

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф.
Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

ОТЧЕТ о курсовой работе

Тема Объектно-ориентированная разработка программ с графическим
пользовательским интерфейсом «сверху-вниз»

Вариант: Интернет-браузер

Обучающегося группы О722Б Головкин Р.В.
группа фамилия и инициалы

Направление подготовки / Информационные системы и технологии
специальность 09.03.02 полное наименование направления подготовки / специальности
индекс

Направленность Технологии разработки информационных систем
образовательной программы профиль / специализация / магистерская программа

Дисциплина (модуль) Информационные технологии и программирование

Руководитель: _____
подпись

_____ Вальштейн К.В.
ученая степень, ученое звание фамилия ИО

Оценка: _____
« _____ » 20 ____ г.

Обучающийся: _____
подпись

_____ Головкин Р.В.
фамилия ИО
« _____ » 20 ____ г.

Санкт-Петербург
2024 г.

РЕФЕРАТ

Отчет 21 с., 17 рис., 5 источн., 1 прил.

ООП, ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС,
НАСЛЕДОВАНИЕ, C#, WPF

Объект разработки – интерактивное приложение с графическим пользовательским интерфейсом.

Целью курсовой работы является разработка интерактивного приложения с графическим пользовательским интерфейсом с использованием фреймворка .Net и технологии WPF [1].

В процессе работы проводились исследования предметной области, были определены основные требования к разрабатываемому приложению, сформирована иерархия классов, выбраны средства разработки, спроектирован пользовательский интерфейс и структура приложения.

В результате разработки было создано интерактивное приложение с графическим пользовательским интерфейсом «Интернет-браузер».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи	5
2 Описание иерархии классов	6
2.1 Класс Tab.....	6
2.2 Класс TabsController.....	8
2.3 Класс HistoryController	9
2.4 Класс FavouritesController.....	11
2.5 Структура FavouritesItem.....	12
2.6 Структура HistoryItem.....	12
2.7 Демонстрация иерархии классов	13
3 Используемые мультимедийные ресурсы и сторонние библиотеки.....	14
4 Демонстрация работы	15
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20
ПРИЛОЖЕНИЕ А Текст программы	21

ВВЕДЕНИЕ

Самым распространенными на данный момент являются веб приложения. Они просты в разработке, доступны без установки на любой платформе, устройстве с любой аппаратной составляющей. Веб приложения доминируют над нативными решениями [2], которые всегда являются более громоздкими и монументальными, требующими большого количество физических ресурсов и действительно необходимыми лишь для узких задач по взаимодействию с компьютером. Для основных повседневных задач обычного пользователя гораздо более удобными являются веб приложения. Для доступа к любому веб приложению необходимо подключение к интернету и веб браузер, от которого, в большинстве своем, зависит быстродействие веб приложений и удобство взаимодействия с ними. В связи с большой актуальностью веб приложений была выбрана тема курсовой работы, задача которой – реализовать веб браузер.

Целью курсовой работы является разработка интерактивного приложения с графическим пользовательским интерфейсом с использованием фреймворка .Net и технологии WPF.

Для достижения поставленной цели необходимо решить следующие задачи:

- описать основные требования к разрабатываемому приложению;
- составить иерархию классов;
- разработать приложение;
- продемонстрировать работоспособность приложения.

1 Постановка задачи

Должен быть реализован Интернет-браузер с поддержкой ввода адреса, нескольких вкладок, закладок, истории просмотров.

На каждой вкладке должен отображаться текущий адрес с возможностью его изменения при открытии вкладки. Также должна быть реализована возможность закрыть вкладку, вернуться на страницу назад и вперед, перезагрузить страницу, добавить адрес в избранное с сохранением читаемого домена. Переход по адресу из адресной строки должен осуществляться нажатием клавиши ввод.

У браузера должна быть главная страница с таблицей избранного и историей просмотра. Должна быть реализована возможность удаления записи из списка избранного и истории, а также переход по хранимой ссылке. Должна быть предусмотрена полная очистка истории. Данные об избранном и истории необходимо сохранять при выходе из приложения.

В результате были сформированы требования, по которым будет разработано приложение.

2 Описание иерархии классов

В процессе разработки приложения были созданы следующие классы:

- Tab – класс, описывающий элементы отображения и логику работы вкладки браузера;
- TabController – класс, описывающий управление вкладками и их отображение;
- HistoryController – класс, описывающий управление историей браузера;
- FavouritesController – класс, описывающий управление закладками пользователя.

Также, были созданы следующие структуры:

- HistoryItem – структура, описывающая элемент записи в истории браузера;
- FavouritesItem – структура, описывающая элемент записи в списке закладок пользователя.

Далее будет приведено описание членов каждого из этих классов и интерфейса.

2.1 Класс Tab

Данный класс содержит следующие публичные свойства:

- Label LabelCurrentUrl – данное свойство используется для хранения и отображения элемента Label в заголовке вкладки;
- TextBox TextBoxQuery – данное свойство используется для хранения и отображения адресной строки внутри вкладки;
- Button ButtonGoBack – данное свойство используется для хранения и отображения кнопки перехода на предыдущую страницу;
- Button ButtonGoForward – данное свойство используется для хранения и отображения кнопки перехода на следующую страницу;
- Button ButtonReload – данное свойство используется для хранения и отображения кнопки перезагрузки текущей страницы;

– `Button ButtonAddToFavourites` – данное свойство используется для хранения и отображения кнопки добавления текущего адреса из адресной строки в избранное;

– `WebView2 WV` – данное свойство используется для хранения и отображения компонента отображения веб страниц.

Также данный класс описывает следующие приватные методы.

`void InitUI()` – данный метод используется для инициализации всех элементов интерфейса, описанных в свойствах класса.

`void InitWV(string link)` – данный асинхронный метод используется для инициализации компонента отображения веб страниц `WebView2`. В качестве параметра принимает ссылку на стартовую страницу.

`void ButtonGoBack_Click(object sender, RoutedEventArgs e)` – данный метод описывает событие нажатия на кнопку перехода на предыдущую страницу. Параметры метода сгенерированы фреймворком.

`void ButtonGoForward_Click(object sender, RoutedEventArgs e)` – данный метод описывает событие нажатия на кнопку перехода на следующую страницу. Параметры метода сгенерированы фреймворком.

`void ButtonReload_Click(object sender, RoutedEventArgs e)` – данный метод описывает событие нажатия на кнопку перезагрузки текущей страницы. Параметры метода сгенерированы фреймворком.

`void ButtonAddToFavourites_Click(object sender, RoutedEventArgs e)` – данный метод описывает нажатие на кнопку добавления текущего адреса из адресной строки в избранное. Параметры метода сгенерированы фреймворком.

`void WV_SourceChanged(object? sender, CoreWebView2SourceChangedEventArgs e)` – данный метод реализует логику, исполняемую при изменении текущего адреса. Параметры метода сгенерированы фреймворком.

`void WV_NavigationCompleted(object sender, WebView2.Core.CoreWebView2NavigationCompletedEventArgs e)` – данный

метод реализует логику, исполняемую при завершении переадресации компонента отображения веб страниц. Параметры метода сгенерированы фреймворком.

`void TextBoxQuery_KeyDown(object sender, KeyEventArgs e)` – данный метод реализует логику, исполняемую при нажатии кнопки ввод. Параметры метода сгенерированы фреймворком.

На рисунке 1 представлена диаграмма класса Tab.

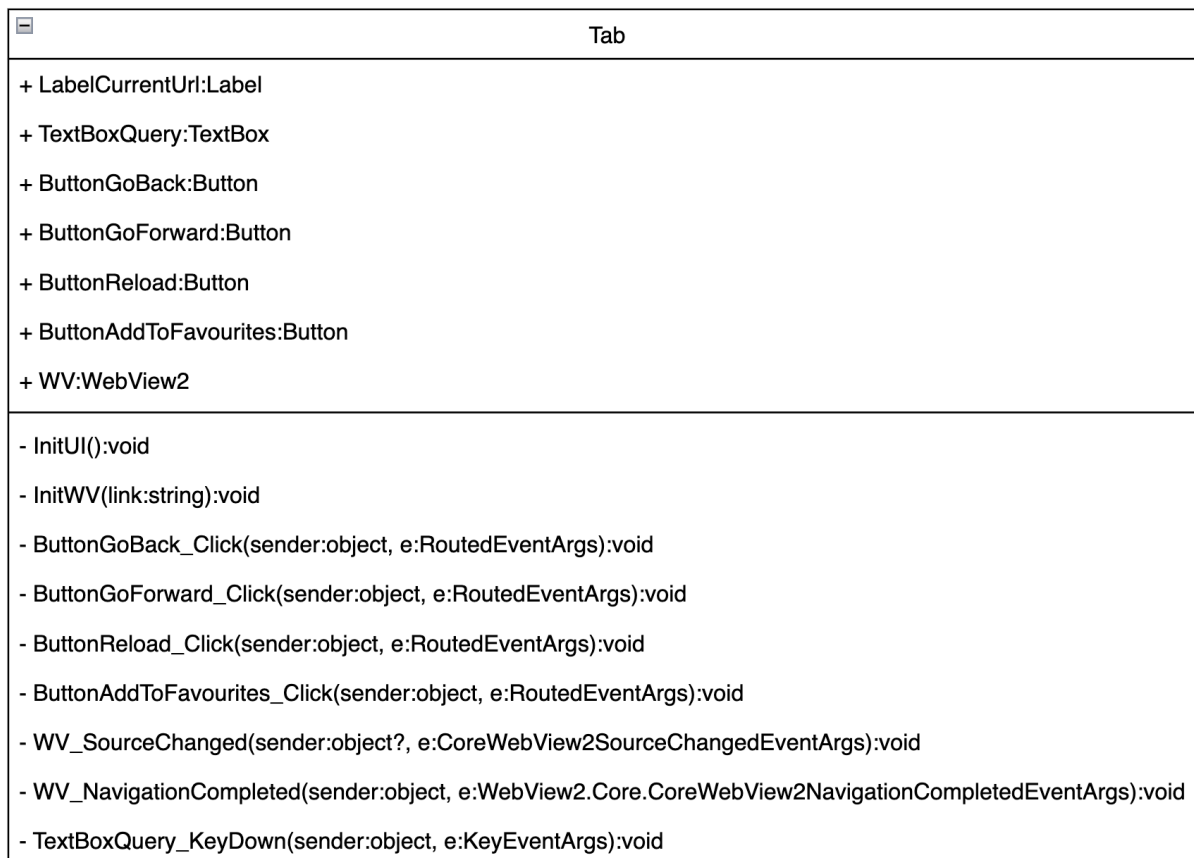


Рисунок 1 – Диаграмма класса Tab

2.2 Класс TabsController

Данный класс содержит следующие публичные свойства:

- `int TabsCount` – данное поле содержит количество открытых вкладок;
- `TabControl tbCtrl` – данное свойство содержит объект `TabControl`, использующийся для отображения вкладок.

Также данный класс содержит следующие публичные методы.

`void AddNewTab(string link)` – данный метод используется для добавления новой вкладки. В качестве параметра принимает ссылку на стартовую страницу. Значение параметра по умолчанию – `http://google.com`.

`void LabelClose_MouseDown(object sender, System.Windows.Input.MouseButtonEventArgs e)` – данный метод реализует закрытие вкладки при нажатии на соответствующую иконку. Параметры метода сгенерированы фреймворком.

На рисунке 2 представлена диаграмма класса `TabController`.

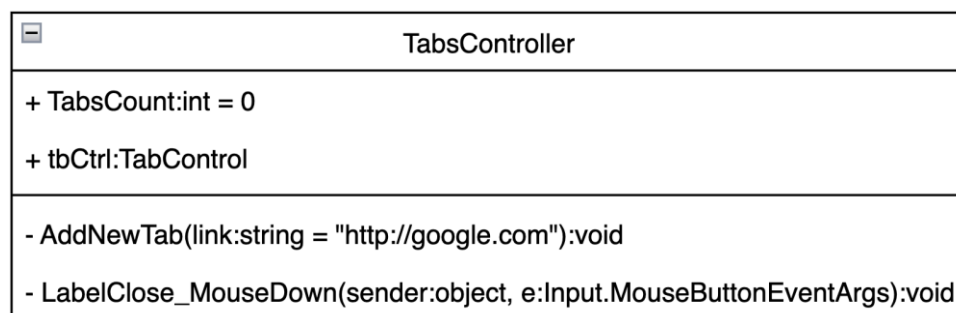


Рисунок 2 – Диаграмма класса `TabController`

2.3 Класс `HistoryController`

Данный класс содержит следующие публичные поля:

- `string HistoryCsvPath` – данное константное поле содержит путь до файла с данными об истории браузера в формате CSV [3];
- `ListView LV` – данное поле содержит элемент отображения списка истории браузера и используется для контролирования отображаемой информации;
- `List<HistoryItem> ListHistory` – данное поле содержит список объектов `HistoryItem` истории браузера.

Также данный класс содержит следующие приватные методы.

`void InitUI()` – данный метод используется для инициализации и отображения элементов истории браузера.

`void RemoveMi_Click(object sender, System.Windows.RoutedEventArgs e)` – данный метод содержит логику нажатия на кнопку удаления в контекстном меню списка истории браузера. Параметры метода сгенерированы фреймворком.

`void OpenMi_Click(object sender, System.Windows.RoutedEventArgs e)` – данный метод содержит логику нажатия на кнопку открытия в контекстном меню списка истории браузера. Параметры метода сгенерированы фреймворком.

`void AddToHistory(HistoryItem item, bool _new)` – данный метод используется для добавления нового элемента в список истории. Параметр `item` типа `HistoryItem` содержит добавляемую запись, параметр `_new` типа `bool` необходим для заполнения списка уже существующими значениями без добавления их в файл, значение по умолчанию – `true`.

`void LoadFromFile(string filePath)` – данный метод используется для загрузки списка истории браузера из файла. Параметр `filePath` типа `string` обозначает путь к файлу для загрузки данных.

`void SaveToFile(string filePath)` – данный метод используется для сохранения списка истории браузера в файл. Параметр `filePath` типа `string` обозначает путь к файлу для сохранения данных.

`void ClearHistory()` – данный метод используется для полной очистки истории браузера.

На рисунке 3 представлена диаграмма класса `HistoryController`.

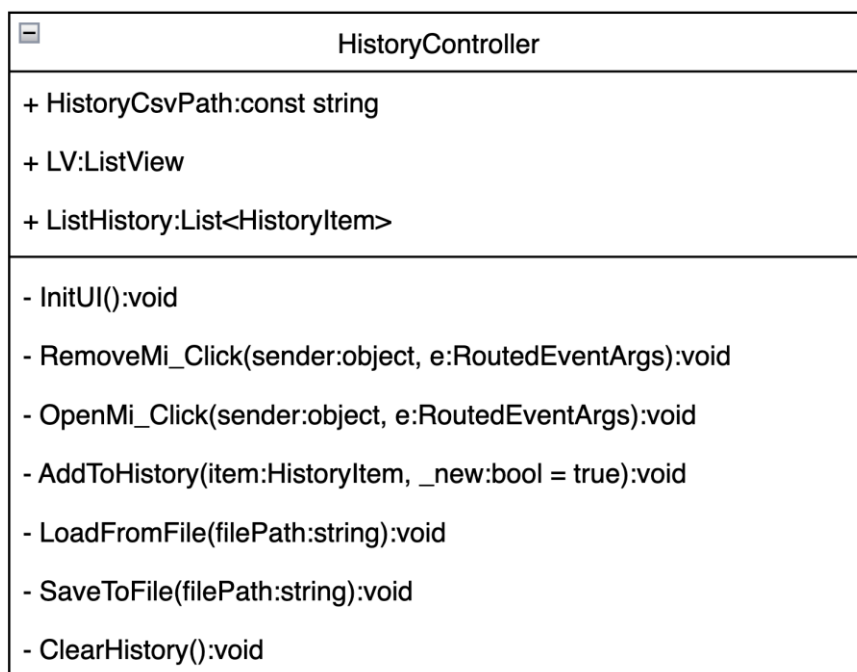


Рисунок 3 – Диаграмма класса `HistoryController`

2.4 Класс FavouritesController

Данный класс содержит следующие публичные поля:

- `string FavouritesCsvPath` – данное константное поле содержит путь до файла с данными об закладках пользователя в формате CSV;
- `ListView LV` – данное поле содержит элемент отображения списка избранного и используется для контролирования отображаемой информации;
- `List<FavouritesItem> ListFav` – данное поле содержит список объектов `FavouritesItem` закладок пользователя.

Также данный класс содержит следующие приватные методы.

`void InitUI()` – данный метод используется для инициализации и отображения элементов закладок пользователя.

`void RemoveMi_Click(object sender, System.Windows.RoutedEventArgs e)` – данный метод содержит логику нажатия на кнопку удаления в контекстном меню списка закладок пользователя. Параметры метода сгенерированы фреймворком.

`void OpenMi_Click(object sender, System.Windows.RoutedEventArgs e)` – данный метод содержит логику нажатия на кнопку открытия в контекстном меню списка закладок браузера. Параметры метода сгенерированы фреймворком.

`void AddToFavourites(FavouritesItem item, bool _new)` – данный метод используется для добавления нового элемента в список закладок пользователя. Параметр `item` типа `FavouritesItem` содержит добавляемую запись, параметр `_new` типа `bool` необходим для заполнения списка уже существующими значениями без добавления их в файл, значение по умолчанию – `true`.

`void LoadFromFile(string filePath)` – данный метод используется для загрузки списка закладок пользователя из файла. Параметр `filePath` типа `string` обозначает путь к файлу для загрузки данных.

`void SaveToFile(string filePath)` – данный метод используется для сохранения списка закладок пользователя в файл. Параметр `filePath` типа `string` обозначает путь к файлу для сохранения данных.

На рисунке 4 представлена диаграмма класса FavouritesController.

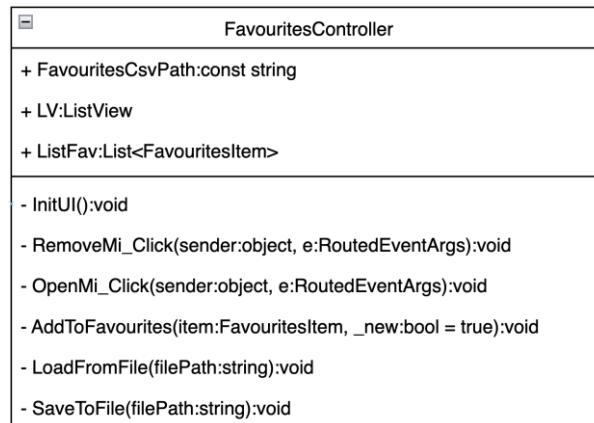


Рисунок 4 – Диаграмма класса FavouritesController

2.5 Структура FavouritesItem

Данная структура содержит следующие публичные поля:

- string Link – данное поле содержит ссылку на страницу, добавленную пользователем в закладки;
- string Title – данное поле содержит название закладки.

На рисунке 5 представлена диаграмма структуры FavouritesItem.

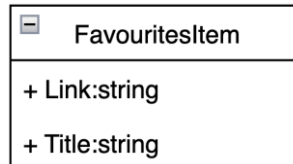


Рисунок 5 – Диаграмма структуры FavouritesItem

2.6 Структура HistoryItem

Данная структура содержит следующие публичные поля:

- string Link – данное поле содержит ссылку на страницу, посещенную пользователем;
- DateTime TimeVisited – данное поле содержит время посещения страницы пользователем.

На рисунке 6 представлена диаграмма структуры HistoryItem.

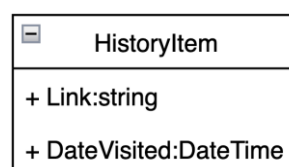


Рисунок 6 – Диаграмма структуры HistoryItem

2.7 Демонстрация иерархии классов

Разработанная иерархия наглядно продемонстрирована на диаграмме классов, приведённой на рисунке 7.

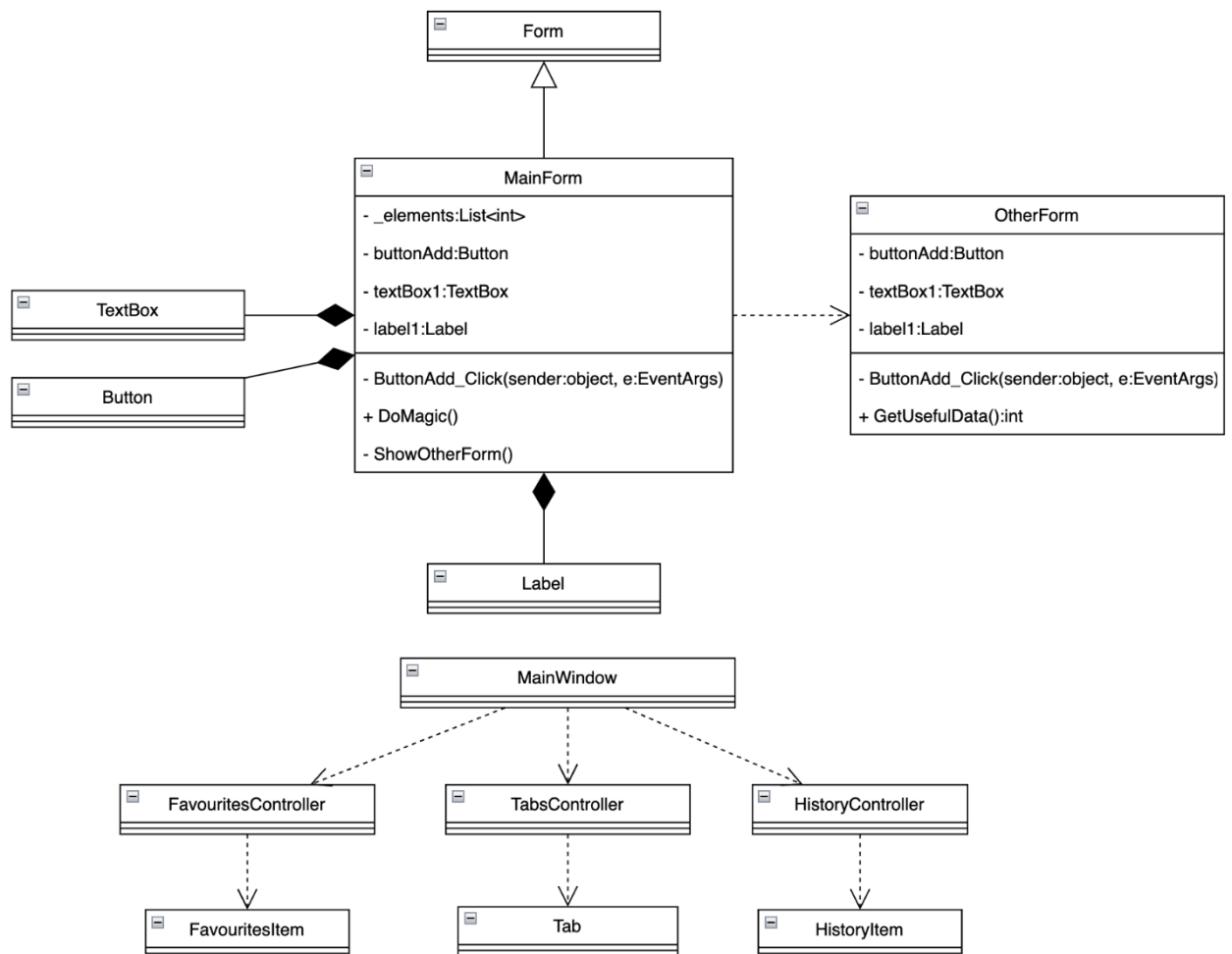


Рисунок 7 – Диаграмма классов иерархии «Веб-браузер»

Разработанная иерархия классов позволит реализовать поставленную задачу в соответствии со сформулированными требованиями.

3 Используемые мультимедийные ресурсы и сторонние библиотеки

Для реализации задания был использован отдельный компонент отображения веб страниц – WebView2 [4].

Разработка собственного компонента отображения является крайне нетривиальной и объемно задачей, создание такового требует большого количества времени и усилий команды разработчиков. В связи с этим, было принято решение использовать готовый компонент отображения.

Использование приведенного компонента позволит ускорить, упростить разработку и повысит качество результирующего продукта.

4 Демонстрация работы

На рисунке 8 изображена стартовая страница браузера со списком избранного и списком истории.

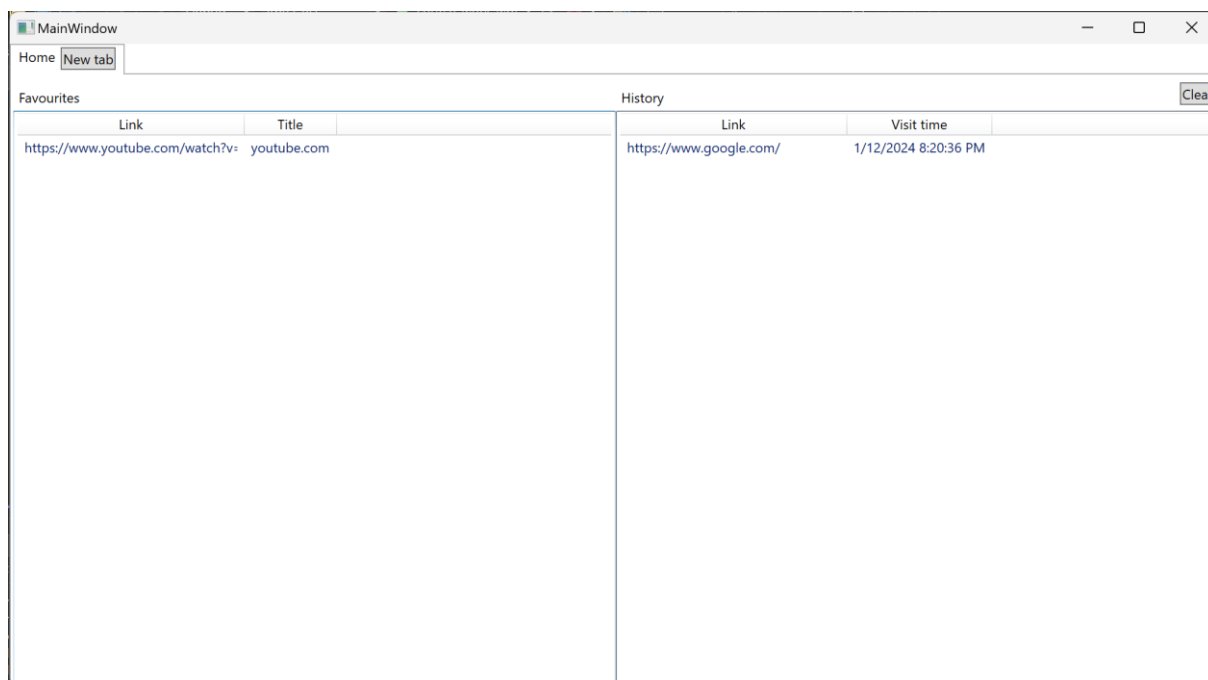


Рисунок 8 – Стартовая страница браузера

На рисунке 9 изображено добавление открытие новой вкладки. Начальный адрес новой вкладки ведет на стартовую страницу поисковой системы google [5].

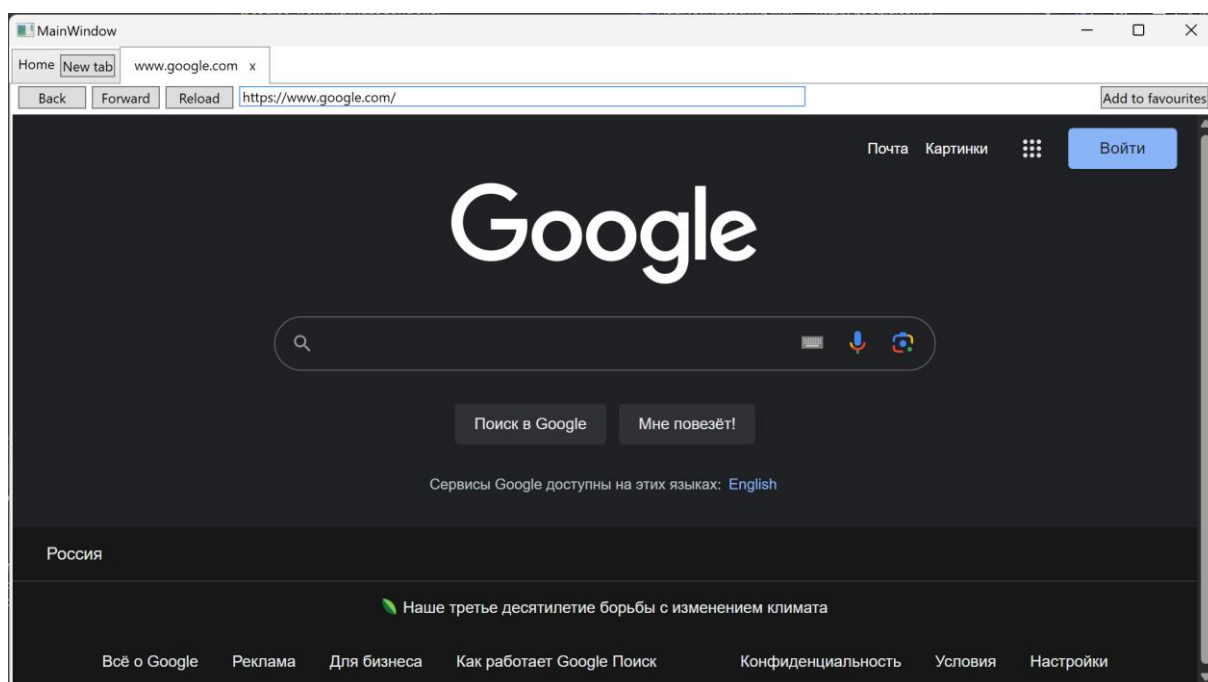


Рисунок 9 – Открытие новой вкладки

На рисунке 10 изображен переход по ссылке из адресной строки.

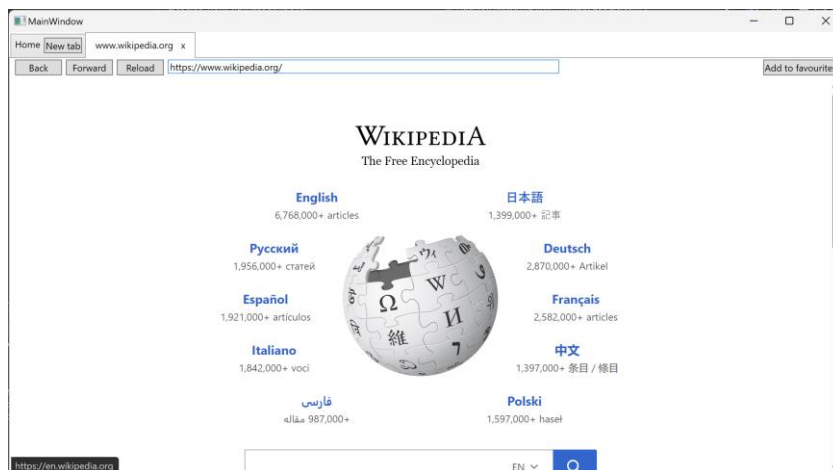


Рисунок 10 – Переход по ссылке из адресной строки

На рисунке 11 изображена обновленная история браузера.

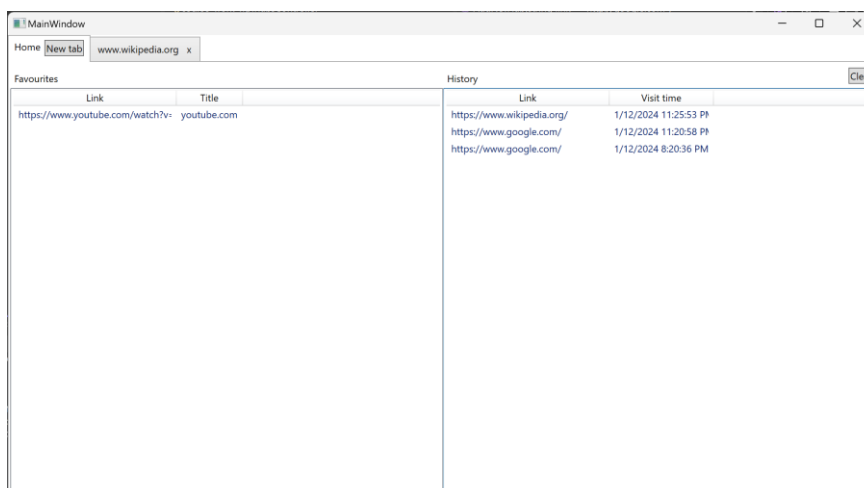


Рисунок 11 – Обновленная история браузера

На рисунке 12 изображено обновление списка закладок пользователя после добавления последней посещенной страницы в закладки.

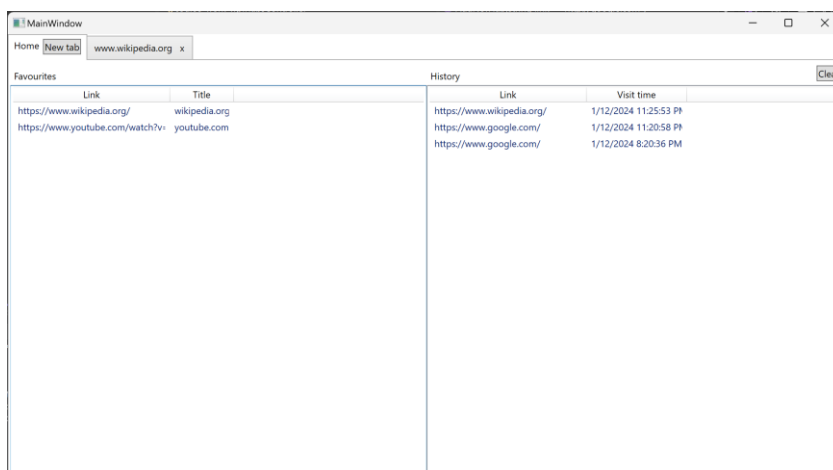


Рисунок 12 – Обновленный список закладок пользователя

На рисунках 13 и 14 изображено контекстное меню с возможности удаления и перехода по хранимой ссылке для списка закладок и истории соответственно.

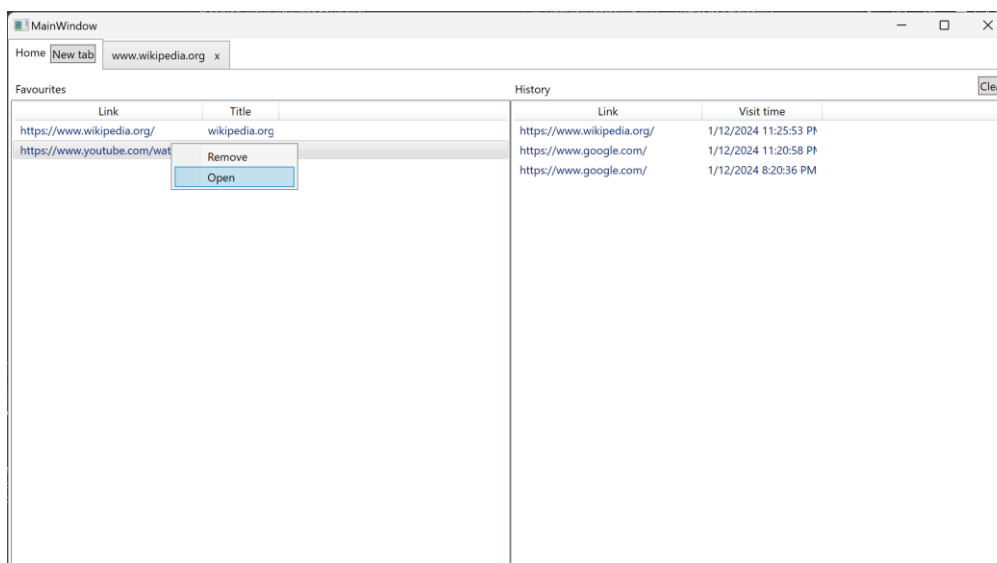


Рисунок 13 – Контекстное меню списка закладок пользователя

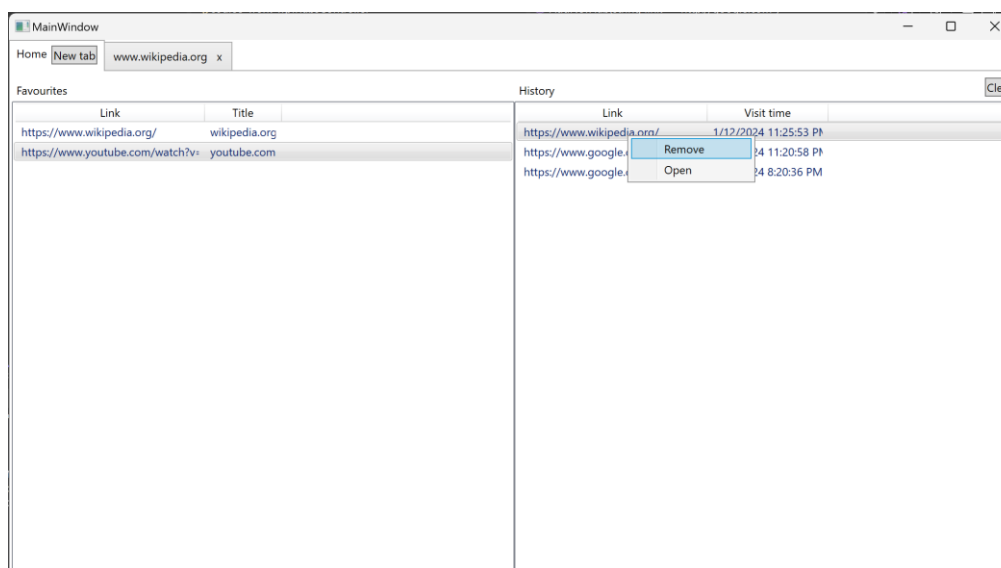


Рисунок 14 – Контекстное меню списка истории браузера

На рисунках 15 и 16 изображены файлы с данными истории браузера и закладками пользователя соответственно.

```
majiaxit@majiaxits-MacBook-Air Resources % cat history.csv
https://www.wikipedia.org/,1/12/2024 11:25:53 PM
https://www.google.com/,1/12/2024 11:20:58 PM
https://www.google.com/,1/12/2024 8:20:36 PM
```

Рисунок 15 – Файл с историей браузера

```
majiaxit@majiaxits-MacBook-Air Resources % cat favourites.csv
https://www.wikipedia.org/,wikipedia.org
https://www.youtube.com/watch?v=-WdYo3WlETY,youtube.com
```

Рисунок 16 – Файл с закладками пользователя

На рисунке 17 изображено окно браузера в полноэкранном режиме с большим количеством вкладок.

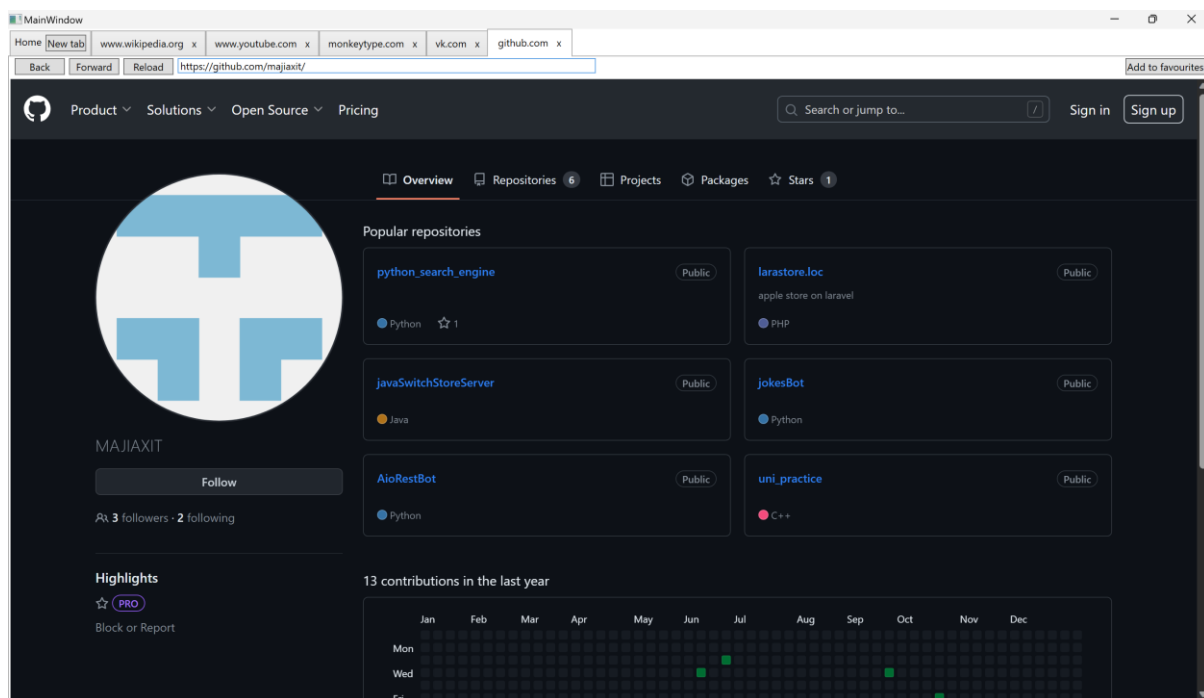


Рисунок 17 – Полноэкранное окно браузера с большим количеством вкладок

Демонстрация возможностей приложения подтверждает его соответствие с выдвинутыми требованиями и поставленной задачей.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были описаны основные требования к разрабатываемому приложению. Была составлена иерархия классов и применены методы разработки «сверху-вниз» для описания иерархии. Была проведена демонстрация и полное тестирование разработанного приложения.

Результатом работы является полноценный веб-браузер, поддерживаемый операционной системой Windows. Исходный код программы находится в Приложении А.

Все задачи курсовой работы выполнены, цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Windows Presentation Foundation. – URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netdesktop-8.0> (дата обращения 01.12.2023).
2. Native app definition. – URL: <https://uxcam.com/glossary/native-app/> (дата обращения 04.12.2023).
3. CSV File format. – URL: <https://docs.fileformat.com/spreadsheet/csv/> (дата обращения 04.12.2023).
4. Компонент отображения веб-страниц. – URL: <https://learn.microsoft.com/ru-ru/microsoft-edge/webview2/get-started/wpf> (дата обращения 02.12.2023).
5. Google Search Engine. – URL: <https://www.google.com> (дата обращения 03.12.2023).

ПРИЛОЖЕНИЕ А

Текст программы

Исходные тексты программы располагаются на прилагаемом электронном носителе.