**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ (ПРАКТИЧЕСКАЯ РАБОТА) №** \_3\_

**Дисциплина: Функциональное программирование**

**Тема:** \_\_\_\_ Применение функционального программирования в TypeScript\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Вариант**

**Выполнил(а): студент(ка) группы 221-374**

**Максимов Юрий Сергеевич**

(Фамилия И.О.)

**Дата, подпись** \_\_03.03.2025 \_\_\_\_\_\_\_\_\_\_

(Дата)(Подпись)

**Проверил:** \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_*

(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_*

(Дата)(Подпись)

**Москва**

**2025**

**Задание**

Разработайте веб-приложение "Калькулятор", которое позволяет пользователю выполнять следующие операции:

* Сложение, вычитание, умножение и деление.
* Возведение в степень.
* Вычисление квадратного корня.

Требования:

* Используйте принципы функционального программирования, такие как иммутабельность данных и чистые функции.
* Используйте функции высшего порядка для обработки данных и создания новых функций.
* Веб-приложение должно быть реализовано с использованием HTML, CSS и TypeScript.
* Интерфейс должен быть интуитивно понятным и удобным для пользователя.

**Код**

**index.html**

|  |
| --- |
| **<!DOCTYPE html>**  **<html lang="ru">**  **<head>**  **<meta charset="UTF-8">**  **<meta name="viewport" content="width=device-width, initial-scale=1.0">**  **<title>Калькулятор</title>**  **<link rel="stylesheet" href="styles.css">**  **</head>**  **<body>**  **<div class="calculator">**  **<div class="display">**  **<div id="operation-display" class="operation-display"></div>**  **<input id="display" type="text" readonly>**  **</div>**  **<div class="buttons">**  **<button class="btn button-num" onclick="handleButtonClick('7')">7</button>**  **<button class="btn button-num" onclick="handleButtonClick('8')">8</button>**  **<button class="btn button-num" onclick="handleButtonClick('9')">9</button>**  **<button class="btn button-op" onclick="handleOperation('divide')">/</button>**  **<button class="btn button-num" onclick="handleButtonClick('4')">4</button>**  **<button class="btn button-num" onclick="handleButtonClick('5')">5</button>**  **<button class="btn button-num" onclick="handleButtonClick('6')">6</button>**  **<button class="btn button-op" onclick="handleOperation('multiply')">\*</button>**  **<button class="btn button-num" onclick="handleButtonClick('1')">1</button>**  **<button class="btn button-num" onclick="handleButtonClick('2')">2</button>**  **<button class="btn button-num" onclick="handleButtonClick('3')">3</button>**  **<button class="btn button-op" onclick="handleOperation('subtract')">-</button>**  **<button class="btn button-op" onclick="handleButtonClick('.')">.</button>**  **<button class="btn button-num" onclick="handleButtonClick('0')">0</button>**  **<button class="btn button-op" onclick="handleOperation('add')">+</button>**  **<button class="btn button-op" onclick="handleOperation('power')">^</button>**  **<button class="btn button-op" onclick="handleSqrt()">√</button>**  **<button class="btn button-op" onclick="handleClear()">C</button>**  **<button class="btn button-op" onclick="handleSignChange()">+/-</button>**  **<button class="btn button-equal" onclick="handleEqual()">=</button>**  **</div>**  **</div>**  **<script src="app.js"></script>**  **</body>**  **</html>** |

styles.css

|  |
| --- |
| **body {**  **font-family: Arial, sans-serif;**  **display: flex;**  **justify-content: center;**  **align-items: center;**  **height: 100vh;**  **margin: 0;**  **background-color: #f4f4f4;**  **}**  **.calculator {**  **width: 300px;**  **background-color: #F0F8FF;**  **padding: 20px;**  **border-radius: 10px;**  **box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);**  **}**  **.display {**  **margin-right: 20px;**  **margin-bottom: 20px;**  **}**  **#operation-display {**  **font-size: 1.2em;**  **color: #888;**  **text-align: right;**  **margin-bottom: 10px;**  **}**  **#display {**  **width: 100%;**  **height: 40px;**  **font-size: 1.5em;**  **padding: 10px;**  **text-align: right;**  **border-radius: 5px;**  **border: 1px solid #ccc;**  **}**  **.button-op {**  **background-color: #D3D3D3;**  **}**  **.button-num {**  **background-color: white;**  **}**  **.button-equal {**  **background-color: #AFEEEE;**  **}**  **.buttons {**  **display: grid;**  **grid-template-columns: repeat(4, 1fr);**  **gap: 10px;**  **}**  **.btn {**  **padding: 20px;**  **font-size: 1.5em;**  **border: none;**  **border-radius: 5px;**  **cursor: pointer;**  **transition: background-color 0.3s;**  **}**  **.button-num:hover {**  **background-color: #ddd;**  **}**  **.button-op:hover {**  **background-color: #888;**  **}**  **.button-equal:hover {**  **background-color: #87CEEB;**  **}** |

**app.ts**

|  |
| --- |
| **const add = (a: number, b: number): number => a + b;**  **const subtract = (a: number, b: number): number => a - b;**  **const multiply = (a: number, b: number): number => a \* b;**  **const divide = (a: number, b: number): number => b !== 0 ? a / b : NaN;**  **const power = (a: number, b: number): number => Math.pow(a, b);**  **const sqrt = (a: number): number => a >= 0 ? Math.sqrt(a) : NaN;**  **const sign = (a: number): number => a \* (-1)**  **type Operation = 'add' | 'subtract' | 'multiply' | 'divide' | 'power' | 'sqrt' | 'sign';**  **let currentInput: string = '';**  **let previousInput: string = '';**  **let currentOperation: Operation | null = null;**  **const updateDisplay = (input: string): void => {**  **const display = document.getElementById('display') as HTMLInputElement;**  **display.value = input;**  **};**  **const updateOperationDisplay = (): void => {**  **const operationDisplay = document.getElementById('operation-display') as HTMLElement;**  **if (previousInput !== '' && currentOperation !== null) {**  **const operationSymbol = getOperationSymbol(currentOperation);**  **operationDisplay.textContent = `${previousInput} ${operationSymbol}`;**  **} else {**  **operationDisplay.textContent = '';**  **}**  **};**  **const getOperationSymbol = (operation: Operation): string => {**  **switch (operation) {**  **case 'add': return '+';**  **case 'subtract': return '-';**  **case 'multiply': return '\*';**  **case 'divide': return '/';**  **case 'power': return '^';**  **case 'sqrt': return '√';**  **default: return '';**  **}**  **};**  **const handleButtonClick = (value: string): void => {**  **currentInput += value;**  **updateDisplay(currentInput);**  **updateOperationDisplay();**  **};**  **const handleClear = (): void => {**  **currentInput = '';**  **previousInput = '';**  **currentOperation = null;**  **updateDisplay('');**  **updateOperationDisplay();**  **};**  **const handleOperation = (operation: Operation): void => {**  **if (currentInput === '') return;**  **if (previousInput !== '') {**  **handleEqual();**  **}**  **currentOperation = operation;**  **previousInput = currentInput;**  **currentInput = '';**  **updateOperationDisplay();**  **};**  **const handleEqual = (): void => {**  **if (previousInput === '' || currentInput === '') return;**  **const prev = parseFloat(previousInput);**  **const current = parseFloat(currentInput);**  **let result: number;**  **switch (currentOperation) {**  **case 'add':**  **result = add(prev, current);**  **break;**  **case 'subtract':**  **result = subtract(prev, current);**  **break;**  **case 'multiply':**  **result = multiply(prev, current);**  **break;**  **case 'divide':**  **result = divide(prev, current);**  **break;**  **case 'power':**  **result = power(prev, current);**  **break;**  **case 'sqrt':**  **result = sqrt(prev);**  **break;**  **default:**  **return;**  **}**  **currentInput = result.toString();**  **previousInput = '';**  **currentOperation = null;**  **updateDisplay(currentInput);**  **updateOperationDisplay();**  **};**  **const handleSqrt = (): void => {**  **if (currentInput === '') return;**  **const num = parseFloat(currentInput);**  **const result = sqrt(num);**    **currentInput = result.toString();**  **updateDisplay(currentInput);**  **updateOperationDisplay();**  **};**  **const handleSignChange = (): void => {**  **if (currentInput === '') return;**  **let currentNumber = parseFloat(currentInput);**  **currentNumber = -currentNumber;**  **currentInput = currentNumber.toString();**  **updateDisplay(currentInput);**  **};** |

**Ссылка на код**

<https://github.com/MaksimovYuriy/FunctionalProgramming/tree/main/Лаб%204>

**Скриншот выполнения**

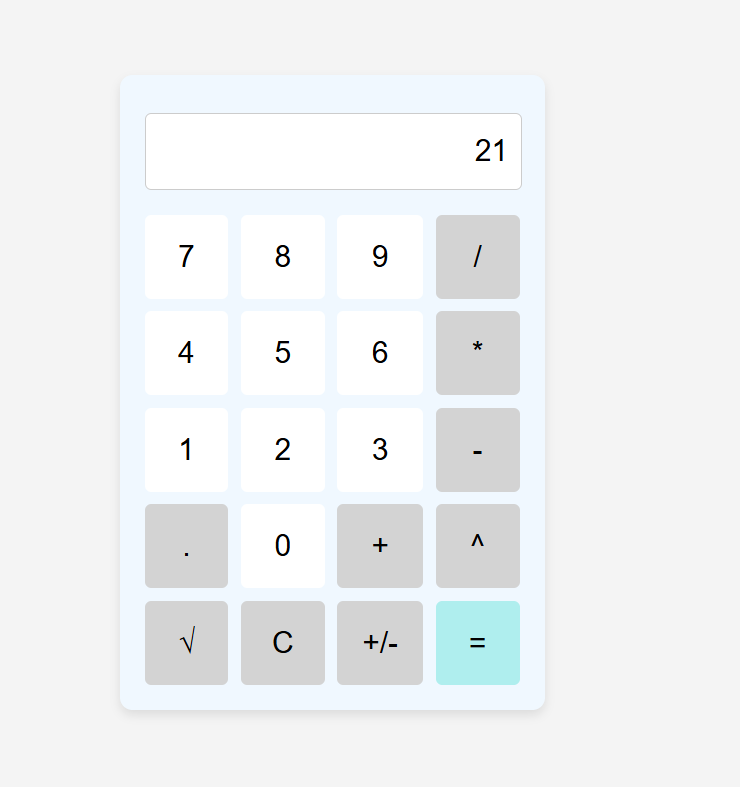


Рис. 1 - Калькулятор