**

***LAB # 09***

To implement Stack as Arrays and linked list and Implement Stack by using Stack class.

December 23, 2024

**

***LAB TASKS***

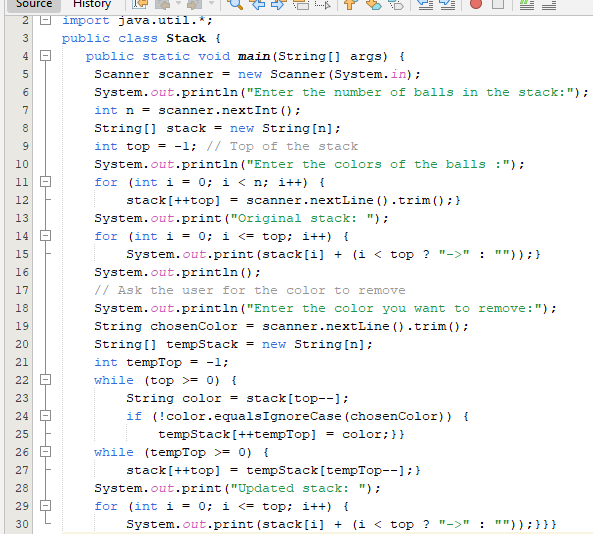
***TASK # 01***

1. Write a program to create Stack (by using array) that can take input random color balls (i.e. red, yellow, orange, green) and store it in order. The user likes any one of the colored ball so he takes out all the balls, one by one, takes only the chosen color and keeps other in order so that he can return them to stack in exactly the same order as before minus the chosen ball. Print both the input and resultant stack. For example:

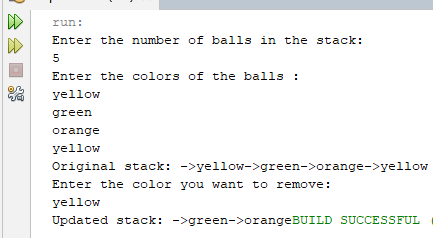
Original stack: Yellow->green->orange->yellow->red

Updated stack: Green->orange->red i.e. yellow is removed

*INPUT*

**

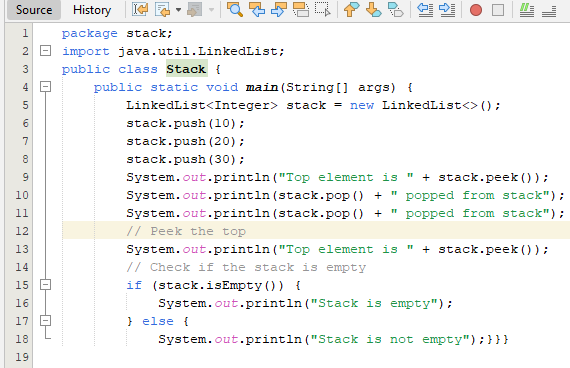
*OUTPUT*

**

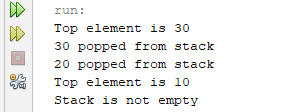
***TASK # 02***

1. Write a program to implement stack (by using linked list) and insert and delete elements in a stack using its operations.

*INPUT*

**

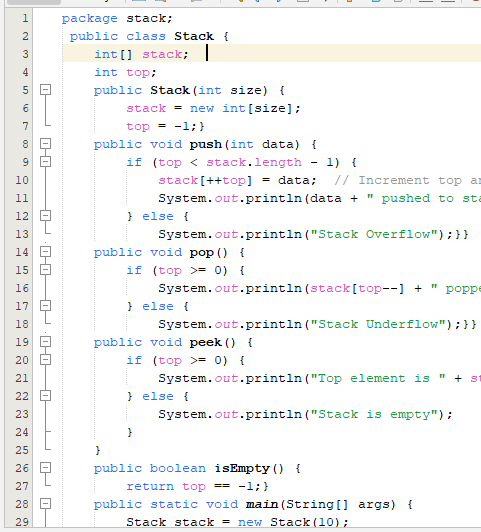
*OUTPUT*

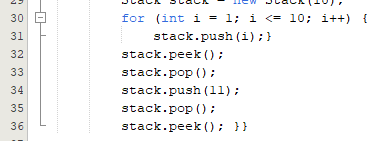
**

***TASK # 03***

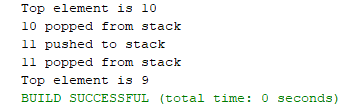
1. Write a program to create stack of 10 elements and perform five more operations on that stack.

*INPUT*

**

**

*OUTPUT*

**

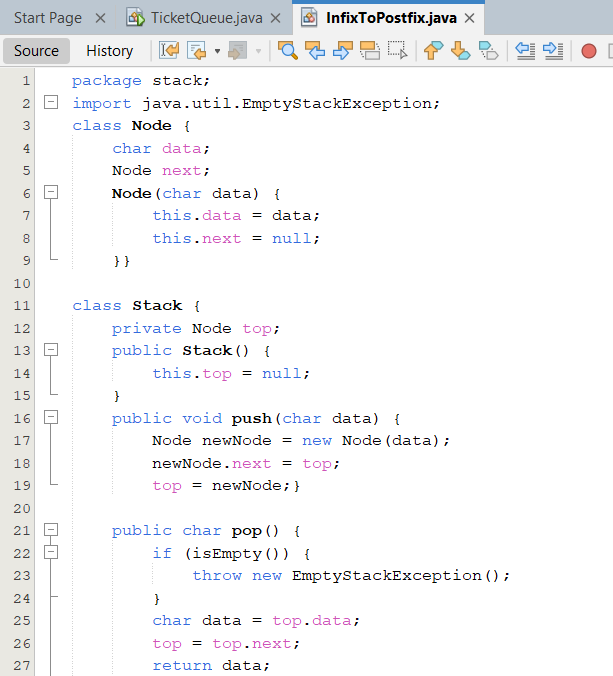
***HOME TASKS***

***TASK # 01***

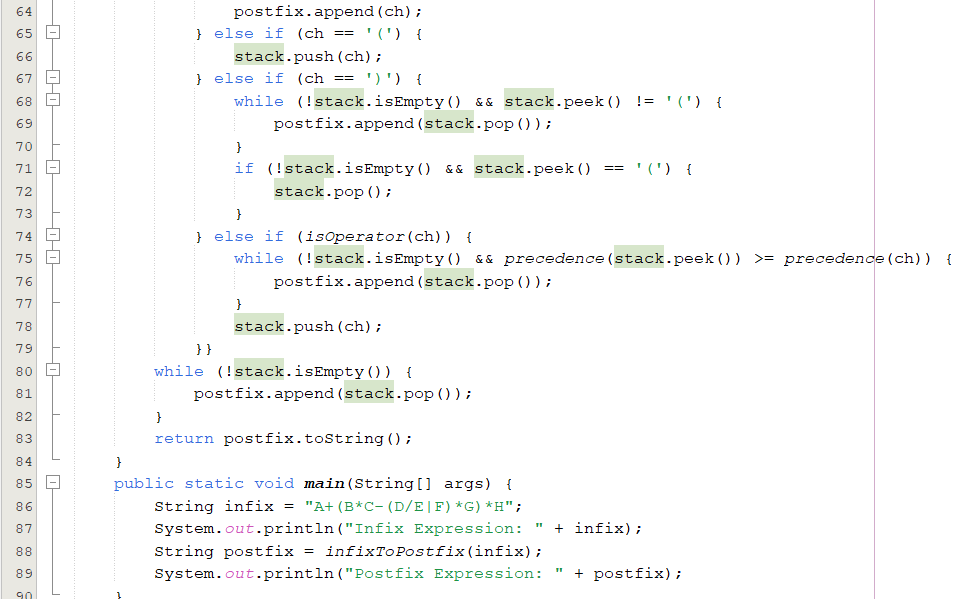
1. Write a program to convert Infix expression into postfix expression by stack using linked list.

**A + ( B \* C - ( D / E | F ) \* G ) \* H**

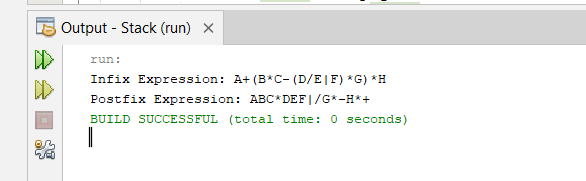
*INPUT*







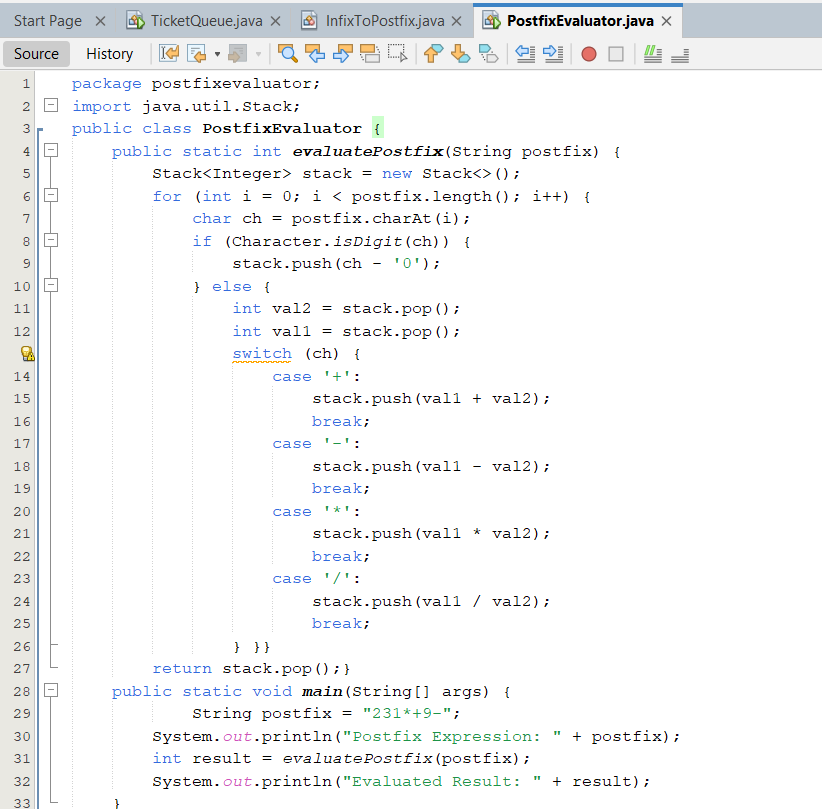
*OUTPUT*



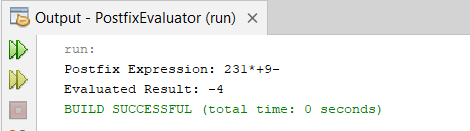
***TASK # 02***

1. Write a program that evaluate following postfix expression by stack using array.

1. **3 1 \* + 9 –**



*OUTPUT*

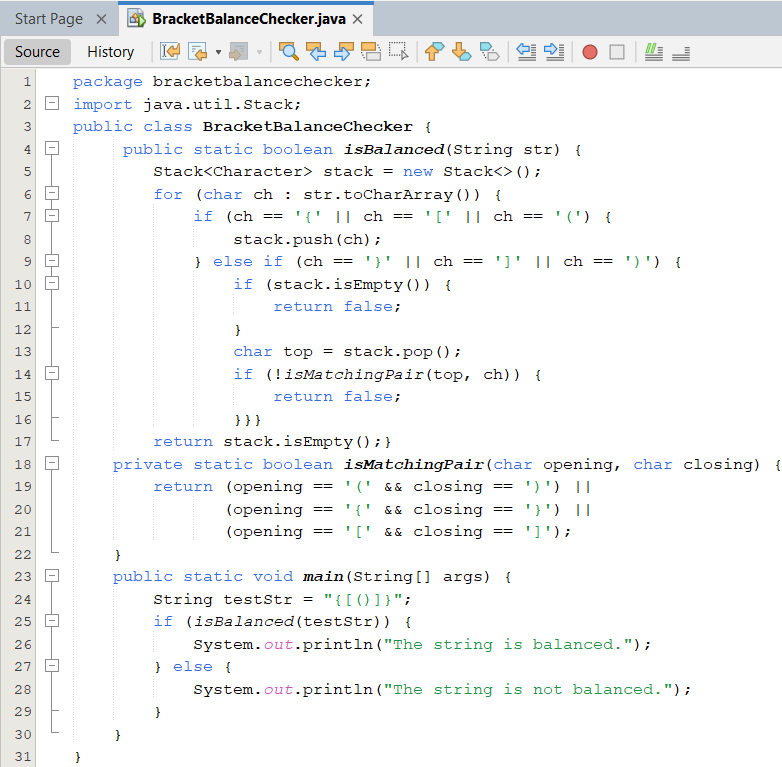


***TASK # 03***

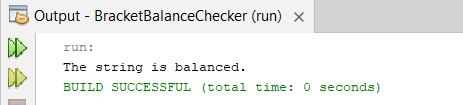
1. You are tasked with writing a program to check if a given string of brackets is balanced. A string is said to be balanced if:
2. Every opening bracket has a corresponding closing bracket.
3. Brackets close in the correct order.

The brackets used are: {, }, [, ], (, ).

*INPUT*

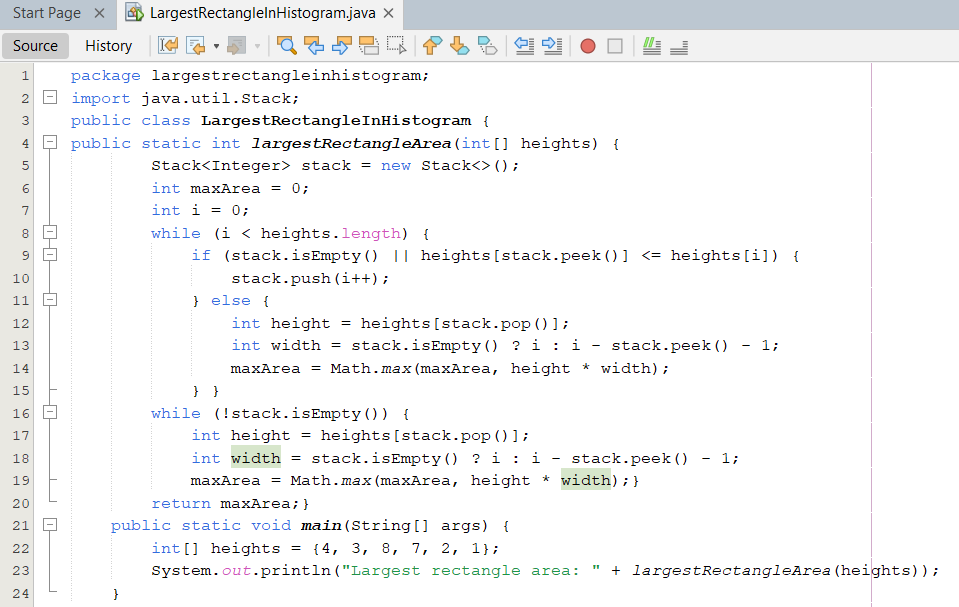


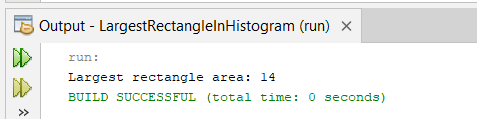
*OUTPUT*



***TASK # 04***

1. Given an array of integers representing the heights of bars in a histogram, find the area of the largest rectangle that can be formed using consecutive bars. Solve this using a **stack**

*OUTPUT*



**Stack ADT Implementation**

***LAB TASKS***

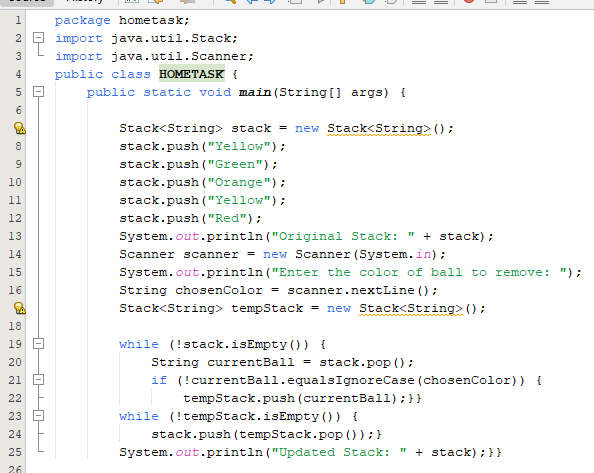
***TASK # 01***

Write a program to create Stack (by using ADT stack) that can take input random color balls (i.e. red, yellow, orange, green) and store it in order. The user likes any one of the colored ball so he takes out all the balls, one by one, takes only the chosen color and keeps other in order so that he can return them to stack in exactly the same order as before minus the chosen ball. Print both the input and resultant stack. For example:

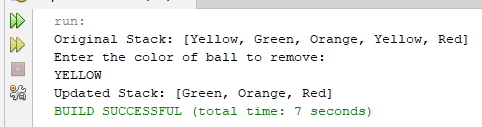
Original stack: Yellow->green->orange->yellow->red

Updated stack: Green->orange->red i.e. yellow is removed

*INPUT*

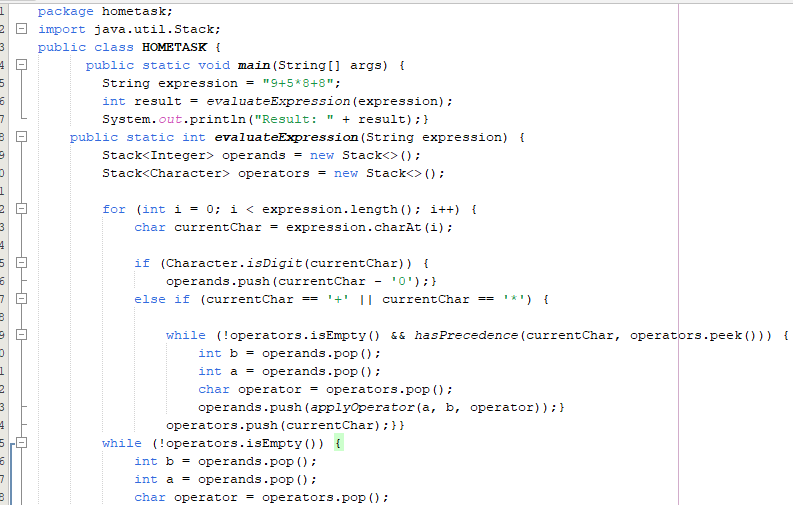
**

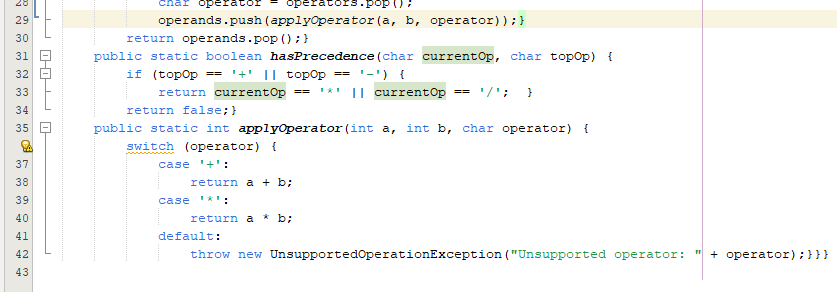
*OUTPUT*

**

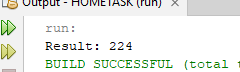
***TASK # 02***

You are required to write a program that evaluates a **simple arithmetic expression** consisting of single-digit integers and the operators + and \*. Use two stacks (one for operands and one for operators) to evaluate the expression.





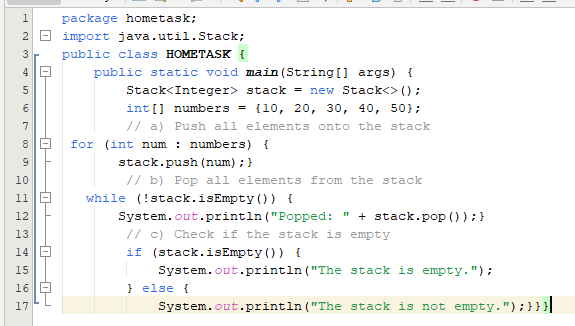
OUTPUT

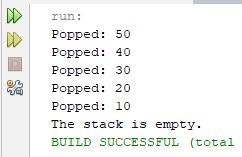


***TASK # 03***

You are given a sequence of integers. Your task is to implement a stack and perform the following operations:

1. **Push** all the elements onto the stack.
2. **Pop** all elements from the stack until it is empty.
3. After performing the pop operations, check if the stack is empty and output whether it is empty or not.

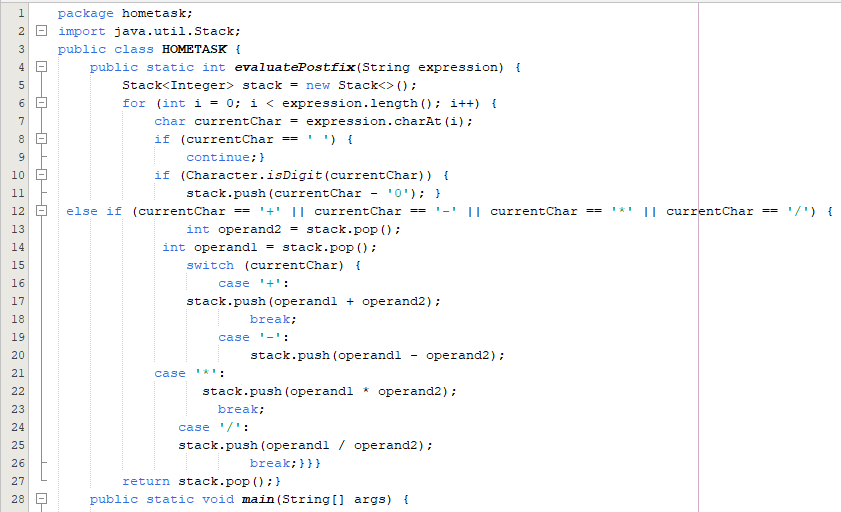
OUTPUT



*HOME TASK*

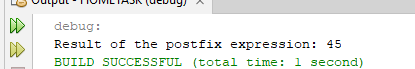
***TASK # 01***

1. Write a program that evaluate following postfix expression by Stack ADT class.

**2 3 1 \* + 9 -** 

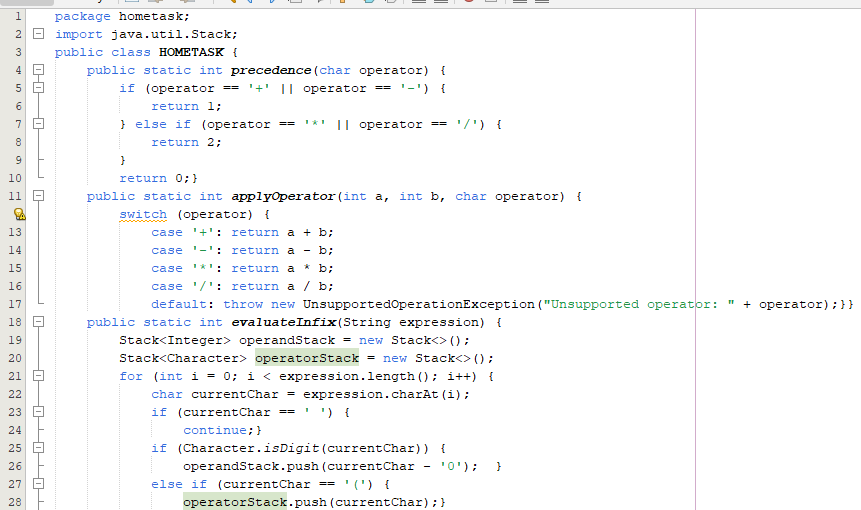


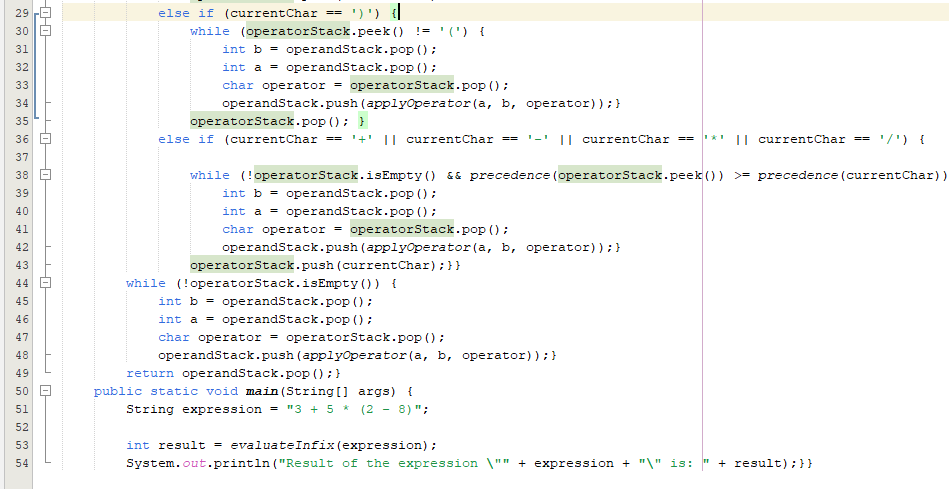
Output



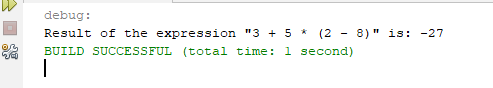
***TASK # 02***

1. You are given an arithmetic expression in **infix notation** (e.g., 3 + 5 \* (2 - 8)). You need to evaluate the expression and return the result. The operators involved are +, -, \*, /, and parentheses ().You must implement a **stack** using the **Stack ADT**



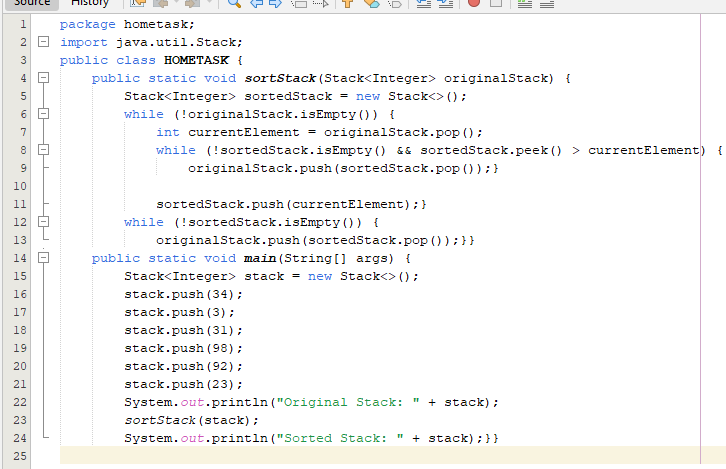


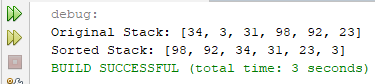
OUTPUT



***TASK # 04***

1. Given a stack of integers, you are to sort the stack in **ascending order** using another stack. You can only push and pop elements from the top of the stack. You are not allowed to use any additional data structures except for the second stack.

******

******