

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
СПЕЦИАЛИЗАЦИЯ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1
курса «Тестирование программного обеспечения»
по теме: «Модульное тестирование»
Вариант № 133204

Выполнил студент:
Тюрин Иван Николаевич
группа: Р33102

Преподаватель:
Письмак А. Е.,
Сорокин Р. Б.

Санкт-Петербург, 2024 г.

Содержание

Лабораторная работа № 1. Модульное тестирование	2
1. Задание варианта № 133204	2
2. Выполнение задания	3
1. Тестирование функции	3
2. Тестирование алгоритма	4
3. Тестирование доменной модели	4
3. Вывод	5

Лабораторная работа № 1

Модульное тестирование

1. Задание варианта № 133204

, , ,

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: 133204

1. Функция $\arcsin(x)$
2. Программный модуль для сортировки массива по алгоритму слиянием (<http://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>)
3. Описание предметной области: *Зажужжал мотор. Тоненький свист перерос в рев воздуха, вырывающегося в черную пустоту, усеянную невероятно яркими светящимися точками. Форд и Артур вылетели в открытый космос, как конфетти из хлопушки. Глава 8*

, , ,

2. Выполнение задания

В соответствии с заданием были выполнены следующие 3 подзадачи:

1. тестирование функции,
2. тестирование алгоритма,
3. тестирование доменной модели.

Код программы для выполнения заданий был выполнен на языке программирования Kotlin с использованием библиотеки Kotest, которая использует внутри себя Junit 5 для JVM-таргетов. Так же были использованы Property-based тесты с генераторами из библиотеки Kotest.

2. 1. Тестирование функции

Функция $\arcsin x$ раскладывается в ряд Тейлора:

$$\arcsin(x) = \sum_{n=0}^{\infty} \frac{((2n)! \cdot x^{(2n+1)})}{(4^n \cdot (n!)^2 \cdot (2n+1))} = x + \sum_{n=1}^{\infty} \frac{((2n)! \cdot x^{(2n+1)})}{(4^n \cdot (n!)^2 \cdot (2n+1))}$$

Для вычисления с помощью языка программирования можно сделать следующую декомпозицию с оптимизацией:

временная переменная

$$T_n = T_{n-1} \cdot \left[\frac{((2n-1) \cdot 2n) \cdot x^2}{4 \cdot n^2} \right] = T_{n-1} \cdot \left[\frac{(2n-1) \cdot x^2}{2 \cdot n} \right]$$

и переменная с результатом

$$R_n = \frac{T_n}{(2n+1)}$$

Так же для можно применить свойство центральной симметрии графика функции $\arcsin x$ и вычислять значение функции только для положительного аргумента и при необходимости инвертировать знак аргумента и результата.

Исследуя далее функцию можно выделить ключевые свойства функции:

- $\arcsin x$ определен лишь при $x \in [-1; 1]$,
- она непрерывна на всей области определения,
- она принимает определенные, табличные значения в следующих точках:

1. $\arcsin(0) = 0$
2. $\arcsin(-1) = -\frac{\pi}{2}$
3. $\arcsin(1) = \frac{\pi}{2}$
4. $\arcsin(-0.5) = -\frac{\pi}{6}$
5. $\arcsin(0.5) = \frac{\pi}{6}$
6. $\arcsin(-\frac{\sqrt{3}}{2}) = -\frac{\pi}{3}$
7. $\arcsin(\frac{\sqrt{3}}{2}) = \frac{\pi}{3}$

Соответствующие ограничения функции были проверены с помощью соответствующих тестов, при этом численные значения проверялись не на строгое соответствие, а с учетом некоторой погрешности. И так как погрешность не оговаривалась в условии задания, она была выбрана эмпирически.

2. 2. Тестирование алгоритма

Требуемый алгоритм сортировки слиянием был написан в рекурсивном стиле.

Основными требованиями которые предъявляются к алгоритму сортировки:

- сортировка не выкидывает элементы коллекции, т.е. размер входной и выходной коллекции совпадают,
- после сортировки массив является отсортированным, т.е. после сортировки любой перестановки упорядоченной коллекции мы должны получить первоначально упорядоченную коллекцию.

Исходные коды алгоритма представлены в репозитории, модуле `task-2-algorithm`: <https://github.com/e1turin/itmo-sw-testing/tree/main/lab-1-unit-testing/task-2-algorithm>.

2. 3. Тестирование доменной модели

Доменная модель составлена в соответствии с приведенным отрывком из текста. В нем были выделены следующие сущности:

- Двигатель «Engine»,
- Шум «Noise» ,

- Космос «Space»,
- Точки (звезды, понятно из контекста) «Point»,
- Эффект «Effect», в частности эффект трансформации в конфетти,
- Персонаж «Character», Форд и Артур.

Каждая доменная сущность имеет свои атрибуты и методы, подробнее можно видеть на изображении 1.1.

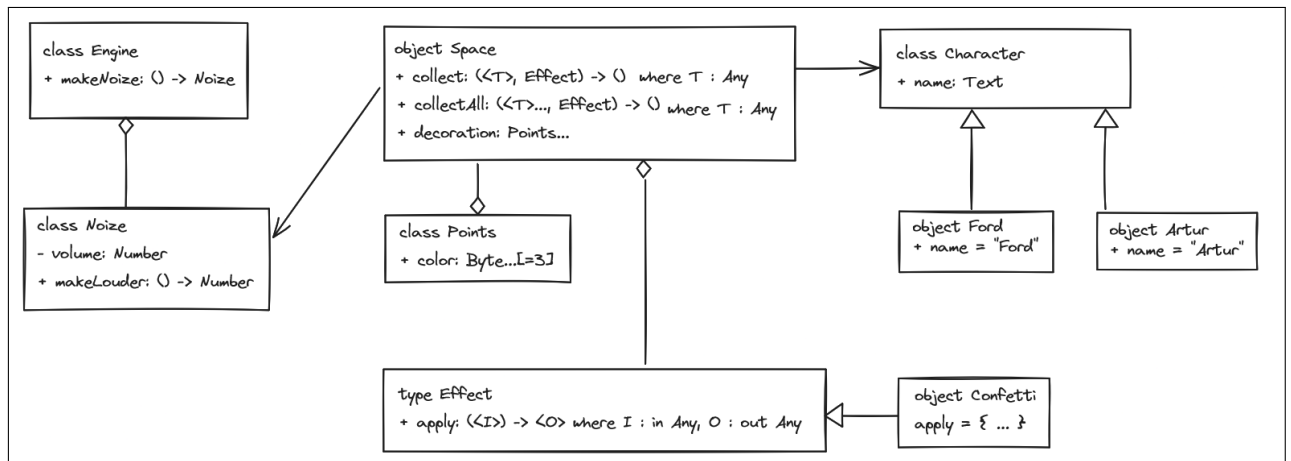


Рис. 1.1: Caption

Собственно модульному тестированию были подвержены методы объекта «Space» и класса «Noise». Космос должен правильно втягивать в себя объекты, т.е. с определенными сайд-эффектами. Шум должен правильным образом увеличиваться. Код можно найти в репозитории в соответствующем модуле **task-3-story**: <https://github.com/elturin/itmo-sw-testing/tree/main/lab-1-unit-testing/task-3-story>.

Тестирование всей истории было не было произведено, т.к. это не является частью задания.

3. Вывод

В соответствии с заданием был разработан программный код для модульного тестирования разработанных программных компонентов. Все исходные коды можно найти в репозитории: <https://github.com/elturin/itmo-sw-testing/>. Во время выполнения лабораторной работы были укреплены навыки программирования, выработаны навыки написания модульных тестов и Property-based тестов; был получен опыт работы с библиотекой Kotest.