

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное
образовательное бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

Дипломный проект допущен к защите
Заместитель директора по
воспитательной и методической работе
_____ к.п.н., Глазырина Т. Г.
« _____ » _____ 2024 г.

ДИПЛОМНЫЙ ПРОЕКТ

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИЗВЛЕЧЕНИЯ
ИНФОРМАЦИИ ИЗ БОЛЬШИХ ОБЪЕМОВ ТЕКСТОВЫХ ДАННЫХ**

Выполнила
Поглазова Валерия Владимировна
студент специальности 09.02.07
Информационные системы
и программирование
группа 21П-1
Форма обучения: очная

(подпись)

Руководитель:
Калинин Арсений Олегович

(подпись)

Дипломный проект защищен

« _____ » _____ 2024 г.

Оценка _____

Секретарь ГЭК _____

Слободской
2024

Нормоконтроль:

Дата: _____

Подпись

Расшифровка подписи

ОГЛАВЛЕНИЕ

	Стр.
ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	6
1.1. Анализ предметной области.....	6
1.2. Техническое задание.....	9
ВЫВОД ПО ГЛАВЕ 1	14
ГЛАВА 2. КОНСТРУКТОРСКАЯ ЧАСТЬ.....	15
2.1 Архитектура программы.....	15
2.2. Описание алгоритмов и функционирования программы.....	18
ВЫВОД ПО ГЛАВЕ 2	21
ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНО-ПРИКЛАДНАЯ ЧАСТЬ.....	22
3.1. Тестирование и опытная эксплуатация программы.....	22
3.2. Руководство пользователя.....	25
ВЫВОДЫ ПО ГЛАВЕ 3.....	31
ЗАКЛЮЧЕНИЕ.....	32
СПИСОК ЛИТЕРАТУРЫ.....	34
ПРИЛОЖЕНИЯ.....	37

ВВЕДЕНИЕ

В наше время наблюдается стремительный рост объемов цифровых данных, значительная часть которых представлена в текстовом формате. Эти данные хранятся в различных форматах и источниках: от документов и веб-страниц до социальных сетей и баз данных. Однако, эта информация часто является неструктурированной или полуструктурированной, что затрудняет её анализ и использование для принятия решений. Извлечение полезной информации из таких массивов данных вручную является трудоемким и неэффективным процессом.

Актуальность данной темы заключается в том, что эффективное извлечение информации из больших объемов текстовых данных является критически важной задачей для многих областей, включая науку, бизнес, государственное управление и другие. Автоматизация этого процесса позволяет значительно сократить время и затраты на анализ данных, повысить точность и объективность результатов, а также открыть новые возможности для принятия информированных решений.

В процессе проведения исследования выявляется, что существующие решения для обработки больших текстовых данных часто обладают недостаточной гибкостью, не поддерживают широкий спектр форматов данных или имеют ограниченную функциональность. Многие инструменты требуют значительных вычислительных ресурсов или обладают высокой стоимостью.

Разработка специализированных программ для извлечения информации из больших объемов текстовых данных становится все более актуальной задачей в связи с постоянным увеличением количества цифровых данных, ростом сложности задач анализа данных и необходимостью повышения эффективности работы с информацией. Это особенно важно в условиях, когда требуется обработка данных из различных источников и в разных форматах.

Таким образом, разработка программы "Извлечение информации из

больших объемов текстовых данных" решает конкретные проблемы неэффективности и трудоемкости ручного анализа текстовых данных, но и подчеркивает актуальность автоматизации процессов обработки больших данных. Программа позволит эффективно извлекать, структурировать и анализировать информацию из различных источников, что существенно повысит производительность и качество работы с данными. Поэтому данную программу целесообразно написать на языке программирования C#, учитывая его широкие возможности для работы с данными, наличие развитых библиотек для обработки текста и высокую производительность.

Язык программирования C# предоставляет разработчику поистине великолепный набор простых в использовании инструментов, позволяющих быстро разрабатывать сложные проекты, создавая приятный и удобный пользовательский интерфейс.

Объект исследования: процесс извлечения структурированной информации из больших объемов неструктурированных текстовых данных.

Предмет исследования: разработка программного обеспечения для извлечения информации из больших объемов текстовых данных

Цель дипломного проекта – заключается в разработке и реализации программного обеспечения, способного эффективно извлекать заданную информацию из больших объемов текстовых данных, преобразовывать ее в удобный для анализа и дальнейшей обработки структурированный формат, обеспечивая высокую скорость и точность извлечения.

Задачи исследования:

- Описать предметную область.
- Разработать технического задание на создание программного продукта.
- Описать архитектуру программы.
- Описать алгоритмы и функционирование программы.
- Провести тестирование и опытную эксплуатацию.

- Разработать руководство оператора

Методы исследования - системный анализ и функциональное моделирование.

Практическая значимость: заключается в создании инструмента, позволяющего автоматизировать и ускорить процесс извлечения и структурирования информации из больших объемов текстовых данных.

ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1. Анализ предметной области

Извлечение информации из текстовых данных — это процесс автоматического извлечения структурированной информации из неструктурированных или полуструктурированных текстовых данных.

Извлечение информации из больших объемов текстовых данных разрабатывалась для автоматизации процесса анализа и извлечения ключевой информации из неструктурированных и полуструктурированных текстовых данных, хранящихся в различных форматах и источниках (документы, веб-страницы, социальные сети и т.д.).

Этот процесс был осуществлен с целью повышения эффективности и производительности обработки больших объемов текстовых данных, снижения трудозатрат на ручной анализ, повышения точности и объективности результатов анализа, а также обеспечения возможности принятия более обоснованных решений на основе полученной информации.

Благодаря извлечению информации из больших объемов текстовых данных можно обнаружить скрытые закономерности и тренды, улучшить принятие решений в бизнесе, науке и государственном управлении, оптимизировать процессы обработки информации, получить конкурентное преимущество за счет быстрого доступа к важной информации, автоматизировать мониторинг СМИ и социальных сетей.

Извлечение информации из больших объемов текстовых данных представляет собой комплексный процесс, включающий в себя этапы загрузки данных, предобработки, извлечения ключевых сущностей, анализ настроений, структурирование данных и визуализации данных.

Цель извлечения состоит в том, чтобы преобразовать неструктурированные или полуструктурированные текстовые данные в

структурированный формат, удобный для дальнейшего анализа и использования, извлечь из них конкретную информацию и выявить закономерности, необходимые для решения конкретных задач пользователя.

Также в процессе анализа было проведено исследование предметной области, в результате которого была создана диаграмма вариантов использования, которая отражает актера (пользователя) и выполняемые им функции (Рисунок 1.1).

На ней видно, что в системе выделяется только один центральный пользователь, который:

- Пользователь – может загружать текстовые данные из различных источников и форматов (файлы, URL); производить фильтрацию; редактировать данные; сохранять результаты анализа в форматах (CSV, JSON, XML) для дальнейшей обработки в других системах.

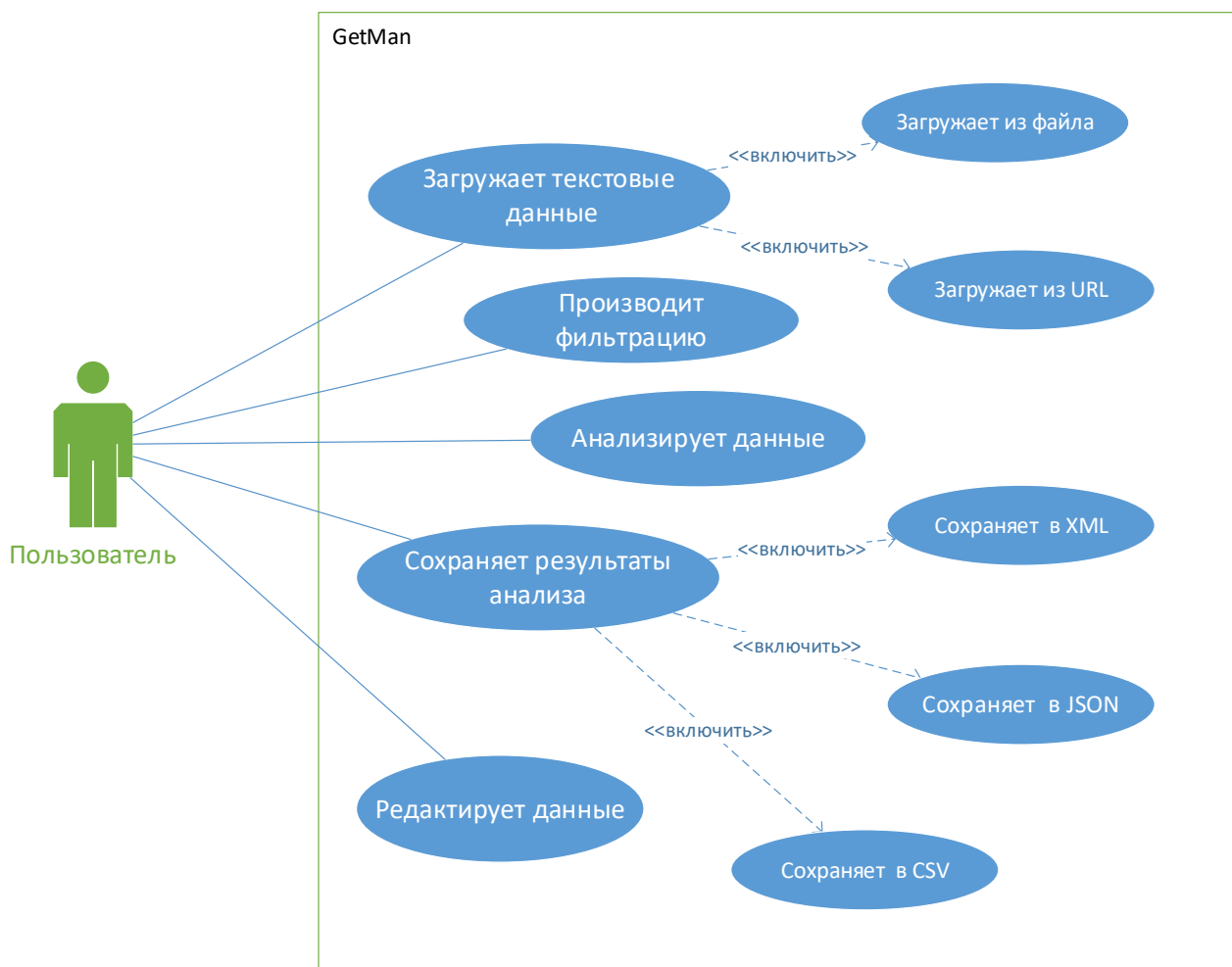


Рисунок 1.1 - Диаграмма вариантов использования

Анализ программы аналога.

Postman — это платформа для разработки и тестирования API. Она представляет собой интегрированную среду разработки (IDE), которая позволяет разработчикам создавать, отправлять и тестировать HTTP-запросы, управлять коллекциями запросов, автоматизировать тестирование, и получать данные о производительности API.

Плюсы Postman:

1. Удобный интерфейс: интуитивно понятный и простой в использовании интерфейс, даже для начинающих разработчиков.
2. Автоматизация запросов: Postman позволяет автоматизировать отправку запросов к веб-сайтам и API, что может сэкономить время при извлечении больших объемов данных.
3. Многофункциональность: Postman поддерживает множество типов запросов (GET, POST, PUT, DELETE и др.), а также различные форматы данных (JSON, XML, и другие). Поддерживает автодополнение, что ускоряет создание запросов.
4. Экспорт данных: Postman позволяет экспортировать полученные данные в различные форматы, такие как CSV и JSON, что упрощает дальнейший анализ.
5. Бесплатная версия с достаточным функционалом: доступна бесплатная версия, которая вполне подходит для многих задач.

Минусы Postman:

1. Ограничения бесплатной версии: бесплатная версия имеет некоторые ограничения по количеству запросов, коллекций и пользователей.
2. Цена платной версии: платная версия может быть довольно дорогой для отдельных разработчиков или небольших команд.
3. Зависимость от интернета: для работы Postman требуется подключение к интернету.

4. Сложность для очень сложных сценариев: для очень сложных и многоуровневых API-сценариев или работы с необычными форматами данных, Postman может показаться несколько ограниченным, особенно в сравнении с инструментами программирования.

5. Ограничения в обработке больших объемов данных: не самый подходящий инструмент для очень больших объемов данных из-за его фокуса на взаимодействие с API и работе с небольшими наборами данных.

6. Слишком много возможностей: для новичка может показаться перегруженным из-за большого набора инструментов и функций, которые не всегда нужны.

1.2. Техническое задание

Техническое задание разрабатывалось на основании заявки (Приложение 1) и требований заказчика, программы «Извлечение информации из больших объемов текстовых данных», анализа предметной области и с учетом требований ГОСТ 19.201-78 [5].

Наименование программы – «GetMan». Программа предназначена для извлечения информации из больших объемов текстовых данных.

Разработка программы ведется на основании учебного плана и перечня тем утвержденных на заседании предметно цикловой комиссии информатики и программирования.

Функциональным назначением программы является извлечение информации из больших объемов текстовых данных.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Загрузка данных;
- Предобработка данных (Очистка данных: Удаление лишних символов, специальных символов и т.д.);

- Извлечение информации;
- Формирование структурированных данных;
- Сохранение результатов.

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организация бесперебойного питания технических средств;
- использование лицензионного программного обеспечения;
- отсутствие вредоносного программного обеспечения, наличие антивирусной программы;
- соблюдение правил и требований по эксплуатации технических средств.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 5 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий себя:

- процессор с тактовой частотой, 1 ГГц, не менее;
- оперативную память объемом 512 Мб, не менее;
- жесткий диск со свободным местом 500 Мб, не менее;
- монитор, с разрешением экрана 1024*768, не менее;
- компьютерная мышь;
- клавиатура;
- принтер.

Исходные коды программы должны быть реализованы на языке C#. В качестве интегрированной среды разработки программы должна быть использована среда программирования Microsoft Visual Studio 2022.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7/8/10/11.

Программное обеспечение поставляется в виде изделия на USB-флэш накопителе.

Требования к транспортировке и хранению должны соответствовать условиям эксплуатации носителей, на которых находится программный продукт.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

Предварительный состав программной документации включает в себя следующие документы:

- техническое задание;
- руководство оператора.

Разработка должна быть проведена в следующие стадии и этапы:

1. Анализ требований:

На стадии анализ требований формулируются цели и задачи проекта. Создается основа для дальнейшего проектирования

2. Проектирование:

На стадии проектирование должны быть выполнены перечисленные ниже этапы работ:

- разработка программной документации;

На этапе разработка программной документации должна быть выполнена разработка технического задания.

При разработке технического задания должны быть выполнены перечисленные работы: постановка задачи, определение и уточнение требований к техническим средствам, определение требований к программе, определение стадий, этапов и сроков разработки программы и документации на нее, выбор языков программирования.

- разработка алгоритма программы;

На этапе разработки алгоритма программы должен быть разработан алгоритм работы программы.

- кодирование;

На стадии кодирования происходит реализация алгоритмов в среде программирования.

- тестирование и отладка.

На стадии тестирование и отладка происходит проверка алгоритмов, реализованных в программе на работоспособность в различных ситуациях. Исправление выявленных ошибок, повторное тестирование.

Приемо-сдаточные испытания должны проводиться при использовании технических средств. Приемка программы заключается в проверке работоспособности программы путем ввода реальных или демонстрационных данных.

Во время приемки работы разработчик предоставляет программу и документацию, которая к ней прилагается. При успешных испытаниях программа вводится в эксплуатацию, а при критических ошибках — возвращается на доработку.

ВЫВОДЫ ПО ГЛАВЕ 1

В этой главе была описана аналитическая часть, содержащая в себе описание извлечения информации из больших объемов текстовых данных, который был изучен перед созданием программного продукта, а также техническое задание, содержащее в себе информацию о программном продукте, его функциях, эксплуатации и требования, которые должны учитываться при создании программы и документации к ней.

Определена значимость и требования к будущему программному обеспечению. В техническом задании были определены основные требования к программному продукту и функциональные характеристики, а также состав программной документации.

ГЛАВА 2. КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1. Архитектура программы

Разработанный программный продукт поставляется в виде установочного файла «getman.exe».

Весь код программы описан в разделе приложений (Приложение 2).

Приложение «GetMan», написанное на языке программирования C# в интегрированной среде разработки Microsoft Visual Studio 2022, работает под управлением операционной системы Windows 11.

В программном продукте реализовано одно главное окно. Окно предоставляет основной функционал программы, и вся работа происходит именно в нем. В основном окне реализованы следующие функции:

- Текстовое поле «Введите URL-адрес или вставьте текст» предназначено для ввода пользователем URL-адреса веб-страницы или файла, который требуется обработать;
- Кнопка «Открыть файл» предназначена для предоставления пользователю возможности выбора файла с локального диска для обработки.
- Кнопка «Отправить» предназначена для отправки введенного URL-адреса или файла в текстовом поле для извлечения данных;
- Список предназначен для отображения результатов обработки введенных данных;
- Выпадающий список предназначен для выбора формата файла, в который пользователь хочет сохранить результаты обработки. Доступные форматы: XML, JSON и CSV;
- Кнопка «Сохранить файл» предназначена для сохранения результатов обработки (отображаемых в списке) в файл выбранного формата (выбранного в выпадающем списке);

- Кнопка «Редактировать» предназначена для редактирования данных, отображаемых в списке.

Программа предоставляет гибкий инструмент для работы с данными в разных форматах, используя библиотеки:

- **Newtonsoft.Json** для JSON.
- **System.Xml** для XML.
- **CsvHelper** для CSV.

Фрагмент кода показан на рисунке 2.2.

```
private string ConvertXmlToJson(string xmlData)
{
    try
    {
        XmlDocument doc = new XmlDocument();
        doc.LoadXml(xmlData);    // Удаляем XML-декларацию, если она есть
        if (doc.FirstChild is XmlDeclaration)
            doc.RemoveChild(doc.FirstChild);
        // Создаем корневой элемент, если его нет
        if (doc.ChildNodes.Count > 1)
        {
            XmlElement root = doc.CreateElement("root");
            while (doc.HasChildNodes)
                root.AppendChild(doc.FirstChild);
            doc.AppendChild(root);
        } // Сериализуем XML в JSON с форматированием
        return JsonConvert.SerializeXmlNode(doc, Formatting.Indented, true);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка преобразования XML → JSON: {ex.Message}");
        return null;
    }
}
```

Рисунок 2.2 – Фрагмент кода преобразования в JSON

Как работает преобразования XML в JSON:

1. **Входные данные:** XML-строка.
2. **Извлечение данных XML:** Загрузка в XmlDocument.
3. **Нормализация:** Удаление декларации и добавление корневого элемента.
4. **Сериализация:** Преобразование в JSON с помощью JsonConvert.SerializeXmlNode().

Код модуля преобразования CSV в JSON показан на рисунке 2.3:

```
private string ConvertCsvToJson(string csvData)
{
    try
    {
        using (var reader = new StringReader(csvData))
        using (var csv = new CsvReader(reader, CultureInfo.InvariantCulture))
        {
            var records = csv.GetRecords<dynamic>().ToList();
            return JsonConvert.SerializeObject(records, Formatting.Indented);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка преобразования CSV → JSON: {ex.Message}");
        return null;
    }
}
```

Рисунок 2.3 – Код модуля работы с CSV в JSON

2.2. Описание алгоритмов и функционирования программы

Алгоритмы программы

Рассмотрим алгоритмы, используемые в программе. Ниже приведено их краткое описание.

- Запуск программы

При запуске программы отображается основное окно.

- Ввод входных данных (URL/файл/текст)

Программа позволяет вводить данные тремя способами: вручную указать URL-адрес, выбрать файл через диалоговое окно по кнопке «Открыть файл» или вставить произвольный текст, и затем определить формат данных (JSON, XML, CSV или TXT) для корректной обработки.)

- Редактирования данных

Функция создана для того, чтобы пользователь мог изменить данные перед сохранением. Нужно нажать на кнопку «Редактировать», после чего открывается модальное окно редактирования. После завершения редактирования изменения сохраняются в основном окне.

- Сохранения данных

Для того, чтобы сохранить данные в выбранный формат (JSON/XML/CSV) нужно нажать на кнопку «Сохранить», и выбрать путь, в котором вы хотите сохранить файл.

На рисунке 2.4 показана схема программы. Данная схема отображает алгоритм работы программы.

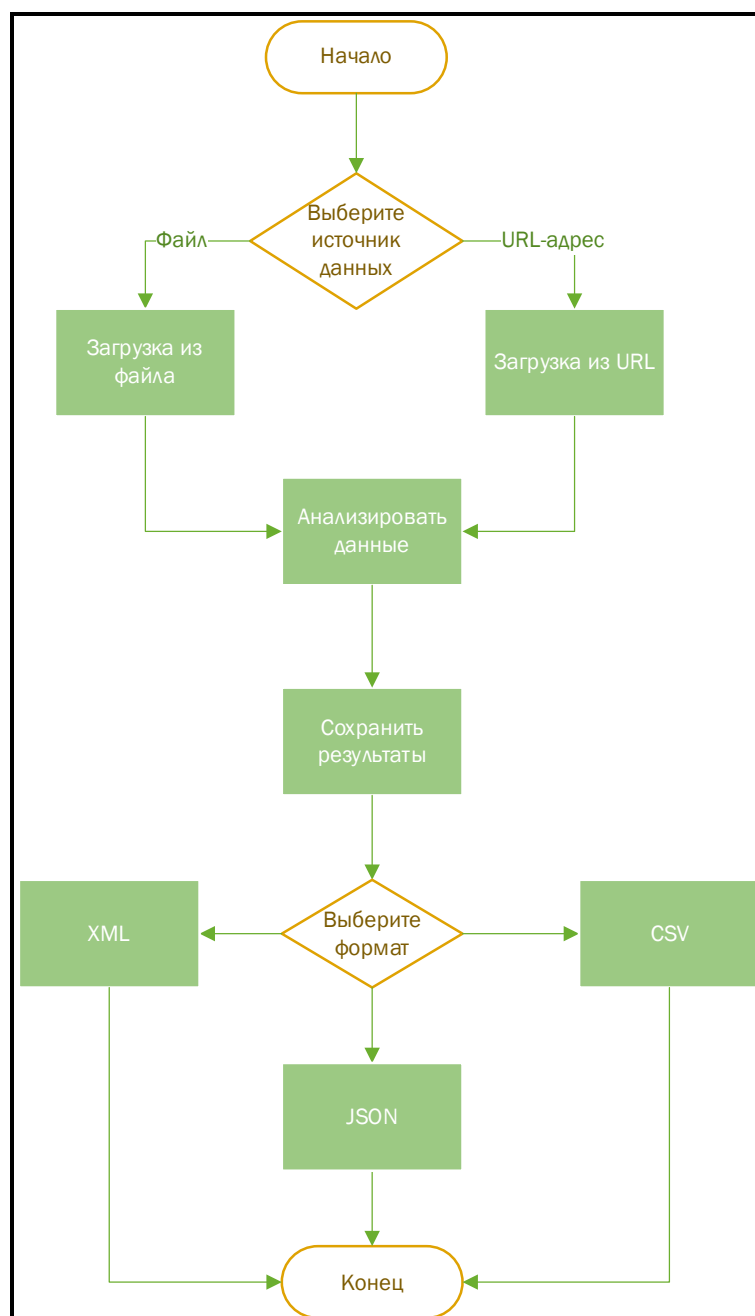


Рисунок 2.4 – Алгоритм работы программы

Функционирование программы

При запуске программы отображается основное окно программы, предназначенное для извлечения информации из текстовых данных.

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организация бесперебойного питания технических средств;

- использование лицензионного программного обеспечения;
- отсутствие вредоносного программного обеспечения, наличие антивирусной программы;
- обучение пользователей правилам безопасности и лучшим практикам работы с программным обеспечением.
- соблюдение правил и требований по эксплуатации технических средств.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 5 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Для корректной работы программы GetMan необходимо выполнить следующие шаги:

1. Установка программы:
 - Запустите установочный файл getman.exe.
 - Следуйте инструкциям мастера установки.

- По завершении программа будет доступна в меню Пуск и на рабочем столе.

2. Дополнительные требования:

- Для работы с интернет-источниками требуется стабильное подключение к сети.
- Для обработки больших файлов (более 100 МБ) рекомендуется 8+ ГБ ОЗУ.

ВЫВОДЫ ПО ГЛАВЕ 2

В результате над конструкторской частью была разработана архитектура созданного программного продукта, описаны алгоритмы и функционирования программы.

Для более наглядного понимания архитектуры программы и алгоритма её работы была построена схема работы программного обеспечения.

Для реализации функционала использовались современные библиотеки, такие как Newtonsoft.Json, System.Xml и CsvHelper, что позволило обеспечить корректное преобразование данных между форматами.

Для обеспечения стабильной работы программы определены требования к системе и условия эксплуатации, необходимые для стабильной работы приложения. Программа адаптирована для работы в операционной системе Windows 10/11.

Вся информация представлена для изучения структуры программы и ее работы.

ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНО-ПРИКЛАДНАЯ ЧАСТЬ

3.1. Тестирование и опытная эксплуатация программы

Для проведения тестирования программы мною было произведено базовое тестирование во время прохождения преддипломной практики, которое показало готовность программы к эксплуатации на предприятие.

- Преобразование данных из CSV в JSON.

Шаги тестирования:

1. Запустить приложение;
2. Ввести входные данных;
3. Нажать на кнопку «Отправить»;

Данные тестирования:

Ожидаемый результат: Данные занесенные в программу, обработаны.

Фактически результат: Данные занесенные в программу, обработаны (Рисунок 3.1).

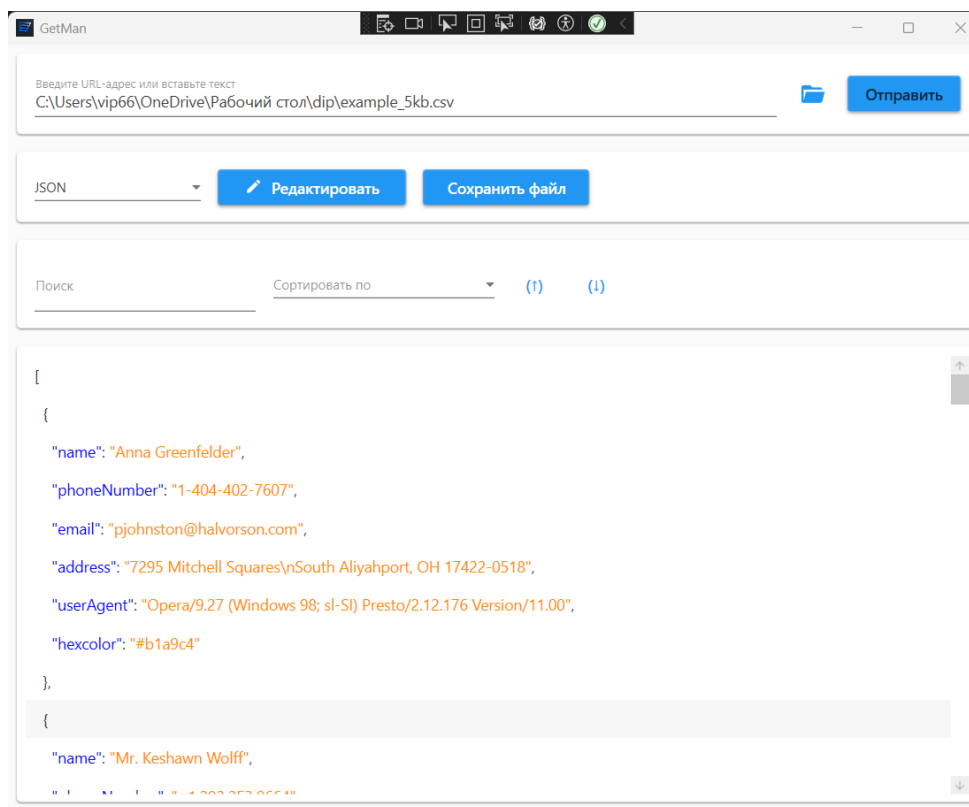


Рисунок 3.1 – Преобразование данных из CSV в JSON

- Преобразование данных из JSON в XML.

Шаги тестирования:

1. Запустить приложение;
2. Ввести входные данных;
3. Нажать на кнопку «Отправить»;

Данные тестирования:

Ожидаемый результат: Данные занесенные в программу, обработаны.

Фактически результат: Данные занесенные в программу, обработаны.

(Рисунок 3.2).

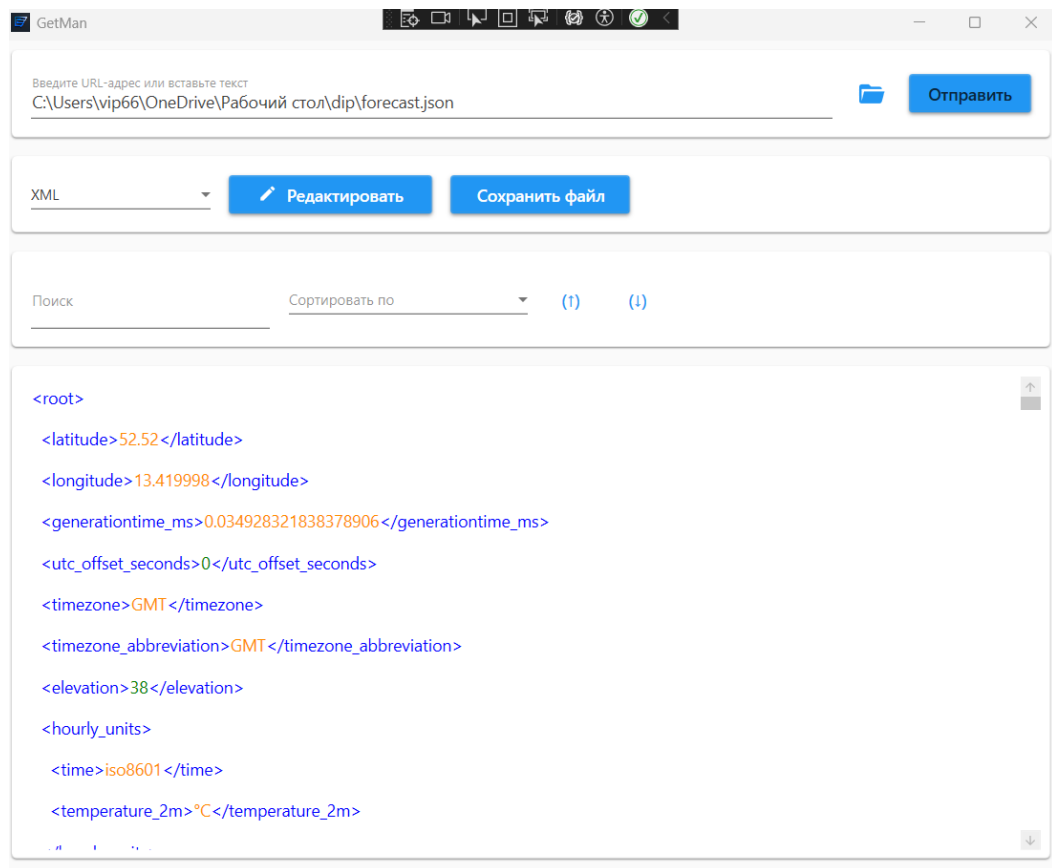


Рисунок 3.2 – Преобразование данных из JSON в XML

- Редактирование данных.

Шаги тестирования:

1. Запустить приложение;
2. Ввести входные данных;

3. Нажать на кнопку «Отправить»;
4. Нажать кнопку «Редактировать»;
5. Нажать кнопку «Сохранить»;

Данные тестирования:

Ожидаемый результат: Данные отредактированы.

Фактически результат: Данные отредактированы. (Рисунок 3.2).

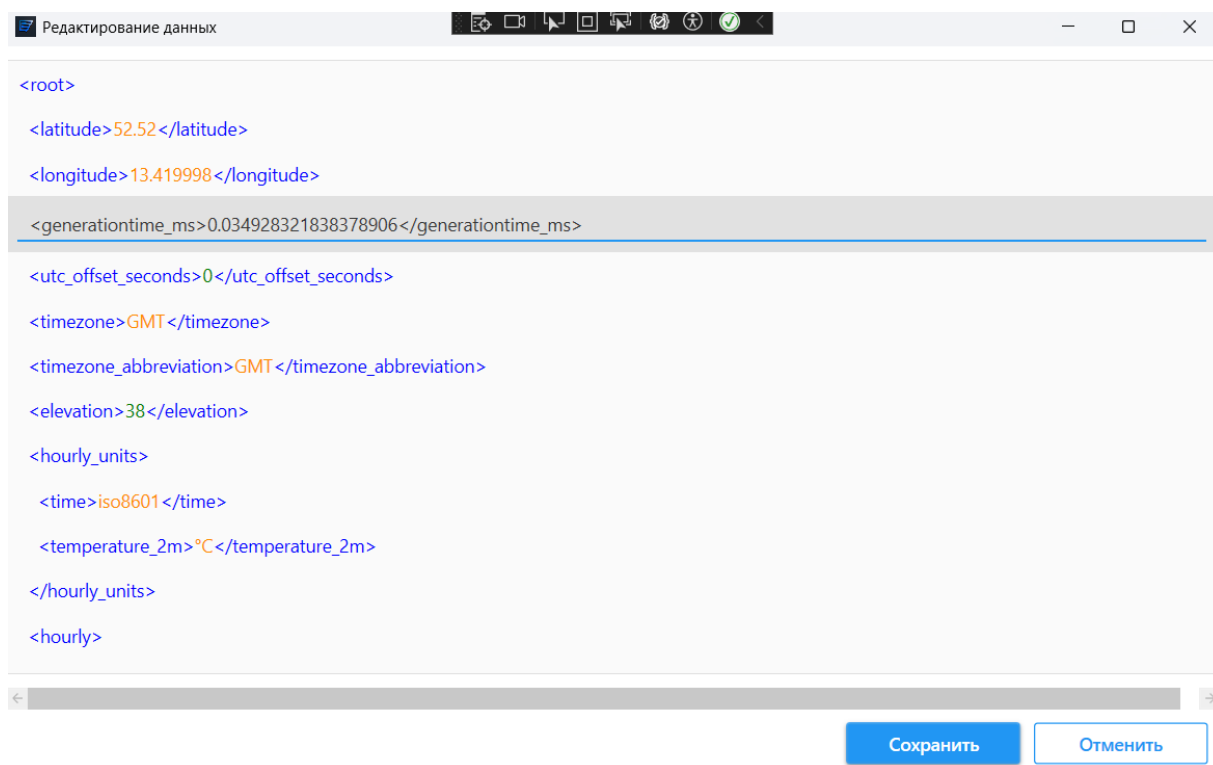


Рисунок 3.2 – Редактирование данных

3.2. Руководство оператора

Функциональным назначением программы является извлечение информации из больших объемов текстовых данных.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Загрузка данных;
- Предобработка данных (Очистка данных: Удаление лишних символов, специальных символов и т.д.);
- Извлечение информации;
- Формирование структурированных данных;
- Сохранение результатов;

Условия выполнения программы

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Минимальный состав технических средств

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:

- процессор с тактовой частотой, 1 ГГц, не менее;
- оперативную память объемом, 512 Мб, не менее;
- жесткий диск со свободным местом 500 Мб, не менее;
- монитор, с разрешением экрана 1024*768, не менее;
- компьютерная мышь;
- клавиатура;
- принтер;

Минимальный состав программных средств.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 10 и выше.

Требования к персоналу (пользователю).

Минимальное количество персонала требуемого для работы программы, должно составлять не менее 1 штатной единицы – пользователь программы, оператор.

Пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы семейства Windows.

Загрузка и запуск программы

Для установки программы необходимо открыть файл «getman.exe» от имени администратора с USB-флэш накопителя (Приложение 3).

В случае успешного запуска программы на рабочем столе будет отображено основное окно программы «GetMan» (Рисунок 3.3).

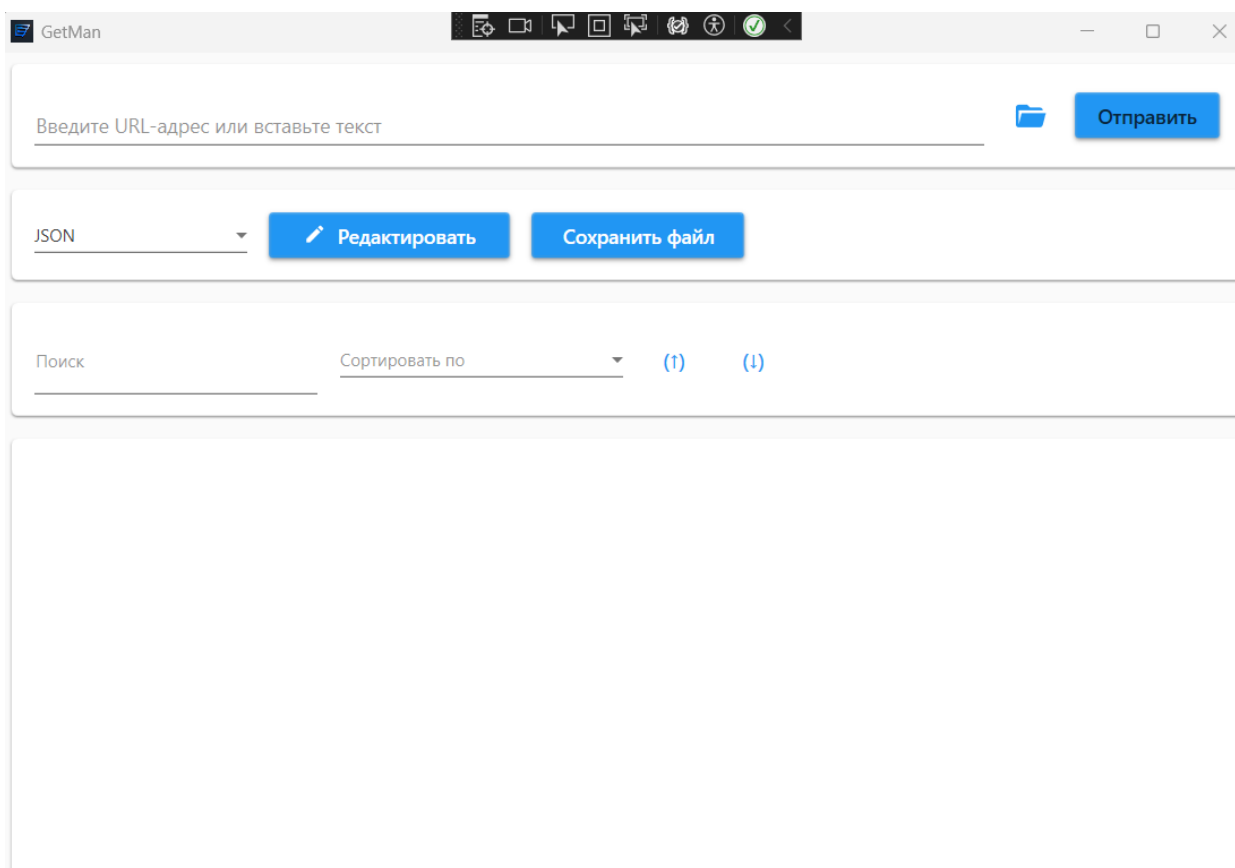


Рисунок 3.3 – Основное окно

В основной странице программы отображается поле для входных данных (URL/Файл/Текст). На этой странице вы открываете файл, отправляете, совершаете поиск, редактируете и сохраняете обработанные данные.

Чтобы открыть файл нужно нажать на иконку с папкой, после этого выбрать файл, который необходимо обработать.

Для того, чтобы отредактировать нужно нажать на кнопку «Редактировать» после чего откроется окно редактирования. (Рисунок 3.5)

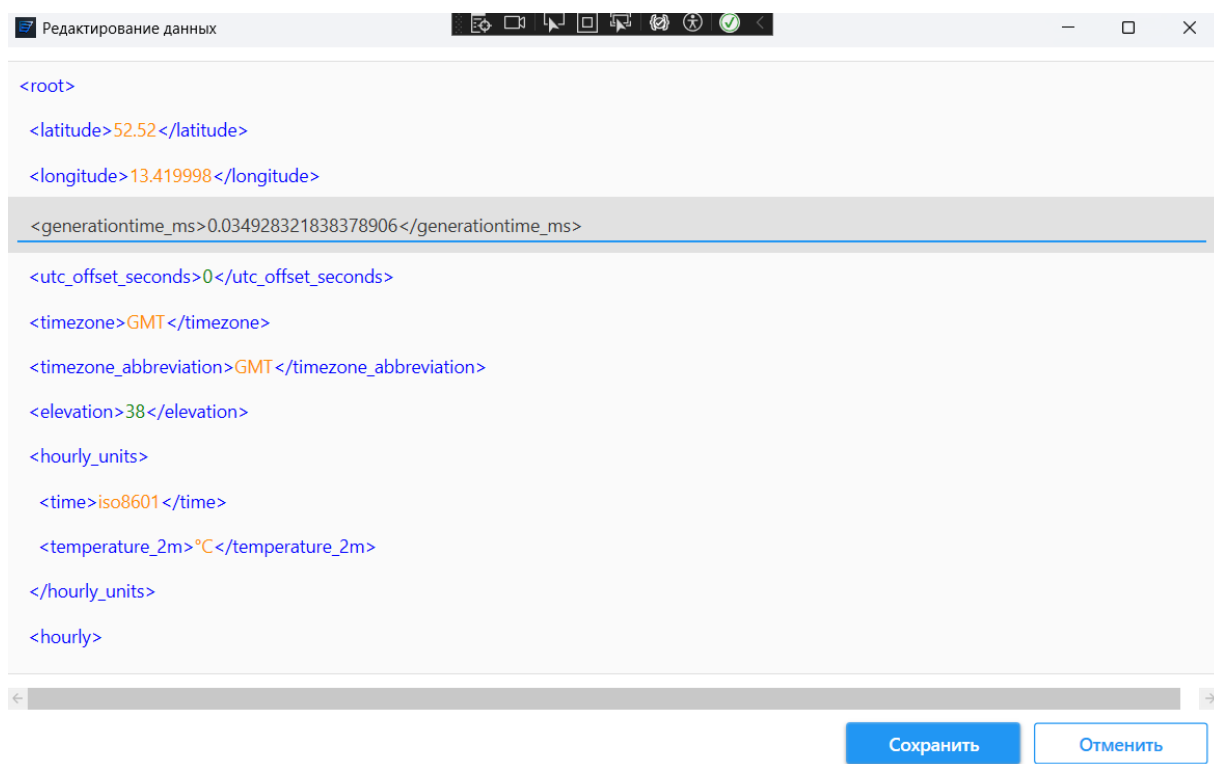


Рисунок 3.5 – Окно редактирования

Для завершения редактирования нужно нажать на кнопку «Сохранить изменения». После чего изменения сохраняются в основном окне.

В основном окне для сохранения полученных результатов с выбранным форматом (JSON/XML/CSV) нужно нажать на кнопку «Сохранить», и выбрать путь, в котором вы хотите сохранить файл. (Рисунок 3.6)

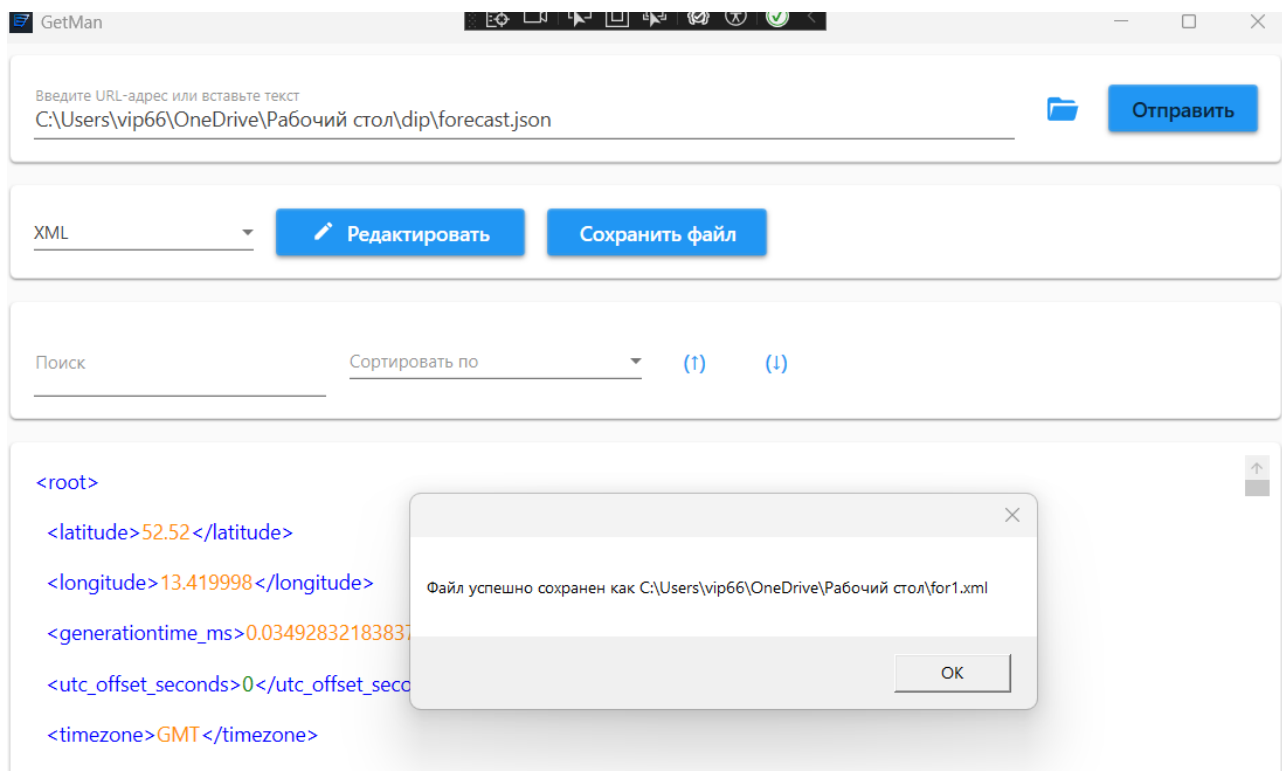


Рисунок 3.6 – Сохранение результатов

ВЫВОДЫ ПО ГЛАВЕ 3

В результате написания третьей главы были проведены тестирования, выявлены и устранены ошибки работы программы. Была описана экспериментально-прикладная часть программной документации. В ходе данной главы была осуществлена опытная эксплуатация.

Проведенное тестирование программы подтвердило ее работоспособность и готовность к эксплуатации. Были проверены все функциональные возможности программы при работе с реальными данными.

Также было выявлено, что запас функциональных возможностей предусматривает использование программы достаточно долго.

На основе созданной программы было написано руководство оператора, которое достаточно понятно описывает алгоритм работы с программой. Руководство оператора разработано на основании ГОСТ 19.505-79[7].

ЗАКЛЮЧЕНИЕ

Основной задачей дипломного проекта было создание программного обеспечения для извлечения данных из текстовых данных.

В ходе написания дипломного проекта был создан программный продукт и техническая документация к нему, описаны алгоритмы и архитектура работы программы.

В ходе работы над аналитической частью были описаны наиболее часто используемые понятия и их поля. В техническом задании описаны функциональные значения, технический состав ЭВМ.

В процессе работы над конструкторской частью была разработана архитектура программы, описан алгоритм работы программы.

Было проведено тестирование программного продукта на наличие ошибок. Во время опытной эксплуатации было разработано руководство пользователя, упрощающее ознакомление пользователя с программой. Программный продукт «GetMan» получил хороший отзыв от куратора.

Разработанный программный продукт имеет визуально приятный интерфейс, в освоении и использовании он достаточно прост.

За время написания дипломного проекта была изучена предметная область, которая помогла правильно составить формы для заполнения документов.

Программный продукт создавался при помощи Microsoft Visual Studio 2022.

При проверке работоспособности программного обеспечения было выявлено, что программа работает верно с различными наборами входных данных.

В результате написания дипломного проекта было разработано программное обеспечение для извлечения информации из текстовых данных.

Все задачи были реализованы, каждая из них выполнялась вовремя и в соответствии с заданием на дипломную работу.

Цели дипломного проекта были успешно достигнуты. Программное обеспечение получило хороший отзыв. Требования заказчика были выполнены в полном объеме.

Благодаря гибкой архитектуре, программа легко поддается модификациям и улучшениям, что открывает возможности для дальнейшего развития и расширения функционала.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.103-77. Единая система программной документации. Обозначение программ и программных документов, введ. 01.07.1978. – М.: Изд-во стандартов, 1978. – 2 с.
2. ГОСТ 19.104-78. Единая система программной документации. Основные надписи, введ. 01.01.1978. – М.: Изд-во стандартов, 1980. – 6 с.
3. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом, введ. 01.01.1980. – М.: Изд-во стандартов, 1989. – 6 с.
4. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению, введ. 01.01.1980. – М.: Изд-во стандартов, 1988. – 3 с.
5. ГОСТ 19.401-78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению, введ. 01.01.1980. – М.: Изд-во стандартов, 1988. – 2 с.
6. ГОСТ 19.505-79. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению, введ. 01.01.1980. – М.: Изд-во стандартов, 1987. – 6 с.

ПРИЛОЖЕНИЯ

Приложение 2

Код приложения

MainWindow.xaml

```
<Window x:Class="Inv.MainWindow"
```

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

```
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
```

```
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

```
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
```

```
xmlns:local="clr-namespace:Inv"
```

```
mc:Ignorable="d"
```

```
Height="550" Width="1100"
```

```
MinHeight="400"
```

```
MinWidth="800"
```

```
x:Name="WindowChrome"
```

```
Title="Учёт спортивного инвентаря | Авторизация"
```

```
Icon="icon.ico"
```

```
Style="{StaticResource VS2012WindowStyle}"
Loaded="Window_Loaded"
Closed="WindowChrome_Closed">
```

```
<Window.Resources>
```

```
<ResourceDictionary>
```

```
<ResourceDictionary.MergedDictionaries>
```

```
<materialDesign:BundledTheme
BaseTheme="Light" PrimaryColor="Blue"
SecondaryColor="LightGreen"
ColorAdjustment="{materialDesign:ColorAdjustment}"/>
```

```
</ResourceDictionary.MergedDictionaries>
```

```
</ResourceDictionary>
```

```
</Window.Resources>
```

```
<Grid>
```

```
<!-- MENU -->
```

```
<Grid x:Name="AllGrid">
```

```
<Grid.ColumnDefinitions>
```

```
<ColumnDefinition MinWidth="180"
MaxWidth="180"/>
```

```
<ColumnDefinition Width="3*"/>
```

```
</Grid.ColumnDefinitions>
```

```
<Grid Grid.Column="0">
```

```
<ScrollViewer Style="{StaticResource
LeftScrollViewer}"/>
```

```
<StackPanel>
```

```
<Grid Height="10">
```

```
<Grid.ColumnDefinitions>
```

```
<ColumnDefinition Width="11*"/>
```

```
<ColumnDefinition Width="0*"/>
```

```
<ColumnDefinition Width="4*"/>
```

```
</Grid.ColumnDefinitions>
```

```
</Grid>
```

```
<Button x:Name="MenuBtn2"
Style="{StaticResource MenuBtn}"
Click="MenuBtn2Click" Content="Учёт"/>
```

```
<Button x:Name="MenuBtn1"
Click="MenuBtn1Click" Style="{StaticResource
MenuBtn}" Content="Инвентарь"></Button>
```

```
<Button x:Name="MenuBtn7"
Click="MenuBtn7Click" Style="{StaticResource
MenuBtn}" Content="Списанное"></Button>
```

```
<Button x:Name="MenuBtn3"
Click="MenuBtn3Click" Style="{StaticResource
MenuBtn}" Content="Экспорт в Word"/>
```

```
<Button x:Name="MenuBtn4"
Click="MenuBtn4Click" Style="{StaticResource
MenuBtn}" Content="Экспорт в Excel"/>
```

Приложение 3

USB-флэш накопитель с материалами проекта

На USB-флэш накопителе располагается:

- Проект приложения (каталог Inv)
- Установщик приложения (файл mysetup.exe)
- Файл дипломного проекта в формате MS Word (Дипломный проект.docx)