ГУАП КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

|  |  |  |
| --- | --- | --- |
| ассистент |  | М.А. Мурашова |
| должность, уч. степень, звание | подпись, дата | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| ОСНОВЫ JAVASCRIPT |
| по дисциплине: WEB-ТЕХНОЛОГИИ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |
| --- | --- | --- |
| СТУДЕНТ ГР. | 4232 | Е.А.Уткина |
|  | подпись, дата | инициалы, фамилия |

Санкт-Петербург

2025

**Содержание**

[1. Цель работы 3](#_Toc192604660)

[2. Задание 3](#_Toc192604661)

[3. Вариант задания (25) 3](#_Toc192604662)

[4. Названия средств, использованных при выполнении работы 4](#_Toc192604663)

[5. Выполнение задания 4](#_Toc192604664)

[6. Листинг программного кода 9](#_Toc192604665)

[7. Выводы 10](#_Toc192604666)

# 1. Цель работы

Знакомство с языком JavaScript.

# 2. Задание

**Базовое задание**

Используя редактор javascript (см. листинг 1) написать программу, которая выполняет задание по варианту (базовая часть) для матрицы произвольного размера (n\*m) и использует массив. Обязательно использовать методы массива. Вывести на страницу матрицу в форме прямоугольника. Размеры матрицы получать через диалог с пользователем.

**Расширенное задание**

Нарисовать заданную вариантом фигуру, используя объект canvas и образец скрипта из листинга 2. Повторяющие фрагменты формировать с помощью циклов.

Размер фигуры должен задаваться пользователем через диалог. Под размером понимается количество элементов по вертикали и горизонтали. На пример, для варианта 1 задание можно сформулировать так: построить фигуру, состоящую из четного количества (n) вертикальных синий линий, которые пересекаются с нечетным количеством (n-1) горизонтальных линий, при чем меньшая половина горизонтальных линий синего цвета, а большая половина красного цвета. Пересечения линий должны образовывать квадраты.

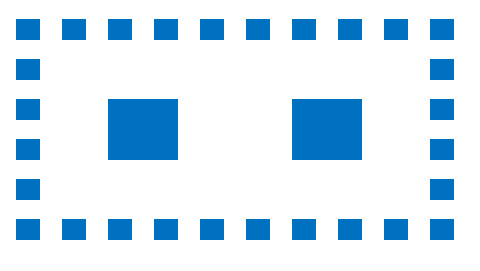
При желании до рисования фигуры можно сформировать матрицу, отражающую заполнение прямоугольника, в который вписана фигура, элементами, а потом визуализировать эту матрицу.

# 3. Вариант задания (25)

Базовое задание:

9) Все элементы матрицы, которые отличаются от заданного f, заменить нулями.

Расширенное задание:



# 4. Названия средств, использованных при выполнении работы

1. Редактор кода Visual Studio Code.

2. Браузер Google Chrome.

# 5. Выполнение задания

Ниже представлены таблицы (1-4) описывающие все переменные, методы и функции.

Таблица 1 – описание всех переменных программ

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| generateMatrix | Function | Функция, создающая матрицу заданного размера с элементами как произведение индексов |
| rows | Number | Количество строк в матрице, введенное пользователем |
| cols | Number | Количество столбцов в матрице, введенное пользователем |
| f | Number | Значение, которое остаётся в матрице, остальные заменяются на 0 |
| matrix | Array (2D) | Исходная матрица, заполненная произведениями индексов |
| initialResult | String | Строковое представление исходной матрицы для вывода с разделением табуляцией |
| modifiedMatrix | Array (2D) | Модифицированная матрица, где элементы, не равные f, заменены на 0 |
| myResult | String | Строковое представление модифицированной матрицы для вывода |
| canvas | Object | Объект канваса для рисования |
| context | Object | 2D-контекст канваса для выполнения операций рисования |
| blue | String | Цвет заливки (синий) |
| i | Number | Счетчик строк для циклов |
| j | Number | Счетчик столбцов для циклов |

Таблица 2 – описание использованных методов массива

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Параметры** | **Назначение** |
| Array.from | { length: n }, callback | Создаёт массив длиной n, заполняя его результатами вызова callback-функции |
| map | callback(element, index) | Создаёт новый массив, применяя callback к каждому элементу исходного |
| join | separator | Объединяет элементы массива в строку с указанным разделителем |

Таблица 3 – описание использованных методов других стандартных объектов

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Параметры** | **Назначение** |
| parseInt() | string | Преобразует строку в целое число, игнорируя нечисловые символы после числа |
| isNaN | value | Проверяет, является ли значение NaN (не числом) |
| prompt | message | Выводит диалоговое окно для ввода данных пользователем, возвращает строку |
| getContext | contextType | Возвращает объект контекста для рисования (тут "2d") |
| clearRect | x, y, width, height | Очищает указанную область канваса |
| fillRect | x, y, width, height | Очищает указанную область канваса |
| getElementById | id | Возвращает элемент DOM с указанным идентификатором |

Таблица 4 – описание использованных стрелочных функций

|  |  |  |
| --- | --- | --- |
| **Стрелочная функция** | **Параметры** | **Назначение** |
| В Array.from для строк | (\_, i) | Создаёт массив строк, где каждая строка — массив произведений (i + 1) \* (j + 1) |
| В Array.from для столбцов | (\_, j) | Заполняет строку значениями (i + 1) \* (j + 1) для каждого столбца |
| В map для initialResult | row | Преобразует каждую строку матрицы в строку с элементами, разделёнными \t |
| В map для modifiedMatrix (внешний) | row | Применяет замену элементов к каждой строке матрицы |
| В map для modifiedMatrix (внутренний) | element | Заменяет элемент на 0, если он не равен f, иначе оставляет без изменений |
| В map для myResult | row | Преобразует каждую строку модифицированной матрицы в строку с \t |

На рисунках 1-4 представлена работа программы базового задания, вычисляющая матрицу согласно варианта задания. У пользователя запрашивается количество строк и столбцов, если же введены не числа, то выводится ошибка, если же всё в порядке – выводится матрица.

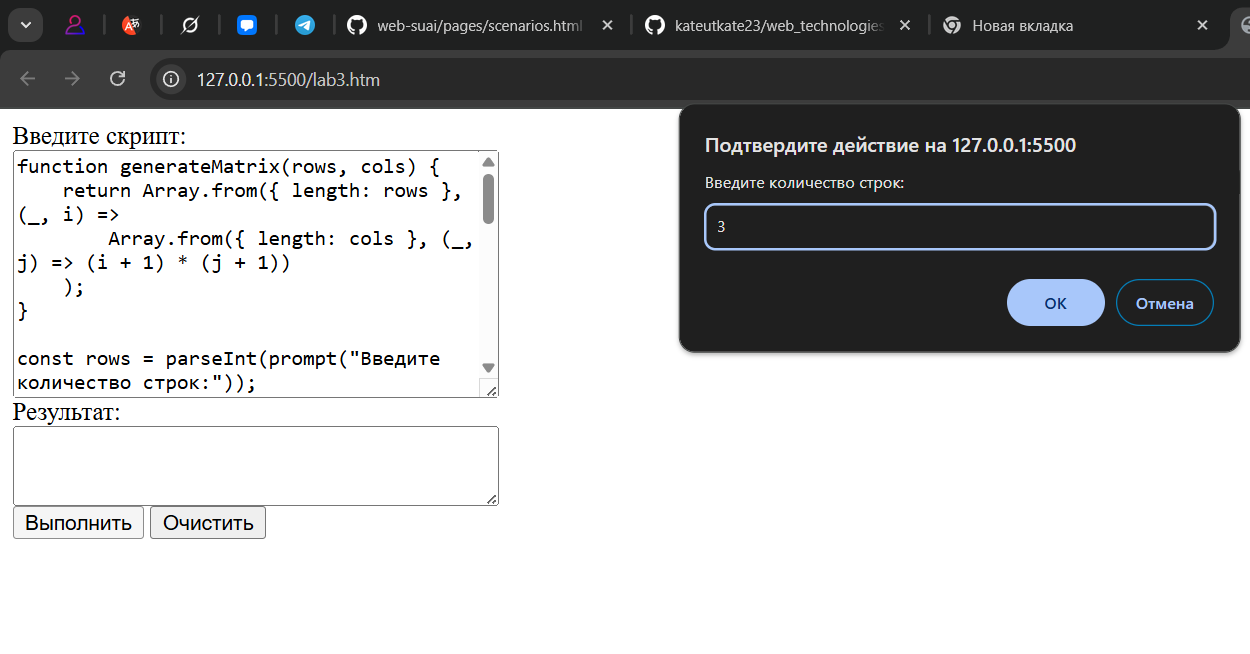


Рисунок 1 – запрос у пользователя количества строк

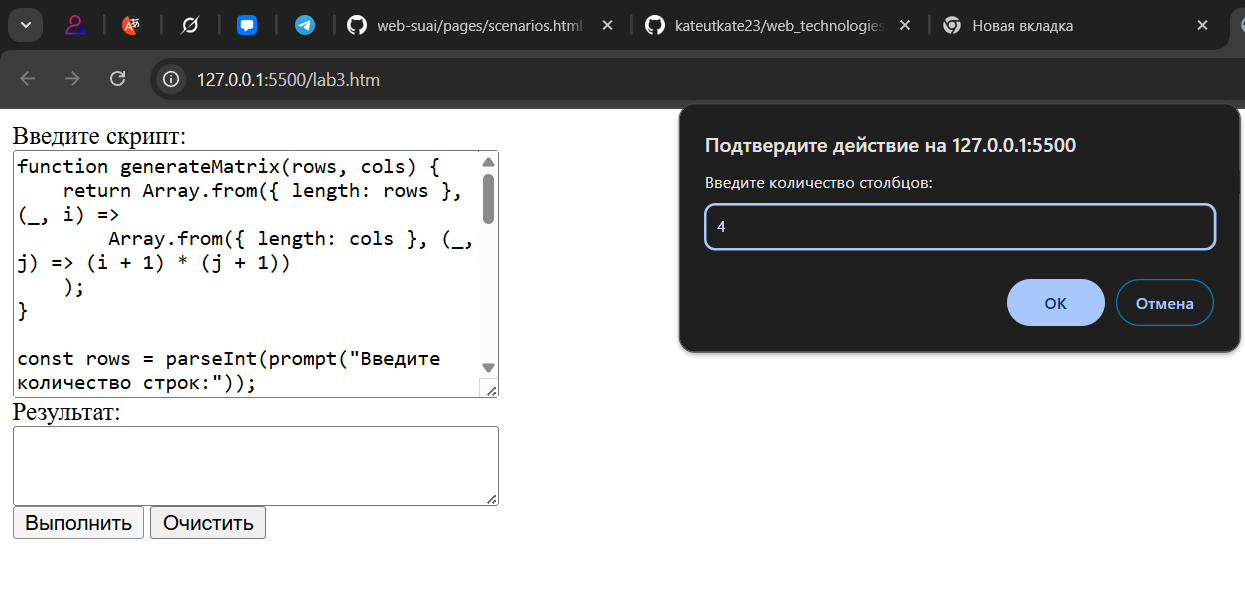


Рисунок 2 – запрос у пользователя количества столбцов

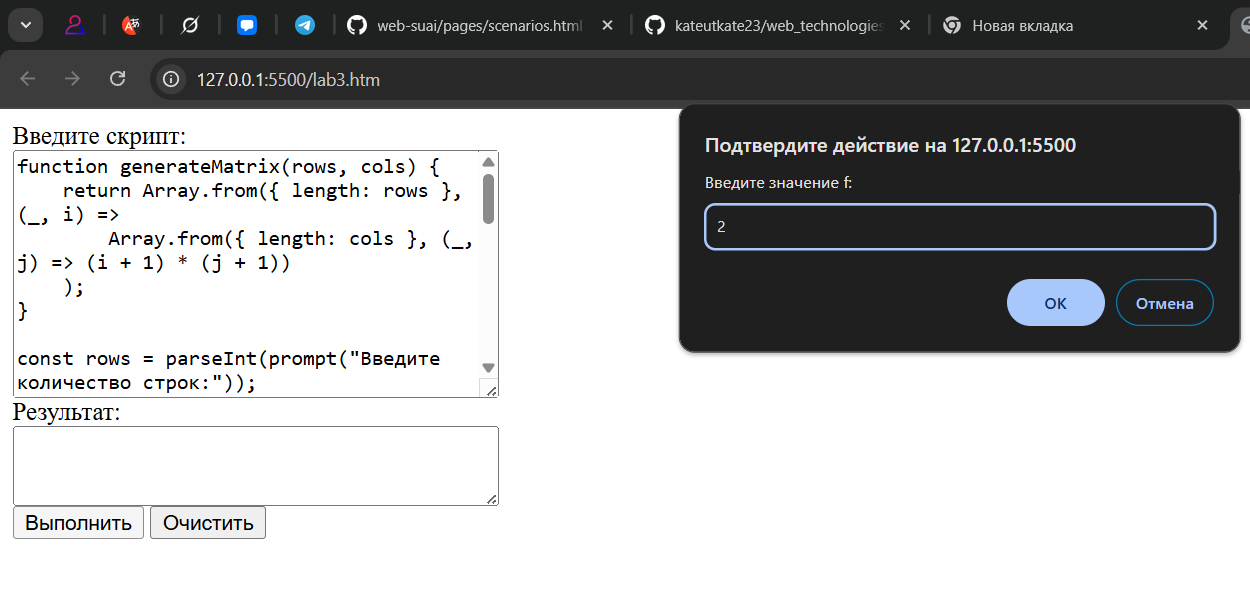


Рисунок 3 – запрос у пользователя значения f

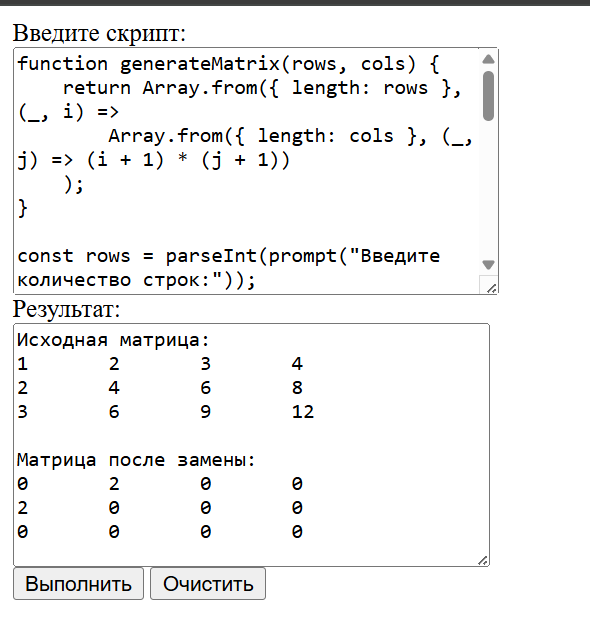
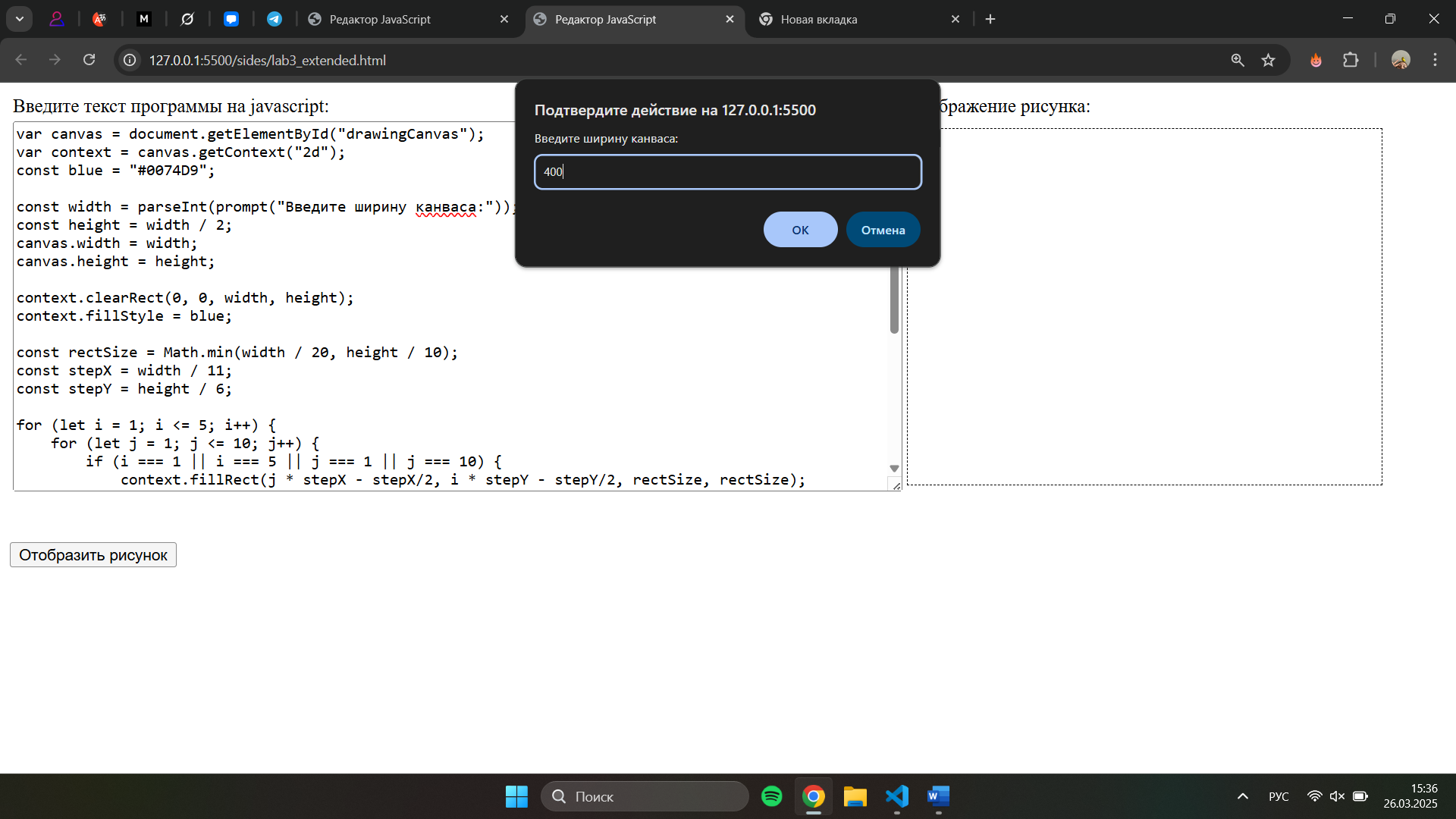


Рисунок 4 – вывод при успешном вводе

На рисунке 5 представлен результат работы программы генерирующий рисунок с заданным в варианте паттерном.



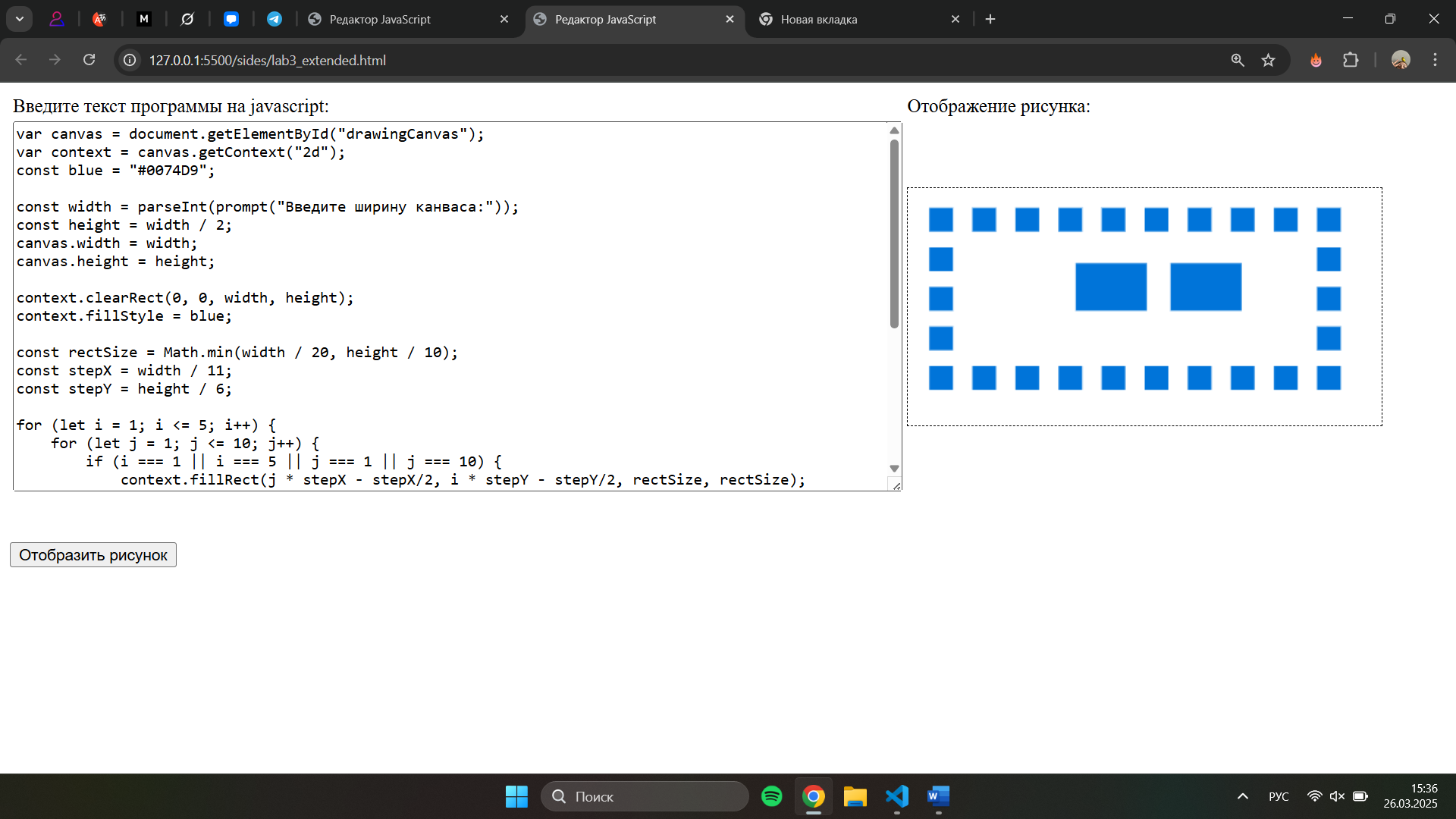


Рисунок 5 – вывод работы расширенного задания на страниц

# 6. Листинг программного кода

**Базовое задание:**

function generateMatrix(rows, cols) {

return Array.from({ length: rows }, (\_, i) =>

Array.from({ length: cols }, (\_, j) => (i + 1) \* (j + 1))

);

}

const rows = parseInt(prompt("Введите количество строк:"));

const cols = parseInt(prompt("Введите количество столбцов:"));

const f = parseInt(prompt("Введите значение f:"));

if (!isNaN(rows) && !isNaN(cols) && !isNaN(f) && rows > 0 && cols > 0) {

const matrix = generateMatrix(rows, cols);

let initialResult = matrix.map(row => row.join('\t')).join('\n');

const modifiedMatrix = matrix.map(row =>

row.map(element => (element !== f ? 0 : element))

);

let myResult = modifiedMatrix.map(row => row.join('\t')).join('\n');

"Исходная матрица:\n" + initialResult + "\n\nМатрица после замены:\n" + myResult;

} else {

"Пожалуйста, введите корректные положительные числа.";

}

**Расширенное задание:**

var canvas = document.getElementById("drawingCanvas");

var context = canvas.getContext("2d");

const blue = "#0074D9";

const width = parseInt(prompt("Введите ширину канваса:"));

const height = width / 2;

canvas.width = width;

canvas.height = height;

context.clearRect(0, 0, width, height);

context.fillStyle = blue;

const rectSize = Math.min(width / 20, height / 10);

const stepX = width / 11;

const stepY = height / 6;

for (let i = 1; i <= 5; i++) {

for (let j = 1; j <= 10; j++) {

if (i === 1 || i === 5 || j === 1 || j === 10) {

context.fillRect(j \* stepX - stepX/2, i \* stepY - stepY/2, rectSize, rectSize);

}

}

}

const mainRectX = stepX - stepX/2;

const mainRectY = stepY - stepY/2;

const mainRectWidth = 9 \* stepX;

const mainRectHeight = 4 \* stepY;

const centerX = mainRectX + mainRectWidth / 2;

const centerY = mainRectY + mainRectHeight / 2 - 20;

context.fillRect(centerX - 40, centerY, 60, 40);

context.fillRect(centerX + 40, centerY, 60, 40);

# 7. Выводы

В процессе выполнения лабораторной работы исследовалась работа с языком JavaScript, включая создание и обработку массивов, а также применение метода canvas для визуализации данных. Была успешно создана программа, генерирующая матрицу, элементы которой определяются как произведение номера строки на номер столбца, с последующим выводом результата на страницу. Для реализации расширенного задания использовался объект canvas, с помощью которого отрисовывалась заданная фигура, что позволило сформировать паттерн на основе данных, введённых пользователем. В ходе работы применялись методы массивов, такие как Array.from(), map() и join(), а также стандартные методы для обработки введённых данных и взаимодействия с DOM.