# **Прикладное программирование**

Практика 23

Библиотека API Yandex maps: хранение данных в файлах.

Для каждой программы, состоящей из нескольких файлов,

создавайте отдельную папку проекта, например project23-1,

project23-2 и т.д., где "23" - номер практического

занятия, 1, 2 и т.д. - номер задания.

Каждую программу, состоящую из одного файла, сохранять

в отдельном файле для каждого упражнения,

например prog23-1, prog23-2 и т.д., где "23" - номер практического

занятия, 1, 2 и т.д. - номер задания.

1. Выполните пример 31 из лекции, создайте проект yapanorama02.

Напишите программу на языке JavaScript с применением

библиотеки API Yandex maps, которая позволяет отображать

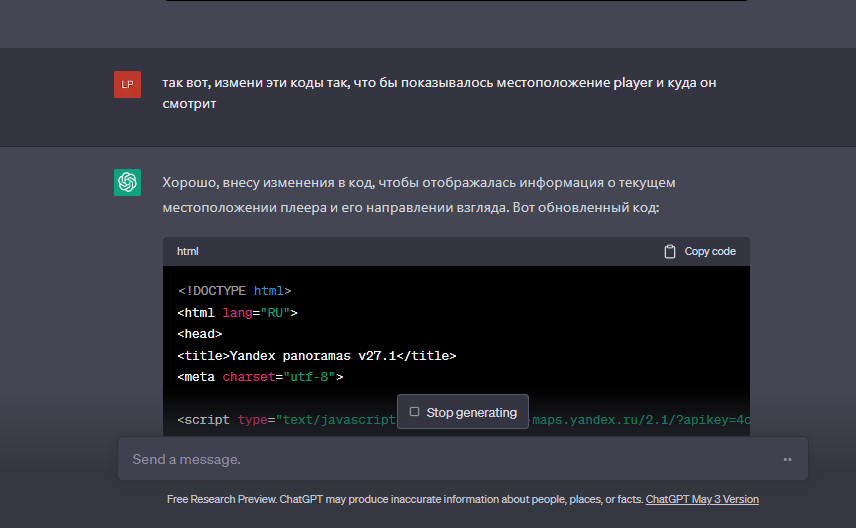
панораму локации с помощью сервиса Яндекс панорамы.

В программе отслеживаются события смены локации и направления взгляда

в плеере панорамы. При переходе в новую локацию координаты заносятся

в двумерный массив. На снимках экрана отобразите работу с консолью

браузера.



<!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>

console.log("Test Yandex panorama v27.1");

var x0 = 55.790927;

var y0 = 49.114453;

var x1 = 55.797391;

var y1 = 49.105993;

var player;

ymaps.ready(init);

function init() {

console.log("init: called");

ymaps.panorama.locate([x0, y0]).done(

function(panoramas) {

console.log("panoramas: " + panoramas);

console.log("panoramas[0]: " + panoramas[0]);

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

showPlayerLocation();

},

function() {}

);

}

function onClickNext() {

console.log("Clicked: Next");

player.moveTo([x1, y1]);

showPlayerLocation();

}

function showPlayerLocation() {

var location = player.getPosition();

var direction = player.getDirection();

var message = "Player Location: " + location + ", Direction: " + direction;

console.log(message);

document.getElementById("player-location").textContent = message;

}

</script>

<style>

html, body {

width: 100%;

height: 100%;

margin: 0;

padding: 0;

}

div.player {

padding: 0;

margin: 0;

height: 75%;

width: 100%;

border: solid 2px red;

}

</style>

</head>

<body>

<h1>Yandex Panorama v.27.1</h1>

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

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

<p id="player-location"></p>

</body>

</html>

console.log("Test Yandex panorama v27.1");

var x0 = 55.790927;

var y0 = 49.114453;

var x1 = 55.797391;

var y1 = 49.105993;

var player;

ymaps.ready(init);

function init() {

console.log("init: called");

ymaps.panorama.locate([x0, y0]).done(

function(panoramas) {

console.log("panoramas: " + panoramas);

console.log("panoramas[0]: " + panoramas[0]);

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

showPlayerLocation();

},

function() {}

);

}

function onClickNext() {

console.log("Clicked: Next");

player.moveTo([x1, y1]).then(function() {

showPlayerLocation();

});

}

function showPlayerLocation() {

player.getPosition().then(function(position) {

player.getDirection().then(function(direction) {

var message = "Player Location: " + position + ", Direction: " + direction;

console.log(message);

document.getElementById("player-location").textContent = message;

});

});

}

4. Напишите программу на языке JavaScript, которая автоматически

генерирует имя файла, увеличивая счетчик. Счетчик - число,

которое является частью имени файла, например 001track.txt,

002track.txt, 003track.txt и т.д. Информация о последнем номере,

сгенерированным программой для имени файла, хранится

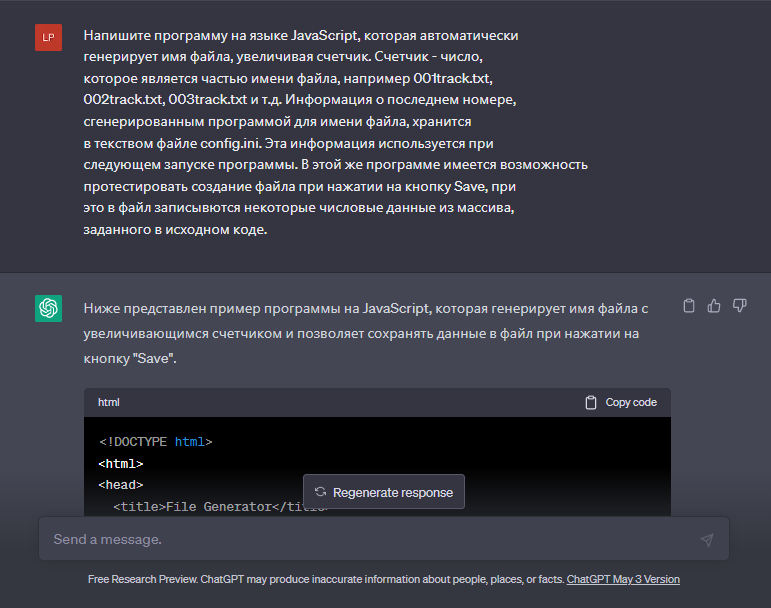
в текством файле config.ini. Эта информация используется при

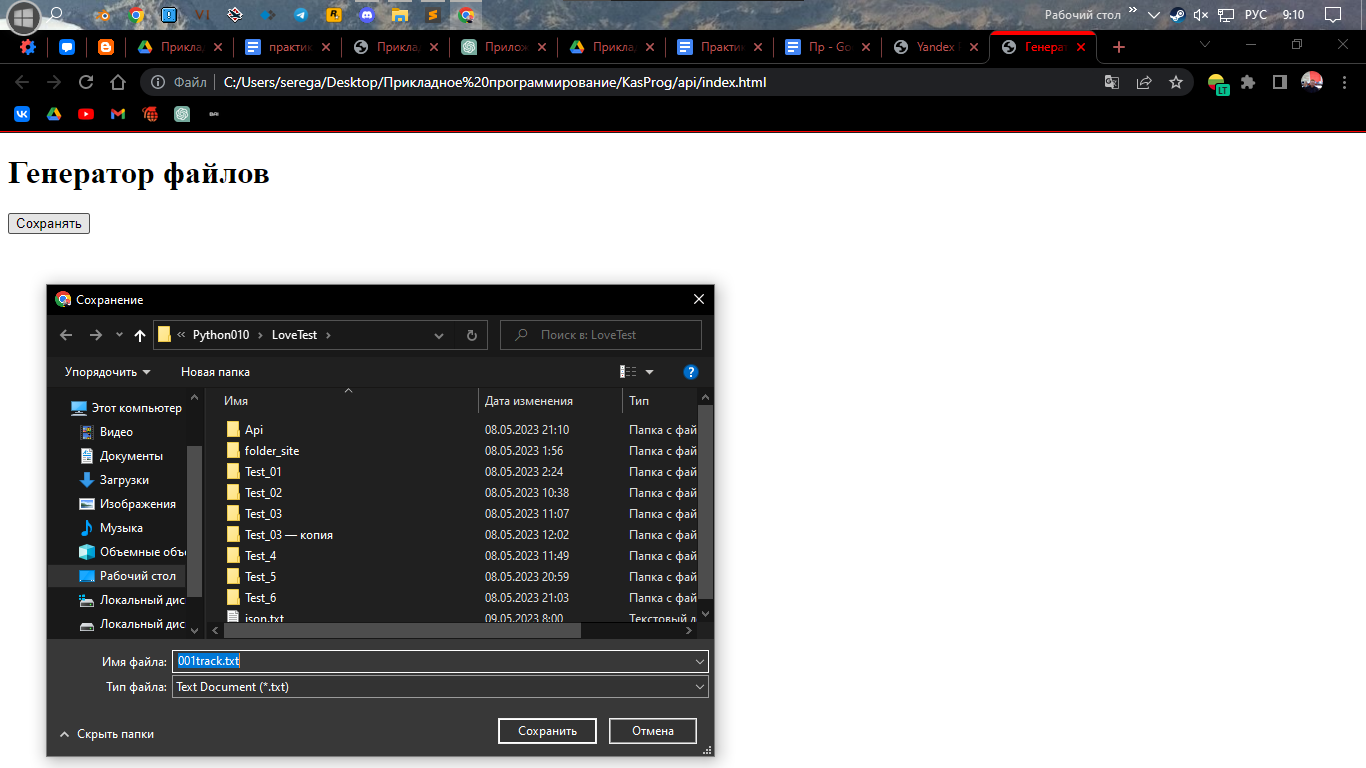
следующем запуске программы. В этой же программе имеется возможность

протестировать создание файла при нажатии на кнопку Save, при

это в файл записывются некоторые числовые данные из массива,

заданного в исходном коде.



<!DOCTYPE html>

<html>

<head>

<title>File Generator</title>

<meta charset="UTF-8">

</head>

<body>

<h1>File Generator</h1>

<button onclick="generateFile()">Save</button>

<script>

// Имя файла и счетчик

var fileName = '';

var counter = 0;

// Функция для генерации имени файла с учетом счетчика

function generateFileName() {

var number = counter.toString().padStart(3, '0');

fileName = number + 'track.txt';

}

// Функция для увеличения счетчика и обновления имени файла

function incrementCounter() {

counter++;

generateFileName();

}

// Функция для сохранения данных в файл

function saveDataToFile() {

// Предположим, что у нас есть массив с числовыми данными

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

// Создаем ссылку на файл

var link = document.createElement('a');

link.href = 'data:text/plain;charset=utf-8,' + encodeURIComponent(data.join('\n'));

link.download = fileName;

link.style.display = 'none';

// Добавляем ссылку на страницу и эмулируем нажатие на ссылку для сохранения файла

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

// Увеличиваем счетчик

incrementCounter();

}

// Функция вызывается при нажатии на кнопку Save

function generateFile() {

saveDataToFile();

console.log('File saved:', fileName);

}

</script>

</body>

</html>