Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное  
учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» (ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

**Практическая работа №8**

по курсу «Технологии разработки серверной части веб-приложений» модуль «Unit-тестирование»

Выполнили

студенты группы КТбо3-4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. И. Селевцова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г. Ю. Полынников

Принял

ассистент кафедры МОП ЭВМ А. А. Жиглатый

**СОДЕРЖАНИЕ**

[**ЦЕЛЬ РАБОТЫ 3**](#_Toc136015541)

[1. Постановка задачи 5](#_Toc136015542)

[2. Алгоритм решения поставленной задачи 6](#_Toc136015543)

[3. Тестовый набор 6](#_Toc136015544)

[4. Юнит-тестирование 7](#_Toc136015545)

[5. Баг-репорты и исправление дефектов 9](#_Toc136015546)

[**ЗАКЛЮЧЕНИЕ 12**](#_Toc136015547)

[**ЛИСТИНГ ПРОГРАММЫ 13**](#_Toc136015548)

## ЦЕЛЬ РАБОТЫ

Научиться применять стратегии при проектировании тестовых наборов для проведения модульного тестирования. Освоить существующие средства автоматизации проведения юнит-тестирования.

**Задание**

1. Выполнить подбор задачи, решение которой может быть оформлено в виде одной отдельной основной функции, не включающей в себя чтение входных данных и вывод результата. Входные данные не должны исчерпываться одним входным параметром. Выбранную задачу обсудить с преподавателем.

2. Выполнить корректную постановку выбранной задачи, которая должна включать: формулировку условия; описание входных данных с указанием их состава, формата представления и области допустимых значений для каждого элемента входных данных; описание выходных данных с указанием их состава и формата представления; примеры входных и выходных данных; возможные способы обработки входных данных из области недопустимых значений; возможные указания и дополнительные требования к реализации.

3. Выполнить оформление решения поставленной задачи в виде одной отдельной основной функции. Допускается наличие вспомогательных 2 функций, реализующих решение подзадач, возникающих в ходе решения поставленной задачи. Также допускается вызов вспомогательных функций из основной. Однако, ввод входных данных и вывод выходных данных должен быть исключен из основной функции, в том числе и посредством вспомогательных функций. Источником входных данных должен являться файл. Вывод результата также необходимо осуществлять в файл.

4. Разработать тестовый набор для проверки корректности решения поставленной задачи (основной функции), используя стратегию, позволяющую минимизировать количество тестовых случаев при сохранении максимально возможного тестового покрытия. По возможности оценить величину тестового покрытия для разработанного тестового набора.

5. Используя встроенные в среду разработки средства юнит-тестирования и разработанный тестовый набор, выполнить юнит-тестирование основной функции, реализующей решение поставленной задачи. При этом необходимо использовать файлы в качестве источника входных и эталонных выходных данных. Не допускается их явное задание в коде проекта юнит-тестирования.

6. При обнаружении дефектов в ходе юнит-тестирования выполнить отладку основной функции и повторное её тестирование.

1. Постановка задачи

Необходимо отсортировать методом пузырька массив размера 5х5, состоящий из целых чисел , и сформировать из отсортированных элементов массив размером 5х5.

**Входные данные**

Массив размером 5х5, состоящий из целочисленных элементов .

Формат текстового представления: пять строк из пяти целых чисел, разделенных пробелами.

**Выходные данные**

Массив размером 5х5.

Формат текстового представления: пять строк из пяти целых чисел, разделенных пробелами.

**Особенность разработки теста**

Необходимо, чтобы входные данные считывались из текстового файла в функции программы, отвечающей за тестирование исполняемого кода.

**Обработка исключительных ситуаций**

Исключение типа ReferenceError, когда функция вызвана без аргументов. Текст сообщения об ошибке: «Нет данных»

Исключение типа ReferenceError, когда функция вызвана с единственным аргументом, являющимся пустым массивом. Текст сообщения об ошибке: «Введен пустой массив».

Исключение типа TypeError, когда функция вызвана с параметром, не являющаяся массивом. Текст сообщения об ошибке: «Введен не массив».

Исключение типа Error, когда функция вызвана с массивом, где недостаточно символов. Текст сообщения об ошибке: «Введен неполный массив».

Исключение типа Error, когда функция вызвана с массивом с избыточным количеством символов. Текст сообщения об ошибке: «Введены избыточные данные».

Исключение типа TypeError, когда функция вызвана с неправильным форматом ввода (например, символы перечислены через запятую или точку с запятой). Текст сообщения об ошибке: «Неверный формат ввода».

Исключение типа TypeError, когда функция вызвана с массивом, имеющим нецелочисленные элементы. Текст сообщения об ошибке: «Неверный тип элементов массива».

Исключение типа TypeError, когда функция вызвана с массивом, чьи элементы выходят за границы диапазона. Текст сообщения об ошибке: «Значения элементов массива находятся вне допустимого диапазона».

**Примеры входных данных:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 668 -714 909 -620 510  -949 -816 411 342 -156  747 948 -317 2 525  145 260 647 -513 975  257 739 -244 -190 76 | -949 -816 -714 -620 -513  -317 -244 -190 -156 2  76 145 257 260 342  411 510 525 647 668  739 747 909 948 975 |

2. Алгоритм решения поставленной задачи

3. Тестовый набор

Для получение минимального количества тестов, имеющих максимальное тестовое покрытие, выделим следующие тесты:

1. Позитивные тесты: 1 тест

2. Тесты граничных значений: 2 теста.

3. Негативные тесты: 9 тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Входные данные | Ожидаемый вывод | Что проверяет тест |
| 1 | 1 -2 0 0 0  -2 0 1 0 0  0 -1 1 -2 0  -2 0 1 2 0  2 1 0 0 0 | -2 -2 -2 -2 -1  0 0 0 0 0  0 0 0 0 0  0 0 0 1 1  1 1 1 2 2 | Типичный случай |
| 2 | 1000 -1000 1000 -1000 1000  -1000 1000 -1000 1000 -1000  1000 -1000 1000 -1000 1000  -1000 1000 -1000 1000 -1000  1000 -1000 1000 -1000 1000 | -1000 -1000 -1000 -1000 -1000  -1000 -1000 -1000 -1000 -1000  -1000 -1000 1000 1000 1000  1000 1000 1000 1000 1000  1000 1000 1000 1000 1000 | Граничные значения интервала |
| 3 | -1000 1000 -1000 1000 -1000  1000 -1000 1000 -1000 1000  -1000 1000 -1000 1000 -1000  1000 -1000 1000 -1000 1000  -1000 1000 -1000 1000 -1000 | -1000 -1000 -1000 -1000 -1000  -1000 -1000 -1000 -1000 -1000  -1000 -1000 -1000 1000 1000  1000 1000 1000 1000 1000  1000 1000 1000 1000 1000 | Граничные значения интервала |
| 4 | Нет данных | Сообщение об ошибке: «Нет данных» | Исключение |
| 5 | ‘’ | Сообщение об ошибке: «Введен пустой массив» |  |
| 6 | ‘string’ (введена строка) | Сообщения об ошибке: «Введен не массив» |  |
| 7 | 1 1 8 9 0  0 9 7 | Сообщение об ошибке: «Введен неполный массив» |  |
| 8 | 1 2 3 4 5 6  7 8 9 10 11  12 13 14 -15  -16 -17 -18 19 20  21 22 23 24 25 | Сообщение об ошибке: «Введены избыточные данные». |  |
| 9 | -3,-3,-3:-3,-3;  -3,-3,-3,-3,-3;  -3,-3,-3,-3,-3;  -3,-3,-3,-3,-3;  -3,-3,-3:-3,-3; | Сообщение об ошибке: «Неверный формат ввода». |  |
| 10 | 65 0 -8 f 44  3.907 -85 262 ! 9  11 11 56 \* 7  777 ‘one’ -12 90 67  ^ -6.98 + 42 -43 | Сообщение об ошибке: «Неверный тип элементов массива». |  |
| 11 | -1001 -1001 -1001 -1001 -1001  -1001 -1001 -1001 -1001 -1001  -1001 -1001 -1001 -1001 -1001  -1001 -1001 -1001 -1001 -1001  -1001 -1001 -1001 -1001 -1001 | Сообщение об ошибке: «Значения элементов массива находятся вне допустимого диапазона». |  |
| 12 | 1001 1001 1001 1001 1001  1001 1001 1001 1001 1001  1001 1001 1001 1001 1001  1001 1001 1001 1001 1001  1001 1001 1001 1001 1001 | Сообщение об ошибке: «Значения элементов массива находятся вне допустимого диапазона». |  |

4. Юнит-тестирование

Для проведения автоматизированного юнит-тестирования была использована библиотека Jest. Тесты описаны в файле index.test.js в виде одного метода, в котором прописаны множество тест-кейсов. Результаты выполнения тестов можно увидеть на скриншоте ниже:

Рисунок 1 – Результат тестирования

Во время выполнения тестов были найдены некоторые ошибки, которые были расписаны и исправлены в следующем разделе.

5. Баг-репорты и исправление дефектов

Дефект №1

|  |  |
| --- | --- |
| Название дефекта | Невыполнение типичного тестового случая |
| Приоритет дефекта | Высокий |
| Серьезность дефекта | Критический |
| Версия системы | ОС Windows 10, NodeJS v18.14.0 |
| Инструкция по воспроизведению дефекта | Входные данные: «8888» |
| Фактический результат | Исключение «Недостаточное количество входных данных» |
| Ожидаемый результат | "Невозможно включить лампочки" |
| Вид, тип, область тестирования | Функциональное, модульное тестирование |
| Вложения |  |

Дефект №2

|  |  |
| --- | --- |
| Название дефекта | Вызывается необработанное исключение при отсутствии необходимых параметров |
| Приоритет дефекта | Высокий |
| Серьезность дефекта | Критический |
| Версия системы | ОС Windows 10, NodeJS v18.14.0 |
| Инструкция по воспроизведению дефекта | Входные данных нет |
| Фактический результат | Ошибка “Cannot read properties of undefined (reading 'length')” |
| Ожидаемый результат | Исключение "Отсутствуют входные данные" |
| Вид, тип, область тестирования | Функциональное, модульное тестирование |
| Вложения |  |

Дефект №3

|  |  |
| --- | --- |
| Название дефекта | Ошибочный вызов исключений при корректных типичных данных |
| Приоритет дефекта | Высокий |
| Серьезность дефекта | Критический |
| Версия системы | ОС Windows 10, NodeJS v18.14.0 |
| Инструкция по воспроизведению дефекта | Входные данные:  "8888" |
| Фактический результат | Исключение "Неправильный формат " |
| Ожидаемый результат | "Невозможно включить лампочки" |
| Вид, тип, область тестирования | Функциональное, модульное тестирование |
| Вложения |  |

6. Исправление дефектов

Для того, чтобы функция проходила все тесты, необходимо внести в неё следующие изменения:

function laba2(a){

    if (a===undefined) throw new Error('Отсутствуют входные данные');

    t = a.length;

    if (t < 4) throw new Error('Недостаточное количество входных данных');

    if (t > 4) throw new Error('Избыточное количество входных данных');

    var s = a.split('');

    if (isNaN(s[0]) || isNaN(s[1]) || isNaN(s[2]) || isNaN(s[3])) throw new Error('неправильный формат');

    colors = [... new Set(s)];

    switch (colors.length) {

      case 4:

        return 4;

      case 3:

        return 4;

      case 2:

        var counter = 0;

        for(var j = 0 ; j < s.length ; ++j){

          if(colors[0] == s[j]){

            counter++;

          }

        }

        return counter == 2 ? 4 : 6;

      case 1:

        return -1;

      default:

        return ('');

    }

  }

Листинг 2 — Исправленный алгоритм решения

После этого юнит-тестирование завершается без ошибок.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 — Пройденное тестирование

## ЗАКЛЮЧЕНИЕ

При выполнении лабораторной работы были получены навыки в базовом юнит-тестировании, подборе тест кейсов и анализе покрытия тестов. Была разработана корректно работающая программа, решающая поставленную задачу, а также были использованы средства автоматизации языка JavaScript (Jest).

## ЛИСТИНГ ПРОГРАММЫ

Файл **index.test.js**

const laba2 = require('./index');

test('тесты', () => {

  expect(laba2('8888')).toBe("Невозможно включить лампочки");

  expect(laba2('1110')).toBe("Включить лампочки можно 6 операциями");

  expect(laba2('3443')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('2106')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('0123')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('9123')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('8071')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('1921')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('3303')).toBe("Включить лампочки можно 6 операциями");

  expect(laba2('9999')).toBe("Невозможно включить лампочки");

  expect(laba2('1820')).toBe("Включить лампочки можно 4 операциями");

  expect(laba2('2229')).toBe("Включить лампочки можно 6 операциями");

});

test ('For 100', () => {

  expect(() =>  laba2('100')).toThrow('Недостаточное количество входных данных');

});

test ('For 28', () => {

  expect(() =>  laba2('28')).toThrow('Недостаточное количество входных данных');

});

test ('For 9', () => {

  expect(() =>  laba2('9')).toThrow('Недостаточное количество входных данных');

});

test ('For ', () => {

  expect(() =>  laba2()).toThrow('Отсутствуют входные данные');

});

test ('For 42642', () => {

  expect(() =>  laba2('42642')).toThrow('Избыточное количество входных данных');

});

test ('For a248', () => {

  expect(() =>  laba2('a248')).toThrow('Неправильный формат');

});

test ('For 0?00', () => {

  expect(() =>  laba2('0?00')).toThrow('Неправильный формат');

});

test ('For 27,5', () => {

  expect(() =>  laba2('27,5')).toThrow('Неправильный формат');

});

test ('For 404H', () => {

  expect(() =>  laba2('404H')).toThrow('Неправильный формат');

});