Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Звіт про виконану роботу до лабораторного заняття № 1

*з курсу*

*Автоматизація тестування*

***Виконав/ла:*** студентка групи41ІПЗ

спеціальності «Інженерія програмного

забезпечення»

Антосевич К.О.

***Викладач:*** Кархут В.Я.

Створіть базовий проект npm init –y Встановіть бібліотеку jest npm install --dev jest Встановіть jest як тестову команду по замовчуванню "scripts": { "test": "jest --verbose" }, 2. Створіть файл math\_operations.js до якого додайте чотири функції для основних математичних операцій – додавання, віднімання, множення, ділення. 3. Створіть math\_operations.test.js в якому для кожної математичної функції напишіть по два юніт тести. Включіть випадки з використання від’ємних та дробових чисел. 4. Створіть math\_operations\_params.test.js в якому створіть параметризовану версію тестів. Додайте як мінімум по три параметри для кожної функції. Використайте метод test.each 5. Створіть файл array\_utils.js в який додайте функції: • Яка приймає масив та повертає число що є сумою елементів масиву; • Яка приймає масив та повертає відфільтрований масив що містить тільки позитивні числа; • Яка приймає масив та повертає відфільтрований масив що містить тільки негативні числа 6. Створіть файл array\_utils.tests.js в якому напишіть тести для вказаних функцій. Використайте функцію beforeAll яка буди створювати тестовий масив перед кожним тестом. 7. Створіть string\_operations.test.js в який додайте функції: • Яка перевіряє чи є текстовий фрагмент паліндромом; • Яка перевіряє чи є текстовий фрагмент анаграмою; 8. Створіть grouped.test.js в якому з використанням функції describe створіть дві групи тестів: • Для файлу array\_utils.js; • Для файлу string\_operations.test.js; 9. Створіть функції та тести до них які перевіряють: • Чи містить масив вказаний об’єкт • Чи містить текстовий фрагмент вказане слово • Чи містить об’єкт який повертає функція вказане поле • Функцію яка приймає масив та зворотній виклик який виконується над елементами масиву. Напишіть тест який перевіряє кількість запусків зворотного виклику. • Функція яка приймає текстовий рядок із фразою та зворотній виклик. Функція перетворює фразу на масив слів передає її у зворотній виклик. Протестуйте з якими аргументами викликається зворотній виклик. • Створіть об’єкт який містить поле імені та поле прізвища, додайте в об’єкт метод introduce який виводить у консоль фразу «Hello, I’m firstname lastname». З використання jest.spyOn преревірте кількість викликів методу

Package.json

{

"name": "test-project",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "jest --verbose"

},

"jest": {

"testEnvironment": "node"

},

"author": "",

"license": "ISC",

"devDependencies": {

"jest": "^26.6.3"

}

}

Math\_operatoins.js

function add(a, b) {

return a + b;

}

function subtract(a, b) {

return a - b;

}

function multiply(a, b) {

return a \* b;

}

function divide(a, b) {

if (b === 0) throw new Error("Division by zero");

return a / b;

}

module.exports = { add, subtract, multiply, divide };

math\_operations.test.js

const { add, subtract, multiply, divide } = require('./math\_operations');

test('adds 1 + 2 to equal 3', () => {

expect(add(1, 2)).toBe(3);

});

test('subtracts 2 - 1 to equal 1', () => {

expect(subtract(2, 1)).toBe(1);

});

test('multiplies 2 \* 3 to equal 6', () => {

expect(multiply(2, 3)).toBe(6);

});

test('divides 10 / 2 to equal 5', () => {

expect(divide(10, 2)).toBe(5);

});

test('division by zero throws error', () => {

expect(() => divide(5, 0)).toThrow("Division by zero");

});

array\_utils.js

function sumArray(array) {

return array.reduce((sum, num) => sum + num, 0);

}

function filterPositive(array) {

return array.filter(num => num > 0);

}

function filterNegative(array) {

return array.filter(num => num < 0);

}

module.exports = { sumArray, filterPositive, filterNegative };

array\_utils.tests.js

const { sumArray, filterPositive, filterNegative } = require('./array\_utils');

describe('Array Utilities', () => {

let array;

beforeAll(() => {

array = [1, -1, 2, -2, 3, -3, 0];

});

test('sum of array', () => {

expect(sumArray(array)).toBe(0);

});

test('filter positive numbers', () => {

expect(filterPositive(array)).toEqual([1, 2, 3]);

});

test('filter negative numbers', () => {

expect(filterNegative(array)).toEqual([-1, -2, -3]);

});

});