Project Assignment

(CSF206)-Advanced Java Programming

Part of the degree of

BACHELOR OF TECHNOLOGY

In

CSE

Submitted to

Mr. Atul Kumar Srivastava

Assistant Professor

SCHOOL OF COMPUTING

Submitted by:

Name: BHAVISH
Roll no: -200102404
Sap Id -1000015397
Section: - D

SCHOOL OF COMPUTING

DIT UNIVERSITY, DEHRADUN

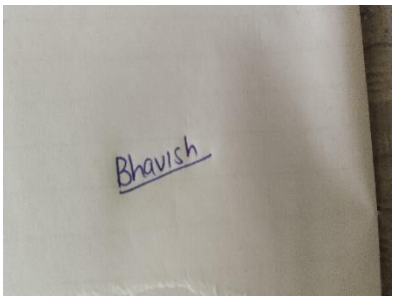
(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

EVEN 2021-22

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the Project Assignment, in partial fulfilment of the requirement as part of the course Advanced Java Programming of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of my work carried out during the period from **11/04/2022** to **24/04/2022** under the guidance of **Mr. Atul Kumar Srivastava**



Date: 24/04/2022

Signature of the Candidate

ACKNOWLEDGEMENT

I thank all my mates and guider who have helped me through this Project .This project cannot be possible without their guidance .I will be cheating myself if I will not take this opportunity to thank all those YouTube teachers and faculties from whom I have learned the basics of this topic.

I sincerely thank my subject teacher **Mr. Atul Kumar Srivastava** Sir for giving me this golden opportunity to perform on this project and for such good guidance throughout my Semester.

Finally ,this project would not have been possible without the confidence ,endurance and support of my family .My family has always been a source of inspiration and encouragement. I wish to thank my parents ,whose love ,teachings and support have brought me this far.

Name: BHAVISH

Roll No: 200102404

SAP ID: 1000015397

TABLE OF CONTENTS:

Problem Statement-

Write a program using JCF (single java file based project with java file name as linlist) to achieve the following:

Write a program that randomly stores 10 integer numbers (to start with) in a LinkedList object. Using this LinkedList object, implement stack and queue operations using GUI as follows: Add 2 Radio Buttons for choosing options (a) Stack and (b) Queue, and 4 JButtons (**Push**, **Pop** for Stack, and **Add, Delete** for Queue).

Once you run the program, it should display a LinkedList of 10 random integers in a text field. You should then be able to choose a radio button of either **Stack** or **Queue**, followed by one of its operations (Push or Pop Buttons for Stack **ONLY**, and Add or Delete Buttons for Queue **ONLY** (Invalid choices like choosing Stack followed by pressing ADD button should not work). The program should then display the modified list (as per the operation undertaken) in the text field.

Note: In case you think you may not be able to implement the GUI, **then use JOptionPane** to enter your commands (as shown below), and show the resultant Linked List in the console. The commands to be entered in showInputDialogue will be as follows:

Push

Pop

Add

Delete

Note that Stack is a LIFO (Last in first out) structure, and Queue is a FIFO (First in first out) structure.

MODULES AND CODE:

```
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;



public class GUI_Application {



public static void main(String[] args) {
new GUI2();
}
}



class GUI2 extends JFrame implements ActionListener{

JFrame frame= new JFrame();
//Panel panel= new Panel();

LinkedList<Integer> list = new LinkedList<Integer>(); //Used to show Original
List
LinkedList<Integer> list1 = new LinkedList<Integer>();//Used to shoe Updated
List

JLabel label;
JLabel label1;

JTextArea text;
JTextArea text1;


JRadioButton r1= new JRadioButton("STACK");
JRadioButton r2=new JRadioButton("QUEUE");



Button b1= new Button("Push");
Button b2= new Button("Pop");
Button b3= new Button("Add");
Button b4= new Button("Delete");
```

```
GUI2(){


Random random = new Random();

for(int i=0;i<10;i++) {
list.add(random.nextInt(100));
}

list1=list;

label = new JLabel("Original Linked List elements are: ");
label1 = new JLabel("Updated Linked List elements are: ");

label.setBounds(10, 10, 100, 100);
label1.setBounds(100, 30, 100, 100);

String Ostr = new String(); //Used to pass list elements as string in
JTextArea Area field
String Ustr1 = new String(); //Used to pass list1 elements as string in
JTextArea Area field

//for loop used to convert each list elements to string
for(int i=0;i<list.size();i++) {
Ostr = Ostr+list.get(i).toString()+"\t";
}

//for loop used to convert each list1 elements to string
for(int i=0;i<list1.size();i++) {
Ustr1 = Ustr1+list.get(i).toString()+"\t";
}

text = new JTextArea(Ostr);
text1 = new JTextArea(Ustr1);

frame.setSize(10000,4000);
frame.setLayout(null);
//frame.add(label);
//frame.add(text);
//frame.add(panel);

label.setBounds(50,50,330,40);
frame.add(label);
label1.setBounds(50,600,330,40);
text.setBounds(300,50,1000,40);
frame.add(text);
text1.setBounds(300,600,1000,40);

// adding radio button r1
```

```
r1.setBounds(500, 250, 330, 40);
r1.setForeground(Color.blue);
//frame.setSize(400,450);

frame.add(r1);

// adding radio button r2
r2.setBounds(1000, 250, 330, 30);
r2.setForeground(Color.blue);
//frame.setSize(400,450);

frame.add(r2);

//adding both radio button in one group...so that only one radio button is
selected at a time
ButtonGroup G = new ButtonGroup();
G.add(r1);
G.add(r2);

// adding button b1
b1.addActionListener(this);
b1.setBackground(Color.green);
b1.setBounds(260, 350, 150, 60);
frame.add(b1);

//frame.setSize(400,450);
b1.setForeground(Color.BLACK);


// adding button b2
b2.addActionListener(this);
b2.setBackground(Color.red);
b2.setBounds(560, 350, 150, 60);
frame.add(b2);

//frame.setSize(400,450);
b2.setForeground(Color.BLACK);


// adding button b3
b3.addActionListener(this);
b3.setBackground(Color.green);
b3.setBounds(860, 350, 150, 60);
frame.add(b3);

//frame.setSize(400,450);
b3.setForeground(Color.BLACK);
```

```
// adding button b4
b4.addActionListener(this);
b4.setBackground(Color.red);
b4.setBounds(1160, 350, 150, 60);
frame.add(b4);

//frame.setSize(400,450);
b4.setForeground(Color.BLACK);

frame.add(label1);
frame.add(text1);

frame.setBounds(1200, 100, 300, 200);
frame.setSize(300, 200);
frame.setLayout(null);

frame.setTitle("BHAVISH ,1000015397");
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public void actionPerformed(ActionEvent e){
try {
if((r1.isSelected()) && (e.getSource()==b1))
{

String num=JOptionPane.showInputDialog(frame,"Enter Value for stack: ");

list1.addLast(Integer.parseInt(num));


String str = new String();
for(int i=0;i<list1.size();i++) {
str = str+list1.get(i).toString()+"\t";
}

text1.setText(null); //used to clear text area
text1.append(str); //used to insert new string in text area

}
else if((r1.isSelected()) && (e.getSource()==b2))
{

list1.removeLast();

String str = new String();
for(int i=0;i<list1.size();i++) {
str = str+list1.get(i).toString()+"\t";
}
```

```
text1.setText(null);
text1.append(str);

}



else if((r2.isSelected()) && (e.getSource()==b3))
{
String num=JOptionPane.showInputDialog(frame,"Enter Value for queue: ");

list1.addLast(Integer.parseInt(num));

String str = new String();
for(int i=0;i<list1.size();i++) {
str = str+list1.get(i).toString()+"\t";
}

text1.setText(null);
text1.append(str);

}

else if((r2.isSelected()) && (e.getSource()==b4))
{
list1.removeFirst();

String str = new String();
for(int i=0;i<list1.size();i++) {
str = str+list1.get(i).toString()+"\t";
}

text1.setText(null);
text1.append(str);
}

else if((r1.isSelected()) && (e.getSource()==b3)) {
throw new Exception();
}
else if((r1.isSelected()) && (e.getSource()==b4)) {
throw new Exception();
}
else if((r2.isSelected()) && (e.getSource()==b1)) {
throw new Exception();
}
else if((r2.isSelected()) && (e.getSource()==b2)) {
throw new Exception();
}
```

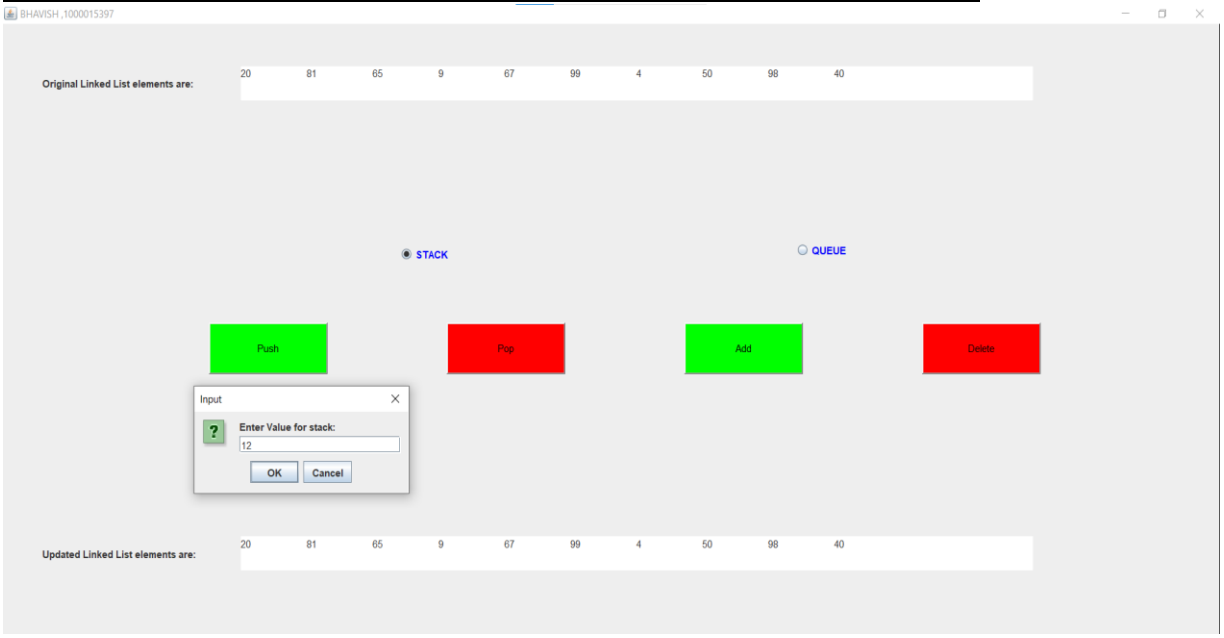
```
}catch(Exception Ex) {

if((r1.isSelected()) && (e.getSource()==b3)) {
JOptionPane.showMessageDialog(frame,"Invalid
Operation!","Alert",JOptionPane.WARNING_MESSAGE);
frame.setDefaultCloseOperation(3);
}
else if((r1.isSelected()) && (e.getSource()==b4)) {
JOptionPane.showMessageDialog(frame,"Invalid
Operation!","Alert",JOptionPane.WARNING_MESSAGE);
frame.setDefaultCloseOperation(3);
}
else if((r2.isSelected()) && (e.getSource()==b1)) {
JOptionPane.showMessageDialog(frame,"Invalid
Operation!","Alert",JOptionPane.WARNING_MESSAGE);
frame.setDefaultCloseOperation(3);
}
else if((r2.isSelected()) && (e.getSource()==b2)) {
JOptionPane.showMessageDialog(frame,"Invalid
Operation!","Alert",JOptionPane.WARNING_MESSAGE);
frame.setDefaultCloseOperation(3);
}
}
}
}
```

OUTPUTS :

1. **The first interface of the output.**

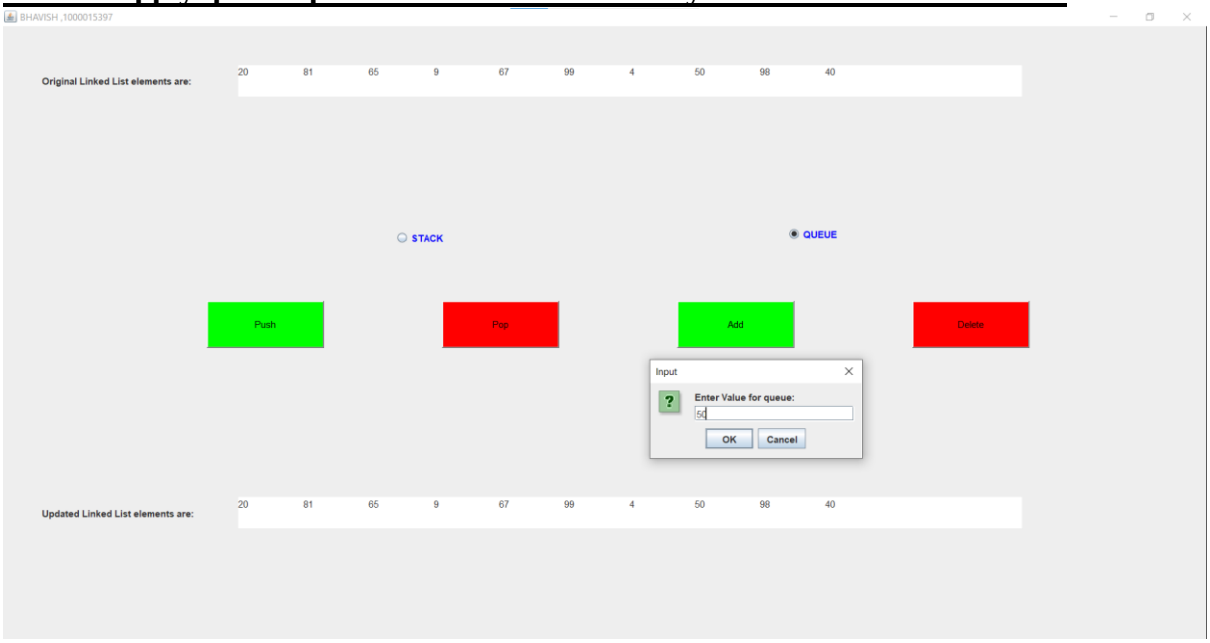
2. **<u>When we perform stack operation and try to push element in linked list</u>**



3. **<u>After push 12 in the linked list at last of list we get update linked list</u>**

4. **<u>Now perform pop operation on linked list and we get 12 remove from last of the list.</u>**

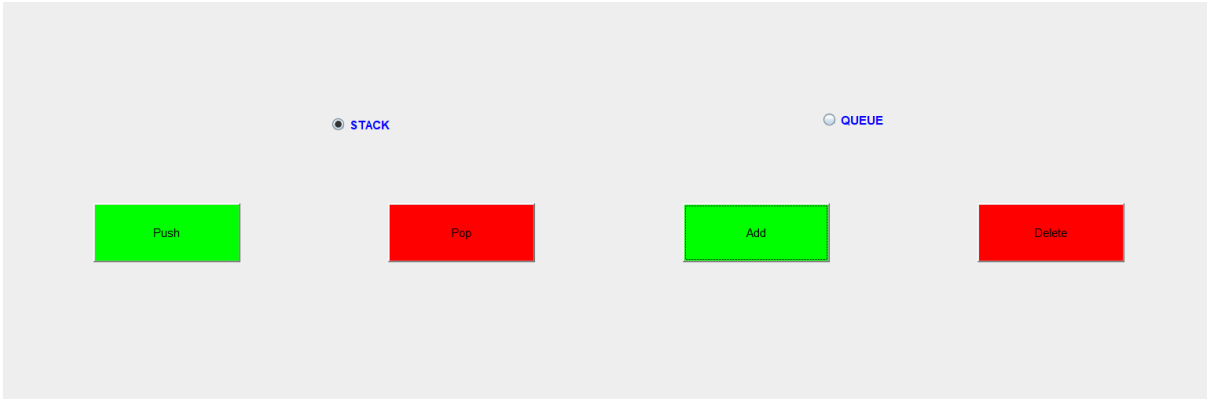
5. **<u>Now we apply queue operation on linked list we try to add new element in list.</u>**



6. **<u>After adding 50 in list at last we get update list.</u>**

7. **<u>Now we perform delete on list and the 20 is deleted from list because in queue it use FIFO .</u>**

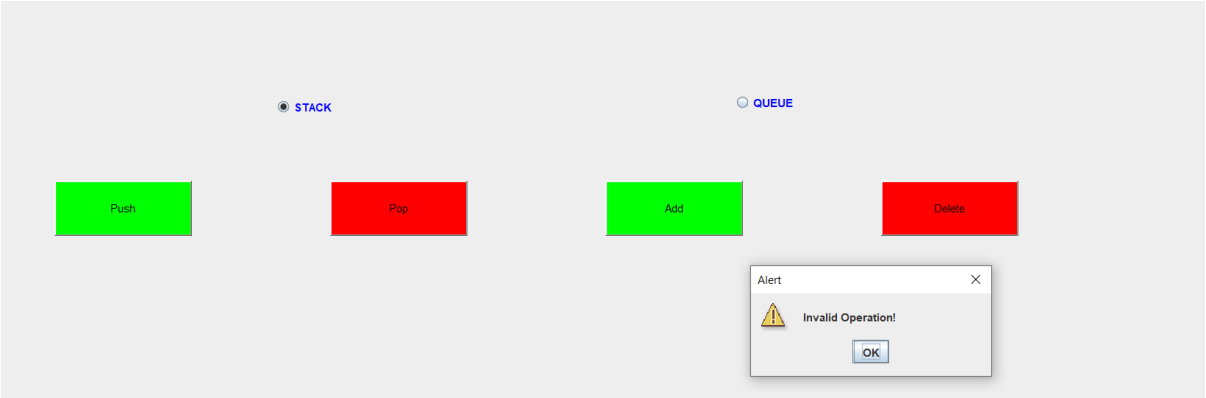
8. **<u>Now we select the stack and try to add the element in list.</u>**



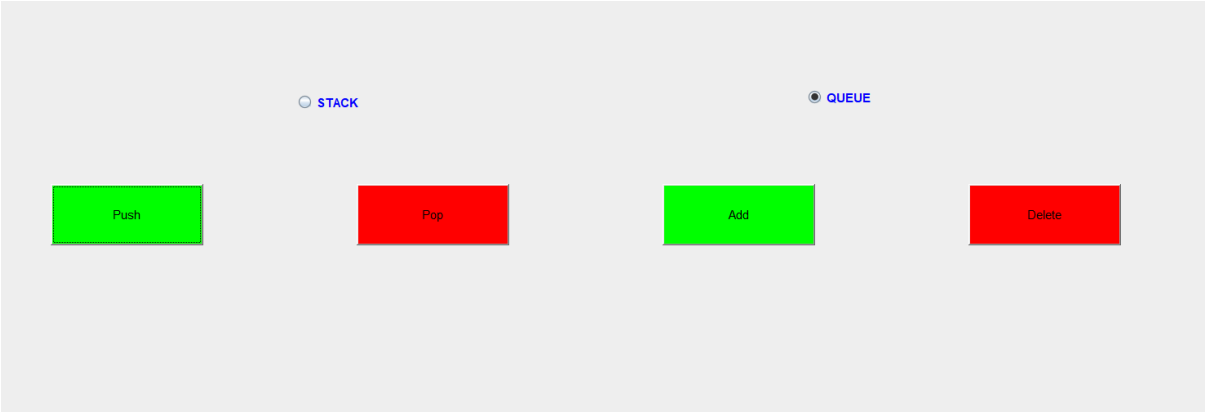
9. **<u>We get invalid operation .</u>**

10. **<u>Now we select stack and try to delete element from list.</u>**

11. **<u>We get invalid operation.</u>**



12. **<u>Now we select queue and try to push the element in list.</u>**



13. **<u>We get invalid operation.</u>**

14. **<u>Now we select queue and try to implement pop on the list .</u>**

15. **<u>We get invalid operation.</u>**

BIBLIOGRAPHY

INTRODUCTION TO JAVA PROGRAMMING AND DATA STRUCTURE

BY Y DANIEL LIANG,

SWING TUTORIAL ON YOUTUBE BY BROCODE CHANNEL

Submitted By: BHAVISH

Sap id: 1000015397

Roll no: 200102404

Signature: