SD2 - Worksheet 1 - 6%

GRIFFITH COLLEGE DUBLIN

Student name:	**Serianu Andrei-Silviu**					
Student number:	**3144757**					
Faculty:	**Computing Science**					
Course:	**BSCH/BSCO/EXCH**			**Stage/year:**	**2**	
Subject:	**Software Development 2**					
Study Mode:	Full time	☑		Part-time		
Lecturer Name:	**Gemma Deery**					
Assignment Title:	**Worksheet 1**					
Date due:	**19.02.2025**					
Date submitted:	**19.02.2025**					

Plagiarism disclaimer:

I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.

I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.

Signed: _____Serianu_____ **Date: ____19.02.2025_____**

Please note: **Students** MUST **retain a hard / soft copy of** ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

Please do not delete the questions.

For each question insert your answer below the question

Task 1

Part 1

```java
package griffith;

public class Grades {

    public int gradesMax(int[] grades) {
        return 0;
    }

    public int gradesTotal(int[] grades) {
        return 0;
    }

    public double gradesAverage(int[] grades) {
        return 0.0;
    }

    public int countFails(int[] grades, int minGrade) {
        return 0;
    }
}
```

```java
package griffith;

import static org.junit.jupiter.api.Assertions.*;

public class GradesTest {
    @Test
    public void testGradesMax() {
        // Placeholder test
    }

    @Test
    public void testGradesTotal() {
        // Placeholder test
    }

    @Test
    public void testGradesAverage() {
        // Placeholder test
    }

    @Test
    public void testCountFails() {
        // Placeholder test
}
}
```

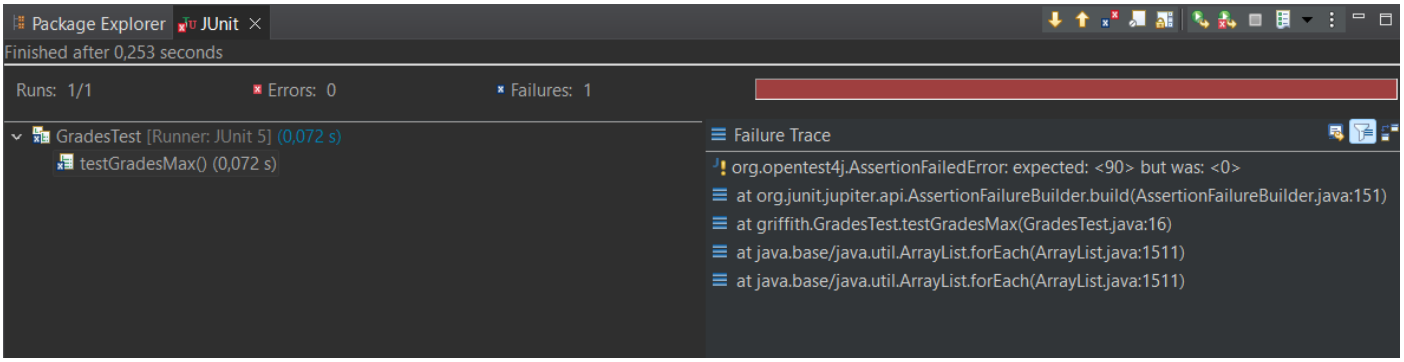
Part 2

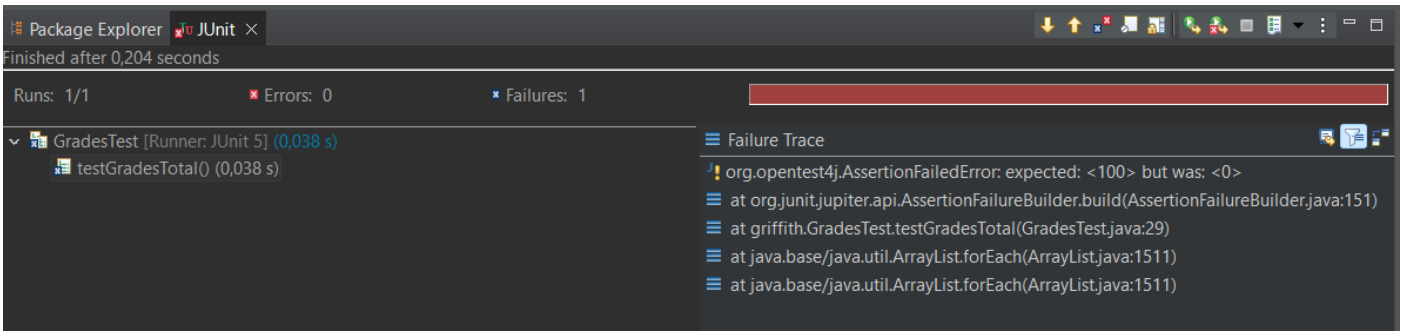
Grades.java *GradesTest.java

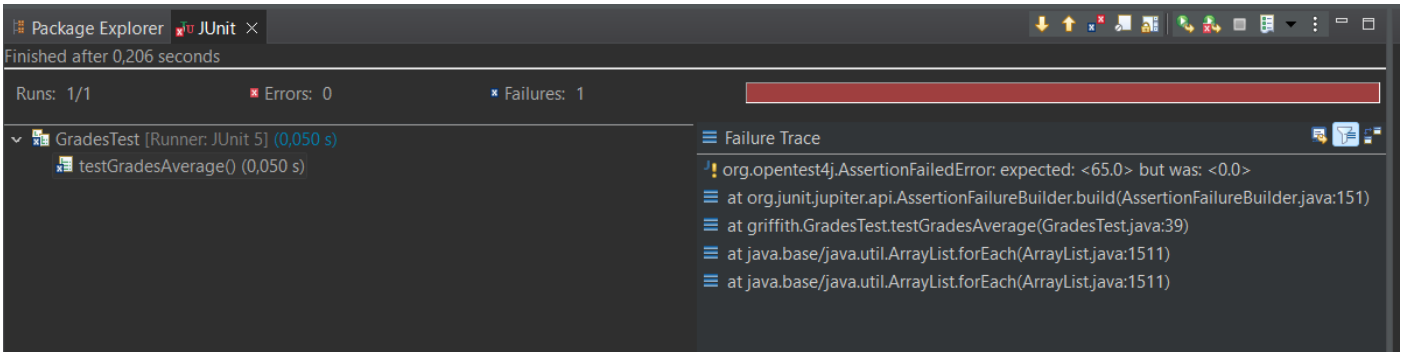
```java
package griffith;

import static org.junit.jupiter.api.Assertions.*;

public class GradesTest {
    @Test
    public void testGradesMax() {
        Grades g = new Grades();
        int[] grades = {45, 78, 90, 32, 88};
        assertEquals(90, g.gradesMax(grades));

        int[] negativeGrades = {-10, -5, -20, -3};
        assertEquals(-3, g.gradesMax(negativeGrades));

        int[] singleElement = {50};
        assertEquals(50, g.gradesMax(singleElement));
    }

    @Test
    public void testGradesTotal() {
        Grades g = new Grades();
        int[] grades = {10, 20, 30, 40};
        assertEquals(100, g.gradesTotal(grades));

        int[] emptyArray = {};
        assertEquals(0, g.gradesTotal(emptyArray));
    }

    @Test
```

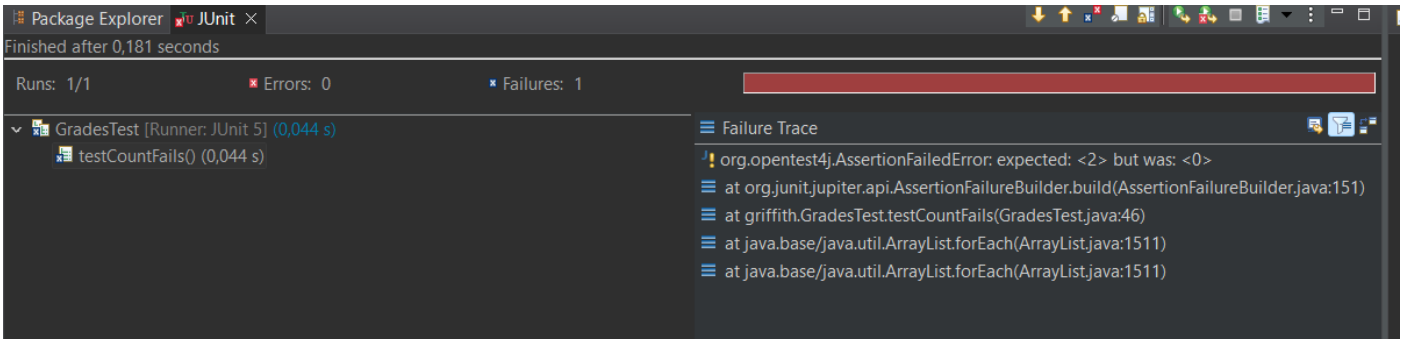
```java
public void testGradesAverage() {
    Grades g = new Grades();
    int[] grades = {50, 60, 70, 80};
    assertEquals(65.0, g.gradesAverage(grades), 0.01);
}

@Test
public void testCountFails() {
    Grades g = new Grades();
    int[] grades = {10, 20, 50, 70};
    assertEquals(2, g.countFails(grades,40));;}
```

The tests failing





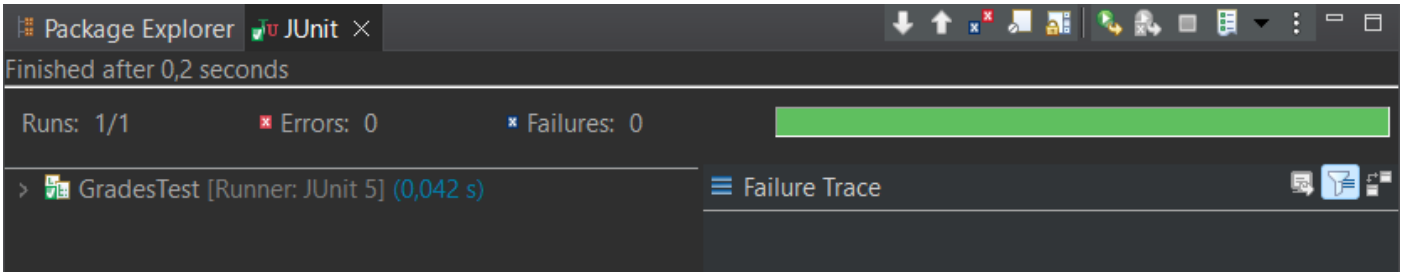




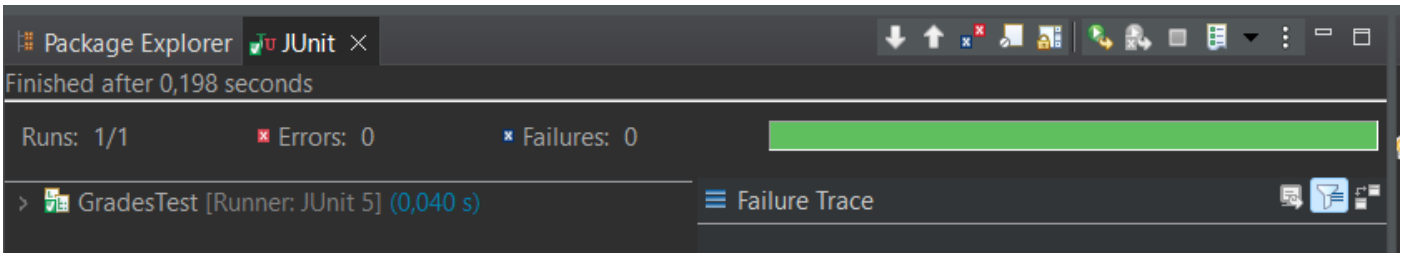
Part 3

Implementing Methods

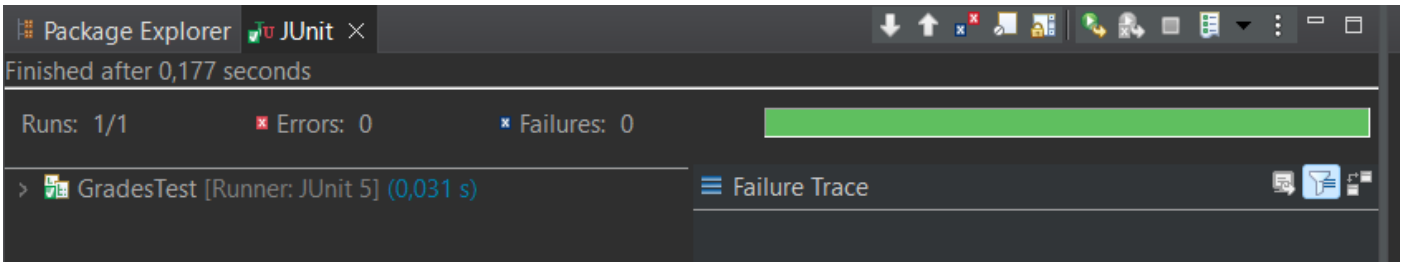
```java
public int gradesMax(int[] grades) {
    if (grades.length == 0) return Integer.MIN_VALUE;
    return Arrays.stream(grades).max().getAsInt();
}
```



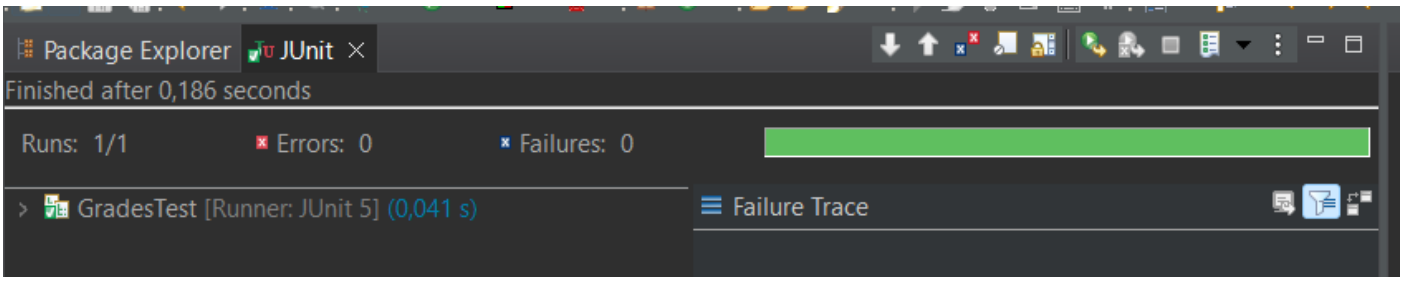

```java
public int gradesTotal(int[] grades) {
    return Arrays.stream(grades).sum();
}
```




```java
public double gradesAverage(int[] grades) {
    if (grades.length == 0) return 0.0;
    return (double) gradesTotal(grades) / grades.length;
}
```



```
    public int countFails(int[] grades, int minGrade) {
        return (int) Arrays.stream(grades).filter(g -> g < minGrade).count();}
}
```



Task 2

Part 1

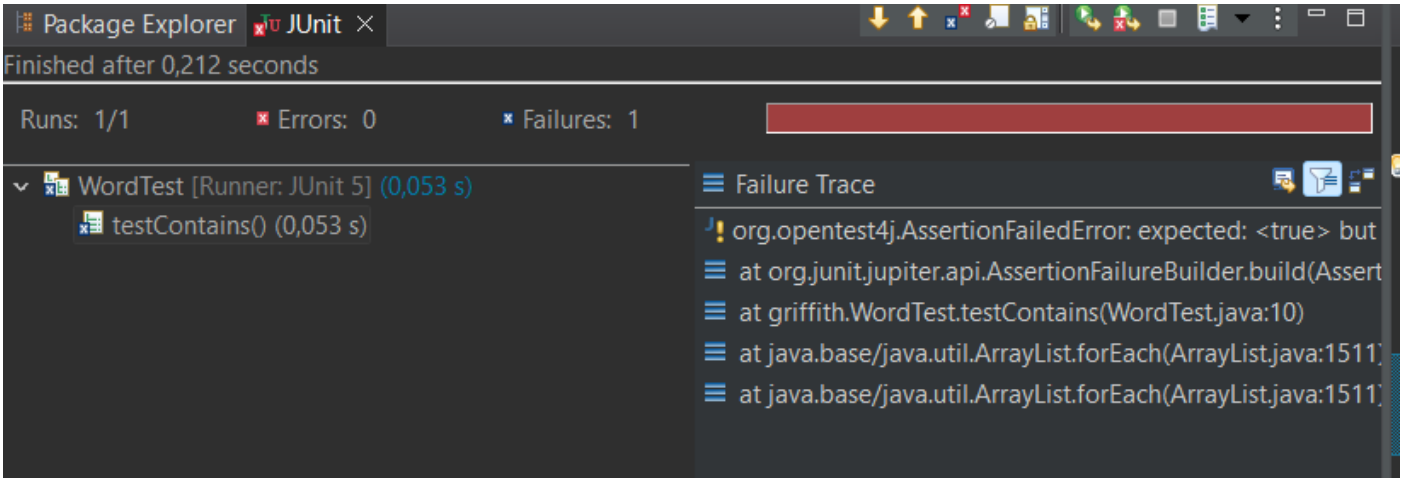
```
Grades.java  GradesTest.java  Word.java  Word
1  package griffith;
2
3  public class Word {
4      private char[] letters;
5
6      public Word(char[] letters) {
7          this.letters = letters;
8      }
9
10     public boolean contains(char symbol) {
11         return false;
12     }
13
14     public int length() {
15         return 0;
16     }
17
18     public char[] getLetters() {
19         return null;}
20 }
```

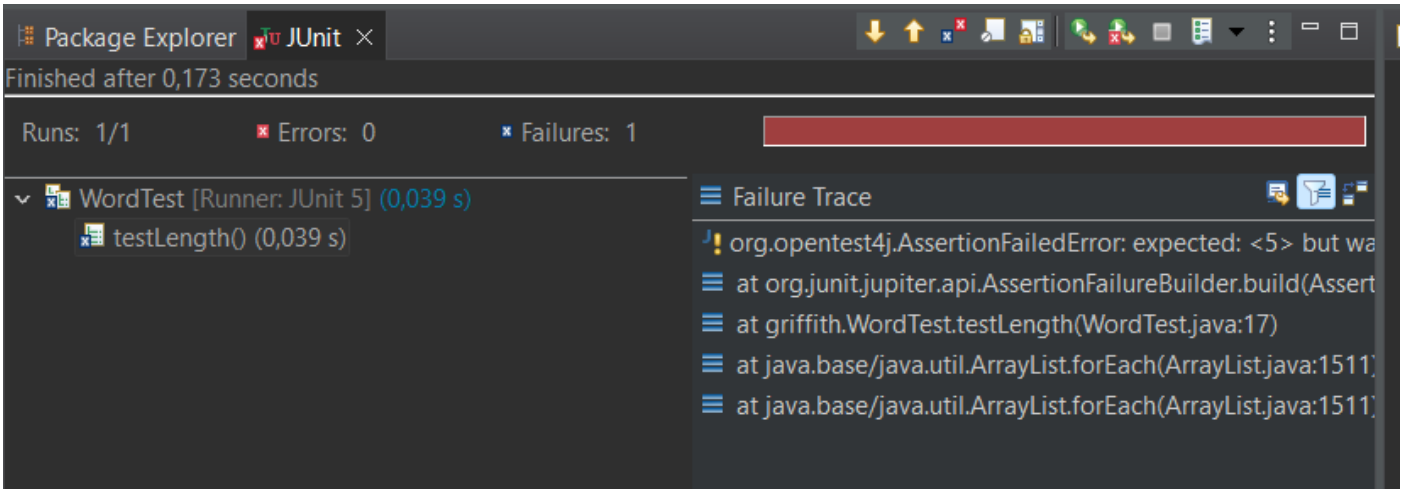
```java
package griffith;

import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class WordTest {
    @Test
    public void testContains() {

    }

    @Test
    public void testLength() {

    }

    @Test
    public void testNotNull() {

}
}
```

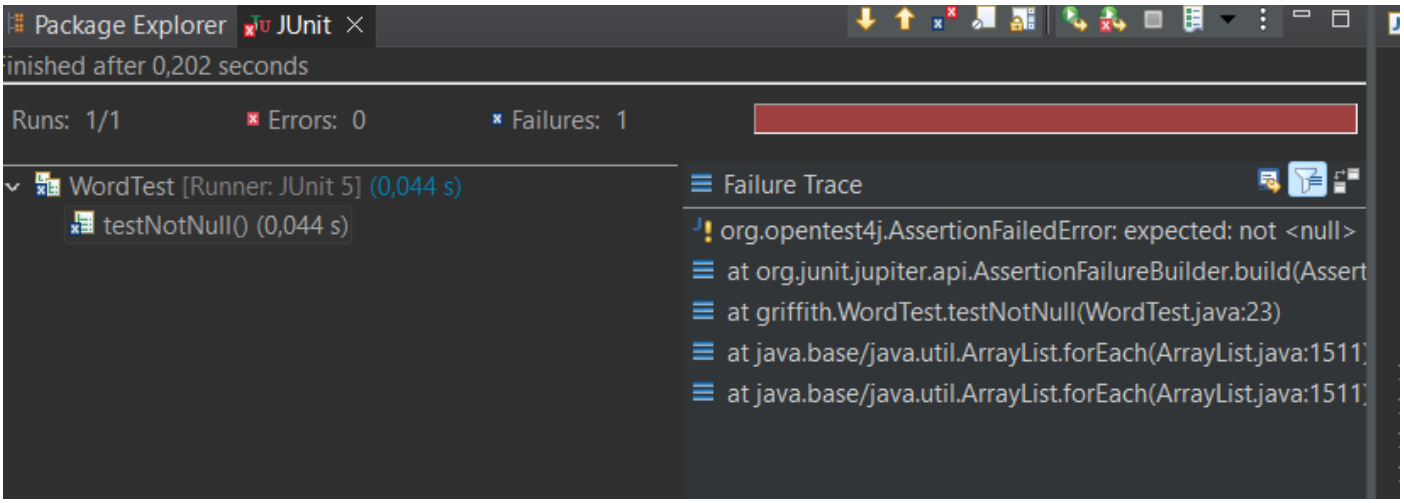
Part 2

```java
package griffith;

public class Word {
    private char[] letters;

    public Word(char[] letters) {
        this.letters = letters;
    }

    public boolean contains(char symbol) {
        return false;
    }

    public int length() {
        return 0;
    }

    public char[] getLetters() {
        return null;}
}
```

The tests failing





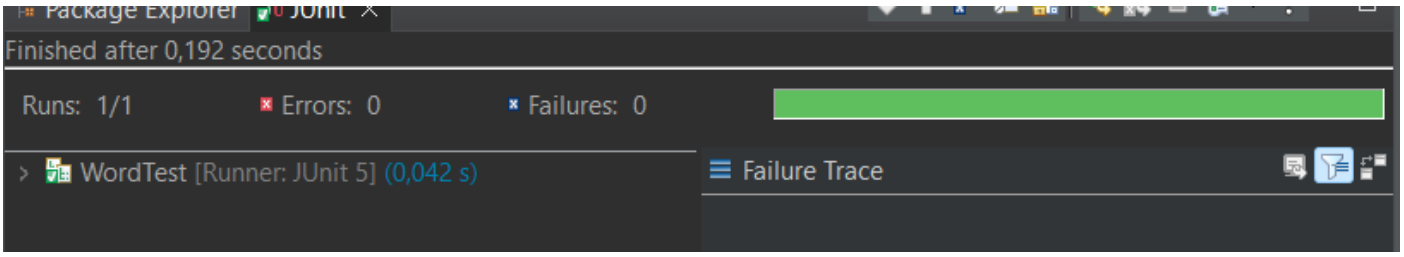


Part 3

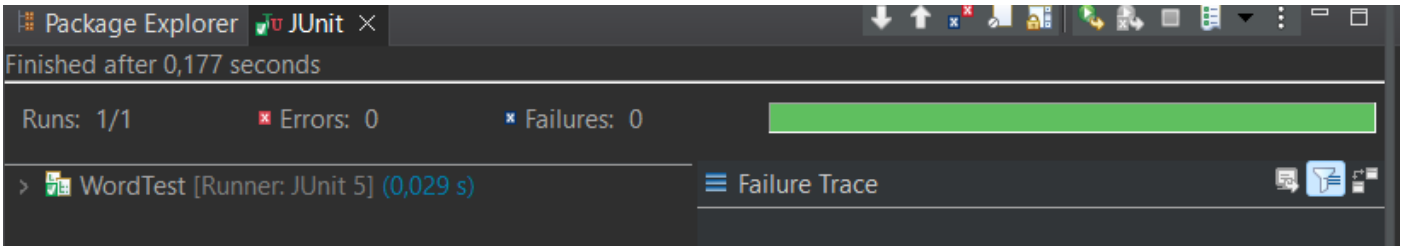
Implementing Methods

SD2 - Worksheet 1 - 6%

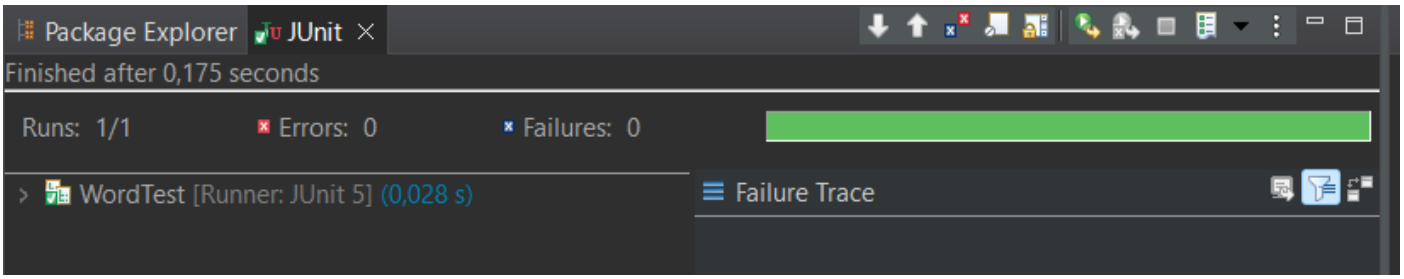
```java
public Word(char[] letters) {
    this.letters = (letters != null) ? letters : new char[]{};
}

public boolean contains(char symbol) {
    for (char c : letters) {
        if (c == symbol) return true;
    }
    return false;
}
```




```java
public int length() {
    return letters.length;
}
```




```java
    public char[] getLetters() {
        return letters;
    }
}
```



GitHub Link : https://github.com/Silviu-Sri/SD2Assigments.git