Отчет

по «Качеству программного обеспечения»

Практическая работа № 3

Выполнил: Чумаченко Михаил

Задание к практической работе:

1. Используя систему тестирования UniTesk сделать пробные наборы тестовых сценариев к лаб. 1 (Но так как UniTesk это классическое, но при этом доисторическое средство – то можно заменить его на любую современную альтернативу, просто разберите как оно работает.)
2. Сделать таблицу контроля выполнения тестовых сценариев
3. Заполнить таблицу согласно полученных результатов

Ход работы:

Для тестирования был выбран Jest, как один из самых популярных фреймворков для тестирования JavaScript. Было написано 8 тестовых сценариев, которые рассматривают все возможные edge-кейсы функции расчета абонентской платы за телефон.

Для тестирования функция была вынесена в отдельный файл.

**index.js**

function calc(tariff, t, K, A, B, C, D){

let to\_pay = -1;

if(tariff === 1){

if(t <= 0 || K <= 0 || A <= 0 || B <= 0){ // If one of the values below zero - display error

return to\_pay;

}

if(t <= K){ // If t <= k - amount = A

to\_pay = A;

}

else{ // if t > K - amount = A + (t-K) \* B

let delta = t - K;

let extraPayment = delta \* B;

to\_pay = A + extraPayment;

}

return to\_pay;

}

if(tariff === 2){

if(t <= 0 || K <= 0 || C <= 0 || D <= 0){ // If one of the values below zero - display error

return to\_pay;

}

if(t <= K){ // If t <= k - amount = C \* t

to\_pay = C \* t;

}

else{ // If t > k - amount = D \* t

to\_pay = D \* t;

}

return to\_pay;

}

return -1;

}

module.exports = calc;

**index.test.js**

const calc = require('./index');

test( 'if tariff !== 1 || !== 2 return -1 v1', () => {

const expectation = -1;

const result = calc(3, 1,1,1,1,1, 1);

expect(result).toBe(expectation);

})

test( 'if tariff !== 1 || !== 2 return -1 v2', () => {

const expectation = -1;

const result = calc(-1, 1,1,1,1,1, 1);

expect(result).toBe(expectation);

})

test('if tariff === 1 variables t, k, a, b should not be less then 0', ()=> {

const expectation = -1;

const result = calc(1, 0,0,0,0,0, 0);

expect(result).toBe(expectation);

})

test('if tariff === 2 variables t, k, c, d should not be less then 0', ()=> {

const expectation = -1;

const result = calc(2, 0,0,0,0,0, 0);

expect(result).toBe(expectation);

})

test('if tariff === 1 and t <= K should return A', ()=> {

const expectation = 234;

const result = calc(1, 1,100,234, 1,0, 0);

expect(result).toBe(expectation);

})

test('if tariff === 1 and t > K should return (A + (t-k) \* B)', ()=> {

const expectation = 235;

const result = calc(1, 101,100,234, 1,0, 0);

expect(result).toBe(expectation);

})

test('if tariff === 2 and t <= K should return C \* t', ()=> {

const expectation = 100;

const result = calc(2, 100,100,0, 0,1, 1);

expect(result).toBe(expectation);

})

test('if tariff === 2 and t >= K should return D \* t', ()=> {

const expectation = 202;

const result = calc(2, 101,100,0, 0,1, 2);

expect(result).toBe(expectation);

})

Для запуска тестов необходимо написать команду “jest” в консоли. После запуска тестов, получаем результаты тестирования, которые отображены в таблице 1.

|  |  |
| --- | --- |
| Название тестового сценария | Результат теста |
| if tariff !== 1 || !== 2 return -1 v1 | Success |
| if tariff !== 1 || !== 2 return -1 v2 | Success |
| if tariff === 1 variables t, k, a, b should not be less then 0 | Success |
| if tariff === 2 variables t, k, c, d should not be less then 0 | Success |
| if tariff === 1 and t <= K should return A | Success |
| if tariff === 1 and t > K should return (A + (t-k) \* B) | Success |
| if tariff === 2 and t <= K should return C \* t | Success |
| if tariff === 2 and t >= K should return D \* t | Success |

Таблица 1 – Результат выполнения тестов

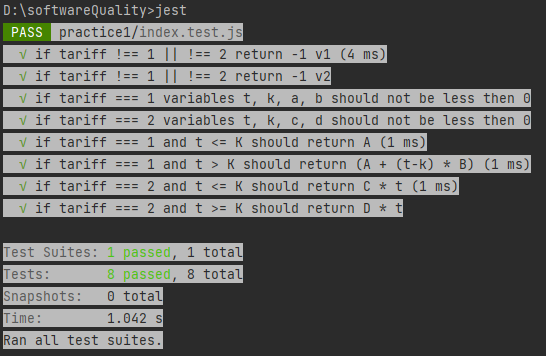


Рисунок 1 – Результат выполнения тестов.