



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

Тема Строки формата JSON. Работа с файлами. Запуск сервера.

Студентка Воякин А. Я.

Группа ИУ7-54Б

Оценка (баллы) _____

Преподаватель Попов А. Ю.

Москва.
2020 г.

Задание 1.1

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 rl = require("readline-sync");

const n = rl.question("Введите кол-во строк: ");
let arr = [];
for (let i = 0; i < parseInt(n); i++) {
    let str = rl.question('№' + (i + 1) + ': ');
    if (str.length % 2 === 0) arr.push(str);
}
let jsonString = JSON.stringify(arr);
fs.writeFileSync("result.json", jsonString);
console.log("Completed");
```

Демонстрация работы программы:

```
Введите кол-во строк: 3
№1: qw
№2: asdfg
№3: zxcvbn
Completed
```

Содержимое файла result.json после выполнения:

```
["qw", "zxcvbn"]
```

Задание 1.2

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");

const vowels = "aeiouyaeёиоуыэя";

function check(str) {
    let ch_num = str.split("").filter(char => vowels.includes(char)).length;
```

```

    if (ch_num === str.length) console.log(str);
  }

  if (!fs.existsSync("input.json")) return;
  let jsonString = fs.readFileSync("input.json", "utf8");
  let arr = JSON.parse(jsonString);
  arr.forEach(str => check(str));

```

Демонстрация работы программы:

Содержимое файла input.json

```

["qwerty", "hello world!", "hrdw", "aeey", "nwtpdr", "null", "oioi", "how are you?", "Moscow"]

```

Результат выполнения программы:

```

aeey
oioi

```

Задание 1.3

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 rl = require("readline-sync");

const ext = rl.question("Введите расширение файлов: ");
const dir = rl.question("Введите адрес папки: ");

let arr = fs.readdirSync(dir);

function check_ext(filename) {
  if (filename.endsWith(ext)) {
    console.log("\n----" + filename + "----");
    let content = fs.readFileSync(dir + '/' + filename, "utf8");
    console.log(content);
  }
}

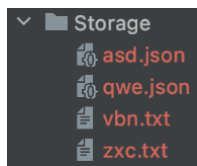
arr.forEach(filename => check_ext(filename));

```

Таблица 3. – Тесты к заданию 1.3

Демонстрация работы программы:

Содержимое директории Storage:



Результат работы программы:

```
Введите расширение файлов: txt
Введите адрес папки: /Users/Alex/Documents/NodeJS/task_3/3/Storage

----vbn.txt----
vbn file txt

----zxc.txt----
zxc file txt
```

Задание 1.4

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 output(name, dir) {
  if (!fs.lstatSync(dir + '/' + name).isDirectory() && name.endsWith(".txt")) {
    let content = fs.readFileSync(dir + '/' + name, "utf8");
    if (content.length <= 10) {
      console.log(name);
    }
  }
}

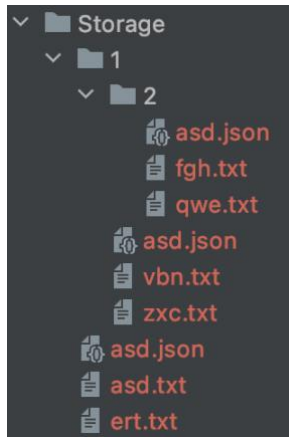
function in_dir(dir) {
  let arr = fs.readdirSync(dir);
  requir(arr, dir);
}

function requir(arr, dir) {
  if (arr.length === 0) return;
  let name = arr.pop();
  if (fs.lstatSync(dir + '/' + name).isDirectory()) {
    in_dir(dir + '/' + name);
  }
  requir(arr, dir);
  output(name, dir);
}

const dir = "/Users/Alex/Documents/NodeJS/TASK 3/4/Storage";
in_dir(dir);
```

Демонстрация работы программы:

Содержимое директории Storage:(подходящие файлы: qwe.txt, vbn.txt, asd.txt)



Результат работы программы:

```
qwe.txt  
vbn.txt  
asd.txt
```

Задание 1.5

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 rl = require("readline-sync");  
  
const n = rl.question("Введите кол-во файлов: ");  
let res_str = "";  
  
for (let i = 0; i < n; i++) {  
    let name = rl.question('№' + (i + 1) + ': ');  
    if (!fs.existsSync(name + ".txt")) {  
        console.log("Файл " + name + ".txt не найден.");  
        i--;  
        continue;  
    }  
    let str = fs.readFileSync(name + ".txt", "utf8").replace(/\n/g, '');  
    res_str += str;  
}  
  
fs.writeFileSync("result.txt", res_str);
```

Демонстрация работы программы:

```
Введите кол-во файлов: 3
№1: qwe
№2: asd
№3: zxc
```

Содержимое файла qwe.txt:

```
We
```

Содержимое файла qwe.txt:

```
hate
```

Содержимое файла qwe.txt:

```
JS!
```

Результат работы программы (содержимое result.txt):

```
We hate JS!
```

Задание 1.6

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 obj = {
  0: undefined
};

let vlog = 0;
while (true) {
  let new_obj = {};
  try {
    JSON.stringify(obj);
  } catch (err) {
    console.log(vlog);
    break;
  }
  vlog++;
  new_obj[vlog] = obj;
  obj = new_obj;
}
```

Результат работы программы:

```
2617
```

Задание 1.7

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");

let jsonstr = fs.readFileSync("input.json", "utf8");
let obj = JSON.parse(jsonstr);
let max = 0;
let pk = undefined;

function req(obj, vloj) {
    for (let key in obj) {
        max = Math.max(max, vloj);
        let loc_max = max;
        req(obj[key], vloj + 1);
        if (vloj === 0 && loc_max < max) pk = key;
    }
}

req(obj, 0);
console.log(JSON.stringify(obj[pk]));
```

Содержимое файла input.json:

```
{
  "a": {
    "a1": 10,
    "a2": {
      "a21": 10,
      "a22": 20
    },
    "a3": true
  },
  "b": {
    "b1": -1.31,
    "b2": true,
    "b3": {
      "b31": {
        "b311": {
          "b3111": 91.31,
          "b3112": [
            -1,
            -2
          ]
        }
      }
    },
    "b32": {
      "b321": [
        10,
        11,
        12,
        13
      ]
    }
  }
}
```

```

        "b322": false
      },
      "b33": 0
    }
  },
  "c": {
    "c1": -1
  }
}

```

Результат работы программы:

```

{"b1":-1.31,"b2":true,"b3":{"b31":{"b311":{"b3111":91.31,"b3112":[-1,-2]}}, "b32":{"b321":[10,11,12,13],"b322":false},"b33":0}}

```

Задание 2.1

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("Server port: ", port);

app.get("/form", 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", 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 res = Math.max(aInt, bInt, cInt);
  const answerJSON = JSON.stringify({result: res});
  response.end(answerJSON);
});

```

Демонстрация работы программы:

```

http://localhost:5015/form?p=form.html

```


localhost

Сравнение трёх чисел

Сравнение трёх чисел

Введите А

10

Введите В

200

Введите С

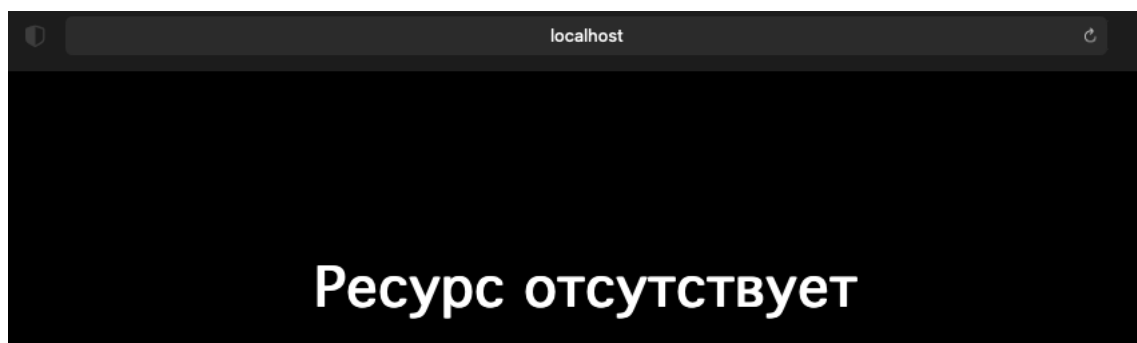
-10

Отправить запрос

```
localhost:5015/compare?a=10&b=200&c=-10  
{"result":200}
```

Запрос несуществующего html:

```
http://localhost:5015/form?p=empty.html
```



Задание 2.2

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("Server port: ", port);

app.get("/form", 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("/choose", function(request, response) {
  let answerJSON = undefined;
  let index = request.query.i;
  if (!fs.existsSync("input.json")) {
    answerJSON = JSON.stringify("Json file not found");
  }
  else {
    let contentJson = fs.readFileSync("input.json", "utf8");
    let content = JSON.parse(contentJson);
    if (index < 0 || index > content.length - 1) {
      answerJSON = JSON.stringify("Invalid index");
    }
    else {
      let res = content[index];
      answerJSON = JSON.stringify({result: res});
    }
  }
  response.end(answerJSON);
});
```

Содержимое файла *input.json*:

```
[
  {
    "a1": 10,
    "a2": {
      "a21": 10,
      "a22": 20
    },
    "a3": true
  },
  {
    "b1": -1.31,
    "b2": true,
    "b3": {
      "b31": {
        "b311": {
          "b3111": 91.31,
          "b3112": [
            -1,
            -2
          ]
        }
      }
    },
    "b32": {
      "b321": [
        10,
        11,
        12,
        13
      ]
    }
  }
]
```

```

    "b322": false
  },
  "b33": 0
}
},
{
  "c1": -1
}
]

```

http://localhost:5015/form?p=form.html

localhost

Содержимое массива по индексу

Введите индекс массива

Отправить запрос

localhost:5015/choose?i=1

```

{"result":{"b1":-1.31,"b2":true,"b3":{"b31":{"b311":{"b3111":91.31,"b3112":[-1,-2]}}, "b32":{"b321":[10,11,12,13],"b322":false}, "b33":0}}}

```

Задание 2.3

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 rl = require("readline-sync")

if (fs.existsSync("template.html")) {
  let temp = fs.readFileSync("template.html", "utf8");
  const act = rl.question("Введите адрес запроса: ");
  temp = temp.replace(/\$action/gi, act.toString());
  const num = rl.question("Введите кол-во названий полей: ");
  let fb = "";
  for (let i = 0; i < parseInt(num.toString()); i++) {
    let name = rl.question("Поле №" + (i + 1) + ": ");
    fb += "<p>\n\t\t<label>\n\t\t\t<span>" + name + "</span> <br>\n\t\t\t<input name=\"" +
name + "\" type=\"text\" " +
    "spellcheck=\"false\" autocomplete=\"off\">\n\t\t\t</label>\n\t\t</p>\n\t\t<br>\n\t";
  }
  temp = temp.replace("$form_body", fb);
  fs.writeFileSync("form.html", temp);
}
else {
  console.log("Нет шаблона html");
}

```

Содержимое шаблона template.html:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Шаблон</title>
  <style>
    body
  </style>
</head>
<body>
<form method="GET" action="$action">
  $form body
  <input type="submit" value="Отправить запрос">
</form>
</body>
</html>
```

Демонстрация работы программы:

```
Введите адрес запроса: /compare
Введите кол-во названий полей: 2
Поле №1: a
Поле №2: b
```

Содержимое результирующего файла form.html:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Шаблон</title>
  <style>
    body
  </style>
</head>
<body>
<form method="GET" action="/compare">
  <p>
    <label>
      <span>a</span> <br>
      <input name="a" type="text" spellcheck="false" autocomplete="off">
    </label>
  </p>
  <br>
  <p>
    <label>
      <span>b</span> <br>
      <input name="b" type="text" spellcheck="false" autocomplete="off">
    </label>
  </p>
  <br>
  <input type="submit" value="Отправить запрос">
</form>
</body>
</html>
```

Задание 2.4

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 express = require("express");

const app = express();
const port = 5015;
app.listen(port);
console.log("Server port: ", port);

app.get("/make_arr", 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 arr = [];
  for (let i = aInt; i <= bInt; i++) {
    if (i % cInt === 0) {
      arr.push(i);
    }
  }
  const answerJSON = JSON.stringify({result: arr});
  response.end(answerJSON);
});
```

Результат работы программы:

```
localhost:5015/make_arr?a=1&b=100&c=3
{"result": [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]}
```

Вывод

В ходе данной лабораторной работы познакомился с форматом JSON, на практике проверил удобность работы с ним, его очень легко парсить, не нужно писать свои функции для этого. Научился создавать сервера, обновил свои знания по созданию html страниц. Изучил Get запросы и научился с ними работать.