Ex.No: 4	**Java Application for ADT Stack using Interfaces**
Date:	

Aim:

To create a Java console application using interface concepts of java for abstract data type stack. Stack operations must be controlled by exception handling techniques.

Algorithm:

```
Step 1   Start the process.
Step 2   Get the Stack’s maximum limit from user.
Step 3   Create an array with the limit and initialize all the elements as -1 and initialize
         current_position with 0
Step 4   Prompt the user with choice for stack operations
         1. PUSH 2.POP 3.PEEK 4.DISPLAY 5.EXIT
Step 5   Get the choice from user and goto step 6
Step 6   If user selects to PUSH
         Step 6.1   Get the number from user to push.
         Step 6.2   Try to assign the value to array assuming that array index is current position.
         Step 6.3   If exception rises display message “Stack is full try to pop someting” and
                    goto step 9.
         Step 6.4   Else increment current position with the value 1 and display element pushed
                    index and goto step 9.
Step 7   If user selects to POP
         Step 7.1   Try to set previous position of current position to -1
         Step 7.2   If exception rises display message “Stack is empty try to push something”
                    and goto step 9.
         Step 7.3   Else decrement current position with the value 1 and display popped element
                    and goto step 9.
Step 8   If user selects to PEEK display element display current_position -1 element in the array
Step 9   If user selects to DISPLAY
         Step 9.1   Display array reversely and elements which is not equal to -1 and goto step 4
Step 10  Exit from the process.
Step 11  Stop the process.
```

Coding

StackOpeations.java [Interface]

```
package com.raja.oopslab.stackadt;

public interface StackOperations {

    boolean push(int number);

    boolean pop();

    void peek();

    void display();
}
```

CustomStack.java

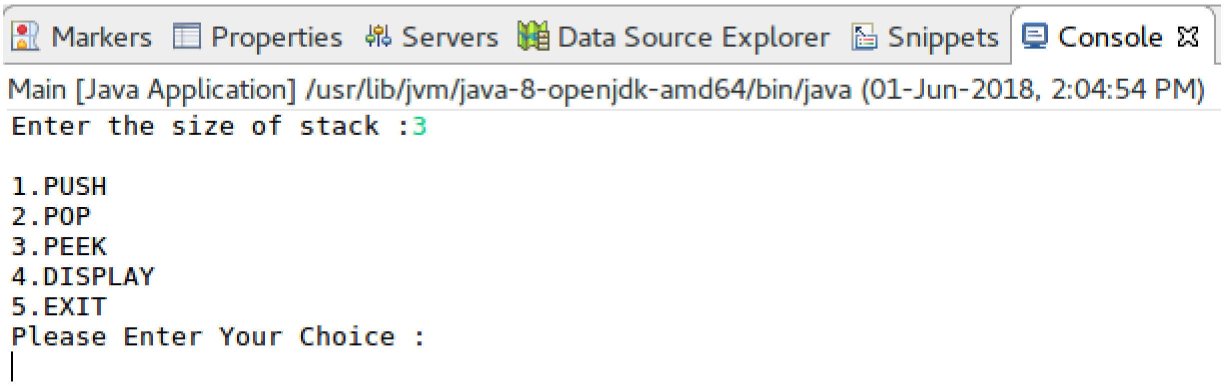
```
package com.raja.oopslab.stackadt;

public class CustomStack implements StackOperations {
    int[] stack_array;
    int limit;
    int current_position = 0;

    public CustomStack(int limit) {
        this.limit = limit;
        stack_array = new int[limit];
        initStack();
    }

    public void initStack() {
        for (int i = 0; i < limit; i++)
            stack_array[i] = -1;
    }

    @Override
    public boolean push(int number) {

        try {

            stack_array[current_position] = number;
            current_position++;
            System.out.println("The element " + number + " pushed in the position " +
current_position);
            display();
            return true;
        }

        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Sorry Stack Full Please do some POP's");
        }
        return false;

    }

    @Override
    public boolean pop() {
        int poped_element;
        try {
            poped_element = stack_array[current_position - 1];
            stack_array[current_position - 1] = -1;
            current_position--;
            System.out.println("Poped element is : " + poped_element);
            display();
            return true;
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Sorry Stack is Empty try to do some push");
        }
        return false;
    }

    @Override
    public void display() {

        System.out.println("\nStack Display");
        System.out.println("*************\n");
        for (int i = limit - 1; i >= 0; i--)
            if (stack_array[i] != -1)
                System.out.println(stack_array[i]);
        System.out.println("\n*************");
    }

    @Override
    public void peek() {
        int peek_element = 0;
        peek_element = stack_array[current_position - 1];
        System.out.println("Peek Element of the Stack is " + peek_element);
    }

}
```

Main.java

```
import java.util.Scanner;
import com.raja.oopslab.stackadt.*;

public class Main {
        public static void main(String[] args) {
                Scanner input = new Scanner(System.in);
                System.out.print("Enter the size of stack :");
                CustomStack mystack = new CustomStack(input.nextInt());
                int choice = 0;
                while (choice != 5) {
                        System.out.println("\n1.PUSH\n2.POP\n3.PEEK\n4.DISPLAY\n5.EXIT");
                        System.out.println("Please Enter Your Choice : ");
                        choice = input.nextInt();
                        switch (choice) {
                        case 1:
                                System.out.println("Enter the Element to PUSH : ");
                                mystack.push(input.nextInt());
                                break;
                        case 2:
                                mystack.pop();
                                break;
                        case 3:
                                mystack.peek();
                                break;
                        case 4:
                                mystack.display();
                                break;
                        case 5:
                                System.out.println("!!! Thank You !!!");
                                break;
                        }

                }
                input.close();
                System.exit(0);
        }

}
```

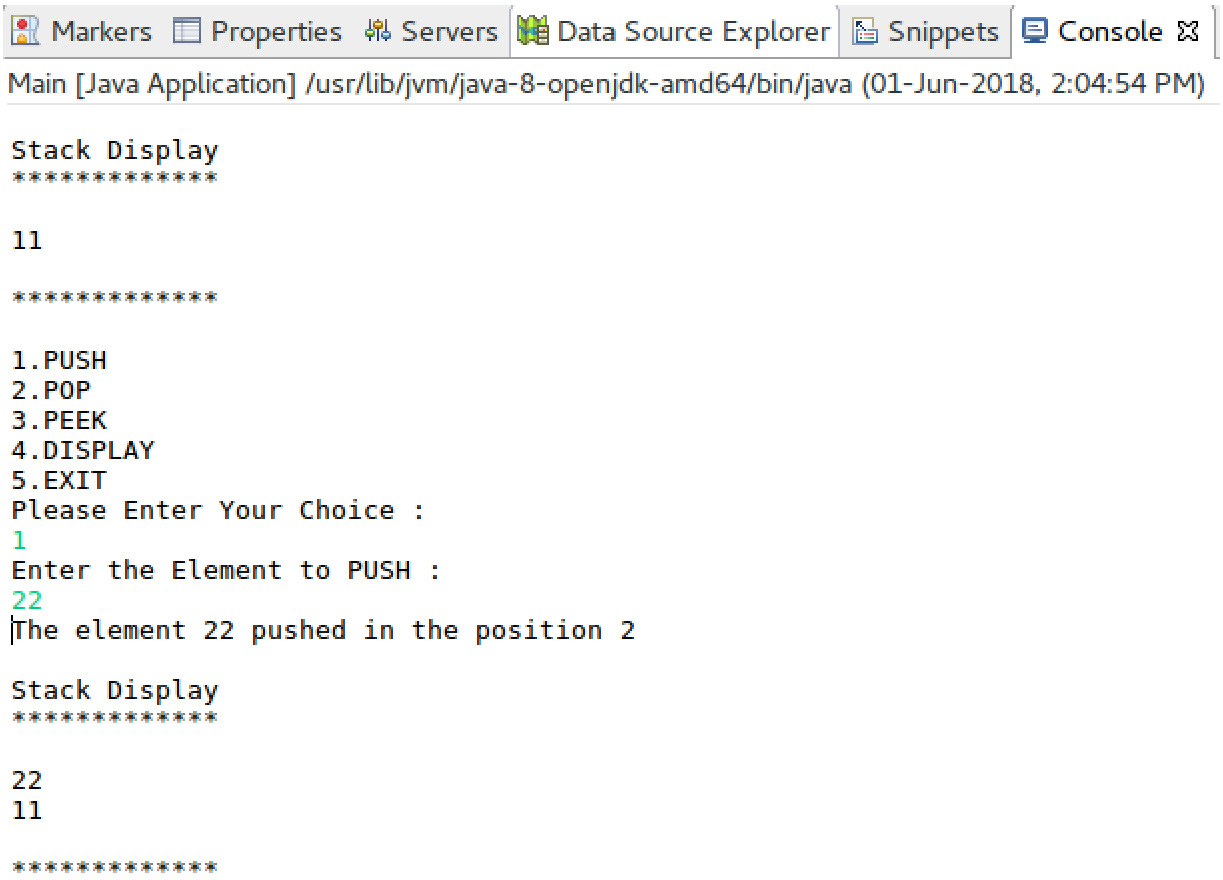
Output:

Choice:



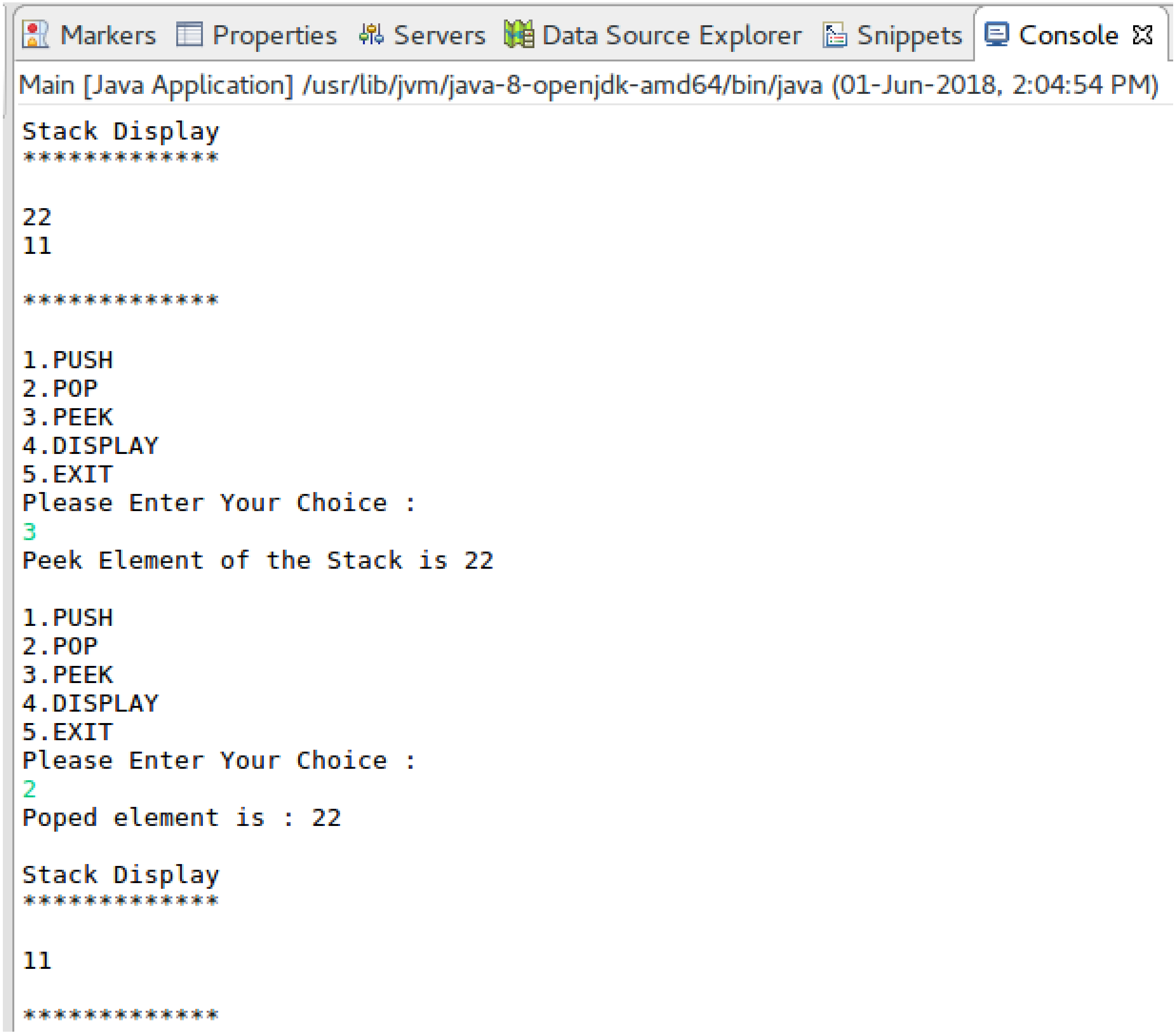
```
Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 2:04:54 PM)
Enter the size of stack :3

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
```

Push Operation:



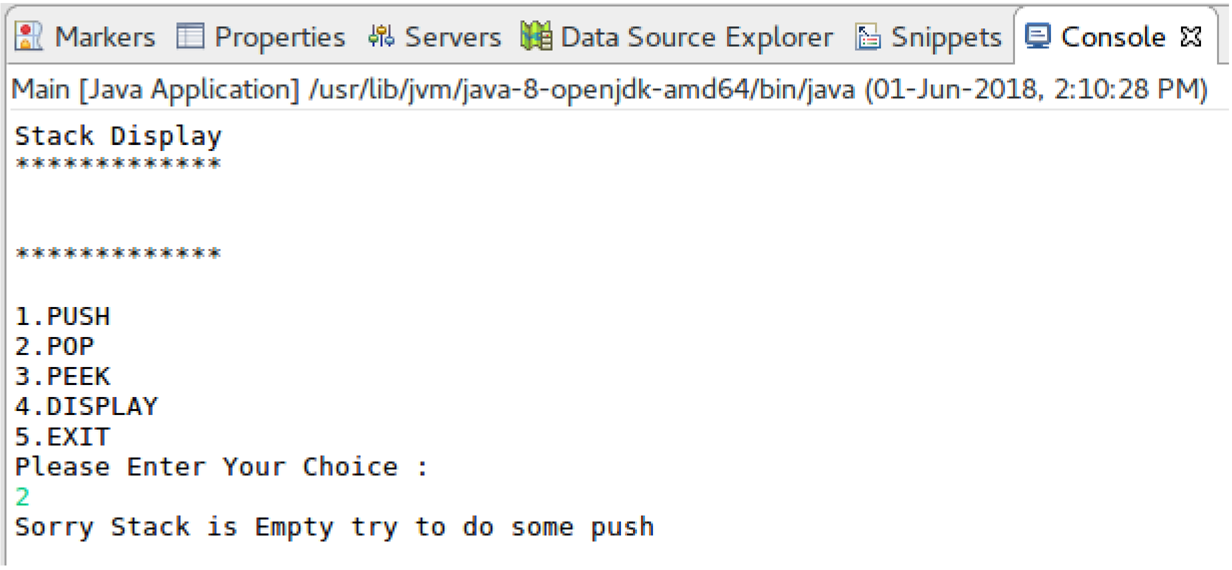
```
Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 2:04:54 PM)

Stack Display
*************

11

*************

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
1
Enter the Element to PUSH :
22
The element 22 pushed in the position 2

Stack Display
*************

22
11

*************
```

Pop and Peek Operation:



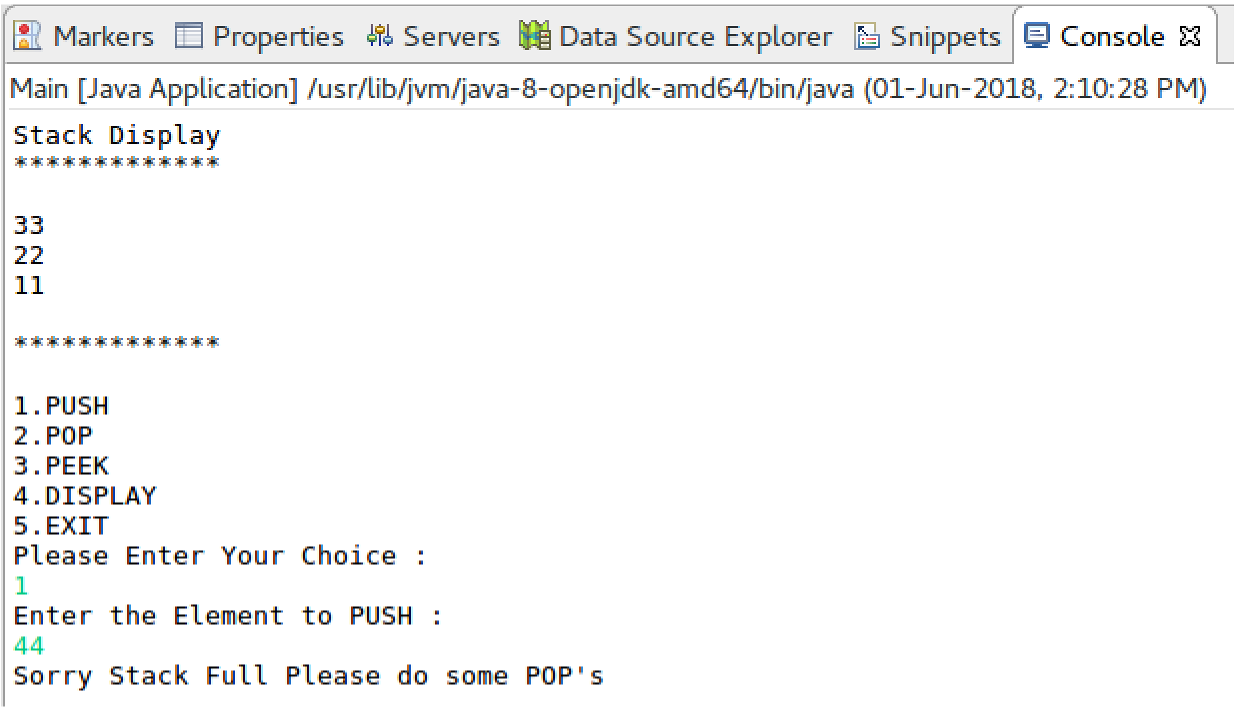
```
Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 2:04:54 PM)
Stack Display
*************

22
11

*************

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
3
Peek Element of the Stack is 22

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
2
Poped element is : 22

Stack Display
*************

11

*************
```

Stack Empty Error:



```
Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 2:10:28 PM)
Stack Display
*************


*************

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
2
Sorry Stack is Empty try to do some push
```

Stack Full Error:



```
Main [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 2:10:28 PM)
Stack Display
*************

33
22
11

*************

1.PUSH
2.POP
3.PEEK
4.DISPLAY
5.EXIT
Please Enter Your Choice :
1
Enter the Element to PUSH :
44
Sorry Stack Full Please do some POP's
```

Result:

The java console application for abstract data type stack using java interface concepts is developed and tested successfully.