DESIGN A JAVA INTERFACE FOR ADT STACK

AIM:

To Design a Java interface for ADT Stack and implement this interface using array, provide necessary exception handling in the implementation.

ALGORITHM:

1. Import the java packages.
2. Design an interface for MyStack with functions push, pop and display.
3. Define a class StackArray to implement the MyStack using array.
4. Define the functions of the interface accordingly and handle the stack overflow and underflow exceptions.
5. Create a class StackAdt and object for a class StackArray in memory and assign it to the reference variable, then the method is invoked.
6. By using Scanner class get the choices for switch statement during runtime.
7. By using switch case statement we can push, pop and display the elements for each choice.

PROGRAM:

*//File Name should be **StackAdt.java***

```
import java.util.Scanner; interface
MyStack
{     public void pop();
public void push();
public void display();
}

class StackArray implements MyStack
{     final static int n=5;
int stack[]=new int[n];
    int top=-1;

   public void push()
    {
        Scanner in;
try
        {
```

```
            in=new Scanner(System.in);
if(top==(n-1))
            {
                System.out.println(" Stack Overflow");
return;             }             else
            {
                System.out.println("Enter the element");
                int ele=in.nextInt();
                stack[++top]=ele;
            }
        }
        catch(Exception e)
        {
            System.out.println("e");
        }
    }

 public void pop()
    {
if(top<0)
        {
            System.out.println("Stack underflow");
return;         }         else
        {
            int popper=stack[top];
top--;
            System.out.println("Popped element:" +popper);
        }
    }

    public void display()
    {
if(top<0)
        {
            System.out.println("Stack is empty");
return;         }         else
        {
           String str=" ";             for(int i=0;
i<=top;  i++)                          str=str+"
"+stack[i]+"                                  -->";
System.out.println("Elements are:"+str);
```

```
        }
    }
}

class StackAdt
{
    public static void main(String arg[])
    {
        Scanner in= new Scanner(System.in);
        System.out.println("Implementation of Stack using Array");
        StackArray stk=new StackArray();
int ch=0;        do
        {
            System.out.println("1.Push 2.Pop 3.Display 4.Exit");
System.out.println("Enter your choice:");
            ch=in.nextInt();
            switch(ch)
            {            case 1:
stk.push();                 break;
case 2:                 stk.pop();
break;            case 3:
stk.display();
break;            case 4:
System.exit(0);
            }
        }
        while(ch<4);
    }
}
```

NOTE:

To Compile, ***javac StackAdt.java*** To Run

java StackAdt

OUTPUT:

```
D:\>javac StackAdt.java

D:\>java  StackAdt
Implementation of Stack using Array
1.Push  2.Pop  3.Display  4.Exit  Enter
your choice:
1
Enter the element
10
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
1
Enter the element
20
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
1
Enter the element
30
1.Push  2.Pop  3.Display  4.Exit
Enter your choice: 1
Enter the element
45
1.Push  2.Pop  3.Display  4.Exit
Enter your choice: 1
Enter the element
55
1.Push  2.Pop  3.Display  4.Exit
Enter your choice: 1
 Stack Overflow

1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Elements are:   10 -->  20 -->  30 -->  45 -->  55 -->
```

1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 2
Popped element:55
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Elements are: 10 --> 20 --> 30 --> 45 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 2
Popped element:45
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Elements are: 10 --> 20 --> 30 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 2
Popped element:30
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Elements are: 10 --> 20 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 2
Popped element:20
1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 3
Elements are: 10 -->

1.Push 2.Pop 3.Display 4.Exit
Enter your choice: 2
Popped element:10
1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Stack is empty

1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
3
Stack is empty

1.Push 2.Pop 3.Display 4.Exit Enter
your choice:
4
D:\>

RESULT:

Thus the Implementation for ADTStack interface using array has been successfully executed.