Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №3

«Функция»

Выполнил студент

группы ИВТАСбд-11

Зюзин Г.А

Проверил Игонин А.Г.

Ульяновск, 2022 г.

# Оглавление.

[Оглавление. 2](#_Toc1)

[Цель работы. 3](#_Toc2)

[Метод выполнения.](#_Toc3)

[Листинг кода. 7](#_Toc4)

[Выводы. 14](#_Toc5)

[Список литературы.](#_Toc6)

# Цель работы.

Написать программу на JavaScript, которая опеределяет: является ли отношение функцией. Вводить необходимо отношение и элементы обоих множеств. Формат ввода: будет текстовое поле, где первый столбец – элементы первого множества, а второй – элементы второго множества. Необходимо выполнить проверку корректности ввода отношения. Программа определяет являются ли отношения между элементами разных множеств функция как первое ко второму, так и второго к первому.

# Метод выполнения.

Данная программа определяет наличие функции между двумя множествами.

Функция – это частный случай бинарных отношений, на которые наложены дополнительные ограничения.

Функцией из множества A в множество В называется бинарное отношение, при котором каждый элемент множества A связан с единственным элементом множества В (Рис. 1).

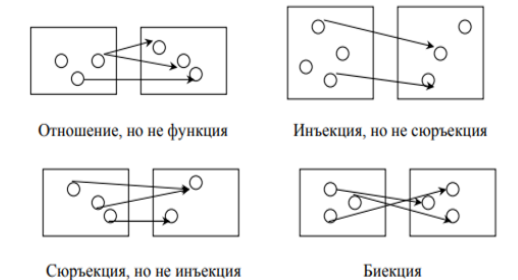
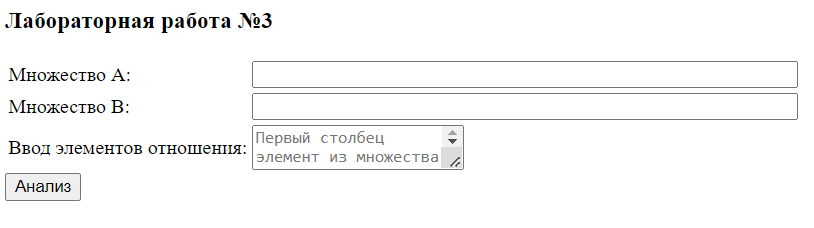


Рис. 1

Рис. 2



В программе используется две строки, где вводятся множества (строка 1 – множества А, 2 – множество B) (Рис. 2 – Вид программы). В эти строки вводятся элементы двух множеств и это те элементы, которые могут использоваться. Если при составлении отношений ввести элемент, которого нет в множестве, то программа выдаст ошибку.

В текстовое поле вводится отношения в два столбца, где первый столбец – элементы множества А, а второй – элементы множества B. Строится матрица, где проставляются 1, которые обозначают, что отношения существуют (Рис. 3 – Пример работы программы).

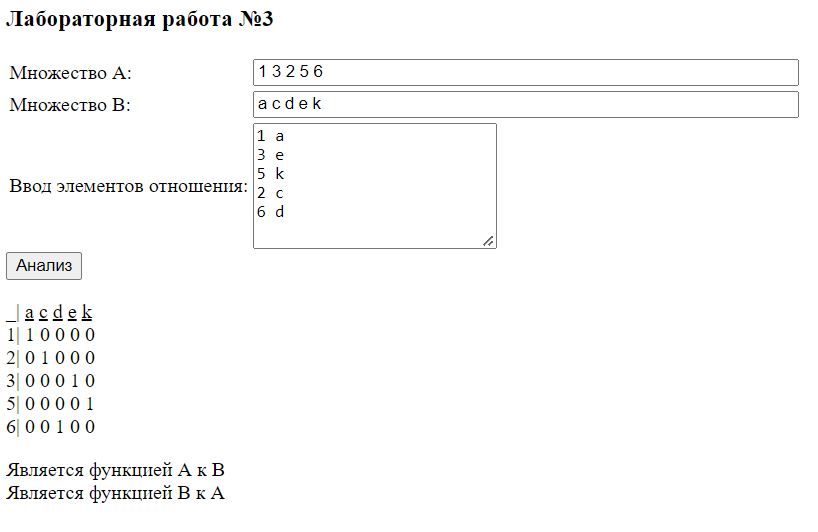
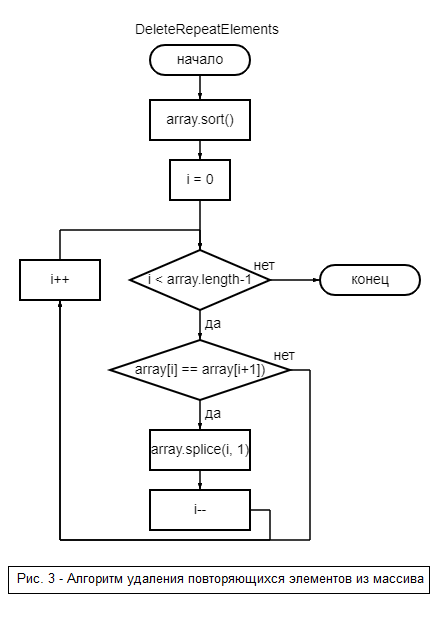


Рис. 3

Благодаря матрице и определяется функция есть ли она или нет.

Программа определяет функцию как из множества A в множество В, так из множества B в множество А. Реализовано это с помощью булевых переменных (Листинг 1).



Листинг 1

let flagFunctionRow = true;

let flagFunctionColumn = true;

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

let countRow = 0;

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

if (matrix[i][j] === 1) {

countRow += 1;

}

if (countRow > 1) {

flagFunctionRow = false;

break;

}

}

if (!flagFunctionRow)

break;

if (countRow === 0)

flagFunctionRow = false;

}

for (let i = 0; i < matrix[0].length; i++) { //Аналогично, только не i j, а j i }

let messageFunctionRow = " ";

let messageFunctionColumn = " ";

if(flagFunctionRow){

messageFunctionRow = "Является функцией А к В";

} else{

messageFunctionRow = "Не является функцией А к В";

}

if(flagFunctionColumn){...} else{...}

# Листинг кода.

let error;

/\*\*

\* Коренктность ввода

\*/

function validate1(str) {

let array = false;

if (str.value.length > 0)

array = str.value.split(' ');

else

{

error = "Введите хотя бы один элемент";

array = false;

}

return array;

}

/\*\*

\* Коренктность ввода отношений

\*/

function validate2(text){

let array;

let str;

array = false;

if(text.value.length > 0) {

array = new Array();

str = text.value.split('\n');

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

if (str[i].length > 0) {

array[i] = str[i].split(' ');

} else{

array = false;

error = "Одна из строк пустая"

}

}

} else{

array = false;

error = "Введите хотя бы ОДНО отношение";

}

return array;

}

/\*\*

\* Удаление повторяющихся элементов множеств

\*/

function DeleteRepeatElements(array)

{

array.sort();

for(let i = 0; i < array.length - 1; i++)

{

if(array[i] == array[i+1])

{

array.splice(i,1); // удаление элемента массива, а на место его массив со следующим индексом

i--;

}

}

}

/\*\*

\* Проверка наличия элементов отношения во множествах

\*/

function check(array1, array2, Array){

let flag = true;

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

if(flag == true){

flag = array1.includes(Array[i][0]);

if(flag == false){

error = "Ошибка: введен элемент, которго нет в одном из множеств";

break

}

}

}

if(flag == true){

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

if(flag == true){

flag = array2.includes(Array[i][1]);

if(flag == false){

error = "Ошибка: введен элемент, которго нет в одном из множеств";

break

}

}

}

}

return flag;

}

function analysis(){

let text = document.getElementById('array');

let str1 = document.getElementById('array1');

let str2 = document.getElementById('array2');

let array1 = validate1(str1);

let array2 = validate1(str2);

let row = array1.length;

let column = array2.length;

let array = validate2(text);

let flag;

if(array != false)

flag = check(array1, array2, array);

if(array1 == false || array2 == false || array == false || flag == false){

alert(error);

} else{

let arrayElementsA = array1;

DeleteRepeatElements(arrayElementsA);

let arrayElementsB = array2;

DeleteRepeatElements(arrayElementsB);

let matrix = new Array();

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

matrix[i] = new Array();

}

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

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

matrix[i][j] = 0;

}

}

let arrayKeyA = {};

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

arrayKeyA[i] = {num : arrayElementsA[i]};

}

let arrayKeyB = {};

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

arrayKeyB[i] = {num : arrayElementsB[i]};

}

/\*\*

\* выставление единиц на места

\*/

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

for(let a = 0; a < arrayElementsA.length; a++){

for(let b = 0; b < arrayElementsB.length; b++){

if (array[i][0] === arrayKeyA[a].num && array[i][1] === arrayKeyB[b].num){

matrix[a][b] = 1;

}

}

}

}

let output = "\_| ";

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

output += "<u>" + arrayElementsB[i] + "</u>" + " ";

}

output += "<br>";

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

output += arrayElementsA[i] + "| ";

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

output += matrix[i][j] + " ";

}

output += "<br>";

}

let flagFunctionRow = true;

let flagFunctionColumn = true;

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

let countRow = 0;

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

if (matrix[i][j] === 1) {

countRow += 1;

}

if (countRow > 1) {

flagFunctionRow = false;

break;

}

}

if (!flagFunctionRow)

break;

if (countRow === 0)

flagFunctionRow = false;

}

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

let countColumn = 0;

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

if (matrix[j][i] === 1) {

countColumn += 1;

}

if (countColumn > 1) {

flagFunctionColumn = false;

break;

}

}

if (!flagFunctionColumn)

break;

if (countColumn === 0)

flagFunctionColumn = false;

}

let messageFunctionRow = " ";

let messageFunctionColumn = " ";

if(flagFunctionRow){

messageFunctionRow = "Является функцией А к В";

} else{

messageFunctionRow = "Не является функцией А к В";

}

if(flagFunctionColumn){

messageFunctionColumn = "Является функцией B к A";

} else{

messageFunctionColumn = "Не является функцией B к A";

}

document.getElementById("Analysis").innerHTML = output;

document.getElementById("Result").innerHTML = messageFunctionRow + "<br>" + messageFunctionColumn;

}

}

# Выводы.

Во время выполнения лабораторной работы, были изучены функции их свойства, разработана программа для их определения функции между двумя множествами. Для разработки был использован язык JS.

# Список литературы.

1. Элементы математической логики. Отношения - <https://ppt-online.org/606090>.
2. Рефлексивные отношения - <https://hmong.ru/wiki/Reflexive_relation>.
3. Транзитивные отношения - <https://studfile.net/preview/4229244/page:3/>
4. Справочник по JavaScript - https://javascript.ru/manual
5. Р. Хаггарти, Дискретная математика для программистов / Техносфера, г. Москва, 2005 г. - 400с.