

SD2 - Worksheet 1 - 6%



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	<input checked="" type="checkbox"/>		Part-time	<input type="checkbox"/>	
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

Serianu Andrei-Silviu 3144757

Task 1

Part 1

```
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;
    }
}
```

```
1 package griffith;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7
8
9
10
11 public class GradesTest {
12     @Test
13     public void testGradesMax() {
14         // Placeholder test
15     }
16
17     @Test
18     public void testGradesTotal() {
19         // Placeholder test
20     }
21
22     @Test
23     public void testGradesAverage() {
24         // Placeholder test
25     }
26
27     @Test
28     public void testCountFails() {
29         // Placeholder test
30     }
31 }
```

SD2 - Worksheet 1 - 6%

Part 2

```
Grades.java *GradesTest.java X
1 package griffith;
2
3 import static org.junit.jupiter.api.Assertions.*;
10
11 public class GradesTest {
12     @Test
13     public void testGradesMax() {
14         Grades g = new Grades();
15         int[] grades = {45, 78, 90, 32, 88};
16         assertEquals(90, g.gradesMax(grades));
17
18         int[] negativeGrades = {-10, -5, -20, -3};
19         assertEquals(-3, g.gradesMax(negativeGrades));
20
21         int[] singleElement = {50};
22         assertEquals(50, g.gradesMax(singleElement));
23     }
24
25     @Test
26     public void testGradesTotal() {
27         Grades g = new Grades();
28         int[] grades = {10, 20, 30, 40};
29         assertEquals(100, g.gradesTotal(grades));
30
31         int[] emptyArray = {};
32         assertEquals(0, g.gradesTotal(emptyArray));
33     }
34
35     @Test
36     public void testGradesAverage() {
37         Grades g = new Grades();
38         int[] grades = {50, 60, 70, 80};
39         assertEquals(65.0, g.gradesAverage(grades), 0.01);
40     }
41
42     @Test
43     public void testCountFails() {
44         Grades g = new Grades();
45         int[] grades = {10, 20, 50, 70};
46         assertEquals(2, g.countFails(grades, 40));
47     }
48 }
```

SD2 - Worksheet 1 - 6%

The tests failing

```
Package Explorer JUnit x
Finished after 0,253 seconds

Runs: 1/1 Errors: 0 Failures: 1

GradesTest [Runner: JUnit 5] (0,072 s)
  testGradesMax() (0,072 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <90> but was: <0>
at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
at griffith.GradesTest.testGradesMax(GradesTest.java:16)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

```
Package Explorer JUnit x
Finished after 0,204 seconds

Runs: 1/1 Errors: 0 Failures: 1

GradesTest [Runner: JUnit 5] (0,038 s)
  testGradesTotal() (0,038 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <100> but was: <0>
at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
at griffith.GradesTest.testGradesTotal(GradesTest.java:29)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

```
Package Explorer JUnit x
Finished after 0,206 seconds

Runs: 1/1 Errors: 0 Failures: 1

GradesTest [Runner: JUnit 5] (0,050 s)
  testGradesAverage() (0,050 s)

Failure Trace
org.opentest4j.AssertionFailedError: expected: <65.0> but was: <0.0>
at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
at griffith.GradesTest.testGradesAverage(GradesTest.java:39)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

```
Package Explorer JUnit x
Finished after 0,181 seconds

Runs: 1/1 Errors: 0 Failures: 1

GradesTest [Runner: JUnit 5] (0,044 s)
  testCountFails() (0,044 s)

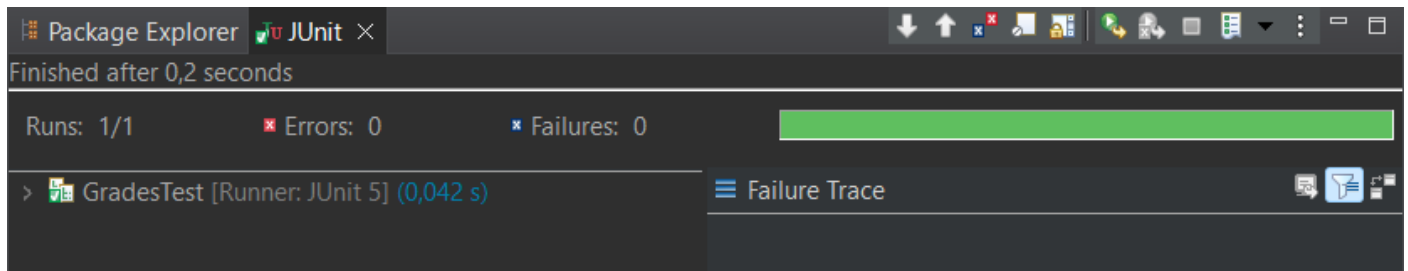
Failure Trace
org.opentest4j.AssertionFailedError: expected: <2> but was: <0>
at org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
at griffith.GradesTest.testCountFails(GradesTest.java:46)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```

SD2 - Worksheet 1 - 6%

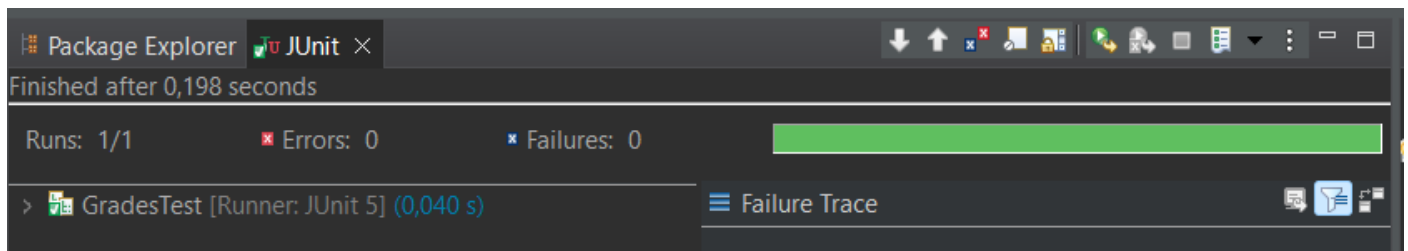
Part 3

Implementing Methods

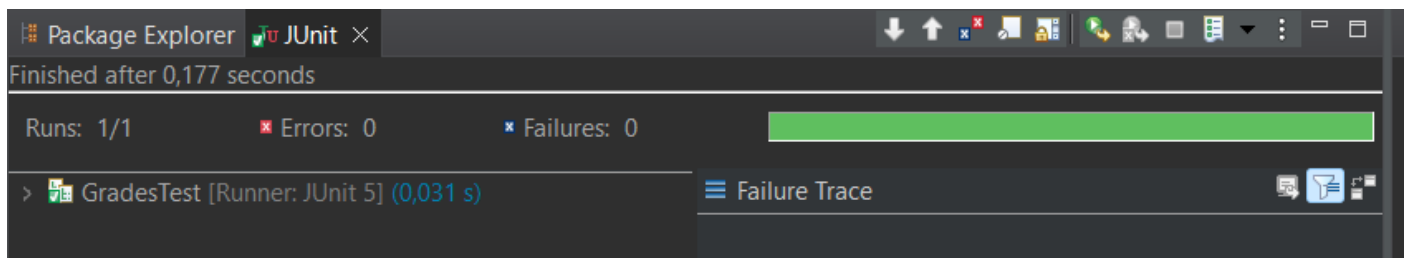
```
public int gradesMax(int[] grades) {  
    if (grades.length == 0) return Integer.MIN_VALUE;  
    return Arrays.stream(grades).max().getAsInt();  
}
```



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

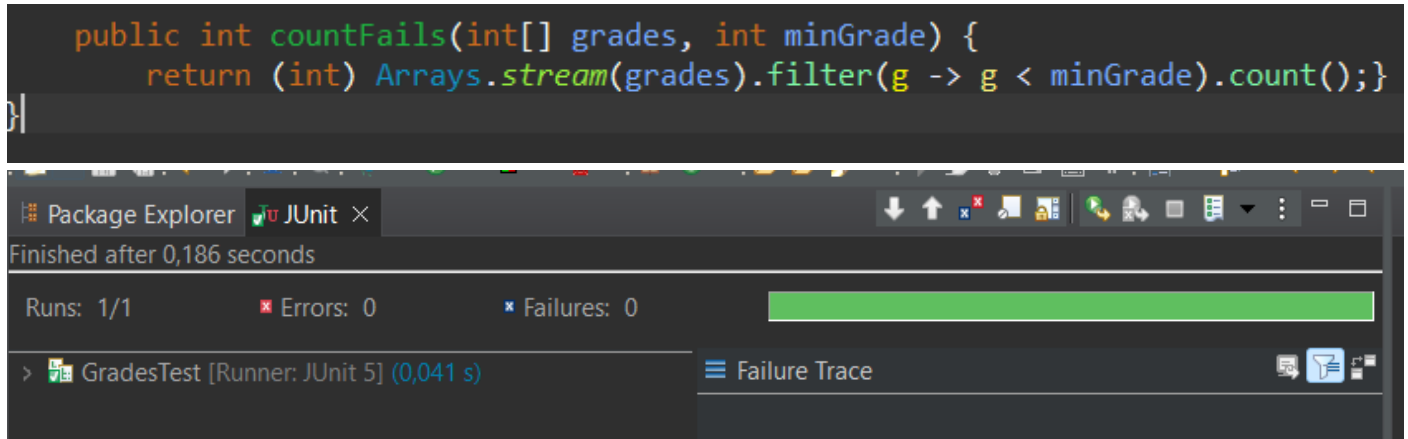


```
public double gradesAverage(int[] grades) {  
    if (grades.length == 0) return 0.0;  
    return (double) gradesTotal(grades) / grades.length;  
}
```



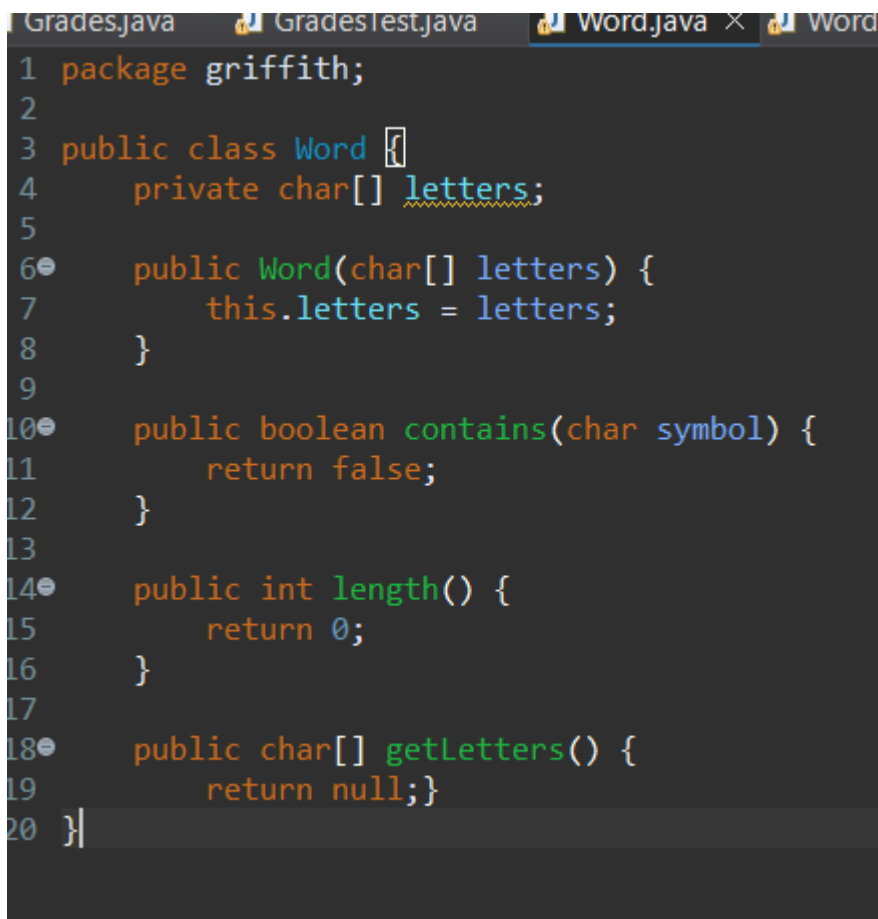
SD2 - Worksheet 1 - 6%

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



Task 2

Part 1



```
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    }  
}
```

SD2 - Worksheet 1 - 6%

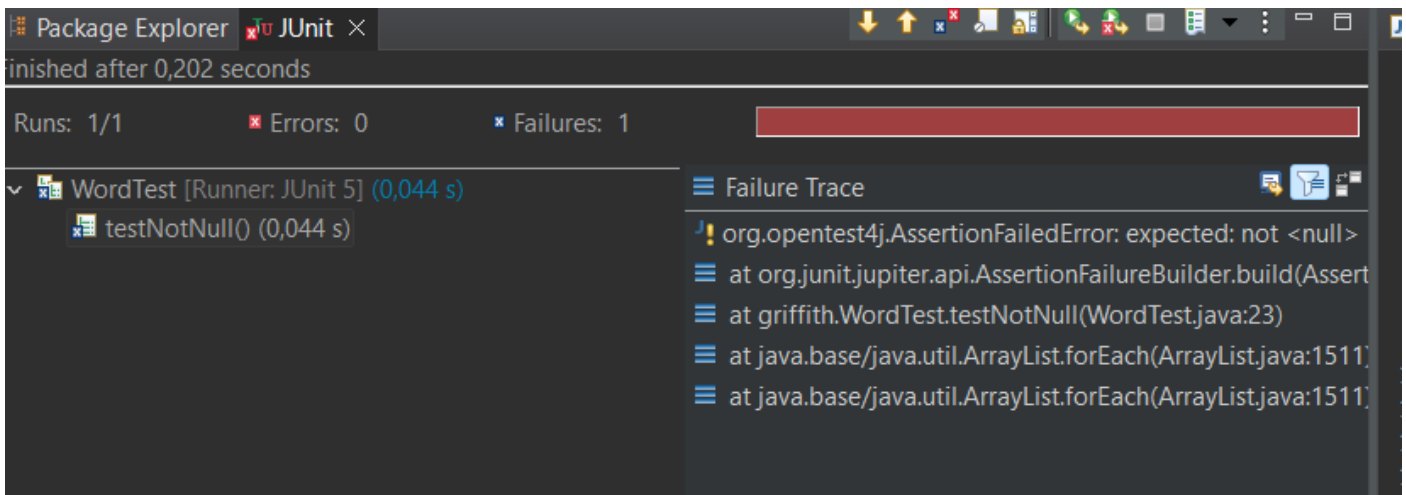
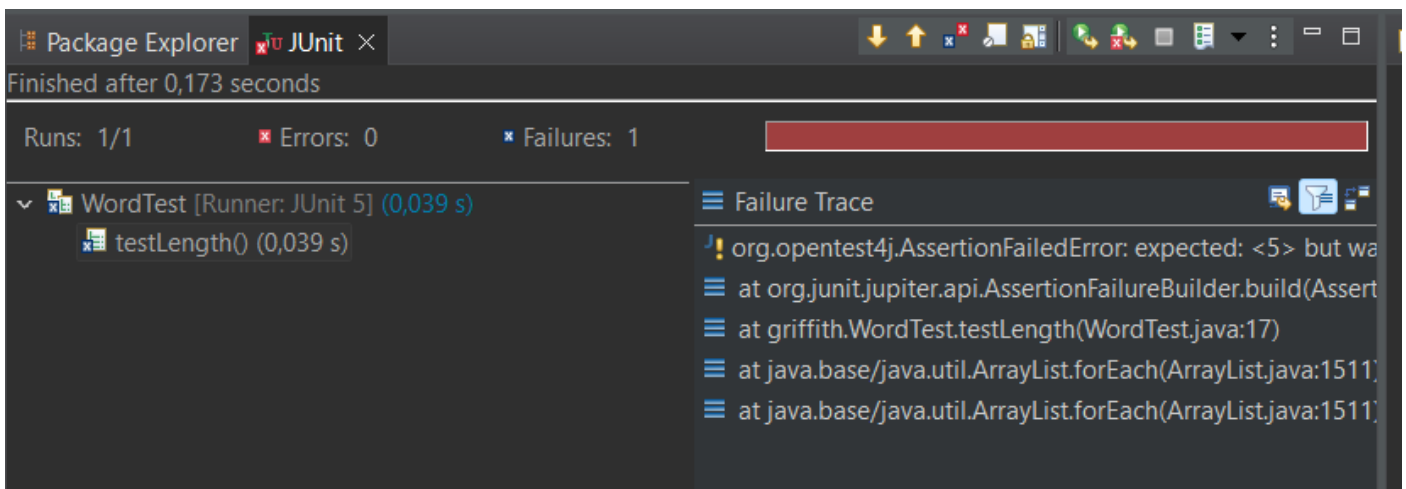
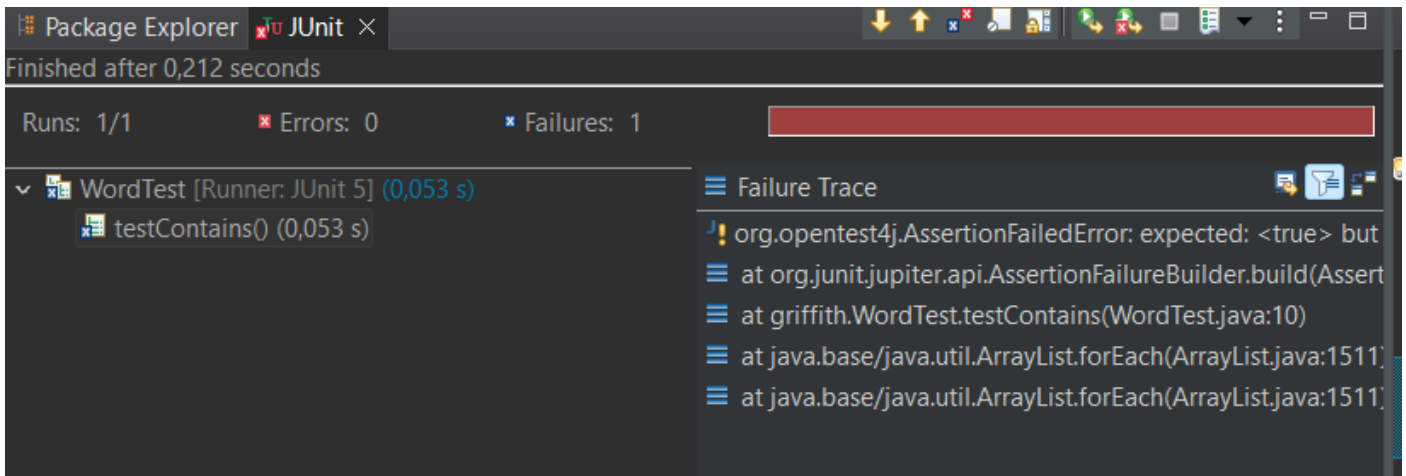
```
1 package griffith;
2
3 import static org.junit.jupiter.api.Assertions.*;
4 import org.junit.jupiter.api.Test;
5
6 public class WordTest {
7     @Test
8     public void testContains() {
9
10    }
11
12     @Test
13     public void testLength() {
14
15    }
16
17     @Test
18     public void testNotNull() {
19
20    }
21 }
```

Part 2

```
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    }
21 }
```

SD2 - Worksheet 1 - 6%

The tests failing



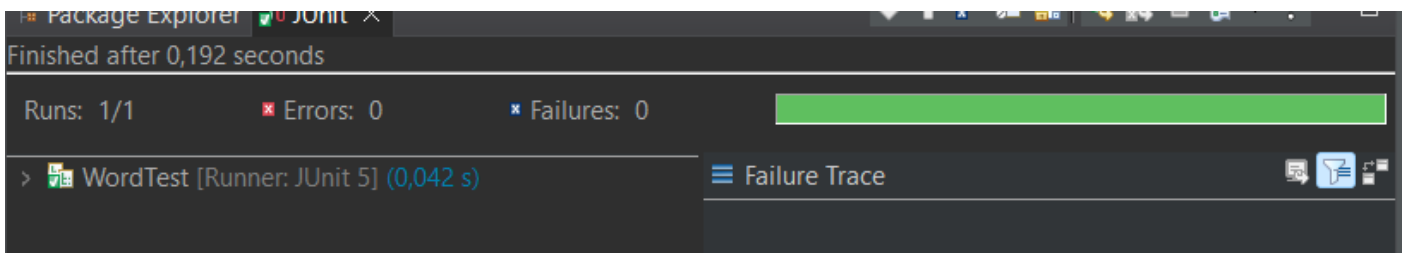
Part 3

Implementing Methods

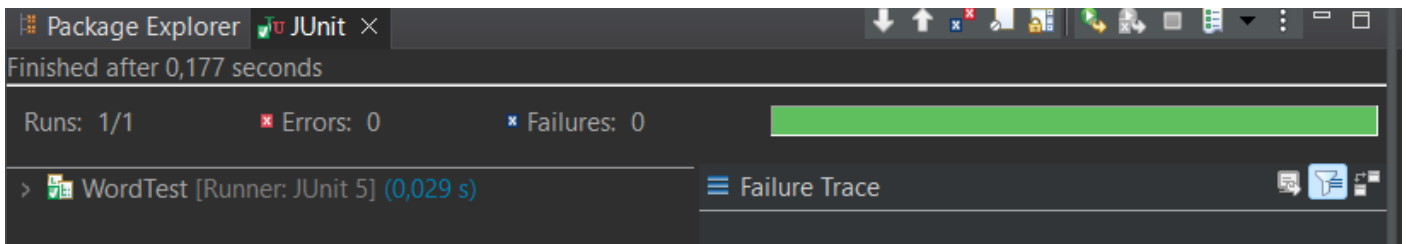
SD2 - Worksheet 1 - 6%

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

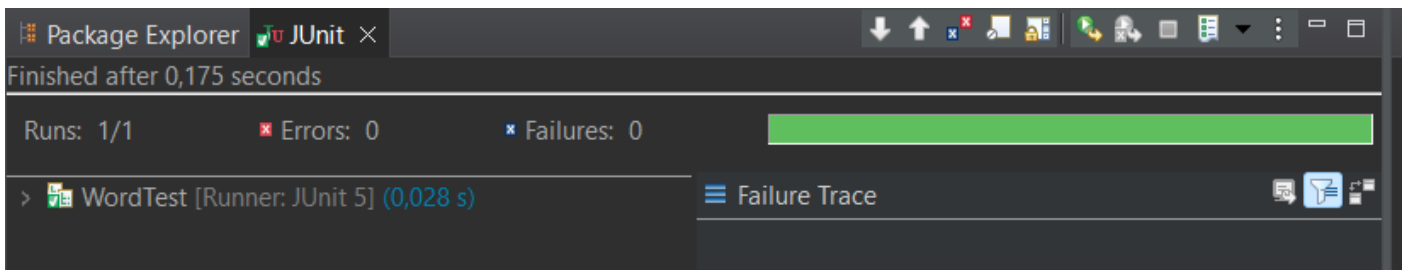
public boolean contains(char symbol) {
    for (char c : letters) {
        if (c == symbol) return true;
    }
    return false;
}
```



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



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



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