

Министерство науки и высшего образования Российской Федерации  
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 333210

Выполнила:  
Студентка Р3332  
Копалина М.А.  
Проверила:  
Наумова Н.А.

г. Санкт-Петербург  
2025г.

# Задание

## Лабораторная работа #1

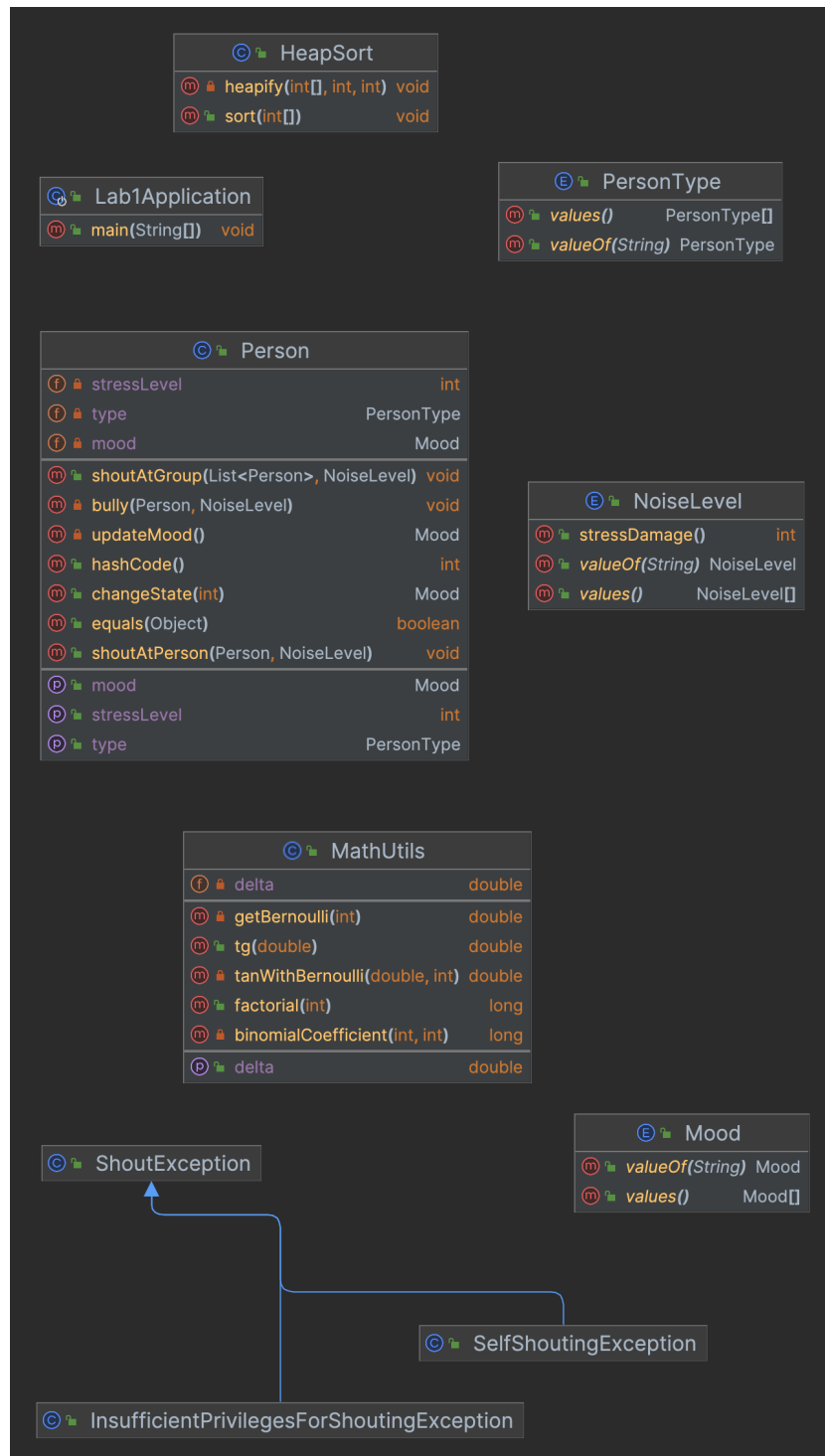
1. Для указанной функции провести модульное тестирование разложения функции в степенной ряд. Выбрать достаточное тестовое покрытие.
2. Провести модульное тестирование указанного алгоритма. Для этого выбрать характерные точки внутри алгоритма, и для предложенных самостоятельно наборов исходных данных 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

Введите вариант:

1. Функция  $\text{tg}(x)$
2. Программный модуль для работы с кучей (<http://www.cs.usfca.edu/~galles/visualization/HeapSort.html>)
3. Описание предметной области:

Простетный Вогон Джельц улыбнулся очень медленно. Он сделал так не ради эффекта, а потому что не мог вспомнить правильную последовательность движения мышц. Он только что побаловал себя освежающей серией воплей на своих пленников, и теперь чувствовал себя отдохнувшим и готовым к небольшой гнусности.

## UML – диаграмма



## Исходный код

[https://github.com/mkklpln/lab1\\_software\\_testing/tree/main](https://github.com/mkklpln/lab1_software_testing/tree/main)

## **Вывод**

В ходе данной лабораторной работы было проведено тестирование математической функции тангенса, реализованной с использованием ряда Бернулли. Для проверки корректности работы метода использовались параметризованные и модульные тесты с применением библиотеки JUnit5. Проведена обработка исключений для некорректных входных данных, а также реализована проверка поведения функции в критических точках. UML-диаграмма классов помогла визуализировать взаимодействие компонентов и структурировать логику вычислений.