1 Задание

Зотов Антон 11-106

1. Скачать минимум 100 текстовых страниц с помощью краулера.
2. Записать каждую страницу (без html-кода) в отдельный текстовый файл.
3. Создать файл index.txt, в котором хранится номер документа и ссылка на страницу.

```
PS C:\Users\1etmehear\RiderProjects\NAZVANIE1\NAZVANIE1> dotnet run -- https://render.ru/xen/  https://www.gameprogrammingpatterns.com/contents.html
1665
Сохранена страница 1: https://render.ru/xen/
53
Страница https://www.gameprogrammingpatterns.com/contents.html содержит менее 1000 слов, пропускаем
1252
Сохранена страница 2: http://render.ru/
1567
Сохранена страница 3: https://render.ru/xen/whats-new/posts/
540
Страница https://render.ru/xen/search/?type=post содержит менее 1000 слов, пропускаем
290
Страница https://render.ru/xen/whats-new/ содержит менее 1000 слов, пропускаем
36
Страница https://render.ru/xen/whats-new/profile-posts/ содержит менее 1000 слов, пропускаем
400
Страница https://render.ru/xen/whats-new/latest-activity содержит менее 1000 слов, пропускаем
4
Страница https://render.ru/ru/artist//about содержит менее 1000 слов, пропускаем
```

Рис.1 – Вызов программы в терминале

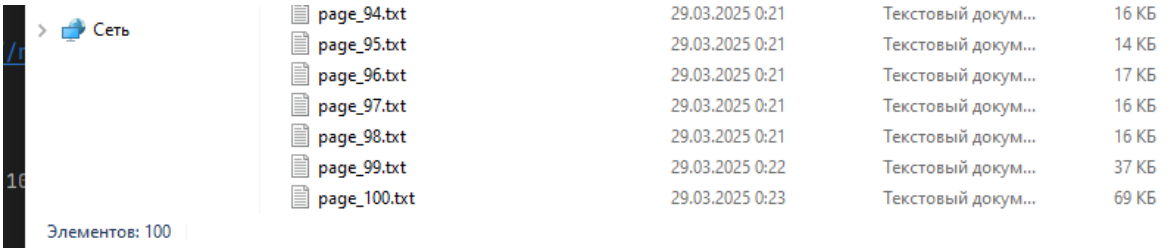

Рис. 2 – Сохраненные страницы

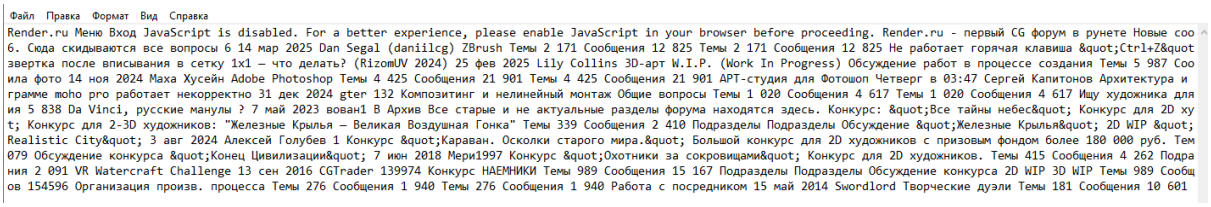
Файл Правка Формат Вид Справка
Render.ru Меню Вход JavaScript is disabled. For a better experience, please enable JavaScript in your browser before proceeding. Render.ru - первый CG форум в рунете Новые соо
6. Сюда скидываются все вопросы 6 14 мар 2025 Dan Segal (daniilcg) ZBrush Темы 2 171 Сообщения 12 825 Темы 2 171 Сообщения 12 825 Не работает горячая клавиша "Ctrl+Z&quot
звертка после вписывания в сетку 1x1 — что делать? (RizomUV 2024) 25 фев 2025 Lily Collins 3D-арт W.I.P. (Work In Progress) Обсуждение работ в процессе создания Темы 5 987 Соо
ила фото 14 ноя 2024 Маха Хусейн Adobe Photoshop Темы 4 425 Сообщения 21 901 Темы 4 425 Сообщения 21 901 АРТ-студия для Фотошоп Четверг в 03:47 Сергей Капитонов Архитектура и
грамме moho pro работает некорректно 31 дек 2024 gter 132 Композитинг и нелинейный монтаж Общие вопросы Темы 1 020 Сообщения 4 617 Темы 1 020 Сообщения 4 617 Ищу художника для
ия 5 838 Da Vinci, русские манулы ? 7 май 2023 ован1 В Архив Все старые и не актуальные разделы форума находятся здесь. Конкурс: "Все тайны небес" Конкурс для 2D ху
t; Конкурс для 2-3D художников: "Железные Крылья — Великая Воздушная Гонка" Темы 339 Сообщения 2 410 Подразделы Подразделы Обсуждение "Железные Крылья" 2D WIP "
Realistic City" 3 авг 2024 Алексей Голубев 1 Конкурс "Караван. Осколки старого мира." Большой конкурс для 2D художников с призовым фондом более 180 000 руб. Тем
079 Обсуждение конкурса "Конец Цивилизации" 7 июн 2018 Мери1997 Конкурс "Охотники за сокровищами" Конкурс для 2D художников. Темы 415 Сообщения 4 262 Подра
ния 2 091 VR Watercraft Challenge 13 сен 2016 CGTrader 139974 Конкурс НАЕМНИКИ Темы 989 Сообщения 15 167 Подразделы Подразделы Обсуждение конкурса 2D WIP 3D WIP Темы 989 Сообщ
ов 154596 Организация произв. процесса Темы 276 Сообщения 1 940 Темы 276 Сообщения 1 940 Работа с посредником 15 май 2014 Swordlord Творческие дуэли Темы 181 Сообщения 10 601

Рис. 3 – Пример страницы

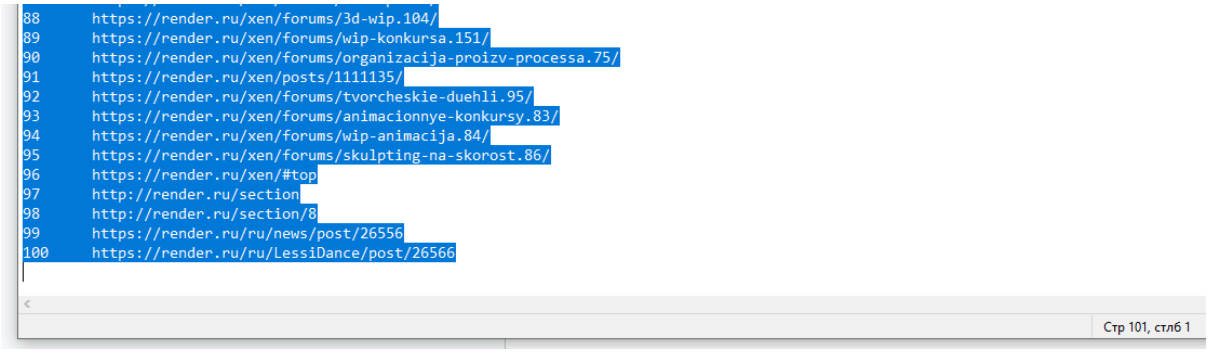

Рис. 4 – index.txt

```
1 usage
class WebCrawler
{
    private readonly List<string> _startUrls;
    private readonly int _minPages;
    private readonly int _minWords;
    private readonly HashSet<string> _visitedUrls = new ();
    private int _pagesDownloaded = 0;
    private const string IndexFile = "index.txt";
    private const string OutputDir = "pages";

    1 usage
    public WebCrawler(List<string> startUrls, int minPages = 100, int minWords = 1000)
    {
        _startUrls = startUrls;
        _minPages = minPages;
        _minWords = minWords;

        Directory.CreateDirectory( path: OutputDir);

        File.WriteAllText(IndexFile,  contents: string.Empty);
    }

    1 usage
    private bool IsValidUrl(string url)
    {
        // Проверяем, что URL валиден и не является файлом (например, .pdf)
        if (!Uri.TryCreate(url, UriKind.Absolute, out var uri))
            return false;

        if (uri.Scheme != Uri.UriSchemeHttp && uri.Scheme != Uri.UriSchemeHttps)
            return false;

        var invalidExtensions = new[] { ".pdf", ".jpg", ".png", ".gif", ".zip", ".doc", ".docx" };
        return !invalidExtensions.Any(ext :string  => uri.AbsolutePath.EndsWith(ext, StringComparison.OrdinalIgnoreCase));
    }
```

Рис. 5 – Конструктор краулера + проверка валидных ссылок

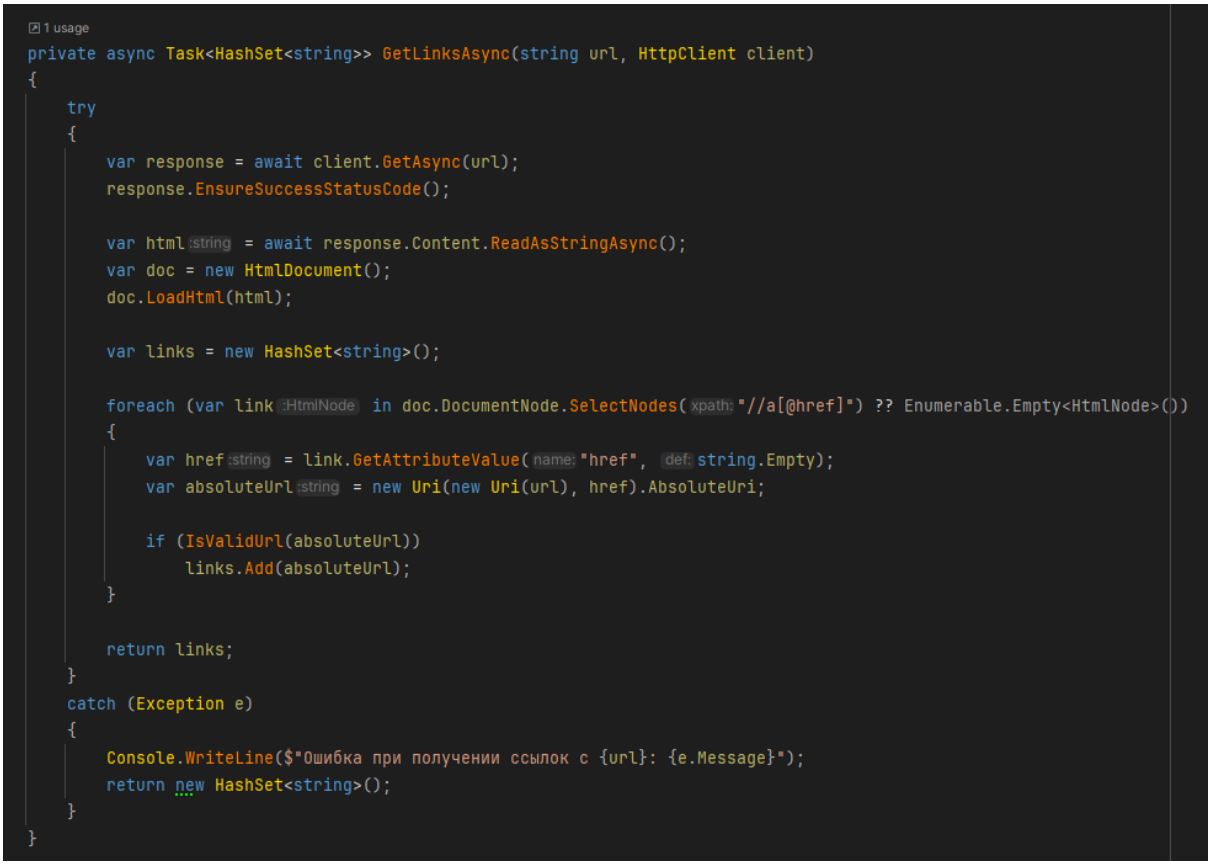

Рис. 6 – Получение ссылок из страницы. Извлекает текстовое содержимое из HTML. Удаляет ненужные элементы (скрипты, стили, навигацию и т.д.). Очищает текст от лишних пробелов и переносов строк

```
private string ExtractText(string html)
    var doc = new HtmlDocument();
    doc.LoadHtml(html);

    var nodesToRemove :HtmlNodeCollection = doc.DocumentNode.SelectNodes( xpath: "//script|//style|//nav|//footer|//head|//iframe");

    foreach (var node in nodesToRemove)
        node.Remove();

    var text = doc.DocumentNode.InnerText;

    // Удаляем лишние пробелы и переносы строк
    text = Regex.Replace( input: text,  pattern: @"\s+",  replacement: " ").Trim();

    return text;
}

1 usage
private bool SavePage(string url, string text)
{
    var words :string[] = text.Split(new[] { ' ', '\t', '\n' }, StringSplitOptions.RemoveEmptyEntries);

    // foreach(var word in words)
    Console.WriteLine(words.Length);

    var wordCount = words.Length;
    if (wordCount < _minWords)
    {
        Console.WriteLine($"Страница {url} содержит менее {_minWords} слов");
        return false;
    }

    _pagesDownloaded++;
    var filename = Path.Combine(OutputDir, $"page_{_pagesDownloaded}.txt");

    File.WriteAllText(filename,  contents: text, Encoding.UTF8);

    File.AppendAllText( path: IndexFile,  contents: $"{_pagesDownloaded}\t{url}{Environment.NewLine}", Encoding.UTF8);

    Console.WriteLine($"Страница сохранена {_pagesDownloaded}: {url}");
    return true;
```

Рис. 7 – Метод для парсинга страницы + сохранение страницы в txt файл. Проверяет количество слов на странице. Если слов достаточно, сохраняет текст в файл

```
1 usage
private async Task<bool> DownloadPageAsync(string url, HttpClient client)
{
    if (_visitedUrls.Contains(url) || _pagesDownloaded >= _minPages)
        return false;

    _visitedUrls.Add(url);

    try
    {
        var response = await client.GetAsync(url);
        response.EnsureSuccessStatusCode();

        var html :string = await response.Content.ReadAsStringAsync();
        var text = ExtractText(html);


        if (string.IsNullOrWhiteSpace(text))
            return false;

        return SavePage(url, text);
    }
    catch (Exception e)
    {
        Console.WriteLine($"Ошибка при скачивании {url}: {e.Message}");
        return false;
    }
}
```

Рис. 8 – Метод для скачивания страницы. Проверяет, не была ли страница уже загружена. Асинхронно скачивает страницу. Извлекает текст и сохраняет его.

```
1 usage
public async Task CrawlAsync()
{
    var handler = new HttpClientHandler
    {
        AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate
    };

    using var client = new HttpClient(handler);
    client.DefaultRequestHeaders.UserAgent.ParseAdd( input: "Mozilla/5.0");
    client.Timeout = TimeSpan.FromSeconds(10);

    var currentLevelUrls = new HashSet<string>(_startUrls);
    var nextLevelUrls = new HashSet<string>();

    while (_pagesDownloaded < _minPages && currentLevelUrls.Count > 0)
    {
        foreach (var url :string in currentLevelUrls)
        {
            if (_pagesDownloaded >= _minPages)
                break;

            if (await DownloadPageAsync(url, client))
            {
                var links :HashSet<string> = await GetLinksAsync(url, client);
                foreach (var link :string in links)
                    nextLevelUrls.Add(link);
            }

            // Небольшая задержка чтобы не перегружать сервер
            await Task.Delay(1000);
        }

        currentLevelUrls = new HashSet<string>( collection: nextLevelUrls.Except(_visitedUrls));
        nextLevelUrls.Clear();
    }

    Console.WriteLine($"Завершено. Скачано {_pagesDownloaded} страниц.");
}
```

Рис. 9 – Основной метод краулера. Имитирует работу браузера, настраивает распаковку сжатых ответов, в цикле с самого начала проходится по стартовым ссылкам. Если скачивание удачно, извлекает все ссылки, добавляет их в основной хэшсет