Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №1

«Операции над множествами»

Выполнил студент

группыИВТАСбд-11

Зюзин Г.А

Проверил Игонин А.Г.

Ульяновск, 2022 г.

# Оглавление.

[Оглавление. 2](#_Toc1)

[Цель работы. 3](#_Toc2)

[Метод выполнения.](#_Toc3)

[Листинг кода. 11](#_Toc4)

[Выводы. 18](#_Toc5)

[Список литературы.](#_Toc6)

# Цель работы.

Написать программу на JavaScript, которая выполняет действия над множествами (объединение, пересечение, дополнение A/В В/A, симметрическая разность) с проверкой элементов на формат ввода, который имеет вид jcib, где j– нечетная цифра, c – цифра, i - чётная цифра, b– буква. В программе ввод элементов множеств осуществляется пользователем.

# Метод выполнения.

Данная программа выполняет операции над двумя множествами. Программа должна совершать следующие действия: объединение, пересечение, дополнение, симметрическая разность.

Объединение ([рис. 1.1](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._1.1)) двух множеств – это множество, которое включает в себя все элементы этих множеств, также называют суммой двух множеств. (A ∪ *В =* {*х : х* ∈ *A* или *х* ∈ *В*}). Функция – union() (Листинг 1).

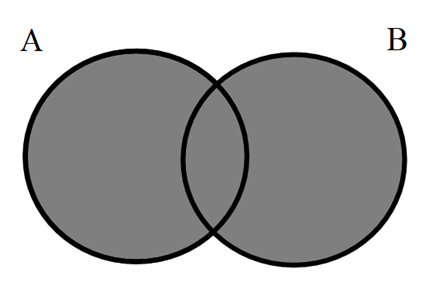


Рис 1.1.

Поскольку результатом данной операции является множество, которое содержит в себе все элементы обоих множеств, то реализация состоит в том, что в один массив засовываются все элементы множеств. После удаления всех повторяющихся элементов, результат – объединения множеств (массив всех элементов).

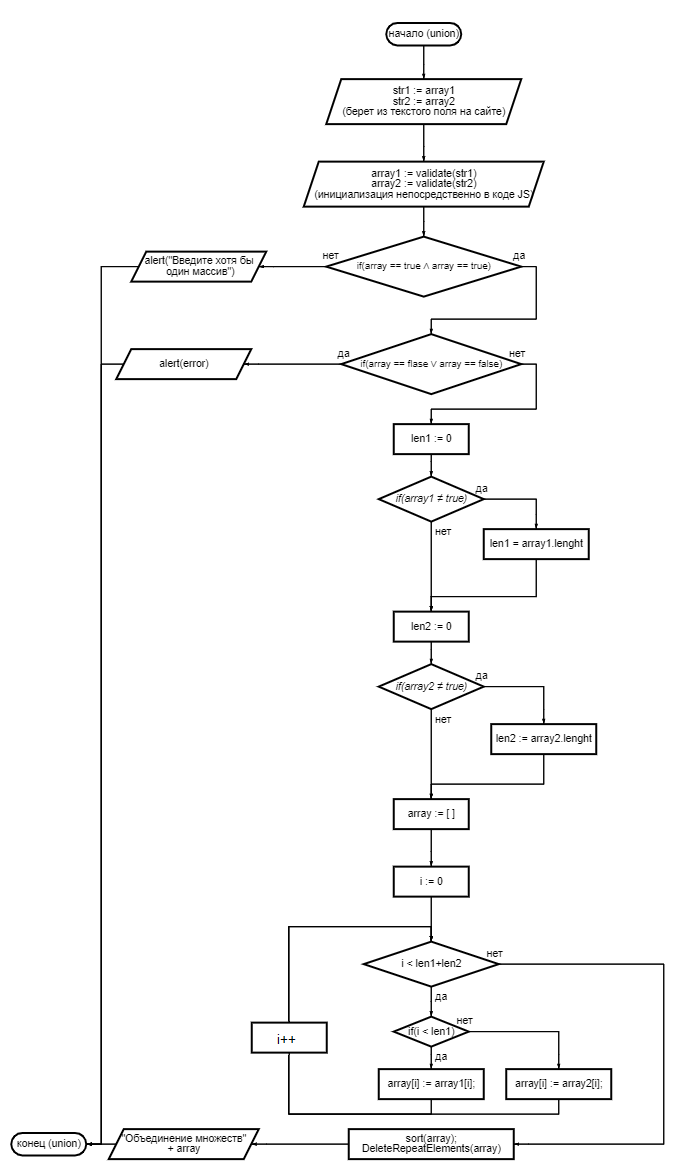


Рис 1.2.

Выше находится блок-схема, которая отображает как работает алгоритм объединения двух множеств ([рис 1.2](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._1.2)).

Листинг 1.

function **union()**{

...

else{

let len1 = 0;

if (array1 != true)

len1 = array1.length;

let len2 = 0;

if (array2 != true)

len2 = array2.length;

let array = [];

for(let i = 0; i < len1+len2; i++){

if (i < len1){

array[i] = array1[i];

}

else{

n = i - len1;

array[i] = array2[n];

}

}

sort(array);

DeleteRepeatElements(array);

document.getElementById("ResultUnion").innerText = "Объединение множеств: " + array;

}

}

Пересечение ([рис 2.1](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._2.1)) двух множеств – это множество, содержащее общие элементы множеств, то есть все элементы этого множества есть в обоих множеств. Также называется произведением множеств. (*А* ⋂ *В =* {*х: х* ∈ ⋂ и *х* ∈ В}). Функция – intersecrion() (Листинг 2).

Для написания данной функции сначала была написана функция CommonElements([рис. 2.2](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._2.1)), так как она пригодится в последующем. На вход она получает два массива, а на выходе массив с общими элементами.

Рис 2.1

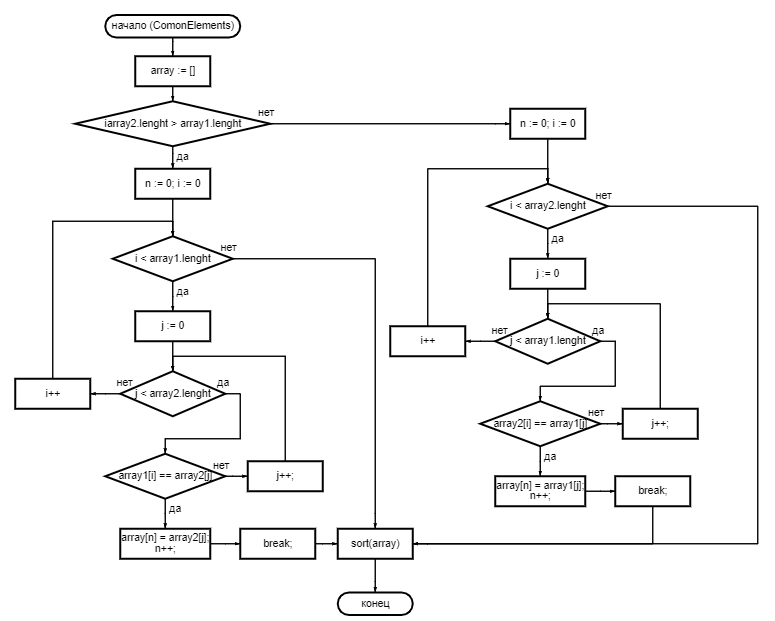
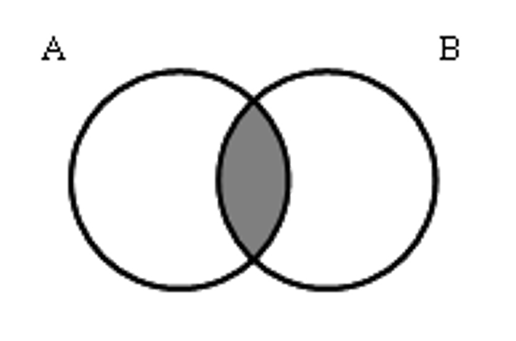


Рис 2.2.

Результатом данной операции является множество, которое содержит в себе все общие элементы обоих множеств, реализация состоит в том, что в один массив засовываются все общие элементы множеств. После удаления всех повторяющихся элементов, результат – пересечение множеств (массив общих элементов).

Листинг 2.

function **intersection()**{

...

else{

...

let array = CommonElements(array1, array2);

sort(array);

DeleteRepeatElements(array);

if(array.length == 0)

document.getElementById("ResultInteraction").innerText = "Общих элементов множеств НЕТ.";

else{...//вывод}

}

function **CommonElements**(array1, array2){

let array = [];

if(array2.length > array1.length){

let n = 0;

for(let i = 0; i < array1.length; i++){

for(let j = 0; j < array2.length; j++){

if(array1[i] == array2[j]){

array[n] = array2[j];

n++;

break;

}

}

}

}

else {...}

sort(array);

return array;

}

Дополнение ([рис. 3.1](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._3.1)) множества A/B – это множество, которое состоит из элементов множества А, которые при этом не принадлежат множеству В. А\В = {х : х ∈ ⋂ и. х ∉ В}. Функция – addition()(Листинг 3).

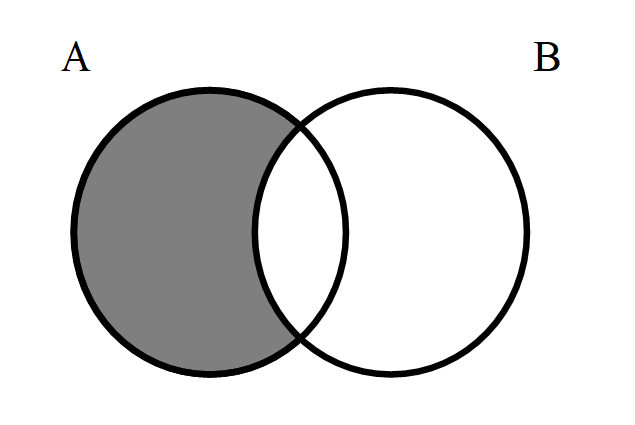


Рис. 3.1

Реализация дополнения множества состоится в том, что сначала находятся все общие элементы двух множеств (с помощью вышеупомянутой функции CommonElement). В зависимости от того для какого множества берётся дополнение, будут удаляться элементы общие элементы из этого множества.

Листинг 3

function **addition()**{

...

else{

...

let ArrayCommonElements = CommonElements(array1, array2);

let n = 0;

for(let i = 0; i < array1.length; i++) {

if(array1[i] == ArrayCommonElements[n]){

array1.splice(i, 1);

n++;

i--;

}

}

n = 0;

for(let i = 0; i < array2.length; i++) {

if(array2[i] == ArrayCommonElements[n]){

array2.splice(i, 1);

n++;

i--;

}

}

if(array2.length == 0 && array1.length == 0)

document.getElementById("ResultAddition").innerText = "А = В";

else if(array2 == true || array2.length == 0){...}

else if(array1 == true || array2.length == 0){...}

else{... }

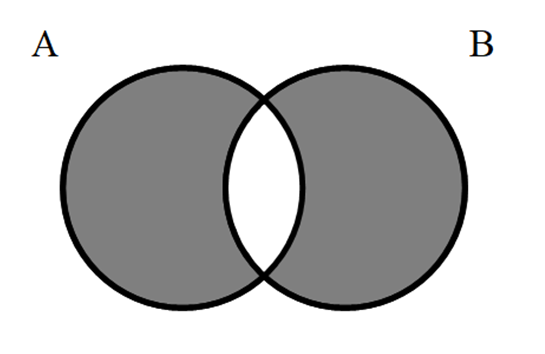
}

}

Симметрическая разность ([рис. 4.1](https://ek6yqv6m7d6trhms_sas_803_34328bddbc4a1e607dd7fd10f9533b90.onlyoffice.disk.yandex.net/7.0.1-62/web-apps/apps/documenteditor/main/index_loader.html?_dc=7.0.1-62&lang=ru&customer=%D0%AF%D0%BD%D0%B4%D0%B5%D0%BA%D1%81.%D0%94%D0%B8%D1%81%D0%BA&headerlogo=https%3A%2F%2Fyastatic.net%2Fs3%2Feditor%2F_%2Feditor_docs_icon_ru_v1.svg&frameEditorId=only-office-app&parentOrigin=https://disk.yandex.ru#_%D1%80%D0%B8%D1%81._4.1)) двух множеств – это множество, которое включает в себя элементы двух множеств, в отношении которых и проводится операция, кроме общих элементов этих множеств. Функция – SymmetricDifference() (Листинг 4).

Происходит это так: в один массив засовываются все элементы множеств. После удаления всех повторяющихся элементов, результат – объединения множеств. Находятся все общие элементы двух множеств (с помощью функции ArrayCommonElemebts) и удаляются из объединения.

Рис. 4.1



Листинг 4

function **SymmetricDifference()**{

...

else {

...

let ArrayCommonElements = CommonElements(array1, array2);

let array = [];

for(let i = 0; i < len1+len2; i++){

if (i < len1){

array[i] = array1[i];

}

else {

let n = i - len1;

array[i] = array2[n];

}

}

sort(array);

DeleteRepeatElements(array);

let n = 0;

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

if(array[i] == ArrayCommonElements[n]) {

array.splice(i, 1);

n++;

i--;

}

}

if(array.length == 0) {//вывод }

else{...}

}

}

Интерфейс данной программы (Рис. 5) состоит из двух полей ввода для каждого множества и четырёх кнопок под каждое действие. Выводом данной программы является результат над множествами (Рис. 6).

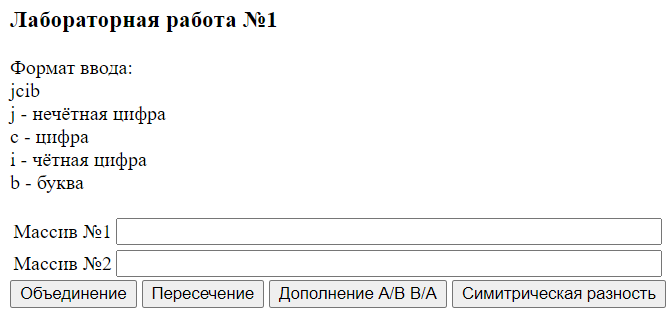
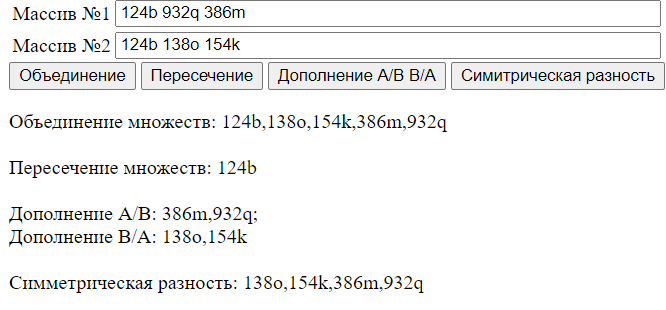


Рис. 5

Рис. 6



# Листинг кода.

/\*\*

\* Дополнительные функции

\*/

/\*\*

\* Алгоритм сортировки массива по возрастанию - пузырьковая сортировка

\*/

function sort(array)

{

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

{

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

{

if (array[j+1] < array[j])

{

let t = array[j+1];

array[j+1] = array[j];

array[j] = t;

}

}

}

}

/\*\*

\* Функция записи, проверки и сортировки элементов массивов

\*/

function validate(str)

{

let array = false;

if (str.value.length > 0)

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

else

array = true;

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

{

if(array[i].length != 4)

{

error = "Элемент множество не может быть такой длины (Подсказка: длина каждого элемента 4).";

array = false;

break;

}

/\*\*

\* Форма элемента - jcib

\* j - нечетная цифра

\* c - цифра

\* i - четная цифра

\* b - буква

\*/

if(array[i][0] % 2 == 0)

{

error = "Некоректный ввод (Подсказка: первый символ всех элементов - нечётная цифра).";

array = false;

break;

}

if (!((array[i][1] >= 48) || (array[i][1] <= 57)))

{

error = "Некоректный ввод (Подсказка: второй символ всех элементов - цифра)";

array = false;

break;

}

if(array[i][2] % 2 != 0)

{

error = "Некоректный ввод (Подсказка: третий символ всех элементов - чётная цифра).";

array = false;

break;

}

if((array[i][3] < 'a') || (array[i][3] > 'z'))

{

error = "Некоректный ввод (Подсказка: четвёртый символ всех элементов - буква)";

array = false;

break;

}

}

sort(array);

DeleteRepeatElements(array);

return array;

}

/\*\*

\* Удаление повторяющихся элементов множеств

\*/

function DeleteRepeatElements(array)

{

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--;

}

}

}

/\*\*

\* Функция, которая возвращает массив общих элементов

\*/

function CommonElements(array1, array2)

{

let array = [];

if(array2.length > array1.length)

{

let n = 0;

for(let i = 0; i < array1.length; i++)

{

for(let j = 0; j < array2.length; j++)

{

if(array1[i] == array2[j])

{

array[n] = array2[j];

n++;

break;

}

}

}

}

else

{

let n = 0;

for(let i = 0; i < array2.length; i++)

{

for(let j = 0; j < array1.length; j++)

{

if(array2[i] == array1[j])

{

array[n] = array1[j];

n++;

break;

}

}

}

}

sort(array);

return array;

}

/\*\*

\* 1.Объединение

\*/

function union()

{

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

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

let array1 = validate(str1);

let array2 = validate(str2);

if(array1 == true && array2 == true)

alert("Введите хотя бы один массив");

if((array1 == false) || (array2 == false))

alert(error);

else

{

let len1 = 0;

if (array1 != true)

len1 = array1.length;

let len2 = 0;

if (array2 != true)

len2 = array2.length;

let array = [];

for(let i = 0; i < len1+len2; i++)

{

if (i < len1)

{

array[i] = array1[i];

}

else

{

n = i - len1;

array[i] = array2[n];

}

}

sort(array);

DeleteRepeatElements(array);

document.getElementById("ResultUnion").innerText = "Объединение множеств: " + array;

}

}

/\*\*

\* 2. Пересечение

\*/

function intersection()

{

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

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

let array1 = validate(str1);

let array2 = validate(str2);

if(array1 == true && array2 == true)

alert("Введите хотя бы один массив");

if((array1 == false) || (array2 == false))

alert(error);

else

{

let len1 = 0;

if (array1 != true)

len1 = array1.length;

let len2 = 0;

if (array2 != true)

len2 = array2.length;

let array = CommonElements(array1, array2);

sort(array);

DeleteRepeatElements(array);

if(array.length == 0)

document.getElementById("ResultInteraction").innerText = "Общих элементов множеств НЕТ.";

else

document.getElementById("ResultInteraction").innerText = "Пересечение множеств: " + array;

}

}

/\*\*

\* 3. Дополнение A/B B/A (разность)

\*/

function addition()

{

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

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

let array1 = validate(str1);

let array2 = validate(str2);

if(array1 == true && array2 == true)

alert("Введите хотя бы один массив");

if((array1 == false) || (array2 == false))

alert(error);

else

{

let len1 = 0;

if (array1 != true)

len1 = array1.length;

let len2 = 0;

if (array2 != true)

len2 = array2.length;

let ArrayCommonElements = CommonElements(array1, array2);

let n = 0;

for(let i = 0; i < array1.length; i++) // Удаление общих элементов

{

if(array1[i] == ArrayCommonElements[n])

{

array1.splice(i, 1);

n++;

i--;

}

}

n = 0;

for(let i = 0; i < array2.length; i++) // Удаление общих элементов

{

if(array2[i] == ArrayCommonElements[n])

{

array2.splice(i, 1);

n++;

i--;

}

}

if(array2.length == 0 && array1.length == 0)

document.getElementById("ResultAddition").innerText = "А = В";

else if(array2 == true || array2.length == 0)

{

document.getElementById("ResultAddition").innerText = "Дополнение A/B: \

" + array1 + "; \nДополнение B/A: Множество В является подмножеством А (или В - пустое множество)";

}

else if(array1 == true || array2.length == 0)

{

document.getElementById("ResultAddition").innerText = "Дополнение A/B:\ " +

"Множество А является подмножеством В (или А - пустое множество)" + "; \nДополнение B/A: " + array2;

}

else

{

document.getElementById("ResultAddition").innerText = "Дополнение A/B: \

" + array1 + "; \nДополнение B/A: " + array2;

}

}

}

/\*\*

\* 4.Симметрическая разность

\*/

function SymmetricDifference()

{

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

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

let array1 = validate(str1);

let array2 = validate(str2);

if(array1 == true && array2 == true)

alert("Введите хотя бы один массив");

if((array1 == false) || (array2 == false))

alert(error);

else

{

let len1 = 0;

if (array1 != true)

len1 = array1.length;

let len2 = 0;

if (array2 != true)

len2 = array2.length;

let ArrayCommonElements = CommonElements(array1, array2);

let array = [];

for(let i = 0; i < len1+len2; i++)

{

if (i < len1)

{

array[i] = array1[i];

}

else

{

let n = i - len1;

array[i] = array2[n];

}

}

sort(array);

DeleteRepeatElements(array);

let n = 0;

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

{

if(array[i] == ArrayCommonElements[n])

{

array.splice(i, 1);

n++;

i--;

}

}

if(array.length == 0)

{

document.getElementById("SymmetricDifference").innerText = "Симметрическая разность: множества\

не имеют различных элементов" + array;

}

else

{

document.getElementById("SymmetricDifference").innerText = "Симметрическая разность: " + array;

}

}

}

# Выводы.

Во время выполнения лабораторной работы, были изучены методы определия свойств отношений, разработан программы для демонстрации. Для разработки был использован язык JS.

# Список литературы.

1. Множества - виды, операции и примеры решения - https://nauka.club/matematika/algebra/mnozhestvo.html
2. Множества и операция над ними - http://www.mathprofi.ru/mnozhestva.html
3. Справочник по JavaScript - https://javascript.ru/manual