Ex No: 06	ADT STACK
Date: 13/08/2019	

Aim:

*To write a java program to design Java interface for ADT Stack and to implement this interface using array.

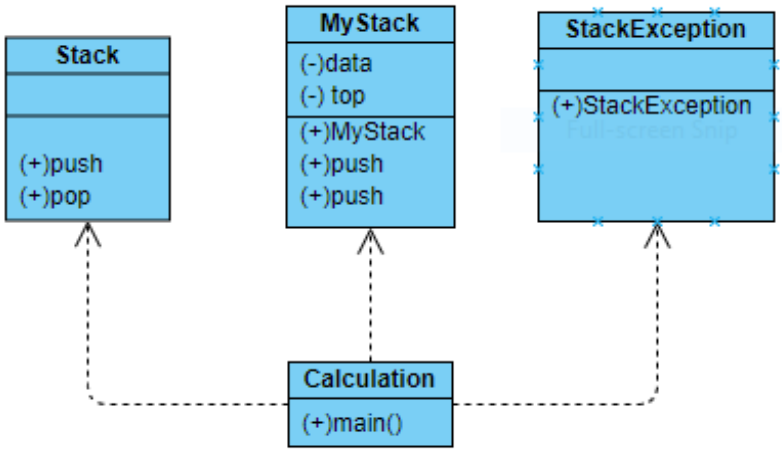
Requirements:

*Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations. Verify the implementation by pushing string data.

Algorithm:

1. Create a package mystack.
2. Create an interface Stack.
3. Create a class MyStack that implements the interface Stack.
4. Create a class StackException which inherits the Exception class.
5. Create a class Calculation and give the data for the stack.
6. Provide the necessary attributes to the class MyStack.
7. Declare case (1) for push a string.
8. Declare case (2) for pop a string.
9. Declare case (3) for exit the application.
10. Stop

Class Diagram:



Program:

```
/**
*Developed by
*D. Sarathi Raj
*212217105054
*Saveetha Engineering College
*sarathiraj852000@gmail.com
*/

package mystack;

public interface Stack {
	public void push(String value1)  throws StackException;
	public String pop() throws StackException;
}
package mystack;

public class StackException extends Exception {
	public StackException(String m)
	{
		super(m);
	}
}
package mystack;

public class MyStack implements Stack {
	private String data[];
	private int top;

	public MyStack(int s)
	{
		top=-1;
```

```
            data=new String[s];
      }
      @Override
      public void push(String v) throws StackException {
            // TODO Auto-generated method stub
            if(top>=(data.length-1))
            {
                  throw new StackException("Stack Full: It is already
having "+(top+1)+" elements");
            }
            top=top+1;
            data[top]=v;
      }

      @Override
      public String pop() throws StackException {
            // TODO Auto-generated method stub
            String result;
            if(top<0)
            {
                  throw new StackException("Stack is empty");
            }
            result=data[top];
            top=top-1;

            return result;
      }

}
package mystack;

import java.util.Scanner;

public class Calculation {
      public static void main(String[] args) {
            String value1;
            int option;
            Stack st;

            Scanner sc=new Scanner(System.in);
            st=new MyStack(5);

            while(true)
            {
                  try
                  {
                        System.out.println("1. Push a string");
                        System.out.println("2. Pop a string");
                        System.out.println("3. Exit");
                        System.out.print("Enter your choice:");

                        option=sc.nextInt();
                        switch(option)
                        {
```

```
                        case 1:
                              System.out.print("Enter a string:");
                              value1=sc.next();
                              st.push(value1);
                              System.out.println("Push completed.");
                              break;
                        case 2:
                              value1=st.pop();
                              System.out.printf("Stack top
value=%s\n",value1);
                              break;
                        default:
                              System.out.print("Please enter a valid
number !!!");
                        }

                        if(option==3)
                        {
                              System.out.print("Thankyou for using stack
application !!!");
                              break;
                        }

                  }catch(StackException e1)
                  {
                        System.out.println("Error:"+e1.getMessage());
                  }catch(NumberFormatException e2)
                  {
                        System.out.println("Error:"+e2.getMessage());
                  }
                  }
            }
}
```

Output:

```
1. Push a string
2. Pop a string
3. Exit
Enter your choice:1
Enter a string:mahendra
Push completed.
1. Push a string
2. Pop a string
3. Exit
Enter your choice:1
Enter a string:singh
Push completed.
1. Push a string
2. Pop a string
3. Exit
Enter your choice:1
Enter a string:dhoni
Push completed.
```

```
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=dhoni
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=singh
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=mahendra
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Error:Stack is empty
1. Push a string
2. Pop a string
3. Exit
Enter your choice:3
Please enter a valid number !!!Thankyou for using stack application !!!
```

Result:

*Thus, the java program for implanting ADT Stack is written and executed successfully.