

# Отчёт по 7 лабе

October 2022

- 1 **Выбранный инструмент**
- 2 **Вид проекта**
- 3 **Программа**
- 4 **Тесты**
- 5 **Результаты**

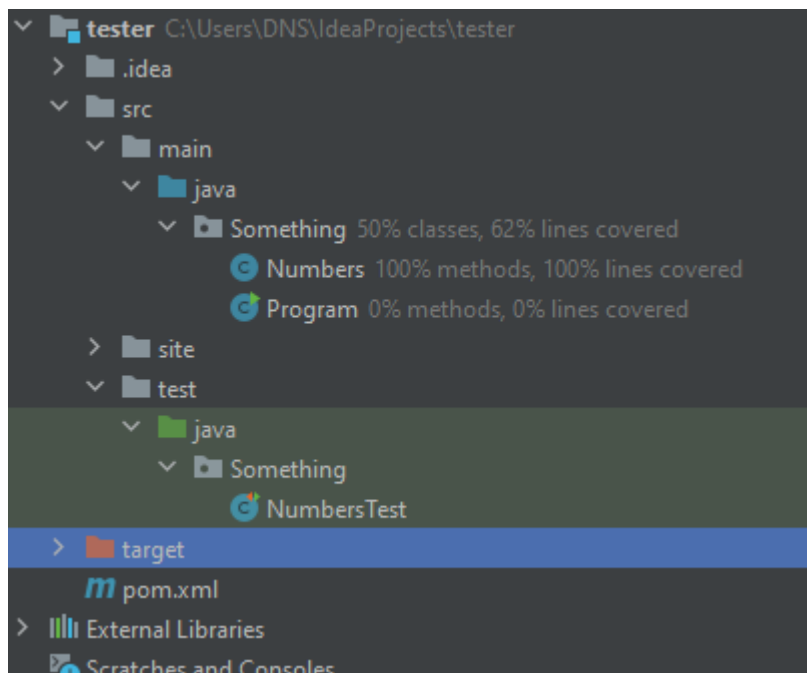
## Выбранный инструмент

**IntelliJ IDEA** — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, с широким набором интегрированных инструментов для рефакторинга, которые позволяли программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов, позволяя сконцентрироваться на функциональных задачах, в то время как IntelliJ IDEA берёт на себя выполнение рутинных операций. Среди прочих возможностей, среда хорошо совместима со многими популярными свободными инструментами разработчиков, такими как CVS, Subversion, Apache Ant, Maven и JUnit

**Apache Maven** — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML

**JUnit** — фреймворк для модульного тестирования программного обеспечения на языке Java.

## Вид проекта



файлы проекта

Папка Java (синяя) - источники. В ней хранится метод и main файл(Program)

Папка Java (зелёная) - тесты. В файле NumbersTest хранятся все написанные тесты, и можно проверить как один тест, так и все сразу

**pom.xml** - это XML-файл, который содержит информацию о конфигурации и деталях проекта, используемых при создании проекта на Maven. Он всегда находится в базовом каталоге проекта. Этот файл также содержит описание задач, список и параметры плагинов.

# Программа

” Папка Java ([синяя](#)) - источники. В ней хранится метод и main файл(Program)”

```
package Something;

public class Numbers {
    12 usages
    public int sum(int number) {
        int result = 0;
        for (int i = 0; i <= number; i++) {
            result += i;
        }
        return result;
    }
}
```

Метод Numbers - считает сумму чисел от 1 до введённого числа

```
package Something;

import Something.Numbers;

public class Program {

    public static void main(String[] args) {
        Numbers numbers = new Numbers();
        int sum5 = numbers.sum(number: 5);
        System.out.println("Sum from 0 till 5 = " + sum5);
    }
}
```

Program - для проверки работы метода

```
Sum from 0 till 5 = 15
```

```
Process finished with exit code 0
```

# Тесты

Всего тестов получилось 11

## Test1

```
public class NumbersTest {

    @org.junit.Test
    public void test() {
        Numbers numbers = new Numbers(); //Всегда положительный
        int actual = numbers.sum( number: 5);
        int expected = 15;
        assertEquals(expected, actual);
        System.out.println("Ожидалось " + expected);
        System.out.println("наше число " + actual);
    }
}
```

## Test2

```
@org.junit.Test
public void test2() {
    Numbers numbers = new Numbers(); //Сумма у randomного числа должна совпасть с выбранным числом
    double i = Math.random()*10;    //В данном случае число 21
    int l = (int) i;
    int actual = numbers.sum(l);
    int expected = 21;
    assertEquals(expected, actual);
    System.out.println("Ожидалось " + expected);
    System.out.println("наше число " + actual);
}
```

## Test3

```
@org.junit.Test
public void test3() {
    Numbers numbers = new Numbers(); // Всегда положительный Expected = actual
    int actual = numbers.sum( number: 4);
    int expected = actual;
    assertEquals(expected, actual);
    System.out.println("Ожидалось " + expected);
    System.out.println("наше число " + actual);
}
```

## Test4

```
@org.junit.Test
public void test4() { // Всегда провальный
    Numbers numbers = new Numbers();
    int actual = numbers.sum( number: 4);
    int expected = 20;
    assertEquals(expected, actual);
}
```

## Test5

```
@org.junit.Test
public void test5() {
    Numbers numbers = new Numbers(); // Краевое, грань
    int actual = numbers.sum( number: 4);
    if (actual > 0 && actual <= 10) {
        boolean man = true; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10" );
    } else {boolean man = false; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10"); fail("Не попадает в диапазон");}
}
```

## Test6

```
@org.junit.Test
public void test6() {
    Numbers numbers = new Numbers();
    int actual = numbers.sum( number: 5); //Крайнее, в диапазоне
    if (actual > 0 && actual <= 10) {
        boolean man = true; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10" );
    }else {boolean man = false; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10"); fail("Не попадает в диапазон");}
}
```

## Test7

```
@org.junit.Test
public void test7() {
    Numbers numbers = new Numbers(); //Крайнее, за диапазоном
    int actual = numbers.sum( number: 5);
    if (actual > 0 && actual <= 10) {
        boolean man = true; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10" );
    }else {boolean man = false; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до 10"); fail("Не попадает в диапазон");}
}
```

## Test8

```
@org.junit.Test
public void test8() {
    double d = Math.random()*50; //Рандомный диапазон
    Numbers numbers = new Numbers();
    int actual = numbers.sum( number: 7);
    if (actual > 0 && actual <= d) {
        boolean man = true; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до " +d);
    }else {boolean man = false; System.out.println("наше число " + actual); System.out.println("Диапазон от 0 до " +d); fail("Не попадает в диапазон");}
}
```



## Test9

```
@org.junit.Test
public void test9() {
    double d = Math.random()*100; //Увеличение диапазона
    Numbers numbers = new Numbers();
    int actual = numbers.sum( number: 7);
    if (actual > 0 && actual <= d) {
        boolean man = true; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d);
    }else {boolean man = false; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d); fail("Не попадает в диапазон");}
}
```

## Test10

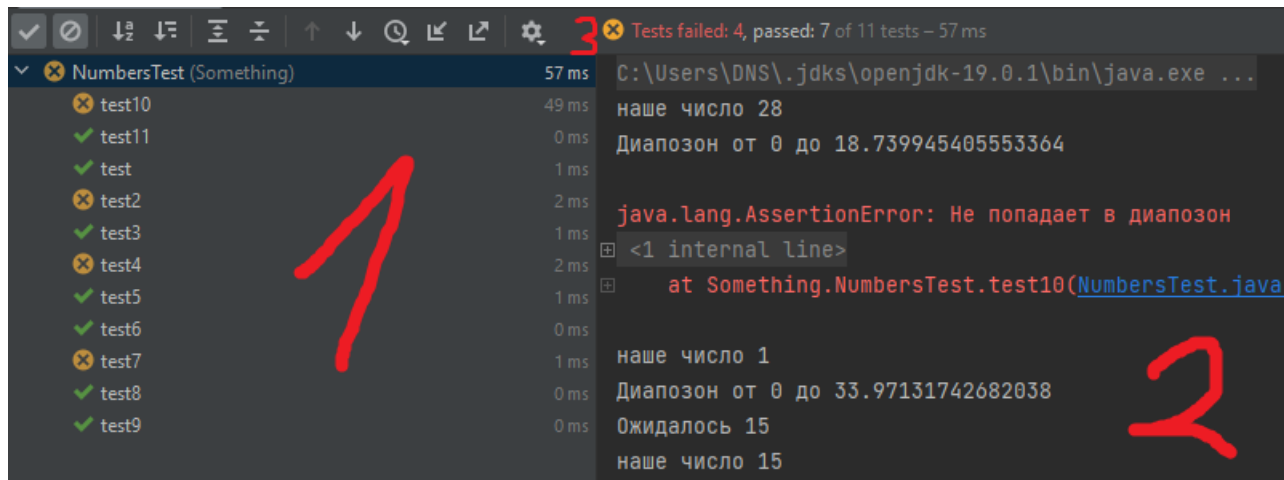
```
@org.junit.Test
public void test10() {
    double d = Math.random()*40; //Уменьшение диапазона
    Numbers numbers = new Numbers();
    int actual = numbers.sum( number: 7);
    if (actual > 0 && actual <= d) {
        boolean man = true; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d);
    }else {boolean man = false; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d); fail("Не попадает в диапазон");}
}
```

## Test11

```
@org.junit.Test
public void test11() {
    double d = Math.random()*50; //Рандомный диапазон рандомное число
    double i = Math.random()*10;
    int l = (int) i;
    Numbers numbers = new Numbers();
    int actual = numbers.sum(l);
    if (actual > 0 && actual <= d) {
        boolean man = true; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d);
    }else {boolean man = false; System.out.println("наше число " + actual);System.out.println("Диапазон от 0 до " +d); fail("Не попадает в диапазон");}
}
```

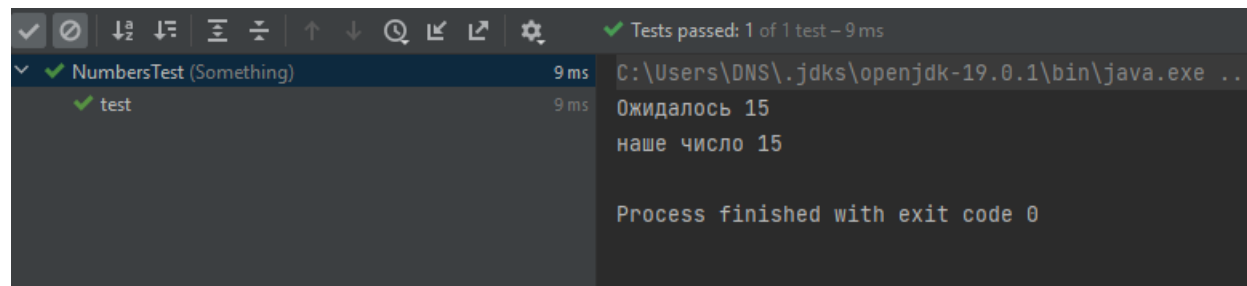
## Результаты

Результат тестов можно узнать как у одного конкретного теста, так и у всех одновременно



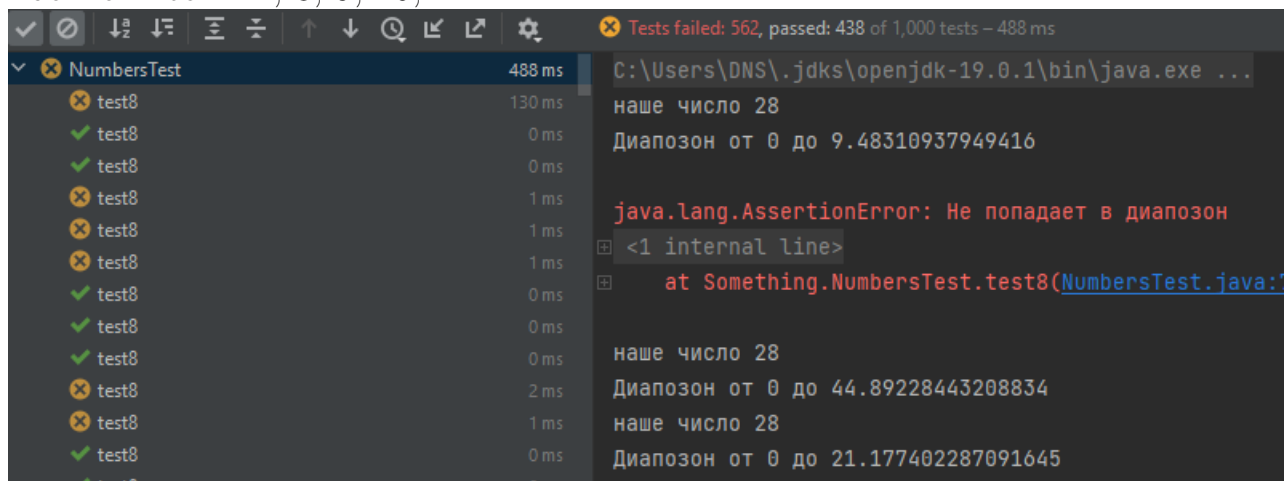
Результат всех тестов

- 1 - Кратко результаты тестирования, графически. Можно кликнуть на каждый тест и посмотреть результат
- 2 - Конкретные результаты тестирования. Можно видеть почему тест пройден или не пройден
- 3 - Краткие результаты тестирования/Статистика пройденных и не пройденных тестов



Результат одного теста

Также есть тесты которые требуют множественного вызова, т.к. данные в них берутся рандомные, к ним относятся тесты 2, 8, 9, 10, 11



Пример вызова 1000 раз теста номер 8

Здесь нас будут интересовать статистические данные, а именно то что 562 теста не прошло тестирование и 438 прошло, из этого можно выявить примерный процент для прохождения теста

## **Шансы прохождения для тестов:**

1. 100%
2. 9,1%
3. 100%
4. 0%
5. 100%
6. 100%
7. 0%
8. 43,8%
9. 71,8%
10. 32,2%
11. 56,9%