

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий Кафедра Информатики и информационных технологий

направление подготовки 09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ РАБОТА) № _3_

Дисципли	на: Функциональ	ьное програ	аммирование			
Гема:	_ <u>Основы ТуреScri</u>	pt для функ	щионального			
программи	<u> трования</u>					
Вариант						
Бириши						
	Выполнил(а): студент(ка) группы 221-374					
	Мако	Максимов Юрий Сергеевич (Фамилия И.О.)				
	Дата	, подпись _		(Подпись)		
	Пров	верил:				
	•		(Фамилия И.О., степень, звание) (Оцен			
	Дата	, подпись _				
			(Дата)	(Подпи	сь)	

Задание

Разработайте набор чистых функций для работы с массивами:

- 1) Функция, которая принимает массив чисел и возвращает новый массив, содержащий только числа, кратные заданному числу.
- 2) Функция, которая принимает массив строк и возвращает новую строку, содержащую все строки, объединенные заданным разделителем.
- 3) Функция, которая принимает массив объектов и возвращает новый массив, отсортированный по значению определенного свойства.

Создайте функцию, которая принимает другую функцию в качестве аргумента и возвращает новую функцию, которая выполняет логирование перед вызовом исходной функции.

Код

```
// Функция, которая принимает массив чисел и возвращает новый
массив,
// содержащий только числа, кратные заданному числу.
function filter multiples(array: number[], multiple: number):
number[] {
    return array.filter(num => num % multiple === 0)
}
// Функция, которая принимает массив строк и возвращает новую
строку,
// содержащую все строки, объединенные заданным разделителем.
function join strs(array: string[], sep: string): string {
    return array.join(sep)
}
// Функция, которая принимает массив объектов и возвращает новый
массив,
// отсортированный по значению определенного свойства.
```

```
function array sort<T>(array: T[], property: keyof T): T[] {
    return [...array].sort((a, b) => a[property] > b[property] ? 1
: -1)
}
// Создайте функцию, которая принимает другую функцию в качестве
аргумента и возвращает новую функцию,
// которая выполняет логирование перед вызовом исходной функции.
function logDecorator<T extends (...args: any[]) => any>(func: T):
T {
    return ((...args: Parameters<T>) => {
        console.log(`Arguments: ${JSON.stringify(args)}`);
        return func(...args);
    }) as T
}
// Примеры
// Числа, кратные 3
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9];
const multiples = filter multiples(numbers, 2);
console.log(multiples);
// Объединение строк
const strings = ['a', 'b', 'c'];
const joinedStrings = join strs(strings, '^^');
console.log(joinedStrings);
// Сортировка объектов по свойству
interface Person {
    name: string;
    age: number;
}
```

Ссылка на код

https://github.com/MaksimovYuriy/FunctionalProgramming/tree/main/Лаб%203

Скриншот выполнения

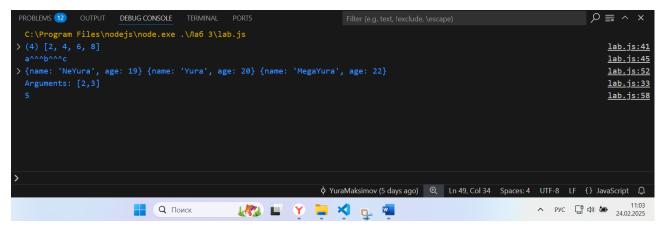


Рис. 1 – Выполнение программы