Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6

Основы работы c JavaScript

тема

|  |  |  |
| --- | --- | --- |
| Преподаватель |  | А.К. Погребников |
|  | подпись, дата | инициалы, фамилия |
| Студент КИ18-17/1б 031831229 |  | В.А. Прекель |
| номер группы, зачетной книжки | подпись, дата | инициалы, фамилия |

Красноярск 2020

# 1 Цель

Освоить возможности динамического управления содержимым веб-страницы при помощи кода на JavaScript.

# 2 Задачи

Выбрать 3 задания из списка и реализовать их в виде функции. Функции должны принимать на вход все необходимые данные (например, последовательности цифр или символов). Вывод реализовать в любом удобном формате. Для выполнения практического задания были выбраны следующие задания:

1. Найти сумму элементов последовательности.

8. Сколько соответствующих элементов двух последовательностей с одинаковым количеством элементов совпадают?

26. Найти разность максимального и минимального элементов последовательности.

# 3 Ход работы

Вместо JavaScript был использован TypeScript, компилируемый в JavaScript. Сначала реализован модуль с самими функциями: для первой задачи использована стрелочная функция, для второй обычная через ключевое слово function, для третьей реализован класс. Для вычисления всех задач использовались функции map и reduce.

Листинг 1 – tasks.ts

// 1. Найти сумму элементов последовательности.  
export const arraySum = (a: number[]) => a.reduce((s, v) => s + v);  
  
// 8. Сколько соответствующих элементов двух последовательностей с одинаковым количеством элементов совпадают?  
export function zipEqualCount(a: number[], b: number[]) {  
 return a.map((elementA, index) => {  
 const elementB = b[index];  
 return { a: elementA, b: elementB };  
 }).reduce((count, value) =>  
 count + (value.a == value.b ? 1 : 0), 0  
 )  
}  
  
// 26. Найти разность максимального и минимального элементов последовательности.  
export class MinMax {  
 min: number;  
 max: number;  
  
 constructor(min: number = Number.MAX\_VALUE, max: number = Number.MIN\_VALUE) {  
 this.min = min;  
 this.max = max;  
 }  
  
 public static findMinMax(items: number[]) {  
 return items.reduce(  
 (accumulator, currentValue) => new MinMax(Math.min(currentValue, accumulator.min), Math.max(currentValue, accumulator.max)),  
 new MinMax()  
 );  
 }  
  
 public static differenceMaxMin(items: number[]) {  
 let res = MinMax.findMinMax(items);  
 return res.max - res.min;  
 }  
}

Была создана отдельная страница, на которой будет ввод и результат выполнения функций.

Листинг 2 – tasks.html

<!DOCTYPE html>  
<html lang="ru">  
  
<head>  
 <meta charset="UTF-8">  
 <meta name="robots" content="index,all">  
 <meta name="author" content="Vladislav Prekel">  
 <meta name="keywords" content="многочлен, полином, уравнение, алгебраический">  
 <meta name="description"  
 content="Многочлены и алгебраические уравнения. Численное решение алгебраических уравнений.">  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
  
 <link type="image/x-icon" href="assets/favicon.ico" rel="icon">  
  
 <script src="https://kit.fontawesome.com/e8a7538d16.js" crossorigin="anonymous"></script>  
  
 <title>Многочлены и алгебраические уравнения (задачи к 6 практической)</title>  
  
 <link href="public/css/custom.css" rel="stylesheet">  
 <link href="css/styles.css" rel="stylesheet">  
</head>  
  
<body id="body">  
 <a class="fixed-bottom btn-primary" id="to\_top" href="#">Наверх</a>  
 <div class="container">  
 <nav class="navbar navbar-expand-lg navbar-dark bg-primary">  
 <a class="navbar-brand" href="index.html">Многочлены и алгебраические уравнения</a>  
 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
 <ul class="navbar-nav ml-auto">  
 <li class="nav-item"><a class="nav-link" href="tasks.html">Задачи</a></li>  
 <li class="nav-item"><a class="nav-link" href="form.html">Форма ввода</a></li>  
 </ul>  
 </div>  
 </nav>  
 <header class="jumbotron">  
 <h1>Задачи для 6 практической</h1>  
 </header>  
 <div class="card card-body mb-4">  
 <form class="form-group">  
 <label>1. Найти сумму элементов последовательности.</label>  
 <input class="form-control" name="task1Input" type="text"> <br>  
 <input class="btn btn-success" type="button" value="Click" onClick="task1(this.form.task1Input.value)">  
 </form>  
 </div>  
 <div class="card card-body mb-4">  
 <form class="form-group">  
 <label>8. Сколько соответствующих элементов двух последовательностей с одинаковым количеством  
 элементов совпадают?</label>  
 <input class="form-control" name="task8Input1" type="text"> <br>  
 <input class="form-control" name="task8Input2" type="text"> <br>  
 <input class="btn btn-success" type="button" value="Click"  
 onClick="task8(this.form.task8Input1.value, this.form.task8Input2.value)">  
 </form>  
 </div>  
 <div class="card card-body mb-4">  
 <form class="form-group">  
 <label>26. Найти разность максимального и минимального элементов последовательности.</label>  
 <input class="form-control" name="task26Input" type="text"> <br>  
 <input class="btn btn-success" type="button" value="Click"  
 onClick="task26(this.form.task26Input.value)">  
 </form>  
 </div>  
 <div class="card card-body mb-4">  
 <h2>Ответ: </h2>  
 <p id="answer">NaN</p>  
 </div>  
 </div>  
 <script src="./public/js/funcs.js" type="module"></script>  
  
 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"  
 integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"  
 crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"  
 integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"  
 crossorigin="anonymous"></script>  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"  
 integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"  
 crossorigin="anonymous"></script>  
</body>  
  
</html>

Для связи функций со страницей создан следующий скрипт:

Листинг 3 – funcs.ts

import \* as tasks from "./tasks.js";  
  
function task1(x: string) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.arraySum(x.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
  
function task8(x: string, y: string) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.zipEqualCount(x.split(" ").map(n => parseInt(n)), y.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
  
function task26(x: string) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.MinMax.differenceMaxMin(x.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
  
declare global {  
 interface Window {  
 task1(x: string): void;  
 task8(x: string, y: string): void;  
 task26(x: string): void;  
 }  
}  
  
window.task1 = task1  
window.task8 = task8  
window.task26 = task26

TypeSctipt может компилироваться в разные спецификации ECMAScript (ES5 более старая, чем ES2015 и отличается, например, отсутствием классов):

Листинг 4 – tasks.js, ES5

// 1. Найти сумму элементов последовательности.  
export var arraySum = function (a) { return a.reduce(function (s, v) { return s + v; }); };  
// 8. Сколько соответствующих элементов двух последовательностей с одинаковым количеством элементов совпадают?  
export function zipEqualCount(a, b) {  
 return a.map(function (elementA, index) {  
 var elementB = b[index];  
 return { a: elementA, b: elementB };  
 }).reduce(function (count, value) {  
 return count + (value.a == value.b ? 1 : 0);  
 }, 0);  
}  
// 26. Найти разность максимального и минимального элементов последовательности.  
var MinMax = /\*\* @class \*/ (function () {  
 function MinMax(min, max) {  
 if (min === void 0) { min = Number.MAX\_VALUE; }  
 if (max === void 0) { max = Number.MIN\_VALUE; }  
 this.min = min;  
 this.max = max;  
 }  
 MinMax.findMinMax = function (items) {  
 return items.reduce(function (accumulator, currentValue) { return new MinMax(Math.min(currentValue, accumulator.min), Math.max(currentValue, accumulator.max)); }, new MinMax());  
 };  
 MinMax.differenceMaxMin = function (items) {  
 var res = MinMax.findMinMax(items);  
 return res.max - res.min;  
 };  
 return MinMax;  
}());  
export { MinMax };

Листинг 5 – funcs.js, ES5

import \* as tasks from "./tasks.js";  
function task1(x) {  
 var a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.arraySum(x.split(" ").map(function (n) { return parseInt(n); })).toString();  
 }  
}  
function task8(x, y) {  
 var a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.zipEqualCount(x.split(" ").map(function (n) { return parseInt(n); }), y.split(" ").map(function (n) { return parseInt(n); })).toString();  
 }  
}  
function task26(x) {  
 var a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.MinMax.differenceMaxMin(x.split(" ").map(function (n) { return parseInt(n); })).toString();  
 }  
}  
window.task1 = task1;  
window.task8 = task8;  
window.task26 = task26;

Листинг 6 – tasks.js, ES2015

// 1. Найти сумму элементов последовательности.  
export const arraySum = (a) => a.reduce((s, v) => s + v);  
// 8. Сколько соответствующих элементов двух последовательностей с одинаковым количеством элементов совпадают?  
export function zipEqualCount(a, b) {  
 return a.map((elementA, index) => {  
 const elementB = b[index];  
 return { a: elementA, b: elementB };  
 }).reduce((count, value) => count + (value.a == value.b ? 1 : 0), 0);  
}  
// 26. Найти разность максимального и минимального элементов последовательности.  
export class MinMax {  
 constructor(min = Number.MAX\_VALUE, max = Number.MIN\_VALUE) {  
 this.min = min;  
 this.max = max;  
 }  
 static findMinMax(items) {  
 return items.reduce((accumulator, currentValue) => new MinMax(Math.min(currentValue, accumulator.min), Math.max(currentValue, accumulator.max)), new MinMax());  
 }  
 static differenceMaxMin(items) {  
 let res = MinMax.findMinMax(items);  
 return res.max - res.min;  
 }  
}

Листинг 7 – funcs.js, ES2015

import \* as tasks from "./tasks.js";  
function task1(x) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.arraySum(x.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
function task8(x, y) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.zipEqualCount(x.split(" ").map(n => parseInt(n)), y.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
function task26(x) {  
 let a = document.getElementById("answer");  
 if (a != null) {  
 a.innerHTML = tasks.MinMax.differenceMaxMin(x.split(" ").map(n => parseInt(n))).toString();  
 }  
}  
window.task1 = task1;  
window.task8 = task8;  
window.task26 = task26;

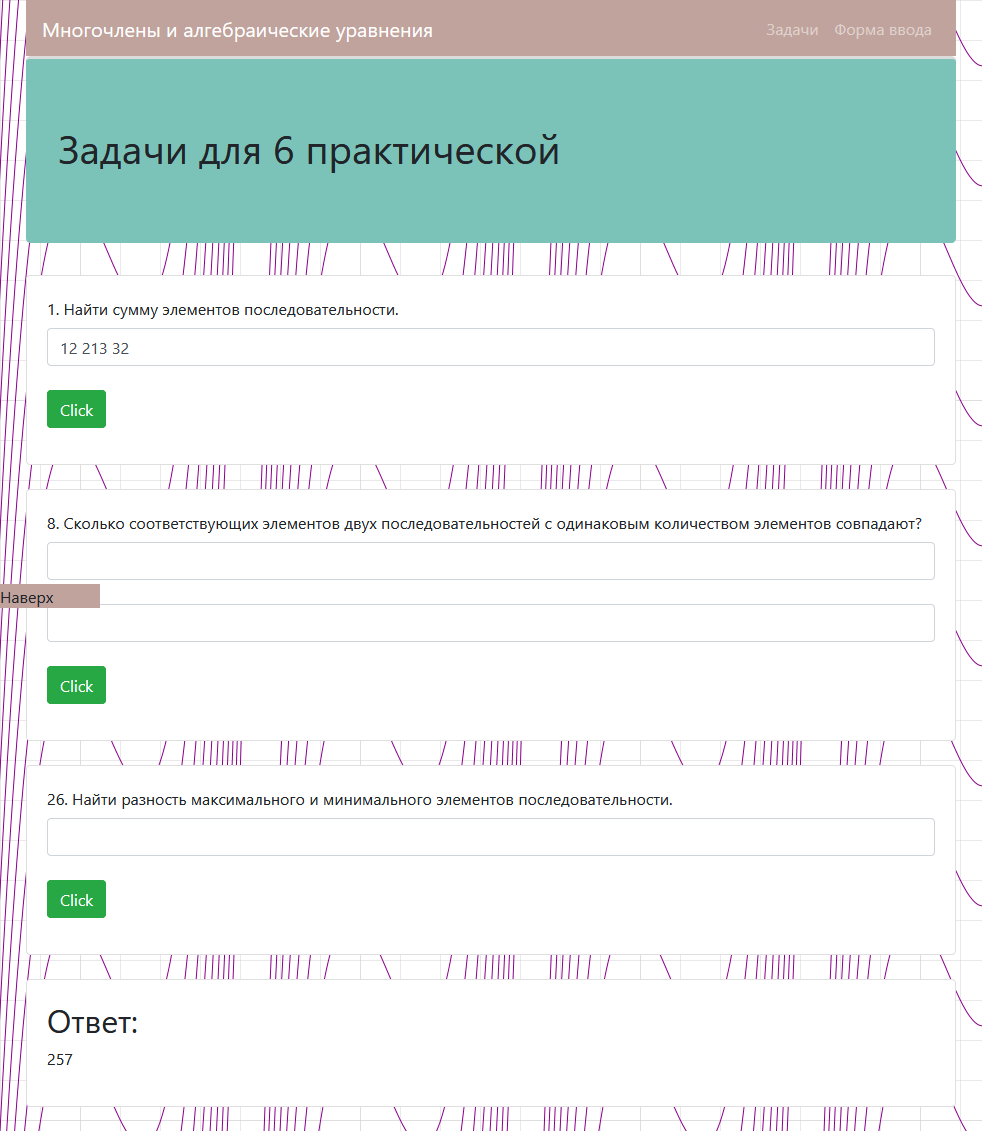


Рисунок 1 – Страница с задачами, первая задача

# 4 Вывод

Были освоены на практике базовые возможности динамического управления содержимым веб-страницы. Сайт доступен по адресу <https://prekel.github.io/WebFrontend/Pr_06/tasks.html>