Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №2

«Свойства отношений»

Выполнил студент

группы ИВТАСбд-11

Абросимов К. С.

Ульяновск, 2022

**Цель работы**

Необходимо написать программу, которая будет определять свойства отношения, введенного бинарной матрицей (состоящей из нулей и единиц), причём матрица должна быть размера n\*n. Определяться должны следующие отношения: рефлексивность, симметричность, кососимметричность (антисимметричность) и транзитивность. Пользователь должен иметь возможность вводить данные, а также должна присутствовать проверка на корректность введённых данных.

**Описание метода выполнения**

Бинарным отношением между элементами множества A и B называется любое подмножество R множества AxB, то есть R ℂ A x B. Отношение рефлексивно, если для любых х из множества А существуют отношение xRx. Отношение симметрично, если для любых х из множества А существуют отношение xRy и yRx. Отношение симметрично, если для любых х и y из множества А существует xRy, из которого следует yRx. Отношение кососимметрично, если для любых х и у из множества А существует xRy и yRx, из которых следует, что x=y. Отношение транзитивно, если для любой тройки элементов x, y, z из А и таких бинарных отношений, как xRy и yRx следует xRz.

Если же с матрицами, то

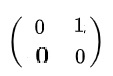
Рефлексивность:



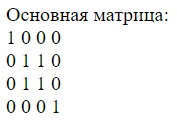
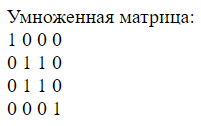
Симметричность:



Антисимметричность – идёт только в одну сторону относительно главной диагонали:



Транзитивность – матрица, умноженная на саму себя, всё равно равна изначальной.

Код проводимых операций будет написан на PHP. PHP - это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере. PHP взаимодействует с html.

Воспользуемся уже созданным файлом с прошлой работы, чтобы установить ссылку на другую лабораторную работу.

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Лабораторные работы</title> - основной заголовок  
 <script type="text/javascript" src="/scripts/scripts.js"></script>  
</head>  
<body>  
 <h1> Мои лабораторные работы </h1>   
 <a href ="mldmLab1.php">Первая лабораторная работа по МЛиДМ</a><br>  
 <a href ="mldmLab2.php">Вторая лабораторная работа по МЛиДМ</a> - ссылки к другим лабораторным работам  
</body>  
</html>

Для того, чтобы пользователь мог вводить данные необходимо написать небольшую html-структуру, состоящую поля для ввода (input), полей для вывода и кнопки, по которой произойдут операции над множествами. Также для полного вывода операций нужно создать значение для вывода.

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Лабораторная работа 2</title>  
 <script type="text/javascript" src="/scripts/scriptslab2.js"></script> - ссылка на другой файл с функциями  
</head>  
<body>  
<h1> Лабораторная работа №2 </h1>  
<form>  
 <table>  
 <tr>  
 <td> Введите бинарную матрицу n\*n </td>  
 <td> <textarea id="mass" value="" rows = "10" cols = "20" size=""></textarea></td> - поле для ввода  
 </tr>  
 <tr>  
 <td colspan="2"> <input type="button" value="Сделать расчёт" onclick="rasschetmatrix();"/></td> - кнопка для выполнения расчётов  
 </tr>  
 </table>  
</form>  
<div id ="Matrix"></div>  
<div id ="outResult"></div>  
</body>  
</html>

Сначала функция “rasschetmatrix” создаёт переменные для считывания информации, введённое пользователем. Но они не являются массивом, а значением. После преобразования значения в двумерный массив, проверяем массив на правильность ввода (по условиям задачи). После этого функция с умножением матрицы на саму себя и последующее определение свойств матрицы.

function rasschetmatrix()  
{  
 var a = ***document***.getElementById('mass'); - получение значения  
 var massmat = [];  
 massmat = needmass( a.value ); - функция преобразования значения в матрицу  
 if(validate(massmat) == false) – проверка ввода  
 {  
 alert(***error\_text***);  
 }  
 else {  
 let doublemat = multiplyMatrix(massmat); - новая матрица, умноженная на саму себя  
 let matrixes = ""; - будущий вывод матриц  
 for (let i = 0; i < massmat.length; i++) {  
 for (let j = 0; j < massmat[i].length; j++) {  
 matrixes += massmat[i][j] + " ";  
 }  
 matrixes += "\n";  
 }  
 let doublematrixes = "";  
 for (let i = 0; i < doublemat.length; i++) {  
 for (let j = 0; j < doublemat[i].length; j++) {  
 doublematrixes += doublemat[i][j] + " ";  
 }  
 doublematrixes += "\n";  
 }  
 ***document***.getElementById('Matrix').innerText = "Основная матрица:\n" + matrixes +  
 "\n" + "Умноженная матрица:\n" + doublematrixes + "\n";  
 let antiSym = true;  
 let refl = true;  
 let sym = true;  
 let tranz = true;  
 for(let i = 0; i < massmat.length; i++) { - основная работа над свойствами матрицы именно здесь:  
 for(let j = 0; j < massmat.length; j++) {  
 if (!(!((massmat[i][j] == 1) && (i != j)) || massmat[j][i] == 0)) {  
 antiSym = false;  
 }  
 if (i == j) {  
 if (massmat[i][j] == 0) {  
 refl = false;  
 }  
 }else {  
 if(massmat[i][j] != massmat[j][i]) {  
 sym = false;  
 }  
 }  
 if ((massmat[i][j] == 0 && doublemat[i][j] == 1) || (massmat[i][j] == 1 && doublemat[i][j] == 0)) {  
 tranz = false;  
 }  
 }  
 }  
 let full\_result = ""; - будущий вывод свойств матрицы  
 if (refl == true) {  
 full\_result = "Данная матрица рефлексивна\n";  
 }  
 else {  
 full\_result = "Данная матрица не рефлексивна\n";  
 }  
 if (sym == true) {  
 full\_result += "Данная матрица симметрична\n";  
 }  
 else {  
 full\_result += "Данная матрица не симметрична\n";  
 }  
 if (antiSym == true) {  
 full\_result += "Данная матрица антисимметрична\n";  
 }  
 else {  
 full\_result += "Данная матрица не антисимметрична\n";  
 }  
 if (tranz == true) {  
 full\_result += "Данная матрица транзитивна\n";  
 }  
 else {  
 full\_result += "Данная матрица не транзитивна\n";  
 }  
 ***document***.getElementById('outResult').innerText = "Свойства матрицы:\n" + full\_result;  
  
 }  
}

Функция needmass преобразует введённое значение в массив, при этом удаляя лишние пробелы и строки.

function needmass( str )  
{  
 var massiv = [];  
 massiv = str.split("\n").map(function (x) { return x.split(" "); });  
  
 for (let i = 0; i < massiv.length; i++) {  
 if (massiv[i] == "" && massiv[i].length == 1) { - удаление лишних строк  
 massiv.splice(i, 1);  
 i--;  
 }  
 }  
 for (let i = 0; i < massiv.length; i++) {  
 for (let j = 0; j <= massiv[i].length; j++) { - удаление лишних пробелов  
 if (massiv[i][j] == "") {  
 massiv[i].splice(j, 1);  
 }  
 }  
 }  
 return massiv;  
}

Функция validate проверяет полученный массив на 3 условия:

* Что он не пустой
* Что его формат n\*n
* Что его значения только 0 и 1

function validate(str)

{

let mass = str;

let flag = true;

if(str.length>0) { - первое условие

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

if (mass.length != mass[i].length) { - второе условие

error\_text = "Матрица неверного формата. Формат n\*n";

flag = false;

break;

}

for (let j = 0; j < mass[i].length; j++) {

if (mass[i][j] != 1 && mass[i][j] != 0) { - третье условие

error\_text = "Матрица должна содержать только элементы 1 или 0";

flag = false;

break;

}

}

}

}

else {

error\_text = "Массив не может быть пустым";

flag = false;

}

return flag;

}

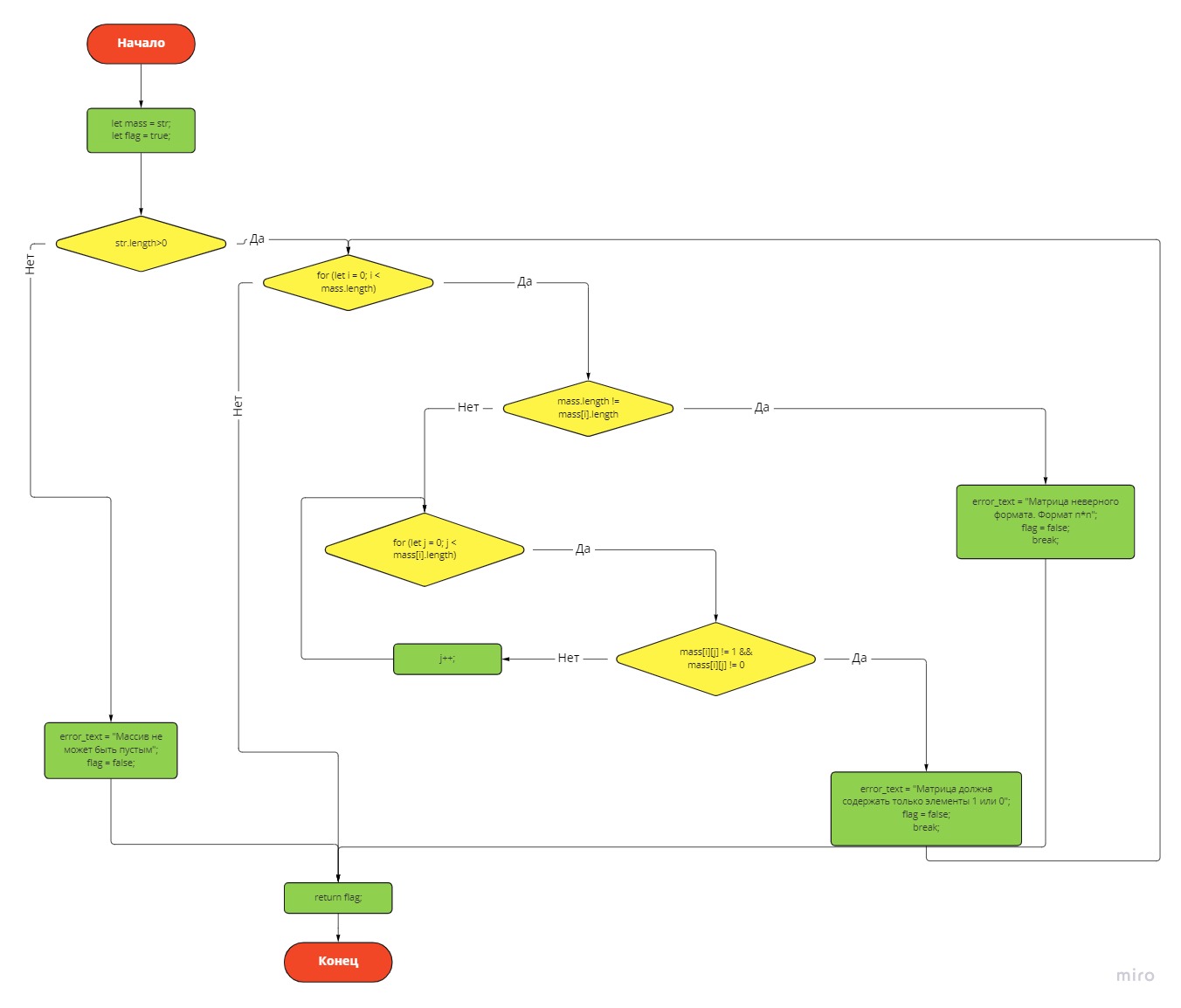


Рис. 1 - функция validate

Функция multiplyMatrix просто умножает матрицу саму на себя.

function multiplyMatrix(n)  
{  
 let result = [];  
 for (let i = 0; i < n.length; i++) {  
 result[i] = [];  
 }  
 for (let i = 0; i < n.length; i++) {  
 for (let j = 0; j < n.length; j++) {  
 let elem = 0;  
 for (let k = 0; k < n.length; k++) {  
 elem += n[j][k] \* n[k][i];  
 }  
 if (elem >= 1) {  
 result[i][j] = 1;  
 } else {  
 result[i][j] = 0;  
 }  
 }  
 }  
 return result;  
}

**Выводы**

В результате проведения работы были изучены основы работы с двумерными массивами в JavaScript, изучено понятие отношения множеств и его свойства (рефлексивность, симметричность, антисимметричность, транзитивность).

# **Список литературы**

(б.д.).

*Metanit*. (б.д.). Получено из JavaScript | Введение в массивы: https://metanit.com/web/javascript/2.5.php

*ppt-online*. (б.д.). Получено из Теория множеств: https://ppt-online.org/5744