

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Мегафакультет: Компьютерных технологий и управления

Направление: 09.03.04 «Программная инженерия»

Лабораторная работа №1

По дисциплине:

«Тестирование программного обеспечения»

Вариант 33132

Выполнила:

студентка группы **Р33112**

Корнишова Евгения
Александровна

Преподаватель:

Харитоновна Анастасия
Евгеньевна

г. Санкт-Петербург

2021 г.

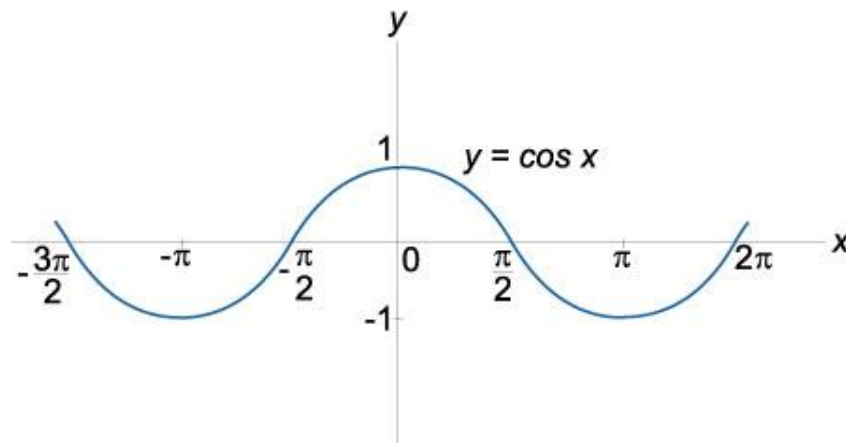
Текст задания:

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.

Код: https://github.com/jane-web-dev/TPO_Lab1

Вариант:

1. Функция $\cos(x)$



$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + \frac{(-1)^{n+1} x^{2n}}{(2n)!} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}, |x| < \infty,$$

Для тестирования были выбраны характерные точки косинуса: от $-\pi$ до π включительно. Тестирование за пределами этих точек при использовании разложения в степенной ряд занимает неприемлемо много времени, поэтому за пределами этого промежутка функция будет возвращать NaN.

При проведении модульного тестирования функции, вычисляющей косинус при помощи разложения в степенной ряд, использовался анализ эквивалентности — функция была разбита на классы эквивалентности, — участки, на которых она ведет себя схожим образом. Были протестированы значения функции на границах этих интервалов и внутри них.

Тестовый сценарий

Входное значение	Ожидаемое значение	Корректность
0.0	1.0	+
π	-1.0	+
$-\pi$	-1.0	+
$\frac{\pi}{2}$	0.0	+
$-\frac{\pi}{2}$	0.0	+
$\frac{\pi}{3}$	0.5	+
$-\frac{\pi}{3}$	0.5	+
$\frac{2\pi}{3}$	-0.5	+
$-\frac{2\pi}{3}$	-0.5	+
-3 π , 5 π , - π - 0.000001, π + 0.000001	NaN	+

Параметризованные тесты

Файл содержит пары <ожидаемое значение $\cos(x)$, x >

cos.csv

1.0,0.0

-1.0,3.14159265358979323846

-1.0,-3.14159265358979323846

0.0,1.57079632679489661923

0.0,-1.57079632679489661923

0.5,1.0471870792210857801758718

0.5,-1.0471870792210857801758718

-0.5,2.0943950814492444683747117436

-0.5,-2.0943950814492444683747117436

NaN,3.14159275358979323846

NaN,-3.14159275358979323846
NaN,31.4159275358979323846
NaN,-31.4159275358979323846

✓ Test Results	58 ms
✓ CosTest	58 ms
✓ test 0,3333PI value()	21 ms
✓ test -0,66666666PI value()	1 ms
✓ test 0,5PI value()	1 ms
✓ test -0,3333PI value()	1 ms
✓ [1] 1.0, 0.0	21 ms
✓ [2] -1.0, 3.14159265358979323846	1 ms
✓ [3] -1.0, -3.14159265358979323846	1 ms
✓ [4] 0.0, 1.57079632679489661923	1 ms
✓ [5] 0.0, -1.57079632679489661923	1 ms
✓ [6] 0.5, 1.0471870792210857801758718	1 ms
✓ [7] 0.5, -1.0471870792210857801758718	1 ms
✓ [8] -0.5, 2.0943950814492444683747117436	1 ms
✓ [9] -0.5, -2.0943950814492444683747117436	1 ms
✓ [10] NaN, 3.14159275358979323846	1 ms
✓ [11] NaN, -3.14159275358979323846	1 ms
✓ [12] NaN, 31.4159275358979323846	0 ms
✓ [13] NaN, -31.4159275358979323846	1 ms
✓ test -PI value()	0 ms
✓ test -0,5PI value()	0 ms
✓ test 0,66666666PI value()	0 ms
✓ test PI value()	1 ms
✓ test values more than PI and less than -PI return N	1 ms
✓ test 0 value()	0 ms

2. Программный модуль для работы с хеш-таблицей с закрытой адресацией (Hash Integer, <http://www.cs.usfca.edu/~galles/visualization/OpenHash.html>)

Закрытая адресация означает, что в случае коллизии будет формироваться список элементов с одинаковым хэшем внутри одной ячейки (бакета) таблицы.

В сценариях тестирования проверялась последовательность действий, проводимая внутри хэш-таблицы (тестирование методом белого ящика).

Тестовые сценарии

Сценарий	Корректность
Положить в таблицу два значения и потом извлечь их же Действия с таблицей размера 5: <ol style="list-style-type: none"> 1. вставка <5, 5> (бакет 0) 2. вставка <11, 11> (бакет 1) 3. поиск 5 	+

<p>4. поиск 11</p> <p>Ожидаемая последовательность: "working with bucket 0", "insert on step 0", "working with bucket 1", "insert on step 0", "working with bucket 0", "found on step 0", "working with bucket 1", "found on step 0"</p>	
<p>Положить в таблицу значение, удалить его и проверить отсутствие</p> <p>Действия с таблицей размера 5:</p> <ol style="list-style-type: none"> 1. вставка <5, 5> (бакет 0) 2. удаление 5 3. поиск 5 <p>Ожидаемая последовательность: "working with bucket 0", "insert on step 0", "working with bucket 0", "found on step 0", "working with bucket 0", "removed on step 0", "working with bucket 0", "not found in bucket"</p>	+
<p>Извлечение элементов при наличии коллизий</p> <p>Действия с таблицей размера 5:</p> <ol style="list-style-type: none"> 1. вставка <1, 1> (бакет 1) 2. вставка <2, 2> (бакет 2) 3. вставка <3, 3> (бакет 3) 4. вставка <4, 4> (бакет 4) 5. вставка <5, 5> (бакет 0) 6. вставка <6, 6> (бакет 1) 7. поиск 1 8. поиск 2 9. поиск 3 10. поиск 4 11. поиск 5 12. поиск 6 <p>Ожидаемая последовательность: "working with bucket 1", "insert on step 0", "working with bucket 2", "insert on step 0", "working with bucket 3", "insert on step 0", "working with bucket 4", "insert on step 0", "working with bucket 0", "insert on step 0", "working with bucket 1", "insert on step 1", "working with bucket 1", "found on step 0", "working with bucket 2", "found on step 0", "working with bucket 3", "found on step 0", "working with bucket 4", "found on step 0", "working with bucket 0", "found on step 0", "working with bucket 1", "found on step 1"</p>	+
<p>Извлечение элементов при наличии коллизии и предыдущего удаления элементов из ячеек с коллизиями</p> <p>Действия с таблицей размера 3:</p> <ol style="list-style-type: none"> 1. вставка <1, 1> (бакет 1) 2. вставка <2, 2> (бакет 2) 3. вставка <3, 3> (бакет 3) 4. вставка <4, 4> (бакет 4) 	+

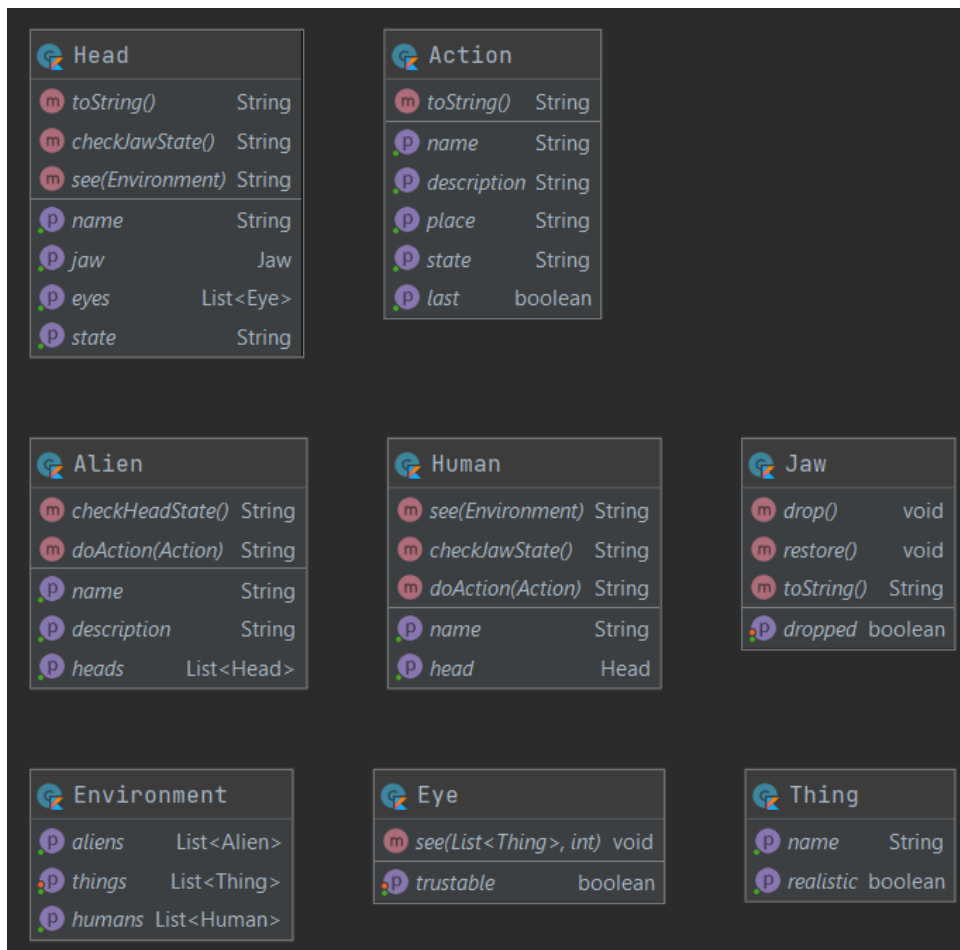
<p>5. вставка <5, 5> (бакет 0) 6. вставка <6, 6> (бакет 1) 7. вставка <7, 7> (бакет 2) 8. вставка <8, 8> (бакет 3) 9. вставка <9, 9> (бакет 4) 10. вставка <10, 10> (бакет 0) 11. удаление 1 12. удаление 2 13. удаление 3 14. удаление 4 15. удаление 5 16. удаление 6 17. поиск 7 18. поиск 8 19. поиск 9 20. поиск 10</p> <p>Ожидаемая последовательность: "working with bucket 1", "insert on step 0", "working with bucket 2", "insert on step 0", "working with bucket 3", "insert on step 0", "working with bucket 4", "insert on step 0", "working with bucket 0", "insert on step 0", "working with bucket 1", "insert on step 1", "working with bucket 2", "insert on step 1", "working with bucket 3", "insert on step 1", "working with bucket 4", "insert on step 1", "working with bucket 0", "insert on step 1", "working with bucket 1", "removed on step 0", "working with bucket 2", "removed on step 0", "working with bucket 3", "removed on step 0", "working with bucket 4", "removed on step 0", "working with bucket 0", "removed on step 0", "working with bucket 1", "removed on step 0", "working with bucket 2", "found on step 0", "working with bucket 3", "found on step 0", "working with bucket 4", "found on step 0", "working with bucket 0", "found on step 0"</p>	
<p>Извлечение элементов элемента из середины цепочки</p> <p>Действия с таблицей размера 1:</p> <p>1. вставка <1, 1> (бакет 1) 2. вставка <2, 2> (бакет 2) 3. вставка <3, 3> (бакет 3) 4. вставка <4, 4> (бакет 4) 5. вставка <5, 5> (бакет 0) 6. удаление 3 7. поиск 2 8. поиск 3 9. поиск 4</p> <p>Ожидаемая последовательность: "working with bucket 0", "insert on step 0", "working with bucket 0", "insert on step 1", "working with bucket 0", "insert on step 2",</p>	<p>+</p>

"working with bucket 0", "insert on step 3",
 "working with bucket 0", "insert on step 4",
 "working with bucket 0", "removed on step 2",
 "working with bucket 0", "found on step 1",
 "working with bucket 0", "not found in bucket",
 "working with bucket 0", "found on step 2"

✓ Test Results	41 ms
✓ HashTest	41 ms
✓ check work of put and get with collision()	36 ms
✓ put and get data from hash table()	0 ms
✓ check correct delete in the middle of sequence()	4 ms
✓ check work with collisions after delete()	1 ms
✓ check data absence after delete from table()	0 ms

3. Описание предметной области:

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.



Были составлены тестовые сценарии, которые проверяют состояние персонажей в зависимости от воздействующих на них внешних факторов.

Тестовые сценарии

Сценарий	Корректность
У Артура отвисает челюсть, если он видит слишком много нереалистичных вещей.	+
Артур не доверяет своим глазам после того, как увидит слишком много нереалистичных вещей.	+
Проверка и вывод в консоль основного сценария задания	+

✓ Test Results	387 ms
✓ DomainTest	387 ms
✓ artur jaw dropped after seeing too much unrealistic things()	339 ms
✓ main plot()	32 ms
✓ artur both eyes became untrusted after seeing too much unrealistic things()	16 ms

Вывод: в ходе выполнения лабораторной работы я закрепила знания, полученные на предмете МиСПИ на втором курсе, и разработала достаточные на мой взгляд тестовые покрытия для трех представленных задач.