| EX: 06 | ADT STACK |
|---|---|
| 31-08-2019 | |

# Aim:

To write a java console application to design a java interface for ADT Stack. Implement this interface using array and to verify the implementation by pushing a string.

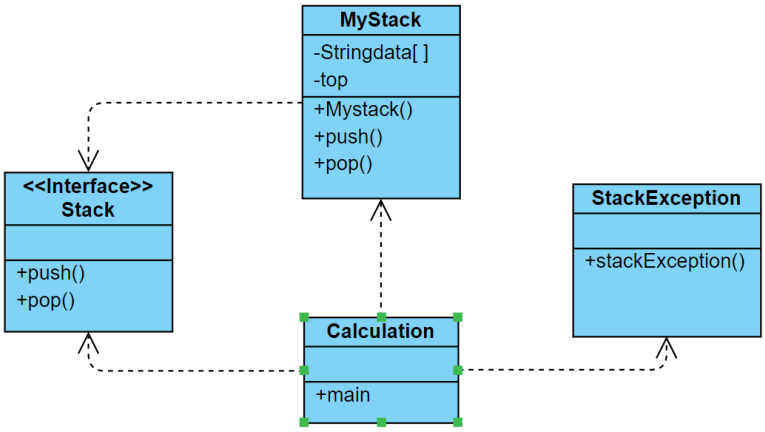
# Requirement:

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

# Algorithm:

Step 1: Create a mystack package.
Step 2: Create a separate class for calculation, mystack and stackexception.
Step 3: Create an interface class stack and implement it to all other class.
Step 4: Create the exception for seperate stackexception in the interface stack.
Step 5: Create a public void push() and string pop() and throw stack exception.
Step 6: Display the results.

# Class Diagram:




# Program:

Calculation.java

```
/*developed by: Sanjai Kumar
* gsanjaik@gmail.com
*/
package mystack;
import java.util.*;

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());
            }
            }
        }
```

```
}
```

MyStack.java

```
/*developed by: Sanjai Kumar
* gsanjaik@gmail.com
*/
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;
     }

}
```

Stack.java

```
/*developed by: Sanjai Kumar
* gsanjaik@gmail.com
*/
package mystack;
```

```java
public interface Stack {
     public void push(String v)  throws StackException;
     public String pop() throws StackException;
}
```

StackException.java

```java
/*developed by: Sanjai Kumar
* gsanjaik@gmail.com
*/
package mystack;

public class StackException extends Exception {

     public StackException(String m)
     {
          super(m);
     }
}
```

# Output:

```
1. Push a String
2. Pop a String
3. Exit
Enter your choice:1
Enter a String:Name
Push completed.
1. Push a String
2. Pop a String
3. Exit
Enter your choice:1
Enter a String:Class
Push completed.
1. Push a String
2. Pop a String
3. Exit
Enter your choice:2
Stack top value=Class
1. Push a String
2. Pop a String
3. Exit
Enter your choice:2
Stack top value=Name
1. Push a String
2. Pop a String
3. Exit
Enter your choice:3
Thanks for using the application!
```

## Result:

Thus the java console application for performing the string operation to push and pop using arraylist and thus the output is verified.