SOFTWARE PROCESS QUALITY

ASSIGNMENT 1

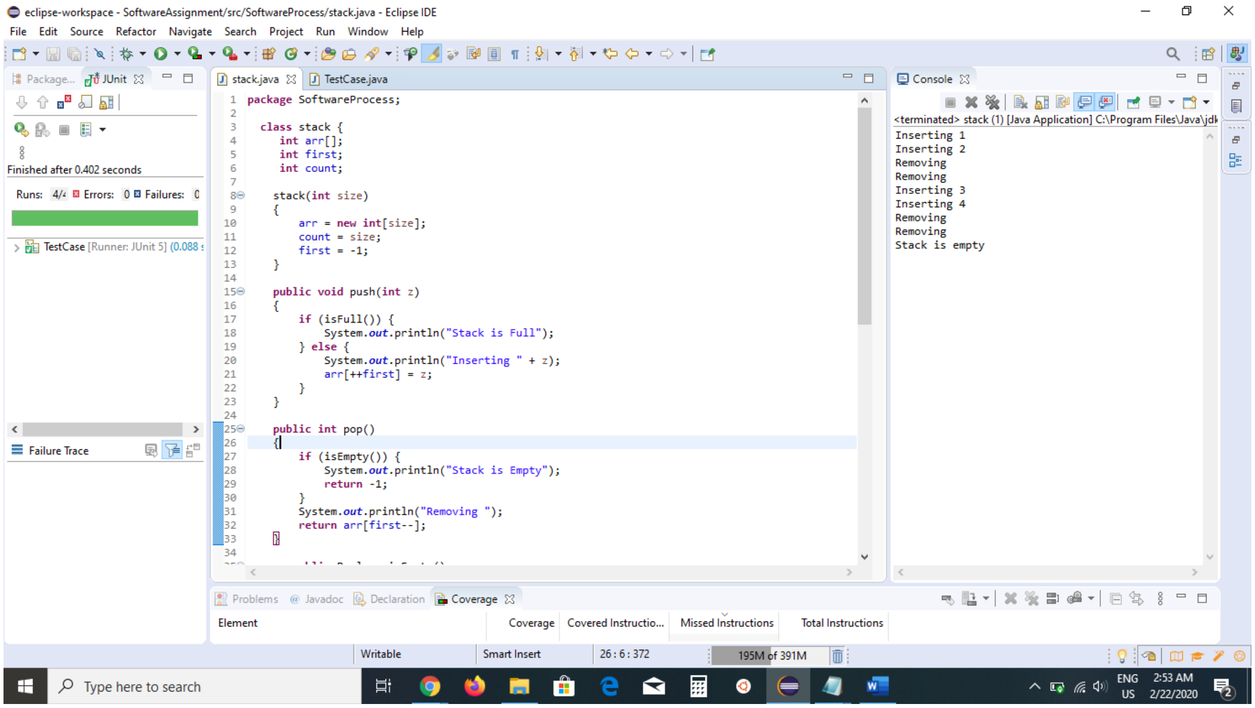
BY: ADITYA GUPTA
19210195

STACK CODE:

```
package SoftwareProcess;

  class stack {
        int arr[];
        int first;
        int count;

       stack(int size)
       {
             arr = new int[size];
             count = size;
             first = -1;
       }

       public void push(int z)
       {
             if (isFull()) {
                    System.out.println("Stack is Full");
             } else {
                    System.out.println("Inserting " + z);
                    arr[++first] = z;
             }
       }

       public int pop()
       {
             if (isEmpty()) {
                    System.out.println("Stack is Empty");
                    return -1;
             }
             System.out.println("Removing ");
             return arr[first--];
       }

          public Boolean isEmpty()
             {
                          return first == -1;
             }

             public Boolean isFull()
             {
                    return first == count - 1;
             }

       public static void main (String[] args)
```

```
        {
                stack stack = new stack(4);

                stack.push(1);
                stack.push(2);

                stack.pop();
                stack.pop();

                stack.push(3);
                stack.push(4);

                stack.pop();
                stack.pop();

                if (stack.isEmpty())
                        System.out.println("Stack is empty");
                else
                        System.out.println("Stack is not empty");
        }
}
```

Output of Stack Code:



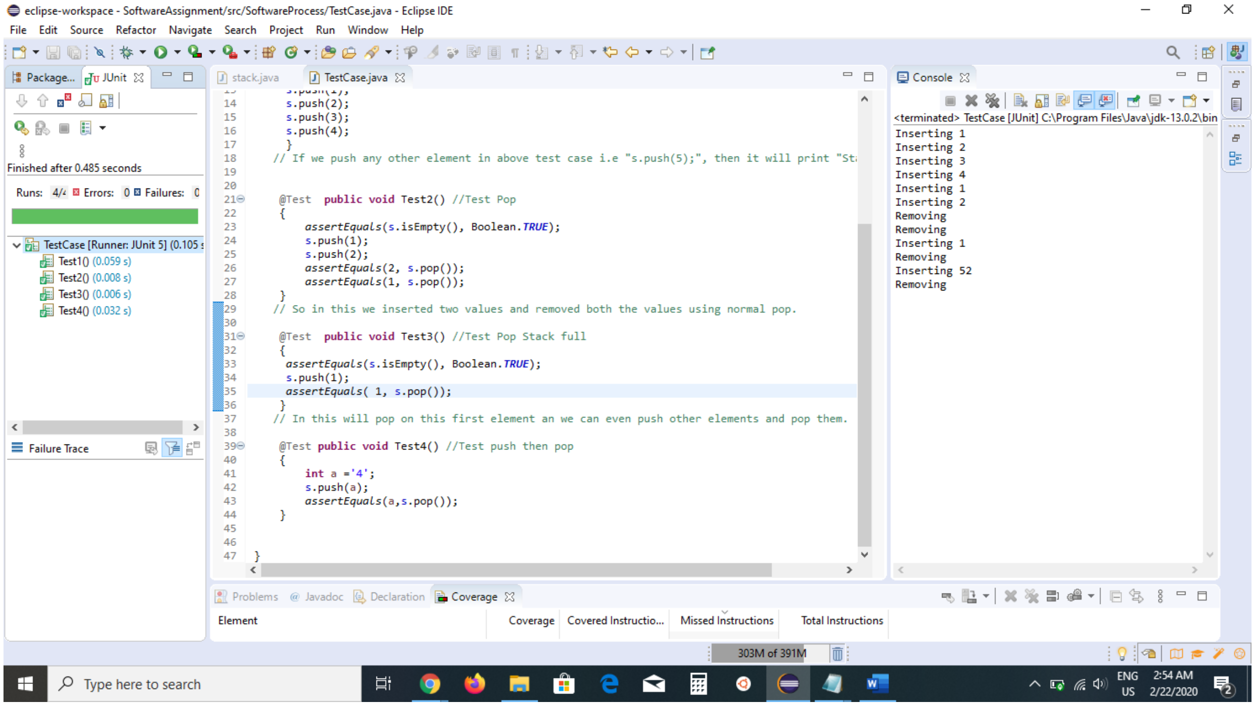
```
package SoftwareProcess;

  class stack {
     int arr[];
     int first;
     int count;

    stack(int size)
    {
        arr = new int[size];
        count = size;
        first = -1;
    }

    public void push(int z)
    {
        if (isFull()) {
            System.out.println("Stack is Full");
        } else {
            System.out.println("Inserting " + z);
            arr[++first] = z;
        }
    }

    public int pop()
    {
        if (isEmpty()) {
            System.out.println("Stack is Empty");
            return -1;
        }
        System.out.println("Removing ");
        return arr[first--];
    }
```

```
<terminated> stack (1) [Java Application] C:\Program Files\Java\jdk
Inserting 1
Inserting 2
Removing
Removing
Inserting 3
Inserting 4
Removing
Removing
Stack is empty
```

JUNIT CODE:

```
package SoftwareProcess;
import static org.junit.Assert.*;
import org.junit.jupiter.api.Test;

 public class TestCase {

       private static int stack_size = 4;
       stack s = new stack(stack_size);

    @Test  public void Test1() //Test Push and Push Stack Full
       {
        assertEquals(s.isEmpty(), Boolean.TRUE);
        s.push(1);
        s.push(2);
        s.push(3);
        s.push(4);
        }
    // If we push any other element in above test case i.e "s.push(5);", then it will
print "Stack is Full".


       @Test  public void Test2() //Test Pop
       {
              assertEquals(s.isEmpty(), Boolean.TRUE);
              s.push(1);
              s.push(2);
              assertEquals(2, s.pop());
              assertEquals(1, s.pop());
       }
       // So in this we inserted two values and removed both the values using normal
pop.

       @Test  public void Test3() //Test Pop Stack full
       {
        assertEquals(s.isEmpty(), Boolean.TRUE);
        s.push(1);
        assertEquals( 1, s.pop());
       }
       // In this will pop on this first element an we can even push other elements
and pop them.

       @Test public void Test4() //Test push then pop
       {
              int a ='4';
              s.push(a);
              assertEquals(a,s.pop());
       }


 }
```

Output of Test Cases:



```
Finished after 0.485 seconds
Runs: 4/4  Errors: 0  Failures: 0

TestCase [Runner: JUnit 5] (0.105 s
    Test1() (0.059 s)
    Test2() (0.008 s)
    Test3() (0.006 s)
    Test4() (0.032 s)
```

```java
        s.push(1);
        s.push(2);
        s.push(3);
        s.push(4);
        }
    // If we push any other element in above test case i.e "s.push(5);", then it will print "St


     @Test  public void Test2() //Test Pop
     {
         assertEquals(s.isEmpty(), Boolean.TRUE);
         s.push(1);
         s.push(2);
         assertEquals(2, s.pop());
         assertEquals(1, s.pop());
     }
    // So in this we inserted two values and removed both the values using normal pop.

     @Test  public void Test3() //Test Pop Stack full
     {
      assertEquals(s.isEmpty(), Boolean.TRUE);
      s.push(1);
      assertEquals( 1, s.pop());
     }
    // In this will pop on this first element an we can even push other elements and pop them.

     @Test public void Test4() //Test push then pop
     {
         int a ='4';
         s.push(a);
         assertEquals(a,s.pop());
     }


 }
```

```
<terminated> TestCase [JUnit] C:\Program Files\Java\jdk-13.0.2\bin
Inserting 1
Inserting 2
Inserting 3
Inserting 4
Inserting 1
Inserting 2
Removing
Removing
Inserting 1
Removing
Inserting 52
Removing
```

ERROR, FAULTS AND FAILURES:

- In (isFull) condition error was shown as it terminated the test code and the remaining test cases were not working and after that it was resolved.
- In Boolean isFull() condition return t element was not resolved.
- Failure was reported when an assertion doesn't have an expected value in any of the (assertsvalue).
- Junit 5 have different libraries to import as compared to Junit 4 otherwise it wasn’t working.

REFERENCES:

- https://www.techiedelight.com/stack-implementation-in-java/
- https://www.tutorialspoint.com/javaexamples/data_stack.htm
- https://cs.lmu.edu/~ray/notes/stacks/