**#8 Вписка (вспомогательное задание для Пиксельхантера)**

**Это дополнительное задание для студентов, которые работают с проектом «Пиксельхантер».**

По ходу игры пользователи видят изображения, которые вписываются в определенные рамки. Проблема заключается в том, что не всегда пропорции изображений в данных совпадают с пропорциями блоков, в которые изображения отрисовываются. В этом задании мы решим проблему таких изображений.

Чтобы решить проблему разных пропорций изображения и контейнера, вам предлагается написать метод, который будет отвечать за подгонку размеров изображения так, чтобы оно могло вписаться в определенные рамки.

Этот метод, должен принимать на вход два объекта со свойствами width и height. Первый объект описывает размеры рамки, в которые должно быть вписано изображение, а второй объект описывает размеры изображения, которые нужно подогнать под рамку. Метод должен возвращать новый объект, который будет содержать измененные размеры изображения.

Ниже представлен код теста, который должен проходить метод. Добавьте код теста в свой проект и сделайте так, чтобы этот тест проходил. Т.о. эта задача решается с помощью TDD:

import assert from 'assert';

// NB! В этом тесте подразумевается, что модуль называется `resize`

// и экспортирует метод `resize`. Если вы назовете метод иначе, учтите

// это в тексте этого теста

import {resize} from './resize';

const createTestForFrame = (frame) => {

const assertRatio = (given, expected) => {

const actual = resize(frame, given);

assert.deepEqual(actual, expected);

};

const createTest = (expected, multiplier) => {

const given = {

width: Math.floor(expected.width \* multiplier),

height: Math.floor(expected.height \* multiplier)

};

it(`shrink ${multiplier}x: ${given.width}x${given.height} => ${expected.width}x${expected.height}`, () => {

assertRatio(given, expected);

});

};

const sequence = (expected) => {

createTest(expected, 8);

createTest(expected, 7);

createTest(expected, 5);

createTest(expected, 4);

createTest(expected, 3);

createTest(expected, 2);

createTest(expected, 1);

};

describe(`Resize into frame: ${frame.width}x${frame.height}`, () => {

describe(`when "width === height"`, () => {

sequence({width: frame.width, height: frame.height});

});

describe(`when "width > height"`, () => {

sequence({width: frame.width, height: Math.floor(frame.height / 2)});

});

describe(`when "width < height"`, () => {

sequence({width: Math.floor(frame.width / 2), height: frame.height});

});

});

};

createTestForFrame({width: 256, height: 256});

createTestForFrame({width: 256, height: 128});

createTestForFrame({width: 468, height: 458});

createTestForFrame({width: 705, height: 455});

createTestForFrame({width: 304, height: 455});

**Почему не подходит кадрирование**

Существует альтернативный способ решить задачу с пропорциями: отрезать часть изображения так, чтобы его пропорции совпали с пропорциями блока, в который нужно вписать изображение. В нашем случае такое решение не подходит, потому что отрезанная часть изображения может быть важной для решения задачи и содержать какие-то признаки, по которым можно опеределить является изображение картиной или фотографией