Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра экономической информатики

Инженерно-экономический факультет

**ЛАБОРАТОРНАЯ РАБОТА №8**

**“** **Основы программирования на стороне сервера. Введение в Node.js. Работа с файлами с использованием модуля fs.”**

|  |  |
| --- | --- |
| Выполнил: | Студент группы 873603 Дадалко А. В. |
| Проверила: | Старший преподаватель кафедры ЭИ Атрощенко Н. А. |

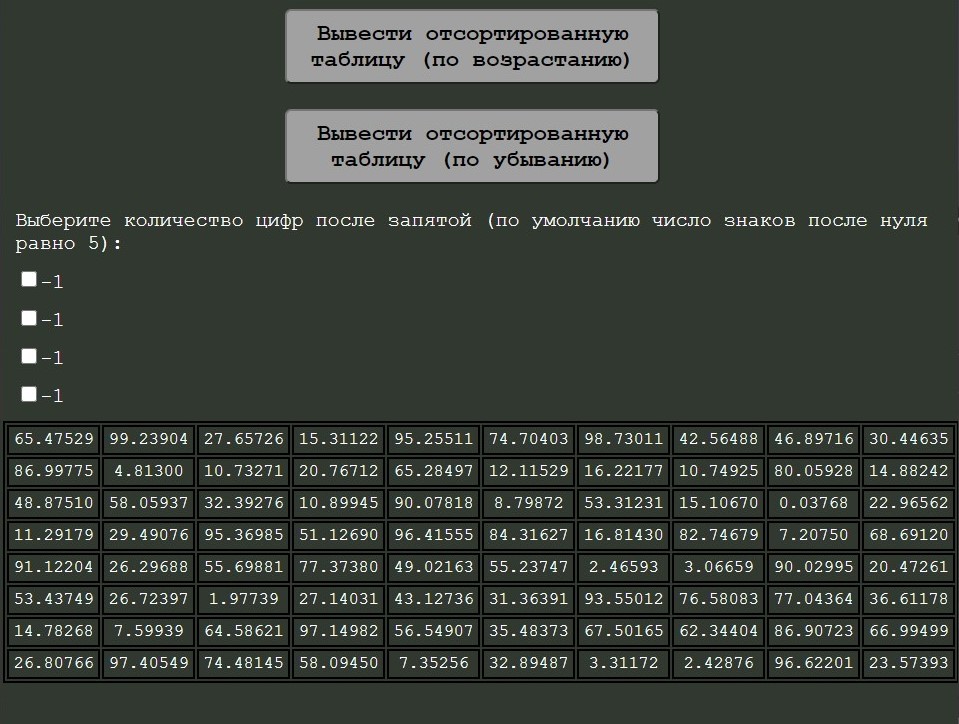
Минск 2020

***Цель работы:*** научиться создавать проекты с использованием сервера Node.js и познакомиться с возможностями файлового модуля fs.

***Индивидуальное задание:*** Создайте массив из 80 дробных десятичных чисел, заполненный случайными значениями от 1 до 10. Все элементы массива разместите в таблице 8x10 на веб-странице. Запишите значения элементов массива до обработки и после обработки в разные файлы. Выведите отсортированный по столбцам массив в таблице по нажатию на соответствующую (по убыванию или возрастанию) кнопку на веб-странице, считав результат из соответствующего файла. Формат вводимых чисел задать с помощью checkbox (количество знаков после запятой). Информация должна считываться с файлов.

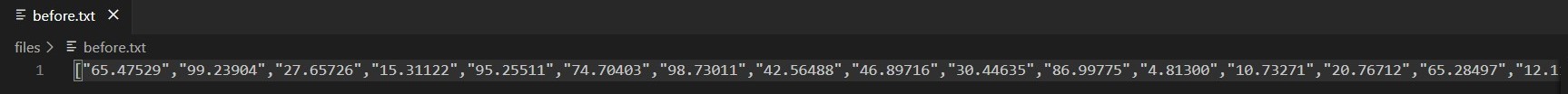
***Выполнение:***

В ходе данной лабораторной работы, в соответствии с вариантом задания, была разработана веб-страница с 2 кнопками, 4 checkbox (для декрементации количества знаков после запятой) и таблицей, содержащей случайные значения в виде дробных десятичных чисел (рисунок 1).



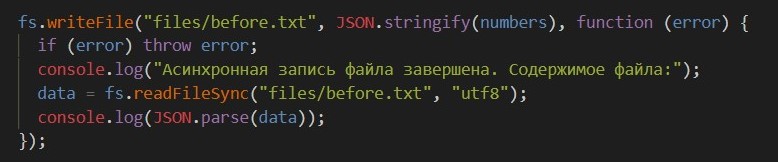
**Рисунок 1 – Созданная веб-страница**

Для генерации случайных значений чисел, был использован метод Math.random()\*100, чтобы выводились числа от 0 до 100. После генерации массива со случайными значениями, происходит запись первоначального массива в текстовый файл под названием “before.txt” (рисунок 2) с помощью модуля fs.

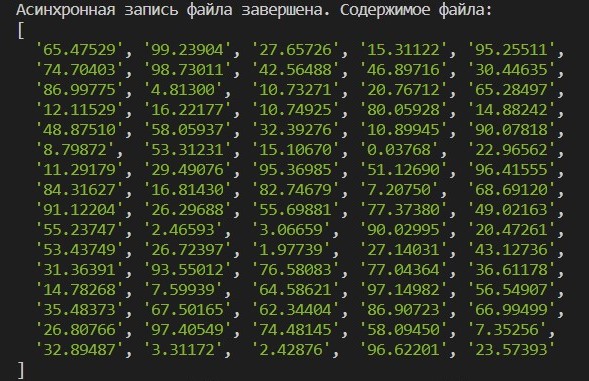


**Рисунок 2 – Текстовый файл “before.txt”**

Модуль fs предоставляет широкие возможности взаимодействия с файлами на стороне сервера. На рисунке 3 представлен фрагмент кода с записью массива в файл “before.txt” и чтением содержимого файла. Для записи был использован метод fs.writeFile() и для чтения – метод fs.readFileSync(). Считанное содержимое файла расположено ниже (рисунок 4).



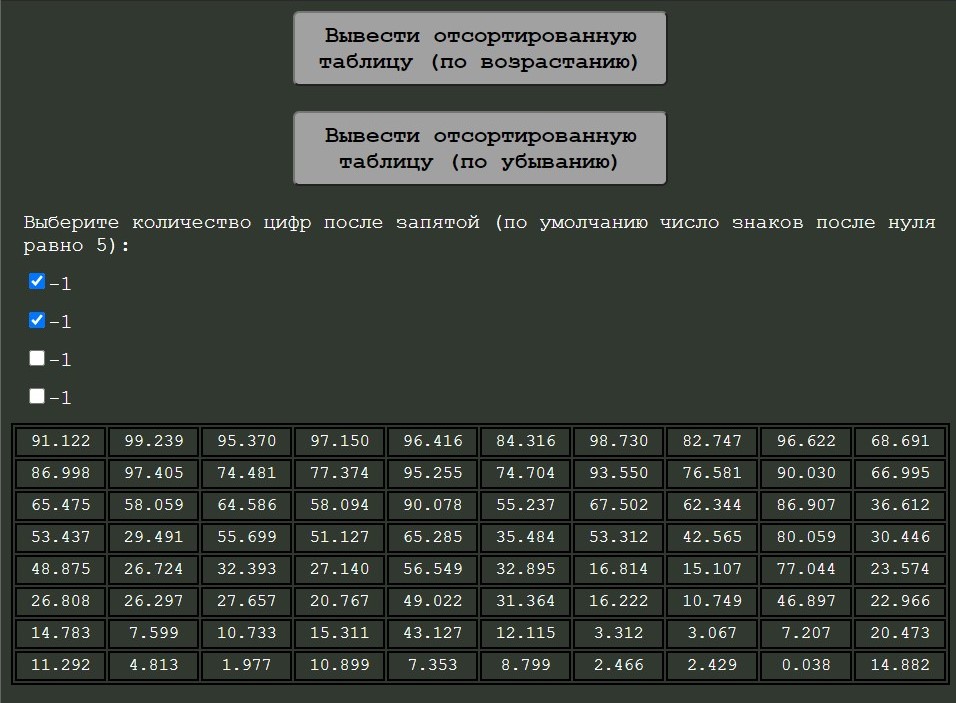
**Рисунок 3 – Запись массива numbers в текстовый файл и чтение файла**

****

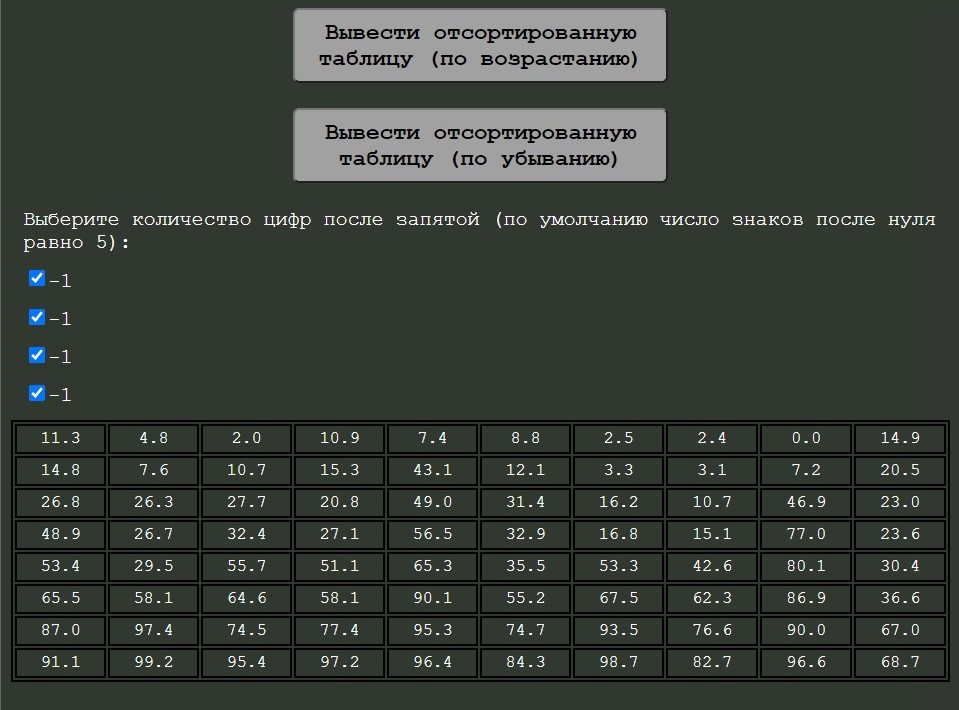
**Рисунок 4 – Содержимое файла**

На веб-странице клиент может подключиться через адресную строку, введя <http://localhost:3000/>. Для создания сервера и прослушивания порта 3000 использовались методы http.createServer() и http.listen() (с указанием значения 3000) соответственно.

По умолчанию, на веб-странице числа выводятся с 5 знаками после запятой. С помощью checkbox можно изменить количество знаков после запятой (число знаков после запятой уменьшается на 1 с каждым отмеченным checkbox). Также, были реализованы 2 кнопки, которые сортируют элементы таблицы по столбцам по возрастанию или убыванию, в зависимости от нажатой кнопки. Два примера работы веб-страницы расположены на рисунках 5 и 6.



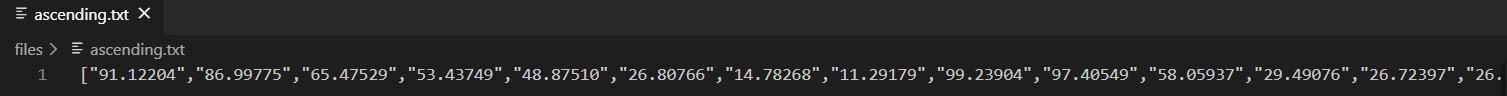
**Рисунок 5 – Результат нажатия кнопки “Вывести отсортированную таблицу (по возрастанию)” с тремя знаками после запятой**

****

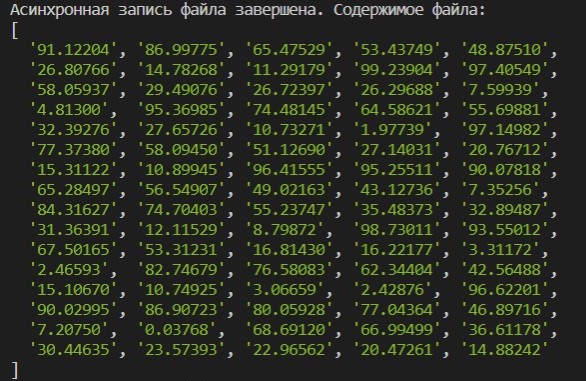
**Рисунок 6 – Результат нажатия кнопки “Вывести отсортированную таблицу (по убыванию)” с одним знаком после запятой**

Массив сортируется по возрастанию и убыванию, а после – записывается в два текстовых файла “ascending.txt” и “descending.txt”. Запись в файлы производится также с помощью метода fs.writeFile() модуля fs.

Содержимое файла с массивом, отсортированным по убыванию, изображено на рисунках 7 и 8.

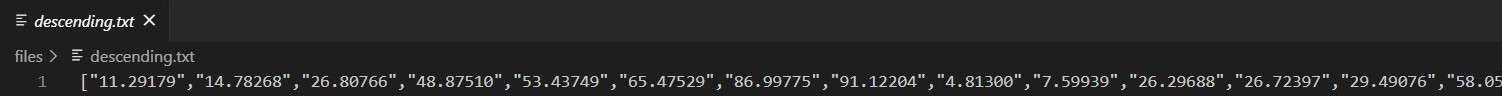


**Рисунок 7 – Содержимое файла “ascending.txt”**

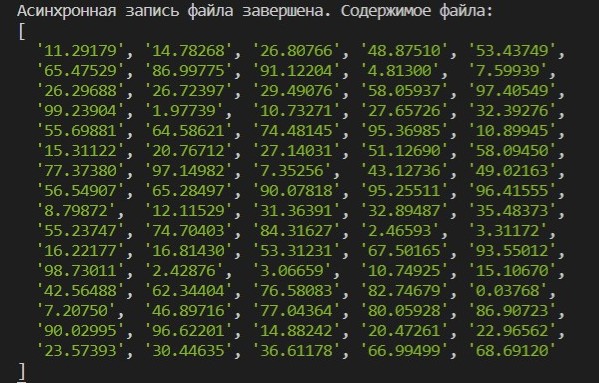


**Рисунок 8 – Содержимое файла “ascending.txt”**

Содержимое файла с массивом, отсортированным по возрастанию, изображено на рисунках 9 и 10.



**Рисунок 9 – Содержимое файла “descending.txt”**

****

**Рисунок 10 – Содержимое файла “descending.txt”**