**Evaluation of Postfix Expression:**

package stack1;

import java.util.Stack;

import java.util.Scanner;

public class postfixEvaluation {

public static int evaluatePostfix(String expression) {

Stack<Integer> stack = new Stack<>();

for (char c : expression.toCharArray()) {

if (Character.isDigit(c)) {

// If the character is a digit, push it onto the stack

stack.push(c - '0'); // Convert char to int and push

} else {

// If the character is an operator, pop two operands from the stack,

// apply the operator, and push the result back onto the stack

int operand2 = stack.pop();

int operand1 = stack.pop();

int result = applyOperator(operand1, operand2, c);

stack.push(result);

}

}

// The final result should be on the top of the stack

return stack.pop();

}

private static int applyOperator(int operand1, int operand2, char operator) {

switch (operator) {

case '+':

return operand1 + operand2;

case '-':

return operand1 - operand2;

case '\*':

return operand1 \* operand2;

case '/':

if (operand2 == 0) {

throw new ArithmeticException("Division by zero");

}

return operand1 / operand2;

default:

throw new IllegalArgumentException("Invalid operator: " + operator);

}

}

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter the Postfix Expression");

String postfixExpression =s.nextLine(); // Example postfix expression

int result = evaluatePostfix(postfixExpression);

System.out.println("Result of the postfix expression: " + result);

}

}