Java Calculator

R Aravind-AM.EN.U4AIE21151
Vishnu Shaji-AM.EN.U4AIE21167
Karthik Nair-AM.EN.U4AIE21138
Akhilesh-AM.EN.U4AIE21108
Mamidi Sowji Krishna-AM.EN.U4AIE21181

Introduction

We have created the application based on our understanding of the object-oriented concepts of inheritance, polymorphism, aggregation. We have also used Data Structures like Stack, Queue, Array and Linked List.

The calculator application performs basic mathematical calculations and is presented to user in a friendly and easy to use way and provides appropriate messages when wrong input is given by the user.

Problem Definition

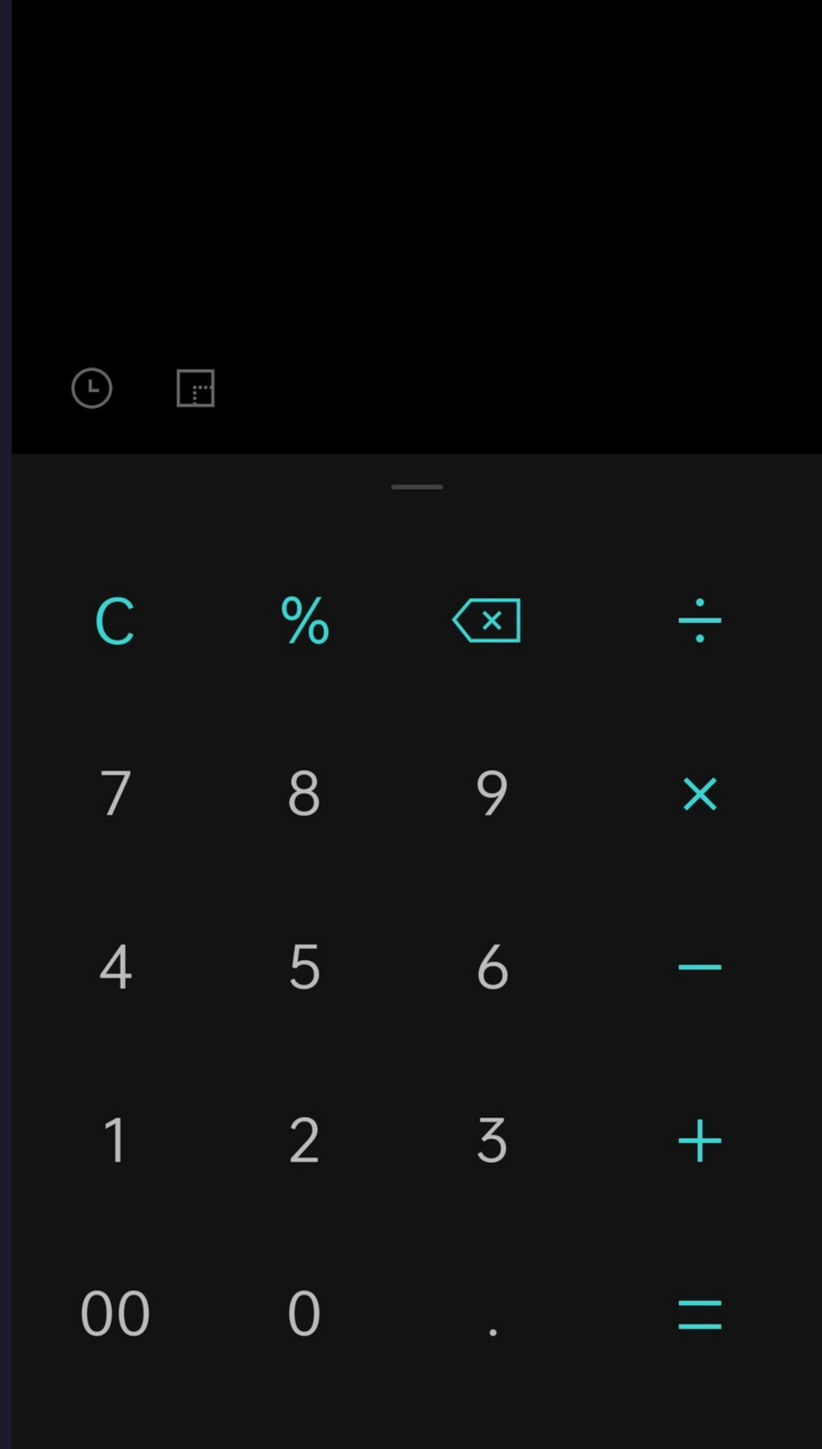
Our main objective in this project is to create a Calculator where people can calculate different values using the given operators.

People can add the value into the calculator memory to use it later and also remove it from the memory.

43! = ?

5483+2314 = ?

438 x 329 = ?



Problem Definition

Difficulty : Deciding which operator to be evaluated first.

Comparison :
Addition First : 20 + 14 * 15 = 510
Multiplication First : 20 + 14 * 15 = 230

Which is correct ?

Problem Approach

Priorities of the operators are predefined.

We will consider five different operators.

Operators	Priority
^ (Exponentiation)	1 (highest)
* (Multiplication) and / (Division)	2
+ (Addition) and – (Subtraction)	3 (lowest)

Infix to Postfix

Infix Expression

A + B / C * D – E / (F + G)

Postfix Expression

A + B / C * D – E / (F G +)

A + B C / * D – E / (F G +)

A + B C / D * – E / (F G +)

A + B C / D * – E (F G +) /

A B C / D * + – E (F G +) /

A B C / D * + E F G + / -

Expression Representation

Infix Notation - an operator is written in between two operands.

a + b

Postfix Notation – the operators appear after operands.

ab+

Postfix Expression Evaluation

Infix Expression : 3 + 5 * (5 / 5) – 2 ^ 2

Postfix Expression : 3 5 5 5 / * + 2 2 ^ -

The expression will be scanned from left to right and as soon as we will encounter an operator, we will apply it to the last two operands.

3 5 5 5 / * + 2 2 ^ -

3 5 5 5 / * + 2 2 ^ -

3 5 5 5 / * + 2 2 ^ -

3 5 5 5 / * + 2 2 ^ -

3 5 5 5 / * + 2 2 ^ - -> Apply '/' to 5 and 5

Postfix Expression Evaluation

3 5 5 5 / * + 2 2 ^ -	- > Apply '/' to 5 and 5
3 5 1 * + 2 2 ^ -	-> Apply '*' to 5 and 1
3 5 + 2 2 ^ -	-> Apply '+' to 3 and 5
3 5 + 2 2 ^ -	-> Apply '+' to 3 and 5
8 2 2 ^ -	-> Apply '^' to 2 and 2
8 4 -	-> Apply '-' to 8 and 4
12	

DATA STRUCTURES USED

1.LinkedList

LinkedList is a sequence of Data Structures, which are connected via links. Here we have used for implementing Stack and Queue in our Calculator.

2. Stack

Stack which is conceptual structure consisting of a set of homogeneous elements and is based on the principle of last in first order out (LIFO). In our Calculator Stack it is used to convert Infix to Postfix and also for Postfix operations.

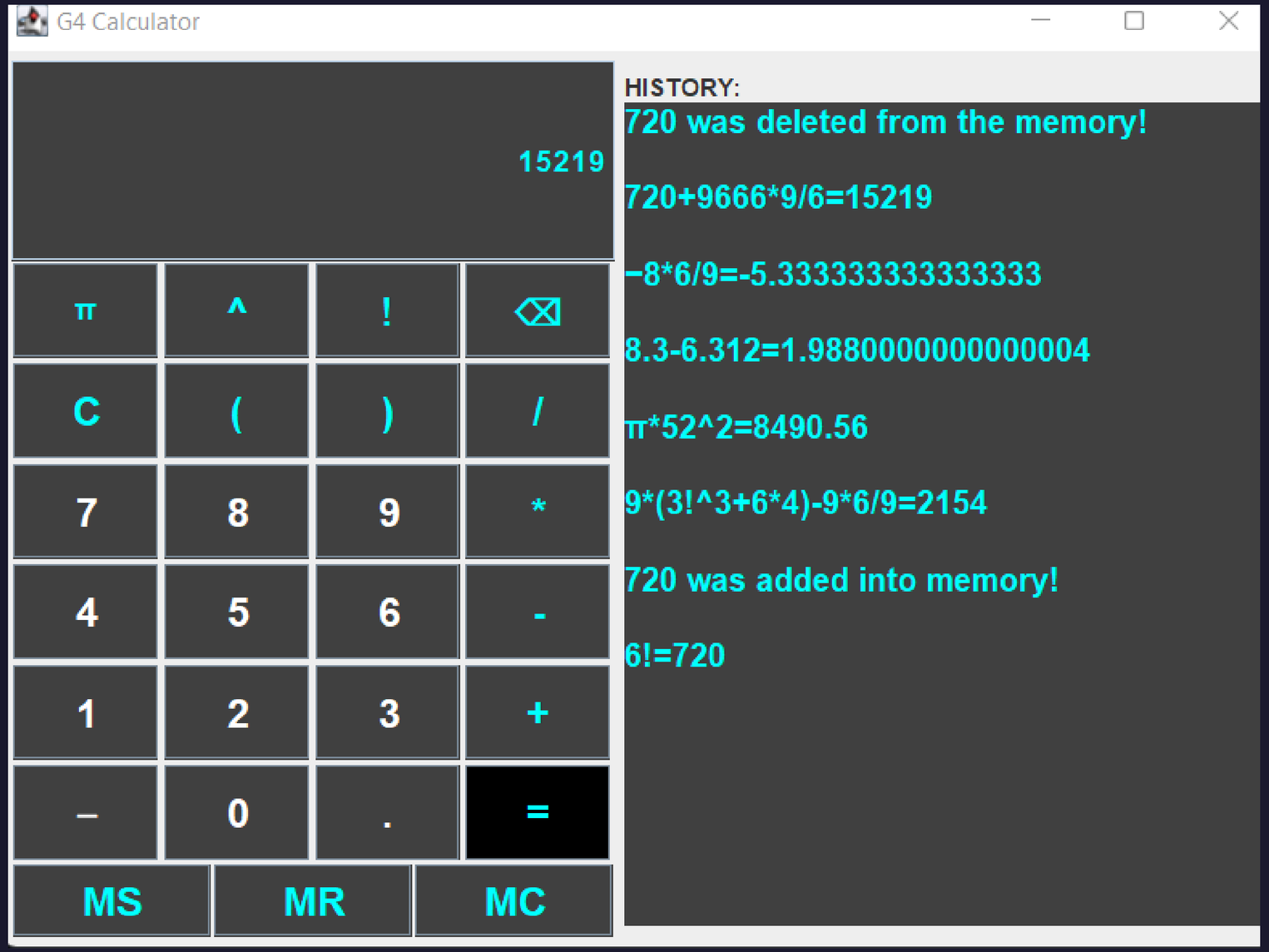
3. Queue

Queue which is a linear structure which follows a particular order in which the operations are performed. In our Calculator we have used Queue to compare the strings.

4. **ArrayList**

ArrayList which is resizable array implementation in java. In our Calculator we have used it to store the values. Infix expression or the input from user in handled using arrays.

Sample Input/Output



Conclusion

We started this project to test our coding skills, but each step was a challenging step that taught us important lessons. This was the first project that we used UI and gave us the opportunity to work with different Data Structures and to use the algorithms we studied this year.

Developing this project made us feel more confident on our skills in programming and algorithms.

THANK YOU