Министерство образования и науки Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Лабораторная работа №2

«Свойства отношения»

Выполнил студент

группы ИВТАПбд-11

Мухаметзянов Т. А.

Ульяновск, 2022

**Оглавление**

[**Постановка задачи 2**](#_Toc103805213)

[**Реализация 3**](#_Toc103805214)

[**Блок-схема 4**](#_Toc103805215)

[**Интерфейс работы 6**](#_Toc103805216)

[**Листинг кода 7**](#_Toc103805217)

[**Выводы 13**](#_Toc103805218)

# Постановка задачи

Разработать программу на JS, определяющую свойства отношения. Формат ввода определяется заданием. Необходимо выполнить проверку корректности ввода отношения. Вариант: матрица N x N.

# Реализация

Структура

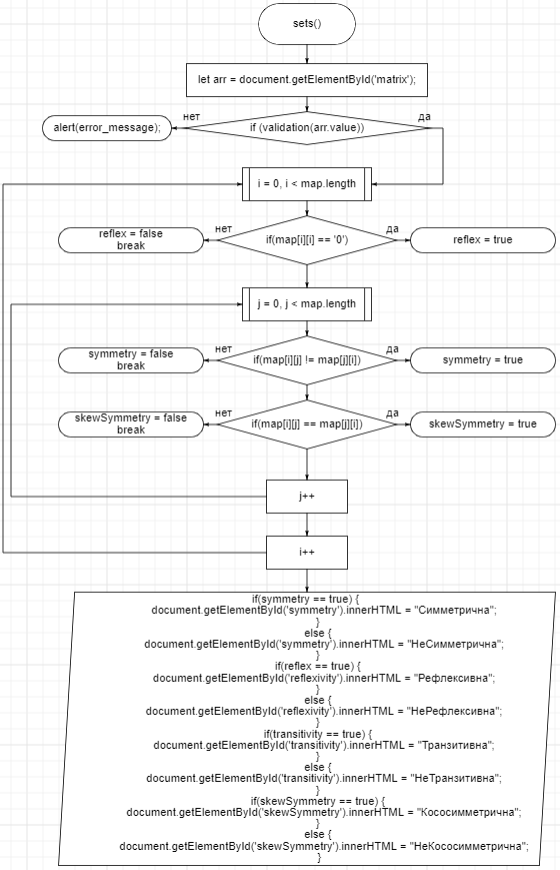
1. HTML страница
2. JS файл с основным скриптом для корректной работы страницы
3. CSS файл со всеми стилями

Лабораторная работа реализована в 3 файла (html, css, js). На сайте пользователь вводит матрицу размером N x N:

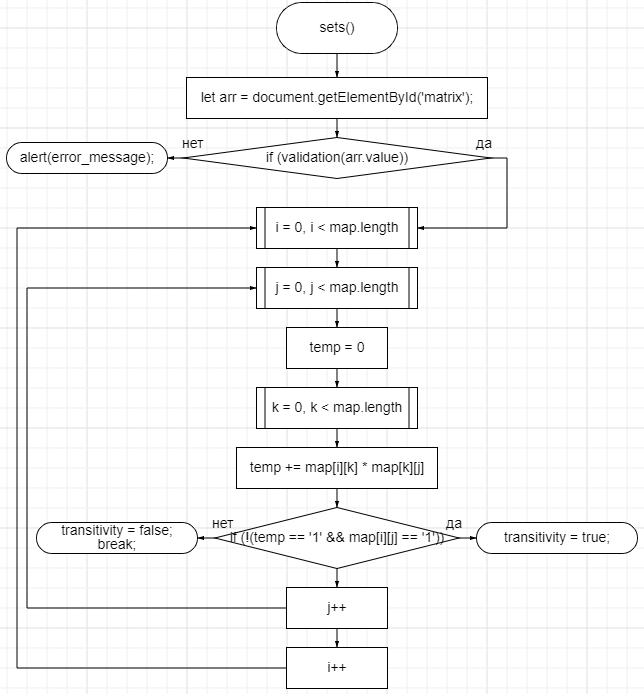
1. Если ввод не проходит валидацию, тогда пользователю выводит ошибку, которую необходим исправить
2. Если ввод корректен, тогда пользователь получает строчки с результатами свойств отношения.

Сайт состоит из номера л\р, поля ввода матрицы, кнопки подтверждения (связанная с основной функцией js) и абзацев для вывода результатов.

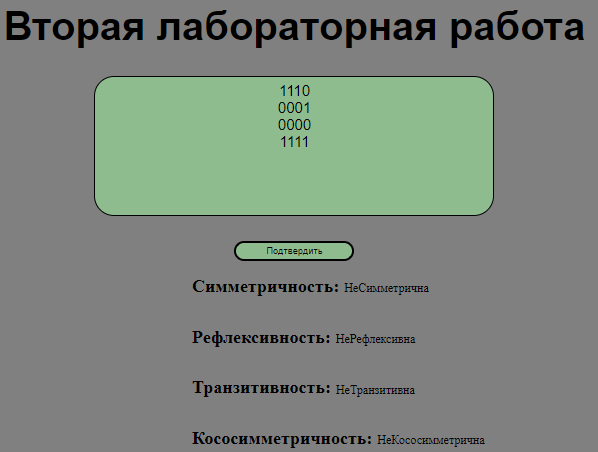
# Блок-схема



Транзитивность



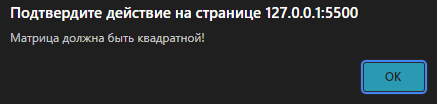
# Интерфейс работы



**Рис-1 Интерфейс**

Состоит из заголовка, поля ввода и кнопки подтверждения ввода, также абзацы для вывода результатов.

Пример ошибки



**Рис-2 Ошибка валидации**

# Листинг кода

HTML

<!DOCTYPE *html*>

<html *lang*="ru">

<head>

    <meta *charset*="UTF-8">

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <link *rel*="stylesheet" *href*="../Styles/dm2Styles.css">

    <script *src*="../JS/dm2Scrypts.js"></script>

    <title>Вторая лабораторная работа</title>

</head>

<body>

    <div *class*="wrap">

        <div *class*="main">

            <h1 *class*="name">Вторая лабораторная работа</h1>

            <form *method*="" *class*="forms" *onsubmit*="return false;">

                <textarea *class*="textar" *id*="matrix" *placeholder*="Введите квадратную матрицу(без пробелов между эл." *form*="form" *name*="input"></textarea>

                <input *class*="sub" *type*="button" *value*="Подтвердить" *onclick*="sets()">

                <div *class*="res">

                    <h2>Симметричность:</h2>

                    <p *id*="symmetry"></p>

                </div>

                <div *class*="res">

                    <h2>Рефлексивность:</h2>

                    <p *id*="reflexivity"></p>

                </div>

                <div *class*="res">

                    <h2>Транзитивность:</h2>

                    <p *id*="transitivity"></p>

                </div>

                <div *class*="res">

                    <h2>Кососимметричность:</h2>

                    <p *id*="skewSymmetry"></p>

                </div>

            </form>

        </div>

    </div>

</body>

</html>

CSS

\*{

    padding: 0;

}

\*,\**:before*,\**:after*{

*-moz-box-sizing*: border-box;

*-webkit-box-sizing*: border-box;

    box-sizing: border-box;

}

*:focus*,*:active*{outline: none;}

a*:focus*,a*:active*{outline: none;}

nav,footer,header,aside{display: block;}

body, html {

    background-color:grey;

    height: 100%;

    margin: 0;

    padding: 0;

}

*.wrap* {

    display: flex;

    min-height: 100%;

    justify-content: center;

    flex-direction: column;

    align-items: center;

    height: 100%;

}

*.main* {

    display: flex;

    justify-content: center;

    align-items: center;

    padding: 10px 0;

    flex-direction: column;

}

*.forms* {

    display: flex;

    flex-direction: column;

    align-items: center;

    height: 100px;

    justify-content: space-between;

}

*.name* {

    font-size: 40px;

    font-family: Arial, Helvetica, sans-serif;

}

*.sub* {

    background-color:darkseagreen;

    height: 20px;

    width: 120px;

    border-radius: 20px;

    border-color: black;

    color: black;

    cursor: pointer;

    margin-top: 10px;

    padding: 3px 0;

}

*.textar* {

    font-size: 15px;

    font-family: Arial;

    background-color:darkseagreen;

    border-radius: 20px;

    text-align: center;

    border-color:rgb(0, 0, 0);

    padding: 5px;

    resize: none;

    overflow: hidden;

    box-sizing: border-box;

    height: 100px;

    min-height: 140px;

    width: 400px;

    margin-bottom: 15px;

  }

*.textar::-webkit-input-placeholder* {

    color:rgb(0, 0, 0);

 }

*.textar:-moz-placeholder* {

    color:rgb(0, 0, 0);

 }

*.textar::-moz-placeholder* {

    color:rgb(0, 0, 0);

 }

*.textar:-ms-input-placeholder* {

    color:rgb(0, 0, 0);

 }

*.results* {

     font-size: 30px;

     font-family: Arial, Helvetica, sans-serif;

 }

*.res* {

     display: flex;

     width: 100px;

     justify-content: space-between;

     margin-right: 105px;

 }

*.res* p {

     padding-top: 8px;

     padding-left: 5px;

 }

JS

var error\_message = "";

// Функция валидации (проверка логического ввода матриц)

function validation(*mas*) {

    let boolu = true;

    if(*mas*.length > 0) {

        let bool = *mas*.split('\n');

        // проверка на валидацию

        for(let i = 0; i < bool.length; i++) {

            for(let j = 0; j < bool.length; j++) {

                if(bool[i][j] != '1' && bool[i][j] != '0') {

                    error\_message = "В матрице могут быть только 0 и 1!";

                    boolu = false;

                    break;

                }

                if(bool.length != bool[i].length) {

                    error\_message = "Матрица должна быть квадратной!";

                    boolu = false;

                    break;

                }

            }

        }

    }

   else  {

       error\_message = "Поле не должно быть пустым!"

       boolu = false;

   }

    return boolu;

}

function sets() {

    let arr = document.getElementById('matrix');

    if (validation(arr.value)) { //true

        let map = arr.value.split('\n');

        let temp; // переменная, которая хранит значение перемножения матрицы саму на себя

        let reflex = true,

            symmetry = true,

            skewSymmetry = true,

            transitivity = true; // переменные логического типа

        for(let i = 0; i < map.length; i++) {

            for(let j = 0; j < map.length; j++) {

                // симметричность

                if(map[i][j] != map[j][i]) { // элементы симметричны отн. главной диагонали

                    symmetry = false;

                    break;

                }

                // кососимметричность

                if(map[i][j] == map[j][i]) { // элементы несимметричны отн. главной диагонали

                    skewSymmetry = false;

                    break;

                }

            }

        }

        for(let i = 0; i < map.length; i++) {

            //рефлексивность

            if(map[i][i] == '0') { // элементы на главной оси равны 1, тут обратное условие

                reflex = false;

                break;

            }

        }

        // транзитивность

        for(let i = 0; i < map.length; i++) {

            for(let j = 0; j < map.length; j++) {

                temp = 0;

                for(let k = 0; k < map.length; k++) {

                    temp += map[i][k] \* map[k][j]; //  перемноение матрицы самой на себя

                }

            }

        }

        if (temp > 1) {

            temp = 1; // если больше 1, то возвращаем бинарный вид

        }

        for(let i = 0; i < map.length; i++) {

            for(let j = 0; j < map.length; j++) {

                for(let k = 0; k < map.length; k++) {

                    if (!(temp == '1' && map[i][j] == '1')) { // значение произведения  и элементы должны быть равны 1, тут обратное

                        transitivity = false;

                        break;

                    }

                }

            }

        }

        // вывод данных в html файл

        if(symmetry == true) {

            document.getElementById('symmetry').innerHTML = "Симметрична";

        }

        else {

            document.getElementById('symmetry').innerHTML = "НеСимметрична";

        }

        if(reflex == true) {

            document.getElementById('reflexivity').innerHTML = "Рефлексивна";

        }

        else {

            document.getElementById('reflexivity').innerHTML = "НеРефлексивна";

        }

        if(transitivity == true) {

            document.getElementById('transitivity').innerHTML = "Транзитивна";

        }

        else {

            document.getElementById('transitivity').innerHTML = "НеТранзитивна";

        }

        if(skewSymmetry == true) {

            document.getElementById('skewSymmetry').innerHTML = "Кососимметрична";

        }

        else {

            document.getElementById('skewSymmetry').innerHTML = "НеКососимметрична";

        }

    }

    // вывод ошибки при validation == false

    else {

        alert(error\_message);

    }

}

# Выводы

С помощью данной лабораторной работы были изучены свойства отношений, а также работа с ними в JS, применение синтаксиса языка javascript. Также приобретены навыки обработки корректного ввода и работы с ошибками.

**Литература**

1. https://habr.com/ru/post/515014/
2. http://mathhelpplanet.com/static.php?p=spetsialnyye-svoystva-binarnykh-otnosheniy