**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: «Современные технологии создания Web-систем»

на тему: **«**Объектная природа языка Javascript**»**

Выполнил: студент гр. ИТИ-11

Манько С.Д.

Принял: ассистент

Карась О.В.

Гомель 2025

**Цель работы:** выполнить практические задания

**Задание**

1. Практические задания
   1. Сгенерировать строку из случайных латинских букв. Строку разбить на символы, каждый символ перевести в верхний регистр, собрать массив из кодов этих символов, оставить только те коды, которые не делятся на 5 и посчитать сумму остатков деления этих кодов на 5. (Указание – использовать методы обхода массивов).
   2. Сгенерировать массив из 1000 случайных чисел в диапазоне s ± p%, отсортировать его по убыванию остатков от деления на 10.
   3. Создать функцию, которая вернет: a. Среднее арифметическое своих аргументов b. Среднее гармоническое своих аргументов
   4. Создать массив из списка имен. Сгенерировать массив из 20 объектов типа «Человек». У каждого объекта должно быть случайным образом выбрано имя, возраст, а также метод, который выводит имя и возраст данного человека. Прогнать методом обхода по этому массиву и заставить каждого человека представиться.

При разработке скриптов использовать объекты. У созданных объектов инициализировать конструкторы и дополнительные функции.

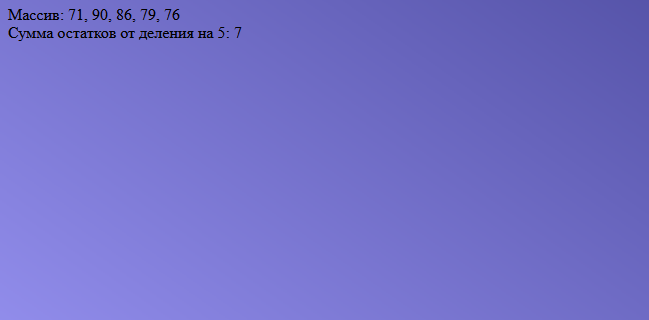


Рисунок 1 – результат выполнения задания 1.1

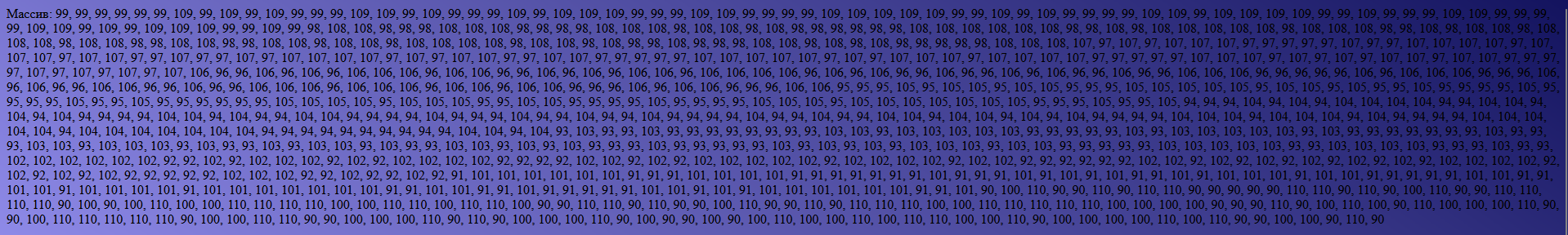


Рисунок 2 – результат выполнения задания 1.2

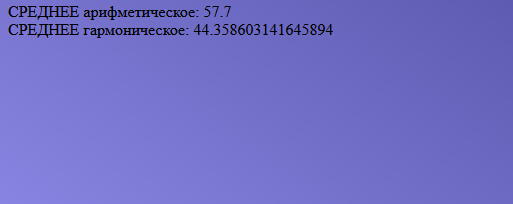


Рисунок 3 – результат выполнения задания 1.3

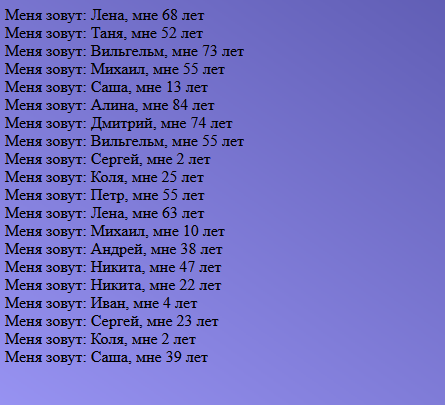


Рисунок 4 – результат выполнения задания 1.4

**Вывод:** в результате лабораторной работы были выполнены практические задания

Приложение A

(Обязательно)

Файл 1.html

<!DOCTYPE html>

<html>

<head>

<title>MSD</title>

<link rel="stylesheet" href="1.css">

</head>

<body>

<div class="Output">

</div>

<div id="resultContainer"></div>

<script src="1.js">

</script>

</body>

</html>

Файл 1.css

body

{

height: 100vh;

background: linear-gradient(35deg, rgb(161, 156, 253) 20%, rgb(19, 19, 94) 100%);

}

.Output

{

display: flex;

justify-content: center;

align-items: center;

}

Файл1.js

function randomString(length) {

let result = '';

const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';

const charactersLength = characters.length;

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

result += characters.charAt(Math.floor(Math.random() \* charactersLength));

}

return result;

}

const result = randomString(5);

let arr = [];

let sum = 0;

console.log(result);

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

arr.push(result[i].toUpperCase().charCodeAt());

}

console.log(arr);

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

if (arr[i] % 5 != 0) {

sum += arr[i] % 5;

}

}

const resultContainer = document.getElementById('resultContainer');

const arrayText = document.createTextNode('Массив: ' + arr.join(', '));

resultContainer.appendChild(arrayText);

resultContainer.appendChild(document.createElement('br'));

const sumText = document.createTextNode('Сумма остатков от деления на 5: ' + sum);

resultContainer.appendChild(sumText);

resultContainer.appendChild(document.createElement('br'));

Файл 2.html

<!DOCTYPE html>

<html>

<head>

<title>MSD</title>

<link rel="stylesheet" href="2.css">

</head>

<body>

<div class="Output">

</div>

<div id="resultContainer"></div>

<script src="2.js">

</script>

</body>

</html>

Файл 2.javasqript

// 1.2. Сгенерировать массив из 1000 случайных чисел в диапазоне s ± p%, отсортировать его по убыванию остатков от деления на 10.

let s = 100;

let p = 10;

let arr = [];

let min = s \* (1 - p / 100); // 90

let max = s \* (1 + p / 100); // 110

for (let i = 0; i < 1000; i++) {

arr.push(Math.floor(Math.random() \* (max - min + 1) + min));

}

arr.sort((a, b) => (b % 10) - (a % 10));

const resultContainer = document.getElementById('resultContainer');

const arrayText = document.createTextNode('Массив: ' + arr.join(', '));

resultContainer.appendChild(arrayText);

resultContainer.appendChild(document.createElement('br'));

Файл 2.css

.Output

{

display: flex;

justify-content: center;

align-items: center;

line-height: 0.1rem;

}

body

{

height: 100vh;

background: linear-gradient(35deg, rgb(161, 156, 253) 20%, rgb(19, 19, 94) 100%);

}

Файл 3.html

<!DOCTYPE html>

<html>

<head>

<title>MSD</title>

<link rel="stylesheet" href="3.css">

</head>

<body>

<div class="Output">

</div>

<div id="resultContainer"></div>

<script src="3.js">

</script>

</body>

</html>

Файл 3.javasqript

// 1.3. Создать функцию, которая вернет: a. Среднее арифметическое своих аргументов b. Среднее гармоническое своих аргументов

let arrAll = []; //Массив для а и b

for (let i = 0; i < 10; i++) {

arrAll.push(Math.floor(Math.random()\* 100));

}

function arithmeticMean() {

let sum = 0;

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

sum += arrAll[i];

}

return sum / arrAll.length; // Возвращаем СРЕДНЕЕ арифметическое

}

function harmonicMean() {

let sum = 0;

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

sum += 1 / arrAll[i];

}

return arrAll.length / sum; // Возвращаем СРЕДНЕЕ гармоническое

}

let arrArithmetic = []

let arrHarmonic = []

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

arrArithmetic.push(arithmeticMean());

arrHarmonic.push(harmonicMean());

}

const arithmetic = arithmeticMean(1);

const harmonic = harmonicMean(1);

let arrhar = []

let arrari = []

arrari.push(arithmetic);

arrhar.push(harmonic);

const resultContainer = document.getElementById('resultContainer');

const arrayText = document.createTextNode('СРЕДНЕЕ арифметическое: ' + arrari.join(', '));

resultContainer.appendChild(arrayText);

resultContainer.appendChild(document.createElement('br'));

const arrayText2 = document.createTextNode('СРЕДНЕЕ гармоническое: ' + arrhar.join(', '));

resultContainer.appendChild(arrayText2);

resultContainer.appendChild(document.createElement('br'));

Файл 3.css

.Output

{

display: flex;

justify-content: center;

align-items: center;

line-height: 0.1rem;

}

body

{

height: 100vh;

background: linear-gradient(35deg, rgb(161, 156, 253) 20%, rgb(19, 19, 94) 100%);

}

Файл 4.html

<!DOCTYPE html>

<html>

<head>

<title>MSD</title>

<link rel="stylesheet" href="4.css">

</head>

<body>

<div class="Output">

</div>

<div id="resultContainer"></div>

<script src="4.js">

</script>

</body>

</html>

Файл 4.javasqript

// 1.4. Создать массив из списка имен. Сгенерировать массив из 20 объектов типа «Человек». У каждого объекта

// должно быть случайным образом выбрано имя, возраст, а также метод, который выводит имя и возраст данного человека. Прогнать методом обхода

// по этому массиву и заставить каждого человека представиться.

let arr = ['Иван', 'Петр', 'Коля', 'Вася', 'Катя', 'Лена', 'Саша', 'Маша', 'Лена', 'Таня', 'Стас', 'Стася', 'Андрей', 'Дмитрий', 'Михаил', 'Никита','Вильгельм','Сергей','Алина' ];

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

introduce() {

console.log(`Меня зовут ${this.name}, мне ${this.age} лет.`);

}

getName() {

return this.name;

}

getAge() {

return this.age;

}

}

let people = [];

for (let i = 0; i < 20; i++) {

let randomIndex = Math.floor(Math.random() \* arr.length);

let name = arr[randomIndex];

let age = Math.floor(Math.random() \* 90) + 1;

let person = new Person(name, age);

people.push(person);

}

for (let person of people) {

person.introduce();

}

const resultContainer = document.getElementById('resultContainer');

for (let person of people) {

const personInfo = document.createTextNode(

`Меня зовут: ${person.getName()}, мне ${person.getAge()} лет`

);

resultContainer.appendChild(personInfo); // Выводим информацию о каждом человеке

resultContainer.appendChild(document.createElement('br'));

}

Файл 4.css

.Output

{

display: flex;

justify-content: center;

align-items: center;

line-height: 0.1rem;

}

body

{

height: 100vh;

background: linear-gradient(35deg, rgb(161, 156, 253) 20%, rgb(19, 19, 94) 100%);

}