МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

Отчёт

о лабораторной работе №3

по дисциплине «Веб-технологии»

Тема: «Базовый JS веб-приложений.»

Выполнил студент 2 курса

группы ПО-11 Сымоник И.А.

Проверил: Михняев А.Л.

**Цель работы**: Изучить базовый JS веб-приложений.

**Ход работы**

**Задание 1:**

function compareNumbers() {

let firstNumber;

let secondNumber;

while (true) {

firstNumber = prompt("Введите первое число:");

if (firstNumber.toLowerCase() === 'q')

break;

secondNumber = prompt("Введите второе число:");

let message;

if (secondNumber.toLowerCase() === 'q')

break;

if (isNaN(firstNumber) || isNaN(secondNumber)) {

message = "Один или оба ввода – не числа.";

alert(message);

break;

}

const firstNumberInt = parseInt(firstNumber);

const secondNumberInt = parseInt(secondNumber);

if (firstNumberInt === secondNumberInt) {

message = "числа равны";

} else if (firstNumberInt < secondNumberInt) {

message = "первое число меньше";

} else {

message = "второе число меньше";

}

alert(message);

}

}

**Задание 2:**

function getFloorAndEntrance(floors, entrances, flatsPerFloor, flat) {

if (typeof floors !== 'number' || floors < 1 || floors > 20) {

throw new Error('Количество этажей должно быть числом от 1 до 20.');

}

if (typeof entrances !== 'number' || entrances < 1 || entrances > 15) {

throw new Error('Число подъездов должно быть числом от 1 до 15.');

}

if (typeof flatsPerFloor !== 'number' || flatsPerFloor < 1 || flatsPerFloor > 5) {

throw new Error('Количество квартир на лестничной площадке должно быть числом от 1 до 5.');

}

if (typeof flat !== 'number' || flat < 1) {

throw new Error('Номер квартиры должен быть числом больше 0.');

}

const entrance = Math.ceil(flat / (flatsPerFloor \* floors));

if (entrance < 1 || entrance > entrances)

throw new Error('Квартиры не существует');

return entrance;

}

function task2() {

try {

const floors = Number(document.getElementById('floors').value);

const entrances = Number(document.getElementById('entrances').value);

const flatsPerFloor = Number(document.getElementById('flatsPerFloor').value);

const flat = Number(document.getElementById('flat').value);

const entrance = getFloorAndEntrance(floors, entrances, flatsPerFloor, flat);

alert(`Номер подъезда: ${entrance}`);

} catch (error) {

alert(error.message);

}

}

**Задание 3:**

function getWeekDay(month, day) {

const daysInMonths = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

if (month < 1 || month > 12) {

throw new Error('Номер месяца должен быть числом от 1 до 12.');

}

if (day < 1 || day > daysInMonths[month - 1]) {

throw new Error(`В ${month}-м месяце нет ${day}-го дня.`);

}

const date = new Date(2022, month - 1, day);

const weekDay = date.toLocaleString('ru', { weekday: 'long' });

return weekDay;

}

function task3() {

try {

const month = Number(document.getElementById('month').value);

const day = Number(document.getElementById('day').value);

const weekDay = getWeekDay(month, day);

alert(`День недели: ${weekDay}`);

} catch (error) {

alert(error.message);

}

}

**Задание 4:**

function calculate(num1, num2) {

const isNum1Even = num1 % 2 === 0;

const isNum2Even = num2 % 2 === 0;

if (isNum1Even && isNum2Even) {

return num1 \* num2;

}

else if (!isNum1Even && !isNum2Even) {

return num1 + num2;

}

else {

if (isNum1Even === true) {

return num2;

}

else {

return num1;

}

}

}

function task4() {

try {

const num1 = Number(document.getElementById('num1').value);

const num2 = Number(document.getElementById('num2').value);

const result = calculate(num1, num2);

alert(`Результат: ${result}`);

} catch (error) {

alert(error.message);

}

}

**Задание 5:**

function sum(a) {

let currentSum = a;

function f(b) {

currentSum += b;

return f;

}

f.toString = function () {

return currentSum;

};

return f;

}

const sum1 = sum(0);

function task5() {

const sumResult = document.getElementById('sum-result');

const inputElement = document.getElementById("sum");

const res = sum1(Number(inputElement.value));

sumResult.innerHTML = res;

}

**Задание 6:**

function range(start, end, step = 1) {

const rangeArray = [];

if (step > 0) {

for (let i = start; i <= end; i += step) {

rangeArray.push(i);

}

} else {

for (let i = start; i >= end; i += step) {

rangeArray.push(i);

}

}

return rangeArray;

}

function task6() {

const start = Number(document.getElementById('start').value);

const end = Number(document.getElementById('end').value);

const step = Number(document.getElementById('step').value);

const result = range(start, end, step);

alert(result);

}

**Задание 7:**

function createMatrix(rows, cols) {

const matrix = [];

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

matrix.push([]);

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

matrix[i].push(Math.floor(Math.random() \* 101));

}

}

return matrix;

}

function printMatrix(matrix) {

for (let i = 0; i < matrix.length; i++) {

console.log(matrix[i].join(' '));

}

}

function sumMatrices(matrix1, matrix2) {

const sumMatrix = [];

for (let i = 0; i < matrix1.length; i++) {

sumMatrix.push([]);

for (let j = 0; j < matrix1[i].length; j++) {

sumMatrix[i].push(matrix1[i][j] + matrix2[i][j]);

}

}

return sumMatrix;

}

function formString(matrix, rows) {

let string = "";

for (let i = 0; i < matrix.length; i++) {

if (i % rows)

string += '\n'

string += matrix[i]

}

return string;

}

function task7() {

const rows = Number(document.getElementById('rows').value);

const columns = Number(document.getElementById('columns').value);

const matrix1 = createMatrix(rows, columns);

printMatrix(matrix1);

alert(formString(matrix1, rows));

const matrix2 = createMatrix(rows, columns);

alert(formString(matrix2, rows));

const sumMatrix = sumMatrices(matrix1, matrix2);

alert(formString(sumMatrix, rows));

}

**Задание 8:**

function parse\_8() {

const inputElement = document.getElementById("arrays");

const res = JSON.parse(inputElement.value);

console.log(res);

return res;

}

function myUnionFunc(arrays) {

let newArr = new Array();

arrays.forEach(element => {

element.forEach((item) => {

newArr.push(item);

})

});

if (newArr.some(Array.isArray)) {

throw ('Фунция работает только с одномерными массивами. Размер массива ' + newArr.flat(Infinity).length);

return;

}

return [...new Set(newArr)];

}

function task8() {

try {

alert(myUnionFunc(parse\_8()));

} catch (error) {

alert('Неверный формат ввода');

alert(error);

}

}

**Задание 9:**

function parse() {

const inputElement = document.getElementById("arrays\_2");

const res = JSON.parse(inputElement.value);

console.log(res);

return res;

}

const MyArrayFlattenFunc = (arr) => {

let result = [];

arr.forEach((element) => {

if (Array.isArray(element)) {

result = result.concat(MyArrayFlattenFunc(element));

} else {

result.push(element);

}

});

result.sort();

return result;

};

function task9(arr) {

try {

alert(MyArrayFlattenFunc(parse()));

}

catch (error) {

alert('Неверный ввод');

}

}

**Задание 10:**

function MyFunctionUnique(input, repeats) {

const map = new Map();

const uniqueValues = [];

for (let i = 0; i < input.length; i++) {

map.set(input[i], (map.get(input[i]) ?? 0) + 1);

}

for (let [key, value] of map.entries()) {

console.log(key + " = " + value)

if (value < repeats)

uniqueValues.push(key);

}

uniqueValues.sort();

return uniqueValues;

}

function task10() {

try {

const str = document.getElementById("str").value;

const repeats = document.getElementById("repeats").value

alert(MyFunctionUnique(str, repeats));

}

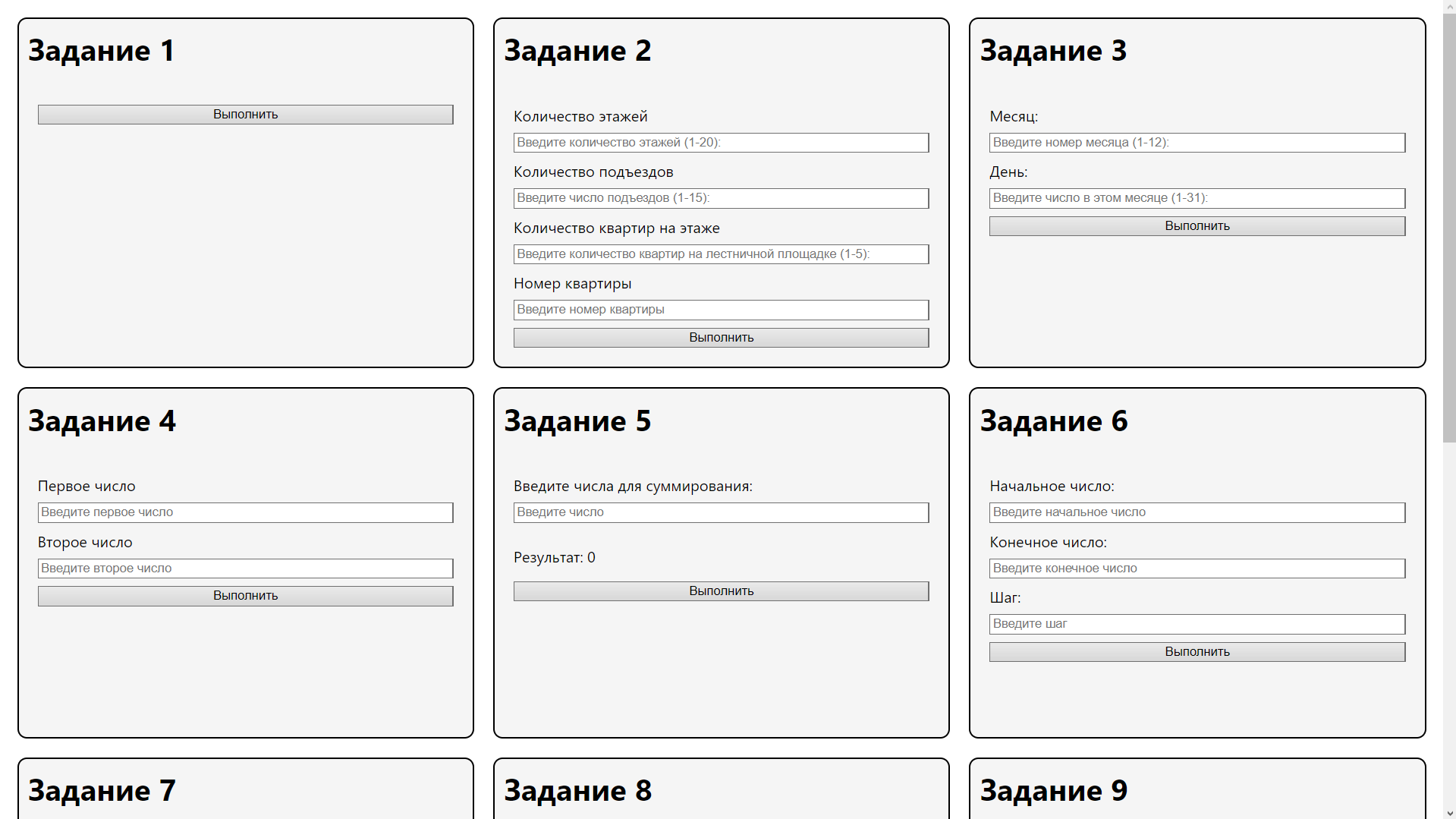
catch (error) {

alert(error)

}

}

**Сверстанная страница:**

****

Изображение выглядит как текст, снимок экрана, диаграмма, программное обеспечение

Автоматически созданное описание

**Вывод:** изучили базовый JS для web приложений.