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 над множествами»

Выполнил студент

группы ИВТАСбд-12

Худойдотов Н. Ш.

Ульяновск, 2022

**Оглавление**

[**Цель работы** 3](#_Toc106093467)

[**Особенности реализации** 3](#_Toc106093468)

[**Блок схема** 3](#_Toc106093469)

[**Интерфейс** 4](#_Toc106093470)

[**Листинг кода** 5](#_Toc106093471)

[**Выводы о проделанной работе** 7](#_Toc106093472)

[**Список литературы** 7](#_Toc106093473)

# **Цель работы**

Требуется написать программу, которая выполняет операции над множествами, а именно пересечение, объединение, симметрическая разница, дополнение. Кроме того, необходимо сделать валидацию данных по заданному формату: jcb, где первый символ – нечётная цифра, второй – цифра, третий – буква. Пользователь имеет возможность ввода данных.

# **Особенности реализации**

Работа выполнена в виде сайта, после ввода данных и нажатия кнопки отправить осуществляется вызов функции через опцию “on click”. Запускается скрипт, который обрабатывает полученные данные и возвращает результат в специальный блок на страничке. Если введенные данные некорректны пользователь получает ошибку с указанием проблемы.

# **Блок схема**

**Main()**

a = document.getElementById("mass1").value;  
b = document.getElementById("mass2").value;

*searchErrors*(a)

&& *searchErrors*(b)

**false**

**false**

*searchErrors*(str)

*alert*(error\_text);

**true**

**true**

const value\_a = new Set(a.split(" "));  
const value\_b = new Set(b.split(" "));

let unification = new Set([...value\_a, ...value\_b]);  
document.getElementById('unification').innerText = "Объединение: " + [...unification.values()] + "\n";

let intersection = new Set([...value\_a].filter(x => value\_b.has(x)));  
document.getElementById('intersection').innerText = "Пересечение: " + [...intersection.values()] + "\n";

let difference = new Set([...value\_a].filter(x => !value\_b.has(x)));  
document.getElementById('difference').innerText = "Разница A/B: " + [...difference.values()] + "\n";  
let difference2 = new Set([...value\_b].filter(x => !value\_a.has(x)));  
document.getElementById('difference2').innerText = "Разница B/A: " + [...difference2.values()] + "\n";

let union = new Set([...value\_a, ...value\_b]);  
let inters = new Set([...value\_a].filter(x => value\_b.has(x)));  
let symmetricDifference = new Set([...union].filter(x => !inters.has(x)));  
document.getElementById('symmetricDifference').innerText = "Симметричная разница: " + [...symmetricDifference.values()];

# **Интерфейс**

Интерфейс программы состоит из двух полей ввода, кнопки для расчета. Результат ввода выводится снизу. Пользователь вводит данные в первые два поля, нажимает кнопку и получает результат ниже.

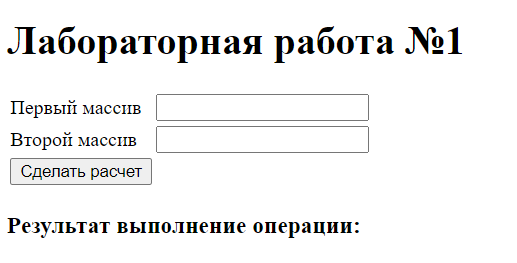
****

Рис. 1 – внешний вид

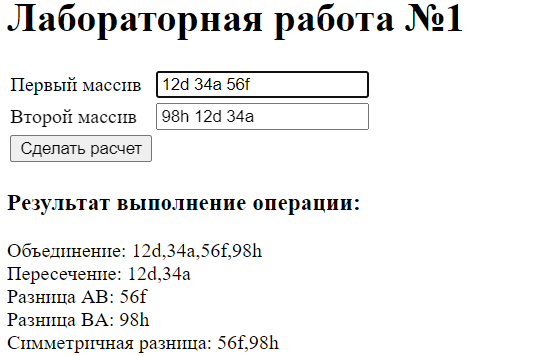


Рис. 2 – Получение результата

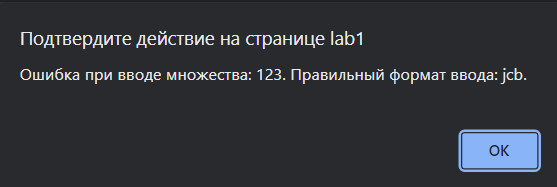


Рис. 2 – Ошибка при вводе

# **Листинг кода**

Первым делом рассмотрим функцию валидации элементов множества. Данная функция принимает оба элементов множества. Для начало проверяем поля ввода, если они пусты то выводится соответствующее сообщение, если нет идем дальше и проверяем на правильный формат ввода. Если элементы прошли проверку, функция возвращает true, иначе – false.

function searchErrors(str)  
{  
 let mass = str.split(" ");  
 if(a.length === 0 || b.length === 0)  
 {  
 ***error\_text*** = "Поля множества не должены быть пустыми.";  
 return false;  
 }  
 for(let i = 0; i < mass.length; i++)  
 {  
 if(!(mass[i][0] % 2 === 1 && (mass[i][1] >= 0 && mass[i][1] <= 9) && (mass[i][2] >= 'a' && mass[i][2] <= 'z') && mass[i].length === 3))  
 {  
 ***error\_text*** = "Ошибка при вводе множества: " + mass[i] + ". Правильный формат ввода: jcb.";  
 return false;  
 }  
 }  
 return true;  
}

Переходим к основной функции к работе операций над множествами.

В переменные **a** и **b** присваиваем значения из полей ввода. Дальше проверяем на валидность и если значения true то идем дальше, если false то выводится сообщения об ошибки.

function main()  
{  
 a = ***document***.getElementById("mass1").value;  
 b = ***document***.getElementById("mass2").value;  
  
 //Проверка на ошибку  
 if(searchErrors(a) && searchErrors(b))  
 {  
 // Получаем два множества из строк, сплитуем по пробелу  
 let value\_a = new ***Set***(a.split(" "));  
 let value\_b = new ***Set***(b.split(" "));

**Первая операция – объединение**. Оператор **...** превращает наш набор в массив, и после объединения двух массивов конструктор **Set** удаляет повторяющиеся элементы.

//Объединяем два множества  
let unification = new ***Set***([...value\_a, ...value\_b]);  
***document***.getElementById('unification').innerText = "Объединение: " + [...unification.values()] + "\n";

**Вторая операция – пересечение.** Здесьвыполняем поиск по одному набору и проверяем, находится ли каждый элемент в другом наборе. Чтобы проверить, есть ли элемент в наборе, мы используем метод **has.**

//Пересечение двух множеств  
let intersection = new ***Set***([...value\_a].filter(x => value\_b.has(x)));  
***document***.getElementById('intersection').innerText = "Пересечение: " + [...intersection.values()] + "\n";

**Третья операция – дополнение.** Здесьвыполняем поиск по одному набору и проверяем, находится ли каждый элемент в другом наборе. Чтобы проверить, есть ли элемент в наборе, мы используем метод **has**, а затем удалим одинаковые элементы.

//Дополнение (Разница) A\B множест  
let difference = new ***Set***([...value\_a].filter(x => !value\_b.has(x)));  
***document***.getElementById('difference').innerText = "Разница A/B: " + [...difference.values()] + "\n";  
//Дополнение (Разница) B\A множест  
let difference2 = new ***Set***([...value\_b].filter(x => !value\_a.has(x)));  
***document***.getElementById('difference2').innerText = "Разница B/A: " + [...difference2.values()] + "\n";

**Четвёртая операция – симметрическая разность.** Здесь обрабатываем два набора **union** и **inters** и ищем симметрическую разностью.

let union = new ***Set***([...value\_a, ...value\_b]);  
let inters = new ***Set***([...value\_a].filter(x => value\_b.has(x)));  
let symmetricDifference = new ***Set***([...union].filter(x => !inters.has(x)));  
***document***.getElementById('symmetricDifference').innerText = "Симметричная разница: " + [...symmetricDifference.values()];

# **Выводы о проделанной работе**

Выполнив данную лабораторную работу, я научился обрабатывать событие клика и запускать с помощью него скрипт, а также обрабатывать введенные пользователем данные и возвращать результат в HTML документ.

# **Список литературы**

1. "Дискретная математика для программистов" Хаггарти.