

package stack;

### NEW ERA UNIVERSITY COLLEGE OF COMPUTER STUDIES

NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

(02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph



#### Marasigan, Vem Aiensi A. -BSCS2

02-25-2022

#### **SOURCE CODE**

```
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
public class BSCS_2_Marasigan_Stack
{
     static Scanner in = new Scanner(System.in);
     static int size = 0;
     static int limit = 0;
     static int choice = 0;
     static int push = 0;
     static JTextField scope = new JTextField();
     public static void main(String[] args)
          boolean repeat = true;
          do {
          try
          {
               System.out.println("[1] Console Mode\t[2]GUI-Beta");
               System.out.print("Choose mode: ");
               choice = in.nextInt();
               switch (choice)
                    case 1: consoleMode(); break;
                    case 2: System.out.println("Stack (Beta) is now launched...");
                              stackGUlbeta(); break;
                    default: System.out.println("Wrong input, Please try again\n"); break;
               if ((choice == 1)|| (choice == 2) )
                    repeat = false;
          catch (Exception e)
               System.out.println("Wrong input, Please try again\n");
          }while (repeat);
     }
     static void consoleMode()
          boolean repeat=true;
          while (repeat)
               try{
                    System.out.print("How many elements will the stack have: ");
                    limit = in.nextInt();
               catch (Exception e){
                    limit = 0; in.next();
                    System.out.println("Wrong input, Please try again\n");
               if (limit > 0)
                    repeat = false;
          }
          int [] stack = new int[limit];
          int choice = 0;
          System.out.println("Stack Created!");
          do {
```



try {

### NEW ERA UNIVERSITY COLLEGE OF COMPUTER STUDIES



NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

```
System.out.println("MENU\n1. PUSH\n2. POP\n3. TOP\n4."
                        + " SIZE\n5. EMPTY\n6. DISPLAY STACK\n7. EXIT");
              System.out.print("Enter Choice [1..7]: ");
              choice = in.nextInt();
              switch (choice)
                   case 1: System.out.print("Enter number to add to the stack: ");
                   push = in.nextInt();
                   pusher(stack, limit, push); displayStack(stack); break;
                   case 2: poper(stack); break;
                   case 3: showTop(stack);break;
                   case 4: stackSize(); break;
                   case 5: stackChecker(stack); break;
                   case 6: displayStack(stack); break;
                   case 7: System.out.println("Thank you po for using my program\n"
                             + "-by Vem Aiensi A. Marasigan");break;
                   default:System.out.println("Wrong input, Please try again\n"); break;
              }
         }
         catch (Exception e)
         {
              choice = 0; in.next();
              System.out.println("Wrong input, Please try again\n");
    }while (choice != 7);
}
static void stackGUIbeta()
    //Note: I learned some of GUI from Alex Lee and Bro Code's Java GUI tutorial
    //All credits to them ^ ^;
    //And I am very Grateful >v< to our Very Kind Professor for extending the deadline
    JFrame createStack = new JFrame();
    scope.setBounds(50, 80, 100, 25);
     JLabel elements = new JLabel("How many elements will the stack have?");
     elements.setBounds(45, 40, 400, 25);
     JButton insert = new JButton("CREATE");
     insert.setBounds(170, 80, 100, 25);
    ActionListener make = new ActionListener()
         public void actionPerformed(ActionEvent e)
                        limit = Integer.parseInt(scope.getText());
                        System.out.println("Stack Created!\n" +
                        limit + " elements can be stored in the stack.\n");
                        int [] stack = new int[limit];
                        JLabel ask=new JLabel("Please type number here:");
                        ask.setBounds(50, 45, 220, 25);
                        JTextField input = new JTextField();
                        input.setBounds(220, 45, 170, 25);
                        JButton button1 = new JButton("PUSH");
                        button1.setBounds(50, 90, 100, 25);
                        ActionListener one = new ActionListener() {
                             public void actionPerformed(ActionEvent e)
                                  push = Integer.parseInt(input.getText());
                              pusher(stack, limit, push); displayStack(stack);
                           }
```



### NEW ERA UNIVERSITY

#### **COLLEGE OF COMPUTER STUDIES**





```
button1.addActionListener(one);
JButton button2 = new JButton("POP");
button2.setBounds(170, 90, 100, 25);
ActionListener two = new ActionListener() {
    public void actionPerformed(ActionEvent e)
     poper(stack);
  };
button2.addActionListener(two);
JButton button3 = new JButton("TOP");
button3.setBounds(290, 90, 100, 25);
ActionListener three = new ActionListener() {
    public void actionPerformed(ActionEvent e)
          showTop(stack);
  };
button3.addActionListener(three);
JButton button4 = new JButton("SIZE");
button4.setBounds(50, 135, 100, 25);
ActionListener four = new ActionListener() {
    public void actionPerformed(ActionEvent e)
         stackSize();
  };
button4.addActionListener(four);
JButton button5 = new JButton("EMPTY");
button5.setBounds(170, 135, 100, 25);
ActionListener five = new ActionListener() {
    public void actionPerformed(ActionEvent e)
         stackChecker(stack);
  };
button5.addActionListener(five);
JButton button6 = new JButton("DISPLAY");
button6.setBounds(290, 135, 100, 25);
ActionListener six = new ActionListener() {
    public void actionPerformed(ActionEvent e)
         displayStack(stack);
button6.addActionListener(six);
JButton button7 = new JButton("END");
button7.setBounds(170, 180, 100, 25);
ActionListener seven = new ActionListener() {
    public void actionPerformed(ActionEvent e)
   {
         System.out.println("\t\t-=StackGUI-Beta=-\n"
                   + "\t by Vem Aiensi Marasigan >.<\n\n\n"
                   + "\n Note: I'm new po to GUI and it may lack features");
     System.exit(0);
button7.addActionListener(seven);
```



**}**;

}

//Methods Here

{

} else

}

} else

}

size++;

System.out.println("Stack is Empty\n");

if (size > 0)

#### **NEW ERA UNIVERSITY** COLLEGE OF COMPUTER STUDIES



NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES (02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph

createStack.setVisible(false); JPanel panel = new JPanel(); panel.setLayout(null); panel.add(button1); panel.add(button2); panel.add(button3); panel.add(button4); panel.add(button5); panel.add(button6); panel.add(button7); panel.add(input); panel.add(ask); JFrame mainWindow = new JFrame(); mainWindow.setTitle("Stack -Beta"); //mainWindow.setDefaultCloseOperation(1); mainWindow.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); mainWindow.setSize(450, 300); mainWindow.setVisible(true); mainWindow.setResizable(false); mainWindow.add(panel); insert.addActionListener(make); JPanel firstPanel = new JPanel(); firstPanel.setLayout(null); firstPanel.add(insert); firstPanel.add(elements); firstPanel.add(scope); createStack.setTitle("Stack -Beta"); //createStack.setDefaultCloseOperation(1); createStack.setSize(350, 200); createStack.setVisible(true); createStack.setResizable(false); createStack.add(firstPanel); static void pusher(int[] stack, int limit, int push) if (size==limit) System.out.println("Stack Overflow");  $System.out.println("Stack data: "+push + " successfully pushed\n");$ stack[size]=push; static void poper(int[] stack) System.out.println(stack[size-1] +" is removed from the stack.\n");



}

### NEW ERA UNIVERSITY COLLEGE OF COMPUTER STUDIES



NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES (02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph

```
}
static void showTop(int[] stack)
     if (size > 0)
     {
          System.out.println(stack[size-1] +" is the top value of the stack.\n");
     }
     else
     {
          System.out.println("Stack is Empty\n");
}
static void stackSize()
{
     if (size>0)
          System.out.println("The stack has a size of "+size + ".\n");
     }
     else
     {
          System.out.println("Size is 0: Stack is Empty.\n");
static void stackChecker(int[] stack)
{
     if (size==0){
          System.out.println("TRUE: The Stack is Empty\n");
     else
          System.out.println("FALSE: The Stack contains " + size + " elements.\n");
}
static void displayStack(int[] stack)
{
     if(size==0)
          System.out.println("Stack is Empty\n");
     }
     else
          System.out.println("The stack contains: ");
          for (int n=0; n<size; n++)
               System.out.print("+----");
          System.out.println("+");
          for (int n=0; n<size; n++)
          {
               System.out.print("|"+stack[n]+"\t");
          System.out.println("|");
          for (int n=0; n<size; n++)
               System.out.print("+----");
          System.out.println("+");
     }
}
```





NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

(02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph

#### **SCREENSHOTS**

BSCS_2_Marasigan_Stack [Java Application] C:\Program	BSCS_2_Marasigan_Stack [Java Application] C:\Program Fi
[1] Console Mode [2]GUI-Beta	MENU
Choose mode: 1	1. PUSH
How many elements will the stack have: 3	2. POP
Stack Created!	3. TOP
MENU	4. SIZE
1. PUSH	5. EMPTY
2. POP	6. DISPLAY STACK
3. TOP 4. SIZE	7. EXIT
5. EMPTY	Enter Choice [17]: 6
6. DISPLAY STACK	The stack contains:
7. EXIT	las ls l
Enter Choice [17]: 1	45 6
Enter number to add to the stack: 45	MENU
Stack data: 45 successfully pushed	1. PUSH
	2. POP
The stack contains:	3. TOP
++	4. SIZE
45	5. EMPTY
++	6. DISPLAY STACK
MENU	7. EXIT
1. PUSH	Enter Choice [17]: 4
2. POP	The stack has a size of 2.
3. TOP	
4. SIZE	MENU
5. EMPTY	1. PUSH
6. DISPLAY STACK 7. EXIT	2. POP
Enter Choice [17]: 1	3. TOP
Enter number to add to the stack: 6	4. SIZE 5. EMPTY
Stack data: 6 successfully pushed	6. DISPLAY STACK
Seach data. O successivity pushed	7. EXIT
The stack contains:	Enter Choice [17]: 1
++	Enter number to add to the stack: 99
45  6	Stack data: 99 successfully pushed
++	•
MENU	The stack contains:
1. PUSH	++
2. POP	45  6  99
3. TOP	++
4. SIZE	MENU
5. EMPTY	1. PUSH
6. DISPLAY STACK	2. POP 3. TOP
7. EXIT	4. SIZE
Enter Choice [17]: 1 Enter number to add to the stack: 23	5. EMPTY
Stack data: 23 successfully pushed	6. DISPLAY STACK
Stack data. 25 successfully pushed	7. EXIT
The stack contains:	Enter Choice [17]: 1
+	Enter number to add to the stack: 12
45  6  23	Stack Overflow
+	The stack contains:
MENU	+
1. PUSH	45  6  99
2. POP	++
3. TOP	MENU
4. SIZE	1. PUSH
5. EMPTY	2. POP 3. TOP
6. DISPLAY STACK	4. SIZE
7. EXIT	5. EMPTY
Enter Choice [17]: 2	6. DISPLAY STACK
23 is removed from the stack.	7. EXIT



#### **NEW ERA UNIVERSITY**

#### **COLLEGE OF COMPUTER STUDIES**



NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

(02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph

```
BSCS_2_Marasigan_Stack [Java Application] C:\Program F<terminated> BSCS_2_Marasigan_Stack [Java Applicat
TOP
                                           MENU
4. SIZE
                                           1. PUSH
EMPTY
                                            POP
6. DISPLAY STACK
                                           3. TOP
EXIT

 SIZE

Enter Choice [1..7]: 3
                                           EMPTY
99 is the top value of the stack.
                                            DISPLAY STACK
                                           7. EXIT
MENU
                                           Enter Choice [1..7]: 6

    PUSH

                                           Stack is Empty
2. POP
3. TOP
                                           MENU

 SIZE

    PUSH

EMPTY
                                           2. POP
DISPLAY STACK
                                           TOP
EXIT

 SIZE

Enter Choice [1..7]:
                                           5. EMPTY
                                           DISPLAY STACK
FALSE: The Stack contains 3 elements.
                                           7. FXIT
                                           Enter Choice [1..7]: 1
MENU
                                           Enter number to add to the stack: 1

    PUSH

                                           Stack data: 1 successfully pushed
2. POP
TOP
                                           The stack contains:
4. SIZE
EMPTY
                                            1
6. DISPLAY STACK
7. EXIT
                                           MENU
Enter Choice [1..7]: 2

    PUSH

99 is removed from the stack.
                                           2. POP
                                           TOP
MENU
                                           SIZE

    PUSH

                                           EMPTY
2. POP
                                           6. DISPLAY STACK
3. TOP
                                           EXIT
4. SIZE
                                           Enter Choice [1..7]: 1
EMPTY
                                            Enter number to add to the stack: 2
6. DISPLAY STACK
                                           Stack data: 2 successfully pushed
7. EXIT
Enter Choice [1..7]: 2
                                           The stack contains:
6 is removed from the stack.
                                           1
                                                 2
MENU
                                           +----

    PUSH

                                           MENU
2. POP

    PUSH

    TOP
    SIZE

                                           2. POP
                                           TOP
EMPTY
                                           SIZE
6. DISPLAY STACK
                                           5. EMPTY
EXIT
                                           6. DISPLAY STACK
Enter Choice [1..7]: 2
                                           EXIT
45 is removed from the stack.
                                           Enter Choice [1..7]: 4
                                           The stack has a size of 2.
MENU
1. PUSH
                                           MENU
2. POP

    PUSH

3. TOP
                                           2. POP

 SIZE

                                           3. TOP
5. EMPTY

 SIZE

DISPLAY STACK
                                           5. EMPTY
7. EXIT
                                           6. DISPLAY STACK
Enter Choice [1..7]: 5
                                           7. EXIT
TRUE: The Stack is Empty
                                           Enter Choice [1..7]: 7
                  Thank you po for using my program
```

Thank you po for using my program -by Vem Aiensi A. Marasigan





NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

		□ Console ×
Stack -Beta — X  How many elements will the stack have?  4 CREATE		BSCS_2_Marasigan_Stack [Java Application] C:\Program Fil  [1] Console Mode [2]GUI-Beta Choose mode: 2 Stack (Beta) is now launched
Stack -Beta - X  Please type number here: 1  PUSH POP TOP  SIZE EMPTY DISPLAY  END		BSCS_2_Marasigan_Stack [Java Application] C:\Program Files\Java\jdk-17\  [1] Console Mode [2]GUI-Beta Choose mode: 2 Stack (Beta) is now launched Stack Created! 4 elements can be stored in the stack.  Stack data: 1 successfully pushed  The stack contains: +
Please type number here:  PUSH POP TOP  SIZE EMPTY DISPLAY  END  Please type number here:  PUSH POP TOP  SIZE EMPTY DISPLAY  END  PLEASE TYPE DISPLAY  END  END	×	BSCS_2_Marasigan_Stack [Java Application] C:\Program Files\Java\java\java\]  [1] Console Mode
(ck):break:		





NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES

Stack -Beta − X	the state of the s
	Stack data: 3 successfully pushed
	The stack contains:
Please type number here:	1   3
Flease type number nere.	Stack data: 9 successfully pushed
PUSH POP TOP	The stack contains:
100	++
SIZE EMPTY DISPLAY	1  3  9
	The stack has a size of 3.
END	9 is removed from the stack.
	3 is the top value of the stack.
(), bicak,	
Stack -Beta − □ X	The stack contains:
	++ Stack data: 9 successfully pushed
Please type number here:	
Please type number nere.	The stack contains:
PUSH POP TOP	1   3   9
POSH	The stack has a size of 3.
SIZE EMPTY DISPLAY	9 is removed from the stack.
SIEE EMPTT DISTERT	
END	3 is the top value of the stack.
E.I.D	The stack contains:
	1  3
	++
(Stack), Dieak,	The stack has a size of 3.
Stack - Beta	
	9 is removed from the stack.
	3 is the top value of the stack.
Please type number here:	The stack contains:
	1  3
PUSH POP TOP	3 is removed from the stack.
SIZE EMPTY DISPLAY	1 is removed from the stack.
	Stack is Empty
END	
lg input, riease try again(n ),	





NO. 9 CENTRAL AVE., NEW ERA, QUEZON CITY, PHILIPPINES (02) 981-4221 | computerstudies@neu.edu.ph | www.neu.edu.ph

Stack -Beta − □ ×	The stack has a size of 3.
	9 is removed from the stack.
	3 is the top value of the stack.
Please type number here:	The stack contains:
Duny Don Ton	1   3
PUSH POP TOP	3 is removed from the stack.
SIZE EMPTY DISPLAY	1 is removed from the stack.
	Stack is Empty
END	TRUE: The Stack is Empty
No commence of the available of	
Stack -Beta	1 1 1
	Stack data: 1 successfully pushed
	The stack contains:
Please type number here: 3	+++
PUSH POP TOP	Stack data: 1 successfully pushed
101	The stack contains:
SIZE EMPTY DISPLAY	1
	Stack Overflow
END	The stack contains:
	1 1 1 1 1
	Stack Overflow
	The stack contains:
	1  1  1  1
	•

-=StackGUI-Beta=by Vem Aiensi Marasigan >.<

Note: I'm new po to GUI and it may lack features

^//Excecuted when END button is pressed^

(I'm very sorry po for using GUI because it's the only thing I had in mind to make inputs without pressing the keyboard's ENTER button)