**Содержание**

[1. Цель лабораторной работы 3](#_3znysh7)

[2. Задание 3](#_2et92p0)

[3. Поэтапное выполнение лабораторной работы 3](#_tyjcwt)

[4. Листинг программы](#_tyjcwt) 3

[5. Вывод программы](#_tyjcwt) 6

[Вывод](#_4d34og8) 7

[Список использованных источников](#_2s8eyo1) 8

**1. Цель лабораторной работы**

Ознакомится с механизмом модульного тестирования для программ, написанных на языке JAVA, изучить основы разработки модульных тестов, изучить основы работы с JUnit и Hamcrest.  
**2. Задание**Написать программу-калькулятор, реализующий функции сложения, вычитания, умножения, деления и подсчёт факториала числа. Осуществить тестирование разрабатываемого проекта, с использованием JUnit и Hamcrest. Создать тестовый класс и реализовать тестирование на проверку сложения малых чисел (например, 0.0000001+0.00000007), больших чисел, деления на ноль, умножения на ноль, проверки подсчёта факториала отрицательного числа, проверки подсчёта факториала числа больше 20.  
**3. Поэтапное выполнение лабораторной работы**1.Установка JDK, IDE, Gradle  
2.Добавление в проекте gradle в build gradle   
dependencies {  
 testCompile group: 'junit', name: 'junit', version: '4.12'  
 testCompile group: 'org.hamcrest', name: 'hamcrest-all', version: '1.3'}  
3.Написание “мертвого” тела кода для реализации калькуляции чисел необходимых в лабораторной.  
4.Написание тест методов.  
5. Компиляция и запуск. (рис.1) **4. Листинг программы.  
Calculator.java**import java.math.BigInteger;  
public class Calculator {  
 public double getSum(double x, double y) {  
 return x + y;  
 }  
 public int getDivide(int x, int y) {  
 return x / y;

}

public int getMultiple(int x, int y) {

return x \* y;

}

public int factorial(int x) {

int result = 1;

if (x > 0) {

for (int i = 1; i <= x; i++) {

result = result \* i;

}

return result;

}

else {

x=x\*-1;

for (int i = 1; i <= x; i++) {

result = result \* i;

}

return result\*-1;

}

}

public String bigFactorial(int x) {

BigInteger fact = new BigInteger("1");

for (int i = 1; i <= x; i++) {

fact = fact.multiply(new BigInteger(i + ""));

}

return fact.toString();

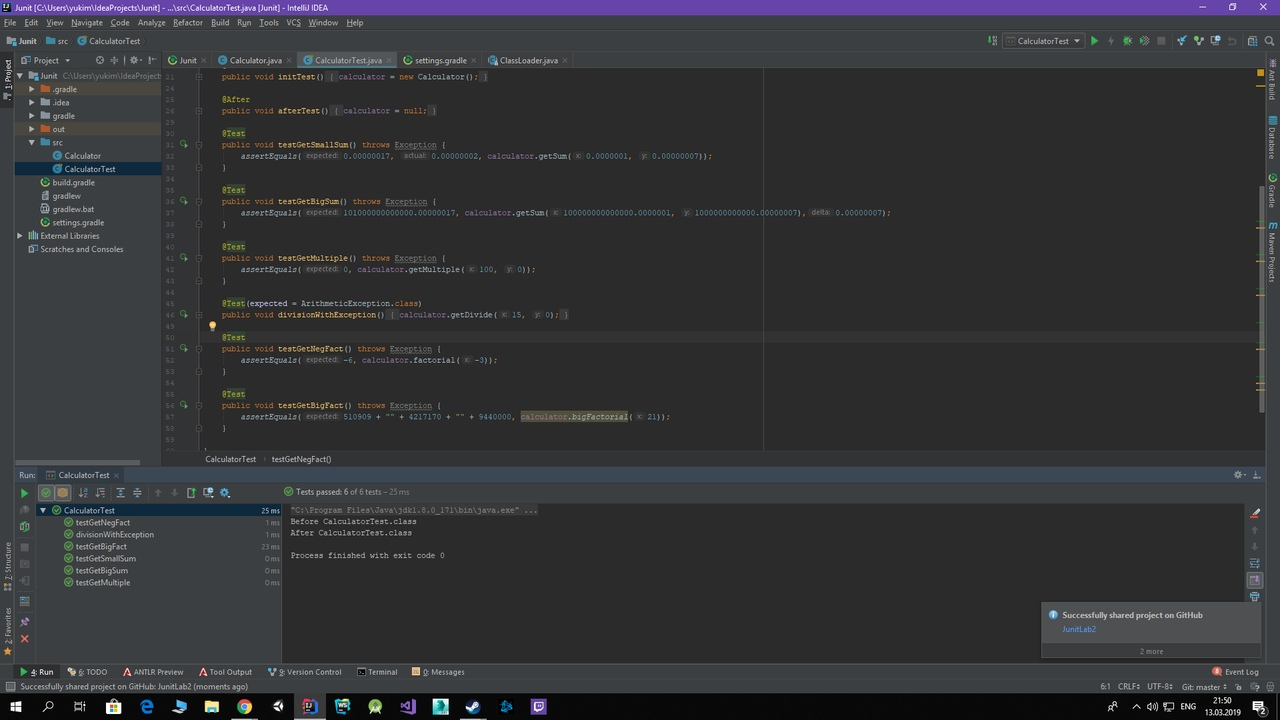
}

}

**CalculatorTest.java**import org.junit.\*;  
import org.hamcrest.SelfDescribing;  
import static org.junit.Assert.\*;  
public class CalculatorTest {  
 private Calculator calculator;  
 @Before  
 public void initTest() {  
 calculator = new Calculator();  
 }  
 @After  
 public void afterTest() {  
 calculator = null;  
 }  
 @Test  
 public void testGetSmallSum() throws Exception { *assertEquals*(0.00000017, 0.00000002, calculator.getSum(0.0000001, 0.00000007));  
 }  
 @Test  
 public void testGetBigSum() throws Exception {

*assertEquals*(101000000000000.00000017, calculator.getSum(100000000000000.0000001, 1000000000000.00000007),0.00000007);  
 }  
 @Test  
 public void testGetMultiple() throws Exception {  
 *assertEquals*(0, calculator.getMultiple(100, 0));  
 }  
 @Test(expected = ArithmeticException.class)  
 public void divisionWithException() {  
 calculator.getDivide(15, 0);  
 }  
 @Test  
 public void testGetNegFact() throws Exception {  
 *assertEquals*(-6, calculator.factorial(-3));  
 }  
 @Test  
 public void testGetBigFact() throws Exception {  
 *assertEquals*(510909 + "" + 4217170 + "" + 9440000, calculator.*bigFactorial*(21));  
 }  
}

**5. Результат работы программы**

****Рисунок 1 — Вывод программы.

**Вывод**

Было произведено ознакомление с механизмом модульного тестирования для программ, написанных на языке JAVA, изучены основы разработки модульных тестов, изучить основы работы с JUnit и Hamcrest. Разработан тест-калькулятор.

Поставленные цель и задачи выполнены.

**Список использованных источников**

1. Чеповский, А.М. Common Intermediate Language и системное программирование в Microsoft .NET [Электронный ресурс] : учебное пособие / А.М. Чеповский, А.В. Макаров, С.Ю. Скоробогатов. — Электрон. дан. — Москва : , 2016. — 398 с. — Режим доступа: https://e.lanbook.com/book/100589. — Загл. с экрана.

2. Голдштейн, С. Оптимизация приложений на платформе .NET [Электронный ресурс] / С. Голдштейн, Д. Зурбалев, И. Флатов ; пер. с англ. Киселев А.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2014. — 524 с. — Режим доступа: https://e.lanbook.com/book/93266. — Загл. с экрана.

3.Чеповский, А.М. Common Intermediate Language и системное программирование в Microsoft .NET [Электронный ресурс] : учебное пособие / А.М. Чеповский, А.В. Макаров, С.Ю. Скоробогатов. — Электрон. дан. — Москва : , 2016. — 398 с. — Режим доступа: https://e.lanbook.com/book/100589. — Загл. с экрана.

4. Крищенко, В.А. Поддержка разработки распределенных приложений в Microsoft .NET Framework [Электронный ресурс] : учебное пособие / В.А. Крищенко, С.В. Горин. — Электрон. дан. — Москва : , 2016. — 249 с. — Режим доступа: https://e.lanbook.com/book/100367. — Загл. с экрана.

5. Суханов, М.В. Основы Microsoft .NET Framework и языка программирования C# [Электронный ресурс] : учебное пособие / М.В. Суханов, И.В. Бачурин, И.С. Майров. — Электрон. дан. — Архангельск : САФУ, 2014. — 96 с. — Режим доступа: https://e.lanbook.com/book/96543. — Загл. с экрана.