ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Соловьева Н.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3  «Основы JavaScript» |
| **по дисциплине: Web-Технологии** |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2024

**Цель работы**: знакомство с языком javascript

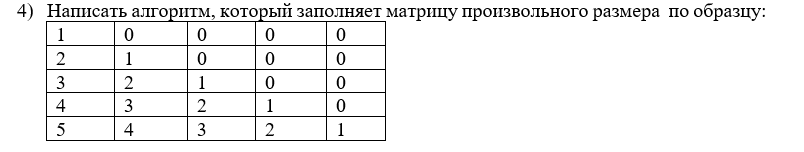
**Базовое задание**

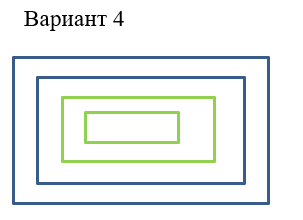
Используя редактор javascript (см. листинг 1) написать программу, которая выполняет задание по варианту (базовая часть) и использует массив. Вывести на экран матрицу в форме прямоугольника. Предусмотреть возможность задавать произвольный размер матрицы через диалог с пользователем.

**Расширенное задание**

Нарисовать заданную вариантом фигуру, используя объект canvas и образец скрипта из листинга 2. Повторяющие фрагменты формировать с помощью циклов. Предусмотреть возможность задавать произвольный размер фигуры через диалог с пользователем.

Вариант 4





Ход выполнения базового задания

Описание переменных программы:

- canvas: переменная типа HTMLCanvasElement, полученная с помощью метода getElementById. Она представляет собой элемент canvas с идентификатором "drawingCanvas".

- context: переменная типа CanvasRenderingContext2D, полученная с помощью свойства getContext объекта canvas. Она представляет собой контекст 2D для рисования на canvas.

- size: переменная типа Number, инициализированная значением 200. Она представляет собой размер каждого прямоугольника, который будет нарисован.

- ratio: переменная типа Number, инициализированная значением 16/9. Она представляет собой соотношение сторон каждого прямоугольника.

- start\_x: переменная типа Number, инициализированная значением 100. Она представляет собой начальную точку x для рисования каждого прямоугольника.

- start\_y: переменная типа Number, инициализированная значением 100. Она представляет собой начальную точку y для рисования каждого прямоугольника.

- i: переменная типа Number, инициализированная значением 0. Она представляет собой счетчик, который увеличивается на 1 при каждом проходе цикла for.

В цикле for выполняются следующие действия:

1. Если значение i больше или равно 2, то цвет линии устанавливается в "green". В противном случае, цвет линии устанавливается в "blue".

2. Вызывается метод beginPath контекста 2D, чтобы начать новый путь.

3. Вызывается метод moveTo контекста 2D, чтобы переместить курсор рисования в точку (start\_x, start\_y).

4. Вызывается метод lineTo контекста 2D, чтобы нарисовать линию от текущей точки до точки (start\_x + size, start\_y).

5. Вызывается метод lineTo контекста 2D, чтобы нарисовать линию от текущей точки до точки (start\_x + size, start\_y + size / ratio).

6. Вызывается метод lineTo контекста 2D, чтобы нарисовать линию от текущей точки до точки (start\_x, start\_y + size / ratio).

7. Вызывается метод lineTo контекста 2D, чтобы нарисовать линию от текущей точки до точки (start\_x, start\_y).

8. Вызывается метод stroke контекста 2D, чтобы нарисовать путь, созданный с помощью метода beginPath.

9. Значение start\_x увеличивается на (size \* (1 - 0.75)) / 2. Это позволяет переместить начальную точку x для следующего прямоугольника.

10. Значение start\_y увеличивается на (size / ratio \* (1 - 0.75)) / 2. Это позволяет переместить начальную точку y для следующего прямоугольника.

11. Значение size уменьшается в 0.75 раза. Это позволяет уменьшить размер следующего прямоугольника.

12. Значение i увеличивается на 1.

Цикл for повторяется 4 раза, что позволяет нарисовать 4 прямоугольника.

function fillMatrix(n) {

let matrix = [];

for (let i = 0; i < n; i++) {

matrix[i] = new Array(n).fill(0);

}

for (let i = 0; i < n; i++) {

for (let j = 0; j <= i; j++) {

matrix[i][j] = i - j;

}

}

return matrix;

}

let size = Number(document.getElementById('size').value);

const resultMatrix = fillMatrix(size);

result = '';

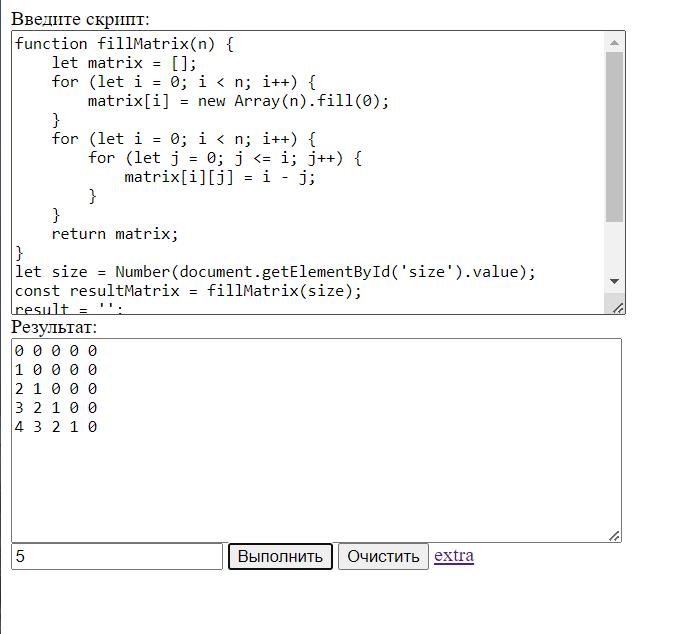
for (let i = 0; i < size; i++) {

result += resultMatrix[i].join(' ')+'\n';

}

result;

**Результат работы**



Выполнение расширенного задания

**Скрипт для расширенного задания**

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

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

let size = Number(document.getElementById('psize').value);;

let ratio = 16/9;

let start\_x = 100;

let start\_y = 100;

for (let i = 0; i < Number(document.getElementById('num').value); i++) {

if (i >= 2) {

context.strokeStyle = "green";

} else {

context.strokeStyle = "blue";

}

context.beginPath();

context.moveTo(start\_x ,start\_y);

context.lineTo(start\_x + size,start\_y);

context.lineTo(start\_x + size,start\_y + size / ratio);

context.lineTo(start\_x,start\_y + size / ratio);

context.lineTo(start\_x,start\_y);

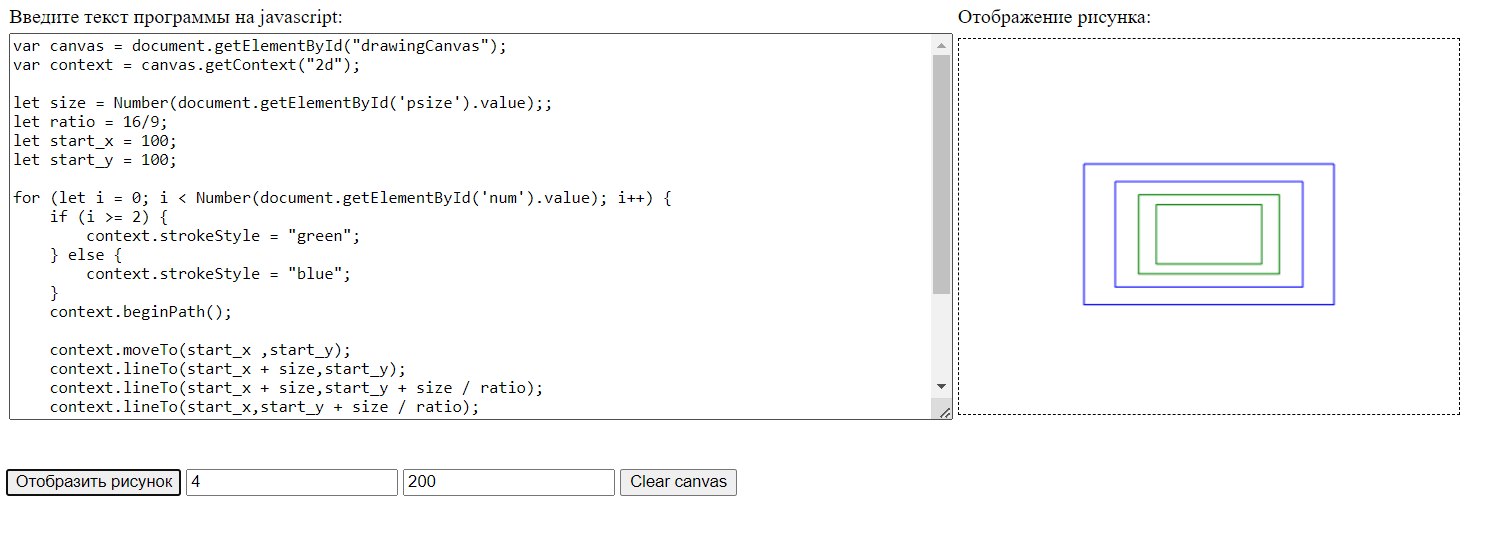
context.stroke();

start\_x += (size \* (1 - 0.75)) / 2;

start\_y += (size / ratio \* (1 - 0.75)) / 2;

size \*= 0.75;

} **Результат работы программы**



Анализ программы и результатов работы

Программа работает корректно и не вызывает трудностей в прочтении