Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

**ОТЧЕТ**

по лабораторной работе №5.

Тема работы: «Тестирование, управляемое данными

(Data-Driven Unit Tests) и Анализ покрытия кода (Code Coverage)».

Наименование дисциплины: «Тестирование и отладка программного обеспечения».

Подготовил:

Студент группы 23-КБ-ПР1

Булгаков В. В.

Краснодар

2025

**Цель работы**: изучить подход к автоматизации процесса тестирования с помощью средств среды разработки Microsoft Visual Studio, а также понятие покрытия кода (Code Coverage) тестами и научиться использовать на практике средства автоматизации определения покрытия.

**Задание:**

1) Используя класс из задания «Лабораторной работы №2. Тестирование методом черноного ящика», реализующий проверку принадлежности точки к области и тестовые наборы из той же работы создать проект тестирования (или добавить еще один тестовый класс в уже используемый тестовый проект) с набором тестов управляемых данными.

2) Запустить проект тестирования и проверить результаты работы.

3) Проанализировать покрытие кода тестами.

4) На основе тестов здания «Лабораторной работы №3. Тестирование методом белого ящика» проанализировать покрытие кода тестами.

5) Составить отчет о результатах проведенного тестирования.

**Ход работы**:

1-2) Добавил класс тестирования в уже используемый проект. Использовал test.each для тестирования, управляемого данными. Проверил результаты работы.

const checkPoint = require('../src/point');

describe('checkPoint - Тестирование, управляемое данными', () => {

    const R = 5;

    const testCases = [

        // Класс 1: точка в 1 области

        [0, 0, '1 область', 'центр круга'],

        [-2, 2, '1 область', 'ниже линии y = x + R'],

        [0, R, '1 область', 'на границе круга (ось Y)'],

        [-R, 0, '1 область', 'на границе круга (ось X)'],

        // Класс 2: точка в 2 области

        [-2, 4, '2 область', 'выше линии y = x + R'],

        [3, -4, '2 область', 'в четвёртом квадранте'],

        [4, -3, '2 область', 'в четвёртом квадранте, на границе'],

        // Класс 3: точка в 3 области

        [6, 0, '3 область', 'вне круга, правее'],

        [0, 6, '3 область', 'вне круга, выше'],

        [4, 4, '3 область', 'вне круга, в первом квадранте'],

        // Класс 4: точка на границе между 1 и 2

        [-1, 4, '1 область', 'на линии y = x + R, относится к 1 области'],

        [-2.5, 2.5, '1 область', 'ещё одна точка на линии y = x + R'],

        // Класс 5: точка на границе между 2 и 3

        [3, -4, '2 область', 'на границе круга, в четвёртом квадранте'],

        [-3.5355339059327378, 3.5355339059327378, '2 область', 'на границе круга, во втором квадранте, выше линии']

    ];

    test.each(testCases)('TestPoint(%i, %i) = %s (%s)', (x, y, expected, description) => {

        const point = new checkPoint(R);

        expect(point.TestPoint(x, y)).toBe(expected);

    });

    // Граничные значений

    const boundaryTestCases = [

        [0, R - 0.001, '1 область', 'внутри круга, близко к границе круга'],

        [0, R + 0.001, '3 область', 'вне круга, близко к границе круга'],

        [-2, 2.999, '1 область', 'внутри круга, близко к линии y = x + R'],

        [-2, 3.001, '2 область', 'выше линии y = x + R, близко к границе'],

        [-1, 3.999, '1 область', 'чуть ниже линии y = x + R'],

        [-1, 4.001, '2 область', 'чуть выше линии y = x + R'],

    ];

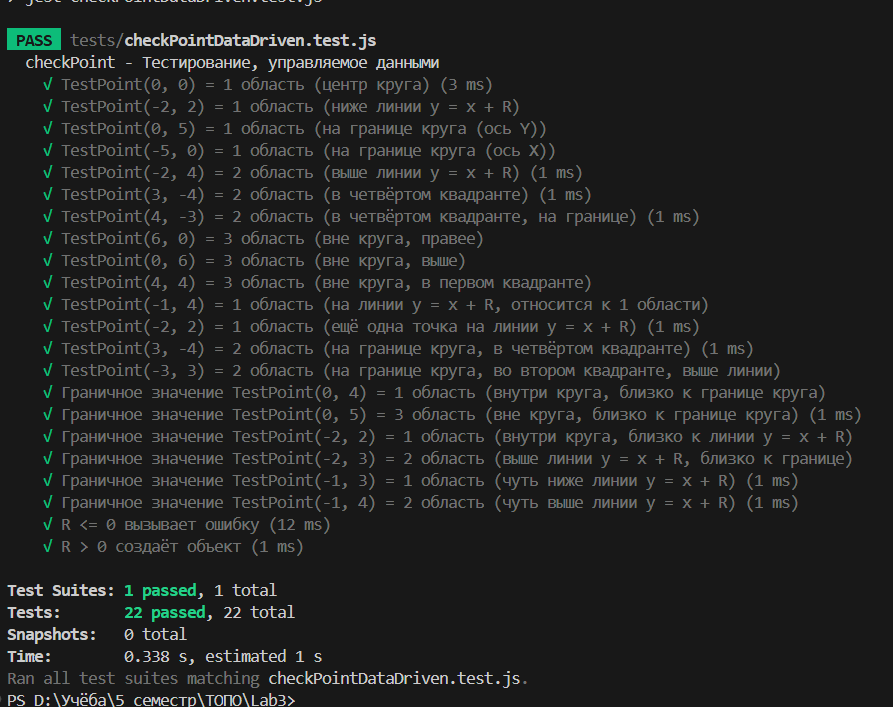
    test.each(boundaryTestCases)('Граничное значение TestPoint(%i, %i) = %s (%s)', (x, y, expected, description) => {

        const point = new checkPoint(R);

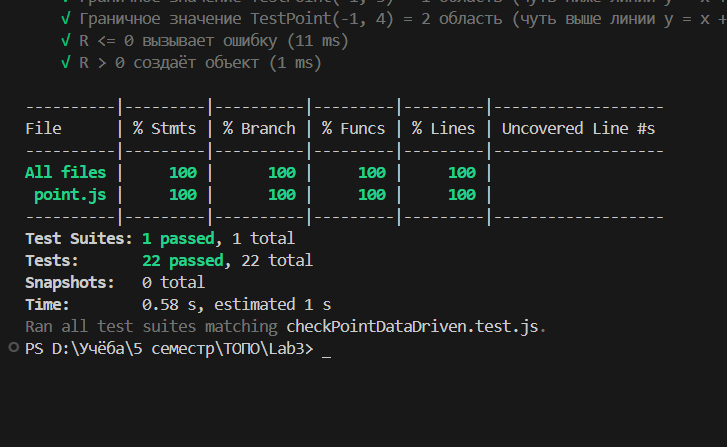
        expect(point.TestPoint(x, y)).toBe(expected);

    });

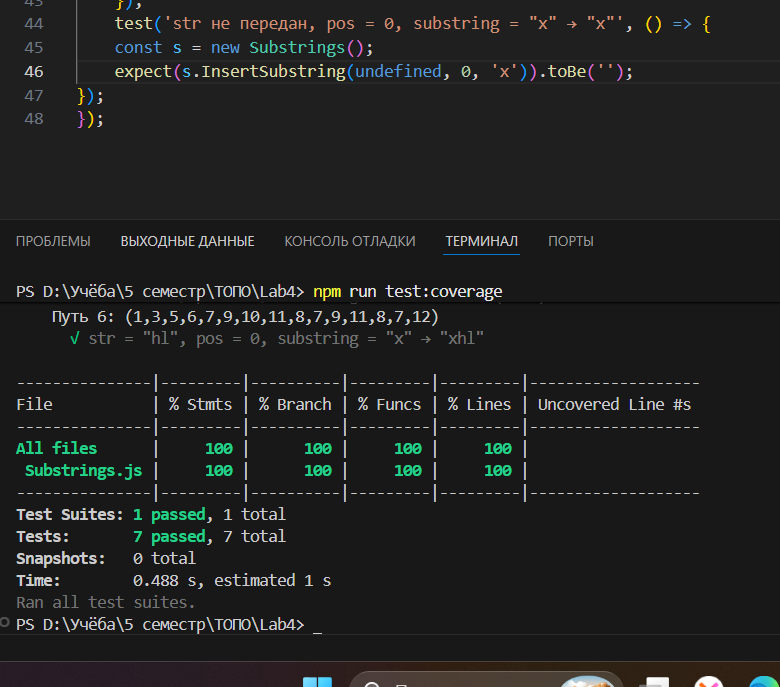
});



3) Проанализировал покрытие кода тестами.



4) Проанализировал покрытие кода тестами из задания “Тестирование методом белого ящика”. Также добавлен один test, где str не передан



Вывод: изучен подход к автоматизации процесса тестирования с помощью средств среды разработки Microsoft Visual Studio, а также понятие покрытия кода (Code Coverage) тестами и был обретён навык использования на практике средств автоматизации определения покрытия.