



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 КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника;

**О Т Ч Е Т**

по лабораторной работе № 2

Дисциплина: Архитектура ЭВМ

Студент

ИУ7-526

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Кузин А.А.

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Попов А.Ю.

(И.О. Фамилия)

Москва, 2020

# Цель

Целью данной лабораторной работы является ознакомление с формат JSON и его обработкой, а также возможностями работы с серверами, используя JavaScript.

## Часть 1

### Задание 1

С клавиатуры считывается число N. Далее считывается N строк. Необходимо создать массив и сохранять в него строки только с четной длиной.

Получившийся массив необходимо преобразовать в строку JSON и сохранить в файл.

### Листинг программы:

```
"use strict"

const fs = require("fs");
const fileName = "result.txt";

const readlineSync = require('readline-sync');
const n = parseInt(readlineSync.question("Input N: "));

let stringArr = [];
for (let i = 0; i < n; i++){
    const line = readlineSync.question("Input string: ");
    if (line.length % 2 === 0)
        stringArr.push(line);
}

fs.writeFileSync(fileName, JSON.stringify(stringArr));
```

### Тесты

Input N: 6

Input string: abc

Input string: def

Input string: 123

Input string: l.z

Input string: asd as

Input string: qwe

**Результат:** ["asd as"]

Input N: 3  
Input string: 12  
Input string: 34  
Input string: 56  
**Результат:** ["12","34","56"]

Input N: 2  
Input string: 1  
Input string: 1  
**Результат:** []

## Задание 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 fileName = "result.txt";
const fs = require("fs");

if (!fs.existsSync(fileName)){
  console.log("File not found!");
}
const jsonArr = fs.readFileSync(fileName, "utf8");

const stringArr = JSON.parse(jsonArr);

function isVowel(letter) {
  const vowels = ["a", "e", "i", "o", "u"];
  let vowel = false;
  for (let j = 0; j < vowels.length; j++) {
    if (letter === vowels[j])
      vowel = true;
  }
  return vowel;
}

function checkOnlyVowels(stringToCheck){
  for (let i = 0; i < stringToCheck.length; i++){
    if (!isVowel(stringToCheck[i]))
      return false;
  }
  return true;
}
```

```
for (let i = 0; i < stringArr.length; i++){
  if (checkOnlyVowels(stringArr[i]))
    console.log(stringArr[i]);
}
```

## Тесты

Содержимое файла

["bcd", "abdce", "bfedc", "aaa", "lo"]

**Результат:** aaa

## Задание 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 fs = require("fs");
const readlineSync = require('readline-sync');
```

```
const fileExtension = readlineSync.question("Input file extension: ");
const directory = readlineSync.question("Input directory: ");
```

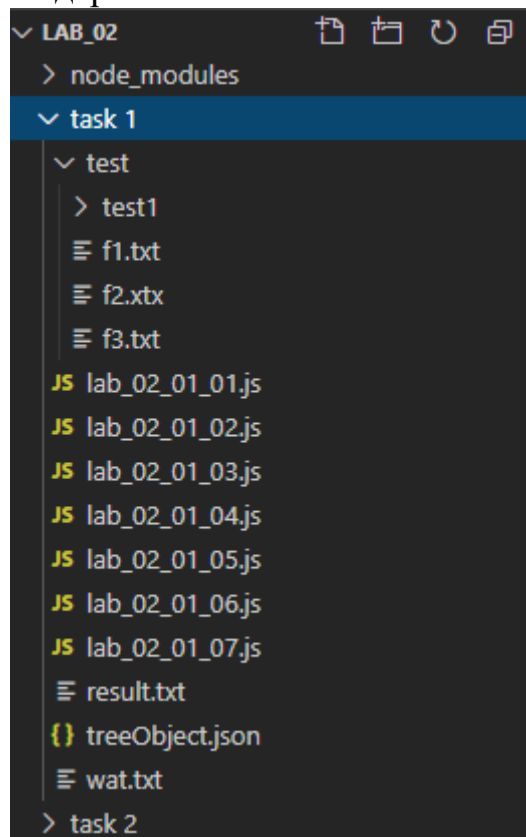
```
const files = fs.readdirSync(directory);
```

```
function isEqual(s1, s2) {
  let l1 = s1.length;
  let l2 = s2.length;
  if (l1 !== l2)
    return false;
  for (let i = 0; i < l1; i++)
    if (s1[i] !== s2[i])
      return false;
  return true;
}
```

```
for (let i = 0; i < files.length; i++){
  const currentExtension = files[i].split(".")[1];
  if (currentExtension && isEqual(currentExtension, fileExtension)){
    console.log("-----\n" + files[i] + ":");
    console.log(fs.readFileSync(directory+files[i], "utf-8"));
  }
}
```

## Тесты

Содержимое папки:



Ввод:

Input file extension: txt

Input directory: test\

Результат:

-----

f1.txt:

razrazraz

-----

f3.txt:

test2

## Задание 4

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 программы

```
"use strict"
```

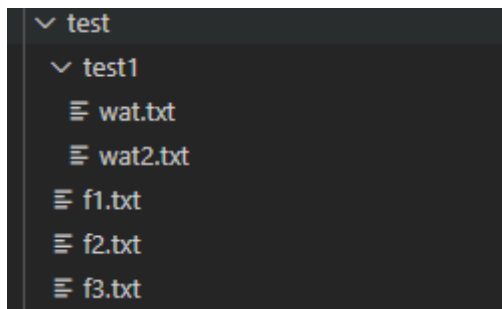
```
const fs = require("fs");
```

```
function findFiles(directory){
  const everything = fs.readdirSync(directory);
  for (let i = 0; i < everything.length; i++){
    const file = everything[i].split(".");
    if (file.length === 1){
      findFiles(directory + "/" + everything[i]);
    }
    else{
      const fileName = directory + "/" + everything[i];
      if (fs.readFileSync(fileName, "utf-8").length <= 10)
        console.log(fileName);
    }
  }
}

findFiles("test");
```

## Тесты:

Структура файлов и папок в папке test:



Содержимое файлов:

f1.txt: razrazraz

f2.txt: Task 2 test or not

f3.txt: test2

wat.txt: Long sentence 12346

wat2.txt: test2

## Результат работы:

test/f1.txt

test/f3.txt

test/test1/wat2.txt

## Задание 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 readlineSync = require('readline-sync');

const n = parseInt(readlineSync.question("Input n: "));

let fileNameArr = [];
for (let i = 0; i < n; i++) {
    const line = readlineSync.question("Input file name: ");
    fileNameArr.push(line);
}

let result = ""
for (let i = 0; i < n; i++){
    const content = fs.readFileSync(fileNameArr[i], "utf-8").split("\n");
    for (let j = 0; j < content.length; j++)
        result += " " + content[j].replace("\r", "");
}

fs.writeFileSync("newresult.txt", result);
```

### Тесты:

Ввод:

Input n: 2  
Input file name: f1.txt  
Input file name: f2.txt  
Содержимое файлов:  
f1.txt: razrazraz  
f2.txt: Task 2 test or not

### Результат работы:

newresult.txt: razrazraz Task 2 test or not

## Задание 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 object = {};

let i = 0;
let curobject = object;

while (true) {
  try{
    curobject["field"] = {};
    curobject = curobject["field"];
    JSON.stringify(object);
    i += 1;
  }
  catch{
    console.log(i);
    break;
  }
}
```

### Тесты:

### Результат работы:

```
E:\study\5sem\arch\lab_02\task 1>node lab_02_01_06.js
964
```

## Задание 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.

## Листинг программы

```
"use strict"
const fs = require("fs");

function findDeepest(object) {
  let level = 1;
  for (let key in object) {
```



```

    if (typeof object[key] === 'object') {
      let depth = findDeepest(object[key]) + 1;
      level = Math.max(depth, level);
    }
  }
  return level;
}

let arr = []

function outputDeepest(object, maxDepth, curDepth=1) {
  if (curDepth === maxDepth){
    return 1;
  }

  for (let key in object) {
    if (typeof object[key] === 'object') {
      if (outputDeepest(object[key], maxDepth, curDepth + 1)){
        arr.push(key);
        return 1;
      }
    }
  }
  return 0;
}

function findOutputDeepest(object) {
  const maxDepth = findDeepest(object);
  outputDeepest(object, maxDepth);
  console.log(arr.reverse());
}

```

```

let fileName = 'treeObject.json';
const objectToLook = JSON.parse(fs.readFileSync(fileName, "utf-8"));
findOutputDeepest(objectToLook);

```

## Тесты:

Содержимое файла treeObject.json:

```

{
  "a": {
    "a": 1,
    "b": 1
  },
  "b": {
    "a": {
      "a": {
        "a": 1
      },
      "b": 1
    },
    "b": 1
  },
}

```

```
    "b": {  
      "a": 1,  
      "b": 1  
    }  
  }  
}
```

**Результат работы:**

```
[ 'b', 'a', 'a' ]
```

## Часть 2

### Задание 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 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("/calculate/max", function (request, response) {  
  const a = parseInt(request.query.a);  
  const b = parseInt(request.query.b);  
  const c = parseInt(request.query.c);
```

```
const max = Math.max(a, b, c);
const answerJSON = JSON.stringify({ result: max });
response.end(answerJSON);
});
```

**Тесты:**

## Вывод максимального из трёх

Введите A

Введите B

Введите C

Отправить запрос

**Результат работы:**

```
{"result":5}
```

## Задание 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 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("/me/item", function (request, response) {
  const i = parseInt(request.query.i);
  const jsonString = fs.readFileSync("array.json");
  const array = JSON.parse(jsonString);
  console.log(array);
  const element = array.array[i]
  const answerJSON = JSON.stringify({ result: element });
  response.end(answerJSON);
});

```

### Тесты:

Содержимое файла array.json: {"array" : [1, 2, 3, 4, 5]}

## Вывод элемента массива

Введите индекс ячейки



### Результат работы:

```
{"result":5}
```

## Задание 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 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);
  }
});

function generateHtml(fields, address) {
  let pageHtml = '<!DOCTYPE html>\n<html>\n<head>\n<meta charset="UTF-8">\n\n<title>Автоматически</title>\n</head>\n<body>';
  pageHtml += '<h1>Автоматически</h1><form method = "GET" action="" + address + "">\n';
  for (let i = 0; i < fields.length; i++) {
    pageHtml += '<p>' + fields[i] + '</p> \n<input name = "' + fields[i] + '" spellcheck = "false" autocomplete = "off" >\n';
  };
  pageHtml += '<br>\n<br>\n<input type="submit" value="Отправить запрос">\n</form>\n</body>\n</html>';
  return pageHtml;
}

app.get("/me/generate", function (request, response) {
  const fields = request.query.fields;
  const address = request.query.address;
  let fieldsArray = fields.split(",");
  let pageHtml = generateHtml(fieldsArray, address);
  response.end(pageHtml);
});

```

## Тесты:

Страница и введённые объекты:

# Формирование HTML развёртки

Введите поля формы

Введите адрес

Сгенерированная страница:

## Автоматически

first

second

third

## Задание 4

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

## 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);
  }
});

function formArray(a, b, c) {
  let result = [];
  for (let i = a; i <= b; i++) {
    if (i % c === 0)
      result.push(i);
  }
  return result;
}

app.get("/output/array", function (request, response) {
  const a = parseInt(request.query.a);
  const b = parseInt(request.query.b);
  const c = parseInt(request.query.c);
  const elements = formArray(a, b, c);
  const answerJSON = JSON.stringify({ result: elements });
  response.end(answerJSON);
});

```

**Тесты:**

## Вывод кратных элементов

Введите A

Введите B

Введите C



**Результат работы:**

```
{ "result": [0, 3, 6, 9] }
```

## **Вывод**

В результате выполнения данной лабораторной работы я получил навыки работы с форматом JSON, его обработкой, работы с файлами, а также навыки поднятия и использование серверов используя express.