РЕФЕРАТ

Пояснительная записка: 89 с., 23 рис., 3 табл., 56 источников, 6 прил.

В данной работе будет рассмотрена разработка сайта на тему «Советская мототехника», который будет предоставлять краткую информацию о выбранной модели мото, а также фото. Сайт создан с помощью технологий HTML и CSS. Страница разделена на фреймовую структуру, которая добавляет динамичности просматриваемому содержимому.

Для примера построения динамического сайта была создана игра «Червяк и яблоки» с использованием скриптового языка JavaScript и тега <canvas>. В игре дана возможность управлять червем, который поедает яблоки, увеличиваясь в размерах. Конец игры наступает при столкновении червя со своим телом.

HTML, CSS, JAVASCRIPT, CANVAS, ИГРА, САЙТ, ФРЕЙМОВАЯ СТРУКТУРА, ДИНАМИЧЕСКИЙ САЙТ, МОТО СССР

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc70663961)

[1 ЯЗЫКИ ПРОГРАММИРОВАНИЯ 5](#_Toc70663962)

[1.1 Языки программирования клиентской части сайта 5](#_Toc70663963)

[1.2 Языки программирования серверной части сайта 6](#_Toc70663964)

[2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ САЙТА 8](#_Toc70663965)

[2.1 Описание предметной области 8](#_Toc70663966)

[2.2 Фреймовая структура сайта 9](#_Toc70663967)

[2.3 Совместное использование HTML и JavaScript. 9](#_Toc70663968)

[ЗАКЛЮЧЕНИЕ 11](#_Toc70663969)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc70663970)

[Приложение А 13](#_Toc70663971)

[Приложение Б 24](#_Toc70663972)

# ВВЕДЕНИЕ

HTML, CSS и JavaScript – это базовые инструменты для построения динамического клиента сайта. В данной работе задействованы их основные возможности для создания сайта про мото СССР и игры «Червяк и яблоки». Теги – это то, из чего состоит язык HTML. Они являются своеобразными командами, которые преобразовываются в визуальные объекты в браузере пользователя. Например, тег <img> используется для размещения изображений на странице. У него есть обязательный атрибут src, в котором указывается ссылка на файл. Простыми словами технология HTML необходима для разметки текста в веб-документе на логические разделы и создания его структуры. Иерархия тегов позволяет удобно и эффективно манипулировать содержимым из сторонних инструментов, например, обращаться к определенному участку текста для изменения цвета. В данной работе с помощью HTML была построена фреймовая структура, а также задействован тег canvas для поэтапной отрисовки динамической страницы игры.

CSS используется создателями веб-страниц для задания представления внешнего вида этих веб-страниц. В работе язык был использован для изменения цветовой палитры сайта, установки пользовательских шрифтов, размещения фото и размера шрифта. Отдельно внимание стоит уделить созданию адаптивности сайта под различные устройства с помощью CSS. Сайт принимает разные вариации внешнего вида в зависимости от дисплея на котором он отображается. Это позволяет предоставить пользователю сайт вне зависимости от используемого им устройства. Сайт хорошо отображается как на мобильных устройствах, так и на больших дисплеях.

Основная задача JavaScript создавать последовательность действий, которые будут приводить к определенному результату. В работе язык был применен для создания логики игры: управление червём, поедание яблок, увеличение червя, реакция на столкновения и прочее.

# 1 ЯЗЫКИ ПРОГРАММИРОВАНИЯ

## 1.1 Языки программирования клиентской части сайта

В работе был использован язык разметки HTML для создания структурированного документа сайта, CSS для оформления документа и JavaScript для создания динамических элементов страницы.

Однако для создания динамических веб-страниц существует множество других языков.

Для создания логики клиентской части веб-приложения используются:

* HTML − расшифровывается как HyperText Markup Language (язык гипертекстовой разметки) стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.
* CSS (Cascading Style Sheets, каскадные таблицы стилей) — CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы. Каскадные таблицы стилей позволяют работать со шрифтами, создавать сложные визуальные эффект и даже последовательную анимацию.
* JavaScript − язык интерфейса, используемый для создания и разработки веб-сайтов, настольных приложений и игр. JavaScript работает во всех браузерах и может работать с программами, которые не размещены в Интернете. Он поддерживает как функциональные, так и объектно-ориентированные стили программирования, и в основном, это ваш подход к созданию потрясающих пользовательских интерфейсов и веб-сайтов / приложений / игр, которые выглядят супер круто. Понимание JavaScript важно, даже если ваше сердце настроено на развитие серверной части. Компоненты, структуры данных и алгоритмы JavaScript применяются практически к любому другому языку.

## 1.2 Языки программирования серверной части сайта

PHP - это скриптовый язык, используемый для быстрого создания динамических веб-страниц. Отличный выбор для frontend и backend разработчиков, чтобы добавить их в арсенал (но особенно для последних), он стоит за такими веб-гигантами, как WordPress и Facebook. PHP позволяет быстро и легко расширять веб-приложения и запускать веб-сайты с повторяющимися серверными задачами (например, обновлять новостные ленты). Он имеет открытый исходный код и очень популярен в среде начинающих компаний, медиа-агентств и электронной коммерции - таких людей, которые часто нанимают новых веб-разработчиков.

Python – язык чрезвычайно прост в освоении и является динамичным, универсальным языком. Хотя он более популярен как язык на заднем плане, с ним можно делать практически все, что вы хотите. Разработанный с целью быть читабельным, простым и, самое главное, забавным, это новый фаворит разработчиков во всех областях индустрии.

Go – рассчитанный на высоконагруженные крупные проекты, поддерживает типобезопасность, возможность динамического ввода данных, а также содержит богатую стандартную библиотеку функций и встроенные типы данных вроде массивов с динамическим размером и ассоциативных массивов. С помощью механизмов многопоточности Go упрощает распределение вычислений и сетевого взаимодействия, а современные типы данных открывают программисту мир гибкого и модульного кода. Программа быстро компилируется, при этом есть сборщик мусора и поддерживается рефлексия.

Ruby — динамический, интерпретируемый язык высокого уровня с открытым исходным кодом. В языке нет строгих правил, ограничивающих возможности и подходы к решению задач. Он обладает элегантным синтаксисом, который приятно читать и легко писать. Рассчитан на малые и средние проекты.

Java – объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems. Изначально новый язык программирования назывался Oak и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения. Сегодня технология используется для создания крупных проектов с длительной поддержкой.

C# – язык программирования С# был разработан группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft. После стал универсальным инструментом в том числе и для веб разработки.

# 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ САЙТА

## 2.1 Описание предметной области

В советское время невероятно стильным и модным личным автотранспортом являлись мопеды. Особенно среди юношества. Очередь на приобретения автомобиля длилась годами, цены кусались. Мотоцикл также стоил дорого и все же требовал гаражного хранения. А мопед, подобно велосипеду, зачастую завозили в квартиру.

На вождение мопеда не требовались права. Стоили мопеды в СССР от 100 рублей.

Продажа мопедов в осуществлялась в магазинах мототехники, почти всегда по предварительной записи. Отмечаться в очереди за мопедом требовалось каждое утро в течение месяца-полугода.

Мопеды производились на Пензенском велосипедном заводе (легкий мопед ЗИФ), Львовском мотозаводе (мопед Верховина, мокик Карпаты), Рижском мотозаводе (мопед Рига, мокик Дельта)

Первый советский мотовелосипед выпустили в 1958 году на Рижском мотозаводе «Саркана Звайгзне»: Рига-16.

Модель оказалась не очень удачной. После практики на чешском мотозаводе JAWA в шестидесятые годы начался серийный выпуск первых мопедов в СССР — Рига-1.

Мопеды Рига постоянно модернизировались. В фильме Приключения Электроника Сыроежкин рассекал именно на мопеде RIGA.

Рижский завод «Саркана Звайгзне» в 1966 году начал производство легкого мопеда «Рига-5» с двигателем Д-5 мощностью 1,2 л. с. У машины очень простая ходовая часть. Механические тормоза мопеда гарантируют быстрое торможение и безаварийную езду. Управление тормозом переднего колеса и дросселем установлено на правой стороне руля, рычаг сцепления — на левой.

## 2.2 Фреймовая структура сайта

Frame — это отдельная область окна, являющаяся по своей структуре законченным документом HTML. Понятие пришло к нам из английского языка и переводится как «рамка». Фрейм — это структура, разделяющая окно браузера на отдельные участки и содержащая в себе самостоятельную и полноценную страницу. Данная технология программирования продолжает вызывать много споров, хотя пик ее популярности прошел из-за появления проблем с удобством сайта. Положительные стороны использования это экономия трафика за счет обновления только одной страницы, новые интересные возможности навигации по странице, вариативность размеров их участков.

С помощью тегов <frameset> и <frame> была сделана фреймовая структура сайта с целью визуально разделить компоненты сайта и расширить функционал интерфейса. Сайт состоит из фреймов:

* «Меню» — предоставляет навигацию по сайту и раздел с информацией об авторе;
* «Контент» — представляет фото выбранного мопеда и краткую информация о нем;
* «Топ» — отображает шапку сайта, при нажатии возвращает на главную страницу;

## 2.3 Совместное использование HTML и JavaScript.

Совместное использование этих двух технологий заключается в том, что JavaScript принимает события, которые происходят на странице HTML, обрабатывает данные нужным образом и возвращает результат. Этим результатом может быть все что угодно, как число, так и новая анимация, а возможно новый созданный тег в документе.

Особенностью использования является необходимость явно указать в веб-документе используемый скрипт. Делается это в теге <script> с указанием пути к скрипту в атрибуте src. Подключение желательно выполнять в конце документа для ускорения загрузки содержимого сайта, так как на загрузку скрипта может понадобиться несколько секунд, что может создать негативное впечатление посетителя.

Для создание игры «Червяк и яблоки» были задействованы возможности работы с графикой HTML. Был создан тег <canvas>, который создает область, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства. При помощи тега <canvas> можно создавать рисунки, анимацию, покадровую отрисовку изображения. JavaScript был использован для создания логики отрисовки нового кадра, например, при нажатии определенной кнопки, экран перерисовывался с соответствующим контекстом или для создания логики взаимодействия червя и яблока на основе координатной системы.

# ЗАКЛЮЧЕНИЕ

Была спроектирована фреймовая структура и разработан сайт о советских мопедах, представляющий краткую информацию с изображениями для каждой модели. При разработке были получены практические навыки создания структуры веб-документа, создания цветовой палитры сайта и разработки динамических страниц.

Средствами HTML, CSS и JavaScript была создана игра «Червяк и яблоки», которая задействует основные возможности работы с изображением через тег <canvas> и использует простые конструкции языка JavaScript для задания логики обновления кадров. В игре пользователь может управлять червем, направляя его к случайно появляющимся на поле яблокам, поедая их червь растет в размерах. При передвижении он не должен сталкиваться с собой или границами игрового поля.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Справочник HTML. [Электронный ресурс] – Режим доступа: http:// htmlbook.ru
2. Электронный учебник JavaScript. [Электронный ресурс] – Режим доступа: https:// learn.javascript.ru/
3. Дэвид Макфарланд. "Новая большая книга CSS"/Дэвид Макфарланд . – Город: Москва, 2016, 256 с.

# Приложение А

ЛИСТИНГ

Страница с контактами

!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Обо мне</title>

</head>

<style>

.page{

background:#c0c0c0;

padding-left: 3%;

padding-top:3%

}

</style>

<div >

<body class ="page" >

<h1 align = "center">Ржавая молодость</h1>

<h2 align = "center">О создателе сайта</h2>

<p><font size = "5" face="Times New Roman">Привет,утомленный путник, ты оказалася на сайте "Ржавая молодость".

Здесь ты узнаешь много интересных фактов о советском мотопроме. </font></p>

<dl><font size = "4" face ="Times New Roman">

<dt> <b> Фио автора: </b> </dt>

<dd>Стрилец Илья Владимирович</dd>

<dt> <b> Адрес: </b> </dt>

<dd>Москва, Большая Садовая, 302-бис, пятый этаж, кв. № 50.</dd>

<dt> <b> Телефон:</b> </dt>

<dd>8-800-555-35-35</dd>

<dt> <b> E-mail: </b> </dt>

<dd>strilets.piupiu@gmail.com</dd>

<dt> <b> Интересы: </b> </dt>

<dd>

Мой список интересов достаточно обширен.

Я интересуюсь компьютерными играми, геймдевом, компьютерными комплектующими, да и техникой в целом, активно слежу за новинками в мире компьютерных технологий.

Люблю смотреть фильмы и сериалы.

Периодически читаю книги, люблю классику, иногда читаю комиксы или мангу.

Уважаю рок-музыку.

Увлекаюсь сборкой советской мототехники, которая является тематикой моего сайта.

</dd>

<dt> <b>Сайт</b> </dt>

<dd> <a href ="https://vk.com/ilyishka\_piu\_piu" target = "\_blank" > Вконтакте </a> <br>

<a href = "https://www.instagram.com/ilyishka.piu.piu/" target = "blank"> Инстаграм</a>

</dd>

</dl>

</font>

</body>

</div>

</html>

Код сайта

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Фреймы</title>

</head>

<frameset rows="20%,\*">

<frame src="top.html" name="TOP" scrolling="no" noresize>

<frameset cols="15%,\*">

<frame src="menu.html" name="MENU">

<frame src="content.html" name="CONTENT">

</frameset>

</frameset>

</html>

menu.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8"

</head>

<body bgcolor="#c0c0c0">

<h2>Меню сайта</h2>

<ul>

<li><a href = "verhovina.html" target = "CONTENT"> Верховина-6</a> </li>

<li><a href = "karpati.html" target = "CONTENT"> Карпаты 2 </a></li>

<li><a href = "riga-13.html" target = "CONTENT"> Рига-13</a></li>

</ul>

</body>

</html>

content.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<h1 >Главная страница</h1>

<img src ="pes.jpg" width="700" height="500">

<p align = "center">

Здравствуйте! Вас приветствует "Ржавая молодость".

Мы зона свободного общения людей, увлечение которых

- малокубатурная мототехника и техническое творчество.

Развитие мопедного сообщества в сети и в оффлайне

- наша главная задача. Рады видеть всех, от скутеристов

до любителей двухколёсного антиквариата.

Добро пожаловать!

</p>

</body>

</html>

top.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body bgcolor="#342c1f">

<div class = "image-center">

<img src = "header.jpg" width = "460" height="255">

</div>

</body>

</html>

verhovina.html

<!DOCTYPE html>

<html>

<head>

<meta charset = "utf-8">

</head>

<body>

<h1 align = "center">Верховина-6</h1>

<img src = "verhovina.jpg" width="650" height="450">

<p>

Модель ЛМЗ-2.158, ЛМЗ-2.158-1. Производилась с 1977 года.<br><br>

Новые накладки на бак, изменённое седло, новые крылья (на более поздних моделях),

новые боковые крышки, двигатели Ковровского мотозавода Ш-57, позднее Ш-58, оснащённый

кикстартером или двигатель S-58 Шяуляйского завода.Начиная с этой модели качество этих мопедов

(мокиков как принято было писать в журнале «За рулём») сильно ухудшилось, что впоследствии привело

к затовариванию и падению спроса на этот вид мототехники. В розничной продаже «Верховина-6» стоила 220 рублей

с крашеными крыльями и ободами колес, 223 рубля с хромированными ободами и 226 рублей с хромированными крыльями и ободами.<br><br>

Двигатель Ш-57 мопеда «Верховина»

Считается самым быстрым мокиком СССР.<br><br>

</p>

</body>

</html>

karpati.html

<!DOCTYPE html>

<html>

<head>

<meta charset = "utf-8">

</head>

<body>

<h1 align = "center">Карпаты-2</h1>

<img src = "karpati.jpg" width="650" height="450">

<p>

Мопед «Карпаты» появился в СССР весной 1981 года.

И почти сразу стал одним из самых популярных транспортных средств того времени.

За время своей жизни он успел претерпеть несколько модернизаций,

которые исправили недочеты первой модели и улучшили его,

как мобильное средство. <br><br>

Производителем данного малого средства передвижения стал украинский завод мотоциклов во Львове.

Прототипом мопеда стала серия мокиков «Верховина».

В 1981 году завод изготавливает первую модификацию данного транспортного средства под названием «Карпаты 1».

Спустя три года в производство запускаются следующие серии мокиков этой марки – «Карпаты 2».

Эта серия была уже модифицированной и улучшенной по сравнению с первым выпуском.

</p>

</body>

</html>

riga-13.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

</head>

<body>

<h1 align = "center"> Рига-13</h1>

<img src = "riga-13.jpg" width = "650" height="450">

<p>Мотозавод «Саркана Звайгзне» в 1983 году приступил к серийному выпуску легкого мопеда «Рига-13»

, заменившего популярную модель «Рига-11». Как и все мопеды этого типа, он не имел коробки

передач и был оснащен вспомогательным педальным приводом. Новая модель отличалось от предыдущей

и внешне, и улучшенными техническими параметрами. Ее важнейшее преимущество — увеличенная

на 0,3 л. с. мощность двигателя — было достигнуто в основном благодаря новой системе

выпуска — выхлопной трубе и глушителю. Это помогло поднять динамические качества машины,

но максимальная скорость была сохранена прежней. Существенно увеличена (с 4 до 5,5 л)

была вместимость топливного бака С заднего колеса он был перенесен на переднюю часть рамы,

что улучшило подачу топлива, дало возможность до конца использовать его запас. Были усилены

механизм привода заднего тормоза и некоторые детали передачи на заднее колесо.

Взамен хребтовой применена рама закрытого типа. Благодаря этому мопед стал на 2 кг легче.

Мопед комплектовался зеркалом заднего вида. «Рига-13» производилась до 1998 года,

это самая массовая модель Рижского мотозавода.

</p>

</body>

</html>

Код JavaScript выполняющий арифметические действия

Текст HTML Документа

<!DOCTYPE html>

<html>

<head>

<meta charset = "utf-8">

</head>

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

<body>

</body>

</html>

Код JavaScript

x = prompt("Введите x")

result = (x \* x) -( 7 \* x) + (10 / (x \* x))- (8\*x) + 12;

alert(result)

Код игры написанной на JavaScript

Текст HTML Документа

!DOCTYPE html>

<html>

<head>

<meta charset = "UTF-8">

<meta http-equiv= "x-ua-compatible" content= "ie=edge">

<meta name= viewport content= "width=device-width initial-scale=1">

<title>Змейка на JavaScript</title>

<style>

body{

margin: 0;

}

#canvas{

display:block;

margin:0 auto;

background-color: black;

}

</style>

</head>

<body>

<canvas id ="canvas" width = "400" height="400"></canvas>

<script src = “game.js”></script>

</body>

</html>

Код JavaScript

let canvas = document.querySelector('#canvas')

let ctx = canvas.getContext('2d')

let width = canvas.width

let height = canvas.height

let score = 0

let blockSize = 10

let widthInBlock = width / blockSize

let heightInBlock = height / blockSize

// добавляем звук

let audioEat = new Audio('music/carrot\_chew\_munch\_001\_28556.mp3')

let audioGameOver = new Audio('music/583699913672e65.mp3')

// Рисуем круг

function circle(centerX, centerY, radius) {

ctx.beginPath()

ctx.arc(centerX, centerY, radius / 2, 0, Math.PI \* 2, false)

ctx.fill()

}

// Создаем счетчик

function drawScore() {

ctx.font = '20px Courier'

ctx.fillStyle = 'white'

ctx.textBaseline = 'top'

ctx.fillText('Счёт: ' + score, blockSize, blockSize)

}

// Создаем конец игры

function gameOver() {

playing = false

ctx.font = '30px Courier'

ctx.fillStyle = 'white'

ctx.textAlign = 'center'

ctx.textBaseline = 'middle'

ctx.fillText('Конец игры', width / 2, height / 2)

}

// сетка

function Block(col, row) {

this.col = col

this.row = row

}

Block.prototype.drawSquare = function(color) {

let x = this.col \* blockSize

let y = this.row \* blockSize

ctx.fillStyle = color

ctx.fillRect(x, y, blockSize, blockSize)

}

Block.prototype.drawCircle = function(color) {

let x = this.col \* blockSize + blockSize / 2

let y = this.row \* blockSize + blockSize / 2

ctx.fillStyle = color

circle(x, y, blockSize)

}

Block.prototype.equal = function(otherBlock) {

return this.col === otherBlock.col && this.row === otherBlock.row

}

let block = new Block()

function Snake() {

this.snakeBody = [

new Block(10, 7),

new Block(9, 7),

new Block(8, 7)

]

this.direction = 'right'

this.nextDirection = 'right'

}

Snake.prototype.drawSnake = function() {

for (let i = 0; i < this.snakeBody.length; i++) {

this.snakeBody[i].drawSquare('green')

}

}

Snake.prototype.move = function() {

let head = this.snakeBody[0]

let newHead

this.direction = this.nextDirection

if (this.direction === 'right') {

newHead = new Block(head.col + 1, head.row)

} else if (this.direction === 'left') {

newHead = new Block(head.col - 1, head.row)

} else if (this.direction === 'up') {

newHead = new Block(head.col, head.row - 1)

} else if (this.direction === 'down') {

newHead = new Block(head.col, head.row + 1)

}

this.checkSelfCollision(newHead)

this.snakeBody.unshift(newHead)

if (newHead.equal(apple.applePosition)) {

score++

if (animationTime > 40) {

animationTime -= 5

}

apple.move(this.snakeBody)

audioEat.play()

} else if (newHead.equal(apple.applePosition2)) {

score += 2

if (animationTime > 40) {

animationTime -= 5

}

apple.move2(this.snakeBody)

audioEat.play()

} else {

this.snakeBody.pop()

animetionTimeBoolean = false

}

}

Snake.prototype.checkSelfCollision = function(newHead) {

for (let i = 1; i < this.snakeBody.length; i++) {

if(newHead.equal(this.snakeBody[i])) {

gameOver()

audioGameOver.play()

}

}

}

Snake.prototype.checkCollision = function() {

for (let i = 0; i < this.snakeBody.length; i++) {

if (this.snakeBody[i].col < 0) {

this.snakeBody[i] = new Block(this.snakeBody[i].col + 40, this.snakeBody[i].row)

} else if (this.snakeBody[i].row < 0) {

this.snakeBody[i] = new Block(this.snakeBody[i].col, this.snakeBody[i].row + 40)

} else if (this.snakeBody[i].row > widthInBlock) {

this.snakeBody[i] = new Block(this.snakeBody[i].col, this.snakeBody[i].row - widthInBlock - 1)

} else if (this.snakeBody[i].col > heightInBlock) {

this.snakeBody[i] = new Block(this.snakeBody[i].col - heightInBlock - 1, this.snakeBody[i].row)

}

}

}

Snake.prototype.setdirection = function(newDirection) {

if (this.direction === 'right' && newDirection === 'left') {

return

} else if (this.direction === 'left' && newDirection === 'right') {

return

} else if (this.direction === 'up' && newDirection === 'down') {

return

} else if (this.direction === 'down' && newDirection === 'up') {

return

}

this.nextDirection = newDirection

}

function Apple() {

this.applePosition = new Block(20, 20)

this.applePosition2 = new Block(15, 15)

}

Apple.prototype.appleDraw = function() {

this.applePosition.drawCircle('red')

this.applePosition2.drawCircle('white')

}

Apple.prototype.move = function(busyBlock) {

let randomCol = Math.floor(Math.random() \* widthInBlock)

let randomRow = Math.floor(Math.random() \* heightInBlock)

this.applePosition = new Block(randomCol, randomRow)

this.applePosition.drawCircle('red')

for (let i = 0; i < busyBlock.length; i++) {

if (busyBlock[i].equal(this.applePosition)) {

this.move()

}

}

}

Apple.prototype.move2 = function(busyBlock) {

let randomCol2 = Math.floor(Math.random() \* widthInBlock)

let randomRow2 = Math.floor(Math.random() \* heightInBlock)

this.applePosition2 = new Block(randomCol2, randomRow2)

this.applePosition2.drawCircle('white')

for (let i = 0; i < busyBlock.length; i++) {

if (busyBlock[i].equal(this.applePosition2)) {

this.move2()

}

}

}

let snake = new Snake()

let apple = new Apple()

let playing = true

let animationTime = 100

function gameLoop() {

ctx.clearRect(0, 0, width, height)

drawScore()

snake.move()

snake.checkCollision()

snake.drawSnake()

apple.appleDraw()

if (playing) {

setTimeout(gameLoop, animationTime)

}

}

gameLoop()

// let setIntervalId = setInterval(function() {

// ctx.clearRect(0, 0, width, height)

// drawScore()

// snake.move()

// snake.checkCollision()

// snake.drawSnake()

// apple.appleDraw()

// }, 100);

let directions = {

37: 'left',

38: 'up',

39: 'right',

40: 'down'

}

window.addEventListener('keydown', (event) => {

let newDirection = directions[event.keyCode]

if (newDirection !== undefined) {

this.direction = snake.setdirection(newDirection)

}

})

# Приложение Б

ЭКРАННЫЕ ФОРМЫ

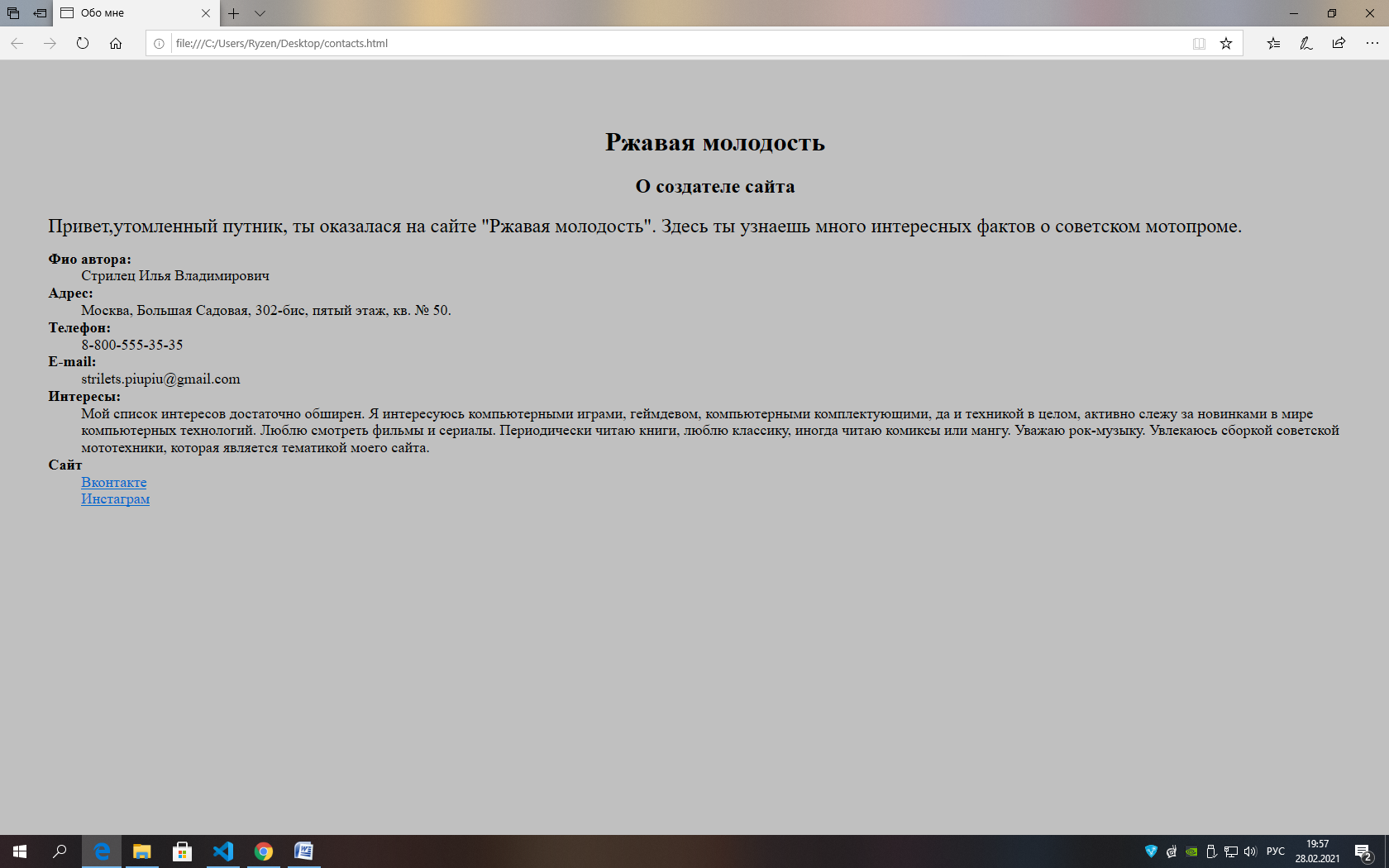


Рисунок Б.1 – Документ contacts.html

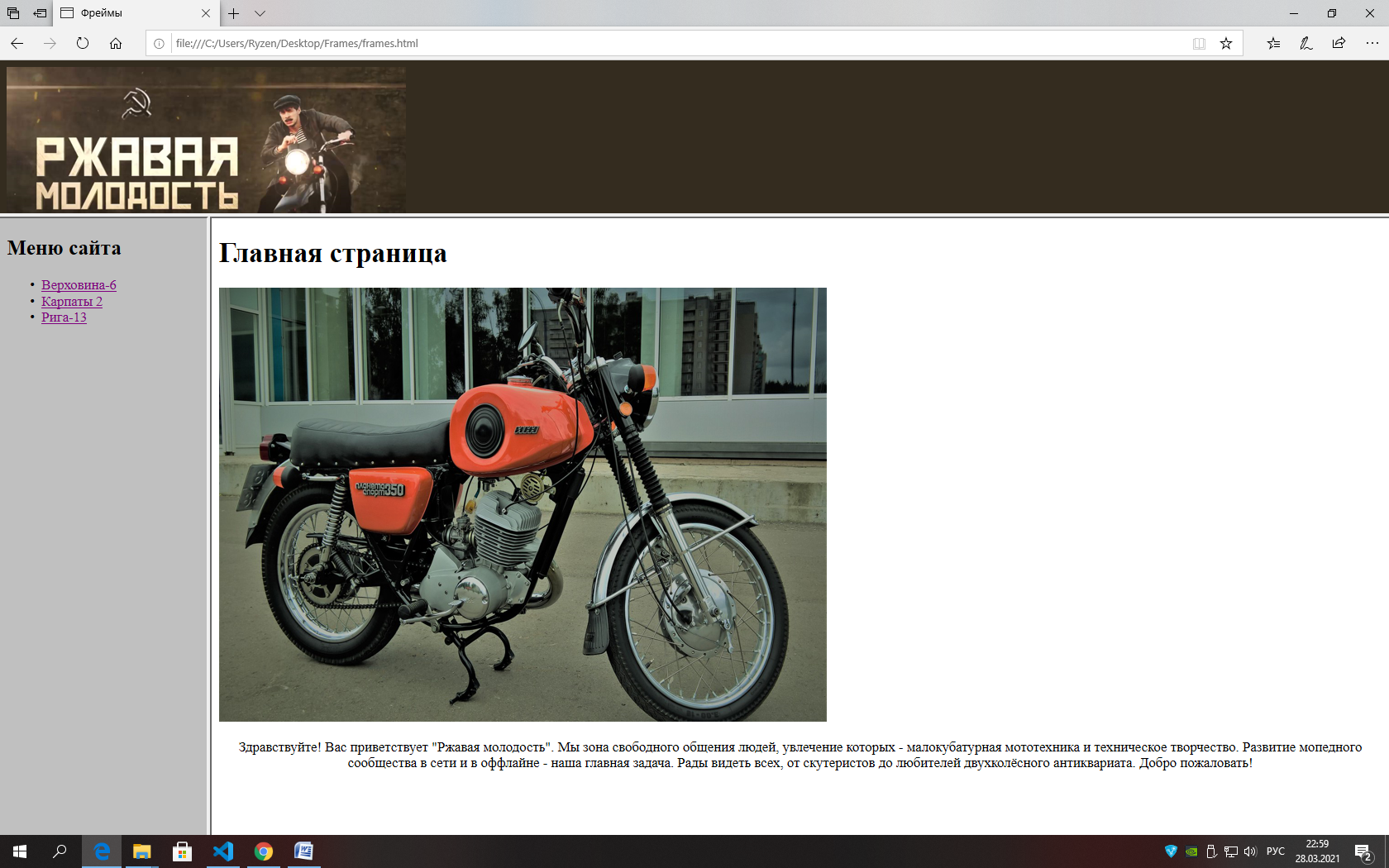


Рисунок Б.2 – Главная страница сайта

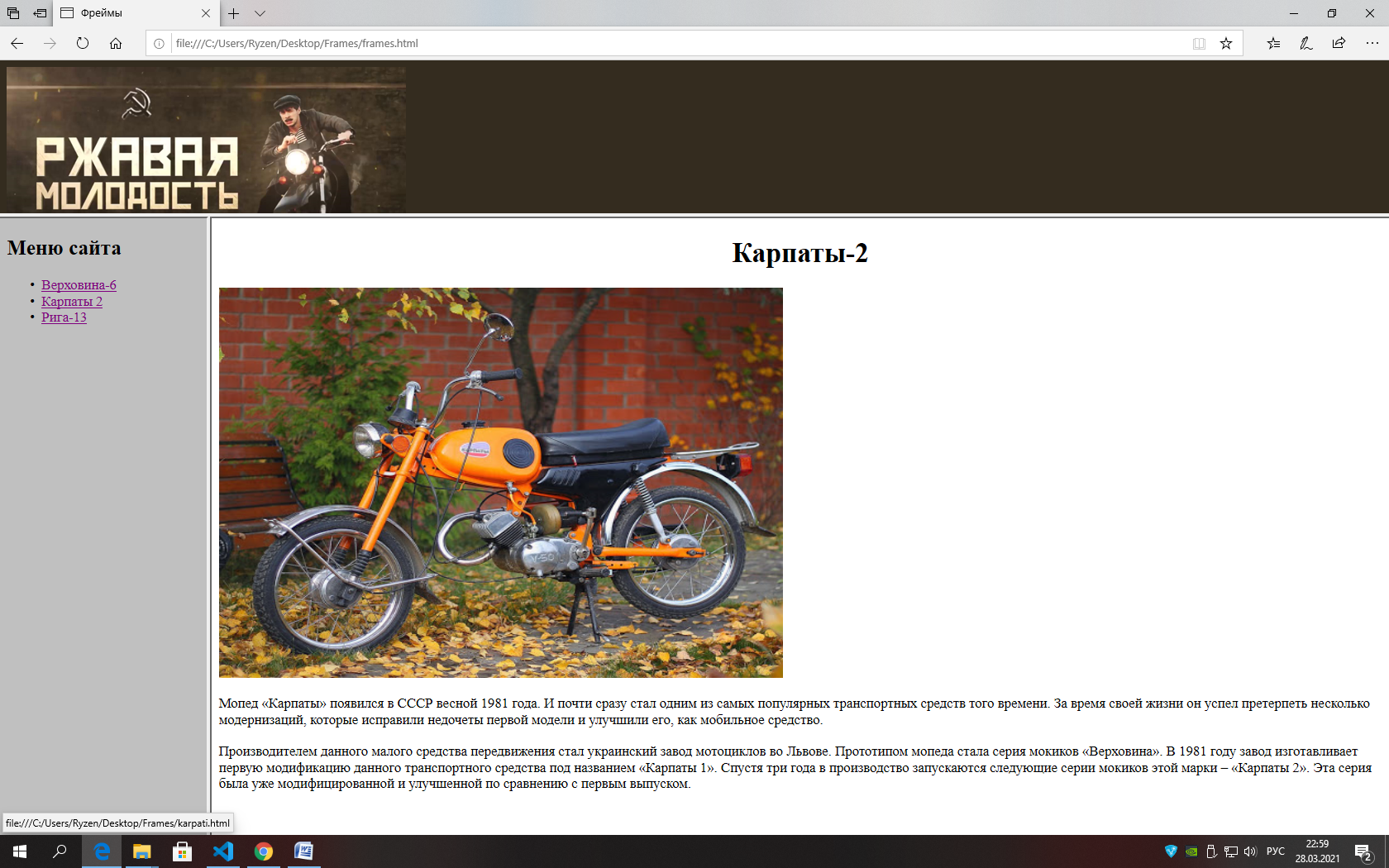


Рисунок Б.3 – Страница посвященная мопеду Карпаты-2

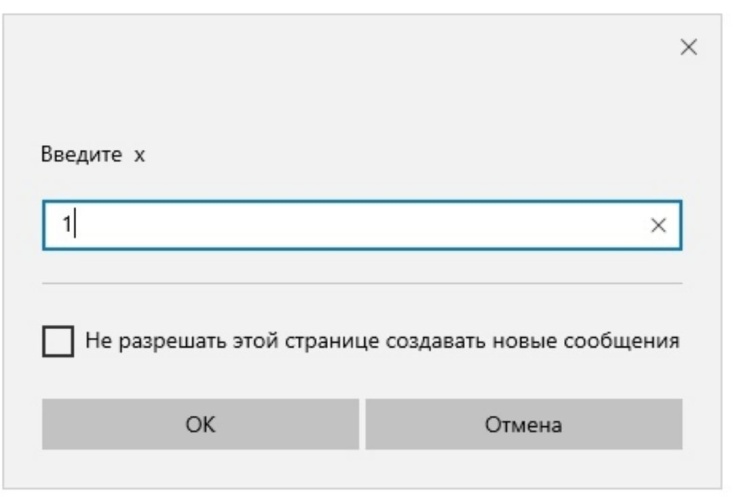


Рисунок Б.4 –Арифметические действия – Ввод

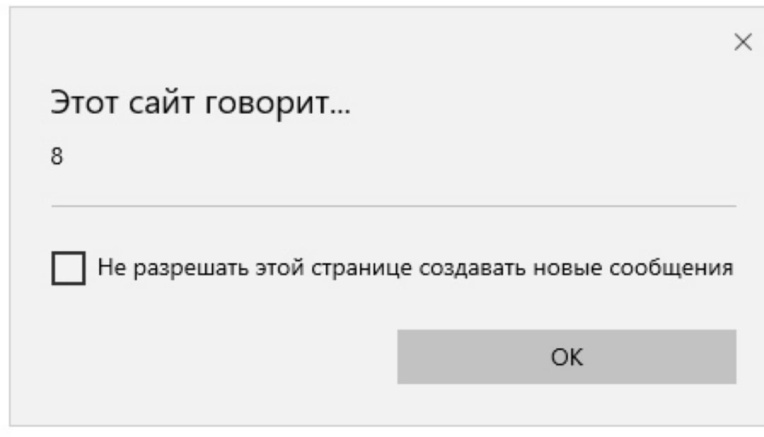


Рисунок Б.5 – Арифметические действия – Вывод

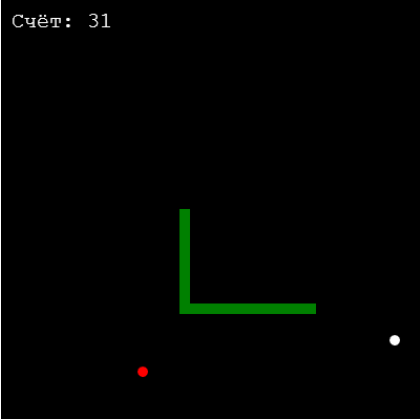


Рисунок Б.6 – Процесс игры – Геймплей

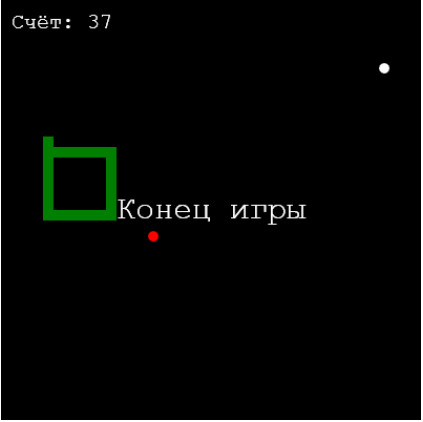


Рисунок Б.6 – Процесс игры – Конец игры