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 №6.

Тема работы: «Тестирование методом серого ящика».

Наименование дисциплины: «Тестирование и отладка программного обеспечения».

Подготовил:

Студент группы 23-КБ-ПР1

Булгаков В. В.

Краснодар

2025

**Цель работы**

Цель работы – изучить подход к тестированию методом серого ящика.

**Задание**

1)Создать класс (в соответствии с вариантом задания из п.5), реализующий преобразование строки из одной системы счисления в другую.

2) Протестировать класс на основе метода серого ящика с использованием средств автоматизации.

3) Составить отчет о результатах проведенного тестирования.

Вариант: перевод из троичной системы счисления в шестнадцатеричную систему счисления.

**Ход работы**

1) Создал класс, реализующий поставленное задание.



2) Выбрал классы эквивалентности, граничные значения и значения путей для прохождения всех критериев покрытия белого ящика.

Правильные классы эквивалентности:

1. Строка: 0-9, A-B.

Неправильные классы эквивалентности:

1. Строка не из перечня: 0-9, A-B.

2. Не строка.

3. Пустая строка.

4. Строка > Number.MAX\_SAFE\_INTEGER

Граничные значения: ‘0’, ‘1’, ‘B’, ‘(Number.MAX\_SAFE\_INTEGER) (702273685B77A27)’, ‘702273685B77A28, ‘702273685B77A26.

Таблица 1 – Пути

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Путь** | **Input** | **Результат** | **Комментарий** |
| 1 | 1,2 | null | Error: Input должен быть не пустой строкой | Невалидный тип данных |
| 2 | 1,3,4,5,6 | “C” | Error: Некорректные входные данные | Символ вне диапазона 0-9, A-B |
| 3 | 1, 3, 4,5,7,8,9,10,12,13,14,16,17,19,11, 10,20 | “0” | “0” | Минимальное значение |
| 4 | 1, 3, 4,5,7,8,9,10,12,13,15,16,17, 19,10, 11, 12,13,14,16,17,19, 11, 10, 20, 21,22,23, 24,25,23,24,25,26 | "1B" | “17” | Пример |
| 5 | 1, 3, 4, 5,7,8,9,10,12,13,14,16,17…18 | "702273685B77A28" | Error: Превышение Number.MAX\_SAFE\_INTEGER | Ошибка |

Таблица 2 – Критерии

|  |  |  |
| --- | --- | --- |
| **№** | **Критерий** | **Пути** |
| 1 | Покрытие операторов | 1, 2, 4 |
| 2 | Покрытие решений | 1, 2, 3, 4, 5 |
| 3 | Покрытие условий | 1, 2, 3, 4, 5 |
| 4 | Покрытий путей | 1, 2, 3, 4, 5 |

const BaseConverter = require('../src/Converter');

describe('Converter', () => {

    let converter;

    beforeEach(() => {

        converter = new BaseConverter();

    });

    describe('Правильные классы эквивалентности', () => {

        test('Строка: 0-9, A-B', () => {

            expect(converter.from12To16('0')).toBe('0');

            expect(converter.from12To16('1')).toBe('1');

            expect(converter.from12To16('9')).toBe('9');

            expect(converter.from12To16('A')).toBe('A');

            expect(converter.from12To16('B')).toBe('B');

        });

    });

    describe('Неправильные классы эквивалентности', () => {

        test('Строка не из перечня: 0-9, A-B', () => {

            expect(() => converter.from12To16('C')).toThrow('Некорректные входные данные');

            expect(() => converter.from12To16('XYZ')).toThrow('Некорректные входные данные');

            expect(() => converter.from12To16('1G')).toThrow('Некорректные входные данные');

        });

        test('Не строка', () => {

            expect(() => converter.from12To16(null)).toThrow('Input должен быть не пустой строкой');

            expect(() => converter.from12To16(undefined)).toThrow('Input должен быть не пустой строкой');

            expect(() => converter.from12To16(123)).toThrow('Input должен быть не пустой строкой');

            expect(() => converter.from12To16({})).toThrow('Input должен быть не пустой строкой');

        });

        test('Пустая строка', () => {

            expect(() => converter.from12To16('')).toThrow('Input должен быть не пустой строкой');

        });

        test('Строка > Number.MAX\_SAFE\_INTEGER', () => {

            expect(() => converter.from12To16('702273685B77A28')).toThrow('Превышение Number.MAX\_SAFE\_INTEGER');

        });

    });

    describe('Граничные значения', () => {

        test('"0"', () => {

            expect(converter.from12To16('0')).toBe('0');

        });

        test('"1"', () => {

            expect(converter.from12To16('1')).toBe('1');

        });

        test('"B"', () => {

            expect(converter.from12To16('B')).toBe('B');

        });

        test('"702273685B77A27" (Number.MAX\_SAFE\_INTEGER)', () => {

            expect(converter.from12To16('702273685B77A27')).toBe('1FFFFFFFFFFFFF');

        });

        test('"702273685B77A28" (Number.MAX\_SAFE\_INTEGER + 1)', () => {

            expect(() => converter.from12To16('702273685B77A28')).toThrow('Превышение Number.MAX\_SAFE\_INTEGER');

        });

        test('"702273685B77A26" (Number.MAX\_SAFE\_INTEGER - 1)', () => {

            expect(converter.from12To16('702273685B77A26')).toBe('1FFFFFFFFFFFFE');

        });

    });

    describe('Пути', () => {

        describe('Путь 1: input = null → Error', () => {

            test('input = null → Error', () => {

                expect(() => converter.from12To16(null)).toThrow('Input должен быть не пустой строкой');

            });

        });

        describe('Путь 2: input = "C" → Error', () => {

            test('input = "C" → Error', () => {

                expect(() => converter.from12To16('C')).toThrow('Некорректные входные данные');

            });

        });

        describe('Путь 3: input = "0" → "0"', () => {

            test('input = "0" → "0"', () => {

                expect(converter.from12To16('0')).toBe('0');

            });

        });

        describe('Путь 4: input = "1B" → "17"', () => {

            test('input = "1B" → "17"', () => {

                expect(converter.from12To16('1B')).toBe('17');

            });

        });

        describe('Путь 5: input = "702273685B77A28" → Error', () => {

            test('input = "702273685B77A28" → Error', () => {

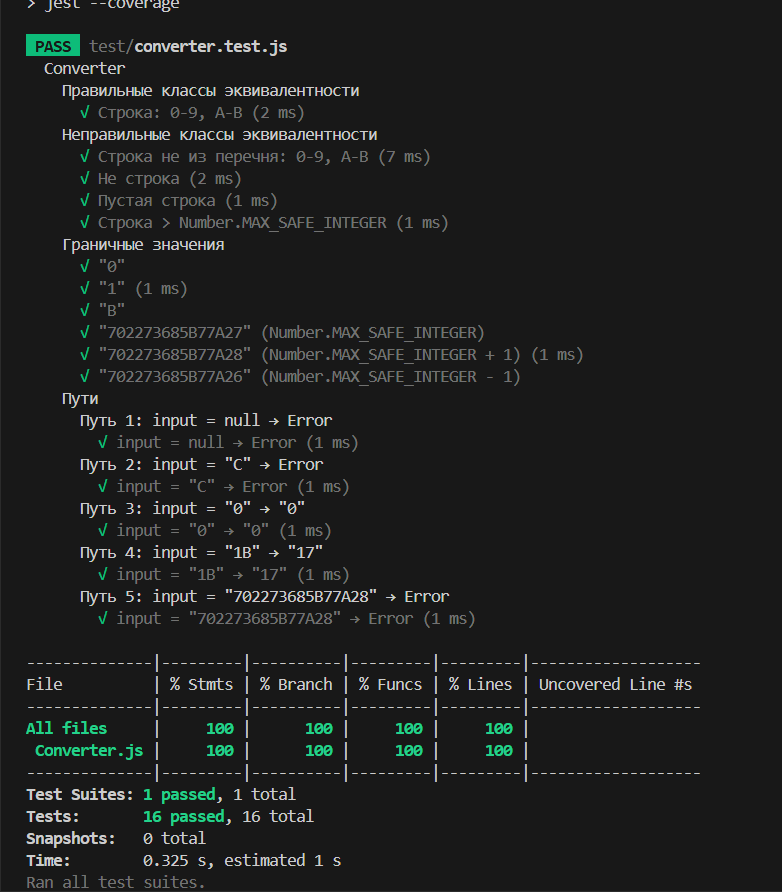
                expect(() => converter.from12To16('702273685B77A28')).toThrow('Превышение Number.MAX\_SAFE\_INTEGER');

            });

        });

    });

});



**Вывод**

В ходе выполнения данной лабораторной работы удалось изучить подход к тестированию методом серого ящика.