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  Язык HTML. Приемы верстки |
| **по дисциплине: Web-Технологии** |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134К |  |  |  | Самарин Д. В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Оглавление

[Цель работы: 3](#_Toc165917864)

[Вариант задания № 16: 3](#_Toc165917865)

[Названия средств, использованных при выполнении работы: 3](#_Toc165917866)

[Описание выполненных пунктов базового задания: 3](#_Toc165917867)

[Таблица с описанием переменных программ 4](#_Toc165917868)

[Скриншоты web-страниц с подписью и номерами рисунков 5](#_Toc165917869)

[Рисунок 1 (базовое задание) 5](#_Toc165917870)

[Рисунок 2(Базовое задание) 5](#_Toc165917871)

[Рисунок 3 (Расширенное задание) 6](#_Toc165917872)

[Рисунок 4 (Расширенное задание) 6](#_Toc165917873)

[Листинг лабораторной работы: 7](#_Toc165917874)

[Первая страница HTML 4: 7](#_Toc165917875)

[Вторая страница HTML 5: 8](#_Toc165917876)

Цель работы: Знакомство с языком javascript

## Вариант задания № 16:

|  |  |  |
| --- | --- | --- |
| № | Базовое задание | Расширенное задание |
| 16 | Определить номер первого столбца матрицы, в котором есть хотя бы один отрицательный элемент. |  |

## Названия средств, использованных при выполнении работы:

Редактор кода VSCode;

Браузер Chrome;

## Описание выполненных пунктов базового задания:

1) Первая страница (Базовое задание):

* [Рисунок 1;](#_Рисунок_1(HTML_4))
* [Рисунок 2;](#_Рисунок_2(HTML_4))
* [Cсылка на листинг;](#_Первая_страница_HTML)

2) Вторая страница (Расширенное задание):

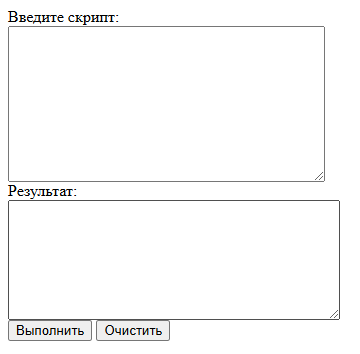
* [Рисунок 3;](#_Рисунок_3_(HTML)
* [Рисунок 4;](#_Рисунок_4_(HTML)
* [Ссылка на листинг;](#_Вторая_страница_HTML)

## Таблица с описанием переменных программ

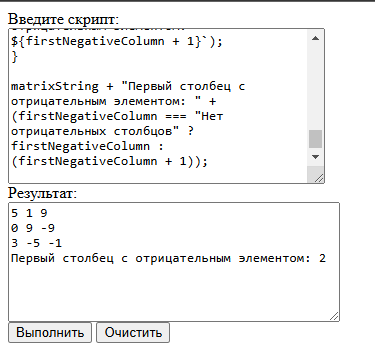
|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Назначение** |
| rows | Числовой | Количество строк в матрице |
| columns | Числовой | Количество столбцов в матрице |
| ans | Число/строка | Результат поиска первого отрицательного числа в строке или сообщение "нет отрицательных чисел" |
| n | Число | Константа, используемая для генерации случайных чисел |
| matrix | Массив | Матрица, содержащая случайные числа |
| i | Число | Переменная для индексации строк матрицы |
| j | Число | Переменная для индексации столбцов матрицы |
| **Имя переменной** | **Тип** | **Назначение** |
| canvas | Объект | Ссылка на элемент холста в HTML, полученный с помощью метода **getElementById** |
| context | Объект | Контекст рисования 2D холста, полученный с помощью метода **getContext** |
| margin | Число | Расстояние между квадратами и отступ от края холста |
| squareSize | Число | Размер стороны квадрата |
| rhombusHeight | Число | Высота ромба в квадратах, введенная пользователем |
| color | Строка | Цвет квадратов, который изменяется в зависимости от столбца ромба (синий или зеленый) |
| counts | Массив | Массив, содержащий количество квадратов для каждого столбца ромба |
| column | Число | Индекс текущего столбца ромба |
| offsetY | Число | Вертикальное смещение от верхнего края холста для каждого столбца ромба |
| x | Число | Горизонтальная позиция начала текущего столбца ромба на холсте |
| y | Число | Вертикальная позиция начала текущего квадрата в текущем столбце ромба на холсте |
| rectHeight | Число | Высота прямоугольника, заполняющего нечетные столбцы ромба (равна высоте столбца ромба) |

В ходе выполнения лабораторной работы были реализованы WEB страница для расширенного и базового заданий.

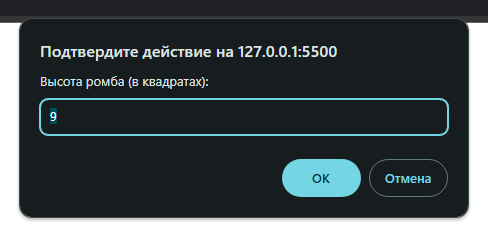
## Скриншоты web-страниц с подписью и номерами рисунков



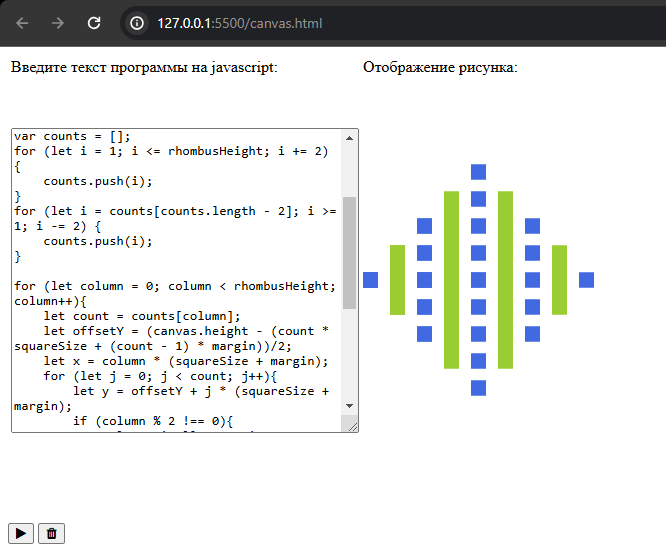
## Рисунок 1 (базовое задание)



## Рисунок 2(Базовое задание)



## Рисунок 3 (Расширенное задание)



## Рисунок 4 (Расширенное задание)

# Листинг лабораторной работы:

## Первая страница:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Редактор JavaScript</title>  <link rel="stylesheet" href="jseditstyle.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">  <meta charset="UTF-8">  <meta name="description" content="Сайт о процессорах">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  </head>  <body>  <div id="content">  Введите количество строк матрицы:<br>  <input type="number" id="rows"><br>  Введите количество столбцов матрицы:<br>  <input type="number" id="columns"><br>  <br>  <button onclick="generateMatrix()">Сгенерировать матрицу</button><br><br>  Результат:<br>  <textarea id="myrezult" rows=3 cols=40></textarea><br>  <div class="controls">  <button id="play" onclick="findFirstNegativeColumn()">Поиск первого отрицательного столбца</button>  <span></span>  <button id="clear" onclick="clearResult()">Очистить результат</button>  </div>  </div>  <script>  function generateMatrix() {  const rows = parseInt(document.getElementById('rows').value);  const columns = parseInt(document.getElementById('columns').value);  const matrix = [];  for (let i = 0; i < rows; i++) {  const row = [];  for (let j = 0; j < columns; j++) {  row.push(Math.floor(Math.random() \* 21) - 10); // генерация случайного числа от -10 до 10  }  matrix.push(row);  }  document.getElementById('myrezult').value = JSON.stringify(matrix);  }  function findFirstNegativeColumn() {  const matrix = JSON.parse(document.getElementById('myrezult').value);  let found = false;  for (let j = 0; j < matrix[0].length; j++) {  for (let i = 0; i < matrix.length; i++) {  if (matrix[i][j] < 0) {  found = true;  document.getElementById('myrezult').value = `Первый отрицательный столбец: ${j + 1}`;  return;  }  }  }  if (!found) {  document.getElementById('myrezult').value = "Отрицательных элементов нет";  }  }  function clearResult() {  document.getElementById('myrezult').value = '';  }  </script>  </body>  </html> |

## Вторая страница:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <link rel="stylesheet" href="canvasstyle.css">  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">  <title>Canvas рисование</title>  <script>  function clear\_holst(){  var canvas = document.getElementById("drawingCanvas");  var context = canvas.getContext("2d");  context.clearRect(0,0,canvas.width,canvas.height);  }  </script>  </head>  <body>  <div id="content">  <table border = 0>  <tr>  <td> Введите текст программы на javascript:</td>  <td>Отображение рисунка: </td>  </tr>  <tr>  <td><textarea id="mycode" rows=20 cols=50></textarea></td>  <td><canvas id="drawingCanvas" width="400" height="400">  Ваш браузер не поддерживает canvas  </canvas>  </td>  </tr>  </table>  <br><br>  <div class="controls">  <button id="play" onclick= "eval(document.getElementById('mycode').value)"><i class="fa fa-play"></i></button>  <span></span>  <button id="clear" onclick="clear\_holst()">  <i class="fa fa-trash"></i>  </button>  </div>  </div>  </body>  </html> |

|  |
| --- |
| var canvas = document.getElementById("drawingCanvas");  var context = canvas.getContext("2d");  let margin = 12;  const squareSize = 15;  let rhombusHeight = parseInt(prompt("Высота ромба (в квадратах): ", "9"));  let color;  var counts = [];  for (let i = 1; i <= rhombusHeight; i += 2) {  counts.push(i);  }  for (let i = counts[counts.length - 2]; i >= 1; i -= 2) {  counts.push(i);  }  for (let column = 0; column < rhombusHeight; column++){  let count = counts[column];  let offsetY = (canvas.height - (count \* squareSize + (count - 1) \* margin))/2;  let x = column \* (squareSize + margin);  for (let j = 0; j < count; j++){  let y = offsetY + j \* (squareSize + margin);  if (column % 2 !== 0){  color = 'yellowgreen';  let rectHeight = count \* squareSize + (count - 1) \* margin;  context.fillStyle = color;  context.fillRect(x, y, squareSize, rectHeight);  break;  }  else{  color = 'royalblue';  context.fillStyle = color;  context.fillRect(x, y, squareSize, squareSize);  }  }  } |

Вывод: выполнив лабораторную работаю №3, я познакомился ближе с javascript.