

Колледж космического машиностроения и технологий

КУРСОВОЙ ПРОЕКТ

По МДК.01.02 «Прикладное программирование»

Тема: «Разработка веб-приложения «текстовый редактор» на JavaScript»

Выполнил студент

Пилипушко Андрей Сергеевич

Группа П2-18

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Проверил преподаватель

Гусятинер Леонид Борисович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Оценка)

Королёв 2021 г.

Оглавление

[**Введение** 3](#_Toc75368423)

[**1.** **Теоретическая часть** 4](#_Toc75368424)

[**1.1.** **Описание предметной области** 4](#_Toc75368425)

[**1.2.** **Описание существующих разработок** 6](#_Toc75368426)

[**1.2.1.** **Notepad++.** 7](#_Toc75368427)

[**1.2.2.** **Блокнот** 8](#_Toc75368428)

[**1.2.3.** **Microsoft Word** 9](#_Toc75368429)

[**2.** **Проектная часть** 11](#_Toc75368430)

[**2.1.** **Диаграмма прецедентов** 11](#_Toc75368431)

[**2.2.** **Выбор инструментов** 12](#_Toc75368432)

[**2.3.** **Диаграмма классов** 14](#_Toc75368433)

[**2.4.** **Описание главного модуля** 15](#_Toc75368434)

[**2.5.** **Описание спецификаций к модулям** 19](#_Toc75368435)

[**2.6.** **Описание модулей** 20](#_Toc75368436)

[**2.7.** **Описание тестовых наборов модулей** 21](#_Toc75368437)

[**2.8.** **Описание применения средств отладки** 23](#_Toc75368438)

[**3.** **Эксплуатационная часть** 24](#_Toc75368439)

[**3.1.** **Руководство оператора** 24](#_Toc75368440)

[**Заключение** 27](#_Toc75368441)

[**Список литературы и интернет-источников** 28](#_Toc75368442)

# **Введение**

Целью данного курсового проекта является написание веб-программы «Текстовый редактор» для упрощения работы с текстовой информацией. Эта тема является актуальной на данный момент, потому что в наше время слишком много информации проходит через головы людей. Данный курсовой проект позволит облегчить запись текстовой информации в веб-браузере. Так же в данном проекте будет простой для понимания интерфейс.

В первой части будет рассмотрена предметная область данной темы, а также несколько продуктов по данной теме.

Во второй части будут рассмотрены инструменты и модули, которые были разработаны, структура программной части и листинги ключевых частей программных модулей.

В третьей части будет рассмотрено руководство для пользователей.

В заключительной части будет приведен To-do лист с планами по доработки программы, а также сделаны общие выводы о получившемся проекте.

# **Теоретическая часть**

## **Описание предметной области**

**Текстовый редактор** — самостоятельная компьютерная программа или компонент программного комплекса (например, редактор исходного кода интегрированной среды разработки или окно ввода в браузере), предназначенная для создания и изменения текстовых данных в общем и текстовых файлов, в частности.

Текстовые редакторы предназначены для работы с текстовыми файлами в интерактивном режиме. Они позволяют просматривать содержимое текстовых файлов и производить над ними различные действия: вставку, удаление и копирование текста, контекстный поиск и замену, сортировку строк, просмотр кодов символов и конвертацию кодировок, печать и т. п.

Часто интерактивные текстовые редакторы содержат дополнительную функциональность, призванную автоматизировать действия по редактированию (от записываемых последовательностей нажатий клавиш до полноценных встроенных языков программирования), или отображают текстовые данные специальным образом (например, с подсветкой синтаксиса).

Текстовые процессоры ориентированы на оформление и форматирование текстов и внедрение в них сторонних объектов (шрифтов, таблиц, формул, графиков т. п.) и характеризуются наличием [WYSIWYG-режимов](https://ru.wikipedia.org/wiki/WYSIWYG). Поскольку в текстовом формате не предусмотрено хранение информации об оформлении текста, текстовые процессоры работают либо с файлами, в которых тексты представлены в обрамлении какого-либо [языка разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) вроде [HTML](https://ru.wikipedia.org/wiki/HTML), либо с файлами в собственных «[двоичных](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB)» форматах.

Многие текстовые редакторы являются редакторами исходного кода, то есть они ориентированы на работу с текстами программ.

Существует три вида тестовых редакторов:

1. Построчный (строковый) текстовый редактор.

Работает с текстом как последовательностью пронумерованных строк, выполняя операции над текстом в указанных строках.

Примером такого редактора может быть edlin, входивший в состав [MS-DOS](https://ru.wikipedia.org/wiki/MS-DOS).

1. Контекстный (строковый) редактор.

примером которого может быть ECCE ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Edinburgh Compatible Context Editor*)[[3]](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80#cite_note-3), выполняет операции над текстом в текущей позиции.

Выполняет операции над текстом в текущей позиции.

1. Экранный текстовый редактор.

Позволяет пользователю перемещать курсоров в тексте с помощью клавиш или других устройств ввода.

## **Описание существующих разработок**

В этом разделе рассмотрены некоторые уже имеющиеся программы для записи и форматирования текста.

Интерфейс таких программ для чаще всего простой и понятный для обычных пользователей.

Цели использования тестового редактора:

1. Создания и изменения текстовых файлов.
2. Печать документов.
3. Поиск отрывков текста.
4. редактирование строк текста;
5. возможность использования различных шрифтов символов;
6. копирование и перенос части текста с одного места на другое или из одного документа в другой;
7. контекстный поиск и замена частей текста;
8. задание произвольных межстрочных промежутков;
9. автоматический перенос слов на новую строку;
10. автоматическая нумерацию страниц;
11. обработка и нумерация сносок;
12. выравнивание краев абзаца;
13. создание таблиц и построение диаграмм;
14. проверка правописания слов и подбор синонимов;
15. построение оглавлений и предметных указателей;
16. распечатка подготовленного текста на принтере в нужном числе экземпляров и т.п.

Основные требования:

1. Запись и редактирование текста.
2. Сохранение текста после выхода из текстового редактора.

### **Notepad++.**

Cвободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса, разметки, а также языков описания аппаратуры VHDL и Verilog.

Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры. Поддерживает открытие более 100 форматов

Базовые возможности:

1. Подсветка синтаксиса.
2. Сворачивание кода.
3. Автодополнение и автоматическое закрытие скобок и тэгов (если активировано).
4. Закладки.
5. Регулярные выражения для поиска и замены.
6. Запись и воспроизведение макросов.
7. Сравнение файлов.
8. Менеджер проектов.
9. Карта документа.
10. Переопределение любых горячих клавиш.
11. Резервное копирование сохраняемых файлов (включается в настройках).
12. Трансформация текста при помощи подключённого плагина TextFX.
13. Поддержка и конвертирование кодировок ANSI, UTF-8 и UCS-2.
14. Блоковое выделение текста, одновременное выделение нескольких разных мест (с Ctrl).
15. Многострочное редактирование (с использованием Alt).

Возможности при установке плагинов:

1. Шаблоны текста (сниппеты), вводимые с помощью сокращений (плагин SnippetPlus).
2. FTP-менеджер (плагин NppFTP).
3. Hex-редактор.
4. Автосохранение (при потере фокуса; через настраиваемый промежуток времени).
5. Проверка орфографии (с использованием GNU Aspell).
6. Симметричное и асимметричное шифрование текста (при установке плагина NppDarkCrypt).
7. Поддержка Zen Coding.
8. Поддержка автоматизации с помощью скриптов: Python, JScript, Lua, и других.
9. Поддержка сохранения в OneDrive и Dropbox.

### **Блокнот**

**Блокнот** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Notepad*) — простой [текстовый редактор](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80), являющийся частью [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), начиная с вышедшей в [1985 году](https://ru.wikipedia.org/wiki/1985_%D0%B3%D0%BE%D0%B4) [Windows 1.0](https://ru.wikipedia.org/wiki/Windows_1.0).

Блокнот использует [оконный класс](https://ru.wikipedia.org/wiki/Windows_API) *EDIT*. Вплоть до вышедшей в [2000 году](https://ru.wikipedia.org/wiki/2000_%D0%B3%D0%BE%D0%B4) [Windows ME](https://ru.wikipedia.org/wiki/Windows_ME) поддерживались только самые базовые функции, многие функции были доступны только из меню, а максимальный размер [файла](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB) составлял 64 [килобайта](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BB%D0%BE%D0%B1%D0%B0%D0%B9%D1%82) (предел класса EDIT). В настоящее время редактор поддерживает контекстную замену, [горячие клавиши](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D1%80%D1%8F%D1%87%D0%B0%D1%8F_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88%D0%B0) (например, Ctrl+S для сохранения файла), снят предел в 64 Кбайт и добавлена поддержка [Юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4).

### **Microsoft Word**

**Microsoft Word** (часто — **MS Word**, **WinWord** или просто **Word**) — [текстовый процессор](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), предназначенный для создания, просмотра, редактирования и форматирования текстов статей, деловых бумаг, а также иных [документов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), с локальным применением простейших форм [таблично](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0)-[матричных](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) алгоритмов. Выпускается [корпорацией Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в составе [пакета](https://ru.wikipedia.org/wiki/%D0%9E%D1%84%D0%B8%D1%81%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82) [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office). Первая версия была написана [Ричардом Броди](https://ru.wikipedia.org/w/index.php?title=%D0%91%D1%80%D0%BE%D0%B4%D0%B8,_%D0%A0%D0%B8%D1%87%D0%B0%D1%80%D0%B4&action=edit&redlink=1) ([Richard Brodie](https://en.wikipedia.org/wiki/Richard_Brodie)) для [IBM PC](https://ru.wikipedia.org/wiki/IBM_PC), использующих [DOS](https://ru.wikipedia.org/wiki/DOS), в 1983 году.

Позднее выпускались версии для [Apple Macintosh](https://ru.wikipedia.org/wiki/Macintosh) (1984), [SCO UNIX](https://ru.wikipedia.org/wiki/SCO_UNIX) и [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows) (1989). Текущей версией является Microsoft Office Word 2019 для [Windows](https://ru.wikipedia.org/wiki/Windows) и [macOS](https://ru.wikipedia.org/wiki/MacOS), а также веб-версия Word Online ([Office Online](https://ru.wikipedia.org/wiki/Office_Online)), не требующая установки программы на компьютер.

Microsoft Word является наиболее популярным из используемых в данный момент текстовых процессоров, что сделало его бинарный [формат документа](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0) стандартом [де-факто](https://ru.wikipedia.org/wiki/%D0%94%D0%B5-%D1%84%D0%B0%D0%BA%D1%82%D0%BE), и многие конкурирующие программы имеют поддержку совместимости с данным форматом. Расширение «[.doc](https://ru.wikipedia.org/wiki/.doc)» на платформе [IBM PC](https://ru.wikipedia.org/wiki/IBM_PC) стало синонимом двоичного формата Word 97—2000. Фильтры экспорта и импорта в данный формат присутствуют в большинстве [текстовых процессоров](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80). Формат документа разных версий Word меняется, различия бывают довольно тонкими. Форматирование, нормально выглядящее в последней версии, может не отображаться в старых версиях программы, однако есть ограниченная возможность сохранения документа с потерей части форматирования для открытия в старых версиях продукта.

Последняя версия MS Word 2007 «использует по умолчанию» формат, основанный на [XML](https://ru.wikipedia.org/wiki/XML), — [Microsoft Office Open XML](https://ru.wikipedia.org/wiki/Microsoft_Office_Open_XML). Спецификации форматов файлов Word 97-2007 были опубликованы Microsoft в 2008 году[[4]](https://ru.wikipedia.org/wiki/Microsoft_Word#cite_note-4)[[5]](https://ru.wikipedia.org/wiki/Microsoft_Word#cite_note-5). Ранее большая часть информации, нужной для работы с данным форматом, добывалась посредством [обратного инжиниринга](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D0%B6%D0%B8%D0%BD%D0%B8%D1%80%D0%B8%D0%BD%D0%B3), поскольку основная её часть отсутствовала в открытом доступе или была доступна лишь ограниченному числу партнёров и контролирующих организаций.

Как и прочие приложения из [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office), Word может расширять свои возможности посредством использования встроенного макроязыка (сначала использовался [WordBasic](https://ru.wikipedia.org/wiki/WordBasic), с версии Word 97 применяется [VBA](https://ru.wikipedia.org/wiki/Visual_Basic_for_Applications) — Visual Basic для приложений). Однако это предоставляет широкие возможности для написания встраиваемых в документы [вирусов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B8%D1%80%D1%83%D1%81) (так называемые «макровирусы»).

Наиболее ярким примером была эпидемия червя [Melissa](https://en.wikipedia.org/wiki/Melissa_(computer_worm)). В связи с этим многие считают разумной рекомендацию всегда выставлять наивысший уровень настроек безопасности при использовании Word (Меню: «Tools>Macro>Security», «Сервис>Макрос>Безопасность…» в локализованных русских версиях). Также нелишним будет использовать [антивирусное программное обеспечение](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%B8%D0%B2%D0%B8%D1%80%D1%83%D1%81%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Первым вирусом, заражавшим документы Microsoft Word, был DMV, созданный в декабре 1994 года Дж. Мак-Намарой для демонстрации возможности создания макровирусов. Первым же вирусом, попавшим в «дикую природу» и вызвавшим первую в мире эпидемию макровирусов (это произошло в июле-августе 1995 года), был Concept.

На документ Microsoft Word могут быть установлены 3 типа паролей:

* Пароль для открытия документа
* Пароль для изменения документа
* Пароль на внесение примечаний и исправлений (для версий 2003 и ниже). В новых версиях MS Word 2007\2010 этот тип пароля называется *«Пароль для Ограничения форматирования и редактирования»*

Вне зависимости от версий MS Word, в которой был создан документ, *«пароль разрешения записи»* и *«пароль на внесение примечаний и исправлений»* можно удалить из документа мгновенно. Эти типы паролей служат не столь для защиты документа, как для обеспечения коллективной работы над документом. А основную защиту документа от несанкционированного доступа обеспечивает *«пароль для открытия документа»*.

В документах ранних версиях MS Word (до MS Office 2003 включительно) пароль для открытия файла может быть без проблем взломан.

В последних версия MS Office 2007 \ 2010 компания Microsoft использует стойкий алгоритм шифрования AES с 128-битным ключом. Формирование ключа происходит путём 50000 \ 100000 кратного применения SHA-1 хеш-функции, что делает перебор паролей крайне медленным, и при использовании стойкого пароля шансы его подобрать близки к нулю.

# **Проектная часть**

## **Диаграмма прецедентов**

В этом разделе представлена диаграмма прецедентов. На диаграмме показаны все возможные функциональные и поведенческие отношения.

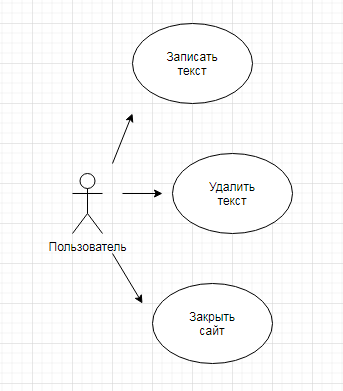


Рисунок 1. Диаграмма прецедентов работы пользователя

## **Выбор инструментов**

При выборе инструментов мной был выбрал язык JavaScript, так как, изначально хотел сделать веб-приложение, так же в проекте присутствует Html и Css. **Проектирование сценария**

В данном разделе приведен сценарий использования программы пользователем.

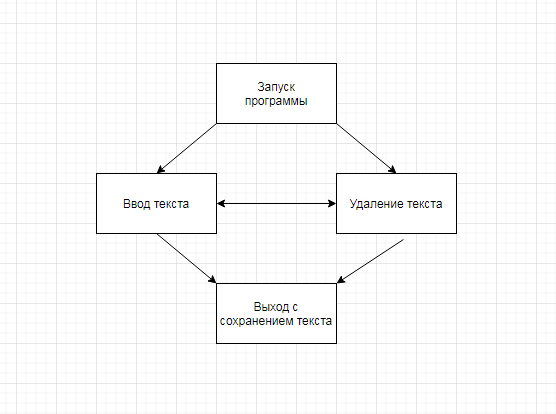


Рисунок 2. Сценарий использования

Пользователь после запуска программы может выполнить 3 действия: начать вводить текст, выйти из программы с сохранением текста или удалить имеющийся текст.

При выборе выхода программа заканчивает свою работу, она сохраняет весь введенный текст, после повторного запуска всё вами написанное останется.

## **Диаграмма классов**

В данном разделе представлен класс, использующийся в проекте.

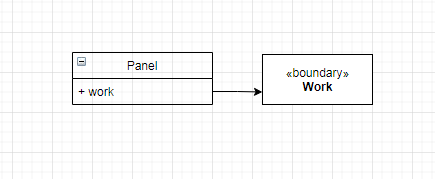


Рисунок 3. Диаграмма классов для проекта

Класс «Panel» является наследником метода Work(), который позволяет сохранять введенный текст.

## **Описание главного модуля**

В главный модуль входит класс Panel который отвечает за функционал сохранения программы.

В классе Panel реализованы следующие методы:

**Листинг 1. Реализация сохранения текста.**

<script>

let Panel = class {

work() {

// если в нашем хранилище уже что-то есть…

if (localStorage.getItem('text\_in\_editor') !== null) {

// …то отображаем его содержимое в нашем редакторе

document.getElementById('editor').innerHTML = localStorage.getItem('text\_in\_editor');

}

// отслеживаем каждое нажатие клавиши и при каждом нажатии выполняем команду

document.addEventListener('keydown', function (e) {

// записываем содержимое нашего редактора в хранилище

localStorage.setItem('text\_in\_editor', document.getElementById('editor').innerHTML);

});

// закончился скрипт

};

};

new Panel().work();

</script>

Данный метод реализует сохранение текста, чтобы после закрытия веб-страницы, все ваши данные остались.

**Листинг 2. Работа со стилями через CSS.**

<style type="text/css">

/\*задаём общие параметры для всей страницы: шрифт и отступы\*/

body {

text-align: left;

margin: 10;

font-family: Courier New, Courier;

font-size: 20px;

background-color: lightgray;

}

/\* оформляем окно редактора \*/

.editorSheet {

width: 80vw;

min-height: 100vw;

margin-left: 10vw;

border: solid;

border-width: 0px;

text-align: left;

background-color: white;

-webkit-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

-moz-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

padding: 15px;

}

h1 {

font-size: 40px;

font-family: Tahoma;

font-weight: 900;

}

/\*закончили со стилями\*/

</style>

Использование CSS в разработке веб-приложения очень нужна вещь, через CSS вы меняете стиль всего сайта.

**Листинг 3. HTML.**

<!DOCTYPE html>

<html> <!-- служебная часть -->

<head>

<!-- заголовок страницы -->

<title>Текстовый редактор</title>

<!-- настраиваем служебную информацию для браузеров -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<!-- задаём CSS-стили прямо здесь же, чтобы всё было в одном файле -->

<style type="text/css">

/\*задаём общие параметры для всей страницы: шрифт и отступы\*/

body {

text-align: left;

margin: 10;

font-family: Courier New, Courier;

font-size: 20px;

background-color: lightgray;

}

/\* оформляем окно редактора \*/

.editorSheet {

width: 80vw;

min-height: 100vw;

margin-left: 10vw;

border: solid;

border-width: 0px;

text-align: left;

background-color: white;

-webkit-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

-moz-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

padding: 15px;

}

h1 {

font-size: 40px;

font-family: Tahoma;

font-weight: 900;

}

/\*закончили со стилями\*/

</style>

<!-- закрываем служебную часть страницы -->

</head>

<!-- началось содержимое страницы -->

<body>

<!-- началась видимая часть -->

<!-- заголовок страницы -->

<h1>Текстовый редактор с автосохранением</h1>

<!-- большой блок для ввода текста: высота в половину, а ширина — во весь экран, назвывается "text\_area", обведено рамкой толщиной в 1 пиксель, выравнивание текста — по левому краю -->

<div id="editor" contenteditable="true" class="editorSheet">

</div>

<!-- закончилась видимая часть -->

<!-- пишем скрипт, который будет постоянно сохранять наш текст -->

<script>

let Panel = class {

work() {

// если в нашем хранилище уже что-то есть…

if (localStorage.getItem('text\_in\_editor') !== null) {

// …то отображаем его содержимое в нашем редакторе

document.getElementById('editor').innerHTML = localStorage.getItem('text\_in\_editor');

}

// отслеживаем каждое нажатие клавиши и при каждом нажатии выполняем команду

document.addEventListener('keydown', function (e) {

// записываем содержимое нашего редактора в хранилище

localStorage.setItem('text\_in\_editor', document.getElementById('editor').innerHTML);

});

// закончился скрипт

};

};

new Panel().work();

</script>

<!-- закончилось содержимое страницы -->

</body>

<!-- конец всего HTML-документа -->

</html>

HTML важный ресурс создания сайта, так как он и в начале кода и в конце, я вставил весь код.

## **Описание спецификаций к модулям**

В данном разделе описан публичны член модулей курсового проекта.

В главном модуле (index.html) содержится публичный метод класса Panel. Так как все эти методы были описаны в разделе 2.6, в данном разделе описание данных методов опущено. В данном модуле также есть один объект:

1. work – объект, который реализует сохранение текста.

В моем коде, есть только один модуль и один метод, так как весь код написан через один модуль.

## **Описание модулей**

Кроме главного модуля программа не содержит более модулей.

Полный код модуля приведен в приложении 1. В этом разделе приведена блок-схема метода, который содержаться в класс Panel из модуля index.html.

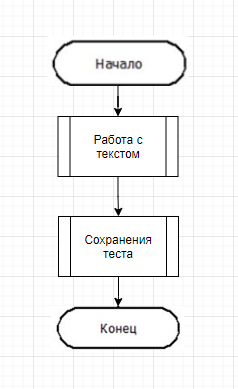
**

Рисунок 4. Блок-схема метода work

## **Описание тестовых наборов модулей**

В этом разделе будут продемонстрированы результаты тестирования «черного ящика».

Тест 1. Ввод текста на русском, английском и чешском языке.

Действия: Отрыть веб-приложение, ввести текст на трех разных языках.

Ожидаемый результат: Распознание символов кириллицы и латиницы.

Результат теста:

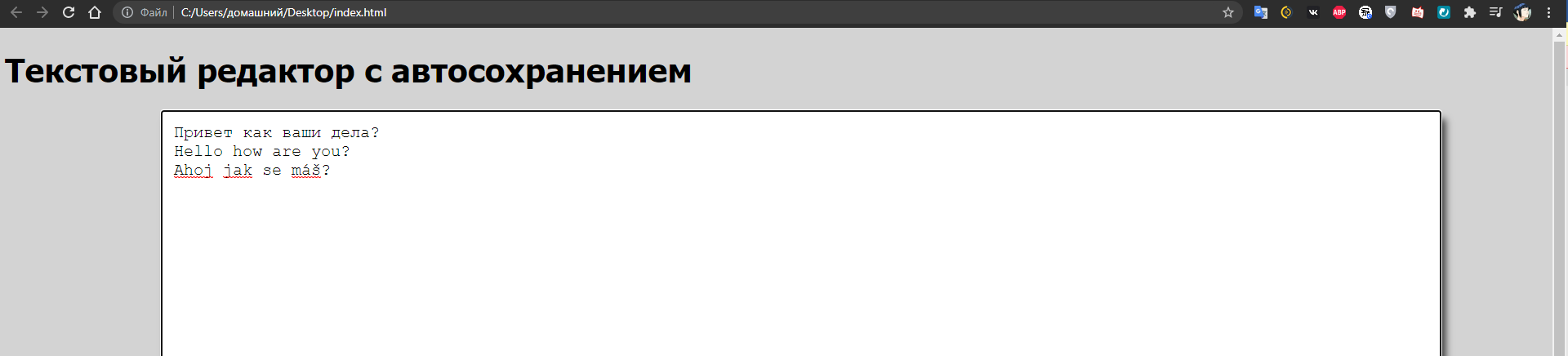


Рисунок 5. Тест 1

Тест 2. Символы юникода.

Действия: Выбор и последующий ввод различных символов юникода.

Ожидаемы результат: Распознание символов юникода.

Результат тесте:

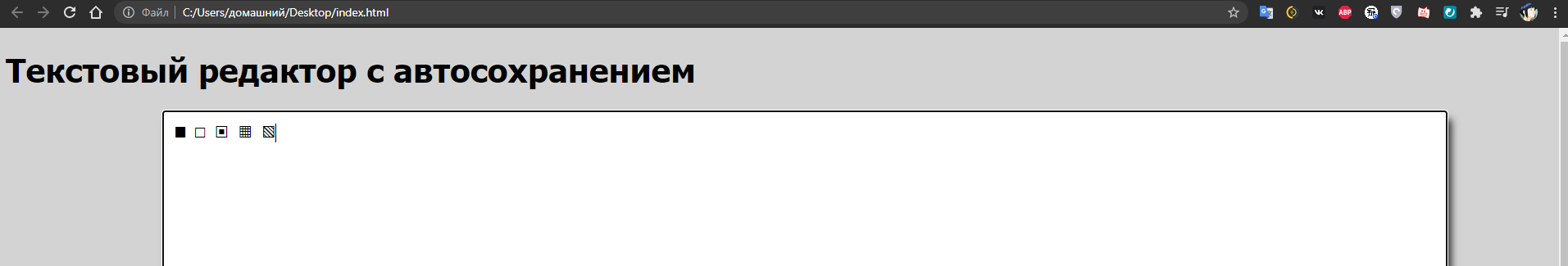


Рисунок 6. Тест 2

Тест 3. Сохранение текста с последующим выходом из веб-приложения.

Действия: Ввод текста, выход из приложения и заново открыть приложение.

Ожидаемый результат: Текст сохранится.

Результат теста:

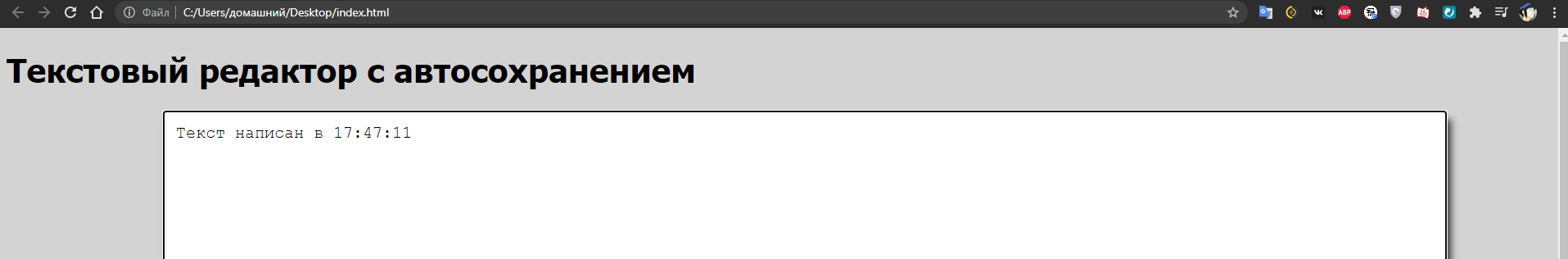


Рисунок 7. Текст до выхода

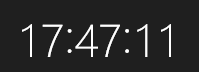


Рисунок 17. Время написания текста

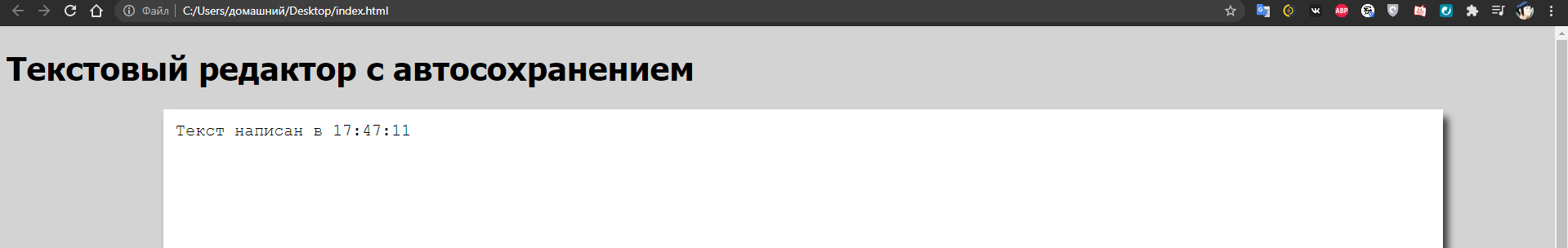


Рисунок 8. Текст после выхода



Рисунок 9. Время после повторного открытия веб-приложения

## **Описание применения средств отладки**

В этом разделе показано умение применять средства отладки.

В ходе написания курсового проекта при попытке запустить скрипт было получено данное сообщение:

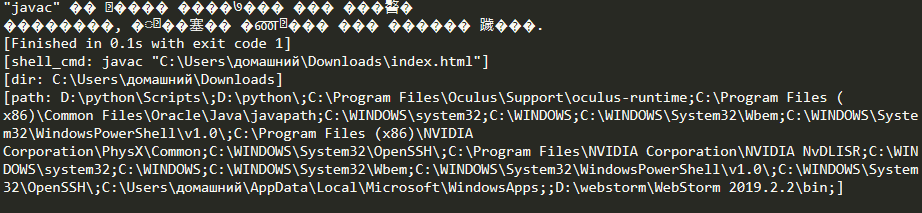


Рисунок 10. До применения средств отладки

После получения данного сообщения были просмотрены 74 и 75 строки модуля index.html и была обнаружена ошибка, которая впоследствии была устранена, а после попытки запуска скрипта, запускается веб-приложение:

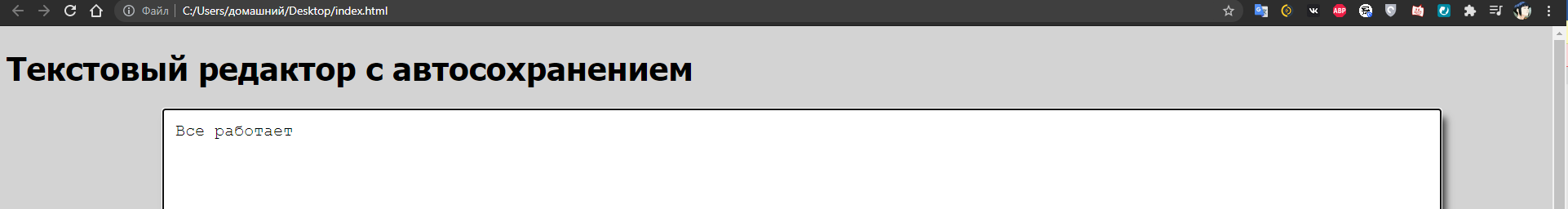


Рисунок 11. После применения средств отладки

## 

# **Эксплуатационная часть**

## **Руководство оператора**

**АННОТАЦИЯ**

В данном программном документе приведено руководство оператора по применению и эксплуатации программы «Текстовый редактор», предназначенной для облегчения пользователям набора текста.

В данном программном документе, в разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В данном программном документе, в разделе «Выполнение программы» указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 [[1]](#footnote-1)1), ГОСТ 19.103-77 [[2]](#footnote-2)2), ГОСТ 19.104-78\* [[3]](#footnote-3)3), ГОСТ 19.105-78\* [[4]](#footnote-4)4), ГОСТ 19.106-78\* [[5]](#footnote-5)5), ГОСТ 19.505-79\* [[6]](#footnote-6)6), ГОСТ 19.604-78\* [[7]](#footnote-7)7)).

1. **Назначение программы**
   1. **Функциональное назначение программы**

Специальное программное обеспечение «Текстовый редактор» используется для удобной записи текста с последующим редактированием его.

* 1. **Эксплуатационное назначение программы**

Специальное программное обеспечение «Текстовый редактор» может эксплуатироваться любым пользователем, который хочет записать текст.

* 1. **Состав функций**
     1. **Функция сохранения текста.**

Эта функция позволяет сохранять текст и не беспокоится о потере введенной информации

1. **Условия выполнения программы**
   1. **Минимальный состав аппаратных средств**

ОС: Windows 10

Процессор: Как минимум 1 ГГц или SoC.

ОЗУ: 1 ГБ (для 32-разрядных систем) или 2 ГБ (для 64-разрядных систем).

Место на жестком диске: 16 ГБ (для 32-разрядных систем) или 20 ГБ (для 64-разрядных систем).

Видеоадаптер: DirectX версии не ниже 9 с драйвером WDDM 1.0.

Дисплей: 800 x 600.

* 1. **Минимальный состав программных средств**

Дополнительные программные средства не требуются.

* 1. **Требование к персоналу (пользователю)**

Конечный пользователь должен уметь пользоваться клавиатурой и уметь запускать приложения.

1. **Выполнение программы**
   1. **Загрузка и запуск программы**

Перед запуском программы вам ничего не нужно создавать, просто запустить из любого удобного места файл и начать ввод текста на любом удобном вам языке, спокойно используя любые символы, можно легко копировать текст с любых источников и не беспокоится о том, что ввод текста получится без распознания символов.

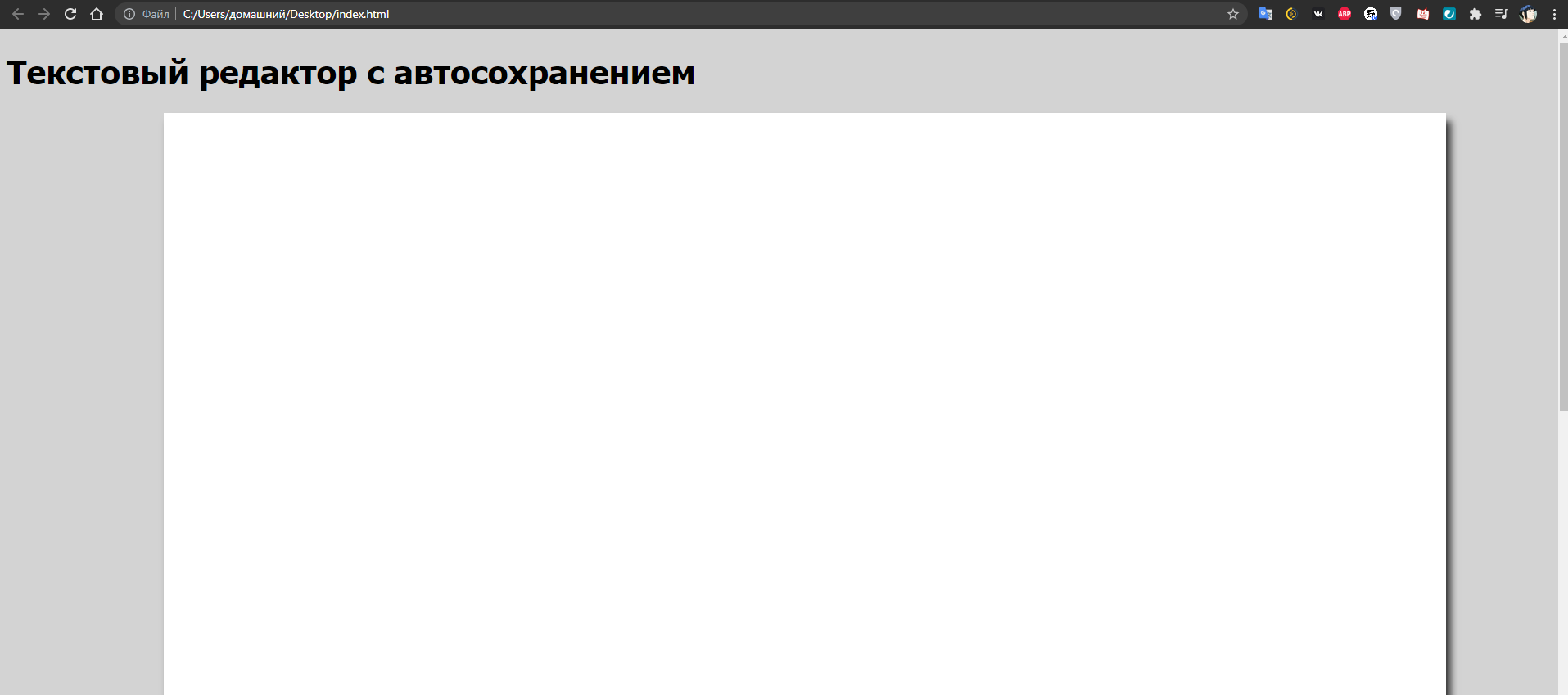
**

Рисунок 12. Веб-приложение

* 1. **Завершение работы программы**

Чтобы завершить работу программы нажмите в браузере кнопку закрытия окна, текст сам сохранится, можете не беспокоится.

# **Заключение**

В результате выполнения курсового проекта была написана программа «Текстовый редактор» для упрощения ввода текста.

В ходе работы были проанализированы предметная область, существующие разработки, посвященные данному направлению, получены практические навыки по созданию сайтов HTML используя CSS и JavaScript.

Также планируется продолжать работу над данным проектом с целью расширения возможностей и удобства приложения для пользователей. Планы по доработкам представлены ниже.

To-do лист:

1. Создание главного меню с реализацией функций из Word.
2. Интеграция баз данных для удобства записей лекций и прочего.
3. Доработка интерфейса с целью упрощения работы с программой.
4. Создать 2 показателя: количество страниц и число слов.
5. Улучшение дизайна интерфейса.

# **Список литературы и интернет-источников**

1. Онлайн-учебник по языку JavaScript

https://learn.javascript.ru

1. Онлайн-учебник по HTML:

<http://htmlbook.ru/samhtml>

1. Онлайн-учебник по CSS:

<http://htmlbook.ru/samcss>

1. Хорстманн, К. С. Современный JavaScript для нетерпеливых : практическое пособие / Кэй С. Хорстманн ; пер. с англ. А. А. Слинкина. - Москва : ДМК Пресс, 2021. - 288 с. - ISBN 978-5-97060-177-8. - Текст : электронный.

<http://znanium.com/catalog/document?id=369593>

# **Приложение 1. Код главного модуля index.html.**

**Листинг 4. Проект**

<!DOCTYPE html>

<html> <!-- служебная часть -->

<head>

<!-- заголовок страницы -->

<title>Текстовый редактор</title>

<!-- настраиваем служебную информацию для браузеров -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<!-- задаём CSS-стили прямо здесь же, чтобы всё было в одном файле -->

<style type="text/css">

/\*задаём общие параметры для всей страницы: шрифт и отступы\*/

body {

text-align: left;

margin: 10;

font-family: Courier New, Courier;

font-size: 20px;

background-color: lightgray;

}

/\* оформляем окно редактора \*/

.editorSheet {

width: 80vw;

min-height: 100vw;

margin-left: 10vw;

border: solid;

border-width: 0px;

text-align: left;

background-color: white;

-webkit-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

-moz-box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

box-shadow: 6px 10px 9px 0px rgba(0, 0, 0, 0.75);

padding: 15px;

}

h1 {

font-size: 40px;

font-family: Tahoma;

font-weight: 900;

}

/\*закончили со стилями\*/

</style>

<!-- закрываем служебную часть страницы -->

</head>

<!-- началось содержимое страницы -->

<body>

<!-- началась видимая часть -->

<!-- заголовок страницы -->

<h1>Текстовый редактор с автосохранением</h1>

<!-- большой блок для ввода текста: высота в половину, а ширина — во весь экран, назвывается "text\_area", обведено рамкой толщиной в 1 пиксель, выравнивание текста — по левому краю -->

<div id="editor" contenteditable="true" class="editorSheet">

</div>

<!-- закончилась видимая часть -->

<!-- пишем скрипт, который будет постоянно сохранять наш текст -->

<script>

let Panel = class {

work() {

// если в нашем хранилище уже что-то есть…

if (localStorage.getItem('text\_in\_editor') !== null) {

// …то отображаем его содержимое в нашем редакторе

document.getElementById('editor').innerHTML = localStorage.getItem('text\_in\_editor');

}

// отслеживаем каждое нажатие клавиши и при каждом нажатии выполняем команду

document.addEventListener('keydown', function (e) {

// записываем содержимое нашего редактора в хранилище

localStorage.setItem('text\_in\_editor', document.getElementById('editor').innerHTML);

});

// закончился скрипт

};

};

new Panel().work();

</script>

<!-- закончилось содержимое страницы -->

</body>

<!-- конец всего HTML-документа -->

</html>

1. 1) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов [↑](#footnote-ref-1)
2. 2) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов [↑](#footnote-ref-2)
3. 3) ГОСТ 19.104-78\* ЕСПД. Основные надписи [↑](#footnote-ref-3)
4. 4) ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам [↑](#footnote-ref-4)
5. 5) ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом [↑](#footnote-ref-5)
6. 6) ГОСТ 19.505-79\* ЕСПД. Руководство оператора. Требования к содержанию и оформлению [↑](#footnote-ref-6)
7. 7) ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом [↑](#footnote-ref-7)