

Национальный исследовательский университет информационных
технологий, механики и оптики

Факультет ПИиКТ

Тестирование программного обеспечения

Лабораторная работа №1

Вариант №3204

Студент

Курносова Ирина Викторовна

Группа Р33301

Преподаватель

Гаврилов Антон Валерьевич

г. Санкт-Петербург

2023

Задание

1. Для указанной функции провести модульное тестирование разложения функции в степенной ряд. Выбрать достаточное тестовое покрытие.
 - Функция $\arccos(x)$
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.
 - Программный модуль для работы с хеш-таблицей с разрешением коллизий методом цепочек (Hash Integer, <http://www.cs.usfca.edu/~galles/visualization/BucketSort.html>)
3. Сформировать доменную модель для заданного текста. Разработать тестовое покрытие для данной доменной модели
 - Артур, нервничая, вошел следом и был ошеломлен, увидев развалившегося в кресле человека, положившего ноги на пульт управления и ковыряющего левой рукой в зубах правой головы. Правая голова, казалось, была всецело занята этим, но зато левая улыбалась широко и непринужденно. Количество вещей, видя которые, Артур не верил своим глазам, все росло. Его челюсть отвисла.

Выполнение:

Часть 1. Функция $\arccos(x)$

$$\arccos(x) = \frac{\pi}{2} - \left[\sum_{n=1}^{\infty} \frac{1 * 3 * \dots * (2n-1) x^{2n+1}}{2 * 4 * \dots * 2n(2n+1)} \right]$$

```
public static double arccos(double x) {
    if (x > 1 || x < -1) {
        return Double.NaN;
    }

    int n = 1;
    double tmp = pow(x, n * 2 + 1) / (2 * n * (2 * n + 1));
    double res = tmp;
    n++;
    while (abs(tmp) > 0.000001) {
        tmp = tmp * pow(x, 2) * pow((2 * n - 1), 2) / ((2 * n) * (2 * n + 1));
        res += tmp;
        n++;
    }
    res += x;
    return PI / 2 - res;
}
```

В качестве эталонных значений при тестировании использовалась функция `Math.acos(x)`.
При сравнении результатов допускалась погрешность: $\delta = 0.01$

Результаты тестирования

✓ ArcCosTest (tests.one)	106 ms
✓ negativeAllowableDouble(double)	82 ms
✓ [1] -0.1	77 ms
✓ [2] -0.15	1 ms
✓ [3] -0.31	1 ms
✓ [4] -0.665	1 ms
✓ [5] -0.7001	1 ms
✓ [6] -0.95	1 ms
✓ positiveInt(int)	2 ms
✓ [1] 0	1 ms
✓ [2] 1	1 ms
✓ [3] 10	
✓ negativeInt(int)	2 ms
✓ [1] -1	1 ms
✓ [2] -2	1 ms
✓ [3] -10	
✓ positiveNearOneDouble(double)	5 ms
✓ [1] 0.9	
✓ [2] 0.91	1 ms
✓ [3] 0.99	1 ms
✓ [4] 0.9901	1 ms
✓ [5] 0.999999	1 ms
✓ [6] 1.000001	1 ms
✓ positiveAllowableDouble(double)	5 ms
✓ [1] 0.1	1 ms
✓ [2] 0.15	1 ms
✓ [3] 0.31	1 ms
✓ [4] 0.665	
✓ [5] 0.7001	1 ms
✓ [6] 0.95	1 ms
✓ negativeNearOneDouble(double)	4 ms
✓ [1] -0.9	1 ms
✓ [2] -0.91	
✓ [3] -0.99	1 ms
✓ [4] -0.9901	1 ms
✓ [5] -0.999999	1 ms
✓ [6] -1.000001	
✓ nearZeroDouble(double)	6 ms
✓ [1] -1.0E-4	1 ms
✓ [2] -1.0E-6	1 ms
✓ [3] -0.029	1 ms
✓ [4] 1.0E-4	1 ms
✓ [5] 1.0E-6	1 ms

Часть 2. Алгоритм формирования хеш-таблицы

Реализована функция сортировки массива путём составления хеш-таблицы. Поиск индекса элемента в хеш-таблице осуществляется по формуле:

$$index(n) = \frac{n * Max_count}{Max_value + 1}$$

Max_count – количество элементов массива

Max_value – максимальный элемент массива

```
private int index(double val) {  
    return (int) Math.floor(Math.abs(val) * size / (MAX_VALUE + 1));  
}
```

Структура самой хеш-таблицы представлена в виде вектора из коллекций SortedMap, где Double key – значение элемента, Integer value – количество его повторений

```
private Vector<SortedMap<Double, Integer>> collection;
```

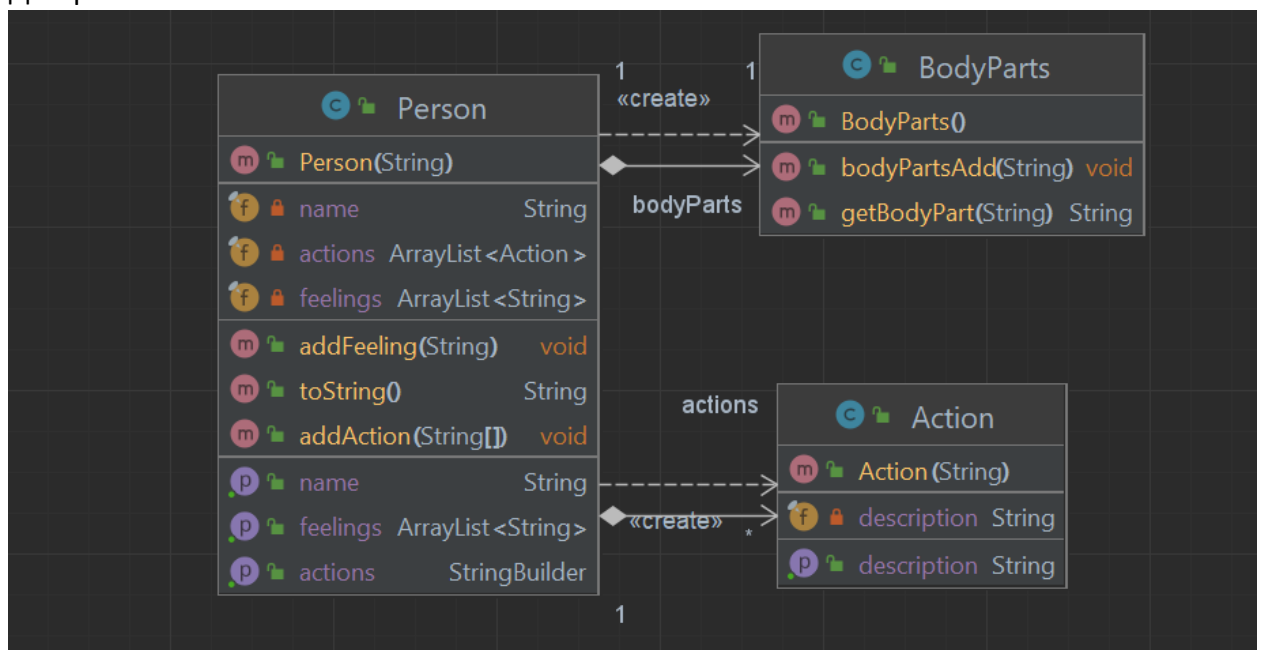
В результате функция возвращает отсортированный массив

Результаты тестирования

✓ HashTableTest (tests.two)	27 ms
✓ checkOneEl()	15 ms
✓ sort()	1 ms
✓ checkNull()	1 ms
✓ arrayRepeat()	3 ms
✓ maxVals()	7 ms

Часть 3. Доменная область

Диаграмма классов



Оригинальный текст	Текст программы
<p>Артур, нервничая, вошел следом и был ошеломлен, увидев развалившегося в кресле человека, положившего ноги на пульт управления и ковыряющего левой рукой в зубах правой головы. Правая голова, казалось, была всецело занята этим, но зато левая улыбалась широко и непринужденно. Количество вещей, видя которые, Артур не верил своим глазам, все росло. Его челюсть отвисла.</p>	<p>Артур вошёл Артур увидел Странный человек Артур не верил своим глаза Его челюсть отвисла [Артур нервничал, Артур был ошеломлён]</p> <p>Странный человек положил на стол ноги Странный человек ковырял левая рука в зубы правая голова правая голова была занята делом левая голова широко улыбалась</p>

Результаты тестирования:

✓ DomainTest (tests.three)	24 ms
✓ checkFeelings()	18 ms
✓ checkActions()	3 ms
✓ checkBodyPart()	
✓ checkToString()	3 ms

Вывод:

В ходе выполнения данной лабораторной работы я познакомилась с библиотекой Junit и реализовала модульное тестирование для различных алгоритмов. Для тестирования были использованы обычные (@Test) и параметризированные (@ParameterizedTest) тесты.