Содержание

1 Текст задания

Задача 10

Из листа клетчатой бумаги размером M*N клеток удалили некоторые клетки. На сколько кусков распадется оставшаяся часть листа?

Пример. Если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.

Ввод входных параметров осуществляется с помощью клавиатуры и мыши.

2 Исходные тексты программы

index.html

```
<!DOCTYPE html>
<html lang="ru">
    <head>
        <title>Лабораторная работа №1 - Удаление клеток</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <script type="text/javascript" src="lib/jquery-2.2.4.min.js"></script>
        <script type="text/javascript" src="js/remsquare.js"></script>
    </head>
    <body>
        <div class="wrapper">
            <div id="condition">
                <h2>Условие задачи</h2>
                <p>Из листа клетчатой бумаги размером M*N клеток удалили некоторые клетки. На сколько кусков распадется оставшаяся часть листа?</p>
                <p>Пример: Если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.</p>
                <p>Примечание: Две клетки не распадаются, если они имеют общую сторону.</p>
            </div>
            <div id="input">
                <h2>Входные параметры</h2>
                <p><label>Число строк (M): <input id="row" type="number"></label></p>
                <p><label>Число столбцов (N): <input id="col" type="number"></label></p>
                <div id="grid">
                    <p>Введите число строк и столбцов в соответствующие текстовые поля.</p>
                </div>
            </div>
```

```
                <div id="output">
                    <h2>Выходные параметры</h2>
                    <p id="answer">Входные параметры не введены...</p>
                </div>
            </div>
        </body>
    </html>
```

style.css

```
    @font-face
    {
      font-family:Roboto;
      src:url(fonts/roboto/Roboto.woff2) format(woff2), url(fonts/roboto/Roboto.woff) format(woff), url(fonts/roboto/Roboto.ttf) format(truetype);
      font-weight:400;
      font-style:normal;
    }

    *
    {
      margin:0;
      padding:0;
    }

    html
    {
      font-size:62.5%;
    }

    body
    {
      font-family:Roboto;
    }

    .wrapper
    {
```

```
  max-width:800px;
  width:calc(100%-20px);
  border-left:1px solid #39c;
  border-right:1px solid #39c;
  margin:0 auto;
}

h2
{
  font-size:2em;
  background-color:#39c;
  color:#fff;
  text-align:center;
  padding:5px 8px;
}

p
{
  font-size:1.6em;
  margin:5px;
}

input
{
  font-size:1em;
}

label
{
  display:block;
  width:100%;
  text-align:center;
}

label > input
{
  width:100%;
  box-sizing:border-box;
```

```
  text-align:center;
  outline:0;
  padding:4px 2px;
}

#condition p
{
  text-align:justify;
}

#output
{
  border-bottom:1px solid #39c;
}

#grid
{
  display:flex;
  flex-wrap:wrap;
  margin:15px auto;
  padding:5px;
}

.cell
{
  border:1px solid #000;
  box-sizing:border-box;
  cursor:pointer;
  background-color:#ddf7ff;
  transition:background-color .1s linear;
}

.cell:hover
{
  background-color:#c9e3ff;
}

.cell.selected
```

```
{
  background-color:#4c8efb;
}

.cell.selected:hover
{
  background-color:#318efd;
}
```

remsquare.js

```
/**
 * Матрица посещений
 * Состояния ячеек сетки в виде двумерного массива
 */
var statesCells = [];

/**
 * Пары узлов
 */
var pair = [];

/**
 * Получить количество строк
 */
function getRow()
{
    return jQuery('#row').val();
}

/**
 * Получить количество столбцов
 */
function getCol()
{
    return jQuery('#col').val();
}
```

```
/**
 * Проверка значения на int
 * @param {int} val
 *   Значение для проверки
 */
function isInt(val)
{
    var result = false;
    if (Math.floor(val).toString() === val && jQuery.isNumeric(val))
    {
        result = true;
    }
    return result;
}

/**
 * Является ли число положительным
 * @param {int} val
 *   Значение для проверки
 */
function numPositive(val)
{
    var result = false;
    if (val >= 0)
    {
        result = true;
    }
    return result;
}

/**
 * Конвертировать одномерный массив в двумерный
 * @param {array} vector
 *   Одномерный массив
 * @param {int} col
 *   Количество ячеек в строке
 */
function convMatrix(vector, col)
```

```
    {
        var matrix = [];
        while (vector.length)
        {
            matrix.push(vector.splice(0, col));
        }
        return matrix;
    }

    /**
     * Построение сетки
     * @param {int} row
     *   Число строк
     * @param {int} col
     *   Число столбцов
     */
    function buildGrid(row, col)
    {
        // Общее количество ячеек
        var totalCells = row * col;
        // Разметка ячеек
        var htmlCells = '';
        // Каскадные стили
        var styles = '.cell {width: calc(100% / ' + col + '); padding-bottom:
calc(100% / ' + col + ');}';
        for (var i = 0; i < totalCells; i++)
        {
            htmlCells += '<div class="cell selected"></div>';
        }

        jQuery('head style').remove();
        jQuery('<style>' + styles + '</style>').appendTo('head');
        jQuery('#grid').html(htmlCells);
    }

    /**
     * Алгоритм поиска в ширину
     * Breadth-First Search (BFS)
```

```
 * @param {int} x
 *   Координата текущей строки
 * @param {int} y
 *   Координата текущего столбца
 * @param {int} length
 *   Всего строк в массиве
 * @param {int} height
 *   Всего столбцов в массиве
 */
function dfs(x, y, length, height)
{

    if (x === length || x === -1 || y === height || y === -1)
    {
        return;
    }
    else if (statesCells[x][y] === -1)
    {
        // Если ячейка пустая
        return;
    }

    // Поиск одинаковых координат
    for (var i = 0; i < pair.length; i++)
    {
        if (pair[i][0] === x && pair[i][1] === y)
        {
            return;
        }
    }

    pair.push([x, y]); // Сохранить текущее местоположение

    dfs(x + 1, y, length, height); // Проверить ячейку слева
    dfs(x - 1, y, length, height); // Проверить ячейку справа
    dfs(x, y + 1, length, height); // Проверить ячейку сверху
    dfs(x, y - 1, length, height); // Проверить ячейку снизу
```

```
}

/**
 * Сохранить координаты в матрице истории посещений
 * @param {int} value
 *   Значение, которое надо сохранить
 */
function saveStateCell(value)
{
    var n = pair.length;
    for (var i = 0; i < n; i++)
    {
        statesCells[pair[i][0]][pair[i][1]] = value;
    }
}

/**
 * Считывание состояния ячеек из сетки
 * @param {int} col
 *   Число колонок
 */
function readGrid(col)
{
    col = parseInt(col);
    var states = []; // Состояния ячеек сетки
    var pieces = 0; // Количество найденных кусков
    var quantity = jQuery('#grid .cell').length;
    if (quantity > 0)
    {
        // Перебор ячеек сетки
        jQuery('#grid .cell').each(function()
        {
            if (jQuery(this).hasClass('selected'))
            {
                states.push(0); // Занятая ячейка
            }
            else
            {
```

```
                states.push(-1); // Пустая ячейка
            }
        });
    }

    // Одномерные массивы конвертировать в двумерные
    statesCells = convMatrix(states, col);

    // Количество строк в матрице
    var m = statesCells.length;

    // Проход по матрице
    for (var i = 0; i < m; i++)
    {
        var row = statesCells[i]; // Элементы строки
        var n = row.length; // Количество столбцов в матрице
        for (var j = 0; j < n; j++)
        {
            if (row[j] === 0)
            { // Ячейка не пустая и не посещенная
                pieces += 1; // Количество кусков
                pair = [];
                dfs(i, j, m, n); // Обработка данных поиском в ширину
                saveStateCell(pieces); // Сохранить координаты
            }
        }
    }

    // Вывести ответ на экран
    jQuery('#answer').html('Количество кусков: <b>' + pieces + '</b>');
}


/**
 * Выполняем действия когда DOM полностью загружен
 */
jQuery(document).ready(function()
{
```

```
            // Отслеживаем изменение числа строк и числа столбцов
            jQuery('#row, #col').on('keyup', function()
            {
                var row = getRow(); // Количество строк на сетке
                var col = getCol(); // Количество столбцов на сетке
                var message = ''; // Сообщение пользователю

                if (row === '' || col === '')
                { // Значения не пустые
                    message = 'Введите число строк и столбцов в соответствующие
текстовые поля.';
                }
                else if (!isInt(row) || !isInt(col))
                { // Являются Int
                    message = 'Входные параметры не являются целочисленными.';
                }
                else if (!numPositive(row) || !numPositive(col))
                { // Положительные числа
                    message = '<b>Ошибка! </b>Входные параметры должны иметь
положительные значения!';
                }
                else if (row > 150 || col > 150)
                {
                    message = '<b>Ошибка! </b>Введите значения текстовых полей
<= 150.';
                }

                if (message === '')
                {
                    buildGrid(row, col); // Построение сетки
                    readGrid(col); // Считывание состояния ячеек из сетки
                }
                else
                {
                    jQuery('#grid').html('<p>' + message + '</p>');
                }
            });
            // Отслеживаем клик по ячейке
```

```
    jQuery('#grid').on('click', '.cell', function()
    {
        var col = getCol(); // Количество столбцов на сетке
        jQuery(this).toggleClass('selected'); // Добавить/удалить класс
        readGrid(col); // Считывание состояния ячеек из сетки
    });
});
```

3 Скриншоты выполнения программы

Рисунок 1 – Ввод входных параметров с клавиатуры

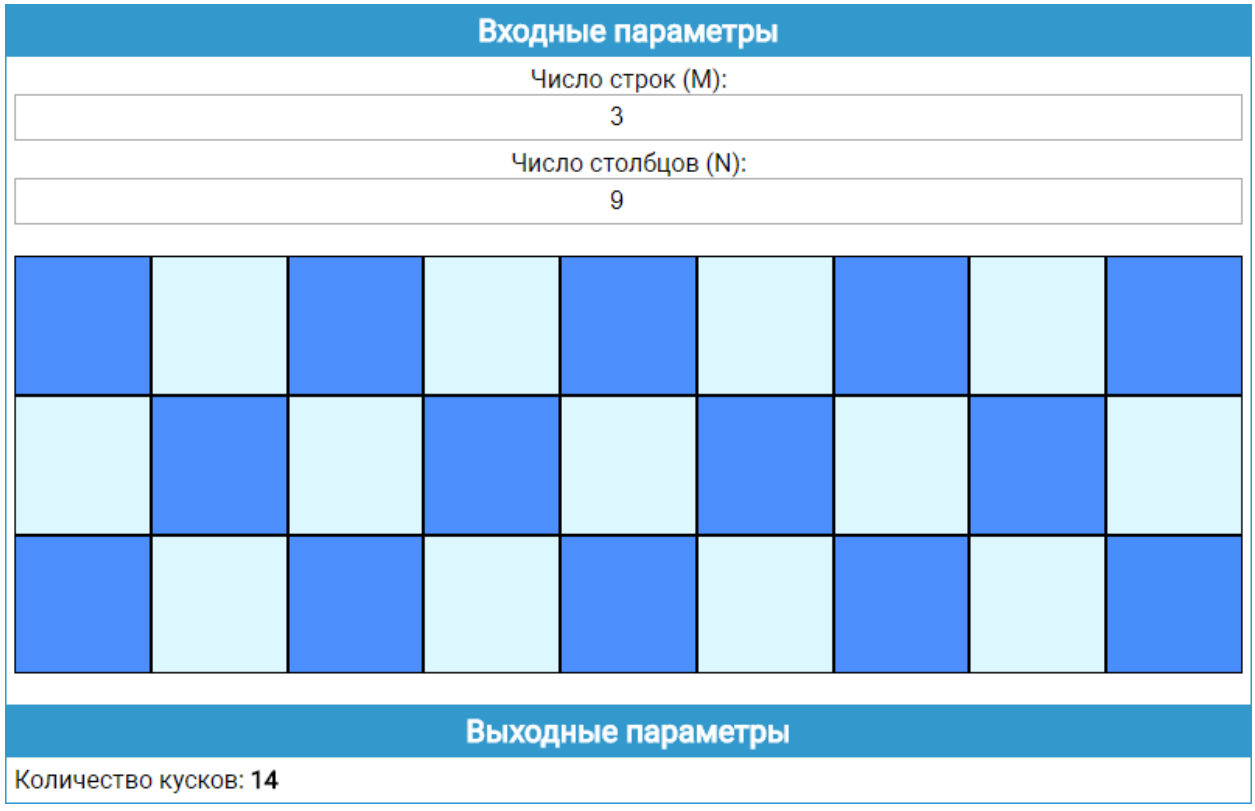

Рисунок 2 – Результат выполнения программы

4 Заключение

В ходе выполнения лабораторной работы изучены принципы работы со списками, стеками и очередями.

5 Используемая литература

1 Методические указания по выполнению лабораторных работ по дисциплине «Технологии программирования» – Нижний Новгород.: НГТУ, 2015. – 18 с.

2 Стандарт предприятия СТП 1-У-НГТУ-2004. Общие требования к оформлению пояснительных записок дипломных и курсовых проектов. – Нижний Новгород.: НГТУ, 2004. – 22 с.