Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Математическая логика и дискретная математика»

Лабораторная работа №1

«Операции с множествами»

Выполнил студент

группы ИВТАПбд-12

Матюнин И.С.

Ульяновск, 2022

Оглавление

Цель работы 2

Описание метода выполнения 2

Код JavaScript: 3

Вывод 13

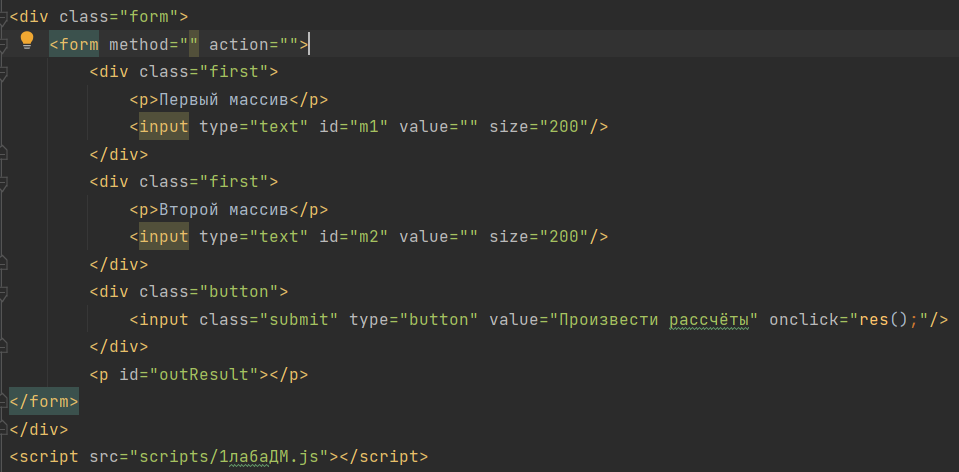
Список использованной литературы 13

# **Цель работы**

Необходимо написать HTML страницу, которая будет проводить операции над множествами: объединение, пересечение. дополнения А\B и B\A и симметрическую разность. Также должна присутствовать валидация (проверка ввода) данных (Первый символ элемента – цифра, второй – четная цифра, третий – нечётная буква и четвертый – буква) и вывод ошибки, если хотя бы одно множество пустое. Данные вводятся пользователем.

# **Описание метода выполнения**

Первым делом, мы создаем HTML документ, в который добавляются поля для ввода данных (форму) содержащее в себе два текстовых поля для ввода данных, и кнопку.

Рис. Основной код HTML-страницы.

Сами операции будут выполняться в функциях, написанных на языке JavaScript и выполняющиеся при нажатии кнопки, которая обращается к главной функции скрипта res().

# **Код JavaScript:**

1. Обявл глобальные переменные, которые будут использоваться во всех функциях.
2. Валидация данных: функция проверки корректность ввода данных. Первым делом массив проверяется на заполненность. Если массив пустой, то возвращается сообщение об ошибке. В случаее корректности ввода, введённая строка разделяется на отдельные элементы через пробел и удаляют повторяющиеся элементы. С помощью цикла и условий проверяется правильность ввода **cijb**, если есть ошибка, то в переменную созданную для вывода ошибки помещается текст самой ошибки с выведенным элементом и уточнением ошибки, чтобы пользователь мог её исправить.

var m1, m2, err;

function count(arr, el) {

let k = 0;

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

if( arr[i] == el)

k++;

return k;

}

//проверка ввода

function check(str){

let arr = false;

if(str.length>0){

arr = str.split(" ");

//Убрать повторяющиеся элементы

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

if (count(arr, arr[i]) > 1){

arr.splice(i, 1);

i--;

}

}

//Проверка на ввод

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

//первый символ

if (arr[i][0] < 1 || arr[i][0] > 9) {

err = "Ошибка при вводе массива!\n" + str + " В элементе " + arr[i];

err += "\nДолжна быть введена цифра!";

err += "\nИсправьте 1 элемент (" + arr[i][0] + ")";

arr = false;

break;

}

//второй символ

if (arr[i][1] % 2 != 0) {

err = "Ошибка при вводе массива!\n" + str + " В элементе " + arr[i];

err += "\nДолжна быть введена чётная цифра!";

err += "\nИсправьте 2 элемент (" + arr[i][1] + ")";

arr = false;

break;

}

//третий символ

if (arr[i][2] % 2 == 0) {

err = "Ошибка при вводе массива!\n" + str + " В элементе " + arr[i];

err += "\nДолжна быть введена нечётная цифра!";

err += "\nИсправьте 3 элемент (" + arr[i][2] + ")";

arr = false;

break;

}

//четвёртый символ

if (arr[i][3] < "a" || arr[i][3] > "z") {

err = "Ошибка при вводе массива!\n" + str + " В элементе " + arr[i];

err += "\nДолжна быть введена буква!";

err += "\nИсправьте 4 элемент (" + arr[i][3] + ")";

arr = false;

break;

}

}

}

else{

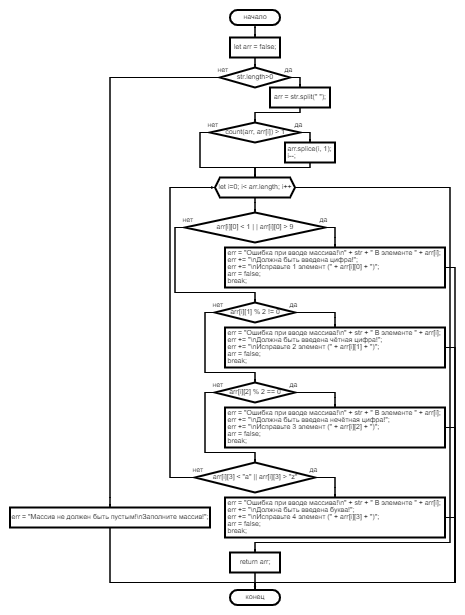
err = "Массив не должен быть пустым!\nЗаполните массив!";

}

return arr;

}

Блок схема:

1. Объединение: функция, объединяющая 2 введенных множества вместе.

//Объединение

function unification(m1, m2){

let result = "";

for(let i=0; i<m1.length; i++){

result += m1[i] +" ";

}

for(let i=0; i<m2.length; i++){

if(m1.indexOf(m2[i]) == -1){

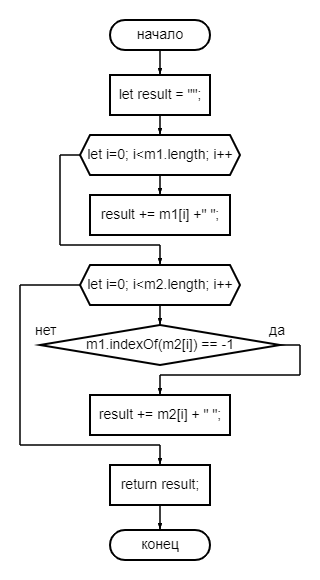
result += m2[i] + " ";

}

}

return result;

}

Блок схема:

1. Пересечение: функция, которая выводит те элементы множества, которые совпали в 2 множествах.

//Пересечение

function intersection(m1, m2){

let result = "";

for(let i=0; i<m2.length; i++)

if(m1.indexOf(m2[i]) != -1){

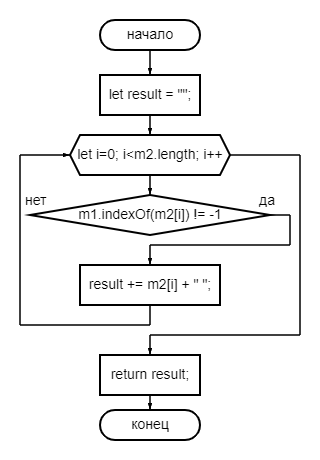
result += m2[i] + " ";

}

}

return result;

}

Блок-схема:

1. Дополнение А\B и B\A: функция, которая выводит те элементы 2 множества, которого нет в 1.

//дополнение массива

function addition(m1,m2){

let result= "";

for(let i=0; i<m1.length; i++){

if(m2.indexOf(m1[i]) == -1){

result += m1[i] + " ";

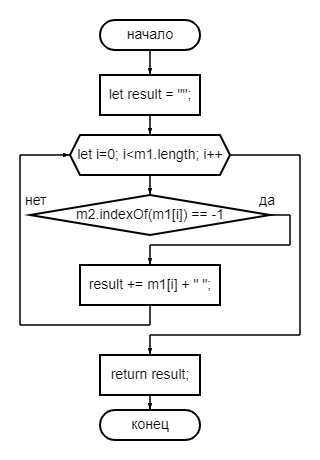
}

}

return result;

}

Блок-схема:



1. Симметрическая разность: функция, выводящая те элементы, которые не пересекаются. Так как симметрическая разность, по своей сути, это соединение дополнений А\B и B\A, функция обратится к функции дополнения с разными входными данными (A\B и B\A).

//симметрическая разность

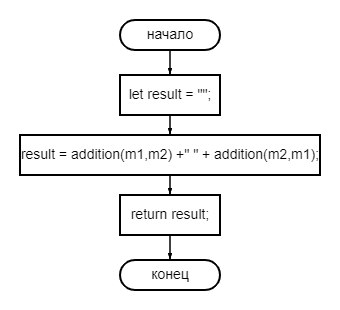
function SymmetricDifference(m1,m2){

let result= "";

result = addition(m1,m2) +" " + addition(m2,m1);

return result;

}

Блок-схема:

1. Конечный результат: функция. которая выводит результат на странице.

//Основая функция

function res() {

let resfull = "";

var errm1 = document.getElementById('m1');

var errm2 = document.getElementById('m2');

if ((m1 = check(errm1.value)) == false){

alert(err);

}

if ((m2 = check(errm2.value)) == false){

alert(err);

}

else{

resfull = "Объединение: " + unification(m1, m2) + "\n";

resfull += "Пересечение: " + intersection(m1, m2) + "\n";

resfull += "Дополнение 1 массива: " + addition(m1, m2) +"\n";

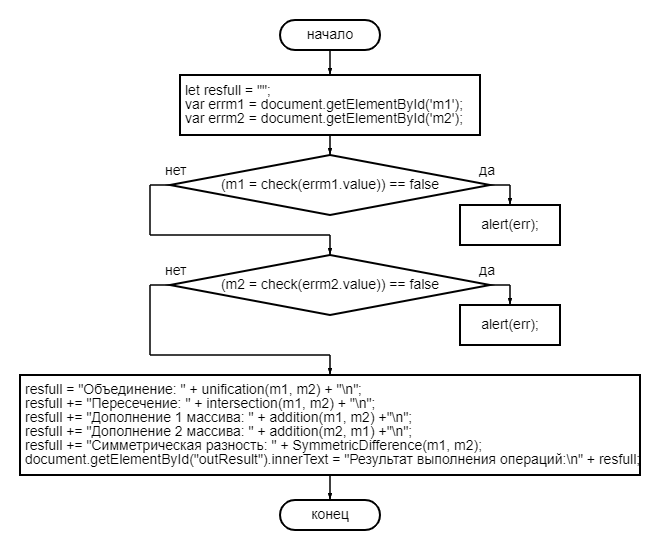
resfull += "Дополнение 2 массива: " + addition(m2, m1) +"\n";

resfull += "Симметрическая разность: " + SymmetricDifference(m1, m2);

document.getElementById("outResult").innerText = "Результат выполнения операций:\n" + resfull;

}

}

Блок-схема:

# **Вывод**

В ходе данной лабораторной работы была разработана программа по реализации основных операций над числовыми множествами средствами языка JavaScript: объединение, пересечение, дополнение А\B и B\A, симметрическая разность.В ходе лабораторной работы были изучены свойства числовых множеств и основы языка программирования JavaScript.

# **Список использованной литературы**

1. Яблонский С.В. Введение в дискретную математику. – М.: Высш. шк., 2003. – 384 с.
2. Даккетт, Джон. HTML и CSS. Разработка и дизайн веб-сайтов/ Джон Дакетт; [пер. с англ. М.А.Райтмана]. - Москва: Эксмо, 2019. - 480 с. - (Мировой компьютерный бестселлер).
3. Даккетт, Джон. JavaScript и jQuery. Интерактивная веб-разработка / Джон Дакетт; [пер. с англ. М.А.Райтмана]. - Москва: Эксмо, 2019. - 640 с.: ил. - (Мировой компьютерный бестселлер).