МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №5

Специальность ПО-5

Выполнила

А. А. Нерода

студентка группы ПО-5

Проверил

А. А. Крощенко

ст. преп. кафедры ИИТ,

Брест 2022

**Вариант 10**

**Цель работы:** освоить приемы тестирования кода на примере использования библиотеки JUnit

**Задание 1 – Введение в JUnit**

• Создаете новый класс и скопируйте код класса Sum;

• Создаете тестовый класс SumTest;

• Напишите тест к методу Sum.accum и проверьте его исполнение. Тест должен проверять работоспособность функции accum.

• Очевидно, что если передать слишком большие значения в Sum.accum, то случится переполнение. Модифицируйте функцию Sum.accum, чтобы она возвращала значение типа long и напишите новый тест, проверяющий корректность работы функции с переполнением. Первый тест должен работать корректно.

public class Sum {

public static int accum ( int ... values ) {

int result = 0;

for ( int i = 0; i < values . length ; i ++) {

result += values [ i ];

}

return result ;

}

}

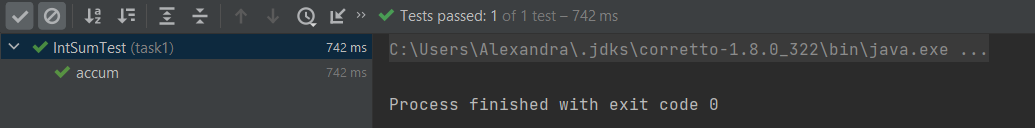
**IntSum.java**

package task1;  
  
public class IntSum {  
  
 public int accum ( int[] values ) {  
 int result = 0;  
 for ( int i = 0; i < values.length; i++) {  
 result += values[i];  
 }  
 return result;  
 }  
}

**IntSumTest.java**

package task1;  
  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class IntSumTest {  
  
 @Test  
 public void accum() {  
 int[] val = new int[]{1, 2, 3, 4, 5};  
 IntSum is = new IntSum();  
 int actual = is.accum(val); // реальность  
 int expected = 15; // ожидание  
 *assertEquals*(expected, actual); // проверка на эквивалентность  
 }  
}

**Результат:**

****

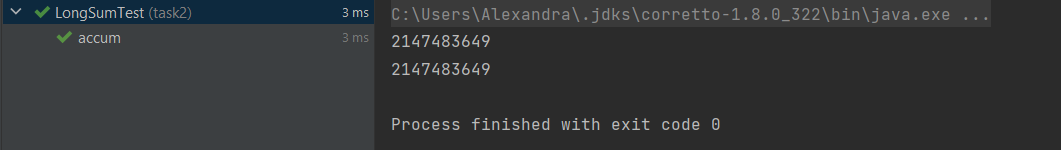
**LongSum.java**

package task2;  
  
public class LongSum {  
  
 public static long accum ( long[] values ) {  
 long result = 0;  
 for ( int i = 0; i < values.length; i++) {  
 result += values[i];  
 }  
 return result;  
 }  
}

**LongSumTest.java**

package task2;  
  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class LongSumTest {  
  
 @Test  
 public void accum() {  
 long[] val = new long[]{Integer.*MAX\_VALUE*, 2};  
 LongSum ls = new LongSum();  
 long actual = ls.*accum*(val); // реальность  
 System.*out*.println(actual);  
 long expected = Integer.*MAX\_VALUE* + 2L; // ожидание  
 System.*out*.println(expected);  
 *assertEquals*(expected, actual); // проверка на эквивалентность  
 }  
}

**Результат:**

****

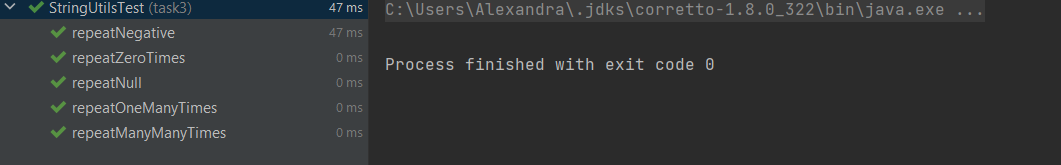
**StringUtils.java**

package task3;  
  
public class StringUtils {  
  
 public String repeat(String pattern, int repeat) {  
 if (repeat < 0) {  
 throw new IllegalArgumentException("Positive parameter expected.");  
 }  
 if (pattern == null) {  
 throw new NullPointerException("Non-nullable pattern expected.");  
 }  
 StringBuilder builder = new StringBuilder();  
 for (int i = 0; i < repeat; ++i) {  
 builder.append(pattern);  
 }  
 return builder.toString();  
 }  
}

**StringUtilsTest.java**

package task3;  
  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class StringUtilsTest {  
 private static StringUtils *stringUtils* = new StringUtils();  
 @Test  
 public void repeatZeroTimes() {  
 *assertEquals*("", *stringUtils*.repeat("j", 0));  
 }  
 @Test  
 public void repeatOneManyTimes() {  
 *assertEquals*("jjjj", *stringUtils*.repeat("j", 4));  
 }  
 @Test  
 public void repeatManyManyTimes() {  
 *assertEquals*("ABCABC", *stringUtils*.repeat("ABC", 2));  
 }  
 @Test  
 public void repeatNegative() {  
 *assertThrows*(IllegalArgumentException.class, () ->  
 *stringUtils*.repeat("ABC", -5));  
 }  
 @Test  
 public void repeatNull() {  
 *assertThrows*(NullPointerException.class, () ->  
 *stringUtils*.repeat(null, 1));  
 }  
  
}

**Результат:**

****