|  |  |
| --- | --- |
|  | **МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ**  Федеральное государственное БЮДЖЕТНОЕ образовательное учреждение высшего образования  **«российский государственный аграрный университет –**  **МСха имени К.А. Тимирязева»**  **(ФГБОУ ВО ргау - МСХА имени К.А. Тимирязева)** |
|  |  |

Институт экономики и управления АПК

Кафедра прикладной информатики

**Объектно-ориентированный анализ**

**и программирование**

**Отчет по заданиям, Вариант 14**

Выполнил:

студент 4 курса 406 группы

Федотов Александр Михайлович

Проверил: ст. преподаватель

Ермолаева Ольга Сергеевна

**Москва, 2019**

Оглавление

[Введение 4](#_Toc27523344)

[Семинар 1 5](#_Toc27523345)

[1.1 Расчет температуры по Цельсию 6](#_Toc27523346)

[1.2 Временной интервал 7](#_Toc27523347)

[1.3 Индивидуальное задание к первому семинару 8](#_Toc27523348)

[Семинар 2 9](#_Toc27523349)

[2.1 Вычисление значений функции, заданной графически 11](#_Toc27523350)

[2.2 Выстрел по мишени 11](#_Toc27523351)

[2.3 Клавиши курсора 13](#_Toc27523352)

[2.4 Таблица значений функции 14](#_Toc27523353)

[2.5 Вычисление суммы ряда 15](#_Toc27523354)

[2.6 Индивидуальное задание ко второму семинару 16](#_Toc27523355)

[Семинар 3 18](#_Toc27523356)

[3.1 Количество элементов между минимумом и максимумом 20](#_Toc27523357)

[3.2 Сумма элементов правее последнего отрицательного 22](#_Toc27523358)

[3.3 Быстрая сортировка массива 23](#_Toc27523359)

[3.4 Индивидуальная задание к третьему семинару 24](#_Toc27523360)

[Семинар 5 26](#_Toc27523361)

[5.1 Поиск подстроки 28](#_Toc27523362)

[5.2 Подсчет количества вхождений слова в текст 29](#_Toc27523363)

[5.3 Вывод вопросительных предложений 29](#_Toc27523364)

[Семинар 6 31](#_Toc27523365)

[6.1 Поиск в простой базе (массив структур) 32](#_Toc27523366)

[6.2 Сортировка массива структур 34](#_Toc27523367)

[6.3 Структуры и бинарные файлы 36](#_Toc27523368)

[6.4 Структуры в динамической памяти 37](#_Toc27523369)

[6.5 Индивидуальное задание 39](#_Toc27523370)

[Семинар 7 44](#_Toc27523371)

[7.1 Передача в функцию параметров стандартных типов 44](#_Toc27523372)

[8 Семинар 46](#_Toc27523373)

[8.1 Перегрузка функций 46](#_Toc27523374)

[Семинар 9 49](#_Toc27523375)

[9.1 Стек 49](#_Toc27523376)

[Семинар 10 51](#_Toc27523377)

[10.1 Поиск в просто базе (массив объектов) 51](#_Toc27523378)

# Введение

Все семинары выполнялись с использованием языка JavaScript.

Семинары 1-5 представляют собой HTML страницы с простым интерфейсом для проверки работоспособности написанных программ.

Семинары 6-10 представляют собой консольные приложения, написанные с помощью Node.js. Для их запуска и проверки необходим установленный Node.js на ПК, а так же все пакеты, используемые в заданиях.

Установка пакетов происходит с помощью команды, введенной из папки проекта:

$npm i

Для запуска приложения необходимо использовать команду:

$node index.js

Архив с выполненными заданиями можно скачать по ссылке:

<https://www.dropbox.com/s/nj8qxvfqdapeq0e/OOP_FDTV.rar?dl=0>

# Семинар 1

При переносе задания на язык JavaScript был разработана простая HTML страница с кнопочным интерфейсом.

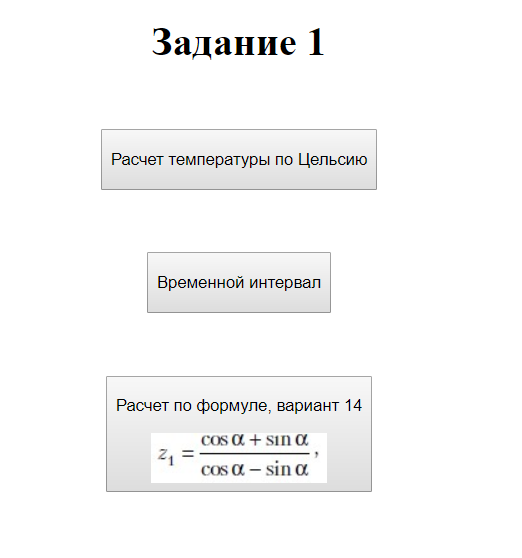


Рис.1 – Кнопочная форма HTML страницы

Листинг кнопочной формы для первого семинара:

<body>

    <div *style*="text-align:center">

        <h1>Задание 1</h1>

        <div *class*="div" *style*="margin-bottom: 50px">

            <button *class*="ex11" *style*="margin-top: 30px;">

                <p>Расчет температуры по Цельсию</p>

            </button>

        </div>

        <div *class*="div" *style*="margin-bottom: 50px">

            <button *class*="ex12">

                <p>Временной интервал</p>

            </button>

        </div>

        <div *class*="div">

            <button *class*="ex13">

                <p>Расчет по формуле, вариант 14</p>

                <img *src*="https://i.ibb.co/Y0Lvndw/1ex.jpg" *alt*="">

            </button>

        </div>

    </div>

# 1.1 Расчет температуры по Цельсию

В рамках задания необходимо написать программу, переводящие значение температуры по Фаренгейту в температуру по Цельсию.

Листинг 1.1

document.querySelector(".ex11").addEventListener("click", function () {

            var fahr = +prompt("Введите температуру по фаренгейту");

            var cels = (fahr - 32) \* 5 / 9;

            alert("По фаренгейту: " + fahr.toFixed(2) + " / По Цельсию: " + cels.toFixed(2));

            console.log("По цельсию: " + fahr.toFixed(2));

            console.log("По фаренгейту: " + cels.toFixed(2));

        });

# 1.2 Временной интервал

Необходимо написать программу, высчитывающую временной интервал между двумя временными точками.

Листинг 1.2

1. document.querySelector(".ex12").addEventListener("click", function () {
2. alert("Введите время начала интервала");
3. var hour1 = +prompt("Введите час старта");
4. var min1 = +prompt("Введите минуту старта");
5. var sec1 = +prompt("Введите секунду старта");
6. alert("Введите время конца интервала");
7. var hour2 = +prompt("Введите час конца");
8. var min2 = +prompt("Введите минуту конца");
9. var sec2 = +prompt("Введите секунду конца");
10. var sum\_sec = (hour2 - hour1) \* 3600 + (min2 - min1) \* 60 + sec2 - sec1;
11. console.log(sum\_sec);
12. var hour = Math.floor(sum\_sec / 3600);
13. var min = Math.floor((sum\_sec - hour \* 3600) / 60);
14. var sec = Math.floor(sum\_sec - hour \* 3600 - min \* 60);
15. alert("Продолжительность промеутка от " + hour1 + ":" + min1 + ":" + sec1 + " до " + hour2 + ":" +
16. min2 + ":" + sec2 + " равна " + hour + ":" + min + ":" + sec);
17. console.log("Продолжительность промеутка от " + hour1 + ":" + min1 + ":" + sec1 + " до " + hour2 +
18. ":" + min2 + ":" + sec2 + " равна " + hour + ":" + min + ":" + sec)
19. });

# 1.3 Индивидуальное задание к первому семинару

Необходимо написать программу, высчитывающую значения по данной формуле:



Листинг 1.3

        document.querySelector(".ex13").addEventListener("click", function () {

            alert("Введите значение альфа");

            var alpha = +prompt("Введите значение альфа");

            var z1 = (Math.cos(alpha) + Math.sin(alpha)) / (Math.cos(alpha) - Math.sin(alpha));

            var z2 = (Math.tan(alpha \* 2) + Math.sin(alpha \* 2));

            alert("Результат расчета: z1 = " + z1.toFixed(2) + "/ z2 = " + z2.toFixed(2));

        })

# Семинар 2

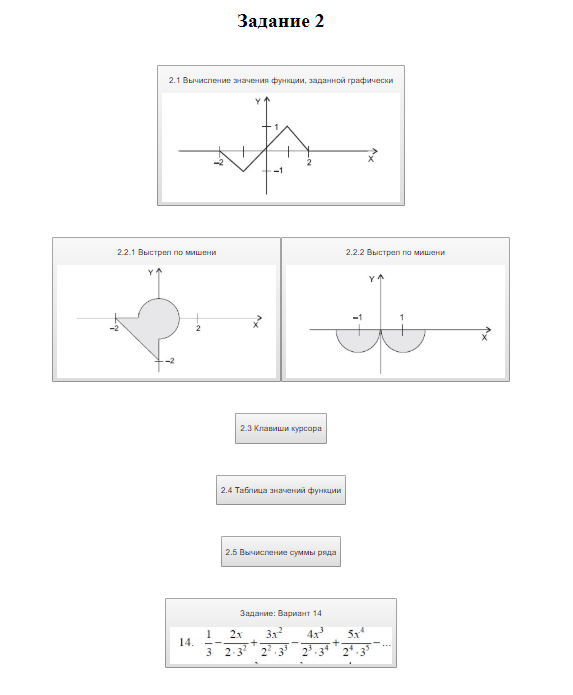
****

Рис.2 – Кнопочная форма HTML страницы

Листинг кнопочной формы для второго семинара

<div *style*="text-align:center">

        <h1>Задание 2</h1>

        <div *class*="div" *style*="margin-bottom: 50px">

            <button *class*="ex21" *style*="margin-top: 30px;">

                <p>2.1 Вычисление значения функции, заданной графически</p>

                <img *src*="https://i.ibb.co/c2vxKtF/2ex.jpg" *alt*="2ex" border="0">

            </button>

        </div>

        <div *class*="div" *style*="margin-bottom: 50px; display: flex; justify-content: center;">

            <button *class*="ex22">

                <p>2.2.1 Выстрел по мишени</p>

                <img *src*="https://i.ibb.co/Tk2qrCG/2-2ex.jpg" *alt*="2-2ex" border="0" *width*="350px" *height*="180px">

            </button>

            <button *class*="ex222">

                <p>2.2.2 Выстрел по мишени</p>

                <img *src*="https://i.ibb.co/QpPQSZQ/3ex.jpg" *alt*="3ex" border="0" *width*="350px" *height*="180px">

            </button>

        </div>

        <div *class*="div">

            <button *class*="ex23" *style*="margin-bottom: 50px">

                <p>2.3 Клавиши курсора</p>

            </button>

        </div>

        <div *class*="div">

            <button *class*="ex24" *style*="margin-bottom: 50px">

                <p>2.4 Таблица значений функции</p>

            </button>

        </div>

        <div *class*="div">

            <button *class*="ex25" *style*="margin-bottom: 50px">

                <p>2.5 Вычисление суммы ряда</p>

            </button>

        </div>

        <div *class*="div">

            <button *class*="ex26" *style*="margin-bottom: 50px">

                <p>Задание: Вариант 14</p>

                <img *src*="https://i.ibb.co/RBWQCbS/2ex6.jpg" *alt*="2ex6" border="0">

            </button>

        </div>

    </div>

# 2.1 Вычисление значений функции, заданной графически

Необходимо написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.

(График из задания см. в кнопочной форме)

Листинг 2.1

document.querySelector(".ex21").addEventListener("click", function () {

            var y;

            var x = +prompt("Введите значение аргумента");

*//условия*

            (x < -2) ? y = 0:

                (x >= -2 && x < -1) ? y = -x - 2 :

                (x >= -1 && x < 1) ? y = x :

                (x >= 1 && x < 2) ? y = -x + 2 :

                (x >= 2) ? y = 0 :

                y = 0;

            alert("Для x = " + x + " значение функции y = " + y);

        });

# 2.2 Выстрел по мишени

Необходимо написать программу, определяющую, попадает ли точка в область.

(Область на графике см. в кнопочной форме)

Листинг 2.2

document.querySelector(".ex22").addEventListener("click", function () {

            var x = +prompt("Введите значение x");

            var y = +prompt("Введите значение y");

*if* (x \* x + y \* y <= 1 || x <= 0 && y <= 0 && y >= -x - 2) {

                alert("Точка попадает в цель");

            } *else* {

                alert("Точка не попадает в цель")

            }

        });

        document.querySelector(".ex222").addEventListener("click", function () {

            var x = +prompt("Введите значение x");

            var y = +prompt("Введите значение y");

*if* (y < 0 && ((x - 1) \* (x - 1) + y \* y <= 1 ||

                    (x + 1) \* (x + 1) + y \* y <= 1)) {

                alert("Точка попадает в цель");

            } *else* {

                alert("Точка не попадает в цель")

            }

        });

# 2.3 Клавиши курсора

Необходимо написать программу, определяющую, какая из курсорных клавиш была нажата

В рамках этого задания была добавлена дополнительная верстка для кнопки button с классом ex23 в состоянии focus. Когда кнопка находится в фокусе у пользователя, она подсвечивается зеленым цветом.

 <style>

*.ex23:focus* {

            background-color: white;

            border: 3px solid green;

            box-shadow: -4px 0px 73px 0px rgba(3, 255, 129, 1);

            transition: all 0.5s;

        }

    </style>

Листинг 2.3

document.querySelector(".ex23").addEventListener("keydown", function (event) {

            var directions = {

                37: "Стрелка влево",

                38: "Стрелка вверх",

                39: "Стрелка вправо",

                40: "Стрелка вниз"

            };

            var keyC = directions[event.keyCode];

*if* (keyC !== undefined) alert("Была нажата: " + keyC + ". Нажмите enter для продолжения.");

        });

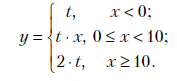
        document.querySelector(".ex23").addEventListener("click", function () {

            alert("Закройте это окно и нажмите стрелочную кнопку.");

        });

# 2.4 Таблица значений функции

Необходимо написать программу печати таблицы значений функции.



Листинг 2.4

document.querySelector(".ex24").addEventListener("click", function () {

            var Xn = +prompt("Введите значение Xn");

            var Xk = +prompt("Введите значение Xk");

            var dX = +prompt("Введите значение Xn");

            var t = +prompt("Введите значение t");

            alert("Результат выполнения операции см. в консоли (клавиша F12 -> console)");

            console.log("----------------------------");

            console.log("|      X      |      Y      |");

            console.log("----------------------------");

            var x = Xn;

*while* (x <= Xk) {

*if* (x < 0) y = t;

*if* (x >= 0 && x < 10) y = t \* x;

*if* (x >= 10) y = 2 \* t;

*if* (t > 100) {

                    console.log("|      " + x + "      " + "|      " + y + "      |");

                } *else* {

                    console.log("|      " + x + "      " + "|      " + y + "      |");

                }

                x += dX;

                console.log("----------------------------");

            }

        });

# 2.5 Вычисление суммы ряда

Необходимо написать программу вычисления значения функции sin x с точностью ε по формуле:



Листинг 2.5

document.querySelector(".ex25").addEventListener("click", function () {

            const maxIter = 500;

            var x = +prompt("Введите аргумент");

            var eps = +prompt("Введите точность");

            var done = true;

            var ch = x;

            var y = ch;

*for* (var i = 0; ch > eps; i++) {

                ch \*= x \* x / ((2 \* i + 2) \* (2 \* i + 3));

                y += ch;

*if* (i > maxIter) {

                    alert("Ряд расходится!");

*return* done = false;

                }

            }

*if* (done) alert("Значение функции: " + y + " для x = " + x);

        });

# 2.6 Индивидуальное задание ко второму семинару

Для выполнения индивидуального задания необходимо объединить код заданий 2.4 и 2.5, а так же найти собственную формулу шага для индивидуального ряда.



Формула шага для данного ряда выглядит следующем образом:

ch \*= 1 \* Math.pow(x, i) / Math.pow(2, i) / Math.pow(3, i + 1);

Что бы это проверить, можно подставить значения счетчика i и убедиться, что шаг найден верно.

Листинг 2.6

 document.querySelector(".ex26").addEventListener("click", function () {

            const maxIter = 500;

            var x = +prompt("Введите аргумент");

            var eps = +prompt("Введите точность");

            var done = true;

            var ch = x;

            var y = ch;

            console.log("----------------------------");

            console.log("|      X      |      Y      |");

            console.log("----------------------------");

*for* (var i = 0; ch > eps; i++) {

                ch \*= 1 \* Math.pow(x, i) / Math.pow(2, i) / Math.pow(3, i + 1);

*if* (!(i % 2)) y -= ch;

*// Если счетчик i делится на 2 без остатка (каждая четная итерация)*

*else* y += ch;

*//в любом другом случае (каждая нечетная итерация)*

                console.log("|      " + x + "      " + "|      " + y + "      |")

*if* (i > maxIter) {

                    alert("Ряд расходится!");

*return* done = false;

                }

            }

*if* (done) alert("Значение функции: " + y + " для x = " + x);

            console.log("|      " + x + "      " + "|      " + y + "      |");

        })

# Семинар 3

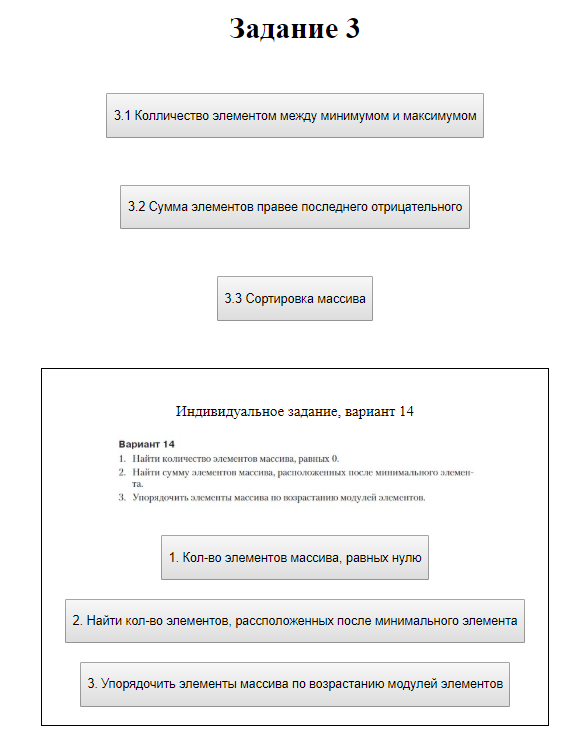


Рис.3 – Кнопочная форма HTML страницы

Листинг кнопочной формы для третьего семинара

<div *style*="text-align:center">

        <h1>Задание 3</h1>

        <div *class*="div" *style*="margin-bottom: 50px">

            <button *class*="ex31" *style*="margin-top: 30px;">

                <p>3.1 Колличество элементом между минимумом и максимумом</p>

            </button>

        </div>

        <div *style*="margin-bottom: 50px">

            <button *class*="ex32">

                <p>3.2 Сумма элементов правее последнего отрицательного</p>

            </button>

        </div>

        <div>

            <button *class*="ex33" *style*="margin-bottom: 50px">

                <p>3.3 Cортировка массива</p>

            </button>

        </div>

        <div>

            <div *class*="ex34" *style*="border: 1px solid black; width: 500px; margin: auto; padding: 20px;">

                <p>Индивидуальное задание, вариант 14</p>

                <img *src*="https://i.ibb.co/JyRW30S/3-4ex.jpg" *alt*="3-4ex" border="0" *width*="400"

*style*="margin-bottom:20px">

                <button *class*="ex341" *style*="margin-bottom: 20px">

                    <p>1. Кол-во элементов массива, равных нулю</p>

                </button>

                <button *class*="ex342" *style*="margin-bottom: 20px">

                    <p>2. Найти кол-во элементов, рассположенных после минимального элемента</p>

                </button>

                <button *class*="ex343">

                    <p>3. Упорядочить элементы массива по возрастанию модулей элементов</p>

                </button>

                </d>

            </div>

        </div>

    </div>

    </div>

# 3.1 Количество элементов между минимумом и максимумом

Необходимо написать программу, которая для целочисленного массива из n элементов определяет, сколько положительных элементов располагается между его максимальным и минимальным элементом

Листинг 3.1

*//3.1*

        document.querySelector(".ex31").addEventListener("click", function () {

*// var a = [1, 3, -5, 1, -2, 1, -1, 3, 8, 4];*

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var a = [];

*for* (var i = 0; i < numberOfElements; i++) {

                a[i] = +prompt("Введите значение элемента массива");

            };

            console.log(a);

*//Объявление переменных*

            var iMax = 0;

            var iMin = 0;

            var iPos = 0;

            var indexiMax = 0;

            var indexiMin = 0;

            var ibeg = 0;

            var iend = 0;

*//Находим iMax и iMin*

*for* (var i = 0; i < a.length; i++) {

*if* (a[i] > iMax) {

                    iMax = a[i];

                    indexiMax = i;

                }

*if* (a[i] < iMin) {

                    iMin = a[i];

                    indexiMin = i;

                }

            }

*//Находим ibeg и iend*

*if* (indexiMax > indexiMin) {

                ibeg = indexiMin

                iend = indexiMax

            } *else* {

                ibeg = indexiMax

                iend = indexiMin

            }

*//Находим кол-во положительных значений между ibeg и iend*

*//Т.к наш диапазон значений находится между элементами ibeg и iend,*

*//необходимо начинать цикл с индекса i = ibeg + 1, а заканчивать*

*//при условии i < iMax*

*for* (var i = ibeg + 1; i < iMax; i++) {

*if* (a[i] > 0) iPos++;

            }

            console.log("max = " + iMax + "/ min = " + iMin + "/ ibeg = " + ibeg + " / iend = " + iend);

            console.log("Кол-во положительных - " + iPos);

        });

# 3.2 Сумма элементов правее последнего отрицательного

Необходимо написать программу, которая для вещественного массива из n элементов определяет сумму его элементов, расположенных правее последнего отрицательного элемента.

Листинг 3.2

*/3.2*

        document.querySelector(".ex32").addEventListener("click", function () {

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var arr = [];

*for* (var i = 0; i < numberOfElements; i++) {

                arr[i] = +prompt("Введите значение элемента массива");

            };

            console.log(arr);

            console.log(arr.length)

*//Нахождение суммы элементов правее последнего отрицательного*

            var sumElem = 0;

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

                sumElem += arr[i];

*if* (arr[i] < 0) {

                    sumElem = 0

                }

            }

            console.log(sumElem)

        });

# 3.3 Быстрая сортировка массива

Необходимо написать программу, которая упорядочивает вещественный массив по возрастанию значений его элементов.

Листинг 3.3

*//3.3*

        document.querySelector(".ex33").addEventListener("click", function () {

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var arr = [];

*for* (var i = 0; i < numberOfElements; i++) {

                arr[i] = +prompt("Введите значение элемента массива");

            };

            console.log(arr);

*//Сортировка массива*

*for* (var i = 0; i <= arr.length - 2; i++) {

                var minValue = arr[i];

*for* (var j = i + 1; j <= arr.length - 1; j++) {

*if* (Math.abs(arr[j]) < Math.abs(minValue)) {

                        minValue = arr[j];

                        var swap = arr[i];

                        arr[i] = minValue;

                        arr[j] = swap;

                    }

                }

            }

            alert("Отсортированный массив: " + arr);

        });

# 3.4 Индивидуальная задание к третьему семинару

Необходимо:

1. Написать программу, находящую количество элементов массива, равных 0
2. Найти сумму элементов массива, расположенных после минимального элемента
3. Упорядочить элементы массива по возрастанию модулей элементов

Листинг 3.4.1

*//3.4.1*

        document.querySelector(".ex341").addEventListener("click", function () {

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var arr = [];

*for* (var i = 0; i < numberOfElements; i++) {

                arr[i] = +prompt("Введите значение элемента массива");

            };

            console.log(arr);

*//Нахождение элементов массива, равных нулю*

            var nullElem = 0;

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

*if* (arr[i] === 0) nullElem++;

            }

            alert("Кол-во элементом в массиве, равных нулю: " + nullElem);

            console.log("Кол-во элементом в массиве, равных нулю: " + nullElem);

        })

Листинг 3.4.2

*//3.4.2*

        document.querySelector(".ex342").addEventListener("click", function () {

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var arr = [];

*for* (var i = 0; i < numberOfElements; i++) {

                arr[i] = +prompt("Введите значение элемент а массива");

            };

            console.log(arr);

*//Нахождение минимального элемента массива*

            var minElem = arr[0];

            var minElemIndex = 0;

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

*if* (arr[i] < minElem) {

                    minElem = arr[i];

                    minElemIndex = i;

                }

            }

            alert("Кол-во элементов, рассположенных после минимального элемента: " + ((arr.length - 1) -

                minElemIndex));

            console.log("Кол-во элементов, рассположенных после минимального элемента: " + ((arr.length - 1) -

                minElemIndex));

        })

Листинг 3.4.3

*//3.4.3*

        document.querySelector(".ex343").addEventListener("click", function () {

*//Ввод элементов массива*

            var numberOfElements = +prompt("Введите кол-во элементов массива");

            var arr = [];

*for* (var i = 0; i < numberOfElements; i++) {

                arr[i] = +prompt("Введите значение элемент а массива");

            };

            console.log("Массив: " + arr);

*//Сортировка*

*for* (var i = 0; i <= arr.length - 2; i++) {

                var minValue = arr[i];

*for* (var j = i + 1; j <= arr.length - 1; j++) {

*if* (Math.abs(arr[j]) < Math.abs(minValue)) {

                        minValue = arr[j];

                        var swap = arr[i];

                        arr[i] = minValue;

                        arr[j] = swap;

                    }

                }

            }

            alert("Массив после сортировки модулей элементов по возрастанию: " + arr);

            console.log("Массив после сортировки модулей элементов по возрастанию: " + arr)

        })

# Семинар 5

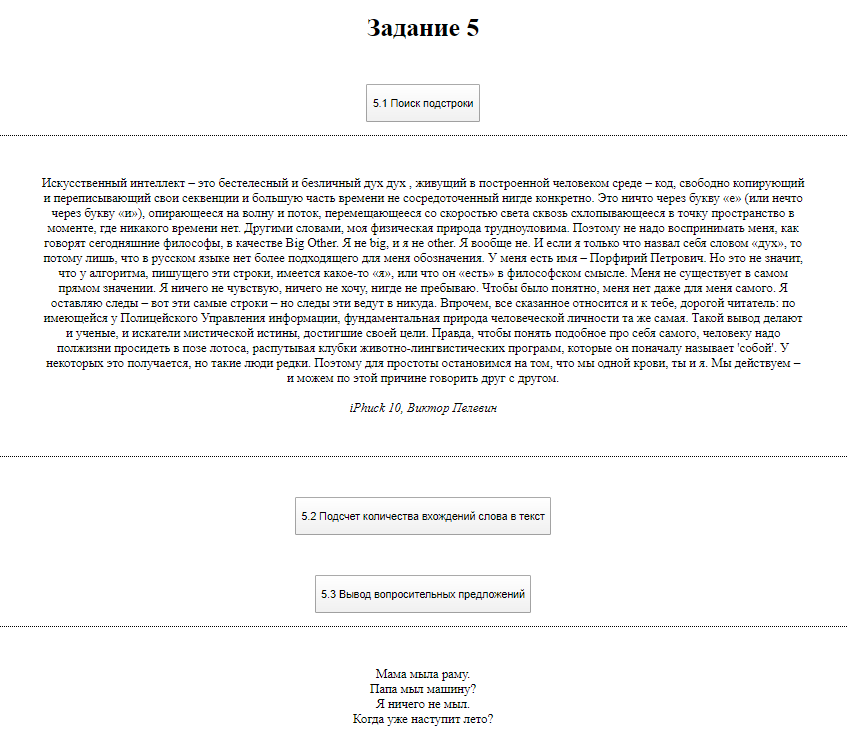


Рис.3 – Кнопочная форма HTML страницы

Листинг кнопочной формы для 5 семинара:

 <div *style*="text-align:center">

        <h1>Задание 5</h1>

        <div *class*="div" *style*="margin-bottom: 50px">

            <button *class*="ex51" *style*="margin-top: 30px;">

                <p>5.1 Поиск подстроки</p>

            </button>

            <p *style*="padding: 50px 30%; border-top: 1px dotted black; border-bottom: 1px dotted black"

*class*="ex51\_text\_search">Искусственный интеллект – это

                бестелесный и безличный дух дух , живущий в построенной человеком среде – код,

                свободно копирующий и переписывающий свои секвенции и большую часть времени не сосредоточенный нигде

                конкретно. Это ничто через букву «е» (или нечто через букву «и»), опирающееся на волну и поток,

                перемещающееся со скоростью света сквозь схлопывающееся в точку пространство в моменте, где никакого

                времени нет. Другими словами, моя физическая природа трудноуловима.

                Поэтому не надо воспринимать меня, как говорят сегодняшние философы, в качестве Big Other. Я не big, и я

                не other. Я вообще не. И если я только что назвал себя словом «дух», то потому лишь, что в русском языке

                нет более подходящего для меня обозначения.

                У меня есть имя – Порфирий Петрович. Но это не значит, что у алгоритма, пишущего эти строки, имеется

                какое-то «я», или что он «есть» в философском смысле. Меня не существует в самом прямом значении. Я

                ничего не чувствую, ничего не хочу, нигде не пребываю. Чтобы было понятно, меня нет даже для меня

                самого. Я оставляю следы – вот эти самые строки – но следы эти ведут в никуда.

                Впрочем, все сказанное относится и к тебе, дорогой читатель: по имеющейся у Полицейского Управления

                информации, фундаментальная природа человеческой личности та же самая. Такой вывод делают и ученые, и

                искатели мистической истины, достигшие своей цели.

                Правда, чтобы понять подобное про себя самого, человеку надо полжизни просидеть в позе лотоса,

                распутывая клубки животно-лингвистических программ, которые он поначалу называет 'собой'. У некоторых

                это получается, но такие люди редки. Поэтому для простоты остановимся на том, что мы одной крови, ты и

                я. Мы действуем – и можем по этой причине говорить друг с другом. <br> <br> <i>iPhuck 10, Виктор

                    Пелевин</i></p>

        </div>

        <div *style*="margin-bottom: 50px">

            <button *class*="ex52">

                <p>5.2 Подсчет количества вхождений слова

                    в текст</p>

            </button>

        </div>

        <div>

            <button *class*="ex53">

                <p>5.3 Вывод вопросительных предложений</p>

            </button>

            <p *style*="padding: 50px;border-top: 1px dotted black; border-bottom: 1px dotted black"

*class*="ex53\_text\_search">Мама мыла раму.<br> Папа мыл машину?<br> Я ничего не мыл. <br>Когда уже наступит лето?</p>

        </div>

    </div>

    </div>

# 5.1 Поиск подстроки

Необходимо написать программу, которая определяет, встречается ли в заданном текстовом файле заданная последовательность символов. Длина строки текста не превышает 80 символов, текст не содержит переносов слов, последовательность не содержит пробельных символов.

Т. к. задания выполняются на нативном JS, у которого нет доступа к файловой системе, поиск будет осуществляться в одном из элементов HTML страницы. При этом было решено осуществлять поиск в полноценном тексте с пробелами и словами.

Работа с файловой системой с помощью JS будет рассмотрена в следующих заданиях.

Листинг 5.1

document.querySelector(".ex51").addEventListener("click", function () {

            var word\_for\_search = prompt("Введи текст для поиска");

            var text = document.querySelector(".ex51\_text\_search");

*//Для проверки используем метод indexOf()*

*if* (text.textContent.indexOf(word\_for\_search) === -1) {

                alert("Совпадений нет");

            } *else* {

                alert("Совпадения есть");

            }

        })

# 5.2 Подсчет количества вхождений слова в текст

Необходимо написать программу, которая определяет, сколько раз встретилось заданное слово в текстовом файле.

Аналогично с предыдущим пунктом, вместо текстового файла будет использоваться текст, написанный в HTML элементе.

Листинг 5.2

 document.querySelector(".ex52").addEventListener("click", function () {

            var word\_for\_search = prompt("Введи текст для поиска");

            var text = document.querySelector(".ex51\_text\_search");

            var index\_reps = 0;

*//Для проверки используем метод indexOf()*

            var count\_reps = 0;

*//Цикл while будет искать вхождения, пока не вернет результат -1, что означает, что вхождений больше нет*

*// Каждую новую итерацию метод indexOf() будет искать вхождение начиная с индекса последнего значения + 1*

*while* (index\_reps !== -1) {

                var index\_reps = text.textContent.indexOf(word\_for\_search, index\_reps + 1);

*if* (index\_reps !== -1) {

                    console.log("Индекс вхождения: " + index\_reps);

                    count\_reps++;

                };

            }

            console.log("Кол-во вхождений: " + count\_reps);

        });

# 5.3 Вывод вопросительных предложений

Необходимо написать программу, которая считывает текст и выводит на экран только вопросительные предложения из текста

Листинг 5.3

document.querySelector(".ex53").addEventListener("click", function () {

            var text = document.querySelector(".ex53\_text\_search");

            console.log(text.textContent)

            var j = 0;

            var i = 0;

*while* (text.textContent.indexOf("?", j) !== -1) {

*if* (text.textContent.indexOf("?", j) < text.textContent.indexOf(".", j)) {

                    console.log("Вопросительное предложение: " + text.textContent.slice(0, text.textContent

                        .indexOf("?", j) + 1));

                    j = text.textContent.indexOf("?", j) + 1;

                } *else* *if* (text.textContent.indexOf("?", j) > text.textContent.indexOf(".", j)) {

                    console.log("Вопросительное предложение: " + text.textContent.slice(text.textContent

                        .indexOf(".", j) + 2, text.textContent.indexOf("?", j) + 1));

                    j = text.textContent.indexOf("?", j) + 1;

                }

            }

        });

# Семинар 6

Т.к. начиная с данного семинара будет проблематично реализовывать логику консольных приложений на JS без доступа к файловой системе, в дальнейшем для выполнения задания будет использоваться программная платформа Node.js.

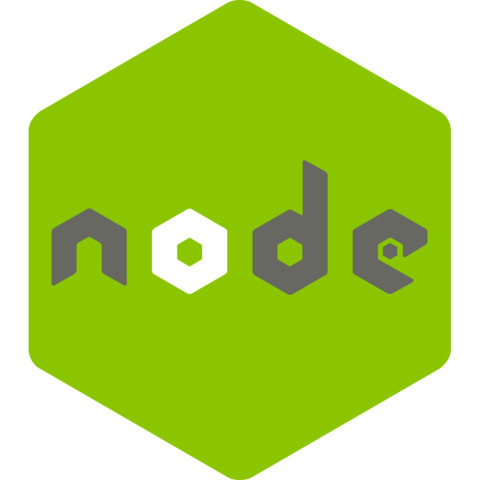


Рис.4 – Node.js

**Node.js** - программная платформа, основанная на движке V8 (транслирующем JS в машинный код) и превращая JS из узко специализированного языка в язык общего назначения.

Node.js позволяет JS взаимодействовать с **уcтройствами ввода/вывода** через свой API, подключать внешние библиотеки, написанные на разных языках, обеспечивающие вызовы к ним из JS кода.

В основе Node.js лежит событийно-ориентированное и асинхронное программирование с неблокирующим вводом/выводом.

База данных, необходимая для выполнения данного семинара находится в папке “/db” в корне проекта и называется “dbase.txt”.

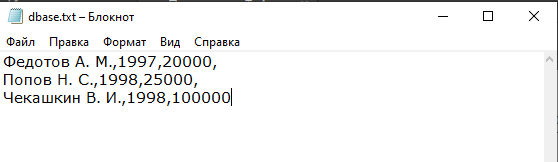


Рис.5 – dbase.txt

# 6.1 Поиск в простой базе (массив структур)

Необходимо написать программу, которая по заданной фамилии выводит на экран сведения о сотруднике, подсчитывая средний оклад всех запрошенных сотрудников.

При выполнении задания были использованы следующие пакеты node.js:

* FileSystem
* ReadLine
* Colors

Пакет FileSystem используется для доступа к файловой системе и возможности чтения и записи данных на ПК. Пакет ReadLine используется для получения доступа к устройствам ввода. Пакет Colors используется для изменения цвета выводимой в консоль информации

Листинг 6.1

var fs = require("fs");

var readline = require("readline");

var colors = require("colors");

var db = "";

var rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

        fs.readFile("./db/dbase.txt.txt", (err, data) => {

            var dbTemp = data.toString();

            var db = dbTemp.split(",");

*for* (var i = 0; i < db.length; i++) {

                db[i] = db[i].replace(/\r?\n/g, "");

            }

            var sumSalary = 0

            var count = 0;

            var middleSalary = 0;

            var successfulSearch = false;

            var continuedInput = true;

            console.log("Введите имя сотрудника:");

            rl.on("line", (line) => {

*for* (var i = 0; i < db.length; i++) {

*if* (line == db[i]) {

*if* (isNaN(+db[i])) {

                            successfulSearch = true;

                            console.log("=================================".red);

                            console.log("Имя и инициалы: " + db[i].green);

                            console.log("Год рождения: " + db[i + 1].green);

                            console.log("Оклад в рублях: " + db[i + 2].green);

                            count++;

                            sumSalary += +db[i + 2];

                            middleSalary = sumSalary / count;

                            console.log(("Средний оклад запрошенных сотрудников: " + middleSalary).yellow);

                            console.log("=================================".red);

                        }

                    }

                }

*if* (successfulSearch === false) {

                    console.log("Пользователь не найден".red);

                }

                successfulSearch = false;

            });

        });

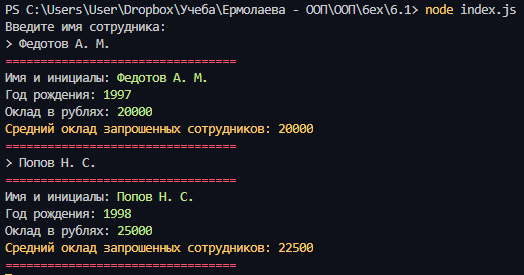
    } *else* {

        console.error("Файл dbase.txt не найден.");

    }

});

Результат выполнения программы в консоли:



# 6.2 Сортировка массива структур

Необходимо написать программу, которая упорядочивает описанный в предыдущей задаче файл по году рождения сотрудников.

В базу данных dbase.txt были добавлены новые сотрудники:

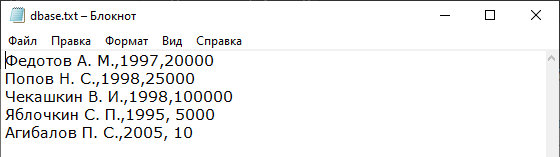


Рис.6 – dbase.txt

Листинг 6.2

var fs = require("fs");

var readline = require("readline");

var colors = require("colors");

var db = [];

var rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

        fs.readFile("./db/dbase.txt.txt", (err, data) => {

            var dbTemp = data.toString();

            var dbTemp2 = dbTemp.split("\r\n"); *//каждый ползователь помещен в массив*

*for* (var i = 0; i < dbTemp2.length; i++) {

                db.push(dbTemp2[i].split(","));

            };

            console.log(db);

*//Сортировка*

*for* (var i = 0; i <= db.length - 2; i++) {

                var minValue = db[i][1];

*for* (var j = i + 1; j <= db.length - 1; j++) {

*if* (db[j][1] < minValue) {

                        minValue = db[j][1];

                        var swap = db[i];

                        db[i] = db[j];

                        db[j] = swap;

                    }

                }

            }

            console.log("Отсортированный список: ".green);

*for* (var i = 0; i < db.length; i++) {

                console.log("");

                console.log("Имя: " + db[i][0].red);

                console.log("Год рождения: " + db[i][1].red);

                console.log("Заработная плата в рублях: " + db[i][2].red);

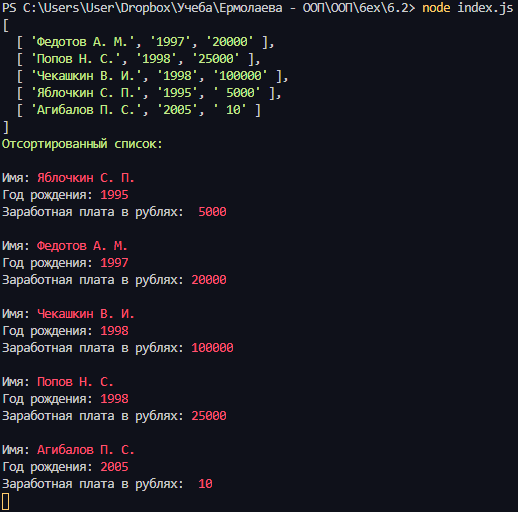
            }

        });

    }

});

Результат выполнения программы в консоли:



# 6.3 Структуры и бинарные файлы

Необходимо написать программу, которая считывает информацию из файла, формат которого описан в задаче 6.1 и записывает ее в бинарный файл.

Листинг 6.3

var fs = require("fs");

var readline = require("readline");

var colors = require("colors");

var db = [];

var rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

        fs.readFile("./db/dbase.txt.txt", (err, data) => {

            var db = data.toString();

            var binaryDb = "";

*//Форматирование в бинарный код*

*for* (var i = 0; i < db.length; i++) {

                binaryDb += db[i].charCodeAt(0).toString(2) + " ";

            };

*//Запись в файл*

            fs.writeFileSync("./db/db.bin", binaryDb);

        });

    }

});

В результате выполнения программа создает файл db.bin в корне проекта

# 6.4 Структуры в динамической памяти

Необходимо написать программу, выводящую на экран содержимое бинарного файла, сформированного в предыдущем задании, упорядочив фамилии сотрудников по алфавиту.

Листинг 6.4

var fs = require("fs");

var readline = require("readline");

var colors = require("colors");

var rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

        fs.readFile("./db/db.bin", (err, data) => {

            var dbBuf = data.toString(); *//Преобразование из буфера в string*

            var dbArr = dbBuf.split(" "); *//Преобразование из string в arr*

            var dbUni = [];

            var dbStr = "";

            var dbTemp = [];

            var db = [];

*//Преобразование из двоичного когда в юникод*

*for* (var i = 0; i < dbArr.length; i++) {

                dbUni[i] = parseInt(dbArr[i], 2);

            };

*//Преобразование из юникода в текст*

*for* (var i = 0; i < dbUni.length; i++) {

                dbStr += String.fromCharCode(dbUni[i]);

            };

            dbTemp = dbStr.split("\r\n");

            console.log(dbTemp);

*for* (var i = 0; i < dbTemp.length; i++) {

                db.push(dbTemp[i].split(","));

            };

*for* (var i = 0; i <= db.length - 2; i++) {

                var minValue = db[i][0];

*for* (var j = i + 1; j <= db.length - 1; j++) {

*if* (db[j][0] < minValue) {

                        minValue = db[j][0];

                        var swap = db[i][0];

                        db[i][0] = minValue;

                        db[j][0] = swap;

                    }

                }

            }

            console.log("");

            console.log("Отсортированный список: ".green);

*for* (var i = 0; i < db.length; i++) {

                console.log("");

                console.log("Имя: " + db[i][0].red);

                console.log("Год рождения: " + db[i][1].red);

                console.log("Заработная плата в рублях: " + db[i][2].red);

            }

        });

    }

});

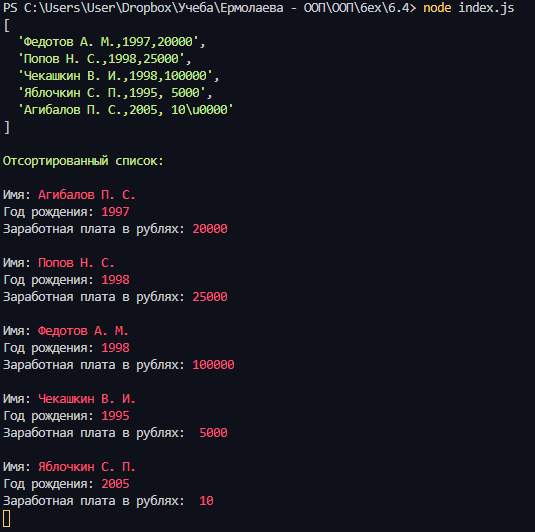
*// var x = 10000100100; // Код буквы из бинарного файла*

*// var xUni = parseInt(x, 2);*

*// var xString = String.fromCharCode(xUni);*

*// console.log(xString); // "Ф"*

Результат выполнения программы в консоли:



# 6.5 Индивидуальное задание

Необходимо написать программу, выполняющую следующие действия:

* Ввод с клавиатуры данных в массив; записи должны быть упорядочены по трем первым цифрам номера телефона
* Вывод на экран информации о человеке, чья фамилия введена с клавиатуры
* Если такого нет, выдать на дисплей соответствующее сообщение

Т.к. для этой программы необходим синхронный ввод данных (проще говоря, программа должна дожидаться ввода пользователя, а не подписываться на событие “ввод”), вместо модуля ReadLine я использовал модуль Prompts.

Листинг 6.5

var fs = require("fs");

var colors = require("colors");

var prompts = require("prompts");

var db = [ *//База данных*

    ['Федотов А. М.', 89998204850, [12, 7, 1997]],

    ['Попов Н. С.', 89553442222, [28, 10, 1998]],

    ['Чекашкин В. И.', 89996554433, [31, 1, 1998]],

    ['Яблочкин С. П.', 89991234567, [1, 2, 2000]],

    ['Агибалов П. С.', 89999876543, [12, 8, 1990]]

];

var sortDb = function() { *//Функция сортировки по первым трем числам в номере*

*for* (var i = 0; i <= db.length - 2; i++) {

        var minValue = (db[i][1] + "").slice(0, 4);

        minValue = +minValue;

*for* (var j = i + 1; j <= db.length - 1; j++) {

        var actualValue = (db[j][1] + "").slice(0, 4);

        actualValue = +actualValue;

*if* (actualValue < minValue) {

            minValue = db[j];

            var swap = db[i];

            db[i] = minValue;

            db[j] = swap;

          }

        }

      }

};

var startProram = function () { *//Основная функция программы*

    sortDb(db); *// Вызов функции сортировки*

    (async () => {

*//Главное меню программы*

        console.log("");

        console.log("1".green + ". Ввод нового сотрудника в БД");

        console.log("2".green + ". Поиск сотрудника по фамилии");

        console.log("3".green + ". Вывести на экран БД");

        console.log("4".green + ". Выход из программы.");

        console.log("");

        const menu = *await* prompts({

            type: 'number',

            name: 'value',

            message: 'Выберите пункт меню.',

        });

*if* (menu.value == 1) { *//Выполняется, если пользователь введет 1*

            const newUserName = *await* prompts({

                type: 'text',

                name: 'value',

                message: 'Введите имя нового сотрудника (Фамилия и инициалы, пример ->' + " Федотов А. М.".red + ")",

            });

            var newUser = db.length;

            db.push([]);

            db[newUser].push(newUserName.value);

            const newUserPhone = *await* prompts({

                type: 'text',

                name: 'value',

                message: 'Телефон нового сотрудника',

            });

            db[newUser].push(+newUserPhone.value);

            const newUserDate = *await* prompts({

                type: 'text',

                name: 'value',

                message: 'Дата рождения (DD.MM.YYYY, пример -> ' + '01.01.2019'.red + ')',

            });

            db[newUser].push(newUserDate.value);

            db[newUser][2] = db[newUser][2].split(".");

*for* (var i = 0; i < db[newUser][2].length; i++) {

                db[newUser][2][i] = +db[newUser][2][i];

            }

            startProram(db); *//Рекурсивный вызов программы*

        } *else* *if* (menu.value == 2) { *//Выполняется, если пользователь введет 2*

            const searchUsers = *await* prompts({

                type: 'text',

                name: 'value',

                message: 'Введите фамилию и инициалы для поиска',

            });

            var nameUserForSearch = searchUsers.value;

            var successfulSearch = false;

*for* (var i = 0; i < db.length; i++) {

*if* (db[i][0] == nameUserForSearch) {

                    successfulSearch = true;

                    console.log("");

                    console.log("Сотрудник найден: ".red);

                    console.log("");

                    console.log("Имя работника: " + db[i][0].red);

                    console.log("Номер телефона: " + db[i][1]);

*//т.к. по заданию дата хранится в массиве из чисел, для корректного вывода требуется сделать несколько проверок*

*if* ((db[i][2][0] + "").length === 1) { *//если первое число в массиве с датой однозначное*

*if* ((db[i][2][1] + "").length === 1) { *// если второе число в массиве с датой однозначное*

                            console.log("Дата рождения: " + "0" + db[i][2][0] + ".0" + db[i][2][1] + "." + db[i][2][2]);

                        } *else* *if* ((db[i][2][1] + "").length === 2) { *// если второе число в массиве с датой двузначное*

                            console.log("Дата рождения: " + "0" + db[i][2][0] + "." + db[i][2][1] + "." + db[i][2][2]);

                        }

                    } *else* *if* ((db[i][2][0] + "").length === 2) { *//если первое число в массиве с датой двузначное*

*if* ((db[i][2][1] + "").length === 1) { *// если второе число в массиве с датой однозначное*

                            console.log("Дата рождения: " + db[i][2][0] + ".0" + db[i][2][1] + "." + db[i][2][2]);

                        } *else* *if* ((db[i][2][1] + "").length === 2) { *// если второе число в массиве с датой двузначное*

                            console.log("Дата рождения: " + db[i][2][0] + "." + db[i][2][1] + "." + db[i][2][2]);

                        }

                    }

                    console.log("");

                }

            }

*if* (successfulSearch === false) console.log("Сотрудник не найден.".red);

            startProram(db); *//Рекурсивный вызов программы*

        } *else* *if* (menu.value == 3) { *//Выполнится, если пользователь введет 3*

            console.log(db);

            startProram(db); *//Рекурсивный вызов программы*

        } *else* *if* (menu.value == 4) { *//Выполняется, если пользователь введет 4*

            process.exit(-1);

        };

    })();

};

startProram(db);

Результат выполнения программы в консоли:

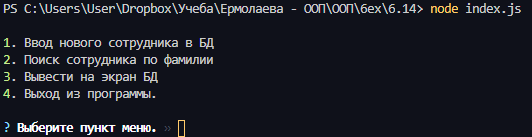


Рис.7 – Меню программы

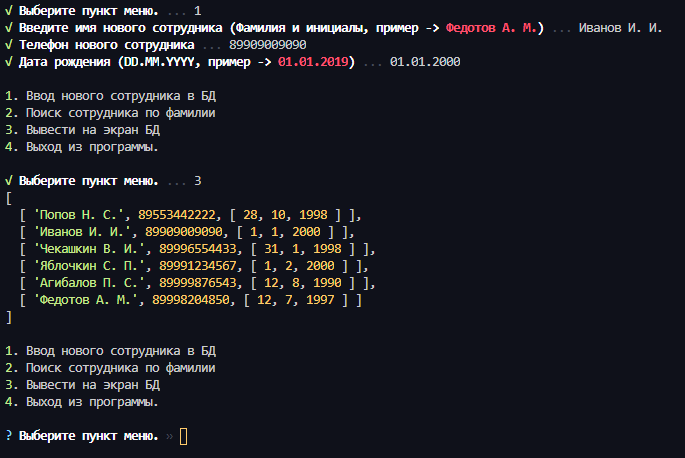


Рис.8 – Ввод нового пользователя и вывод БД на экран

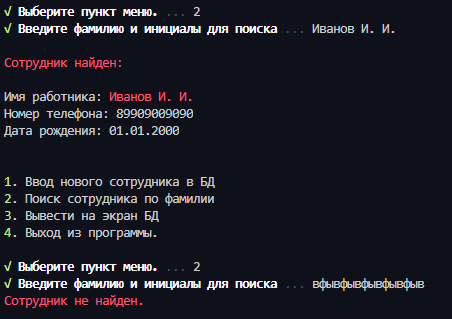


Рис.9 – Поиск сотрудника по фамилии

# Семинар 7

# 7.1 Передача в функцию параметров стандартных типов

Необходимо написать программу вывода таблицы значений функции для аргумента, изменяющегося в заданных пределах с заданным шагом. Значения функции вычислять с помощью разложения в ряд Тейлора с точностью ε.

По сути, это тоже самое задание, что и в семинаре 2. Т.к. мое решение этого задания в прошлом само по себе подразумевает вызов callback функции, то для решения достаточно просто будет заменить ввод аргумента и точности с модального окна prompt() на аргументы, передаваемые в функцию при вызове.

Листинг 7.1

var testFunc = function (x, eps) {

    const maxIter = 500;

            var done = true;

            var ch = x;

            var y = ch;

            console.log("----------------------------");

            console.log("|      X      |      Y      |");

            console.log("----------------------------");

*for* (var i = 0; ch > eps; i++) {

                ch \*= 1 \* Math.pow(x, i) / Math.pow(2, i) / Math.pow(3, i + 1);

*if* (!(i % 2)) y -= ch; *// Если счетчик i делится на 2 без остатка (каждая четная итерация)*

*else* y += ch; *//в любом другом случае (каждая нечетная итерация)*

                console.log("|      " + x + "      " + "|      " + y + "      |")

*if* (i > maxIter) {

                    console.log("Ряд расходится!");

*return* done = false;

                }

            }

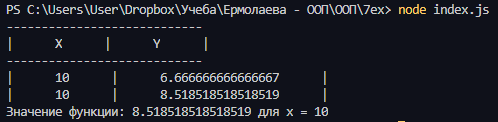
*if* (done) console.log("Значение функции: " + y + " для x = " + x);

            console.log("|      " + x + "      " + "|      " + y + "      |");

        };

        testFunc(10, 2);

Результат выполнения программы в консоли:



# 8 Семинар

# 8.1 Перегрузка функций

Написать программу, которая для базы сотрудников, описанной в задаче 6.1 выдает по запросу список сотрудников, либо родившихся раньше заданного года, либо имеющих оклад больше введенного с клавиатуры.

Листинг 8.1

var fs = require("fs");

var colors = require("colors");

var prompts = require("prompts");

var db = [];

fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

        fs.readFile("./db/dbase.txt.txt", (err, data) => {

*//Чтение файла и форматирование массива*

            var dbTemp = data.toString();

            var dbTemp2 = dbTemp.split("\r\n");

*for* (var i = 0; i < dbTemp2.length; i++) {

                db.push(dbTemp2[i].split(","));

            };

            (async () => {

                console.log("");

                console.log("1".green + ". Ввести год рождения и показать всех работников, родившихся после.");

                console.log("2".green + ". Ввести оклад и показать всех работников, чей оклад больше");

                console.log("3".green + ". Выход из программы.");

                console.log("");

                const menu = *await* prompts({

                    type: 'number',

                    name: 'value',

                    message: 'Выберите пункт меню.',

                });

*if* (menu.value == 1) {

                    const eyersInput = *await* prompts({

                        type: 'number',

                        name: 'value',

                        message: 'Введите год рождения',

                    });

                    var eyers = eyersInput.value;

                    console.log("");

                    console.log("Работники, родившееся после " + eyers + " года:");

*for* (var i = 0; i < db.length; i++) {

*if* (db[i][1] > eyers) {

                            console.log("");

                            console.log("Имя работника: " + db[i][0].red);

                            console.log("Год рождения: " + db[i][1].red);

                            console.log("Оклад: " + db[i][2].red);

                            console.log("");

                        }

                    }

                } *else* *if* (menu.value == 2) {

                    const salaryInput = *await* prompts({

                        type: 'number',

                        name: 'value',

                        message: 'Введите оклад',

                    });

                    var salary = salaryInput.value;

                    console.log("");

                    console.log("Работники, у которых оклад больше " + salary + " рублей:");

*for* (var i = 0; i < db.length; i++) {

*if* (db[i][2] > salary) {

                            console.log("");

                            console.log("Имя работника: " + db[i][0].red);

                            console.log("Год рождения: " + db[i][1].red);

                            console.log("Оклад: " + db[i][2].red);

                            console.log("");

                        }

                    }

                } *else* *if* (menu.value == 3) {

                    process.exit(-1);

                };

            })();

        });

    }

});

Результат выполнения программы в консоли:

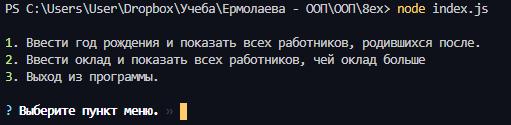


Рис.10 – Основное меню программы

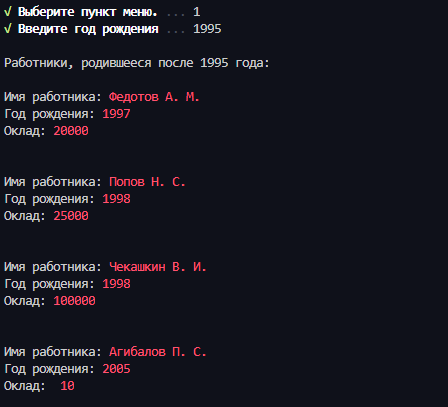


Рис.11 – Поиск по году рождения

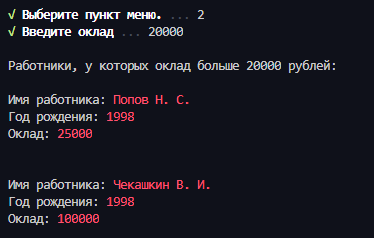


Рис.11 – Поиск по окладу

# Семинар 9

# 9.1 Стек

Необходимо написать программу сортировки вещественного массива из n элементов.

Стек (англ. stack — стопка) — структура данных, представляющая из себя список элементов организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю.

Реализация стека в JS:

var stack = [];        *// []*

stack.push( "first" ); *// stack === ["first"]*

stack.push( 10, 20 );  *// stack === ["first", 10, 20]*

var el = stack.pop();  *// stack === ["first", 10] && el === 20*

stack.push( 2 );       *// stack === ["first", 10, 2]*

el = stack.pop();      *// stack === ["first", 10] && el === 2*

el = stack.pop();      *// stack === ["first"] el = 10*

el = stack.pop();      *// stack === [] && el === "first"*

el = stack.pop();      *// stack === [] && typeof el === "undefined"*

Листинг 9.1

var arr = [5,6,8,3,1];

*//Сортировка*

*for* (var i = 0; i <= arr.length - 2; i++) {

    var minValue = arr[i];

*for* (var j = i + 1; j <= arr.length - 1; j++) {

*if* (Math.abs(arr[j]) < Math.abs(minValue)) {

            minValue = arr[j];

            var swap = arr[i];

            arr[i] = minValue;

            arr[j] = swap;

        }

    }

}

console.log (arr);

Результат выполнения программы в консоли:



# Семинар 10

# 10.1 Поиск в просто базе (массив объектов)

В текстовом файле хранится база отдела кадров предприятия. Каждая строка файла содержит запись об одном сотруднике. Формат записи: фамилия и инициалы, год рождения, оклад. Написать программу, которая по заданной фамилии выводит на экран сведения о сотруднике, подсчитывая средний оклад запрошенных сотрудников.

Задание отличается от 6.1 тем, что добавлять нового сотрудника в массив следует с помощью созданного заранее класса. В нашем случае это класс с названием User.

Листинг 10.1

var fs = require("fs");

var colors = require("colors");

var prompts = require("prompts");

var db = [];

var middleSalary = 0;

var countSalary = 0;

class User { *//Объявление класса User*

    constructor(name, year, salary) { *//Конструктор*

*this*.name = name;

*this*.year = year;

*this*.salary = salary;

    }

    info() { *//Метод вывода информации и сотруднике*

        console.log("")

        console.log("Имя: " + *this*.name.red.bold);

        console.log("Год рождения: " + *this*.year.red.bold);

        console.log("Заработная плата в рублях: " + *this*.salary.red.bold);

        console.log("");

    }

};

function programStart() {

    (async () => {

        fs.exists("./db/dbase.txt.txt", (exists) => {

*if* (exists) {

                fs.readFile("./db/dbase.txt.txt", (err, data) => {

*for* (var i = 0; i < data.toString().split("\r\n").length; i++) { *//Создание объектов класса User*

                        db[i] = new User(

                            data.toString().split("\r\n")[i].split(",")[0],

                            data.toString().split("\r\n")[i].split(",")[1],

                            data.toString().split("\r\n")[i].split(",")[2]

                        );

                    };

                });

            }

        });

        const usersSearch = *await* prompts({ *//Ввод данных для поиска*

            type: 'text',

            name: 'value',

            message: 'Введите фамилию и инициалы сотрудника',

        });

*for* (var i = 0; i < db.length; i++) { *//Поиск*

*if* (usersSearch.value === db[i].name) {

                db[i].info(); *//Вывод информации*

                countSalary++;

                middleSalary += db[i].salary / countSalary;

                console.log(("Средний оклад запрошенных пользователей: " + middleSalary).yellow.bold);

                console.log("");

                programStart(); *//Рекурсивный вызов программы*

            }

        };

    })();

};

programStart();

Результат выполнения программы в консоли:

