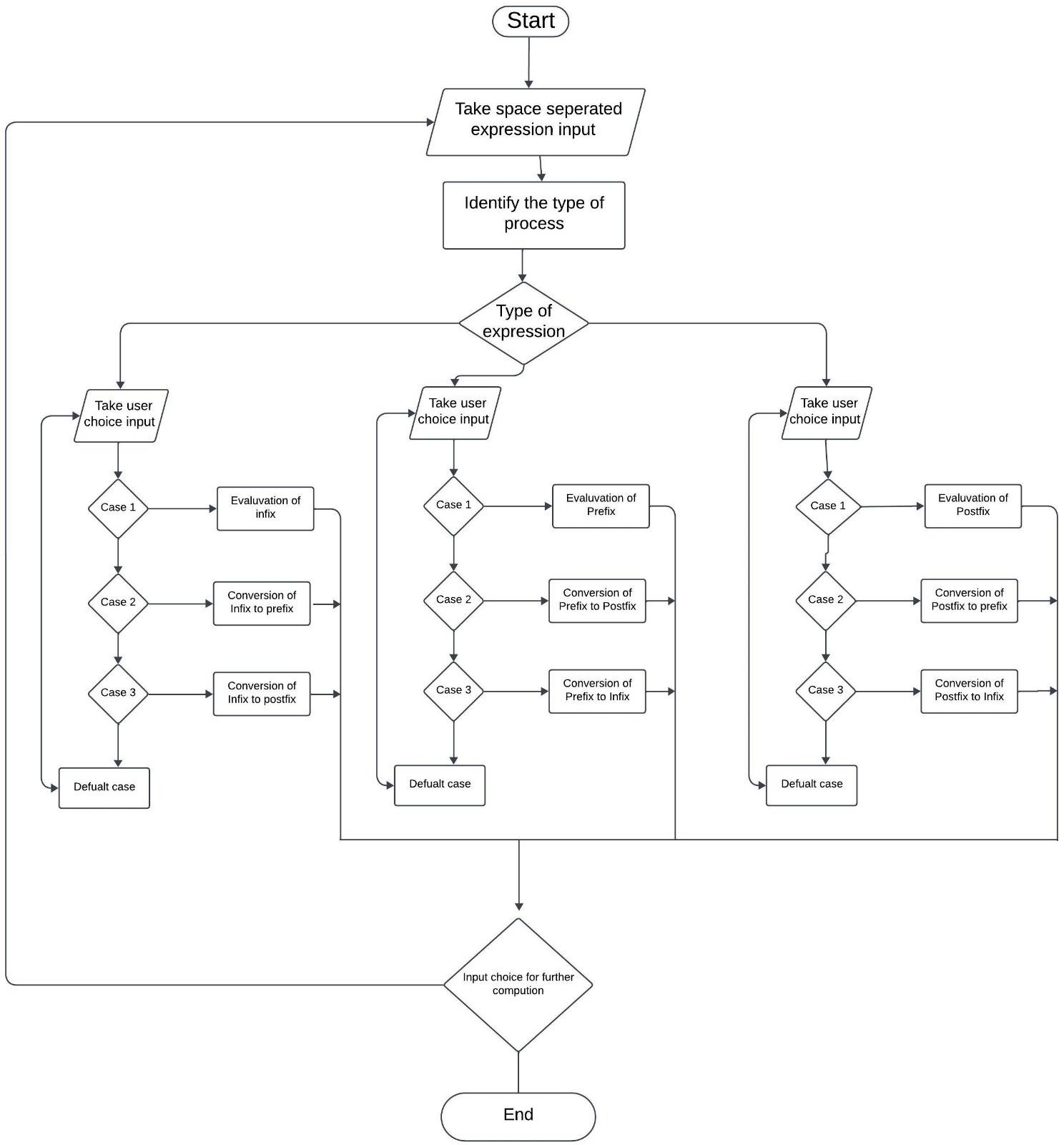
* + The project is written in Java and should be compatible with different operating systems (Windows, Linux, macOS). Ensure that the chosen IDE and JDK are compatible with the operating system of choice.

### Version Control (Optional):

* + If collaborative development or version control is a consideration, using a version control system like Git and platforms like GitHub, GitLab, or Bitbucket can enhance collaboration and code management.

# Design

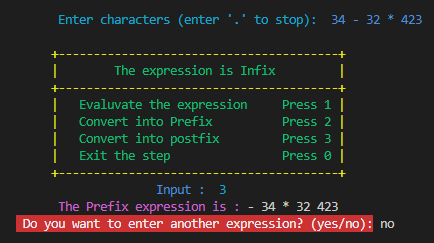
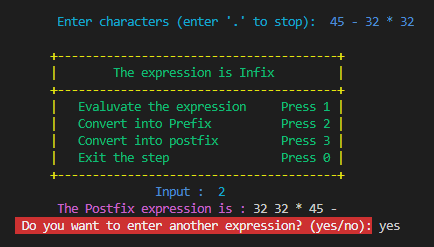
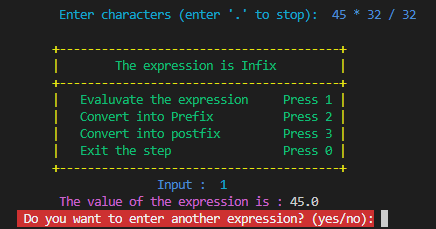
## UML class diagram

* 1. Flow Diagram

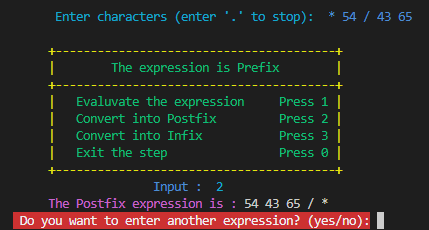
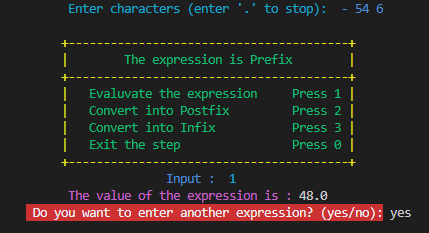
# Output

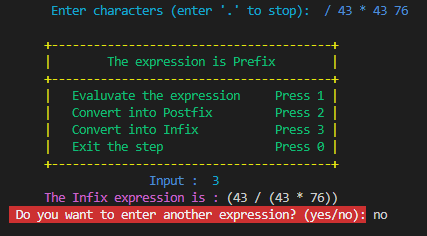
### Infix Expressions :

[

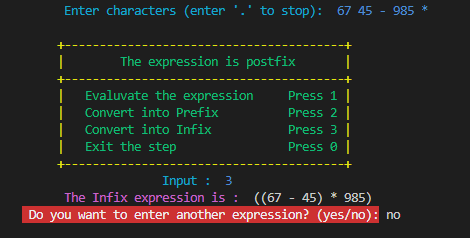
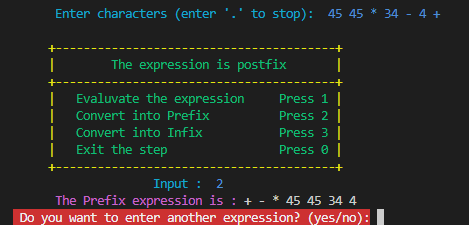
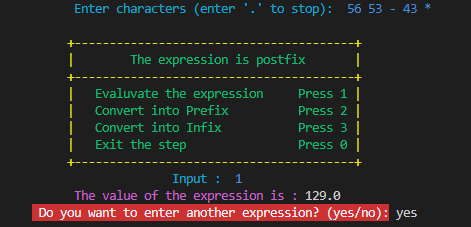


### Prefix Expressions:





Postfix Expressions:



# Conclusion and Future Scopes

The Comprehensive Expression Evaluator and Converter not only meets the immediate need for a versatile arithmetic calculator but also sets the stage for future expansions. Future enhancements could include:

* **Support for Additional Operators:** Extend the calculator's capabilities by incorporating a broader range of mathematical operators and functions.
* **Graphical User Interface (GUI):** Develop a graphical interface for a more intuitive user experience, especially for users less familiar with command-line interfaces.
* **Expression Simplification:** Implement algorithms to simplify complex expressions, providing users with more concise and readable results.
* **Integration with External Tools:** Explore opportunities to integrate the calculator with other tools or systems, enhancing its utility in diverse contexts.

The modular and extensible design of the calculator ensures adaptability to emerging requirements and positions it as a valuable asset for users engaging with mathematical expressions in various formats.