EXP.NO:06	ADT STACK 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 implementatin by pushing and poping string data

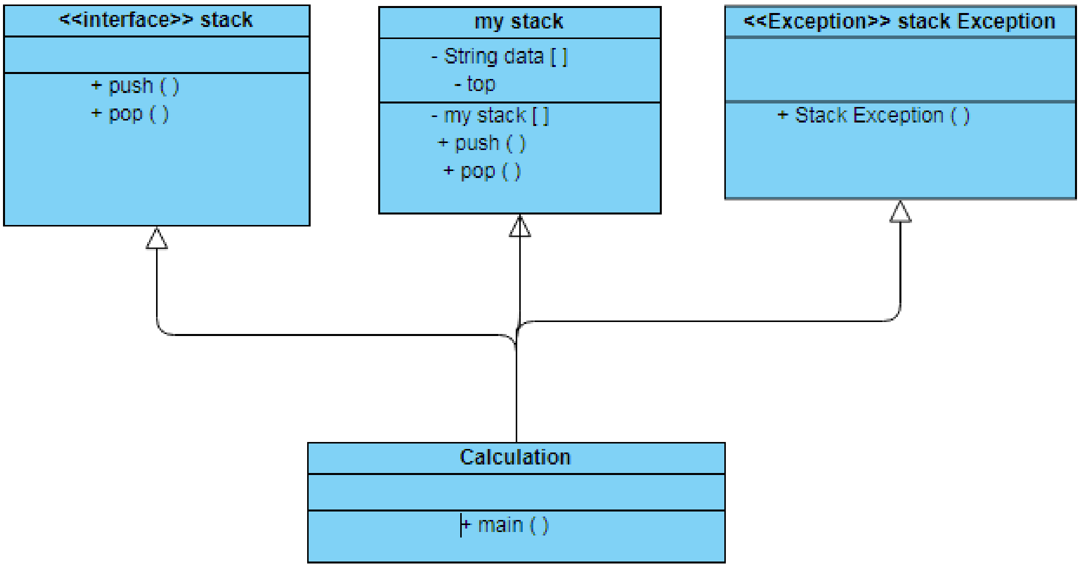
REQUIREMENT:

-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:

```
/***
* EXPERIMENT-06
*developed by Nithishkumar
*Saveetha Engineering College
*jpnithishkumar@gmail.com
*/
package mystack;

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

```
/***
* EXPERIMENT-06
*developed by Nithishkumar
*Saveetha Engineering College
*jpnithishkumar@gmail.com
*/
package mystack;

public class StackException extends Exception {

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

```
/***
* EXPERIMENT-06
*developed by Nithishkumar
*Saveetha Engineering College
*jpnithishkumar@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;
     }

}




/***
* EXPERIMENT-06
*developed by Nithishkumar
*Saveetha Engineering College
*jpnithishkumar@gmail.com
*/
package mystack;
import java.util.*;

public class Calculation {

     public static void main(String[] args) {
          String value1;
          int option;
          Mystack 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:nithish
Push completed.
1. Push a string
2. Pop a string
3. Exit
Enter your choice:1
Enter a string:kumar
Push completed.
1. Push a string
2. Pop a string
3. Exit
Enter your choice:1
Enter a string:palanisamy
Push completed.
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=palanisamy
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=kumar
1. Push a string
2. Pop a string
3. Exit
Enter your choice:2
Stack top value=nithish
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 Java program for ADT stack and to implement this interface using array by providing necessary handling in both the implementatin by pushing and poping string data is done.