

Let's Analyze

Code:

```
1  import java.util.Arrays;
2  import javax.swing.*;
3
4
5
6  public class TheStack {
7
8      private String[] StackArray;
9
10     private int stackSize;
11     private int topOfStack = -1;
12
13
14
15     TheStack(int size){
16         stackSize = size;
17
18         StackArray = new String[size];
19
20         Arrays.fill(StackArray, val: "-1");
21     }
22
23
24     public String displayTheStack(){
25         String output = "";
26         for(int i=0; i<45;i++) {
27             output += "=";
28         }
29         output += "\n";
30
31         for(int i = 0; i < 10; i++){
32
33             output += "| " + StackArray[i];
34
35         }
36
37         output += "|\n";
38
39         for(int i=0; i<45;i++) {
40             output += "=";
41         }
42         output += "\n";
43     }
```

```
44         return output;
45
46
47     }
48
49
50
51     public void push(String input){
52
53         if (topOfStack < stackSize){
54
55             topOfStack++;
56
57             StackArray[topOfStack] = input;
58
59         }
60
61         else {
62             String output = "";
63             output += "Sorry but the stack is full.";
64             output += displayTheStack();
65             output += "PUSH " + input + " was added to the stack.";
66             JOptionPane.showMessageDialog(parentComponent: null, output);
67         }
68
69     }
70
71
72
73     public String pop() {
74
75         String output = "";
76         if(topOfStack >= 0) {
77
78             output += displayTheStack();
79             output += "POP " + StackArray[topOfStack] + " was removed from the Stack.\n";
80             JOptionPane.showMessageDialog(parentComponent: null, output);
81             StackArray[topOfStack] = "-1";
82
83             return StackArray[topOfStack--];
84
85         }
```

```
86     else {
87
88         output += displayTheStack();
89
90
91         output += "Sorry but the stack is empty.";
92         JOptionPane.showMessageDialog(parentComponent: null, output);
93         return "-1";
94     }
95 }
96
97 }
98
99
100
101 public String peek() {
102     String output = "";
103     output += displayTheStack();
104     output += "PEEK " + StackArray[topOfStack] + " is at the top of the stack.\n";
105
106     JOptionPane.showMessageDialog(parentComponent: null, output);
107     return StackArray[topOfStack];
108 }
109
110
111
112
113 public void pushMany(String multipleValues) {
114
115     String[] tempString = multipleValues.split(regex: " ");
116
117     for(int i = 0; i < tempString.length; i++) {
118
119         push(tempString[i]);
120
121     }
122 }
123
124
125
126 public void popAll() {
127
```

```

128         for(int i = topOfStack; i >= 0; i--) {
129             pop();
130         }
131     }
132 }
133
134 }
135
136
137
Run | Debug
138 public static void main(String[] args) {
139
140     TheStack theStack = new TheStack(size: 10);
141     int menuChoice;
142     do{
143
144         menuChoice = Integer.parseInt(JOptionPane.showInputDialog("Press 1 to Push\nPress 2 to Pop\nPress 3 to Display\nPress 4 to PushMany\nPress 5 to P
145
146         switch(menuChoice){
147             case 1:
148                 String value = JOptionPane.showInputDialog(message: "Enter string:");
149                 theStack.push(value);
150                 break;
151             case 2:
152                 theStack.pop();
153                 break;
154             case 3:
155                 JOptionPane.showMessageDialog(parentComponent: null,theStack.displayTheStack());
156                 break;
157             case 4:
158                 String values = JOptionPane.showInputDialog(message: "Enter values");
159                 theStack.pushMany(values);
160                 break;
161             case 5:
162                 theStack.popAll();
163                 break;
164             case 6:
165                 theStack.peak();
166                 break;
167             case 7:
168                 System.exit(status: 1);
169         }

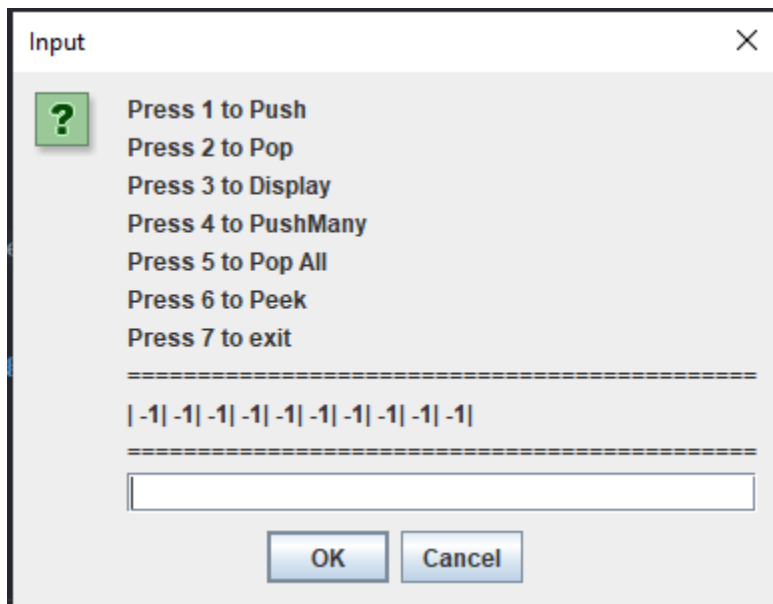
```

```

151         case 2:
152             theStack.pop();
153             break;
154         case 3:
155             JOptionPane.showMessageDialog(parentComponent: null, theStack.displayTheStack());
156             break;
157         case 4:
158             String values = JOptionPane.showInputDialog(message: "Enter values");
159             theStack.pushMany(values);
160             break;
161         case 5:
162             theStack.popAll();
163             break;
164         case 6:
165             theStack.peek();
166             break;
167         case 7:
168             System.exit(status: 1);
169     }
170 }while(menuChoice != 7);
171
172
173
174
175
176
177
178
179 }

```

Output:



Input

×

?

Press 1 to Push

Press 2 to Pop

Press 3 to Display

Press 4 to PushMany

Press 5 to Pop All

Press 6 to Peek

Press 7 to exit

=====

| -1| -1| -1| -1| -1| -1| -1| -1| -1|

=====

OK

Cancel

Input

×

?

Enter string:

OK

Cancel

Input

×

?

Press 1 to Push

Press 2 to Pop

Press 3 to Display

Press 4 to PushMany

Press 5 to Pop All

Press 6 to Peek

Press 7 to exit

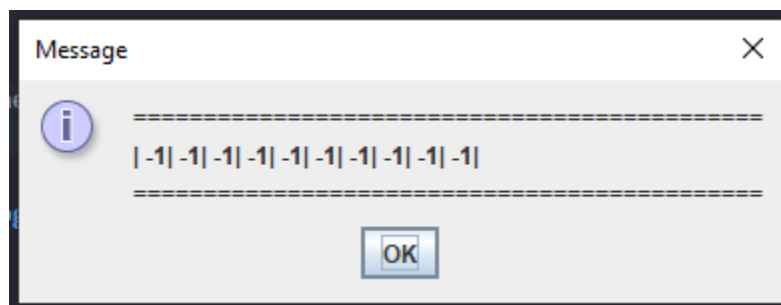
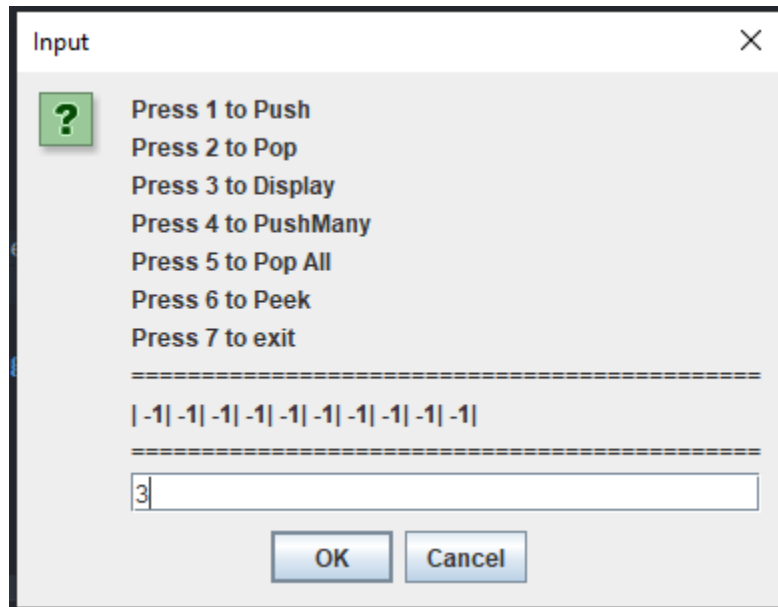
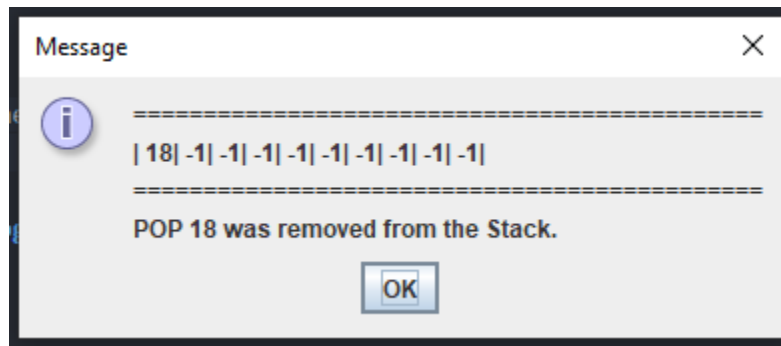
=====

| 18| -1| -1| -1| -1| -1| -1| -1| -1| -1|

=====

OK

Cancel



Input

×

?

Press 1 to Push

Press 2 to Pop

Press 3 to Display

Press 4 to PushMany

Press 5 to Pop All

Press 6 to Peek

Press 7 to exit

=====

| -1| -1| -1| -1| -1| -1| -1| -1| -1|

=====


4

OK

Cancel




Input ✕

 Enter values

19 20 30 40 50 60

OK Cancel

Input ✕

 Press 1 to Push  
Press 2 to Pop  
Press 3 to Display  
Press 4 to PushMany  
Press 5 to Pop All  
Press 6 to Peek  
Press 7 to exit

=====


| 19| 20| 30| 40| 50| 60| -1| -1| -1| -1|

=====

5

OK Cancel

Message ✕

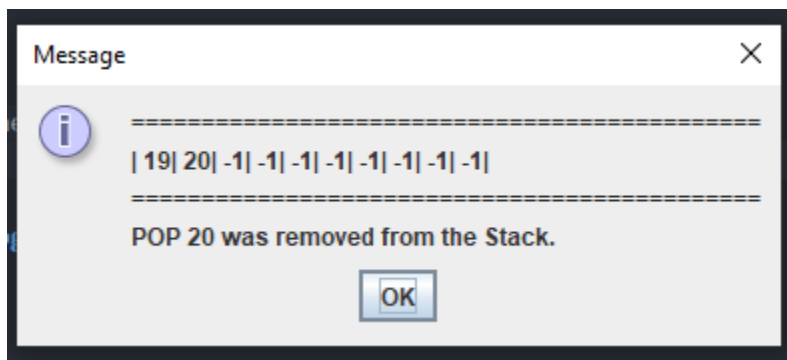
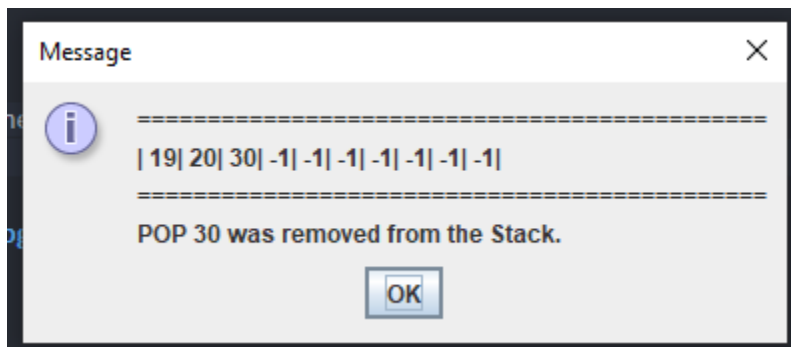
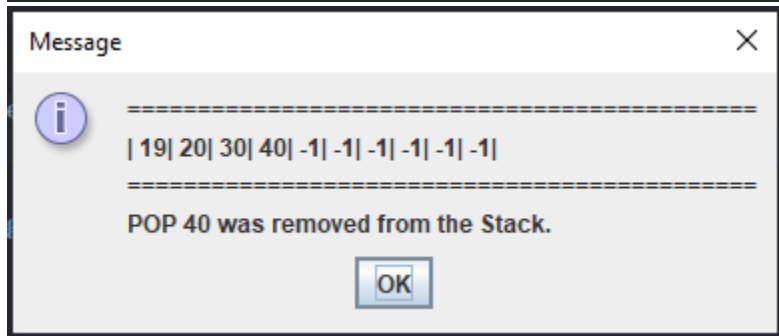
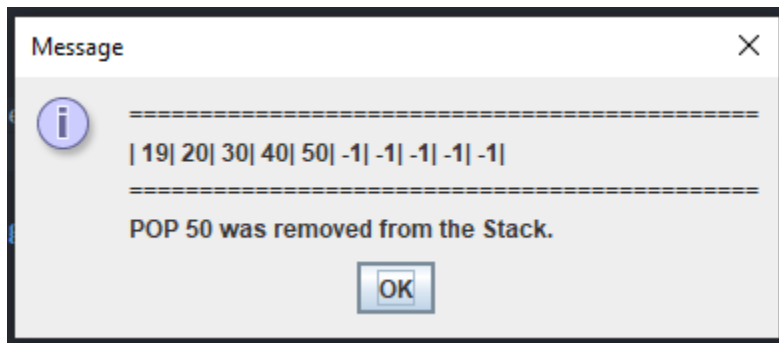
 =====

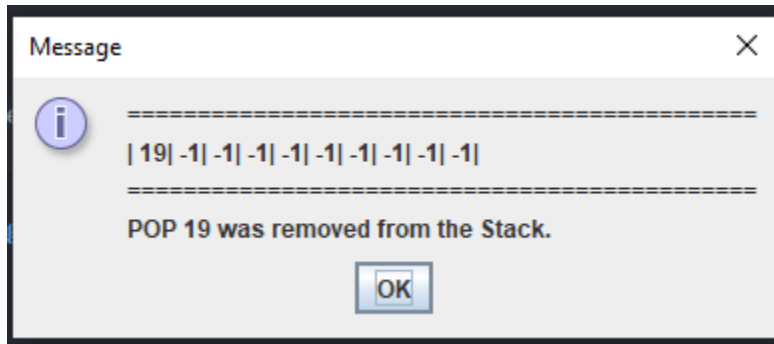
| 19| 20| 30| 40| 50| 60| -1| -1| -1| -1|

=====

POP 60 was removed from the Stack.

OK





After theStack.pushMany("10 20 30 40"); (For peek demonstration)

