SD2 - Worksheet 1 - 6%

GRIFFITH COLLEGE DUBLIN

Student name:	**Enkhbaatar Idersaikhan**					
Student number:	**3112121**					
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:						
Date submitted:						

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: ____________________ **Date:** ____________________

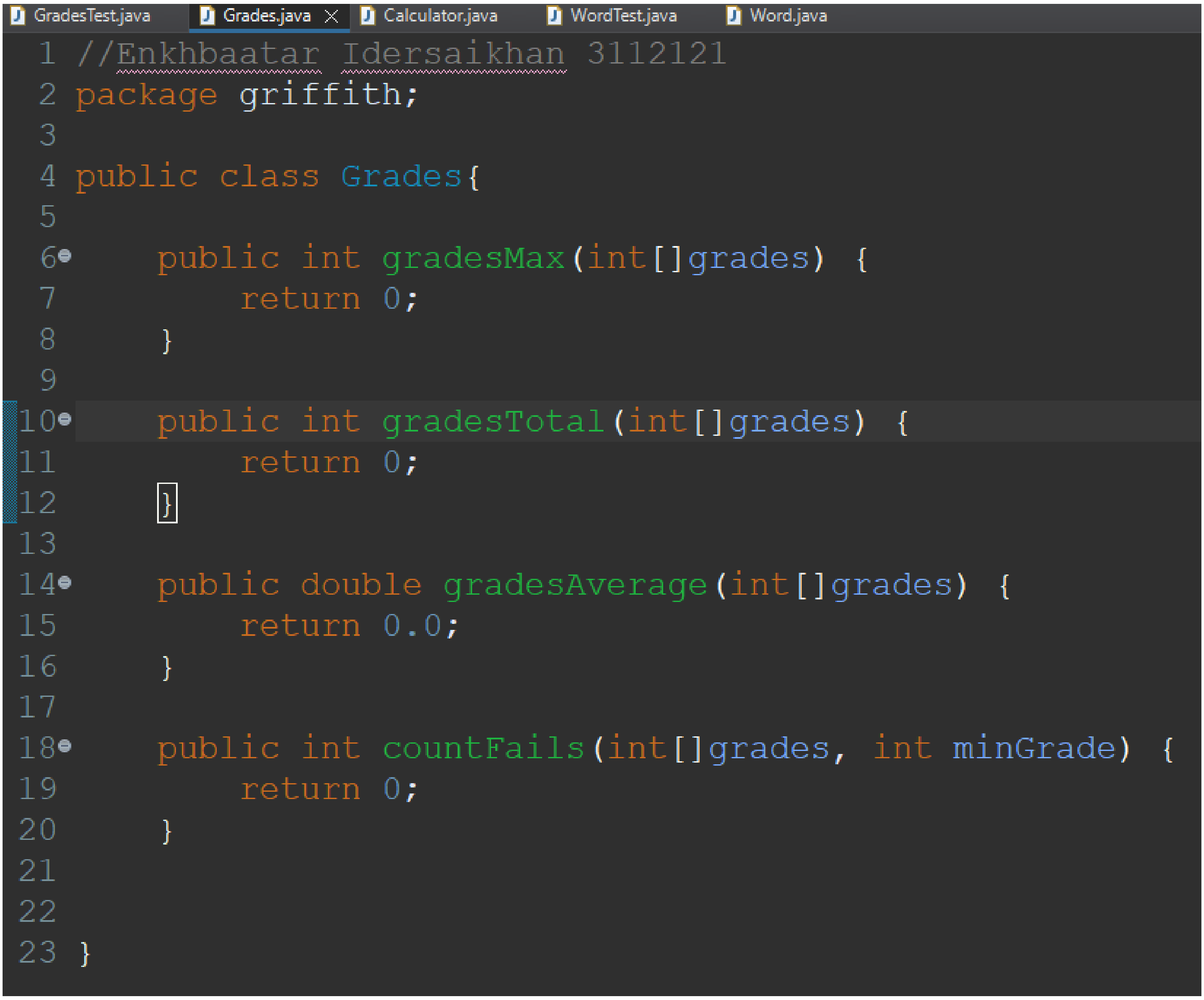
Please note: **Students** MUST **retain a hard / soft copy of** ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

Please do not delete the questions.

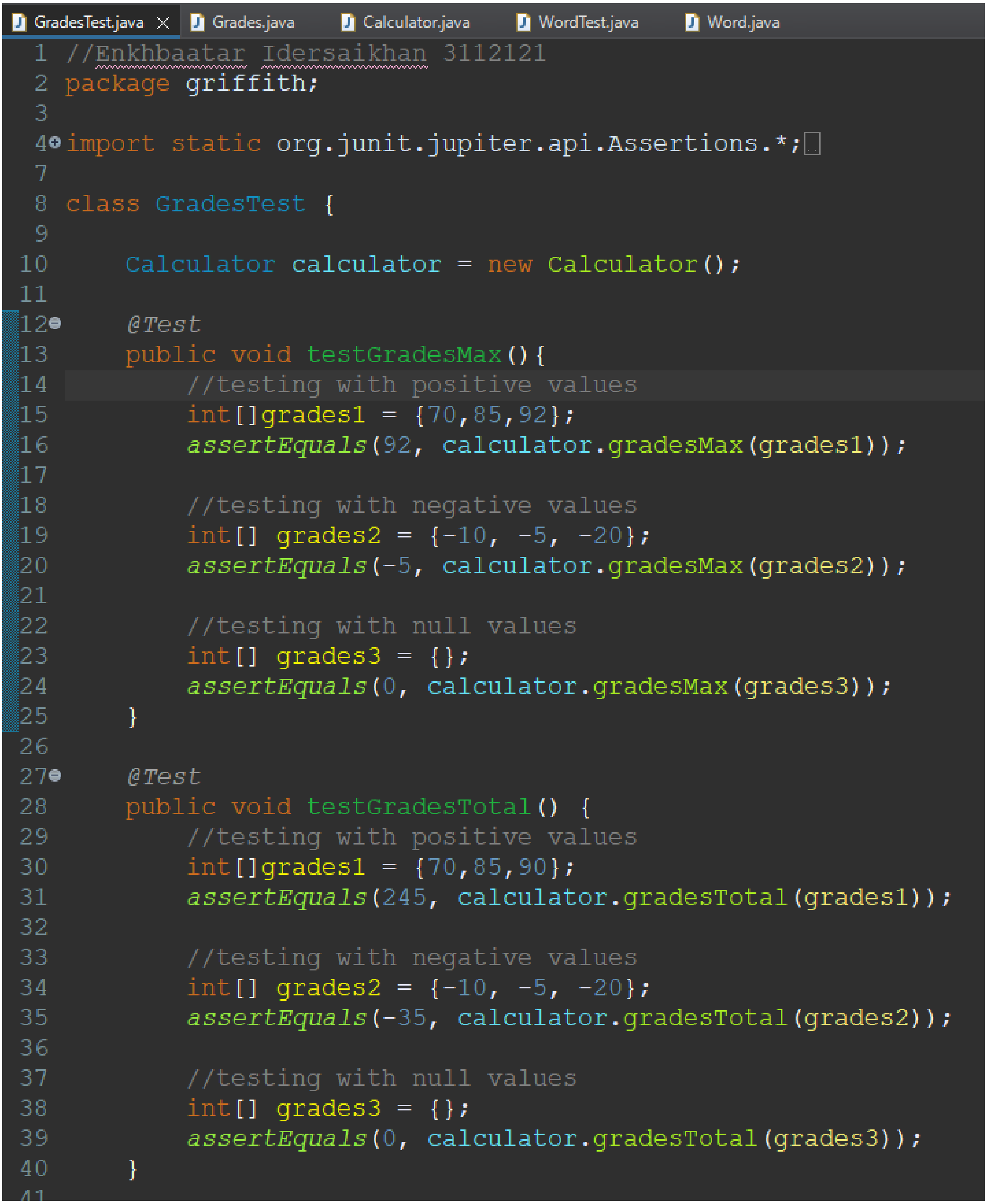
For each question insert your answer below the question

Tasks: 1

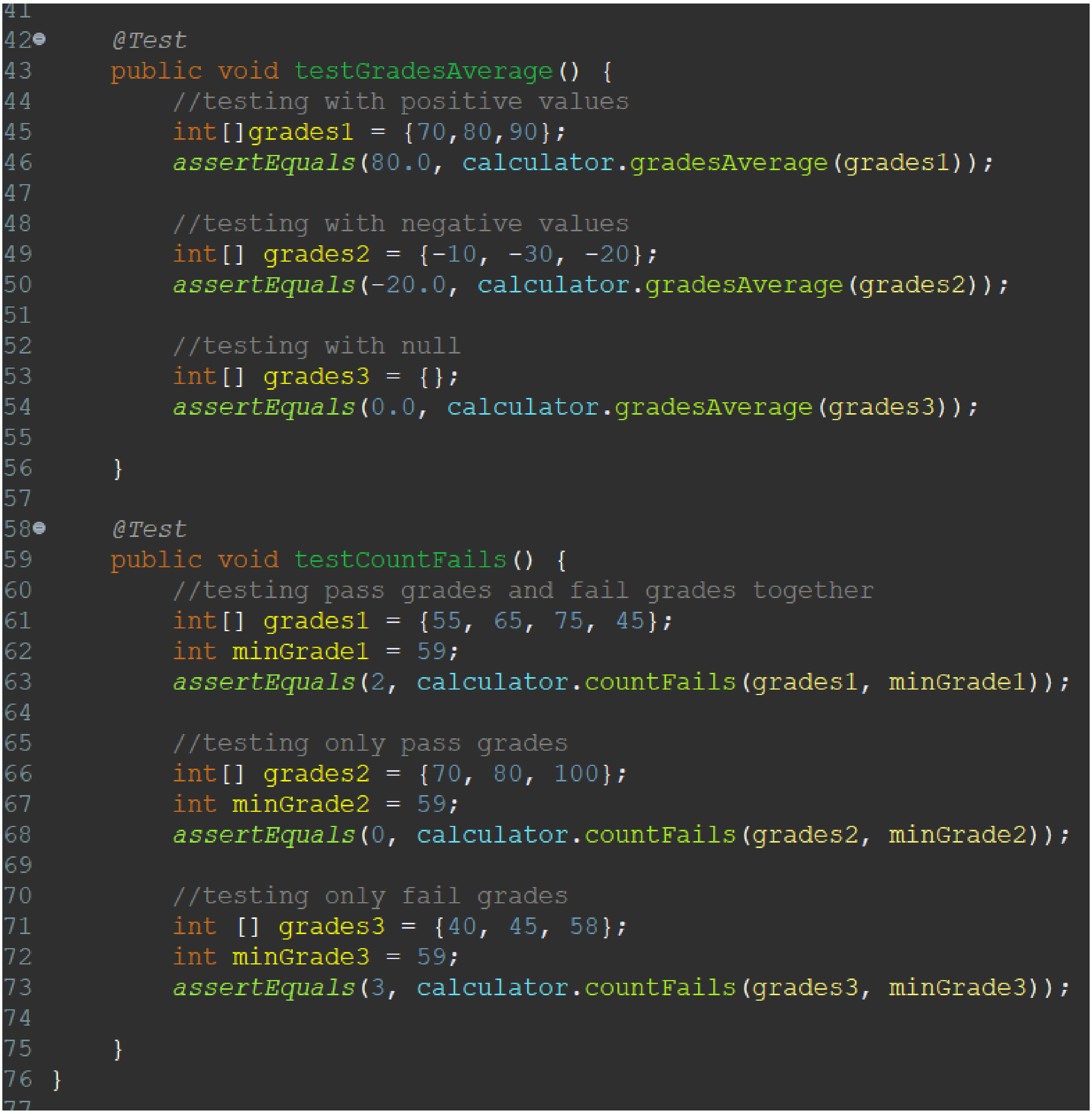
Part One



```java
//Enkhbaatar Idersaikhan 3112121
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;
    }


}
```

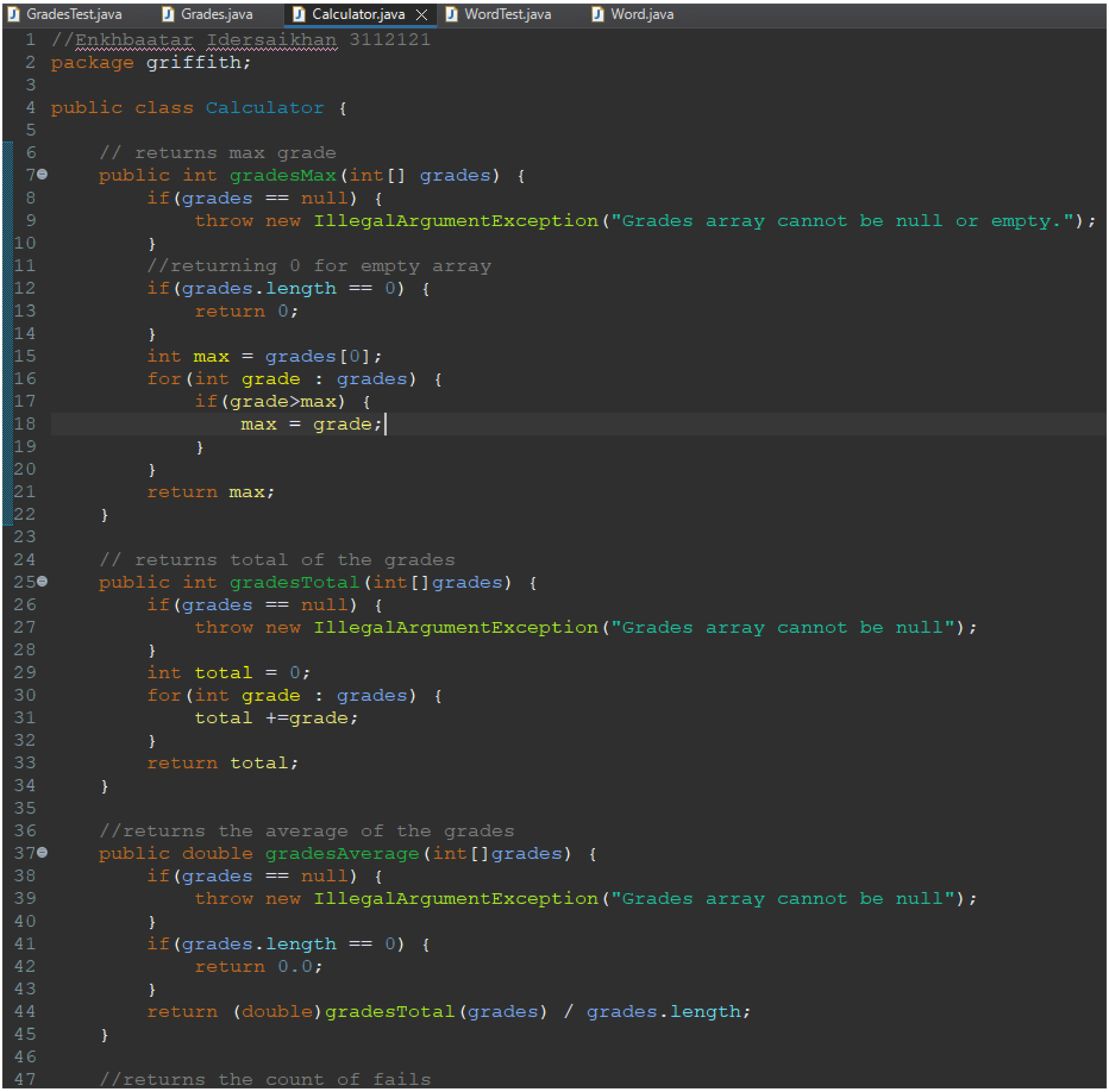


```java
//Enkhbaatar Idersaikhan 3112121
package griffith;

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

class GradesTest {

    Calculator calculator = new Calculator();

    @Test
    public void testGradesMax(){
        //testing with positive values
        int[]grades1 = {70,85,92};
        assertEquals(92, calculator.gradesMax(grades1));

        //testing with negative values
        int[] grades2 = {-10, -5, -20};
        assertEquals(-5, calculator.gradesMax(grades2));

        //testing with null values
        int[] grades3 = {};
        assertEquals(0, calculator.gradesMax(grades3));
    }

    @Test
    public void testGradesTotal() {
        //testing with positive values
        int[]grades1 = {70,85,90};
        assertEquals(245, calculator.gradesTotal(grades1));

        //testing with negative values
        int[] grades2 = {-10, -5, -20};
        assertEquals(-35, calculator.gradesTotal(grades2));

        //testing with null values
        int[] grades3 = {};
        assertEquals(0, calculator.gradesTotal(grades3));
    }
```

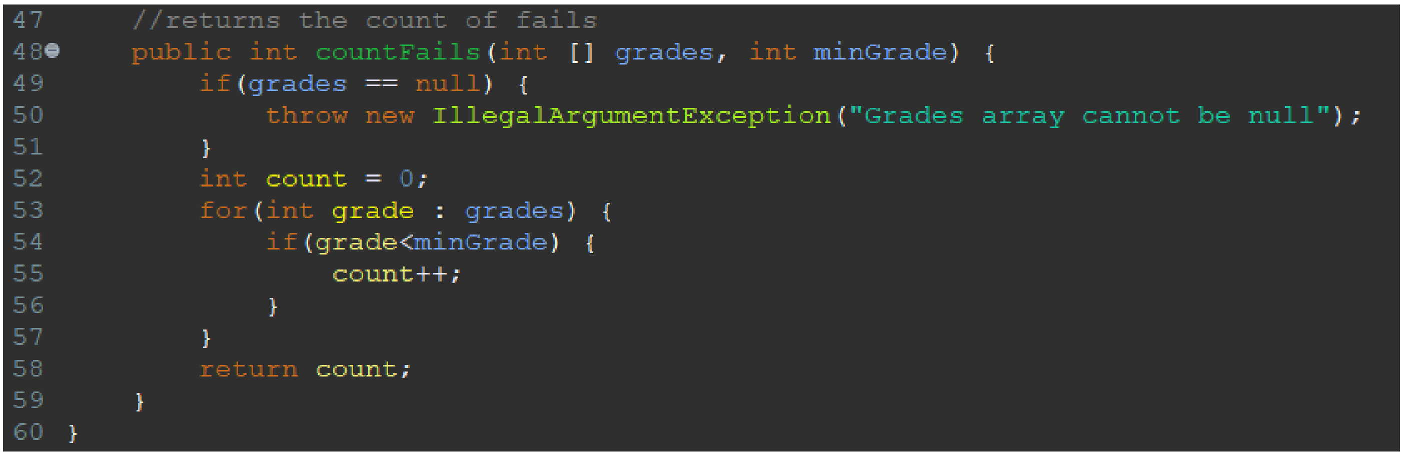


```
    @Test
    public void testGradesAverage() {
        //testing with positive values
        int[]grades1 = {70,80,90};
        assertEquals(80.0, calculator.gradesAverage(grades1));

        //testing with negative values
        int[] grades2 = {-10, -30, -20};
        assertEquals(-20.0, calculator.gradesAverage(grades2));

        //testing with null
        int[] grades3 = {};
        assertEquals(0.0, calculator.gradesAverage(grades3));

    }

    @Test
    public void testCountFails() {
        //testing pass grades and fail grades together
        int[] grades1 = {55, 65, 75, 45};
        int minGrade1 = 59;
        assertEquals(2, calculator.countFails(grades1, minGrade1));

        //testing only pass grades
        int[] grades2 = {70, 80, 100};
        int minGrade2 = 59;
        assertEquals(0, calculator.countFails(grades2, minGrade2));

        //testing only fail grades
        int [] grades3 = {40, 45, 58};
        int minGrade3 = 59;
        assertEquals(3, calculator.countFails(grades3, minGrade3));

    }
}
```

Part 3

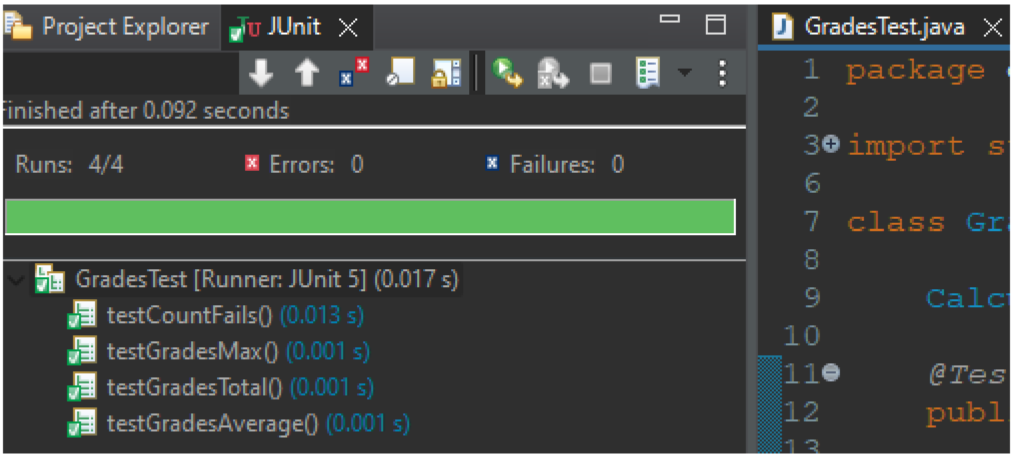


```java
//Enkhbaatar Idersaikhan 3112121
package griffith;

public class Calculator {

    // returns max grade
    public int gradesMax(int[] grades) {
        if(grades == null) {
            throw new IllegalArgumentException("Grades array cannot be null or empty.");
        }
        //returning 0 for empty array
        if(grades.length == 0) {
            return 0;
        }
        int max = grades[0];
        for(int grade : grades) {
            if(grade>max) {
                max = grade;
            }
        }
        return max;
    }

    // returns total of the grades
    public int gradesTotal(int[]grades) {
        if(grades == null) {
            throw new IllegalArgumentException("Grades array cannot be null");
        }
        int total = 0;
        for(int grade : grades) {
            total +=grade;
        }
        return total;
    }

    //returns the average of the grades
    public double gradesAverage(int[]grades) {
        if(grades == null) {
            throw new IllegalArgumentException("Grades array cannot be null");
        }
        if(grades.length == 0) {
            return 0.0;
        }
        return (double)gradesTotal(grades) / grades.length;
    }

    //returns the count of fails
```



```java
    //returns the count of fails
    public int countFails(int [] grades, int minGrade) {
        if(grades == null) {
            throw new IllegalArgumentException("Grades array cannot be null");
        }
        int count = 0;
        for(int grade : grades) {
            if(grade<minGrade) {
                count++;
            }
        }
        return count;
    }
}
```

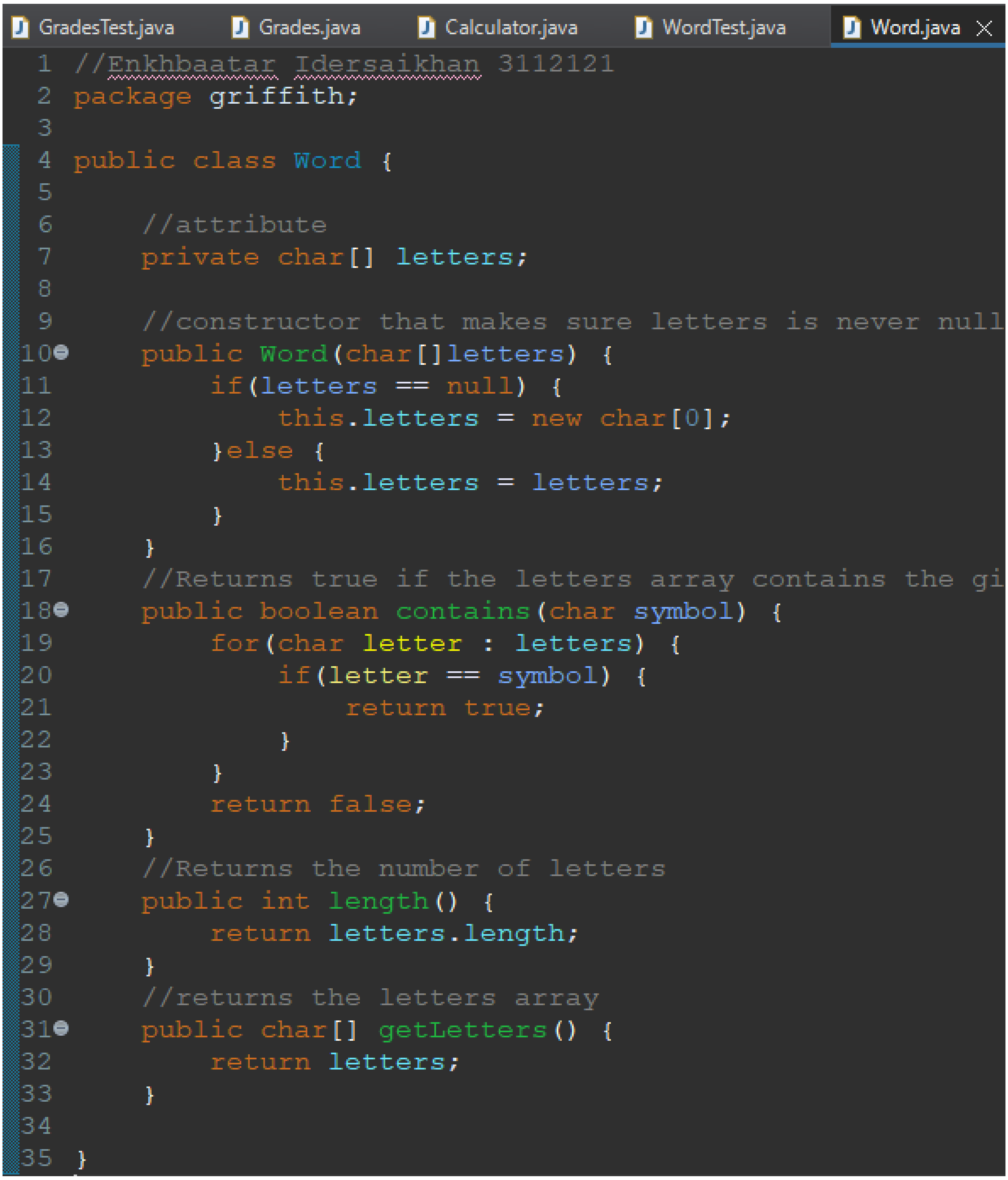
TEST



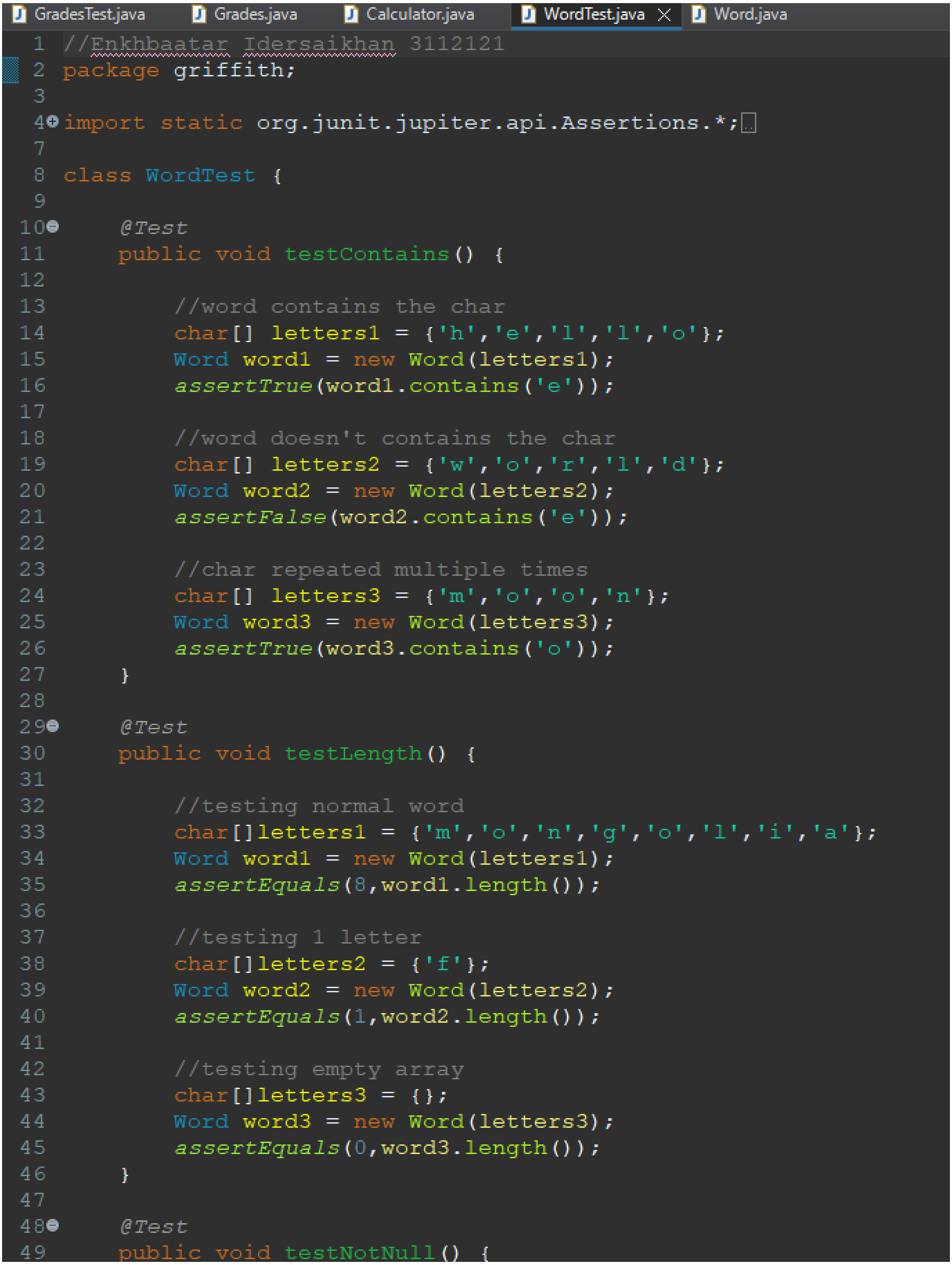
```
inished after 0.092 seconds

Runs:  4/4        Errors:  0        Failures:  0

GradesTest [Runner: JUnit 5] (0.017 s)
    testCountFails() (0.013 s)
    testGradesMax() (0.001 s)
    testGradesTotal() (0.001 s)
    testGradesAverage() (0.001 s)
```

```
package 

import s

class Gr

    Calc

    @Tes
    publ
```

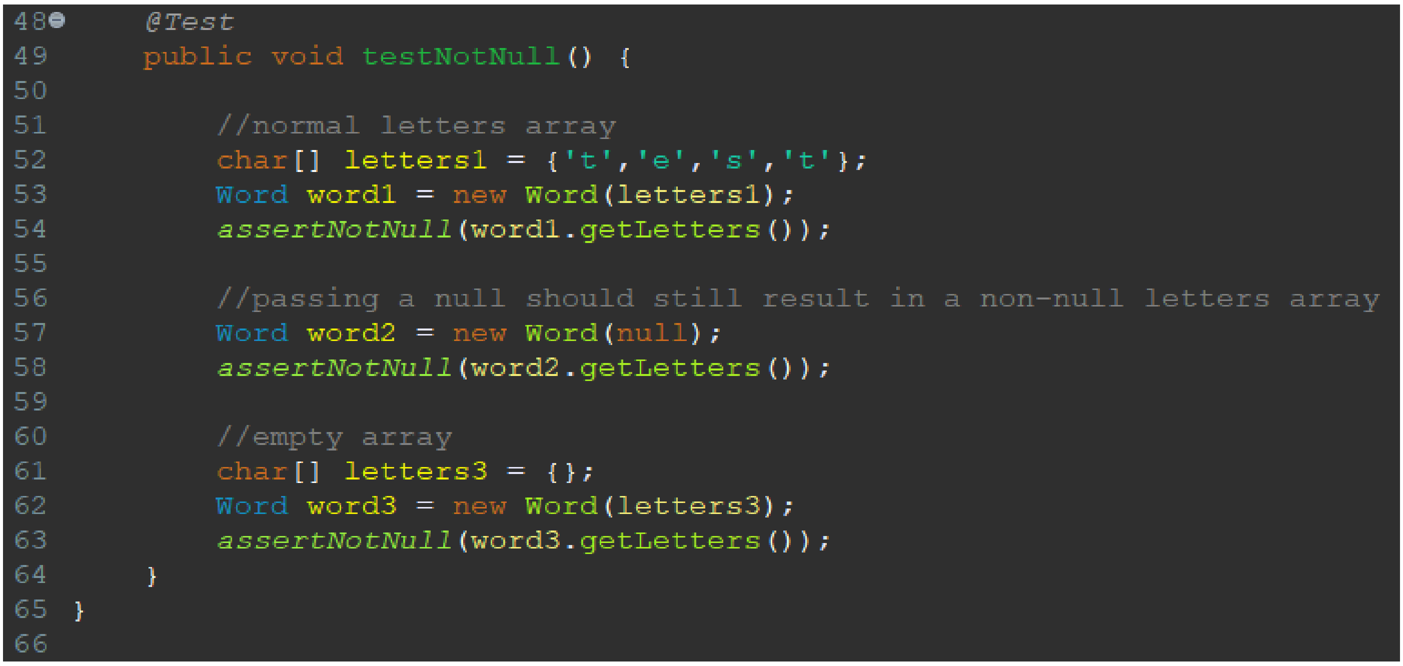
Task 2.



```java
//Enkhbaatar Idersaikhan 3112121
package griffith;

public class Word {

    //attribute
    private char[] letters;

    //constructor that makes sure letters is never null
    public Word(char[]letters) {
        if(letters == null) {
            this.letters = new char[0];
        }else {
            this.letters = letters;
        }
    }
    //Returns true if the letters array contains the gi
    public boolean contains(char symbol) {
        for(char letter : letters) {
            if(letter == symbol) {
                return true;
            }
        }
        return false;
    }
    //Returns the number of letters
    public int length() {
        return letters.length;
    }
    //returns the letters array
    public char[] getLetters() {
        return letters;
    }

}
```

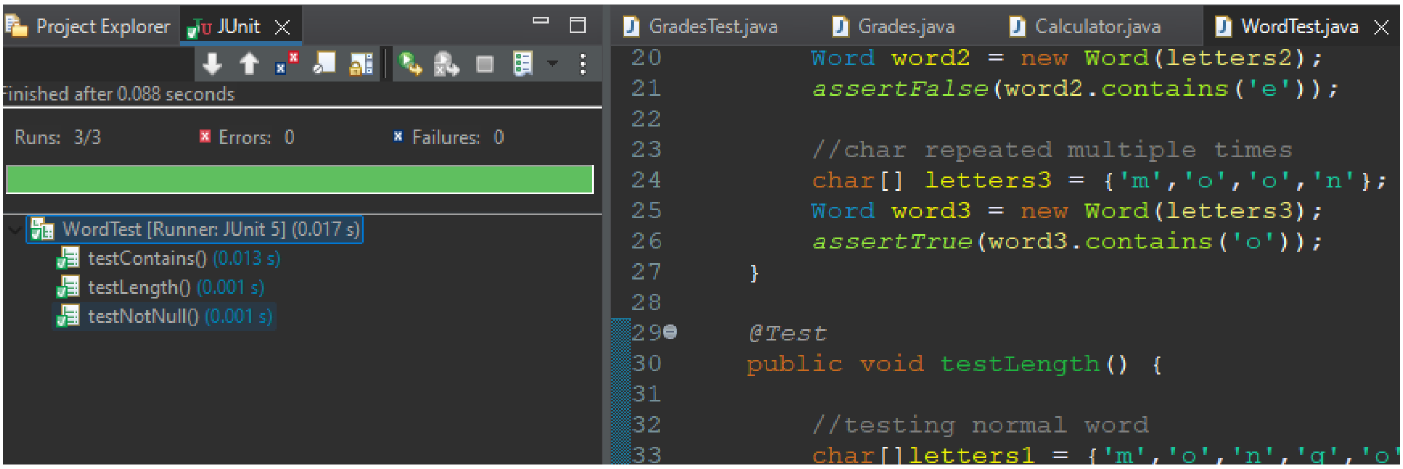


```java
//Enkhbaatar Idersaikhan 3112121
package griffith;

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

class WordTest {

    @Test
    public void testContains() {

        //word contains the char
        char[] letters1 = {'h','e','l','l','o'};
        Word word1 = new Word(letters1);
        assertTrue(word1.contains('e'));

        //word doesn't contains the char
        char[] letters2 = {'w','o','r','l','d'};
        Word word2 = new Word(letters2);
        assertFalse(word2.contains('e'));

        //char repeated multiple times
        char[] letters3 = {'m','o','o','n'};
        Word word3 = new Word(letters3);
        assertTrue(word3.contains('o'));
    }

    @Test
    public void testLength() {

        //testing normal word
        char[]letters1 = {'m','o','n','g','o','l','i','a'};
        Word word1 = new Word(letters1);
        assertEquals(8,word1.length());

        //testing 1 letter
        char[]letters2 = {'f'};
        Word word2 = new Word(letters2);
        assertEquals(1,word2.length());

        //testing empty array
        char[]letters3 = {};
        Word word3 = new Word(letters3);
        assertEquals(0,word3.length());
    }

    @Test
    public void testNotNull() {
```



```
    @Test
    public void testNotNull() {

        //normal letters array
        char[] letters1 = {'t','e','s','t'};
        Word word1 = new Word(letters1);
        assertNotNull(word1.getLetters());

        //passing a null should still result in a non-null letters array
        Word word2 = new Word(null);
        assertNotNull(word2.getLetters());

        //empty array
        char[] letters3 = {};
        Word word3 = new Word(letters3);
        assertNotNull(word3.getLetters());
    }
}

```

TEST



```
Finished after 0.088 seconds

Runs:  3/3          Errors:  0          Failures:  0

WordTest [Runner: JUnit 5] (0.017 s)
    testContains() (0.013 s)
    testLength() (0.001 s)
    testNotNull() (0.001 s)
```

```
        Word word2 = new Word(letters2);
        assertFalse(word2.contains('e'));

        //char repeated multiple times
        char[] letters3 = {'m','o','o','n'};
        Word word3 = new Word(letters3);
        assertTrue(word3.contains('o'));
    }

    @Test
    public void testLength() {

        //testing normal word
        char[]letters1 = {'m','o','n','g','o
```

My user handle for GitHub is : SylerEdd

GitHub link to worksheet 1: https://github.com/SylerEdd/JUnit