

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет Программной инженерии и компьютерной техники

Отчёт по лабораторной работе №1

Тестирование программного обеспечения

Вариант №12136

Выполнил: студент группы Р3314

Силинцев В.В.

Преподаватель: Наумова Н.А.

Санкт-Петербург 2026

Содержание

| | |
|-------------------------------|---|
| Задание..... | 3 |
| Разработанное приложение..... | 5 |
| UML диаграмма..... | 6 |
| Выводы..... | 7 |

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.

Задание по варианту:

1. Функция $\arccos(x)$
2. Программный модуль для поиска кратчайшего пути в неориентированном графе по алгоритму Дейкстры (<http://www.cs.usfca.edu/~galles/visualization/Dijkstra.html>)
3. Описание предметной области: «Всех швырнуло к стенам, из которых тут же выскочили воздушные подушки. Несколько секунд, прижатые к стенам силами инерции, они не могли пошевелиться и лишь хватали ртами воздух. Зафод в отчаянии извернулся и, дотянувшись, пнул небольшой тумблер на навигационной панели.»

Дополнительные требования:

1. Используем JUnit 5
2. Не забываем убедиться в 100% code coverage для всех пунктов задания
3. Для тестирования мат. функции пользуемся параметризованными тестами
4. Табличные значения должны быть получены не из `java.lang.Math`
5. Добавьте UML к доменной модели (п.3) в отчет
6. Модель из п.3 не выводит текст в консоль, как на ЛР3 по программированию. Задача -- сделать модель с состояниями, и оперировать этими состояниями при составлении тестов. Если не хватает сущностей из исходного текста, текст можно расширить

Разработанное приложение

Полный исходный код приложения: <https://github.com/vvlaads/Software-Testing-1>.

UML диаграмма

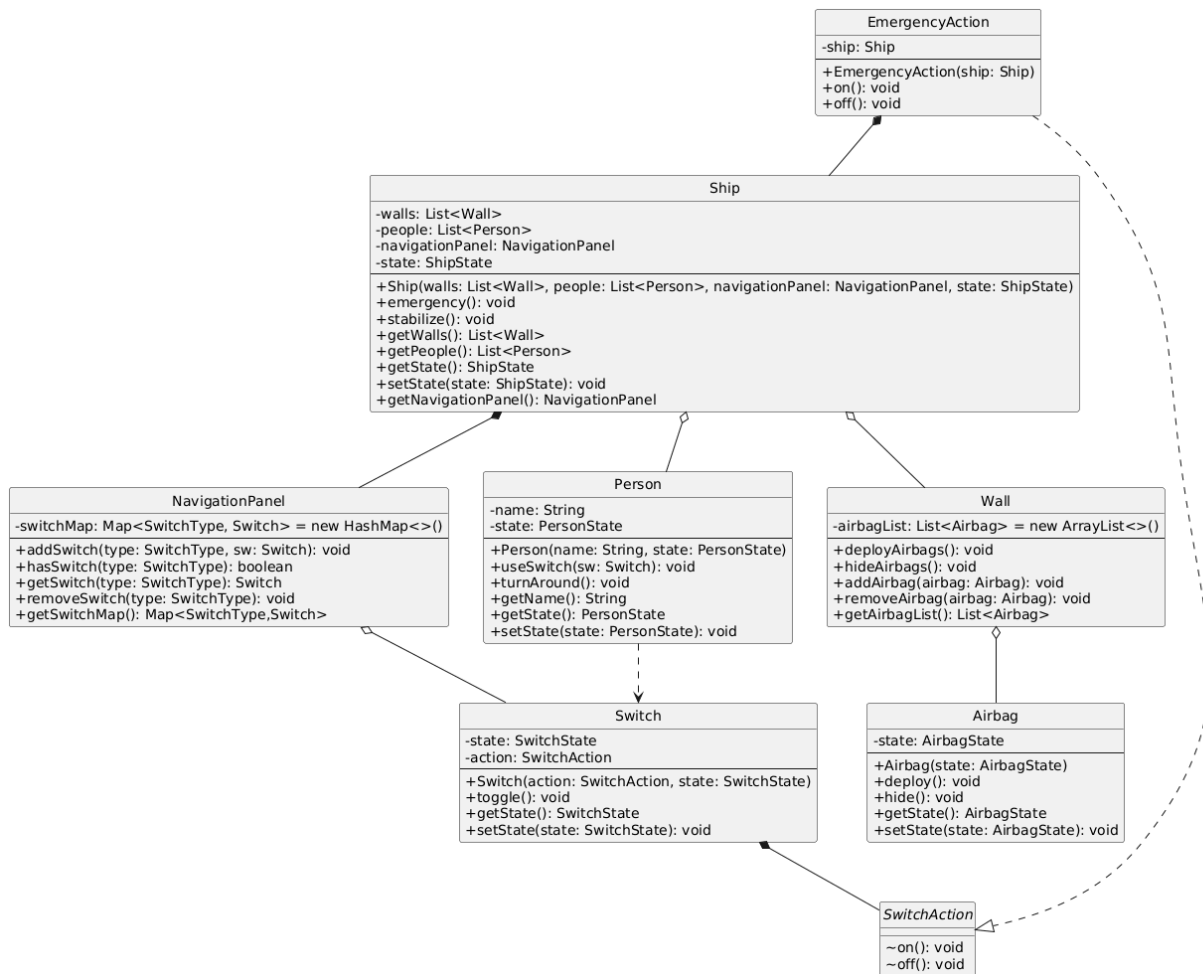


Рисунок 1: UML-диаграмма пункта 3

Выводы

В ходе работы я изучил тестирование с помощью JUnit 5, освоил написание юнит-тестов для разных классов и методов, а также использование параметризованных тестов для проверки функций на множестве входных данных. Научился измерять тестовое покрытие кода и добиваться 100% покрытия. Для упрощения тестов использовал TestFactory для создания готовых объектов, что позволило эффективно проверять сложные сценарии.