

УНИВЕРСИТЕТ ИТМО

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Лабораторная работа №1

ФИО студентов:

Готовко Алексей Владимирович

Руденко Илья Александрович

Вариант: 235713

Направление подготовки: 09.03.04 (СППО)

Учебная группа: РЗЗ101

ФИО преподавателя: Машина Екатерина Алексеевна

Санкт-Петербург

2024г.

Содержание

1	Задание	2
2	Выполнение работы	2
2.1	Тестирование функции	2
2.2	Тестирование алгоритма	2
2.3	Тестирование доменной модели	3
2.4	Исходный код	4

1 Задание

1. Для указанной функции провести модульное тестирование разложения функции в степенной ряд. Выбрать достаточное тестовое покрытие.

Функция $\cos(x)$.

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, [visualization](https://www.cs.usfca.edu/~galles/visualization/BucketSort.html)).

<https://www.cs.usfca.edu/~galles/visualization/BucketSort.html>

3. Сформировать доменную модель для заданного текста. Разработать тестовое покрытие для данной доменной модели.

Описание предметной области:

Голова робота, сидящего в углу, сначала резко дернулась вверх, а затем едва заметно закачалась из стороны в сторону. Он тяжело поднялся на ноги и сделал то, что показалось бы постороннему наблюдателю героической попыткой пересечь комнату. Он остановился перед Триллиан и посмотрел, как будто, сквозь ее левое плечо.

2 Выполнение работы

2.1 Тестирование функции

Для функции $\cos(x)$ рассматривается разложение в ряд Тейлора до седьмого члена в точке $x_0 = 0$.

В тесте сравниваются значения функции `Math.cos()` и функции, считающей значение по вышеописанному ряду, в точках -2.046 , -1.0 , 0.0 , 1.0 , 1.28 с погрешностью 10^{-4} .

2.2 Тестирование алгоритма

Для тестирования имплементированного алгоритма был создан специальный класс `FunnyInteger`, каждая сущность которого характеризуется числом типа `int`. Функция `hashCode()` возвращает остаток от деления числа на 7.

Тестовые сценарии:

- Добавление значений в указанном порядке: 12, 3, 1, 10, 5, 12.

Ожидаемые пары `index` \rightarrow `bucket`:

1 \rightarrow 1,

3 \rightarrow 3, 10,

5 \rightarrow 5, 12.

- Добавление значений в указанном порядке: 32, 25, 18, 4, 11.

Ожидаемые пары `index` \rightarrow `bucket`:

4 \rightarrow 4, 11, 18, 25, 32.

- Добавление значений в указанном порядке: 7, 14, `null`, `null`.

Ожидаемые пары `index` \rightarrow `bucket`:

0 \rightarrow `null`, 7, 14.

- Добавление значений в указанном порядке: 14, 9, 2, 7, 10, null.

Удаление значений.

Ожидаемые пары Value to delete → was deleted:

14 → true,

14 → false,

1 → false.

Ожидаемые пары index → bucket:

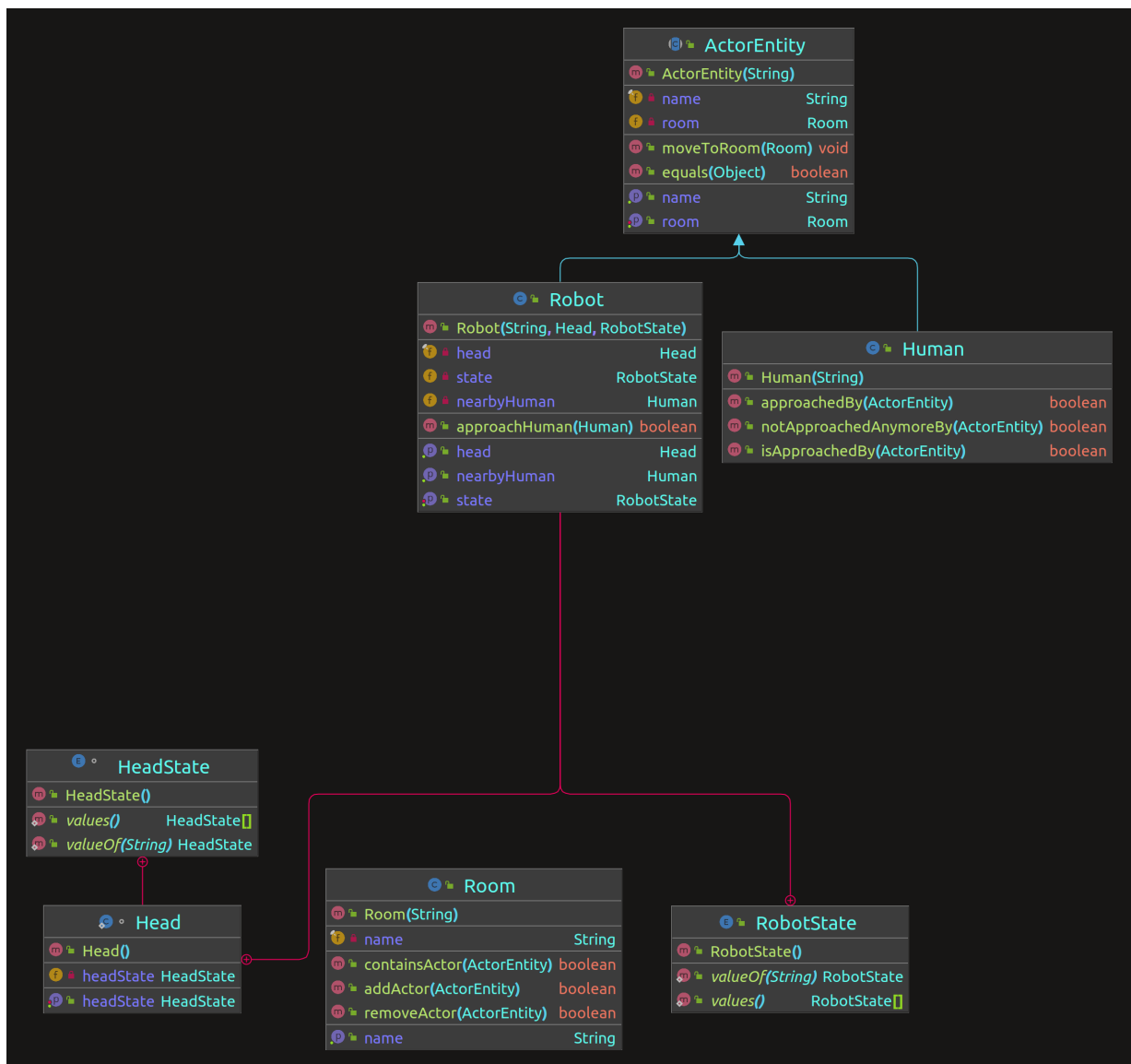
0 → null, 7,

2 → 2, 9,

3 → 10.

2.3 Тестирование доменной модели

UML-диаграмма классов доменной модели, построенной по данному тексту:



Код тестирования модели:

```
1 Room room = new Room("room");
2 Robot robot = new Robot(
3     "robot",
4     new Robot.Head(),
5     Robot.RobotState.SITTING
6 );
7 Human human = new Human("Trillian");
8
9 Assertions.assertFalse(robot.approachHuman(human));
10 Assertions.assertEquals(Robot.RobotState.SITTING, robot.getState());
11 Assertions.assertEquals(Robot.Head.HeadState.STILL, robot.getHead().getHeadState());
12 Assertions.assertNull(robot.getNearbyHuman());
13
14 robot.setRoom(room);
15 human.setRoom(room);
16 Assertions.assertTrue(room.containsActor(robot));
17 Assertions.assertTrue(room.containsActor(human));
18 Assertions.assertTrue(robot.approachHuman(human));
19 Assertions.assertEquals(Robot.RobotState.MOVING, robot.getState());
20 Assertions.assertEquals(Robot.Head.HeadState.SWINGING, robot.getHead().getHeadState());
21 Assertions.assertTrue(human.isApproachedBy(robot));
22 Assertions.assertEquals(human, robot.getNearbyHuman());
23
24 Assertions.assertTrue(robot.approachHuman(null));
25 Assertions.assertFalse(human.isApproachedBy(robot));
26 Assertions.assertEquals(Robot.RobotState.STAYING, robot.getState());
```

2.4 Исходный код

Доступен по [ссылке](https://github.com/xGodness/itmo/tree/main/3-year/software-testing/1-lab) (<https://github.com/xGodness/itmo/tree/main/3-year/software-testing/1-lab>).