**Работа №1. Операции над множествами**

**Цель работы.**

Необходимо написать программу, которая будет проводить операции над множествами, а именно нахождение: их объединения, пересечения, дополнения A/B и A/B, симметрическую разность. Должна присутствовать проверка на корректность введённых данных. Первый символ элемента – цифра, второй – буква, третий и четвертый – четная цифра. Пользователь имеет возможность ввода данных.

**Описание метода выполнения.**

В данной лабораторной работе необходимо выполнить следующие операции над множествами: объединение, пересечение, симметрическая разница, дополнение первого до второго и наоборот.

Объединением двух множеств A и B называется множество, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из этих множеств - либо A, либо B.

Пересечением двух множеств A и B называется множество, состоящее из всех тех и только тех элементов, которые принадлежат каждому из множеств A и B.

Дополнением A/B называется множество, состоящее из всех элементов множества A, которые не принадлежат B.

Симметрической разносностью множеств A и B называется множество, состоящее из всех тех и только тех элементов универсального множества, которые либо принадлежат A и не принадлежат B, либо наоборот, принадлежат B, но не A.

В начале необходимо создать html-структуру, с которой будет взаимодействовать пользователь. В данной лабораторной работе она состоит из следующих элементов:

* Двух полей для ввода множеств. Данные поля создаются при помощи тега <input>.
* Раскрывающегося списка, в котором пользователь имеет возможность выбрать ту операцию, которую он желает произвести с множествами. Данный список создается при помощи тега <select>. <td><select id="operations">  
   <option value="1">Объединение</option>  
   <option value="2">Пересечение</option>  
   <option value="3">Дополнение A/B</option>  
   <option value="4">Дополнение B/A</option>put  
   <option value="5">Сумметрическая разность</option>  
  </select></td>
* Кнопки, которая также создается при помощи тега <input>. После её нажатия программа начинает работать с множествами и выполнять выбранную операцию.
* Текстового поля, на которое выводится полученное после преобразований множество. Данное текстовое поле создается с помощью тега <textarea>.

Последующие формирование нового множества, исходя из выбранной операции, происходит в JavaScript файле. По нажатию кнопки программа начинает выполнять функцию “massivs”, которая в первую очередь проверяет корректность введенных множеств.

Вызывается функция “validate”. Если в множестве была допущена ошибка, данная функция выведет пользователю соответствующее предупреждение. function validate(str, p) {  
 let mass = false  
 if(str.length>0)  
 {  
 mass = str.split(' ')  
 for (let i = 0; i < mass.length; i++)  
 {  
 if ((mass[i][0]<'0')||(mass[i][0]>'9')||(mass[i][0] == null)) {  
 alert('Ошибка при вводе множества: '+ str +' в элементе '+ mass[i]);  
 mass= false;  
 }  
 if ((mass[i][1]<'a')||(mass[i][1]>'z')||(mass[i][1] == null)) {  
 alert('Ошибка при вводе множества: '+ str +' в элементе '+ mass[i]);  
 mass = false;  
 }  
 if (((mass[i][2]<'0')||(mass[i][2]>'9'))||(((mass[i][2]>='1')&(mass[i][2]<='9'))&(mass[i][2]%2 != 0))||(mass[i][2] == null)) {  
 alert('Ошибка при вводе множества: '+ str +' в элементе '+ mass[i]);  
 mass = false;  
 }  
 if (((mass[i][3]<'0')||(mass[i][3]>'9'))||(((mass[i][3]>='0')&(mass[i][3]<='9'))&(mass[i][3]%2 != 0))||(mass[i][3] == null)) {  
 alert('Ошибка при вводе множества: '+ str +' в элементе '+ mass[i]);  
 mass = false;  
 }  
 }  
 }  
 else  
 alert(p+'-ое множество не должно быть пустым');  
 return mass;  
}

Далее полученные строки проверяются на наличие лишних пробелов в конце строки, если такие есть функция “delete\_space” их удаляет и возвращает полученную строку. Это важно, так как далее деление строки на элементы массива будет проводится через пробел.

function delete\_space(str) {  
 if (str[str.length-1] == ' ') {  
 while (str[str.length-1] == ' ') {  
 str = str.substring(0, str.length-1);  
 }  
 }  
 return str;  
}

После полученные строки проверяются на наличие одинаковых членов. Если в одной строке имеются повторяющийся элементы, их необходимо удалить. Для этого в программе используется функция “delete\_elements”.

function delete\_elements(str, str1, mass) {  
 mass = str.split(' ');  
 str += ' ';  
 str1 = '';  
 for (let i=0; i < mass.length; i++) {  
 if ((count\_element(mass, mass[i]) > 1)&(mass[i] != null)) {  
 str1 += mass[i] + ' ';  
 let str\_search = mass[i];  
 for (let j = 0; j < mass.length; j++){  
 if (mass[j] == str\_search) {  
 mass[j] = null;  
 }  
 }  
 }  
 else  
 if (mass[i] != null)  
 str1 += mass[i] + ' ';  
 }  
 str1 = delete\_space(str1);  
 return str1;  
  
}

Далее исходя из того, какую операцию выбрал пользователь, программа переходит к соответствующей функции.

Операция объединения.

В качестве переменных в данной функции выступают две строки с множествами, пустая третья строка и пустой массив. Третей строке присваивается значение первой строки, пробела и второй строки соответственно. Далее выполняется деление по пробелу, полученные элементы записываются в пустой массив. После создается цикл, которой рассматривает каждый элемент массива на наличие в строке точно таких же элементов. Если данный элемент в массиве такой один он записывается в строку result (глобальная переменная), если же нет - то обнуляется.

function operation\_1(str1, str2, str3, mass)  
{  
 str3 = str1 + ' ' + str2;  
 mass = str3.split(' ');  
 for (let i = 0; i < mass.length; i++) {  
 if (count\_element(mass, mass[i])==1) {  
 ***result*** += mass[i] + ' ';  
 }  
 else{  
 mass[i] = null;  
 }  
 }  
}

Операция пересечения.

В качестве переменных в данной функции выступают две строки и два пустых массива. Первый и второй массивы записываются соответствующие строки с разделением по пробелу. Далее каждый элемент первого массива поверяется на наличие точно такого же элемента во втором массиве. Если один и тот же элемент есть и в первом, и во втором массиве, то он записывается в строку с результатом.

function operation\_2(str1, str2, mass1, mass2)  
{  
 mass1 = str1.split(' ');  
 mass2 = str2.split(' ');  
 for (let i = 0; i < mass1.length; i++) {  
 if (count\_element(mass2, mass1[i]) >= 1) {  
 ***result*** += mass1[i] + ' ';  
 }  
 }  
}

Операция дополнения A/B (операция дополнения B/A работает аналогично).

В качестве переменных в данной функции выступают две строки и два пустых массива. Первый и второй массивы записываются соответствующие строки с разделением по пробелу. Далее запускается цикл, которой проверяет, если данный элемент первого массива в массиве второго, если нет, то элемент записывается в строку с результатом.

function operation\_3(str1, str2, mass1, mass2)  
{  
 mass1 = str1.split(' ');  
 mass2 = str2.split(' ');  
 for (let i = 0; i < mass1.length; i++) {  
 if (count\_element(mass2, mass1[i]) == 0) {  
 ***result*** += mass1[i] + ' ';  
 }  
 }  
}

Операция симметрической разности.

В качестве переменных в данной функции выступают две строки и два пустых массива. Первый и второй массивы записываются соответствующие строки с разделением по пробелу. Далее запускается цикл, которой проверяет, если данный элемент первого массива в массиве второго, если нет, то элемент записывается в строку с результатом. После запускается ещё один цикл, который уже наоборот ищет элементы второго массива, не встречающиеся в первом массиве.

function operation\_5(str1, str2, mass1, mass2)  
{  
 mass1 = str1.split(' ');  
 mass2 = str2.split(' ');  
 for (let i = 0; i < mass1.length; i++) {  
 if (count\_element(mass2,mass1[i]) == 0) {  
 ***result*** += mass1[i] + ' ';  
 }  
 }  
 for (let i = 0; i < mass2.length; i++) {  
 if (count\_element(mass1,mass2[i]) == 0) {  
 ***result*** += mass2[i] + ' ';  
 }  
 }  
}

**Список использованных источников.**

<https://guruweba.com/html/teg-select-html-vypadayuschiy-spisok/> - информация про использование тега <select>, создание раскрывающегося списка в html