

Лабораторная работа #1

Тестирование программного обеспечения

Группа: РЗ3101

Студенты: Бухаров Д. П., Никашкин А. В.

Преподаватель: Машина Е. А.

Вариант №235723

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

Задание:

1. Функция $\arccos(x)$
2. Программный модуль для обхода ориентированного графа методом поиска в глубину (<http://www.cs.usfca.edu/~galles/visualization/DFS.html>)
3. Описание предметной области:

И поскольку это далеко не самое естественное положение для кита, то у этого не очень мало времени на то, чтобы успеть свыкнуться с осознанием того, что оно к пришло, чтобы свыкнуться с осознанием того, что оно уже больше не кит.



Выполнение:

Были реализованы классы и методы, обеспечивающие подсчет функции $\arccos(x)$ через степенной ряд, алгоритм обхода графа в глубину DFS и разработана доменная модель по описанию области.

Код нахождения $\arccos(x)$:

```

public static double acos(double value) {
    if(Double.isNaN(value)||Math.abs(value)>1) {
        return Double.NaN;
    }

    double acos=0;
    double epsilon=0.001;
    double x=Math.abs(value);
    double prev;

    long kFact=1;
    long kFact2=1;
    int k=0;

    while(true) {
        prev=acos;
        acos+=(Math.pow(2,-k)*Math.pow(1-x,k)*kFact2/Math.pow(2,k))/
            (kFact+2*k*kFact);

        k++;
        kFact*=k;
        kFact2=k+(k%2==0?1:0);

        if(Math.abs(acos-prev)<epsilon){
            break;
        }
    }

    if(value>0) {
        return acos*Math.sqrt(2-2*x);
    }
    return Math.PI-acos*Math.sqrt(2)*Math.sqrt(1-x);
}

```

Исходный код arcos

Исходный код dfs

Исходный код доменной модели

Код junit тестов:

```

@ParameterizedTest
@ValueSource(doubles = {-1.1, -10.0})
@DisplayName("check negative values")
public void checkNegativeValues(double value){
    Assertions.assertEquals(Math.acos(value),CustomMath.acos(value),epsilon);
}

```

Тесты `arcos`

Тесты `dfs`

Тесты доменной модели

Вывод:

В ходе выполнения данной лабораторной работы я познакомился с `junit` тестами, написал подсчет функции `arcos(x)`, поиска в глубину `dfs` и доменную модель по тексту, покрыл весь код `junit` тестами. Использовал параметризованные, автоматические, повторяющиеся тесты и различные виды `assert` 'ов. Уверен, полученная информация пригодится мне в будущем.