# Project Assignment

# (CSF206)-Advanced Java Programming

Part of the degree of

**BACHELOR OF TECHNOLOGY**

**In**

**CSE**

**Submitted to**

Dr. Anurag Srivastava

Assistant Professor

SCHOOL OF COMPUTING

**Submitted by:**

Ashish Bisht, Sap ID-1000016307

Tushar Garg, Sap ID-1000016480

Manas Goel, Sap ID-1000016211

Abhay Twagi, Sap ID-1000015738

## 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.**

# 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 **11/04/2023** to **24/04/2023** under the guidance of **Dr. Anurag Srivastava.**

# 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.

The commands to be entered in showInputDialogue will be as follows:

Push

Pop

Add

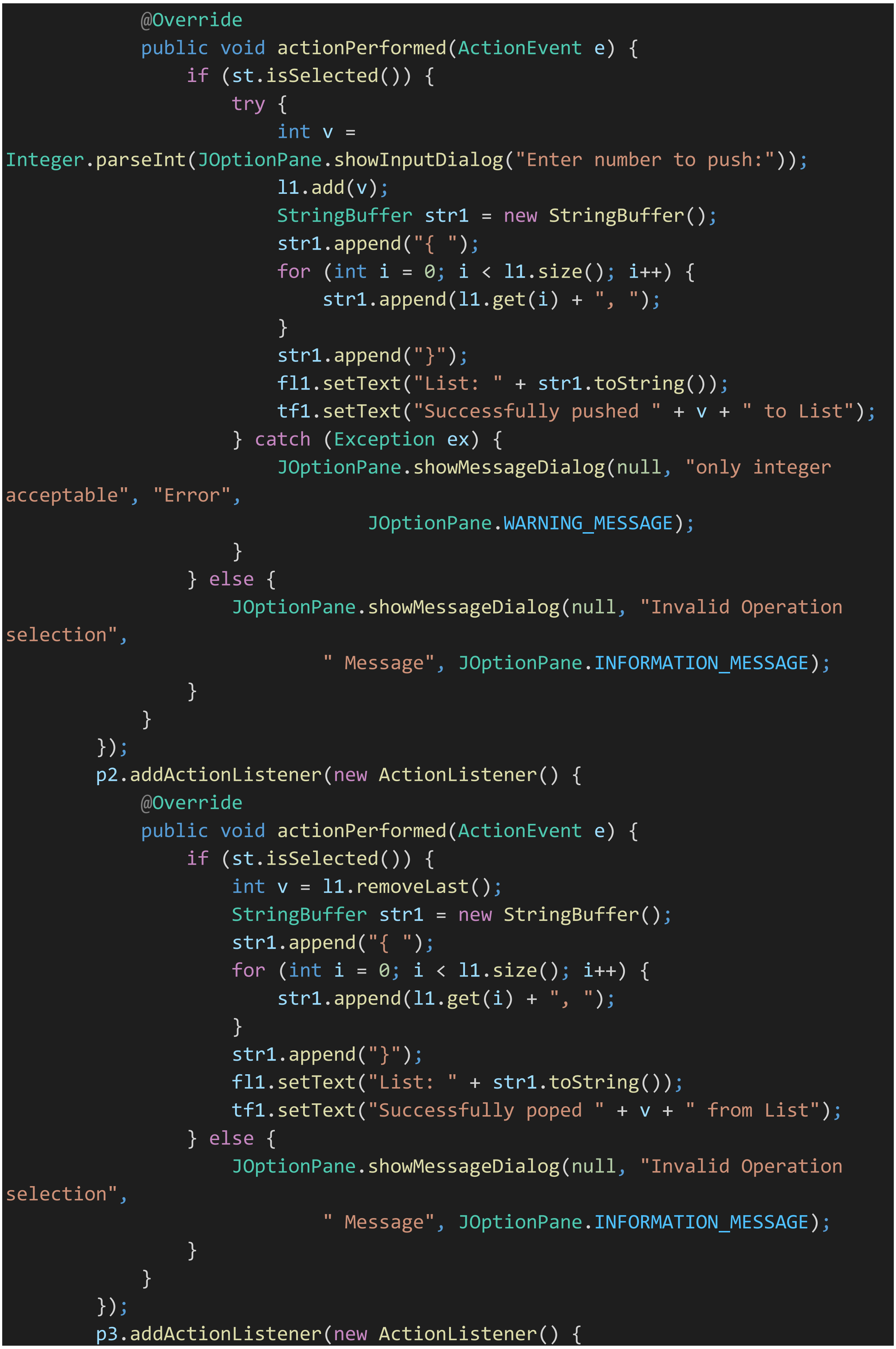
Delete

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

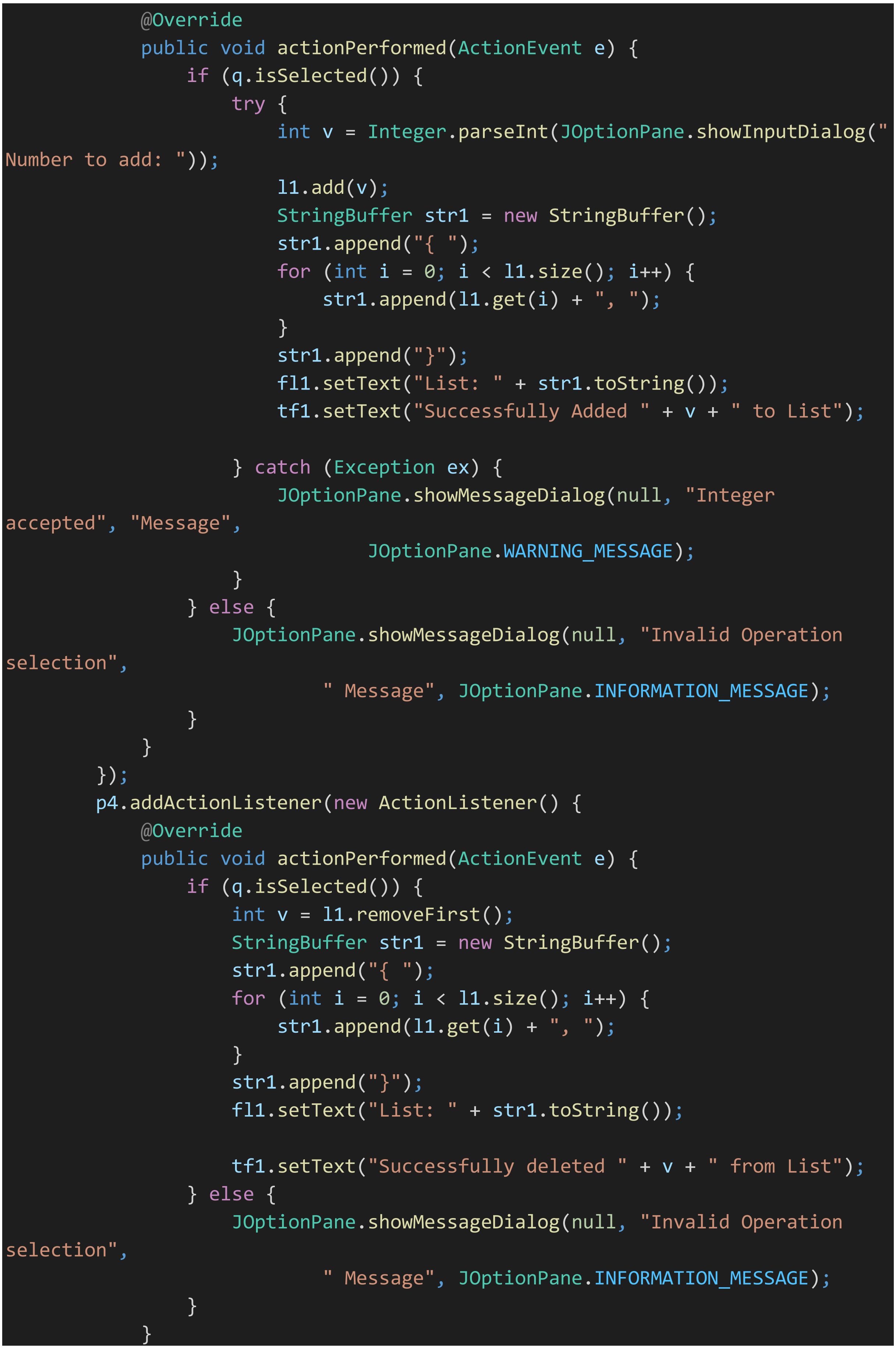
Also, above your code, **mention in 2-3 lines of comments**, what exactly you have achieved, in case you could not complete the desired output, and **whether you used GUI or chose to use JOptionPane.**

## Source Code:

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

public class proch {
    public static void main(String[] args) {
        LinkedList<Integer> l1 = new LinkedList<Integer>();
        for (int i = 0; i < 20; i++) {
            int n = (int) (Math.random() * 10 + 1);
            l1.add(n);
        }
        JFrame f1 = new JFrame();
        f1.setTitle(" Project Assignment");
        f1.setLayout(new FlowLayout(FlowLayout.CENTER));
        JTextField fl1 = new JTextField();
        fl1.setPreferredSize(new Dimension(550, 100));
        fl1.setHorizontalAlignment(JTextField.CENTER);
        StringBuffer str = new StringBuffer();
        str.append("{ ");
        for (int i = 0; i < l1.size(); i++) {
            str.append(l1.get(i) + ", ");
        }
        str.append("}");
        fl1.setText("List: " + str.toString());
        JRadioButton st = new JRadioButton("Stack");
        st.setPreferredSize(new Dimension(100, 100));
        JRadioButton q = new JRadioButton("Queue");
        q.setPreferredSize(new Dimension(100, 100));
        ButtonGroup gr = new ButtonGroup();
        gr.add(st);
        gr.add(q);
        JButton ps = new JButton("push");
        JButton p2 = new JButton("pop");
        JButton p3 = new JButton("Add");
        JButton p4 = new JButton("Delete");
        ps.setPreferredSize(new Dimension(100, 50));
        p2.setPreferredSize(new Dimension(100, 50));
        p3.setPreferredSize(new Dimension(100, 50));
        p4.setPreferredSize(new Dimension(100, 50));
        JTextField tf1 = new JTextField();
        tf1.setPreferredSize(new Dimension(400, 50));
        tf1.setHorizontalAlignment(JTextField.CENTER);
        ps.addActionListener(new ActionListener() {
```

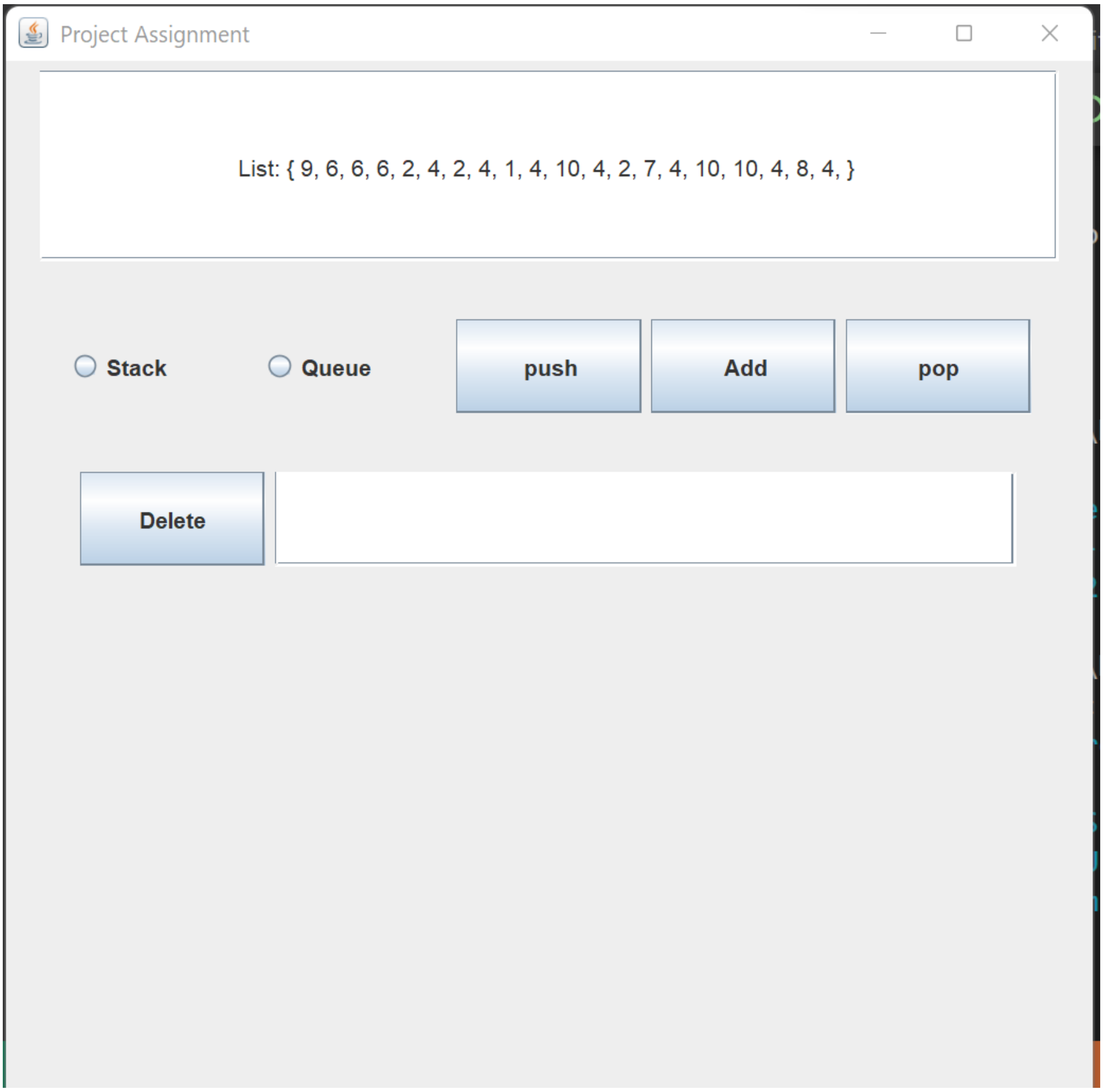


```
            @Override
            public void actionPerformed(ActionEvent e) {
                if (st.isSelected()) {
                    try {
                        int v =
Integer.parseInt(JOptionPane.showInputDialog("Enter number to push:"));
                        l1.add(v);
                        StringBuffer str1 = new StringBuffer();
                        str1.append("{ ");
                        for (int i = 0; i < l1.size(); i++) {
                            str1.append(l1.get(i) + ", ");
                        }
                        str1.append("}");
                        fl1.setText("List: " + str1.toString());
                        tf1.setText("Successfully pushed " + v + " to List");
                    } catch (Exception ex) {
                        JOptionPane.showMessageDialog(null, "only integer
acceptable", "Error",
                                JOptionPane.WARNING_MESSAGE);
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Operation
selection",
                            " Message", JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });
        p2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (st.isSelected()) {
                    int v = l1.removeLast();
                    StringBuffer str1 = new StringBuffer();
                    str1.append("{ ");
                    for (int i = 0; i < l1.size(); i++) {
                        str1.append(l1.get(i) + ", ");
                    }
                    str1.append("}");
                    fl1.setText("List: " + str1.toString());
                    tf1.setText("Successfully poped " + v + " from List");
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Operation
selection",
                            " Message", JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });
        p3.addActionListener(new ActionListener() {
```

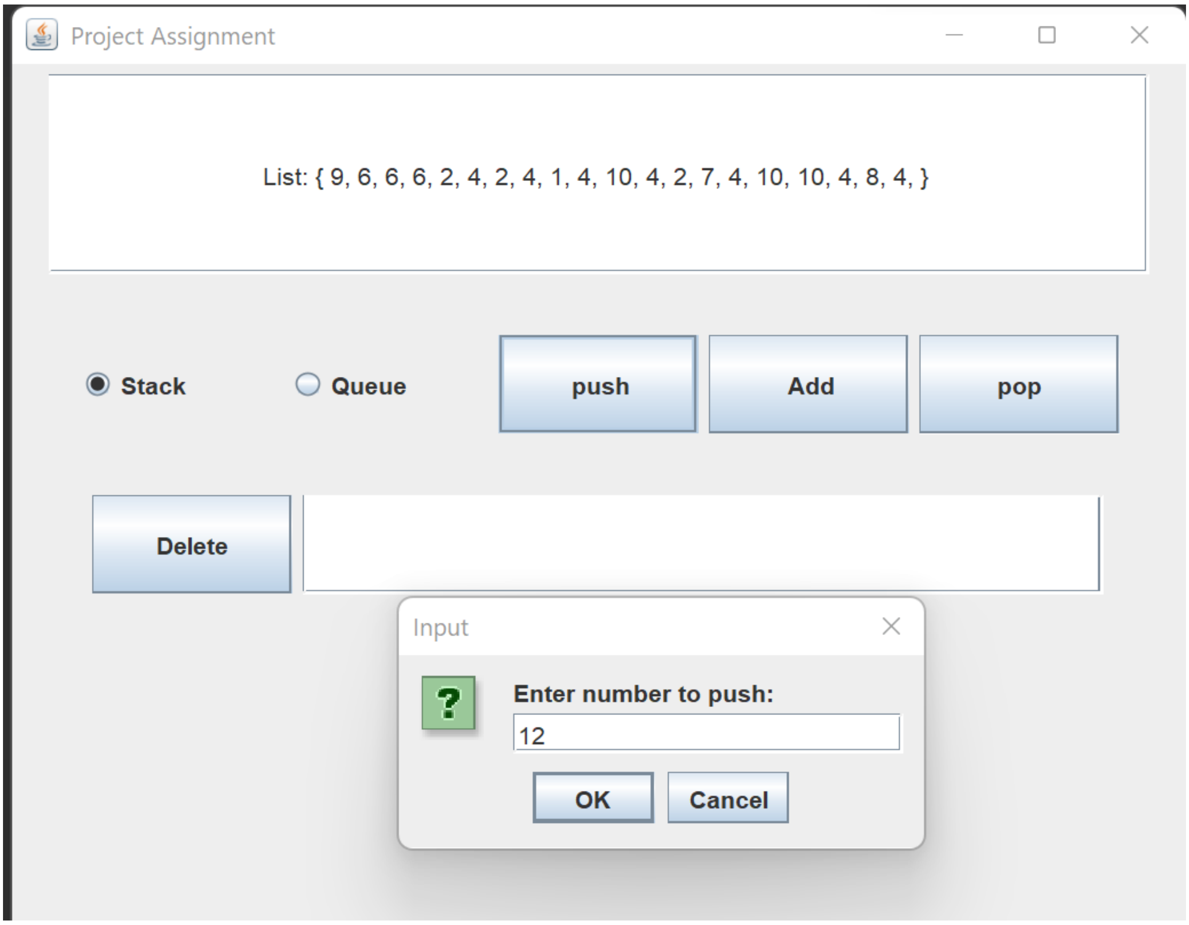


```
            @Override
            public void actionPerformed(ActionEvent e) {
                if (q.isSelected()) {
                    try {
                        int v = Integer.parseInt(JOptionPane.showInputDialog("
Number to add: "));
                        l1.add(v);
                        StringBuffer str1 = new StringBuffer();
                        str1.append("{ ");
                        for (int i = 0; i < l1.size(); i++) {
                            str1.append(l1.get(i) + ", ");
                        }
                        str1.append("}");
                        fl1.setText("List: " + str1.toString());
                        tf1.setText("Successfully Added " + v + " to List");

                    } catch (Exception ex) {
                        JOptionPane.showMessageDialog(null, "Integer
accepted", "Message",
                                JOptionPane.WARNING_MESSAGE);
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Operation
selection",
                            " Message", JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });
        p4.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (q.isSelected()) {
                    int v = l1.removeFirst();
                    StringBuffer str1 = new StringBuffer();
                    str1.append("{ ");
                    for (int i = 0; i < l1.size(); i++) {
                        str1.append(l1.get(i) + ", ");
                    }
                    str1.append("}");
                    fl1.setText("List: " + str1.toString());

                    tf1.setText("Successfully deleted " + v + " from List");
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Operation
selection",
                            " Message", JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });
        f1.add(fl1);
        f1.add(st);
        f1.add(q);
        f1.add(ps);
        f1.add(p3);
        f1.add(p2);
        f1.add(p4);
        f1.add(tf1);
        f1.setSize(600, 600);
        f1.setResizable(true);
        f1.setVisible(true);
    }
}
```

## Outputs:



```
List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, }
```

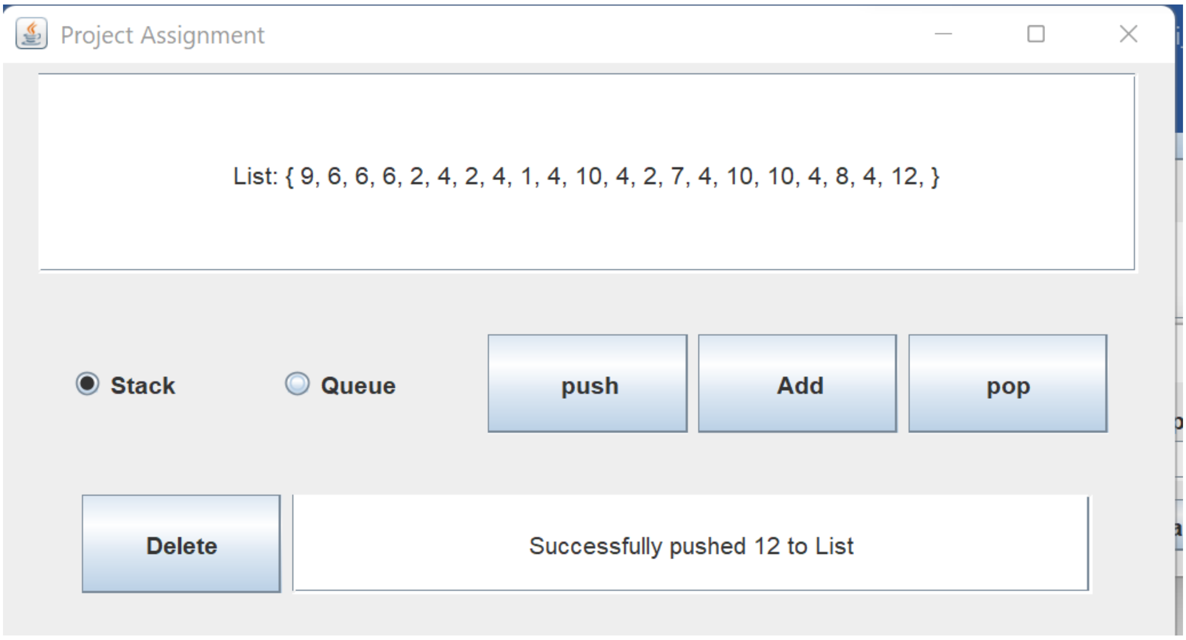


List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, }

● Stack ○ Queue

Enter number to push:

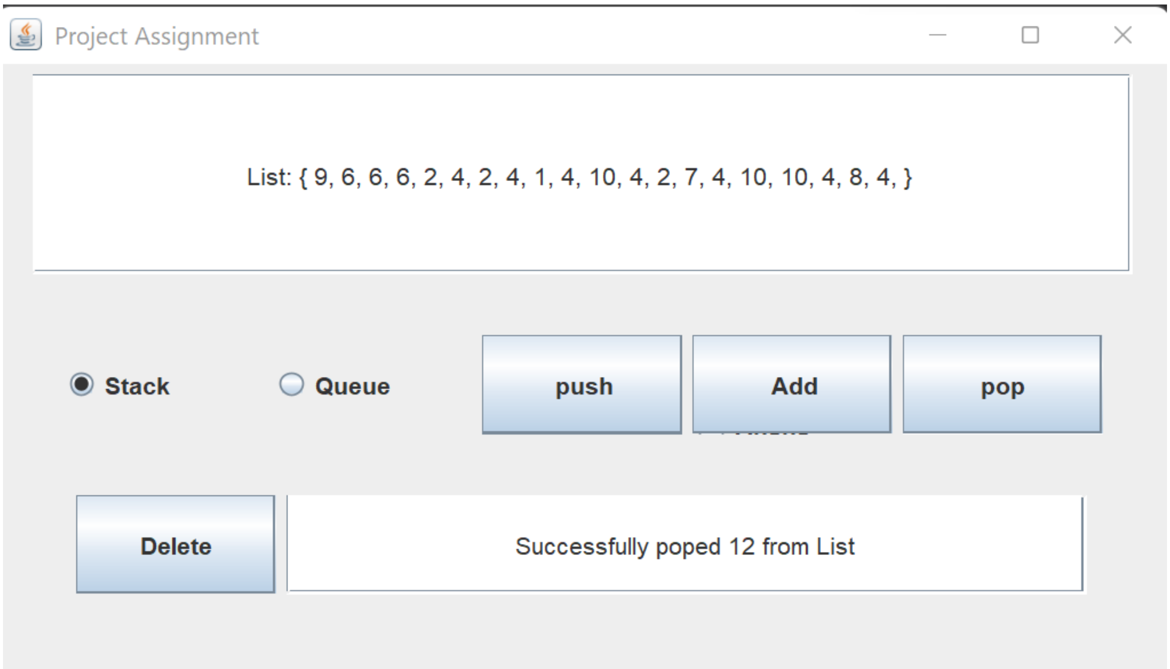
12



List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, 12, }

● Stack ○ Queue

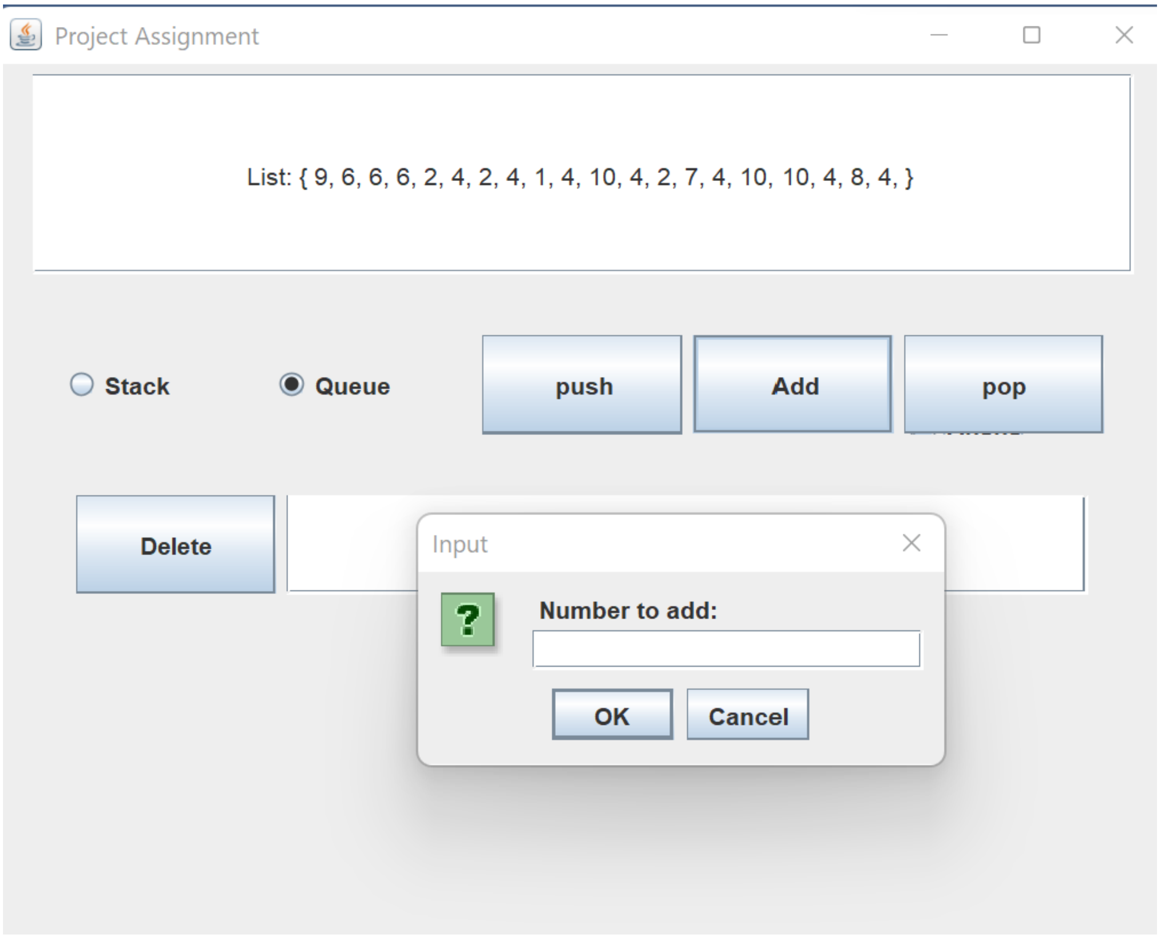
Successfully pushed 12 to List



List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, }

● Stack ○ Queue

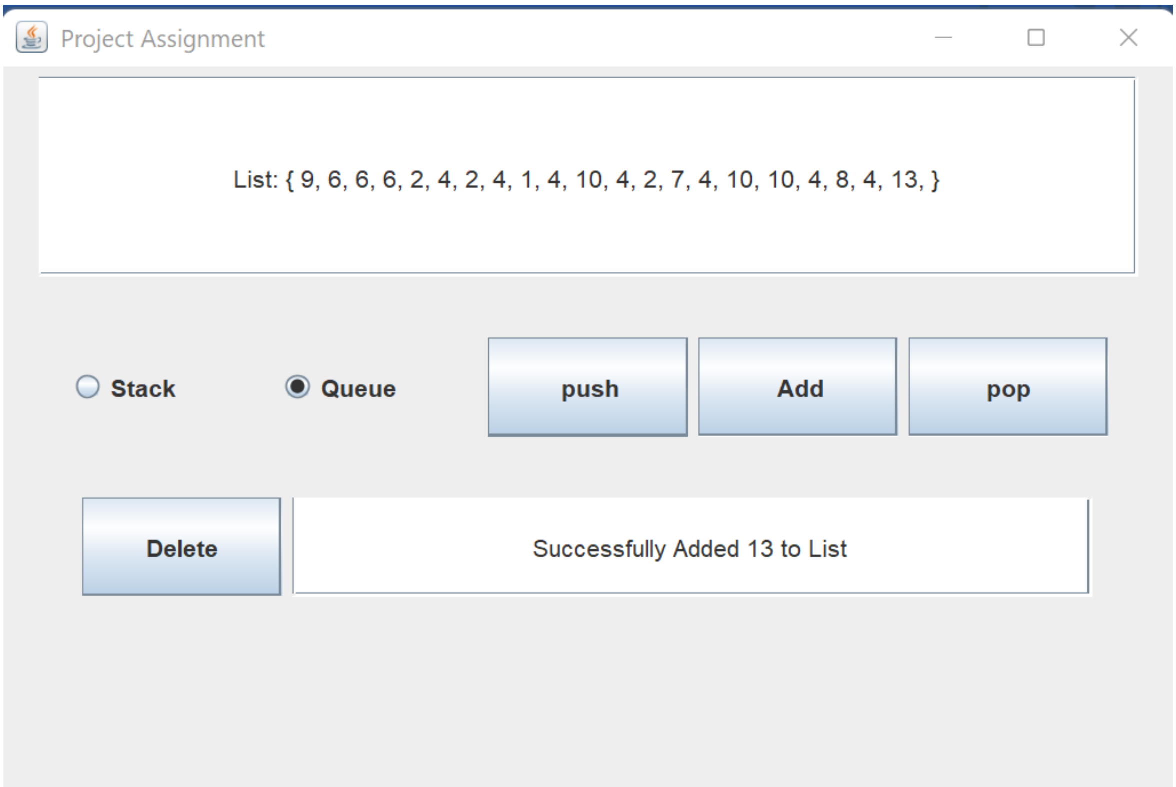
Successfully poped 12 from List



List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, }

○ Stack ● Queue

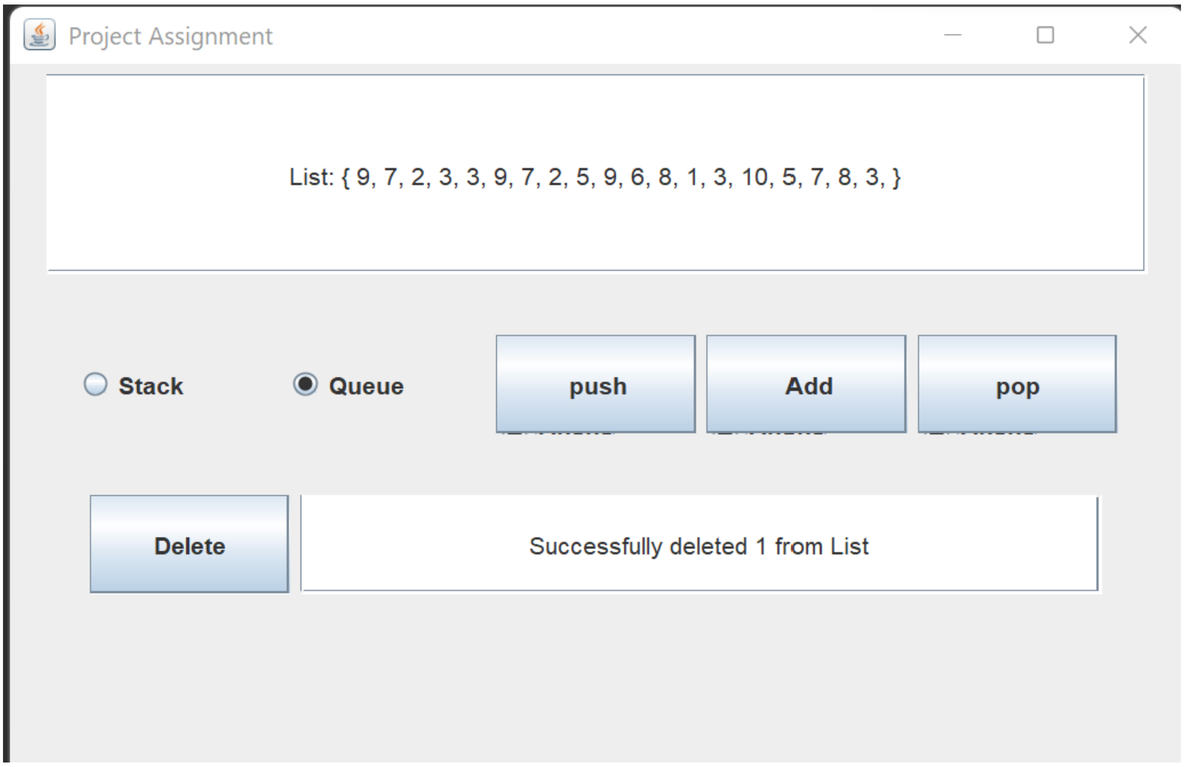
Number to add:



List: { 9, 6, 6, 6, 2, 4, 2, 4, 1, 4, 10, 4, 2, 7, 4, 10, 10, 4, 8, 4, 13, }

○ Stack ● Queue

Successfully Added 13 to List



List: { 9, 7, 2, 3, 3, 9, 7, 2, 5, 9, 6, 8, 1, 3, 10, 5, 7, 8, 3, }

○ Stack ● Queue

Successfully deleted 1 from List

# BIBLIOGRAPHY:

- INTRODUCTION TO JAVA PROGRAMMING AND DATA STRUCTURE BY Y DANIEL LIANG,
- www.javatpoint.com
- www.geeksforgeeks.com
- www.stackoverflow.com

# GITHUB:

Ashish-192/Java-project (github.com)