Прикладное программирование

Практика 22

Библиотека API Yandex maps: инкрементный подход к разработке web-приложения

Для каждой программы, состоящей из нескольких файлов,

создавайте отдельную папку проекта, например project22-1,

project22-2 и т.д., где "22" - номер практического

занятия, 1, 2 и т.д. - номер задания.

Каждую программу, состоящую из одного файла, сохранять

в отдельном файле для каждого упражнения,

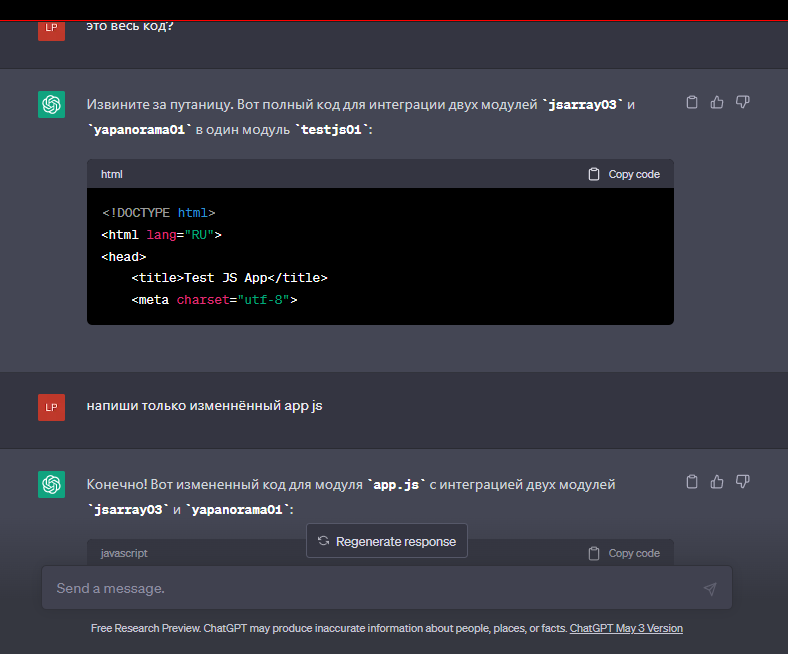
например prog22-1, prog22-2 и т.д., где "22" - номер практического

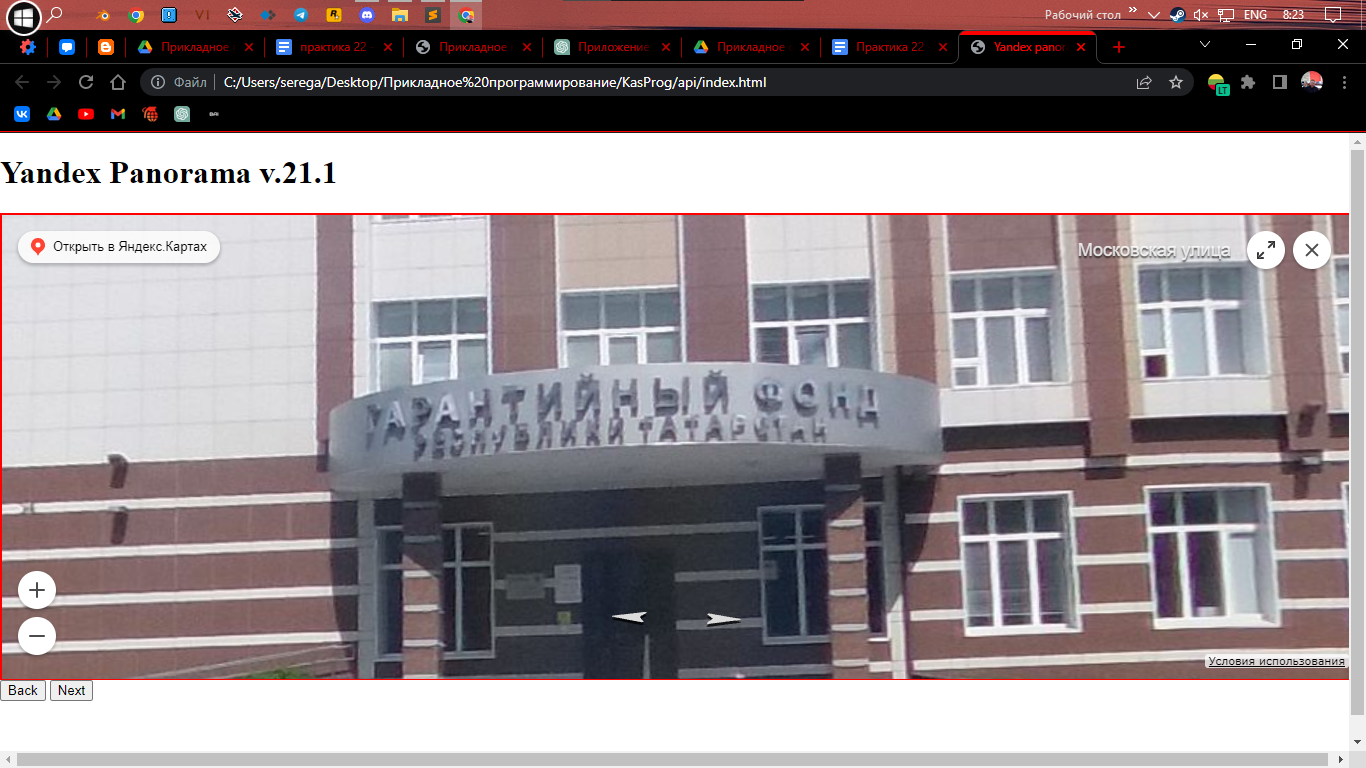
занятия, 1, 2 и т.д. - номер задания.

1. По примеру 28 из лекции создайте приложение проекта testjs01.

Применить инкрементный подход к разработке web-приложения

на языке JavaScript на примере проекта yapanorama01.





<!DOCTYPE html>

<html lang="RU">

<head>

<title> Yandex panoramas v27.1 </title>

<meta charset="utf-8">

<script type="text/javascript" src="http://api-maps.yandex.ru/2.1/?apikey=4c447df8-2105-4390-81f1-740562d9cc5d&lang=ru-RU"></script>

<script src="main.js"></script>

<style>

html,body {

width:100%;

height:100%;

margin:0;

padding:0;

}

div.player1{

padding:0;

margin:0;

height:75%;

width:100%;

border: solid 2px red;

}

</style>

</head>

<body>

<h1>Yandex Panorama v.21.1</h1>

<div id="player" class="player1"></div>

<button onclick="onClickBack()">Back</button>

<button onclick="onClickNext()">Next</button>

</body>

</html>

app,js  
console.log("Test JS App");

// Координаты локаций

var locations = [

{ x: 55.790927, y: 49.114453 }, // Location 1

{ x: 55.797391, y: 49.105993 } // Location 2

];

// Индекс текущей локации

var currentLocationIndex = 0;

// Создаем панорамный плеер

var player;

// Функция инициализации

function init() {

console.log("init:called");

ymaps.ready(function () {

ymaps.panorama.locate([locations[currentLocationIndex].x, locations[currentLocationIndex].y]).done(function (panoramas) {

player = new ymaps.panorama.Player("player", panoramas[0], { direction: [180, 15] });

});

});

}

// Функция перехода к следующей панораме

function nextPanorama() {

currentLocationIndex = (currentLocationIndex + 1) % locations.length;

ymaps.panorama.locate([locations[currentLocationIndex].x, locations[currentLocationIndex].y]).done(function (panoramas) {

player.setPanorama(panoramas[0]);

});

}

// Функция перехода к предыдущей панораме

function previousPanorama() {

currentLocationIndex = (currentLocationIndex - 1 + locations.length) % locations.length;

ymaps.panorama.locate([locations[currentLocationIndex].x, locations[currentLocationIndex].y]).done(function (panoramas) {

player.setPanorama(panoramas[0]);

});

}

// Функция увеличения чисел массива на 1

function incrementArray(numbers) {

return numbers.map(function (number) {

return number + 1;

});

}

// Пример использования функции incrementArray

var numbers = [1, 2, 3, 4, 5];

var incrementedNumbers = incrementArray(numbers);

console.log("Incremented Numbers:", incrementedNumbers);

// Инициализация при загрузке страницы

init();

2. Выполните пример 29 из лекции, создайте проект jsarray04.

Географические координаты локации хранятся в одномерном и двумерном массивах.

Напишите программу на языке JavaScript, в которой задан массив

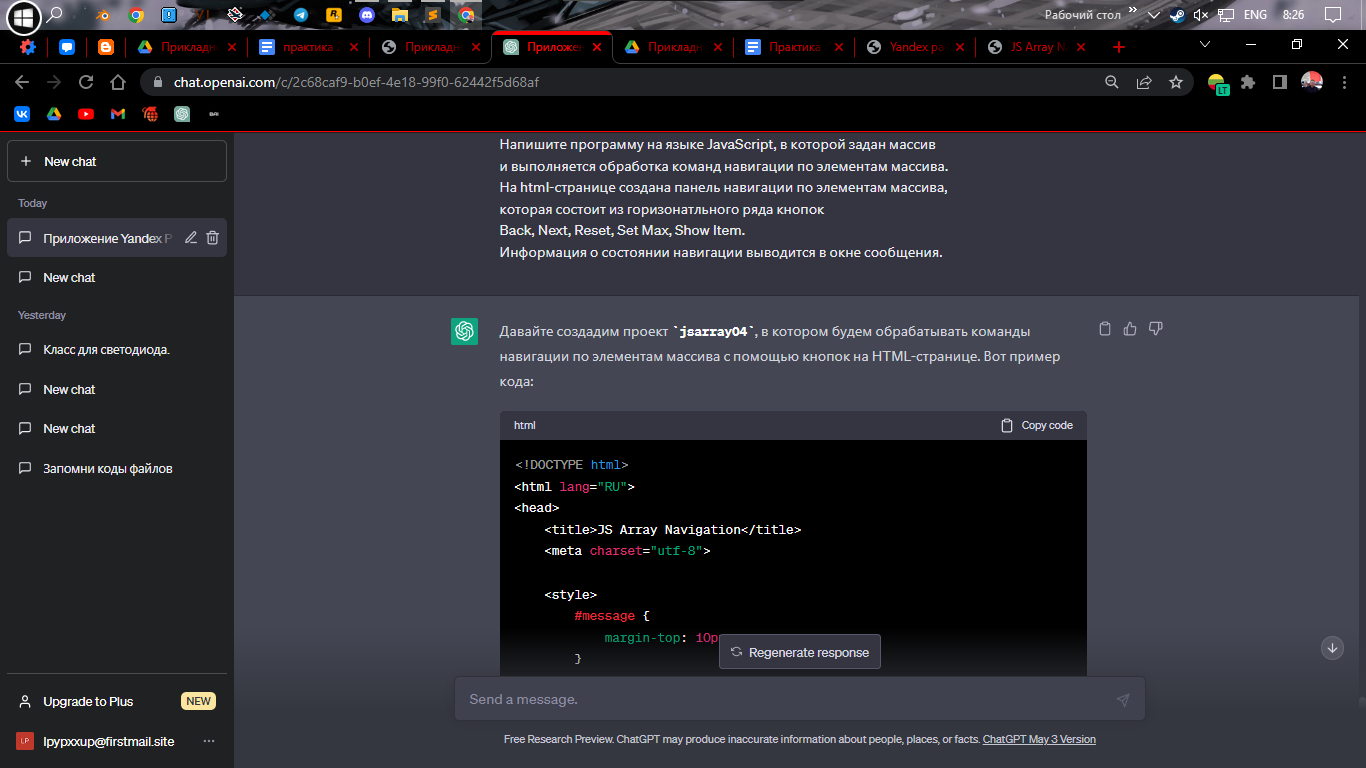
и выполняется обработка команд навигации по элементам массива.

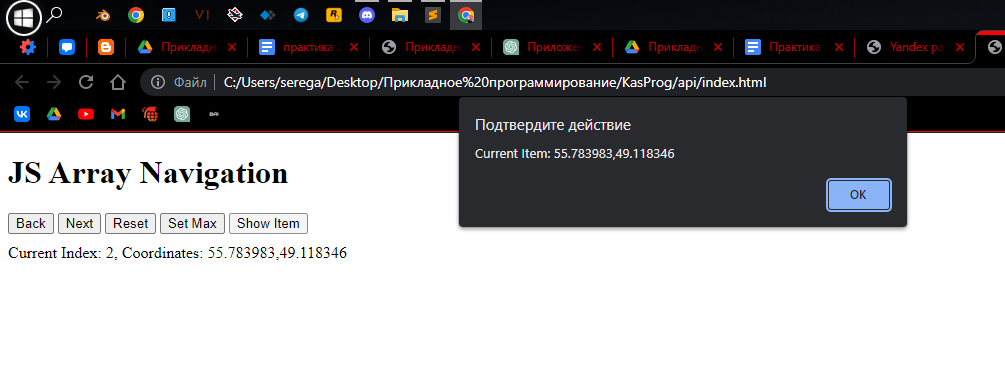
На html-странице создана панель навигации по элементам массива,

которая состоит из горизонатльного ряда кнопок

Back, Next, Reset, Set Max, Show Item.

Информация о состоянии навигации выводится в окне сообщения.





<!DOCTYPE html>

<html lang="RU">

<head>

<title>JS Array Navigation</title>

<meta charset="utf-8">

<style>

#message {

margin-top: 10px;

}

</style>

</head>

<body>

<h1>JS Array Navigation</h1>

<button id="btnBack">Back</button>

<button id="btnNext">Next</button>

<button id="btnReset">Reset</button>

<button id="btnSetMax">Set Max</button>

<button id="btnShowItem">Show Item</button>

<div id="message"></div>

<script src="app.js"></script>

</body>

</html>

console.log("JS Array Navigation");

// Географические координаты локаций

var coordinates = [

[55.790927, 49.114453], // Location 1

[55.797391, 49.105993], // Location 2

[55.783983, 49.118346] // Location 3

];

// Индекс текущего элемента массива

var currentIndex = 0;

// Кнопки навигации

var btnBack = document.getElementById('btnBack');

var btnNext = document.getElementById('btnNext');

var btnReset = document.getElementById('btnReset');

var btnSetMax = document.getElementById('btnSetMax');

var btnShowItem = document.getElementById('btnShowItem');

// Окно сообщения

var message = document.getElementById('message');

// Функция обновления состояния навигации

function updateNavigationState() {

message.innerHTML = 'Current Index: ' + currentIndex + ', Coordinates: ' + coordinates[currentIndex];

}

// Функция обработки команды Back

function navigateBack() {

if (currentIndex > 0) {

currentIndex--;

updateNavigationState();

}

}

// Функция обработки команды Next

function navigateNext() {

if (currentIndex < coordinates.length - 1) {

currentIndex++;

updateNavigationState();

}

}

// Функция обработки команды Reset

function navigateReset() {

currentIndex = 0;

updateNavigationState();

}

// Функция обработки команды Set Max

function navigateSetMax() {

currentIndex = coordinates.length - 1;

updateNavigationState();

}

// Функция обработки команды Show Item

function showCurrentItem() {

alert('Current Item: ' + coordinates[currentIndex]);

}

// Назначение обработчиков событий для кнопок

btnBack.addEventListener('click', navigateBack);

btnNext.addEventListener('click', navigateNext);

btnReset.addEventListener('click', navigateReset);

btnSetMax.addEventListener('click', navigateSetMax);

btnShowItem.addEventListener('click', showCurrentItem);

// Инициализация состояния навигации

updateNavigationState();

3. Воспроизведите программу из примера 30, проект jsarray05.

Географические координаты локации хранятся

в одномерном и **двумерном** массивах.

Напишите программу на языке JavaScript, в которой задан массив

и выполняется обработка команд навигации по элементам массива.

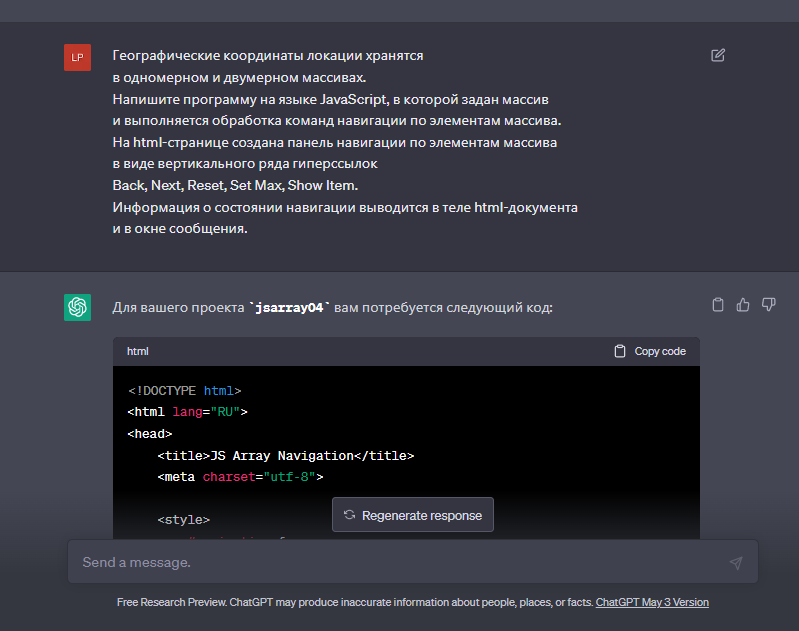
На html-странице создана панель навигации по элементам массива

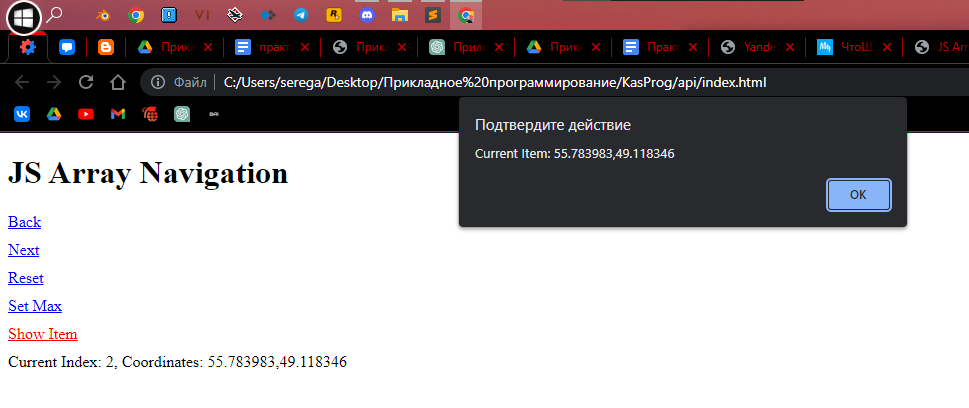
в виде вертикального ряда **гиперссылок**

Back, Next, Reset, Set Max, Show Item.

Информация о состоянии навигации выводится в теле html-документа

и в окне сообщения.





<!DOCTYPE html>

<html lang="RU">

<head>

<title>JS Array Navigation</title>

<meta charset="utf-8">

<style>

#navigation {

display: flex;

flex-direction: column;

gap: 10px;

}

#message {

margin-top: 10px;

}

</style>

</head>

<body>

<h1>JS Array Navigation</h1>

<div id="navigation">

<a href="#" id="linkBack">Back</a>

<a href="#" id="linkNext">Next</a>

<a href="#" id="linkReset">Reset</a>

<a href="#" id="linkSetMax">Set Max</a>

<a href="#" id="linkShowItem">Show Item</a>

</div>

<div id="message"></div>

<script src="app.js"></script>

</body>

</html>

console.log("JS Array Navigation");

// Географические координаты локаций

var coordinates = [

[55.790927, 49.114453], // Location 1

[55.797391, 49.105993], // Location 2

[55.783983, 49.118346] // Location 3

];

// Индекс текущего элемента массива

var currentIndex = 0;

// Ссылки навигации

var linkBack = document.getElementById('linkBack');

var linkNext = document.getElementById('linkNext');

var linkReset = document.getElementById('linkReset');

var linkSetMax = document.getElementById('linkSetMax');

var linkShowItem = document.getElementById('linkShowItem');

// Окно сообщения

var message = document.getElementById('message');

// Функция обновления состояния навигации

function updateNavigationState() {

message.innerHTML = 'Current Index: ' + currentIndex + ', Coordinates: ' + coordinates[currentIndex];

}

// Функция обработки команды Back

function navigateBack(event) {

event.preventDefault();

if (currentIndex > 0) {

currentIndex--;

updateNavigationState();

}

}

// Функция обработки команды Next

function navigateNext(event) {

event.preventDefault();

if (currentIndex < coordinates.length - 1) {

currentIndex++;

updateNavigationState();

}

}

// Функция обработки команды Reset

function navigateReset(event) {

event.preventDefault();

currentIndex = 0;

updateNavigationState();

}

// Функция обработки команды Set Max

function navigateSetMax(event) {

event.preventDefault();

currentIndex = coordinates.length - 1;

updateNavigationState();

}

// Функция обработки команды Show Item

function showCurrentItem(event) {

event.preventDefault();

alert('Current Item: ' + coordinates[currentIndex]);

}

// Назначение обработчиков событий для ссылок

linkBack.addEventListener('click', navigateBack);

linkNext.addEventListener('click', navigateNext);

linkReset.addEventListener('click', navigateReset);

linkSetMax.addEventListener('click', navigateSetMax);

linkShowItem.addEventListener('click', showCurrentItem);

// Инициализация состояния навигации

updateNavigationState();

4. По примерам 27 и 29 из лекции выполните интеграцию

двух программных модулей проектов yapanorama01 jsarray04

в один моудуль, в котором координаты локаций хранятся

в двумерном массиве, переход в плеере панорам к новым локациям

осуществляется с помощью кнопок навигационной панели.