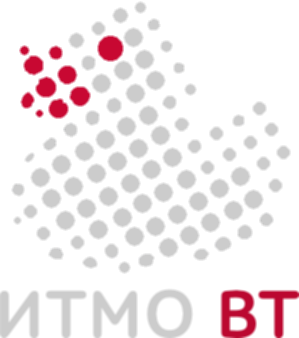
Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа №1

по дисциплине

«Тестирование программного обеспечения»



Выполнил: Студент группы P33112

Почикалин Владислав

Преподаватель: Харитонова Анастасия Евгеньевна

Санкт-Петербург

2021 г.

**Изображение выглядит как текст

Автоматически созданное описаниеВариант 33152:**

**Задание №1:**

Так как функция sin(x) бесконечна и периодична, при x -> (-inf, +inf), то ее степенной ряд будет сходится к ней на всей оси OX.

Формула степенного ряда функции sin(x):

Изображение выглядит как текст, антенна

Автоматически созданное описание

В качестве вида степенного ряда был выбран ряд Маклорена.

Так как область сходимости – вся ось OX, то никаких ограничений на X накладываться не будет, следовательно необходимо протестировать только то, насколько значение ряда близко к значению функции sin(x) при одинаковом x.

Для этого нужно брать во внимание погрешность вычислений, а также количество членов ряда Маклорена, так как их количество влияет на точность сравнения.

Беря эти факты во внимание, а также то, что при больших мантиссах чисел в Java, их значение становится NaN, было выбрано наиболее сбалансированное количество членов ряда(10) для таких вычислений.

В качестве проверки совпадения значений, был выбран метод Assert.assertEquals(), которые позволяет использовать аргумент delta, указывающий на то, с какой погрешностью числа будут считаться равными.

[**Код**](https://github.com/Whatislove118/TPO_LABS/blob/master/src/com/lab1/FirstExercise.java)

**Задание №2:**

Для того, чтобы провести модульное тестирование данной кучи, нужно вначале понять, что это за вид кучи.

Биноминальная куча – состоит из биноминальных деревьев(фактически, биноминальная куча – это набор куч не связанных друг с другом)

Биноминальные деревья бывают разных порядков, однако стоит понимать одну главную вещь – биноминальная куча размера К представляет собой набор биноминальных деревьев с попарно разными порядками. (если размер K – то b0 + b1 + b2 … + bk) Важно отметить, что по мере добавления данных в кучу порядок постепенно увеличивается. При каждом таком увеличении вершина биноминальной кучи связывает с собой кучу текущего порядка. Таким образом, при постепенно добавлении данных в биноминальное дерево, данная операция будет происходить все реже (тк после связывания главной

вершины с бинарным деревом К порядка, вначале строится отдельно от вершины биноминальной кучи биноминальное

дерево до тех пор, пока порядок К не будет равен текущему порядку биноминальной кучи; после этого главная вершина связывает себя с данным бинарным деревом, и теперь для повторения данной процедуры необходимо построить дерево высшего порядка, чем текущее значение) Бинарные деревья также наследуют данное свойство биноминальной кучи.

Также вершины биноминального дерева хранят информацию о минимальном значении в дереве.

Характерные точки:

* Операция добавления вершины
* Удаление заданной вершины
* Изменение вершины
* Операция нахождения минимума
* Удаление наименьшего

Операция добавления вершин:

|  |  |  |  |
| --- | --- | --- | --- |
| Условия тестирования | Начальные данные | Результат работы алгоритма | Эталонный результат работы алгоритма |
| k = 3, все числа разные  и отсортированы | 1, 2, 3, 4, 5, 6, 7, 8 |  |  |
| k = 2, все числа разные  и не отсортированы | 2, 3, 4, 1 |  |  |
| k = 2, все значения равны | 1, 1, 1, 1 |  |  |

Пояснения:

* Результат равен эталонному
* Результат равен эталонному, тк вершина 1 при присоединении будет наименьшей по значению, поэтому именно она станет главной вершиной
* Результат равен эталонному, тк в операции добавления биноминального дерева не возникает коллизий

Операция удаления:

|  |  |  |  |
| --- | --- | --- | --- |
| Условия тестирования | Начальные данные | Результат работы алгоритма | Эталонный результат работы алгоритма |
| k = 2, удаление нижних(  наибольших)  вершин | 5 1 6 7 |  |  |
| k = 2, все числа одинаковые | 1, 1, 1, 1 |  |  |
| k = 2, удаление промежуточной  вершины | 3, 6, 9, 4 |  |  |

Изменение вершины:

|  |  |  |  |
| --- | --- | --- | --- |
| Условия тестирования | Начальные данные | Результат работы алгоритма | Эталонный результат работы алгоритма |
| k = 2, изменение нижней(  наибольшей)  вершины | 7 2 4 9 |  |  |
| k = 2, изменение промежуточной  вершины на наименьшую | 1 2 3 4 |  |  |

Операция нахождения минимума и удаление минимального обьекта:

Данные операции были обьединены, тк по сути это одна и таже задача в биноминальной куче, только применяемые в разных ситуациях

|  |  |  |  |
| --- | --- | --- | --- |
| Условия  тестирования | Начальные данные | Результат работы алгоритма | Эталонный результат работы алгоритма |
| k = 3, удаляется первая 1 (главная вершина), происходит перестроение  (последнее добавленное дерево отсоединяется от новой главной вершины) | 1 1 1 1 1  1 1 1 |  |  |
| k = 2,  удалится главная вершина, а вторым удалится наименьшая вершина из 2 (текущая главная вершина и главная вершина присоединяемого дерева). В данном случае вершина 2 | 1, 2, 4, 3 |  |  |

Пояснения:

* Результат равен эталонному, тк удалилась 1 единица(главная вершины) и произошло перестроение, тк удалилась ссылка на главную вершину
* Результат равен эталонному, тк при 1 удалении удалилась главная вершина, отсоединилось дерево, но тк при отсоединении дерева она все равно остается в куче, то удалилось наименьшее из двух вершин. В данном случае вершина 2

**Задание №3:**

Основная функция данного текста – показать действия персонажа Артур по отношению к Форду. Была описана доменная модель, с учетом основных действий персонажей.

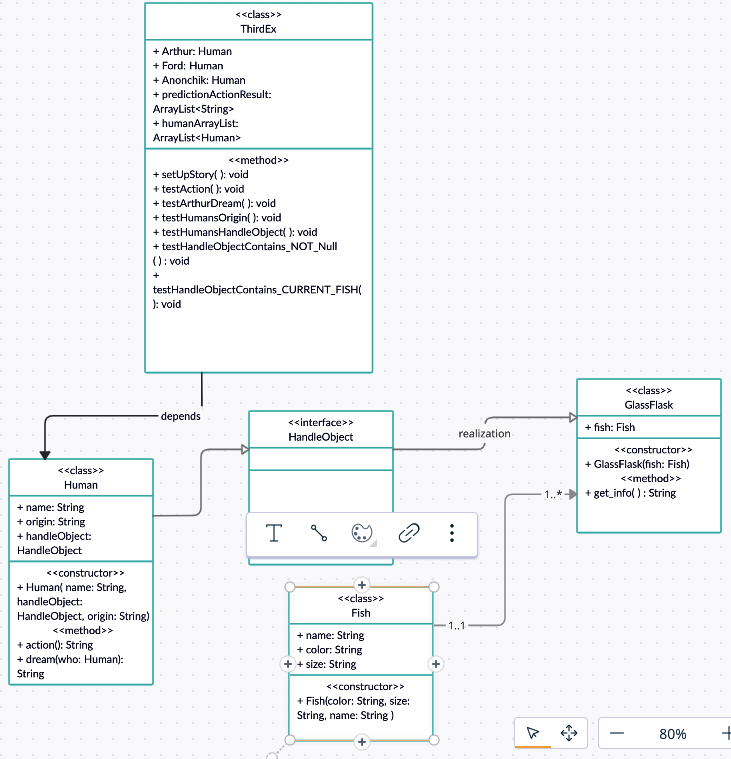
Для полного тестового покрытия потребовалось 6 тестовых методов, которые проверяют, совпадает ли результат работы программы с эталонным значением (в данном контексте – строка из истории). Из-за небольшой специфики данной истории (это касается ее размера) не получилось до конца раскрыть ее, что очень сильно повлияло на количество тестов.

Для проверки использовались методы:

Assert.assertEquals(), Assert.assertNull(), Assert.assertNotNull

[**Код**](https://github.com/Whatislove118/TPO_LABS/blob/master/src/com/lab1/ThirdEx.java)

Доменная модель:

****

**Вывод**

В ходе выполнения работы были изучены разные подходы к проведению тестирование, такие как метод белого и черного ящика. Также было изучено понятия тестового сценария и тестового случая.