Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Факультет информационных технологий и кибербезопасности

Кафедра информационных систем и программирования

**ОТЧЕТ**

по лабораторной работе №12.

Тема работы: «Тестирование программ при отсутствии исходного кода методом черного ящика».

Наименование дисциплины: «Тестирование и отладка программного обеспечения».

Подготовил:

Студент группы 23-КБ-ПР1

Булгаков В .В.

Краснодар

2025

**Цель работы**

Цель работы – закрепить принципы тестирования методом черного ящика и использования внешних библиотек.

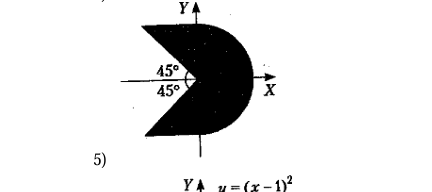
**Задание**

1) Подключить внешнюю библиотеку к новому проекту.

2) В код части классов библиотеки целенаправленно были внесены одна или несколько ошибок. Часть классов работает корректно.

3) Создать проект с модульными тестами для подключенной библиотеки и написать тесты для класса Area реализующей проверку попадания в закрашенную область фигуры в соответствии с вариантом задания из пункта 5. Необходимо обнаружить ошибки в коде (если они есть) и указать при каких условиях они возникают.

4) Оформить отчёт.



**Ход работы**

1-3) Создал внешнюю библиотеку (npm-пакет), написав класс Area.js, собрав библиотеку с помощью webpack. Упаковал библиотеку в .tgz – npm-пакет, который можно подключить в другой проект.

class Area {

    constructor(R) {

        this.R = R;

    }

    IsPointInArea(x, y) {

        if (x \* x + y \* y > this.R \* this.R) {

            return false;

        }

        if (x >= 0) {

            return true;

        }

        if ((y >= -x && y >= 0) || (y < -x && y < 0)) {

            return true;

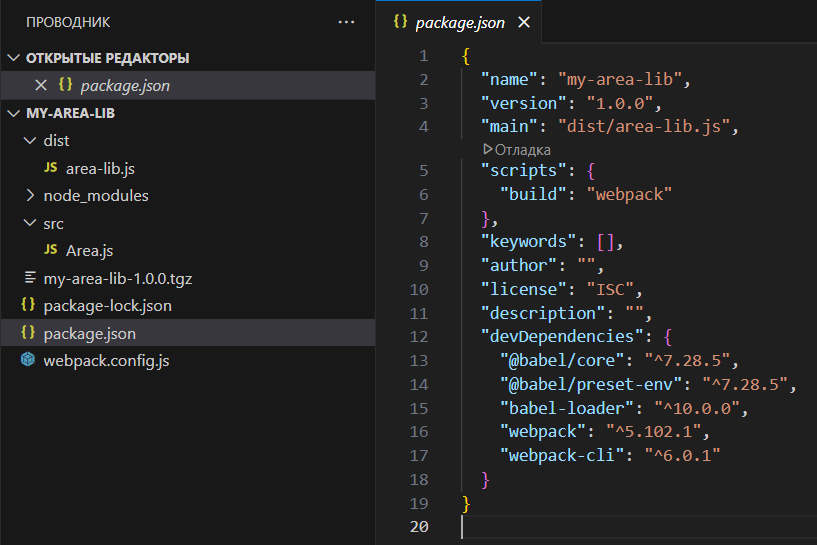
        }

        return false;

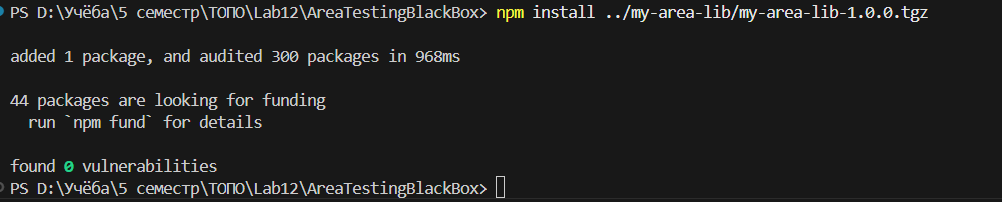
    }

}

module.exports = Area;



Создал новый проект, установил внешнюю созданную библиотеку



Определил классы эквивалентности:

1. Правильные классы эквивалентности.

a. Закрашенная область (включая точки на окружности и на фигуре).

1. Неправильные классы эквивалентности.

a. Всё, что не попадает в закрашенную область.

Написал тесты для подключенной библиотеки согласно классом эквивалентности.

const { describe, test, beforeEach, expect } = require('@jest/globals');

const {AreaLib} = require('my-area-lib');

describe('AreaLib', () => {

    let Area;

    beforeEach(() => {

        Area = new AreaLib(5);

    });

    describe('Точки, входящие в область', () => {

        test('Центр (0, 0)', () => {

            expect(Area.IsPointInArea(0, 0)).toBe(true);

        });

        test('Правая полуплоскость: (3, 3)', () => {

            expect(Area.IsPointInArea(3, 3)).toBe(true);

        });

        test('Правая полуплоскость: (5, 0) — на границе', () => {

            expect(Area.IsPointInArea(5, 0)).toBe(true);

        });

        test('Левая полуплоскость, верхний сектор: (-3, 3)', () => {

            expect(Area.IsPointInArea(-3, 3)).toBe(true);

        });

        test('Левая полуплоскость, верхний сектор: (-5, 0)', () => {

            expect(Area.IsPointInArea(-5, 0)).toBe(false);

        });

        test('Левая полуплоскость, нижний сектор: (-4, -3)', () => {

            expect(Area.IsPointInArea(-4, -3)).toBe(true);

        });

        test('Левая полуплоскость, нижний сектор: (0, -5) — на границе', () => {

            expect(Area.IsPointInArea(0, -5)).toBe(true);

        });

    });

    describe('Точки, не входящие в область', () => {

        test('Вне круга: (6, 0)', () => {

            expect(Area.IsPointInArea(6, 0)).toBe(false);

        });

        test('Вне круга: (-6, 0)', () => {

            expect(Area.IsPointInArea(-6, 0)).toBe(false);

        });

        test('Левая полуплоскость, верхний сектор: (-5, 5) — вне сектора', () => {

            expect(Area.IsPointInArea(-5, 5)).toBe(false);

        });

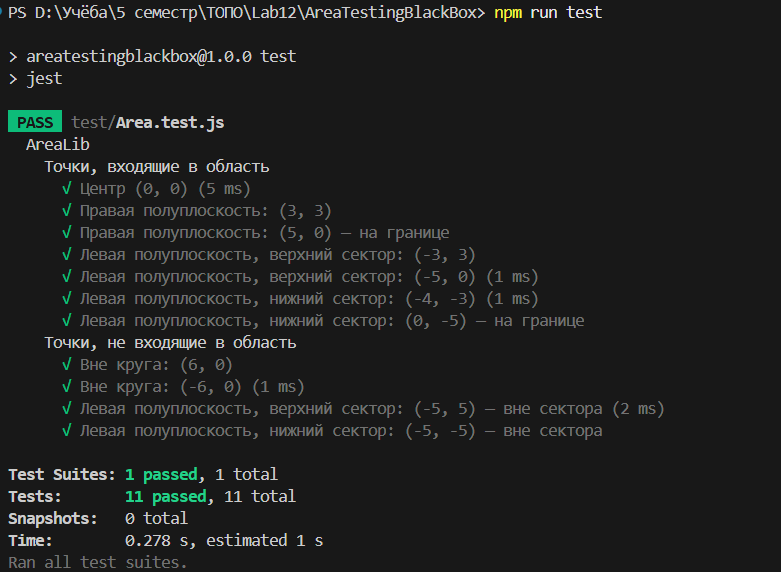
        test('Левая полуплоскость, нижний сектор: (-5, -5) — вне сектора', () => {

            expect(Area.IsPointInArea(-5, -5)).toBe(false);

        });

    });

});

****

**5 Вывод**

В ходе выполнения данной лабораторной работы удалось закрепить принципы тестирования методом черного ящика и использования внешних библиотек.