



# УНИВЕРСИТЕТ ИТМО

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
“Национальный исследовательский университет ИТМО”

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

## **ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине  
“Тестирование программного обеспечения”

Вариант: 618.

*выполнил:*

Студент группы Р33131

**Коновалов Арсений Антонович**

*Преподаватель*

**Харитонов Анастасия Евгеньевна**

г. Санкт-Петербург  
2024 г.

# 1   Оглавление

(Ссылки кликабельны)	
Задание.....	3
Выполнение.....	3
Исходный код.....	5
Вывод.....	3

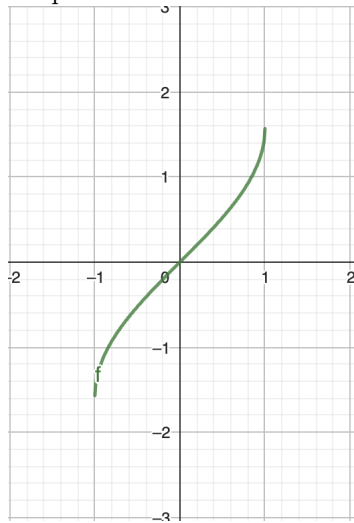
## 2 Задание

- 1 - Для указанной функции провести модульное тестирование разложения функции в степенной ряд. Выбрать достаточное тестовое покрытие.
- 2 - Провести модульное тестирование указанного алгоритма. Для этого выбрать характерные точки внутри алгоритма, и для предложенных самостоятельно наборов исходных данных записать последовательность попадания в характерные точки. Сравнить последовательность попадания с эталонной.
- 3 - Сформировать доменную модель для заданного текста. Разработать тестовое покрытие для данной доменной модели

## 3 Выполнение

### 3.1 Задание 1:

Так как функция определена на отрезке от -1 до 1 достаточным тестовым покрытием будет являться покрытие семи точек.



[Код функции](#)

Тестовый класс:

```
src/tp01.py
1 @ParameterizedTest(name = "arcsin({0})")
2 @DisplayName("TestTask1")
3 @ValueSource(doubles = {-2 * PI, -PI, -0.5 * PI, 0, 0.5 * PI, PI, 1.5 * PI, 2
    * PI})
4 void checkPoints(double param) {
5     assertAll(
6         () -> assertEquals(Math.asin(param), Arcsin.calculate(param, 15),
9             0.001)
7     );
8 }
```

### 3.2 Задание 2:

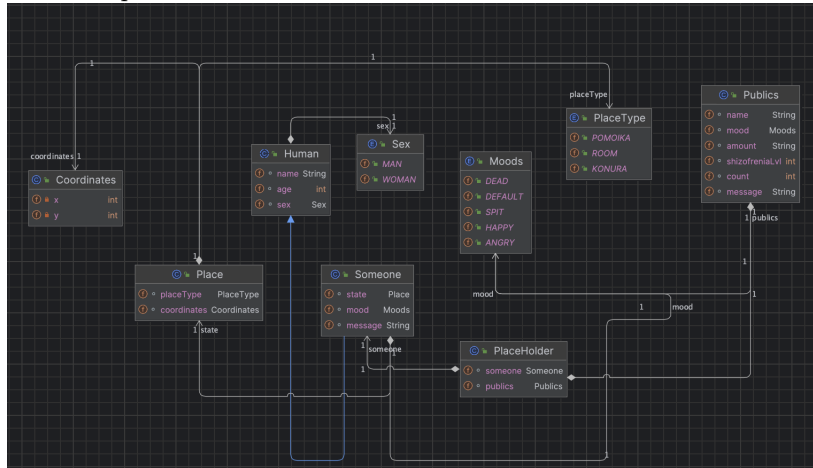
Исходный код алгоритма MergeSort:

src/tpo2\_src.py

```
1 public class MergeSort {
2     public static void merge(int[] arr, int l, int m, int r) {
3         int n1 = m - l + 1;
4         int n2 = r - m;
5
6         int[] L = new int[n1];
7         int[] R = new int[n2];
8
9         System.arraycopy(arr, l, L, 0, n1);
10        for (int j = 0; j < n2; ++j)
11            R[j] = arr[m + 1 + j];
12
13        int i = 0, j = 0;
14
15        int k = l;
16        while (i < n1 && j < n2) {
17            if (L[i] <= R[j]) {
18                arr[k] = L[i];
19                i++;
20            } else {
21                arr[k] = R[j];
22                j++;
23            }
24            k++;
25        }
26
27        while (i < n1) {
28            arr[k] = L[i];
29            i++;
30            k++;
31        }
32
33        while (j < n2) {
34            arr[k] = R[j];
35            j++;
36            k++;
37        }
38    }
39    public static int[] sort(int[] arr, int l, int r) {
40        if (arr == null || arr.length == 0){
41            throw new IllegalArgumentException();
42        }
43        if (l < r) {
44            int m = (l + r) / 3;
45
46            sort(arr, l, m);
47            sort(arr, m + 1, r);
48
49            merge(arr, l, m, r);
50        }
51        return arr;
52    }
53 }
```

src/tpo2.py

Учитывая скромность предметной области, имеющейся в варианте, было принято решение дополнить ее своей фантазией:



src/tpo3.py

Код модели

Можно найти по ссылке [на Github](#)

Выполняя первую лабораторную работу по дисциплине тестирование программного обеспечения я научился использовать библиотеку junit и обучился основам работы с тестами в Java.