

EXP.NO:06	ADT APPLICATION
DATE:19.08.19	

### AIM:

To design a java program for ADT stack and to implement this interface using array by providing necessary handling in both the implementation by pushing and popping string data

### REQUIREMENT:

- knowledge of push and pop
- Exception handling
- Handling of array
- Interface implementation

### ALGORITHM:

STEP 1: Start

STEP 2: create classes Mystack, Stack, Calculation and StackException

STEP 3: Define StackException with string in it

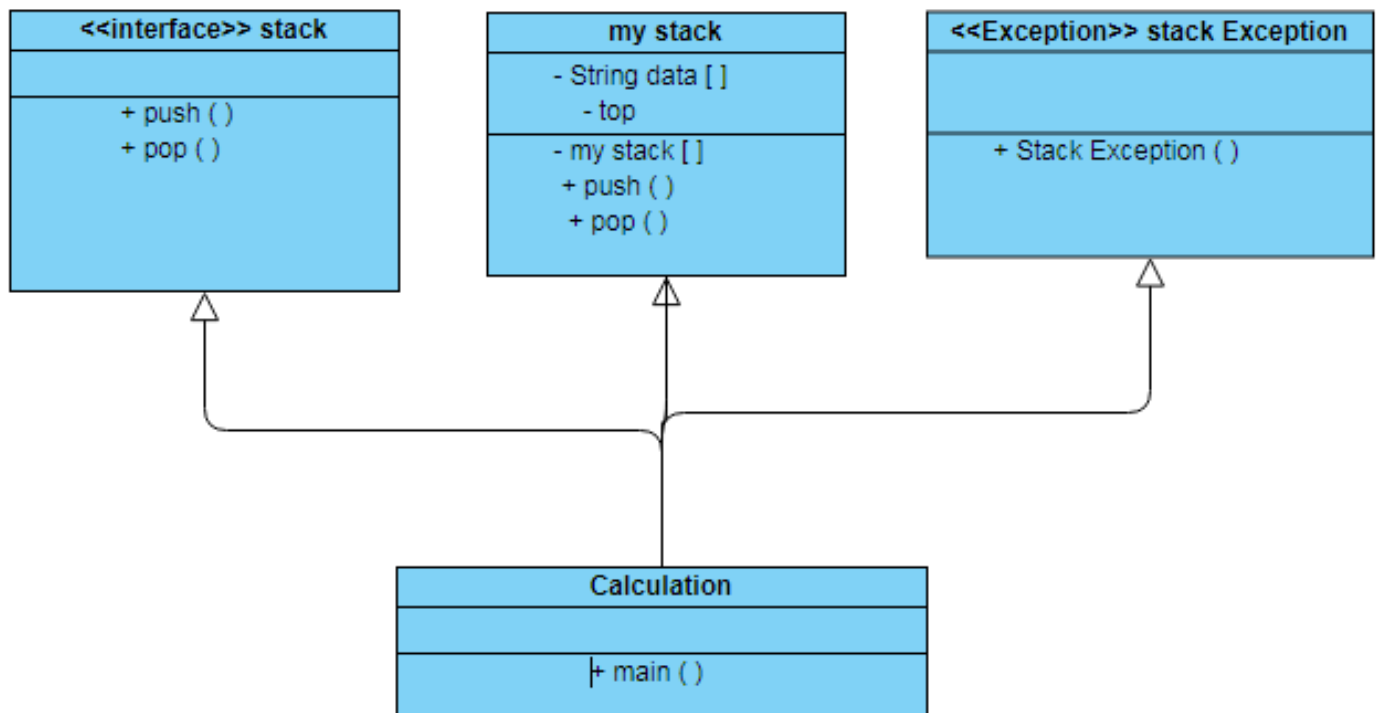
STEP 4: Define the interface by using throw exception

STEP 5: To add data define the data type i.e. string for describing different cases define the operation of each case to meet the requirement

STEP 6: Finish the coding with calculation class coding for the required output

STEP 7: Stop

### CLASS DIAGRAM:



### PROGRAM:

```

/**created by
 *R.Neha,

```

```

* eee-b, 212217105049
* Saveetha Engineering College.
*/
package mystack;
public interface Stack {
    public void push(String v) 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
    {
        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
    {
        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:uday
Push completed.
1. Push a String
2. Pop a String
3. Exit
Enter your choice:1
Enter a String:hari
Push completed.
1. Push a String

```

```
2. Pop a String
3. Exit
Enter your choice:1
Enter a String:jaswanth
Push completed.
1. Push a String
2. Pop a String
3. Exit
Enter your choice:2
Stack top value=jaswanth
1. Push a String
2. Pop a String
3. Exit
Enter your choice:2
Stack top value=hari
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:**Hence,A java program for ADT stack and to implement this interface using array by providing necessary handling in both the implementatin by pushing and popping string data is done