

Министерство науки и высшего образования Российской Федерации

Севастопольский государственный университет

Высшая технологическая школа «Севастопольский приборостроительный  
институт»

Факультет ИТ

**ОТЧЁТ**

по лабораторной работе №3

ИССЛЕДОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА К  
ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ JAVASCRIPT

Выполнил:

ст. гр. ИС/б-22-2-о

Мовенко К. М.

Проверил:

Строганов В. А.

Севастополь

2026

## 1. ЦЕЛЬ РАБОТЫ

Изучение особенностей объектно-ориентированного подхода разработки ПО, получение практических навыков разработки ПО с использованием JavaScript.

## 2. ЗАДАНИЕ

- 2.1. Изучить основные средства языка JavaScript для разработки программ с использованием объектно-ориентированного подхода;
- 2.2. Выполнить две задачи на языке JavaScript согласно своему варианту;
- 2.3. Разработать тестовые примеры;
- 2.4. Выполнить отладку программ;
- 2.5. Сформулировать выводы, проанализировав отличия объектного подхода от разработки программ с использованием декларативных и императивных парадигм;
- 2.6. Оформить отчёт по проделанной работе;

## 3. ВАРИАНТ ЗАДАНИЯ

Создать одномерный и двухмерный массивы. Для однокамерного массива нужно добавить минимум 10 элементов, для матрицы – 16 элементов.

Вариант – 1.

Найти сумму всех элементов в массиве.

Удалить все строки матрицы, в которых есть отрицательные элементы.

## 4. ХОД РАБОТЫ

Перед началом работы были рассмотрены основные понятия объектно-ориентированного программирования. Изучено то, как работать с классами и объектами на языке JavaScript.

В среде Visual Studio Code был написан код на языке JavaScript, который задаёт в программе одномерный массив из 10 элементов, через конструктор заносит его в объект класса NumberArray а затем обращается к его методу sum() для нахождения суммы элементов (листинг 4.1).

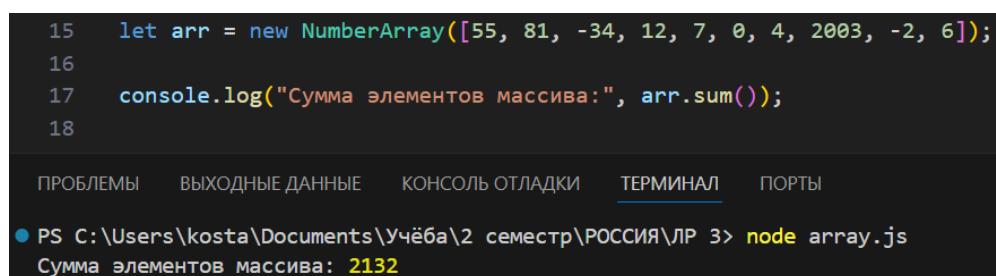
Листинг 4.1 – Сумма одномерного массива через метод класса

```
class NumberArray {
    constructor(values) {
        this.values = values;
    }

    sum() {
        let total = 0;
        for (let i = 0; i < this.values.length; i++) {
            total += this.values[i];
        }
        return total;
    }
}

let arr = new NumberArray([55, 81, -34, 12, 7, 0, 4, 2003, -2, 6]);
console.log("Сумма элементов массива:", arr.sum());
```

В итоге программа успешно выводит на экран сумму элементов исходного списка (рисунок 4.1).



```
15  let arr = new NumberArray([55, 81, -34, 12, 7, 0, 4, 2003, -2, 6]);
16
17  console.log("Сумма элементов массива:", arr.sum());
18

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

● PS C:\Users\kosta\Documents\Учёба\2 семестр\РОССИЯ\ЛР 3> node array.js
Сумма элементов массива: 2132
```

Рисунок 4.1 – Запуск первой программы

Далее была написана программа на JavaScript, которая хранит двумерный массив 4x4 в параметре класса NumberMatrix, а также предоставляет метод для фильтрации строк с отрицательными элементами (листинг 4.2).

Метод `getPositive()` использует функциональный код из предыдущей работы чтобы выбрать из матрицы все строки, в которых не содержатся отрицательные значения.

#### Листинг 4.2 – Фильтрация матрицы вызовом метода

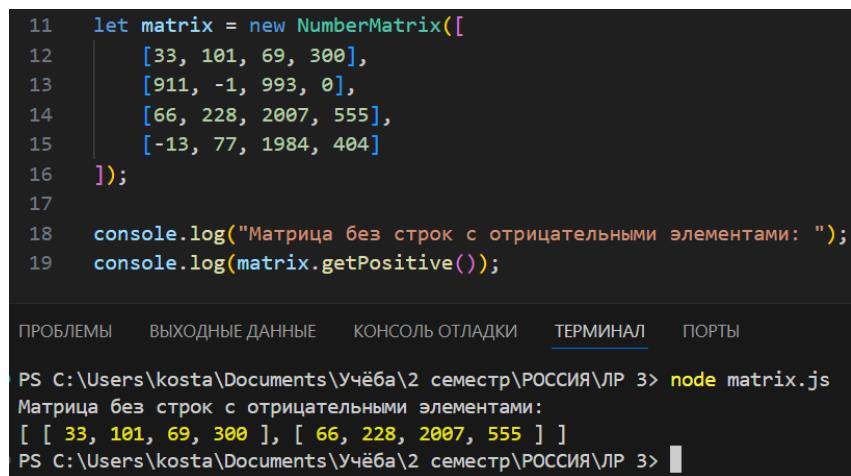
```
class NumberMatrix {
    constructor(values) {
        this.matrix = values;
    }

    getPositive() {
        return this.matrix.filter(row => !row.some(x => x < 0))
    }
}

let matrix = new NumberMatrix([
    [33, 101, 69, 300],
    [911, -1, 993, 0],
    [66, 228, 2007, 555],
    [-13, 77, 1984, 404]
]);

console.log("Матрица без строк с отрицательными элементами: ");
console.log(matrix.getPositive());
```

В итоге программа выводит вариант исходной матрицы без строк с отрицательными элементами (рисунок 4.2).



```
11 let matrix = new NumberMatrix([
12     [33, 101, 69, 300],
13     [911, -1, 993, 0],
14     [66, 228, 2007, 555],
15     [-13, 77, 1984, 404]
16 ]);
17
18 console.log("Матрица без строк с отрицательными элементами: ");
19 console.log(matrix.getPositive());
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

PS C:\Users\kosta\Documents\Учёба\2 семестр\РОССИЯ\ЛР 3> node matrix.js  
Матрица без строк с отрицательными элементами:  
[ [ 33, 101, 69, 300 ], [ 66, 228, 2007, 555 ] ]  
PS C:\Users\kosta\Documents\Учёба\2 семестр\РОССИЯ\ЛР 3>

Рисунок 4.2 – Запуск второй программы

## 5. ВЫВОД

В ходе работы была рассмотрена парадигма объектно-ориентированного программирования: её ключевые идеи, принципы, способы реализации.

Объектный подход к программам был применён на практике при написании кода на языке JavaScript. Полученные программы создают простейшие классы для хранения динамических структур данных и реализуют обращение к ним через методы.

Поскольку предложенные задания не раскрывают всех возможностей ООП, наглядно показать преимущества данного подхода перед обычным структурным программированием не удалось. Однако представление данных в виде объектов иерархии классов является мощным инструментом для формализованного представления данных и организации их взаимодействия за счёт богатого инструментария (опции, структуры, паттерны и др.).