|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_ Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_ Программное обеспечение ЭВМ и информационные технологии \_\_\_\_\_\_

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Дисциплина Архитектура ЭВМ

Студент Мишина Елена Викторовна

*фамилия, имя, отчество*

Группа ИУ7-54Б

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Попов А.Ю.

*подпись, дата фамилия, и.о.*

*2020 г.*

Оглавление

[Цель работы: 3](#_Toc54790147)

[Task 3 4](#_Toc54790148)

[1. Задание 1 4](#_Toc54790149)

[2. Задание 2 5](#_Toc54790150)

[3. Задание 3 6](#_Toc54790151)

[4. Задание 4 8](#_Toc54790152)

[5. Задание 5 9](#_Toc54790153)

[6. Задание 6 10](#_Toc54790154)

[7. Задание 7 11](#_Toc54790155)

[Task 4 13](#_Toc54790156)

[1. Задание 1 13](#_Toc54790157)

[2. Задание 2 15](#_Toc54790158)

[3. Задание 3 16](#_Toc54790159)

[4. Задание 4 18](#_Toc54790160)

[Вывод: 21](#_Toc54790161)

# Цель работы:

Написание программ, демонстрирующих работу с форматом JSON, с файлами на языке JavaScript, а также демонстрирующих умение запускать сервер, получать страницу HTML и работать с формами.

# Task 3

# Задание 1

**Условие задачи:**

С клавиатуры считывается число N. Далее считывается N строк. Необходимо создать массив и сохранять в него строки только с четной длинной. Получившийся массив необходимо преобразовать в строку JSON и сохранить в файл.

**Листинг программы:**

"use strict";

const readlineSync = require('readline-sync')

function read\_number(){

    const number = readlineSync.question("Input count of strings: ")

    return parseInt(number)

}

function read\_string(){

    const str = readlineSync.question("Input string: ")

    return str

}

function create\_arr(number){

    const obj = {};

    obj.mas = []

    let count = 0

    for(let i = 0; i < number; i++){

        let a = read\_string()

        if (a.length % 2 === 0)

        {

            obj.mas[count] = a

            count++

        }

    }

    return obj

}

function write\_file(obj, filename){

    const jsonMAS = JSON.stringify(obj)

    const fs = require("fs")

    const nameString = filename

    fs.writeFileSync(nameString, jsonMAS);

}

**Тестирование:**

function main(){

    let number = read\_number()

    let obj = create\_arr(number)

    write\_file(obj, "result.txt")

}

main()

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_1.js

Input count of strings: 5

Input string: fghjk

Input string: dvbnm

Input string: ertyui

Input string: ghj

Input string: fghjkl;



# Задание 2

**Условие задачи:**

Необходимо считать содержимое файла, в котором хранится массив строк в формате JSON. Нужно вывести только те строки на экран, в которых содержатся только гласные буквы.

**Листинг программы:**

"use strict";

function read\_file(filename){

    const fs = require("fs")

    const nameString = filename

    const contentString = fs.readFileSync(nameString, "utf8")

    const obj = JSON.parse(contentString)

    return obj

}

function check\_vowels(obj){

    var vowels = "aeiouyq"

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

        let count = 0

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

            for (let k = 0; k < vowels.length; k++){

                if (obj.mas[i][j] === vowels[k]){

                    count++

                }

            }

        if (count === obj.mas[i].length){

            console.log(obj.mas[i])

            }

        }

    }

}

**Тестирование:**

function main(){

    const obj = read\_file("text.txt")

    check\_vowels(obj)

}

main()



PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_2.js

Aeqy

# Задание 3

**Условие задачи:**

С клавиатуры считывается строка - название расширения файлов. Далее считывается строка - адрес папки. Необходимо перебрать все файлы в папке и вывести содержимое файлов, у которых расширение совпадает с введенным расширением.

**Листинг кода:**

" "use strict";

const readlineSync = require('readline-sync')

const fs = require("fs");

function read\_extension(){

    const extension = readlineSync.question("Input file extension: ")

    return extension

}

function read\_folder(){

    const folder = readlineSync.question("Input folder: ")

    return folder

}

function read\_name\_files(folder){

    const arr = fs.readdirSync(folder);

    return arr

}

function get\_info(arr, extension){

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

        const fileName = arr[i];

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

            if(arr[i][j] === "."){

                var temp = ''

                for(let k = j + 1; k < arr[i].length; k++){

                    temp += arr[i][k]

                }

                if (temp === extension){

                    const contentString = fs.readFileSync(arr[i], "utf8");

                    console.log(contentString);

                }

            }

        }

    }

}

**Тестирование:**

function main(){

    get\_info(read\_name\_files(read\_folder()), read\_extension())

}

main()

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_3.js

Input folder: ./

Input file extension: txt

14555 gghj

cvbn xcvbn ertyuj

dfghjkl fdghjkl;m fghjkl;

{"mas":["ertyui"]}

{"mas":[ "Ford", "BMW", "Fiat", "aeqy" ]}

{"mas":[ "Ford", "BMW", "Fiat", "aeqy" ]}

# Задание 4

**Условие задачи:**

Дана вложенная структура файлов и папок. Все файлы имеют расширение "txt". Необходимо рекурсивно перебрать вложенную структуру и вывести имена файлов, у которых содержимое не превышает по длине 10 символов.

**Листинг кода:**

"use strict";

const fs = require("fs");

const folder = "./test";

const arr = fs.readdirSync(folder);

function check\_txt(filename){

    for(let j = 0; j < filename.length; j++){

        if(filename[j] === "."){

            var temp = ''

            for(let k = j + 1; k < filename.length; k++){

                temp += filename[k]

            }

            if (temp === "txt"){

                return true

            }

        }

    }

    return false

}

function file\_recursion(folder){

    const arr = fs.readdirSync(folder);

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

        if (check\_txt(arr[i]) === true){

            const contentString = fs.readFileSync(folder + "/" + arr[i], "utf8");

            if (contentString.length <= 10){

                console.log(arr[i])

            }

        }

        else {

            file\_recursion(folder + "/" + arr[i]);

        }

    }

}

**Тестирование:**

function main(){

    file\_recursion(folder)

}

main()

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_4.js

test\_txt.txt

# Задание 5

**Условие задачи:**

С клавиатуры считывается число N. Далее считывается N строк - имена текстовых файлов. Необходимо склеить всё содержимое введенных файлов в одну большую строку и сохранить в новый файл.

**Листинг кода:**

"use strict";

const readlineSync = require('readline-sync')

const number = readlineSync.question("Input N: ")

const fs = require("fs");

var text = ''

for(let i = 0; i < parseInt(number); i++){

    const name = readlineSync.question("Input file name: ")

    const contentString = fs.readFileSync(name, "utf8");

    text += contentString

}

fs.writeFileSync("result3\_5.txt", text);

**Тестирование:**

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_5.js

Input N: 3

Input file name: a.txt



Input file name: b.txt



Input file name: c.txt





# Задание 6

**Условие задачи:**

Написать код, который позволяет определить максимальный возможный уровень вложенности друг в друга полей в объекте, чтобы данный объект можно было преобразовать в строку формата JSON. Ответом является целое число.

**Листинг кода:**

let a = {}

let count = 0;

let flag = true;

while (count < 100000 && flag) {

    try {

        const jsonString = JSON.stringify(a);

        a = {"item": a};

        if (count < 10) {

            console.log(a);

        }

    } catch (err) {

        flag = false;

        console.error(err.message);

    }

    count++

}

console.log("result: ",count);

**Тестирование:**

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_6.js

{ item: {} }

{ item: { item: {} } }

{ item: { item: { item: {} } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

{ item: { item: { item: [Object] } } }

Maximum call stack size exceeded

result: 966

# Задание 7

**Условие задачи:**

Из файла считывается строка в формате JSON. В этой строке информация об объекте, в котором находится большое количество вложенных друг в друга полей. Объект представляет из себя дерево. Необходимо рекурсивно обработать дерево и найти максимальную вложенность в дереве. Необходимо вывести на экран ветку с максимальной вложенностью.

**Листинг кода:**

'use strict';

function GetDeepestBranch(obj) {

    let bestEl = null

    let bestTrack = {}

    let depth = 0

    if (obj === null || typeof obj != 'object') {

        return [obj, 0]

    }

    for (const [key, val] of Object.entries(obj)) {

        const [elTrack, elDepth] = GetDeepestBranch(val)

        if (elDepth + 1 > depth) {

            bestEl = key

            bestTrack = elTrack

            depth = elDepth + 1

        }

    }

    return [{[bestEl] : bestTrack}, depth]

}

const fs = require('fs')

const path = require('path')

const readlineSync = require('readline-sync');

const fIn = 'text.txt'

const content = fs.readFileSync(fIn, 'utf8')

const obj = JSON.parse(content)

const [res, \_] = GetDeepestBranch(obj)

console.log(JSON.stringify(res, null, 2))

**Тестирование:**

PS C:\Users\mishi\Desktop\EVM\task3\_2> npm start

> task3\_2@1.0.0 start C:\Users\mishi\Desktop\EVM\task3\_2

> node task3\_7.js

{

"user": {

"username": "noobishe"

}

}

# Task 4

# Задание 1

**Условие задачи:**

Запустить сервер. Реализовать на сервере функцию для сравнения трёх чисел и выдачи наибольшего из них. Реализовать страницу с формой ввода для отправки запроса на сервер.

**Листинг программы:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/compare/num", function(request, response) {

    const a = request.query.a;

    const b = request.query.b;

    const c = request.query.c;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

    const cInt = parseInt(c);

    if (aInt > bInt && aInt > cInt){

        const answerJSON = JSON.stringify({result: aInt});

        response.end(answerJSON);

    }

    else if (bInt > aInt && bInt > cInt){

        const answerJSON = JSON.stringify({result: bInt});

        response.end(answerJSON);

    }

    else{

        const answerJSON = JSON.stringify({result: cInt});

        response.end(answerJSON);

    }

});

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Задание 1</title>

</head>

<body>

    <h1>Страница A</h1>

    <form method="GET" action="/compare/num">

        <p>Введите A</p>

        <input name="a" spellcheck="false" autocomplete="off">

        <p>Введите B</p>

        <input name="b" spellcheck="false" autocomplete="off">

        <p>Введите C</p>

        <input name="c" spellcheck="false" autocomplete="off">

        <br>

        <br>

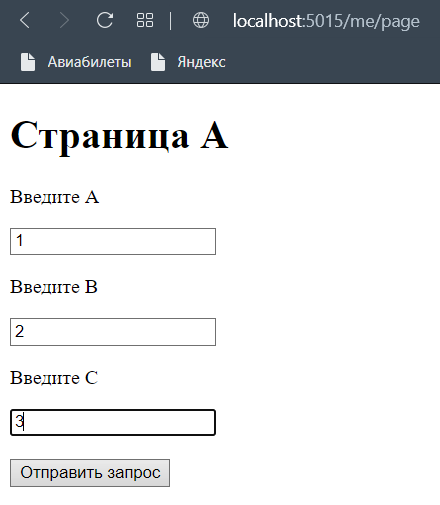
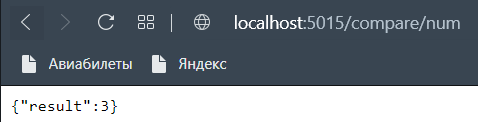
        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

**Тестирование:**

# Задание 2

**Условие задачи:**

Запустить сервер. На стороне сервера должен храниться файл, внутри которого находится JSON строка. В этой JSON строке хранится информация о массиве объектов. Реализовать на сервере функцию, которая принимает индекс и выдает содержимое ячейки массива по данному индексу. Реализовать страницу с формой ввода для отправки запроса на сервер.

**Листинг программы:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/mas", function(request, response) {

    const index = request.query.index;

    const indexInt = parseInt(index);

    const fs = require("fs");

    const nameString = "test.txt";

    const contentString = fs.readFileSync(nameString, "utf8");

    const mas = JSON.parse(contentString);

    if (indexInt < mas.length){

        const answerJSON = JSON.stringify({result: mas[indexInt]});

        response.end(answerJSON);

    }

});

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Задание 2</title>

</head>

<body>

    <h1>Страница A</h1>

    <form method="GET" action="/mas">

        <p>Введите индекс</p>

        <input name="index" spellcheck="false" autocomplete="off">

        <br>

        <br>

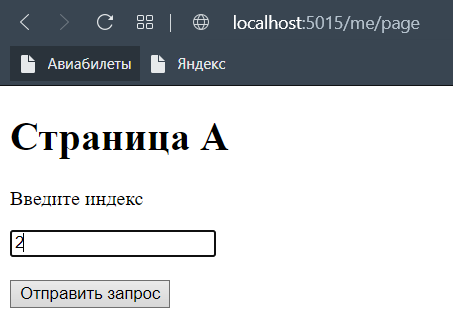
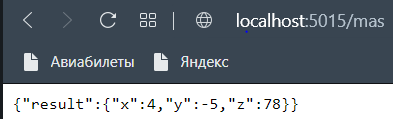
        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

**Тестирование:**

# Задание 3

**Условие задачи:**

Написать программу, которая на вход получает массив названий полей и адрес запроса (куда отправлять). Программа должна генерировать HTML разметку страницы, в которую встроена форма для отправки запроса.

**Листинг кода:**

"use strict";

const readlineSync = require('readline-sync')

function read\_string(){

    const str = readlineSync.question("Input string: ")

    return str

}

const fs = require("fs")

function create\_page(arr\_name, adress){

    let data\_1 = `<!DOCTYPE html>

    <html>

    <head>

        <meta charset="UTF-8">

        <title>Задание 3</title>

    </head>

    <body>

        <h1>Страница A</h1>

        <form method="GET" action="`

    data\_1 += adress + '">\n'

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

        data\_1 += '            <input name="' + arr\_name[i] + '" spellcheck="false" autocomplete="off"></input>\n'

    }

    data\_1 += `            <br>

            <br>

            <input type="submit" value="Отправить запрос">

        </form>

</body>

</html>`

fs.writeFileSync("page.html", data\_1)

}

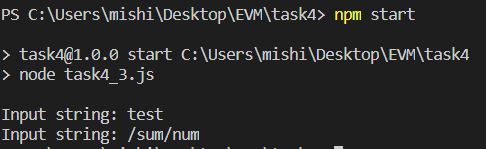
let name = read\_string()

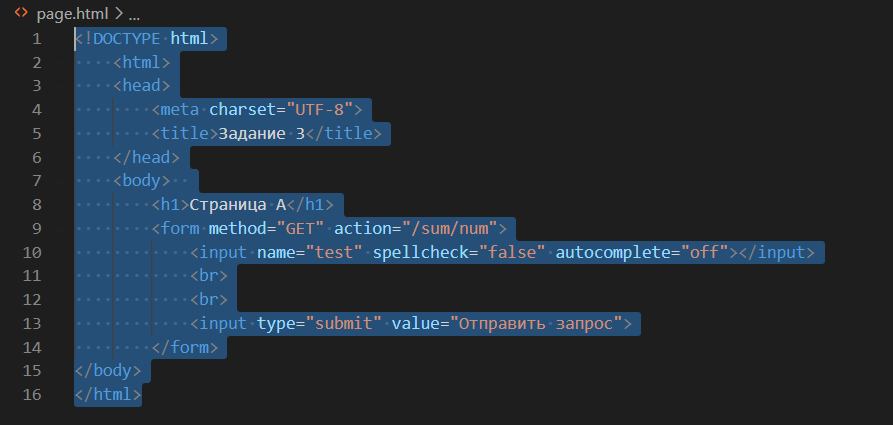
let url = read\_string()

name = name.split(' ')

create\_page(name, url)

**Тестирование:**





# Задание 4

**Условие задачи:**

Запустить сервер. Реализовать на сервере функцию, которая принимает на вход числа A, B и C. Функция должна выдавать массив целых чисел на отрезке от A до B, которые делятся на C нацело.

**Листинг кода:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/mas/num", function(request, response) {

    const a = request.query.a;

    const b = request.query.b;

    const c = request.query.c;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

    const cInt = parseInt(c);

    let mas = []

    let count = 0

    for (let i = aInt; i <= bInt; i++){

        if (i % cInt === 0){

            mas[count] = i

            count++

        }

    }

    const answerJSON = JSON.stringify({result: mas});

    response.end(answerJSON);

});

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Задание 4</title>

</head>

<body>

    <h1>Страница A</h1>

    <form method="GET" action="/mas/num">

        <p>Введите A</p>

        <input name="a" spellcheck="false" autocomplete="off">

        <p>Введите B</p>

        <input name="b" spellcheck="false" autocomplete="off">

        <p>Введите C</p>

        <input name="c" spellcheck="false" autocomplete="off">

        <br>

        <br>

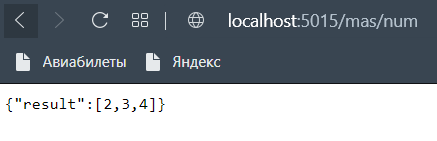
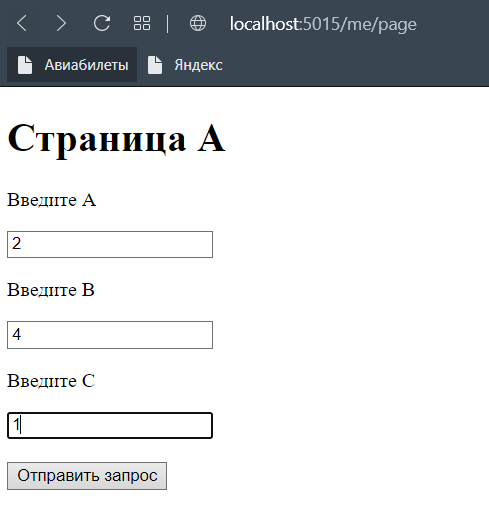
        <input type="submit" value="Отправить запрос">

    </form>

</body>

</html>

**Тестирование:**



# Вывод:

Были приобретены новые знания по JavaScript. Все задачи, представленные в данной лабораторной работе, на знание файлов и серверов выполнены успешно.