Содержание

[**ВВЕДЕНИЕ** 3](#_Toc528727426)

[**Описание задания, цель работы** 4](#_Toc528727427)

[**Ход выполнения работы** 5](#_Toc528727428)

[**Вывод** 9](#_Toc528727429)

[**Список литературы** 10](#_Toc528727430)

[**Приложение** 11](#_Toc528727431)

# **ВВЕДЕНИЕ**

В современном JavaScript появился новый, «более красивый» синтаксис для классов.Новая конструкция class – удобный «синтаксический сахар» для задания конструктора вместе с прототипом.На данный момент использование классов являются одним из самых популярных способов структурирования программных проектов.

В результате выполнения лабораторной работы необходимо решить задачу №2 лабораторной работы №3 с применением ОО стиля.

# **Описание задания, цель работы**

Заданием лабораторной работы является решение задачи согласно варианту, используя средства JavaScript.

Целью лабораторной работы решение задачу №2 лабораторной работы №3 с применением ОО стиля.

В соответствии с заданием необходимо запросить у пользователя размерность матрицы, а также число, на которое ее необходимо умножить.

Для достижения поставленной цели необходимо реализовать класс matrix. Класс должен иметь методы:

* возвращающий заданную строку/столбец o Изменяющий значение заданного элемента
* класс должен хранить свойства:
* размерность матрицы
* список свойств матрицы (квадратная, единичная).

Свойства должны автоматически пересчитываться при изменении значений матрицы.

Для выполнения лабораторной работы был выбран вариант№8.

# **Ход выполнения работы**

Вариант №8. Второе задание данного варианта предполагает работу с матрицами: «Умножение матрицы на константу».

Для выполнения лабораторной работы создадим два файла: index.htmlи lab4.js.

Файл index.htmlбудет содержать базовую разметку страницы. Исходный код данного файла имеет следующую структуру:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Лабораторнаяработа №4</title>

</head>

<body>

<div id="app"></div>

<scriptsrc="lab4.js"></script><!--подключение выполнения скрипта на js-->

</body>

</html>

Строка<scriptsrc="lab4.js"></script>говорит о том, что файл lab.jsнаходится в одной папке с файлом index.html. Когда браузер открывает HTML-страницу, и видит <script> - он первым делом выполнит код, а только потом продолжит чтение страницы дальше.

После того, как начнется чтение файла lab.jsпроизойдет обращение к следующему коду:

document.getElementById("app").innerHTML = `

<div>

<p> Укажите размер матрицы </p>

<input type='number' id='first' min='0'> x

<input type='number' id='second' min='0'>

<p><button id='get-values'>Создатьматрицу</button></p>

</div>

<div id='matrix'></div>

<hr/>

<div id='buttonConst'></div>

<divid='result'></div>

`;

Данная конструкция кода позволяет элементы для ввода размерности матрицы. Результат работа показан на рисунке 1.

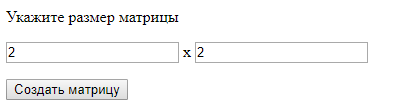


Рисунок 1 – Текст лабораторной работы

Для того, чтобы вывести полученный результат,воспользуемся следующей конструкцией:

createInput(i) {

const input = document.createElement("input");

input.setAttribute("type", "number");

input.style.width = "50px";

input.style.height = "50px";

input.className = "matrix-value" + " l-" + i;

return input;

}

После ввода размерности матрицы получим поля ввода, которые показаны на рисунке 2.

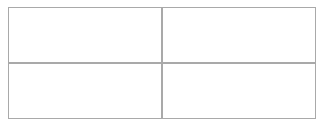


Рисунок 2 – Созданные поля для ввода элементов матрицы

Получим аргументы с строки ввода и переведемих в число:

let matrixA = parseInt(matrixAvalue, 10);

let matrixB = parseInt(matrixBvalue, 10);

this.size = [matrixA, matrixB];

Создадим матрицу из полученного массива данных

createMatrixOfInput(sizeA, sizeB) {

const div = document.createElement("div");

let arr = Array(sizeB)

.fill("")

.map(item =>this.createLine(sizeA));

arr.map(item =>div.appendChild(item));

returndiv;

}

После того как введем элементы матрицы, константу и нажмем кнопку «Посчитать» выдается результат, который показан на рисунке 3.

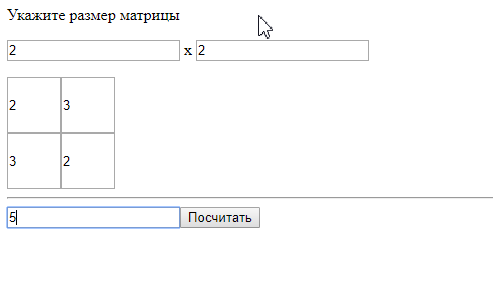


Рисунок 3 – Результат выполнения

За выполнение умножения матрицы на константу отвечает следующий код:

multMatrixNumber(a, A) {

var m = A.length,

n = A[0].length,

B = [];

for (vari = 0; i< m; i++) {

B[i] = [];

for (var j = 0; j < n; j++) {

B[i][j] = a \* A[i][j];

}

}

return B.filter(i =>i[0]);

}

render() {

constgetValue = document.getElementById("get-values");

getValue.addEventListener("click", this.handleCreateMatrix);

}

}

constnewMatrix = new Matrix();

newMatrix.render();

# **Вывод**

В процессе выполнения лабораторной работы была переписана лабораторная работа №3 по работе с матрицами с использованием ОО стилей.В ходе написания лабораторной работы была изученыконструкции по работе с классами, управление размерностью матрицы, а также свойства матриц.

Результатом работы стала реализованная лабораторная работа в ОО стиле.

# **Список литературы**

1. <https://learn.javascript.ru>
2. <https://developer.mozilla.org/ru/docs/Web/JavaScript>

# **Приложение**

**Исходный код файла lab.js**

document.getElementById("app").innerHTML = `

<div>

<p> Укажите размер матрицы </p>

<input type='number' id='first' min='0'> x

<input type='number' id='second' min='0'>

<p><button id='get-values'>Создатьматрицу</button></p>

</div>

<div id='matrix'></div>

<hr/>

<div id='buttonConst'></div>

<div id='result'></div>

`;

class Matrix {

constructor(name) {

this.size = [];

this.property = {};

this.constInputValue = Number;

this.matrix = [];

this.isMatrixCreated = false;

this.handleCreateMatrix = this.handleCreateMatrix.bind(this);

this.multMatrixNumber = this.multMatrixNumber.bind(this);

this.createInput = this.createInput.bind(this);

this.createLine = this.createLine.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

this.createMatrixOfInput = this.createMatrixOfInput.bind(this);

}

//получение данных с полей ввода

handleCreateMatrix(e) {

constmatrixAvalue = document.getElementById("first").value;

constmatrixBvalue = document.getElementById("second").value;

//превращение полученного аргумента с строки и перевод его в число

let matrixA = parseInt(matrixAvalue, 10);

let matrixB = parseInt(matrixBvalue, 10);

this.size = [matrixA, matrixB];

if (this.isMatrixCreated) return;

const matrix = document.getElementById("matrix");

constconstInput = document.createElement("input");

constconstSubmit = document.createElement("button");

constboxConst = document.getElementById("buttonConst");

//кнопка

this.resultElement = document.getElementById("result");

constSubmit.innerText = "Посчитать";

constInput.setAttribute("placeholder", "Введитеконстанту");//полевводаконстантынакоторуюнужноумножитьматрицу

if (!matrixA || !matrixB || matrixA< 1 || matrixB< 1) return;

this.constInput = constInput;

matrix.appendChild(this.createMatrixOfInput(matrixA, matrixB));

boxConst.appendChild(constInput);

boxConst.appendChild(constSubmit);

constSubmit.addEventListener("click", this.handleSubmit);

this.isMatrixCreated = true;

e.target.remove();

}

//создание матрицы из полученного массива данных

createMatrixOfInput(sizeA, sizeB) {

const div = document.createElement("div");

let arr = Array(sizeB)

.fill("")

.map(item =>this.createLine(sizeA));

arr.map(item =>div.appendChild(item));

return div;

}

handleSubmit() {

constconstInputValue = parseInt(this.constInput.value, 10);

constconstanta = constInputValue;

constarrElement = document.querySelectorAll(".matrix-value");

let result = [];

for (let i = 0; i<this.size[1]; i++) {

constarrElems = document.querySelectorAll(`.l-${i}`);

constarr = [...arrElems];

constarrValue = arr.map(i =>parseInt(i.value, 10));

result.push(arrValue);

}

const matrix = this.multMatrixNumber(constanta, result);

this.matrix = matrix;

const output = matrix.map(item => {

return `[ ${item} ]`;

});

this.resultElement.innerText = `[ ${output} ]`;

}

//занесение данных в столбец

getColumn(number) {

return this.matrix.filter(i =>i[number]);

}

//занесение данных в строку

getRow(number) {

returnthis.matrix[number];

}

//формирование матрицы

changeValue(column, row, value) {

this.matrix[row][column] = value;

}

//вывод на экран полей для ввода данных матрицы

createInput(i) {

const input = document.createElement("input");

input.setAttribute("type", "number");

input.style.width = "50px";

input.style.height = "50px";

input.className = "matrix-value" + " l-" + i;

returninput;

}

createLine(sizeA) {

const div = document.createElement("div");

let arr = Array(sizeA)

.fill("")

.map((item, i) =>this.createInput(i));

arr.map(item =>div.appendChild(item));

returndiv;

}

//выполнение умножения матрицы на костанту

multMatrixNumber(a, A) {

var m = A.length,

n = A[0].length,

B = [];

for (vari = 0; i< m; i++) {

B[i] = [];

for (var j = 0; j < n; j++) {

B[i][j] = a \* A[i][j];

}

}

return B.filter(i =>i[0]);

}

render() {

constgetValue = document.getElementById("get-values");

getValue.addEventListener("click", this.handleCreateMatrix);

}

}

constnewMatrix = new Matrix();

newMatrix.render();