|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования 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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Архитектура ЭВМ»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 2**

|  |  |
| --- | --- |
| **Студент** \_\_Аникин И. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Группа** \_\_ИУ7-51Б\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Преподаватель** \_\_Попов А. Ю.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

Москва.

2020 г.

Оглавление

[Задание 3.1 3](#_Toc52146684)

[Задание 3.2 3](#_Toc52146685)

[Задание 3.3 4](#_Toc52146686)

[Задание 3.4 5](#_Toc52146687)

[Задание 3.5 6](#_Toc52146688)

[Задание 3.6 6](#_Toc52146689)

[Задание 3.7 7](#_Toc52146690)

[Задание 4.1 9](#_Toc52146691)

[Задание 4.2 10](#_Toc52146692)

[Задание 4.3 11](#_Toc52146693)

[Задание 4.4 13](#_Toc52146694)

Цель работы: знакомство с форматом JSON, работа с текстовыми файлами, использование фреймворка express.

# Задание 3.1

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 readline = require('readline-sync');

const fs = require('fs');

let n = readline.questionInt("Input n: ");

let a = [];

let s;

for (let i = 0; i < n; i++) {

    s = readline.question("Input " + (i+1) + " string: ");

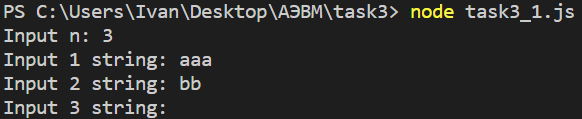
    if (s.length % 2 == 0)

        a.push(s);

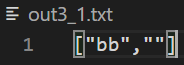
}

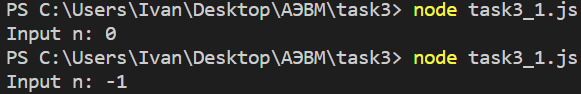
fs.writeFileSync("out3\_1.txt", JSON.stringify(a));

Примеры работы



Содержимое файла out3\_1.txt:





Выходной файл в данном случае пуст.

# Задание 3.2

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"

const fs = require('fs');

function allVowels(s) {

let vowels = "aeiouy";

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

let f = false;

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

if (s[i] === vowels[j])

f = true;

if (!f)

return false;

}

return true;

}

let filename = "out3\_1.txt";

let a = JSON.parse(fs.readFileSync(filename));

let n = a.length;

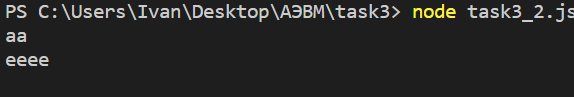
for (let i = 0; i < n; i++) {

if (allVowels(a[i]))

console.log(a[i]);

}

Примеры работы



Содержимое входного файла: [“aa”, “bbb”, “eeee”, “”].

# Задание 3.3

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 программы

"use strict"

const readline = require("readline-sync");

const fs = require("fs");

const path = require("path");

let extension = readline.question("Input extension: ");

let dirpath = readline.questionPath("Input directory path: ");

let files = fs.readdirSync(dirpath);

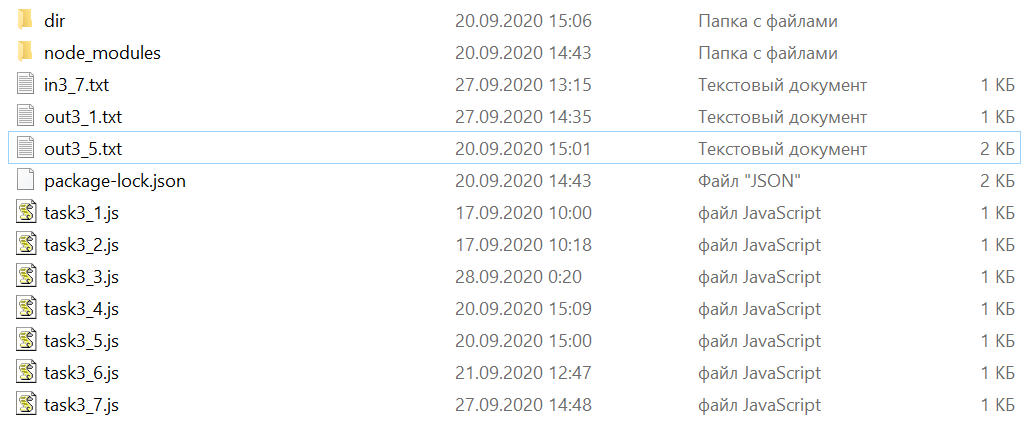
for (let i = 0; i < files.length; i++)

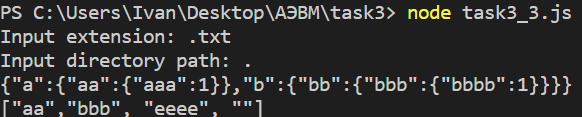
if (path.extname(files[i]) === extension)

console.log(fs.readFileSync(files[i], "utf-8"));

Примеры работы

Работа на примере следующей директории





Выводится содержимое файлов in3\_7.txt, out3\_1.txt, out3\_5.txt.

# Задание 3.4

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

Листинг программы

"use strict"

const fs = require("fs");

const path = require("path");

function f(start\_path) {

let files = fs.readdirSync(start\_path);

let n = files.length;

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

if (fs.statSync(start\_path + '/' + files[i]).isDirectory()) {

f(start\_path + '/'+ files[i]);

} else {

if (fs.readFileSync(start\_path + '/' + files[i]).length <= 10)

console.log(files[i]);

}

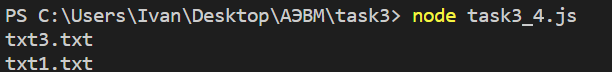
}

}

f("./dir");

Пример работы

Для примера была создана директория dir. Внутри директории находится папка dir1.txt и два файла txt1.txt , txt2.txt. Внутри папки dir1.txt находится файл txt3.txt. Файлы txt1.txt и txt3.txt пустые. В файле txt2.txt строка из 21 символа.



# Задание 3.5

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 программы

"use strict"

const fs = require("fs");

const readline = require("readline-sync");

let n = readline.questionInt("Input n: ");

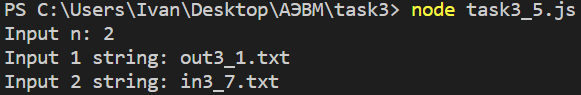
let s = "";

for (let i = 0; i < n; i++)

s += fs.readFileSync(readline.questionPath("Input " + (i+1) + " string: "));

fs.writeFileSync("out3\_5.txt", s);

Пример работы



Содержимое файла out3\_5.txt

["aa","bbb", "eeee", ""]{"a":{"aa":{"aaa":1}},"b":{"bb":{"bbb":{"bbbb":1}}}}

# Задание 3.6

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 программы

"use strict"

let i = 0;

function f(obj) {

try {

JSON.stringify(obj);

} catch (error) {

console.log(i);

return;

}

i++;

obj = {x: obj};

f(obj);

}

f({})

Пример работы



# Задание 3.7

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.

"use strict"

const fs = require("fs");

const datapath = "./in3\_7.txt";

let nested = 0;

let max\_nested = 0;

function f(obj) {

let keys = Object.keys(obj);

let n = keys.length;

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

nested++;

if (nested > max\_nested)

max\_nested = nested;

f(obj[keys[i]]);

nested--;

}

return max\_nested;

}

let arr = []

function g(obj) {

let keys = Object.keys(obj);

let n = keys.length;

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

arr.push(keys[i]);

nested++;

if (nested == max\_nested)

console.log(arr.join('.'));

g(obj[keys[i]]);

nested--;

arr.pop();

}

return max\_nested;

}

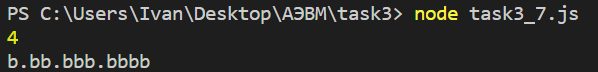
let obj = JSON.parse(fs.readFileSync(datapath));

console.log(f(obj));

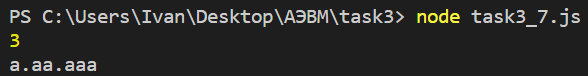
g(obj);

Пример работы

Исходный файл: {"a":{"aa":{"aaa":1}},"b":{"bb":{"bbb":{"bbbb":1}}}}



Исходный файл: {"a":{"aa":{"aaa":1}},"b":{"bb": 1}}



# Задание 4.1

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 express = require("express");

const fs = require("fs");

const app = express();

const port = 5015;

app.listen(port);

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

app.get("/", 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("/maxthree", 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);

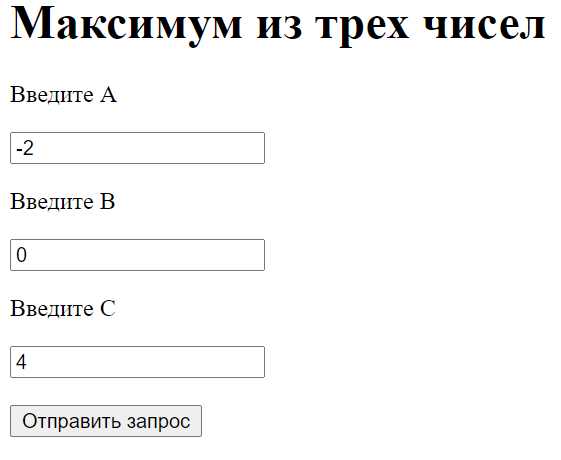
    const minInt = Math.max(aInt, bInt, cInt);

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

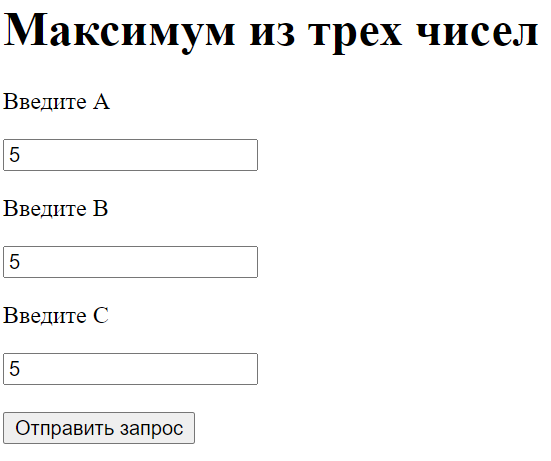
    response.end(answerJSON);

});

Примеры работы









# Задание 4.2

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 express = require("express");

const fs = require("fs");

const app = express();

const port = 5015;

app.listen(port);

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

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

    const index = request.query.index;

    const i = parseInt(index);

    let arr = JSON.parse(fs.readFileSync("data.txt"));

    let answerJSON;

    if (i >= 0 && i < arr.length)

        answerJSON = JSON.stringify({result: arr[i]});

    else

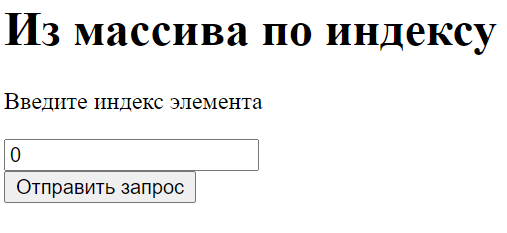
        answerJSON = JSON.stringify({error: "Index out of bounds!"});

    response.end(answerJSON);

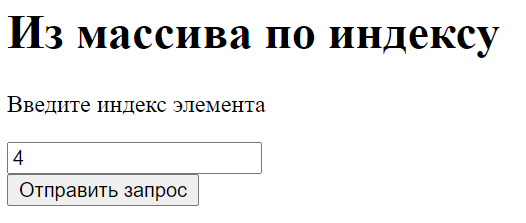
});

Примеры работы

Массив для работы: [1, 2, 3, 4 5]









# Задание 4.3

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 программы

"use strict";

const express = require("express");

const fs = require("fs");

const app = express();

const port = 5015;

app.listen(port);

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

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

const arr = JSON.parse(request.query.array);

const to = request.query.to;

let s = "<form method=GET action=\"" + to +"\">\n"

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

s += "<p> Input value of field " + arr[i] + "</p>\n"

s += "<input name=\"" + arr[i] + "\" spellcheck=\"false\" autocomplete=\"off\">\n";

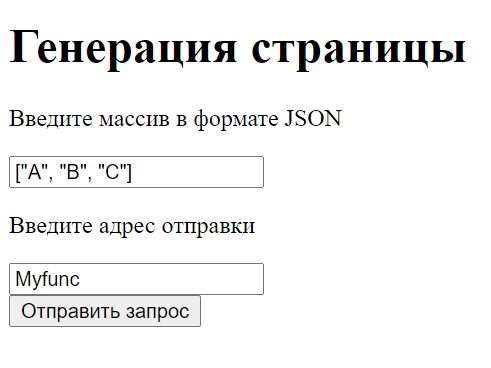
}

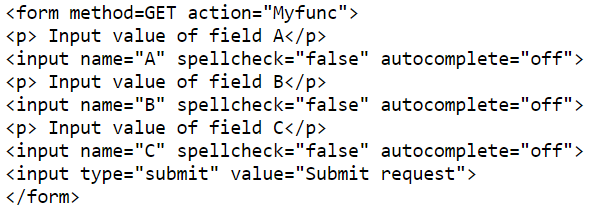
s += "<input type=\"submit\" value=\"Submit request\">\n</form>";

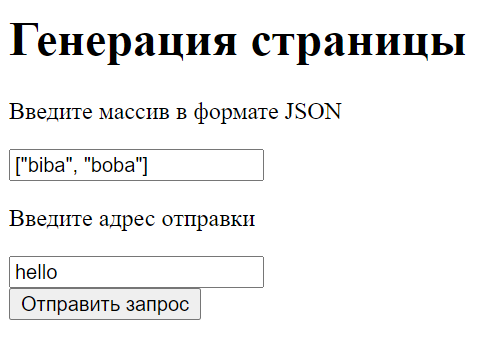
response.end(s);

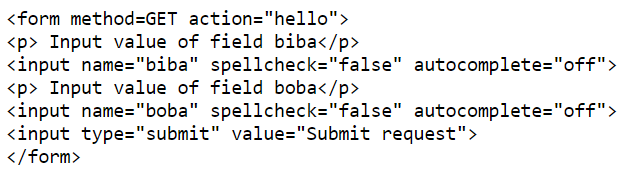
});

Примеры работы









# Задание 4.4

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 программы

"use strict";

const express = require("express");

const fs = require("fs");

const app = express();

const port = 5015;

app.listen(port);

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

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

const a = parseInt(request.query.a);

const b = parseInt(request.query.b);

const c = parseInt(request.query.c);

let arr = [];

for (let i = a; i <= b; i++)

if (i % c == 0)

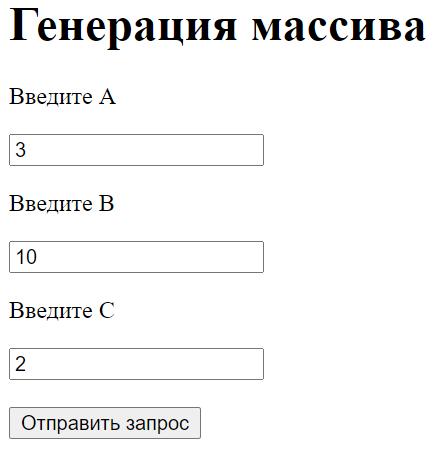
arr.push(i);

let answerJSON = JSON.stringify({result: arr});

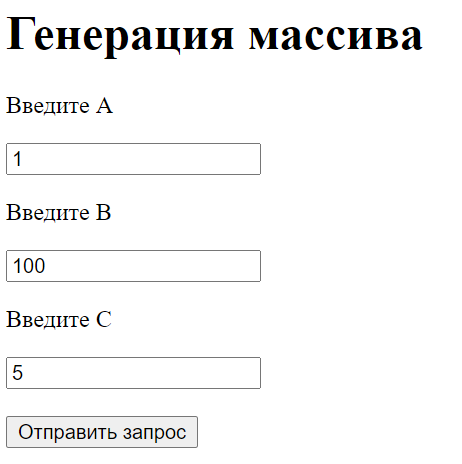
response.end(answerJSON);

});

Примеры работы









Вывод: в ходе данной лабораторной работы были изучен формат JSON, приобретены навыки работы с фреймворком express и текстовыми файлами.