

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ Прикладная математика и информатика _____
(наименование)

_____ 09.03.03 Прикладная информатика _____
(код и наименование направления подготовки / специальности)

_____ Разработка программного обеспечения _____
(направление (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения для парсинга данных» _____

Обучающийся _____ Б.И. Бегматов _____
(Инициалы Фамилия) (личная подпись)

Руководитель _____ к.т.н., доцент, О.В. Аникина _____
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Выпускная квалификационная работа на тему: «Разработка программного обеспечения для парсинга данных».

Ключевые слова: автоматизированная информационная система, база данных, программирование, программное обеспечение.

Объём работы - 76 страниц, на которых представлены 15 рисунков, 10 таблиц. При написании работы использовалось 56 источников.

Объектом исследования является ООО «Виват Интеллект».

Предметом исследования являются бизнес-процессы, связанные с обработкой и анализом данных, извлекаемых из различных источников информации.

Цель исследования - разработать автоматизированную информационную систему, позволяющую эффективно парсить, обрабатывать и анализировать данные для повышения качества принимаемых решений в деятельности организации.

Практической ценностью в работе считается то, что предложенное программное обеспечение позволит ускорить процессы получения и обработки информации, сократить количество ошибок, связанных с ручным вводом данных, а также улучшить доступ к актуальным данным для сотрудников компании.

Разработанный программный продукт может использоваться в любой организации, где существует процесс сбора, анализа и обработки данных, включая маркетинговые исследования, финансовый анализ, мониторинг интернет-ресурсов и др.

В первой главе произведен анализ предметной области, в которой рассматриваются характеристика объекта исследования, функциональное моделирование существующих бизнес-процессов «КАК ЕСТЬ», анализ существующих разработок для решения данной задачи, выработка требований информационному, программному и аппаратному обеспечению.

Во второй главе произведено проектирование программного обеспечения, т.е. существующие разработки для решения обозначенной задачи, функциональное моделирование существующих бизнес-процессов «КАК ДОЛЖНО БЫТЬ», описание функциональных требований проекта, а также описание структуры данных.

В третьей главе произведена разработка и реализация проектных решений, включая выбор языка программирования, технологий, архитектуры системы и реализацию функционала.

В четвертой главе произведена оценка и обоснование экономической эффективности проекта, и другие ключевые финансовые показатели. На основании полученных аналитических данных будет обоснована целесообразность и оправданность инвестиций, вложенных в реализацию данного проекта.

Оглавление

Введение.....	5
Глава 1 Анализ предметной области для разработки ПО.....	8
1.1 Характеристика объекта исследования.....	8
1.2 Функциональное моделирование существующих бизнес-процессов «КАК ЕСТЬ»	12
1.3 Анализ существующих разработок для решения данной задачи	15
1.4 Выработка требований информационному, программному и аппаратному обеспечению.....	18
Глава 2 Проектирование программного обеспечения.....	21
2.1 Существующие разработки для решения обозначенной задачи	21
2.2 Функциональное моделирование существующих бизнес-процессов «КАК ДОЛЖНО БЫТЬ»	23
2.3 Описание функциональных требований проекта.....	24
Глава 3 Программная реализация модуля	33
3.1 Язык программирования Python.....	33
3.2 Среда разработки	38
3.3 Выбор Framework/Библиотеки	41
3.4 Веб -сервер	43
3.5 Spider – запросы	46
3.6 Реализация модуля.....	49
Глава 4 Обоснование экономической эффективности проекта	61
Заключение	71
Список используемых источников.....	73

Введение

Когда обрабатывать вручную огромные объёмы информации — долго и сложно, на помощь приходят парсеры. Анализ данных с собственных ресурсов и веб-сайтов конкурентов — способ развивать бизнес и увеличивать с помощью парсинга. Его активно используют как небольшие, так и крупные компании. Однако собирать и обрабатывать вручную огромные объёмы информации — долго и сложно. На помощь бизнесу приходит парсинг, который позволяет получить необходимое количество данных, систематизировать их и затем использовать в работе.

Слово парсинг, или, по-другому, синтаксический анализ, означает сбор и преобразование разрозненных данных по заданным условиям в читаемый формат. Иначе говоря, это инструменты, которые автоматически собирают интересующую вас информацию с веб-страниц. Специальные программы так и называются — парсеры.

Парсинг данных применяют в разных сферах бизнеса. В электронной коммерции можно парсить данные о ценах на товары у конкурентов, в финансовой отрасли — о котировках акций и других финансовых показателях, в медицинской — о заболеваниях и лечении. С помощью парсинга решают различные задачи, и в зависимости от цели выбирают информацию, которую нужно спарсить, в связи с чем тема, выбранная для выпускной работы, является актуальной

Объектом исследования является ООО «Виват Интеллект».

Предметом исследования являются бизнес-процессы, связанные с обработкой и анализом данных, извлекаемых из различных источников информации.

Цель исследования - разработать автоматизированную информационную систему, позволяющую эффективно парсить, обрабатывать и анализировать данные для повышения качества принимаемых решений в деятельности организации.

Задачи исследования:

- выполнить характеристику объекта исследования и функциональное моделирование существующих бизнес-процессов, анализ существующих разработок для решения задач, выработка требований информационному и программному обеспечению;
- рассмотреть существующие разработки для решения данной задачи;
- описать функциональных требований проекта, описание структуры данных;
- выполнить проектирование программного обеспечения системы и его реализации;
- рассчитать экономическую эффективность проекта.

Новизна исследования заключается в том, что полученный опыт позволяет оптимизировать исследуемый процесс, теоретическая база разработки применима практически к любой компании малого и среднего бизнеса.

Практической ценностью в работе считается то, что предложенное программное обеспечение позволит ускорить процессы получения и обработки информации, сократить количество ошибок, связанных с ручным вводом данных, а также улучшить доступ к актуальным данным для сотрудников компании.

Разработанный программный продукт может использоваться в любой организации, где существует процесс сбора, анализа и обработки данных, включая маркетинговые исследования, финансовый анализ, мониторинг интернет-ресурсов и др.

Работа состоит из введения, 4 глав, заключения и списка использованной литературы.

В первой главе произведен анализ предметной области, в которой рассматриваются характеристика объекта исследования, функциональное моделирование существующих бизнес-процессов «КАК ЕСТЬ», анализ

существующих разработок для решения данной задачи, выработка требований информационному, программному и аппаратному обеспечению.

Во второй главе произведено проектирование программного обеспечения, т.е. существующие разработки для решения обозначенной задачи, функциональное моделирование существующих бизнес-процессов «КАК ДОЛЖНО БЫТЬ», описание функциональных требований проекта, а также описание структуры данных.

В третьей главе произведена разработка и реализация проектных решений, включая выбор языка программирования, технологий, архитектуры системы и реализацию функционала.

В четвертой главе произведена оценка и обоснование экономической эффективности проекта, и другие ключевые финансовые показатели. На основании полученных аналитических данных будет обоснована целесообразность и оправданность инвестиций, вложенных в реализацию данного проекта.

В заключение даны выводы о проделанной работе.

Глава 1 Анализ предметной области для разработки ПО

1.1 Характеристика объекта исследования

ООО «Виват Интеллект» было основано в 2008 году и находится по адресу: Самарская область, город Тольятти, улица Маршала Жукова, дом 18.

Компания представляет собой объединение профессионалов, стремящихся к совершенствованию в области разработки и поддержки решений на платформе 1С. Задача ООО «Виват Интеллект» — обеспечить идеальные условия как для специалистов, так и для клиентов, чтобы совместно достигать высоких результатов и наслаждаться процессом решения каждой задачи, для достижения поставленной цели. Основная миссия ООО «Виват Интеллект» заключается в предоставлении качественных услуг по автоматизации управления и учета на предприятиях, а также культурном сопровождении программного обеспечения 1С.

В ООО «Виват Интеллект» работают высококвалифицированные специалисты с основательным опытом в сфере автоматизации.

Успешные сотрудники получают отличные условия труда, доступ к самым современным технологиям и инструментам, возможность участия в обучающих программах и стажировках, работая в современном офисе с дружелюбной атмосферой. ООО «Виват Интеллект» предлагает компенсации на топливо и корпоративную сотовую связь, а также гибкие условия работы и поддержку лояльным работникам, подход к каждому члену команды.

На протяжении 10 лет компания ООО «Виват Интеллект» успешно реализует сложные IT-проекты на платформе 1С: Предприятие для разнообразных российских компаний. ООО «Виват Интеллект» освоил и применяет лучшие решения 1С, включая:

1С: Управление производственным предприятием,

1С: Управление торговлей,

1С: Комплексная автоматизация,

1С: ERP Управление предприятием 2,

1С: Управление холдингом.

Развертывание и эксплуатация информационной инфраструктуры предприятия нередко требуют профессиональной помощи со стороны сторонних IT-компаний. ООО «Виват интеллект» предлагает корпоративным клиентам ряд услуг и сервисов, решающих весь спектр задач, связанных с использованием различных продуктов 1С. Таким образом, основными услугами предприятия являются:

- внедрение 1С: ERP Разработка любой тематики: производство, торговля, общепит или сфера услуг. Анализирование потребности, сбор данных о компании и предложение услуг двух видов: внедрение системы бизнес-анализа для принятия управленческих решений и ведем проекты по оптимизации и реинжинирингу бизнес-процессов. Помогаем бизнесу увеличить доход и сократить издержки до 30%;
- реальная автоматизация. Автоматизация бизнеса и обучение;
- удаленная работа в 1С. В современном мире многие компании как малого и среднего, так и крупного бизнеса ведут деятельность в новом формате «удаленно». ООО «Виват интеллект» вместе с фирмой 1С, может предложить несколько вариантов работы в 1С облаке.
- внедрение 1С. ООО «Виват интеллект» специализируется на внедрении и настройке программного обеспечения на базе программы 1С. Штат состоит из высококвалифицированных специалистов, каждый из которых сертифицирован в 1С. ООО «Виват интеллект» реализовал более 100 крупных проектов;
- сопровождение 1С. Услуги по сопровождению 1С предоставляются только по договору информационно-технологического сопровождения (договор ИТС) на регулярной основе. Объем сервисов, услуг и стоимость договора определяются тарифным планом, либо видами и объемами работ, необходимыми клиенту, на основании тарифов;

- маркировка товаров. ООО «Виват интеллект» подключает, автоматизирует, настраивает и сопровождает бизнес-процессы, связанные с маркировкой;
- восстановление порядка в бухучете. ООО «Виват интеллект» устраняет проблемы в ведении учета в Программе 1С, восстанавливает как частично, так и комплексно всю базу клиента. Параллельно настраивает системы программного продукта для эффективного учета и налогообложения;
- бюджетирование. Бюджетирование в системах 1С помогает прогнозировать расходы и составлять бюджет на разные периоды времени, а значит – способствует развитию предприятия и соблюдению принятых управленческих решений;
- консультации по работе в 1С. 1С 8.3 – огромный программный комплекс, в который входят десятки инструментов для управления предприятием или ведения бизнеса. Не всегда в штате компании есть специалисты, которые могут полноценно заниматься 1С-разработкой, а программное обеспечение нуждается в настройке и обслуживании;
- внедрение 1С УНФ. Лучшее решение для среднего бизнеса до 30 пользователей 1С, для компаний, занимающихся производством и реализацией собственной продукции, которым необходима учетная система, где все процессы будут слаженными, документы оформлены в едином стандарте, финансовая отчетность такая, которую хочет видеть руководитель;
- аренда 1С. В этот сложный период многие компании малого и среднего бизнесы вынуждены вести деятельность в новом формате удаленно. ООО «Виват интеллект» предлагает несколько вариантов удобной организации удаленной работы с программами 1С.

Запущены проекты Бюджетирование и Казначейство, МСФО БИТ.ФИНАНС.

Организационная структура предприятия представлена на рисунке 1.

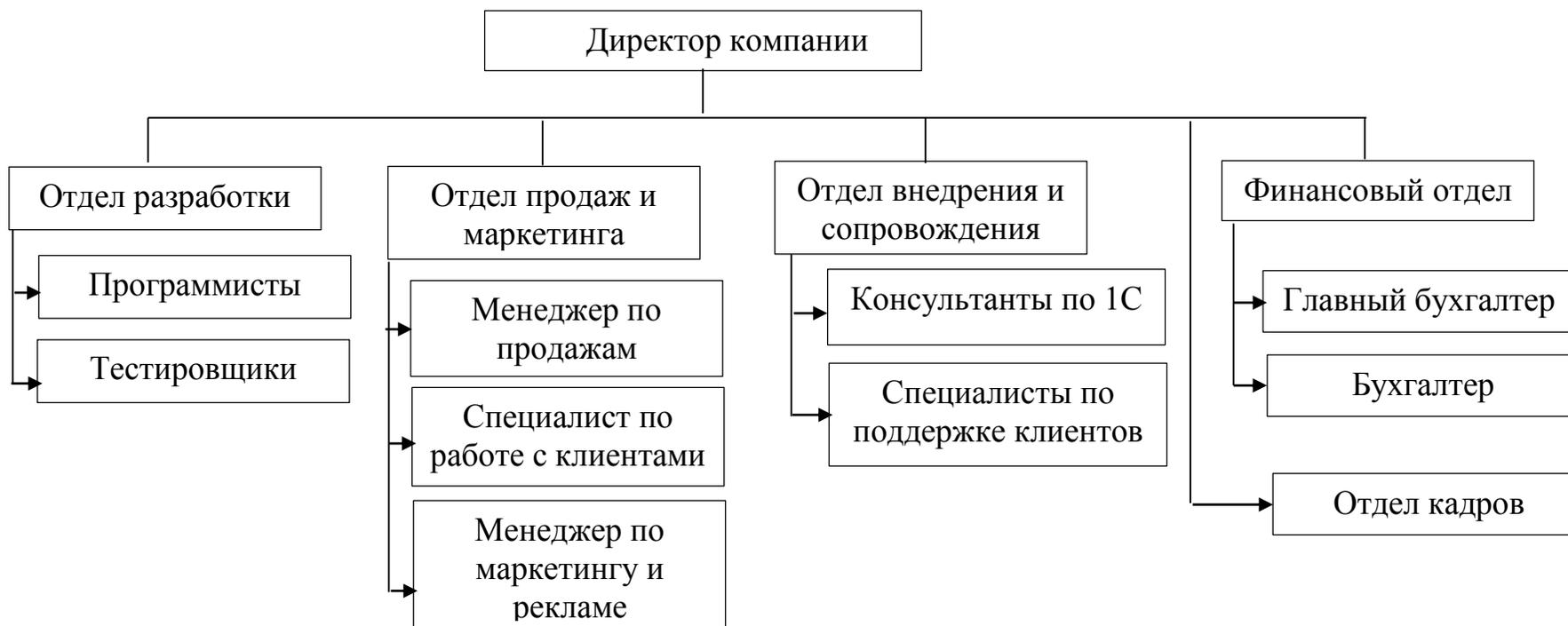


Рисунок 1 – Организационная структура ООО «Виват интеллект»

Такая структура позволяет ООО «Виват интеллект» эффективно работать и обеспечивать высокое качество услуг для клиентов.

Компетенции сотрудников:

- специалисты имеют сертификаты 1С, подтверждающие их уровень знаний;
- ООО «Виват Интеллект» сопровождает крупные компании (от 500 пользователей, 200 юридических лиц);
- опыт внедрения МСФО (международной системы финансовой отчетности).

Высококвалифицированные программисты готовы решить любые задачи:

- более 62% клиентов ООО «Виват Интеллект» рекомендовали компанию своим друзьям и знакомым, партнерам и клиентам;
- настроен электронный документооборот с 65% клиентов;
- каждый новый сервис ООО «Виват Интеллект» тестирует на себе, прежде чем предложить его своим клиентам;
- 300 организаций, более 1000 рабочих мест автоматизированы компанией ООО «Виват Интеллект».

1.2 Функциональное моделирование существующих бизнес-процессов «КАК ЕСТЬ»

Создание функциональной и процессной модели для внедрения парсинга в компании «Виват Интеллект» — это важный шаг к оптимизации рабочих процессов и улучшению автоматизации. Можем использовать нотацию BPMN (Business Process Model and Notation), которая является широко используемой в бизнес-анализе и позволяет визуально представлять процессы.

Модель «Как есть»

Процессный блок-схема «Как есть» (Visio).

Начало процесса: курсор приходит к обработке данных (файл, веб-страница).

Сбор данных: специалист вручную собирает данные из различных источников (документы, веб-страницы, базы данных).

Обработка данных: специалист обрабатывает собранные данные, убирая лишнюю информацию.

Передача данных: обработанные данные передаются в следующий этап (например, для анализа или хранения).

Конечный этап: полученные данные анализируются и формируются отчеты.

Процессная схема «Как есть» (рисунок 2).

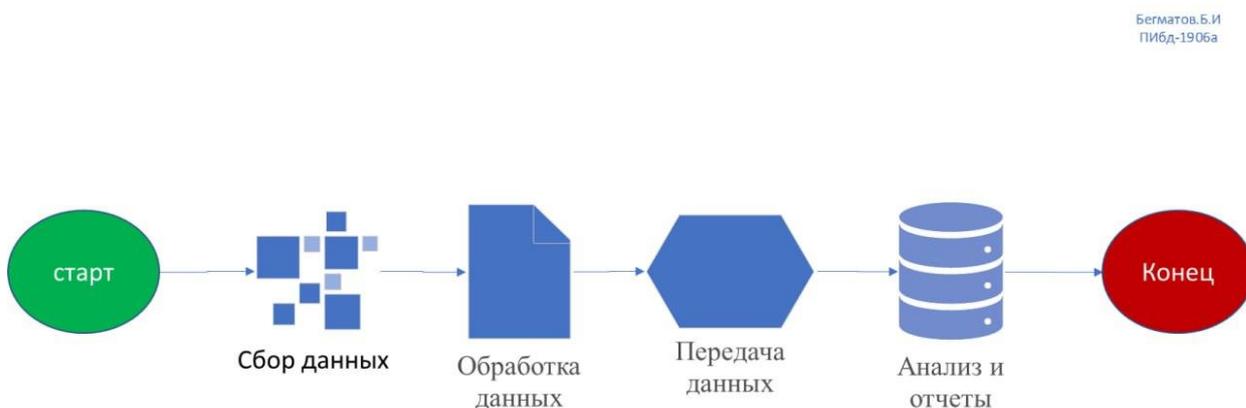


Рисунок 2 - Процессная схема «Как есть»

Недостатки процессной схемы «Как есть» в компании ООО «Виват Интеллект» при сборе и обработке данных можно выделить следующие ключевые моменты:

Время и трудозатраты: ручное выполнение всех этапов, включая сбор, обработку и анализ данных, требует значительного количества времени и усилий со стороны сотрудников. Это может привести к задержкам в получении актуальной информации.

Человеческий фактор: участие человека в процессе сбора и обработки данных увеличивает вероятность ошибок, таких как опечатки, неверная

интерпретация информации или пропуск важных данных.

Низкая эффективность: приходится разрабатывать и поддерживать рутинные процедуры, которые не добавляют ценности бизнес-процессам. Это может привести к снижению общей производительности компании.

Ограниченная масштабируемость: с увеличением объема информации и количества источников данных, тяжело адаптировать существующие процессы. Расширение объемов может потребовать дополнительных ресурсов и времени.

Зависимость от квалификации сотрудников: качество итоговых данных зависит от уровня квалификации и опыта специалистов. Если работники уходят или меняются, знание и навыки парсинга могут быть потеряны.

Информационная безопасность: ручной сбор и обработка данных могут повышать риски утечек информации или неправомерного доступа к конфиденциальным данным.

Отсутствие стандартизации: ручные процессы зачастую не стандартизированы, что может приводить к несоответствиям в качестве и формате собранных данных, затрудняющих их последующий анализ.

Невозможность оперативной обработки данных: в условиях быстроменяющегося рынка, ручная обработка данных может не успевать за событиями, что приводит к упущенным возможностям и снижению конкурентоспособности.

Недостаточная адаптация к изменениям: отсутствие гибкой системы парсинга не позволяет быстро адаптироваться к новым источникам данных или изменению требований рынка.

Таким образом, процессная схема «Как есть» демонстрирует явные недостатки, которые могут быть решены с помощью перехода на автоматизированную модель, что обеспечит лучшую эффективность, скорость и сокращение ошибок в обработке данных.

1.3 Анализ существующих разработок для решения данной задачи

Реализация парсинга может значительно упростить и автоматизировать рабочие процессы. Ниже приведем несколько примеров и разработок, которые иллюстрируют использование парсинга в 1С.

Автоматизация сбора цен и остатков товаров со складов конкурентов.

Ситуация: Розничные сети использовали парсинг для мониторинга цен на товары, предлагаемые конкурентами, а также остатков на складах.

Решение: С помощью HTTP-запросов и библиотек для парсинга HTML-страниц, разработали обработку, которая совершала периодические запросы к сайтам конкурентов, извлекала информацию о ценах и остатках. Полученные данные автоматически загружались в 1С, что позволяло использовать их для анализа и оптимизации своей ценовой политики [2, с. 110].

Импорт данных из внешних систем и баз данных

Ситуация: Один из производств нуждался в регулярном обновлении данных о поставках и характеристиках товаров из внешних систем, которые не имели интеграции с 1С.

Решение: Программа парсинга была разработана для извлечения данных из API и веб-интерфейсов этих систем. Работая на платформе 1С, она автоматически собирала информацию о поставках и импортировала её в базу данных предприятия в назначенное время.

Анализ социальных сетей и новостных сайтов

Ситуация: для компаний, работающих в сфере маркетинга, важно было следить за общественным мнением о своих продуктах.

Решение: Разработанная система парсинга собирала данные из социальных сетей и новостных ресурсов, определяла упоминания бренда, анализировала настроение постов и комментариев. Эти данные затем использовались в 1С для формирования отчетов и анализа эффективности рекламных кампаний [6, с. 90].

Мониторинг наличия документов на сайте клиентов

Ситуация: Поставщики нужны в контроле подписанных документов и спецификаций у своих клиентов.

Решение: Создана обработка на 1С, осуществляющая парсинг веб-страниц клиентов для поиска и проверки наличия документов, необходимых для дальнейшей работы. Информация выводится на панель управления и используется для управления отношениями с клиентами.

Импорт курсов валют и финансовой информации

Ситуация: для бухгалтерского учета необходимо регулярно получать актуальные курсы валют.

Решение: Парсинг данных с веб-сайтов, предоставляющих финансовую информацию (например, сайты банков), возможность получать и обновлять курсы валют в 1С. Это обеспечивает автоматическое обновление данных без ручного ввода [9, с. 42].

Эти примеры показывают, как парсинг может улучшить бизнес-процессы и повысить их эффективность в компаниях, работающих с программой 1С. Реализация парсинга требует технических навыков программирования на платформе 1С, но результаты могут значительно облегчить работу сотрудников и улучшить качество информации, используемой в управлении.

Рассмотрим в деталях, как парсинг может быть интегрирован в 1С, когда он может быть необходим, а также какие преимущества он может дать.

Возможности парсинга в 1С

Парсинг — это процесс извлечения данных из различных источников, таких как веб-страницы, документы, электронные таблицы и многое другое. В 1С можно реализовать парсинг следующими способами [9, с. 134]:

Использование встроенных инструментов для работы с Web-запросами: 1С позволяет делать HTTP-запросы для получения информации из открытых API или веб-сайтов.

Использование сторонних библиотек или платформ для работы с HTML и XML, например, Python или специализированные инструменты парсинга,

которые могут взаимодействовать с 1С.

Написание собственных обработчиков парсинга на языке 1С:
Предприятие.

Парсинг может понадобиться в следующих ситуациях:

Автоматизация сбора данных: при необходимости регулярно собирать данные из определенных источников (например, цены конкурентов, обновления в системах учета и т.д.).

Анализ информации: для получения статистики, маркетинговой информации или другой аналитики из внешних источников.

Импорт данных: для загрузки данных из внешних источников в систему 1С, включая внешние базы данных, веб-службы или интернет-ресурсы.

Преимущества парсинга в 1С:

- увеличение эффективности процессов. Сокращение времени на сбор данных: Автоматизация процесса парсинга позволяет существенно уменьшить время на получение информации, по сравнению с ручными методами;
- минимизация ошибок. Устранение человеческого фактора при вводе данных уменьшает риск ошибочного ввода информации;
- актуальность информации. Регулярное обновление данных: Парсинг позволяет легко обновлять данные в 1С на постоянной основе, что особенно важно для бизнес-анализа и оперативного реагирования на изменения;
- принятие обоснованных решений. Доступ к актуальной информации: благодаря парсингу руководители и аналитики могут принимать более обоснованные решения, опираясь на актуальные рыночные данные и аналитику;
- конкурентные преимущества. Мониторинг рынка: Парсинг может использоваться для отслеживания цен и предложений конкурентов, что позволяет более эффективно управлять ценовой политикой.

Использование парсинга в программе 1С может существенно повысить

эффективность учета и управления бизнес-процессами. Он позволяет автоматизировать получение данных, обеспечивать их актуальность и качество. Реализовав парсинг, компания сможет принимать более обоснованные решения и укрепить свою позицию на рынке.

1.4 Выработка требований информационному, программному и аппаратному обеспечению

Для успешной реализации системы парсинга данных, интегрируемой с 1С, необходимо разработать четкие требования к информационному, программному и аппаратному обеспечению.

Требования к информационному обеспечению

Определение источников данных: необходимо четко определить, из каких источников будет производиться парсинг (веб-страницы, API, файлы и т. д.).

Сигналы и триггеры для парсинга: определить, по какому принципу будет инициироваться процесс парсинга (по тегам и по запросу, где будет установлен тайминг или в ответ на определенные события).

Проверка и валидация данных: установить требования к валидации собранных данных, чтобы гарантировать их корректность и целостность.

Форматы данных: определить форматы, в которых данные будут поступать и сохраняться, чтобы обеспечить совместимость с 1С (например, JSON, XML, CSV и т. д.).

Уровни доступа: определить уровни доступа к системе парсинга для разных категорий пользователей, чтобы обеспечить безопасность информации.

Требования к программному обеспечению

Базовая функциональность: система парсинга должна обеспечивать возможность автоматизированного сбора, обработки и хранения данных с установленной частотой.

Интерфейс интеграции с 1С: необходимо создать API или использовать существующий сервисный интерфейс для интеграции с 1С, чтобы передавать обработанные данные напрямую в систему.

Обработка ошибок: оборудование должно иметь функции логирования и уведомления о сбоях в процессе парсинга.

Наличие регулярных обновлений: программное обеспечение должно иметь возможность обновления для исправления ошибок и добавления новых функций.

Интерфейс пользователя: необходимо разработать удобный интерфейс для настройки и управления процессами парсинга, а также для мониторинга их работы.

Поддержка различных источников данных: программа должна быть способна работать с различными форматами данных и источниками информации.

Требования к аппаратному обеспечению

Производительность: сервер должен обладать достаточной вычислительной мощностью (ЦП, ОЗУ) для обработки больших объемов данных и обеспечения параллельного парсинга.

Хранилище данных: необходимо предусмотреть достаточное количество дискового пространства для хранения собранных и обработанных данных, а также обеспечивать возможность архивации.

Сетевое оборудование: серверы и рабочие станции должны быть подключены к стабильной и надежной сети для обеспечения быстрого доступа к необходимым ресурсам и источникам данных.

Резервирование и отказоустойчивость: необходимость создания системы резервирования для предотвращения потери данных и минимизации времени простоя системы.

Выработка требований к информационному, программному и аппаратному обеспечению для парсинга, интегрированного с 1С, позволит создать эффективную и надежную систему, которая улучшит процессы

обработки данных и обеспечит более качественный анализ информации. Подробное определение требований на этапе проектирования поможет минимизировать возможные проблемы в дальнейшем и обеспечить соответствие системы бизнес-потребностям компании.

Вывод по первой главе

В первой главе дана характеристика ООО «Виват Интеллект», которая специализируется на предоставлении услуг по автоматизации услуг управления учета на базе программного обеспечения 1С. Основные услуги включают внедрение ERP-систем, автоматизацию бизнес-процессов, удаленную работу в 1С, восстановление учета, бюджетирование и консультирование. Рассматривая модель «Как есть» ООО «Виват Интеллект» мы выявили, что данные собираются и обрабатываются вручную, т.е. специалист вручную собирает информацию из различных источников, сортирует и фильтрует данные вручную, а также из сформированных и обработанных данных, составляют отчет

Недостатками текущей модели были: время и трудозатраты, т.е. ручные процессы занимали много времени и усилий, высокая вероятность ошибок, низкая эффективность, трудно адаптировать процессы при увеличении объема данных, качество данных зависит от уровня квалификации специалистов, риск утечки информации, невозможность оперативной обработки данных. Переход на автоматизированную модель позволило бы улучшить эффективность, скорость обработки данных и сократить количество ошибок. Реализация парсинга в компании «Виват Интеллект» значительно упрощает и автоматизирует рабочие процессы.

Были выработаны требования к информационному, программному и аппаратному обеспечению. Была представлена программа, которая будет работать с веб-сайтами, API (как сервис) и с файлами, также будет предоставлена защита данных и соблюдение стандартов безопасности. Так как данная программа является веб-сервисом – она работает с различными программами.

Глава 2 Проектирование программного обеспечения

2.1 Существующие разработки для решения обозначенной задачи

Для создания парсинга данных, который будет интегрирован в систему 1С, существует несколько подходов и решений. Важно рассмотреть как уже существующие разработки, так и возможности для их адаптации к специфическим требованиям бизнеса.

Некоторые существующие разработки для парсинга данных как веб-сервиса можем рассмотреть в таблице 1.

Таблица 1 - Существующие разработки для парсинга данных как веб-сервиса

Инструмент	Описание
Web Scraper.io	Расширение для Chrome, позволяющее визуально программировать парсинг сайтов, имеет большой набор опций.
Data miner	Расширение для Chrome и Edge, упрощающее сбор данных с сайтов с помощью визуального интерфейса.
SiteAnalyzer	Программа для проверки технических и SEO-данных сайтов, работает на Windows.
ParserOk	Парсинг сайтов с помощью VBA (макросов) в Excel, позволяет импортировать данные по заранее созданным шаблонам.
Octoparse	Программа для ПК, автоматизирующая парсинг, удобна для пользователей без опыта в программировании.
iDatica	Сервис для создания кастомных парсеров на основе запроса, подходит для пользователей без навыков программирования.
A-Parser	Многофункциональный инструмент для профессионального использования, парсит данные в неограниченном объёме.

Эти инструменты предлагают различные уровни функциональности и удобства, подходящие для разных требований и навыков пользователей, от простых визуальных интерфейсов до мощных профессиональных решений.

Важно выбрать инструмент, соответствующий специфическим требованиям бизнеса, а также уровня подготовки сотрудников. Web Scraper.io и Data miner отлично подходят для пользователей, которым нужен простой и визуально ориентированный инструмент, тогда как A-Parser и SiteAnalyzer предоставляют более продвинутое возможности для профессионалов. Подробный анализ и сравнение этих инструментов могли бы дать наиболее эффективное решение для бизнеса. Таким образом данные решения помогают сократить время и усилия, необходимые для сбора данных, обеспечивая при этом высокую точность и актуальность информации.

Также стоит обратить внимание на обработку ошибок и исключений, так как веб-страницы могут измениться, что повлияет на работу парсинга.

Но также рассмотрим рекомендации по созданию собственного парсинга:

Собственный модуль: если текущие решения не отвечают всем требованиям, возможно, стоит рассмотреть разработку собственного модуля парсинга, адаптированного под ваши конкретные нужды.

Оптимизация: необходимо исследовать требования к обработке данных и построить оптимизированные алгоритмы и структуры для хранения.

Тестирование: Одной из важнейших частей будет тестирование на обработку ошибок и изменение структуры веб-страниц.

Существующие разработки для создания парсинга в 1С предлагают множество возможностей. Компании стоит тщательно анализировать доступные решения, учитывая свои потребности, технические возможности и уровень интеграции с уже существующими системами. Важно выбирать надежные, проверенные решения или, при необходимости, разрабатывать собственные адаптированные варианты, чтобы обеспечить эффективный процесс парсинга данных.

Также необходимо учитывать затраты на разработку и поддержку собственного модуля. Регулярное обновление и адаптация системы к изменениям на целевых веб-страницах обеспечат ее долгосрочную

эффективность и устойчивость.

2.2 Функциональное моделирование существующих бизнес-процессов «КАК ДОЛЖНО БЫТЬ»

Рассмотрим внедрение системы парсинга, которая автоматизирует сбор и обработку данных.

Процессный блок-схема «Как должно быть» (Visio).

Начало процесса: активация рабочего процесса парсинга (можно по расписанию или по запросу).

Парсинг данных: автоматизированная система парсинга считывает данные из необходимых источников.

Форматирование и очистка данных: автоматизированная система очищает и форматирует данные.

Сохранение данных: Обработанные данные автоматически сохраняются в базу данных.

Анализ данных: аналитическая система использует новые данные для генерации отчетов и бизнес-анализа.

Обратная связь: специалисты получают уведомления о новых данных и отчетах. Процессная схема «Как должно быть» представлена на рисунке 3.

Бегматов.Б.И
ПИБд-1906а



Рисунок 3 - Процессная схема «Как должно быть»

Преимущества новой модели:

- скорость и эффективность: автоматизация процессов парсинга значительно уменьшит время, необходимое для сбора и обработки данных;
- снижение ошибок: устранение ручного ввода данных, что снижает вероятность ошибок;
- увеличение производительности: специалисты могут сосредоточиться на анализе данных и принятии бизнес-решений, вместо рутинной работы по сбору информации;
- гибкость и адаптивность: возможность легко добавлять новые источники данных и менять правила парсинга.

Для успешного внедрения системы парсинга в ООО «Виват Интеллект» компания должна будет уделить внимание выбору технологии парсинга, обучению сотрудников и созданию инфраструктуры для обработки данных. Проведение регулярных мониторингов и доработок системы также требуется для обеспечения максимальной эффективности и повышения качества получаемых данных.

2.3 Описание функциональных требований проекта

Для успешного внедрения системы парсинга данных в компанию «Виват Интеллект» необходимо четко определить бизнес-цели и требования к IT-проекту.

Это позволит составить календарный план, в котором будут указаны основные этапы реализации проекта, ключевые задачи и сроки выполнения. Ниже приведены примеры бизнес-целей и требований к проекту.

Основные цели проекта разработки ПО представлены в таблице 2.

Таблица 2- Основные цели и задачи проекта разработки ПО

№	Цель проекта	Задачи
1	Автоматизация процессов сбора данных	<ul style="list-style-type: none"> – сократить время на ручной сбор данных из различных источников (веб-сайты, API, документы); – уменьшить вероятность ошибок, связанных с ручным вводом данных.
2	Повышение качества собранных данных	<ul style="list-style-type: none"> – обеспечить высокую степень актуальности и полноты собранной информации; – устранить дублирование данных и обеспечить единое хранение и форматирование.
3	Увеличение производительности сотрудников	<ul style="list-style-type: none"> – освободить сотрудников от рутинной работы, позволяя им сосредоточиться на анализе данных и принятии бизнес-решений; – обучить персонал работе с новыми инструментами и данными.
4	Поддержка принятия бизнес-решений	<ul style="list-style-type: none"> – обеспечить доступ к актуальным данным для менеджмента и аналитиков; – ускорить процессы отчетности и анализа.
5	Снижение издержек	<ul style="list-style-type: none"> – снизить операционные затраты на сбор и обработку данных путем автоматизации процессов; – оптимизировать расход ресурсов, необходимых для высококачественного анализа данных.

Рассмотрим подробнее бизнес-цели IT-проекта.

Автоматизация процессов сбора данных:

- сократить время на ручной сбор данных из различных источников (веб-сайты, API, документы). Уменьшить механический труд и превратить сбор информации на автоматический;
- уменьшить вероятность ошибок, связанных с ручным вводом данных. Уменьшить тем самым человеческий фактор влияния точность при сборе информации.

Повышение качества собранных данных:

- обеспечить высокую степень актуальности и полноты собранной информации. Не всегда человек может обеспечить возможность

обзора информации по всему интернет, что актуализирует автоматический сбор информации при парсинге и его полноту;

- устранить дублирование данных и обеспечить единое хранение и форматирование. Также преимущество автоматизированного сбора информации в том, что аннулируются продублированные источники и сама информация, что делается намного качественнее при парсинге данных нежели при влиянии фактора человека.

Увеличение производительности сотрудников:

- освободить сотрудников от рутинной работы, позволяя им сосредоточиться на анализе данных и принятии бизнес-решений. Естественно, бизнес решения будут более правильными и человеческий фактор может повлиять уже на возможности пересмотра тех или иных ошибок при анализе отобранной информации;
- обучить персонал работе с новыми инструментами и данными. Это даст возможность сократить рутинную работу обработки информации и внести свободный промежуток времени на обработку более важных решений.

Поддержка принятия бизнес-решений:

- обеспечить доступ к актуальным данным для менеджмента и аналитиков. Это не только дает возможность воспользоваться информацией большинству сотрудникам при их параллельном доступе к информации, но и решать собственные задачи вне зависимости от передачи информации ответственного сотрудника;
- ускорить процессы отчетности и анализа. Также предусматривает составление отчетности каждого сотрудника в зависимости от его поставленных задач.

Снижение издержек:

- снизить операционные затраты на сбор и обработку данных путем

автоматизации процессов;

- оптимизировать расход ресурсов, необходимых для высококачественного анализа данных.

Требования к IT-проекту.

Функциональные требования

- парсинг данных: задача по сбору данных с различных веб-сайтов. Это ключевая функция, позволяющая автоматизировать процесс получения нужной информации из веб-источников;
- интеграция с API 1С: взаимодействие с системой 1С через API. Это необходимо для автоматического обмена данными между парсером и системой 1С, что повышает эффективность работы;
- поддержка форматов данных: необходимость обработки данных в форматах JSON и XML. Это гарантирует, что собранные данные будут легко интегрироваться с другими системами и платформами;
- безопасность данных: требования по обеспечению защиты данных. Это включает в себя меры по предотвращению несанкционированного доступа и защите данных от потерь и утечек;
- интуитивный интерфейс: простота использования интерфейса для пользователей. Это важно для обеспечения удобства работы с системой;
- документация и инструкции: предоставление подробной документации и инструкций для пользователей, что облегчает внедрение и использование системы.

Нефункциональные требования:

- надежность: включает в себя обработку ошибок и восстановление после сбоев. Это обеспечивает стабильную работу системы и минимизирует возможные простои;

- предотвращение потерь данных: меры по защите данных от потерь. Это важно для обеспечения целостности и сохранности данных;
- регулярные обновления: поддержка и обновление программного обеспечения. Это необходимо для поддержания актуальности системы и исправления возможных ошибок;
- производительность: оптимизация времени ответа и скорости обработки данных. Это позволяет системе работать быстро и эффективно;
- асинхронные запросы и кэширование: использование асинхронных запросов для улучшения производительности. Это помогает обрабатывать большие объемы данных без значительных задержек;
- легкость обслуживания: обеспечение легкости обслуживания и обновления сервиса. Это включает в себя удобство внесения изменений и дополнений в систему;
- мониторинг и логирование: механизмы для отслеживания производительности и выявления проблем. Это позволяет своевременно обнаруживать и устранять неполадки в работе системы;
- масштабируемость: возможность расширения системы для обработки большего объема данных. Это важно для адаптации системы к растущим потребностям бизнеса;
- совместимость: совместимость с другими системами и текущими конфигурациями ИС. Это обеспечивает интеграцию с уже существующими решениями в компании;
- соответствие стандартам безопасности: гарантирует, что система будет соответствовать установленным нормативам и защищать данные пользователей.

Эти требования обеспечивают полную и эффективную функциональность системы парсинга данных, а также соответствие всем необходимым стандартам надежности, производительности и безопасности.

Методология FURPS+ используется для классификации и анализа требований к системе. Вот как можно представить требования в таблице 3.

Таблица 3 - Методология FURPS+

Категория	Требования
Functionality (функциональность)	Парсинг данных с различных веб-сайтов Интеграция с API 1С Поддержка различных форматов данных (JSON, XML) Обеспечение безопасности данных
Usability (применимость)	Интуитивно понятный интерфейс Поддержка различных браузеров и устройств Наличие документации и инструкций для пользователей
Reliability (надежность)	Обработка ошибок и восстановление после сбоев Предотвращение потери данных Регулярные обновления и поддержка программного обеспечения
Performance (производительность)	Оптимизация времени ответа и скорости обработки данных Использование асинхронных запросов и кэширования
Supportability (пригодность к эксплуатации)	Легкое обслуживание и обновление сервиса Механизмы мониторинга и логирования
Дополнительные требования	Масштабируемость Совместимость с другими системами Соответствие стандартам безопасности и конфиденциальности данных

Эта таблица поможет структурировать требования к системе парсинга данных, обеспечив ее функциональность, удобство использования, надежность, производительность, поддержку и соответствие дополнительным

требованиям, согласно методологии FURPS+.

Далее рассмотрим проектирование программного обеспечения на рисунке 4.

Интеграция системы 1С с помощью API позволяет обмениваться данными в режиме реального времени, обеспечивая мгновенное получение и отправку информации. Интерфейс, имеющий два режима — стандартный и пользовательский, — подключен к парсинг программе, что позволяет пользователям выбирать между простым и продвинутым функционалом.

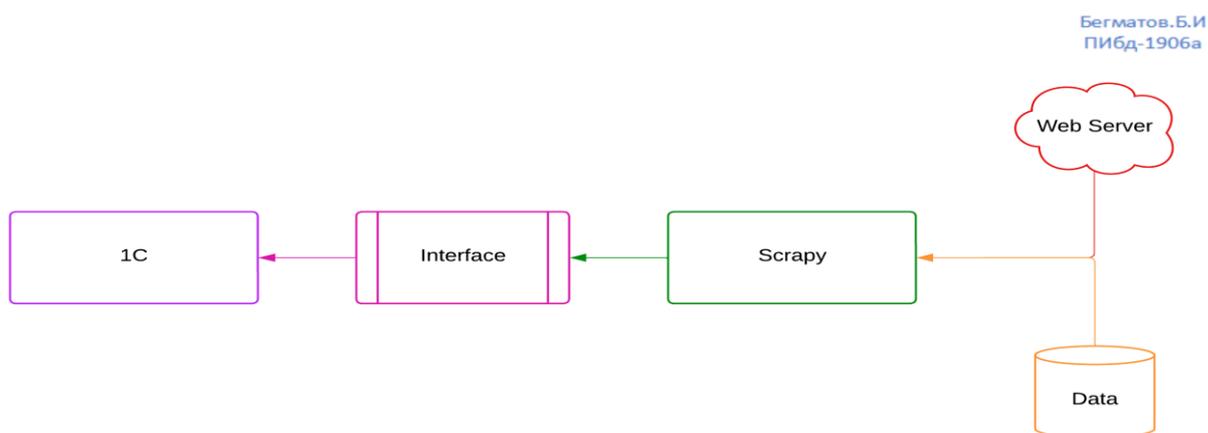


Рисунок 4 – Проект программного обеспечения

Программа для парсинга работает через модульные структуры, где каждый модуль отвечает за свою часть функционала, что обеспечивает устойчивость и эффективность выполнения задач.

Веб-сервер подключен к программе и обеспечивает её стабильную работу, обеспечивая доступность сервиса в любое время

Календарный план проекта можем рассмотреть в таблице 4.

Таблица 4 - Календарный план проекта

Исследование рынка и выбор технологий для парсинга.	1-2 месяца
Разработка модулей парсинга и интеграция	4-6 месяца
Доработка системы	1 месяц
Интеграция с 1С и настройка хранения данных	1-2 месяца
<ol style="list-style-type: none"> 1. Обучение и внедрение: 2. Проведение обучения для сотрудников компании 3. Внедрение системы в рабочие процессы 4. Поддержка и мониторинг (непрерывно) 5. Обеспечение технической поддержки и регулярное обновление системы 6. Проведение регулярных мониторингов работы системы и корректировка, если это необходимо. 	1-2 месяца
7. Итог запланированный период времени	8-13 месяцев

Внедрение системы парсинга данных в компанию «Виват Интеллект» требует четкого понимания бизнес-целей и требований к проекту. Это позволит разработать качественный календарный план, включающий все ключевые этапы работы, и обеспечит успешное выполнение проекта в срок. Конечная цель — повысить эффективность работы компании за счет автоматизации и оптимизации процессов анализа данных. Общая длительность проекта занимает 1 год и 1 месяц по плану, но на его реализацию могут действовать многие факторы, что могут удлинить или сократить время реализации проекта.

Вывод по второй главе

Проанализированы различные инструменты и сервисы для парсинга данных, такие как Web Scraper.io, Data Miner, SiteAnalyzer, ParserOk, Octoparse, iDatica и A-Parser. Эти инструменты предлагают разные возможности, от простого визуального программирования парсинга до профессиональных многофункциональных решений. Они позволяют

автоматизировать сбор данных с веб-сайтов, а также интегрировать данные с другими системами, включая 1С. Важно выбрать подходящий инструмент, учитывая специфические потребности и технические возможности бизнеса.

Представлено функциональное моделирование «КАК ДОЛЖНО БЫТЬ». Преимущества новой модели:

- значительное уменьшение времени на сбор и обработку данных;
- исключение ручного ввода данных;
- увеличение производительности;
- легкость добавления новых источников данных и изменения правил парсинга.

Для успешного внедрения системы парсинга данных в компанию «Виват Интеллект» необходимо четко определить бизнес-цели и требования к IT-проекту. Основные цели проекта включают автоматизацию процессов сбора данных, повышение их качества, увеличение производительности по сбору данных, поддержку принятия бизнес-решений и снижение издержек., что кратко было представлено и сгруппировано по методологии FURPS+. Данный проект рассчитан в итоге на 8-13 месяцев.

Таким образом, веб-сервис парсинга позволит автоматизировать сбор и обработку данных, значительно снизив время и трудозатраты, увеличив точность и актуальность информации. Это позволит сотрудникам сосредоточиться на анализе данных и принятии бизнес-решений, улучшит качество отчетности и снизит операционные издержки. Кроме того, веб-сервис обеспечит гибкость и адаптивность, позволяя легко добавлять новые источники данных и изменять правила парсинга в зависимости от потребностей бизнеса.

Глава 3 Программная реализация модуля

3.1 Язык программирования Python

Python – это высокоуровневый язык программирования, язык известен своей простой и читаемой синтаксической структурой, что делает его доступным даже для новичков. Благодаря минималистичному синтаксису код легко понимать и поддерживать, что особенно важно при разработке и модификации веб-парсеров.

Python является одним из самых популярных языков программирования и часто используется для веб-парсинга благодаря своей простоте, мощным библиотекам и активному сообществу. Вот более подробный обзор, как Python применяется для веб-парсинга, включая его использование в 1С.

Python предлагает множество мощных библиотек для веб парсинга. Множество библиотек, доступных в Python, упрощают задачи, которые могли бы быть сложными в других языках. К примеру:

- BeautifulSoup;
- Scrapy;
- Selenium.

Еще один из больших преимуществ Python – Большое сообщество. Активное сообщество разработчиков и обширная документация делают поиск решений и примеров проще. Таким образом:

Веб-парсинг с помощью Python

Веб-парсинг — это процесс извлечения данных с веб-сайтов. Python предлагает несколько мощных инструментов и библиотек для этой задачи:

BeautifulSoup — это библиотека для парсинга HTML и XML-документов. Она предоставляет простые методы для навигации по дереву документа и извлечения данных. Преимущества BeautifulSoup включают:

Простота использования: позволяет легко находить теги и их содержимое.

Устойчивость к ошибкам: экстрагирует данные даже из плохо сформированного HTML.

Python имеет еще одно преимущество это гибкость. Python позволяет быстро прототипировать и изменять код, что полезно в условиях, когда требования могут меняться

Пример использования:

```
```python from bs4 import BeautifulSoup import requests
url = 'http://example.com' response = requests.get(url) soup =
BeautifulSoup(response.content, 'html.parser')
title = soup.find('title').text print(title) ```
```

Scrapy — это мощный фреймворк для веб-парсинга, который позволяет создавать веб-скреперы. Он предоставляет множество возможностей, включая работу с асинхронными запросами.

Подход к проектированию: позволяет организовывать проект, включающий пауков, обработчики и предметы.

Поддержка сложных проектов: Scrapy подходит для сбора данных с множества страниц и сайтов одновременно.

Пример создания паука в Scrapy:

```
python import scrapy
class MySpider(scrapy.Spider): name = "myspider" start_urls =
['http://example.com']
def parse(self, response):
 title = response.css('title::text').get()
 yield {'title': title}
```

Selenium — это инструмент для автоматизации веб-браузеров. Он может быть использован для парсинга динамических страниц, которые требуют взаимодействия с JavaScript.

Интерактивность: позволяет имитировать действия пользователя, такие как нажатие кнопок и заполнение форм.

Поддержка различных браузеров: работает с Chrome, Firefox и другими

браузерами.

Пример использования Selenium:

```
python from selenium import webdriver
driver = webdriver.Chrome() driver.get('http://example.com') title =
driver.title print(title) driver.quit() ``
```

Использование Python для парсинга в 1С

1С — это платформа для автоматизации бизнеса, и часто возникает необходимость интеграции данных из веб-ресурсов. Python может быть использован для этого в следующем ключе:

Экспорт данных: Собранные с помощью Python данные могут быть импортированы в 1С для дальнейшей обработки.

Упрощение интеграции: Python-скрипты могут совершать HTTP-запросы к API 1С или отправлять данные на сервер 1С через форматы, такие как JSON или XML.

Пример интеграции Python и 1С

Предположим, у вас есть данные, собранные с веб-сайта, которые вы хотите отправить в 1С. Вот простой пример: ``python import requests import json

```
data = {'title': 'Пример заголовка', 'content': 'Содержимое статьи'} url_1c =
'http://1c-server/api/endpoint'
```

```
response = requests.post(url_1c, json=data) if response.status_code == 200:
print('Данные успешно отправлены в 1С') else: print('Ошибка при отправке
данных:', response.status_code) ``
```

Python является очень удобным инструментом для веб-парсинга благодаря своим библиотекам, гибкости и большому сообществу разработчиков. Использование Python для парсинга вместе с платформой 1С открывает большие возможности для автоматизации рабочих процессов и интеграции данных, что делает его полезным навыком для разработчиков и аналитиков.

При парсинге веб-страниц важно соблюдать правила и этику:

robots.txt: перед началом парсинга проверьте файл robots.txt на сайте. Он указывает, какие страницы можно или нельзя парсить.

Частота запросов: нельзя отправлять слишком много запросов за короткий период. Это может перегрузить сервер и привести к блокировке IP. Использовать управление с задержками между запросами.

При веб-парсинге важно обрабатывать возможные ошибки:

Неизвестный ответ: если сервер возвращает ошибку (например, 404 или 500), ваш код должен это учитывать.

Сетевые ошибки: обработка возможных сетевых сбоев и повторных попыток запросов.

Хранение данных

Необходимо определить способ хранения извлеченных данных.

Форматы: JSON, CSV, базы данных (например, SQLite, PostgreSQL).

Ограничение данных: необходимо избегать дублирования данных, необходимо использование первичных ключей.

Если необходимо быстро собрать данные с множества страниц, рассматривается использование технологий, позволяющих выполнять запросы параллельно или асинхронно:

Threading / Multiprocessing: Используйте модули threading или multiprocessing для параллельной обработки.

Asyncio: В сочетании с библиотеками, такими как aiohttp, вы можете выполнять асинхронные HTTP-запросы.

Сложные сценарии парсинга

Работа с JavaScript: для парсинга динамических веб-страниц (где данные загружаются с помощью JavaScript) можно использовать.

Selenium: как упоминалось ранее, он позволяет загружать страницы и взаимодействовать с JavaScript.

Puppeteer: это порт Puppeteer для Python, который также управляет браузером и предоставляет возможность парсинга динамического контента.

API. Иногда лучше использовать API (если он предоставляется) вместо

парсинга HTML. API обычно предоставляет данные в структурированном формате (например, JSON), что облегчает их обработку.

При интеграции с другими системами данные собираются и отправляются в 1С, уточняется, какие системы будут использовать эти данные в дальнейшем:

Автоматизация отчетности: создание отчетов на основе собранных данных.

Мониторинг и аналитика: использование данных для принятия бизнес-решений.

Для углубления знаний в веб-парсинге и интеграции с 1С, можно использовать некоторые ресурсы, которые могут помочь:

- книги и онлайн-курсы: специализированные книги по Python и веб-парсингу, а также курсы на Coursera, Udemy или edX;
- документация библиотек: чтобы узнать о новых возможностях и методах;
- сообщество: участие в форумах и сообществах, таких как Stack Overflow, чтобы получать помощь и обмениваться опытом;
- для сбора цен на товары: парсинг сайтов для мониторинга цен на товары и услуг, например, для сравнения цен или аналитики конкурентов;
- анализ отзывов: сбор отзывов с различных платформ для анализа общественного мнения о продуктах или услугах;
- автоматизация публикаций: сбор контента с одного источника с целью автоматической публикации в другом месте, например, агрегирование новостей.

Веб-парсинг на Python — это мощный инструмент для извлечения и анализа данных из интернета. С правильным подходом и уважением к ресурсам, вы можете эффективно использовать эти данные для улучшения бизнес-процессов и принятия решений. Интеграция с 1С позволяет автоматизировать процессы и повысить эффективность работы организации.

## 3.2 Среда разработки

PyCharm — это кроссплатформенная интегрированная среда разработки (IDE) для языка программирования Python, хорошо поддерживает разработку как для Data Science, так и для Data Engineering, предлагая мощные инструменты для работы с большими объемами данных и автоматических задач.

В нашем случае, будем использовать Pycharm для разработки и работы с парсингом.

PyCharm действительно является одной из самых популярных интегрированных сред разработки (IDE) для языка Python. Она предоставляет широкий спектр возможностей, которые делают ее отличным выбором для разработки как в области Data Science, так и в Data Engineering. Рассмотрим подробнее, что именно делает PyCharm мощным инструментом для парсинга и работы с данными.

Рассмотрим основные функции PyCharm.

Поддержка языка Python. PyCharm предлагает обширные возможности для разработки на Python, включая:

- автодополнение кода: Умное автодополнение помогает ускорить процесс написания кода и снижает вероятность ошибок;
- подсветка синтаксиса: Удобная подсветка синтаксиса делает код более читаемым;
- инспекции кода: PyCharm автоматически анализирует код на наличие ошибок и предлагает рекомендации по его улучшению.

Работа с проектами:

- структурирование проектов: PyCharm позволяет легко организовать файлы и каталоги проектов, что особенно важно для больших приложений;
- управление зависимостями: Интеграция с инструментами, такими как pip и virtualenv, упрощает установку библиотек и управление

окружениями.

Инструменты отладки:

- встроенный отладчик позволяет устанавливать точки останова, просматривать значения переменных и шаг за шагом выполнить код, что упрощает поиск ошибок;
- поддержка различных библиотек для логирования упрощает анализ процессов.

Рассмотрим специальные возможности для Data Science и Data Engineering.

Интеграция с библиотеками. PyCharm поддерживает такие библиотеки и фреймворки, как:

- Pandas: для работы с табличными данными;
- NumPy: для работы с многомерными массивами и матрицами;
- Matplotlib и Seaborn: для визуализации данных;
- Scrapy и BeautifulSoup: для веб-парсинга данных.

Jupyter Notebooks

PyCharm поддерживает работу с Jupyter Notebooks, что позволяет писать, тестировать и визуализировать данные непосредственно в среде разработки. Это особенно удобно для исследовательского анализа данных.

Рассмотрим возможности интеграции.

Системы контроля версий. Поддержка Git и других систем контроля версий интегрирована в IDE, что позволяет легко управлять версиями вашего кода, отслеживать изменения и работать в команде.

Базы данных. PyCharm предоставляет инструменты для подключения к различным базам данных.

SQL и NoSQL: Вы можете взаимодействовать с базами данных напрямую из IDE, выполнять запросы и анализировать результаты.

Поддержка тестирования

PyCharm имеет встроенные инструменты для написания и запуска

тестов. Юнит-тесты: легко интегрируются с библиотеками, такими как `unittest` и `pytest`. Тестирование производительности: можно проводить тесты на производительность, чтобы убедиться, что ваш код работает эффективно.

Удобство работы с парсингом.

Постоянные задачи. При разработке парсеров может потребоваться автоматизация различных задач, таких как:

- очистка HTML-страниц;
- обработка и сохранение полученных данных в удобном формате (например, CSV или база данных);
- PyCharm позволяет создавать и запускать скрипты для автоматизации этих процессов, а также предоставляет инструменты для мониторинга их выполнения.

Визуализация данных. После парсинга и обработки данных можно использовать встроенные средства визуализации для создания графиков и отчетов, что делает анализ данных более наглядным и доступным.

Сообщество и ресурсы. PyCharm имеет обширное сообщество пользователей и множество обучающих ресурсов, включая:

Документация. Подробная документация на русском и английском языках помогает разобраться в функционировании таких инструментов.

Форумы и блоги: Сообщества пользователей делятся опытом, предложениями и советами по использованию PyCharm в различных сценариях.

Таким образом, PyCharm — это мощная среда разработки для Python, обеспечивающая все необходимые инструменты для эффективной работы с веб-парсингом и обработкой данных. Его функции помогают ускорить процесс разработки, облегчить отладку и тестирование, а также улучшить качество кода. Использование PyCharm в проектах Data Science и Data Engineering значительно упрощает рабочие процессы и повышает продуктивность.

### 3.3 Выбор Framework/Библиотеки

Python имеет богатый набор библиотек для веб-парсинга, каждая из которых предназначена для выполнения определенных задач:

BeautifulSoup - Одна из простых и популярных библиотек для парсинга HTML и XML документов. Она позволяет легко извлекать данные из веб-страниц, используя структуру дерева, и предоставляет интуитивный интерфейс для поиска и навигации по элементам.

Scrapy – это фреймворк для веб-скрейпинга, который позволяет легко создавать эффективные и масштабируемые парсеры. Scrapy обеспечивает управление запросами, обработку и сохранение данных, а также поддержку асинхронного выполнения кода. Кроме того, поддерживает автоматизацию повторяющихся задач и управление парсингом большого количества страниц, включая планирование и ротацию IP-адресов.

Selenium – Лучшее парсинга динамических страниц, которые используют JavaScript для генерации контента, Selenium предоставляет интерфейс для управления веб-браузером. Это позволяет эмулировать действия пользователя, такие как клики и ввод текста, что делает возможным извлечение данных с таких страниц.

Но выберем Scrapy, так как выполняет ряд ключевых функций:

- автоматизация;
- масштабируемость;
- работа с прокси серверами;
- работа с большого количества страниц.

Данный фреймворк Scrapy, выполняет все наши условия.

Scrapy — это мощный и гибкий фреймворк для веб-парсинга, который предлагает множество преимуществ, особенно когда речь идет о парсинге данных из системы 1С. Вот более подробное описание его преимуществ:

**Автоматизация процессов**

Scrapy позволяет автоматизировать процесс сбора данных, что

существенно экономит время и усилия. Вместо выполнения рутинных задач вручную можно настроить Scrapy для автоматического извлечения информации с заданных страниц по расписанию. Это особенно полезно для периодического сбора данных из 1С для анализа или отчетности.

#### Масштабируемость

Scrapy спроектирован с учетом возможности масштабирования. Вы можете легко расширять проекты, добавляя новые пауки (спайдеры) для различных источников данных. Благодаря структуре фреймворка, можно обрабатывать множество запросов параллельно, что значительно увеличивает скорость сбора данных. Это особенно полезно, если нужно извлечь данные из больших объемов информации, хранящейся в системе 1С.

#### Работа с прокси-серверами

Scrapy поддерживает работу с прокси-серверами, что позволяет обойти ограничения, наложенные на IP-адреса при выполнении запросов. Это важно для парсинга сайтов, которые могут блокировать запросы из одного IP после нескольких попыток. Использование прокси также помогает сохранить анонимность при сборе данных и снизить риск блокировки ваших запросов.

#### Обработка большого количества страниц

С помощью Scrapy можно настроить проект для обработки большого количества страниц одновременно. Фреймворк позволяет использовать асинхронные запросы, что значительно ускоряет процесс извлечения данных. Можно разрабатывать сложные стратегии обхода (например, обработка пагинации или сбор данных из категорий) без необходимости ручного управления каждой страницей.

#### Гибкость в работе с данными

Scrapy предоставляет множество инструментов для работы с данными, которые собираются:

Обработка данных: можно использовать встроенные пайплайны Scrapy для очистки и обработки данных перед их сохранением в нужном формате (например, в CSV, JSON или в базу данных).

Модульное выполнение: Scrapy позволяет разбивать проект на модули (пауки, пайплайны и расширения), что упрощает поддержку и расширение кода.

#### Расширяемость

Scrapy обладает возможностью расширения через использование пользовательских классов и middleware, что позволяет добавлять специфичные функции, такие как обработка ошибок или специальные методы аутентификации при работе с 1С.

#### Поддержка различных форматов данных

Scrapy поддерживает множество выходных форматов, таких как JSON, CSV, XML и интеграцию с базами данных (например, SQLite, MongoDB). Это позволяет легко сохранять собранные данные в нужном формате без дополнительных преобразований.

#### Сообщество и документация

Scrapy имеет активное сообщество и обширную документацию. Это означает, что легко найти примеры и решения распространенных проблем, что ускоряет процесс разработки и облегчает обучение.

Таким образом использование Scrapy для парсинга данных из системы 1С предоставляет множество преимуществ, которые делают его отличным выбором для автоматизации и масштабируемого сбора информации. Благодаря своей гибкости, поддержке асинхронных запросов и возможности обработки больших объемов данных, Scrapy позволяет эффективно извлекать и обрабатывать данные, что в свою очередь может значительно улучшить бизнес-процессы и аналитические задачи.

### **3.4 Веб -сервер**

Для загрузки веб-страницы браузер отправляет запрос на сервер, в пространстве памяти которого хранятся все документы. В текущей работе используется веб-сервер Apache. Apache — свободный веб-сервер с открытым

исходным кодом, является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и так далее.

Apache, широко известный как Apache HTTP Server, является одним из самых популярных и надежных веб-серверов в мире. Разработанный и поддерживаемый Apache Software Foundation, он предлагает множество функций и возможностей, что делает его отличным выбором для различных веб-приложений. Вот более подробное описание его характеристик и возможностей:

#### Кроссплатформенность

Apache может работать на различных операционных системах, включая:

Linux: наиболее популярная среда для развертывания веб-серверов. Apache отлично интегрируется с различными дистрибутивами Linux.

Windows: позволяет разрабатывать и тестировать веб-приложения на локальной машине.

Unix и BSD-системы: также поддерживаются и широко используются в серверных инфраструктурах.

Mac OS: позволяет разработчикам тестировать свои веб-приложения в среде macOS.

#### Надежность

Apache славится своей стабильностью и надежностью. Сервер может функционировать без сбоев в течение длительных периодов времени, что делает его идеальным для использования в коммерческих и критически важных веб-приложениях.

#### Гибкость и конфигурируемость

Apache обладает высокой гибкостью, что позволяет настроить его под конкретные нужды:

Конфигурационные файлы: Apache использует текстовые конфигурационные файлы, что позволяет пользователям легко изменять настройки сервера. Особенно полезен файл `.htaccess`, который может находиться в любом каталоге и может содержать директивы конфигурации.

Модули: Сервер поддерживает архитектуру модулей, что позволяет подключать дополнительные функциональные возможности по мере необходимости. Например, можно добавлять модули для работы с SSL, аутентификацией, пересылкой, кешированием и многими другими функциями.

#### Поддержка различных протоколов

Apache поддерживает различные протоколы, включая HTTP/1.1, HTTPS (HTTP Secure) и, с недавними обновлениями, можно настроить поддержку HTTP/2. Это позволяет улучшить производительность и безопасность передачи данных.

#### Безопасность

Apache предлагает множество функций безопасности:

Аутентификация пользователей: с помощью модулей, таких как `mod_auth_basic` и `mod_auth_digest`, можно настроить аутентификацию на уровне директории.

SSL и TLS: Поддержка шифрования данных с использованием SSL и TLS, что позволяет обеспечивать безопасность передаваемой информации.

Настройка прав доступа: позволяет ограничивать доступ к определенным ресурсам или директориям на основе заданных условий.

#### Улучшенные метрики и логирование

Apache предоставляет средства для отслеживания и анализа работы сервера:

Логи доступа и ошибок: позволяют собирать информацию о запросах (логирование доступа) и отслеживать ошибки (логирование ошибок), что упрощает процесс отладки и мониторинга.

Модули для анализа: существуют модули, которые интегрируют службы

анализа, такие как AWStats, для представления статистики посещаемости.

Поддержка сред разработки

Apache хорошо работает в комбинации с различными языками программирования и платформами, такими как:

PHP: наиболее популярный язык для веб-разработки.

Python: часто используется с фреймворками, такими как Django и Flask.

Perl, Ruby и другие: поддержка различных скриптовых языков для программирования на стороне сервера.

Комьюнити и поддержка

Apache обладает большим сообществом разработчиков и пользователей, что обеспечивает доступ к документированным ресурсам, форумам и статьям. Существуют различные руководства и материалы, которые охватывают все аспекты установки, конфигурации и оптимизации сервера.

Таким образом, Apache HTTP Server представляет собой мощный инструмент для развертывания и управления веб-приложениями. Его надежность, гибкость, возможности безопасности и поддержка множества платформ делают его идеальным выбором для как небольших, так и крупных веб-проектов. С помощью Apache можно легко адаптировать веб-сервер под любую задачу, что обеспечит стабильную и эффективную работу вашего веб-сайта или приложения.

### **3.5 Spider – запросы**

В Scrapy, спайдер (spider) — это специальный класс, который управляет процессом сборки данных с веб-сайтов. Он создает и отправляет HTTP-запросы к целевым URL, обрабатывает ответы и извлекает интересные данные. Давайте рассмотрим, как реализуются HTTP-запросы в Scrapy и как с ними работает спайдер.

Рассмотрим работу запросов в Scrapy.

Определение начальных URL (start\_urls)

Начальные URL — это список веб-страниц, с которых начинается сканирование. Спайдер отправляет запросы к этим URL и обрабатывает полученные ответы.

```
start_urls = ['http://example.com']
```

Метод `start_requests()`

Метод `start_requests()` позволяет отправлять кастомизированные запросы. Это полезно, если необходимо изменить способ формирования запросов перед их отправкой.

```
def start_requests(self):
 for url in self.start_urls:
 yield scrapy.Request(url=url, callback=self.parse)
```

Метод `parse()`

Метод `parse()` — основной метод, который обрабатывает ответ от HTTP-запроса. Обычно этот метод используется для извлечения данных из HTML-контента, который был загружен.

```
def parse(self, response):
 page_title = response.css('title::text').get()
 self.log(f'Title страницы: {page_title}')
```

Отправка дополнительных запросов

Внутри метода `parse()` или других методов можно отправлять дополнительные запросы.

Используя `scrapy.li` или `scrapy.FormRequest` для создания запросов `def parse(self, response):`

```
Обработка текущего ответа
...
Отправка запроса на другую страницу
next_page = response.css('a.next::attr(href)').get()
if next_page is not None:
 yield response.follow(next_page, self.parse)
```

HTTP-методы

Scrapy поддерживает различные HTTP-методы. Вы можете использовать scrapy.Request для GET-запросов и scrapy.FormRequest для POST-запросов.

```
Пример GET-запроса
```

```
yield scrapy.Request('http://example.com', callback=self.parse)
```

```
Пример POST-запроса
```

```
yield scrapy.FormRequest('http://example.com/form', formdata={'key':
'value'}, callback=self.parse)
```

Заголовки и куки

Вы можете добавлять заголовки и куки к запросам, чтобы имитировать запросы от браузера или для аутентификации.

```
yield scrapy.Request(url='http://example.com',
 headers={'User-Agent': 'Mozilla/5.0'},
 cookies={'session_id': '1234'},
 callback=self.parse)
```

Рассмотрим пример использования полного кода спайдера, который выполняет HTTP-запросы на веб-сайт и обрабатывает ответы:

```
import scrapy
```

```
class MySpider(scrapy.Spider):
```

```
 name = "my_spider" # Название спайдера
```

```
 start_urls = ['http://example.com'] # Начальные URL
```

```
 def parse(self, response):
```

```
 # Извлечение заголовка страницы
```

```
 page_title = response.css('title::text').get()
```

```
 self.log(f'Title страницы: {page_title}')
```

```
 # Извлечение всех ссылок на странице
```

```
 for href in response.css('a::attr(href)').getall():
```

```
yield response.follow(href, self.parse) # Переход по каждой ссылке
```

```
def start_requests(self):
```

```
 for url in self.start_urls:
```

```
 yield scrapy.Request(url=url, callback=self.parse)
```

# Чтобы запустить, сохраните этот код в файле `my_spider.py` и выполните команду:

```
scrapy crawl my_spider
```

Запуск спайдера

Сохраните код в файл `my_spider.py` в директории вашего проекта Scrapy (обычно в `spiders`).

Запустите спайдера из командной строки:

```
scrapy crawl my_spider
```

Таким образом, Scrapy предоставляет мощные инструменты для выполнения HTTP-запросов и обработки ответов. Основная логика заключается в создании запросов, их отправке и обработке полученных данных. Можно легко настраивать запросы, добавлять куки и заголовки, а также управлять процессом сканирования с помощью различных методов спайдера.

### **3.6 Реализация модуля**

Рассмотрим таблицу, объясняющую, почему используем указанные инструменты и платформы для парсинга в 1С, а также их возможности (таблица 5).

Эти инструменты и платформы делают процесс парсинга в 1С более эффективным и управляемым.

Таблица 5 – Выбор технологий для парсинга

Инструмент/ Платформа	Причина использования для парсинга в 1С	Возможности
Python	Предлагает множество мощных библиотек для веб-парсинга. Упрощает сложные задачи.	Множество библиотек для парсинга (BeautifulSoup, Scrapy, lxml) Простота в использовании Мощные возможности для обработки данных
Scrapy	Мощный фреймворк для создания веб-скреперов. Поддерживает асинхронные запросы и множество форматов выхода.	Поддержка JSON, CSV, XML Интеграция с базами данных (SQLite, MongoDB) Возможность асинхронных запросов Легкость сохранения данных в нужном формате
PyCharm	Популярная интегрированная среда разработки (IDE) для Python. Отличный выбор для Data Science и Data Engineering	Поддержка отладки и тестирования. Интеграция с системами контроля версий Удобный интерфейс для работы с кодом Расширяемость за счет плагинов
Apache HTTP Server	Мощный инструмент для развертывания и управления веб-приложениями. Подходит для любых масштабов проектов.	Надежность и гибкость Возможности безопасности Поддержка множества платформ Адаптация веб-сервера под любые задачи

Рассмотрим реализацию программного обеспечения схематично на рисунке 5.

Интеграция 1С с помощью API. Этот блок отвечает за связь между 1С и другими системами через API. Функционал: Использование API для передачи данных между 1С и внешними системами.

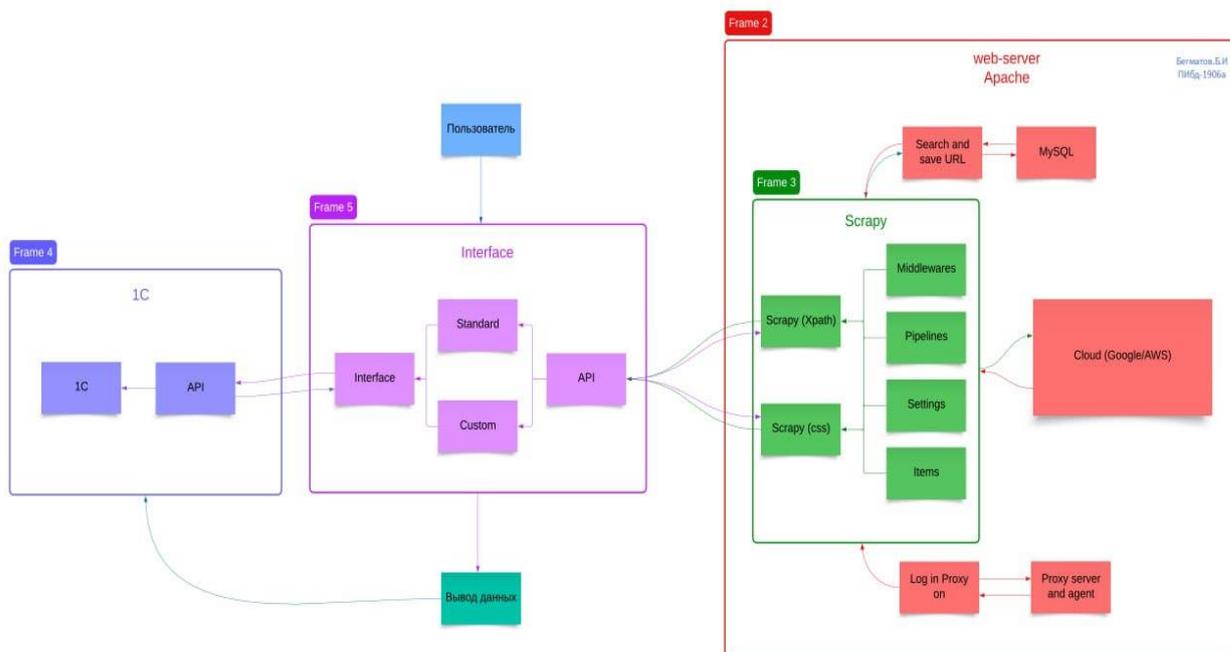


Рисунок 5 – Схема реализации программного обеспечения

Компоненты:

- API-клиенты, которые обращаются к IC;
- API-серверы, которые обрабатывают запросы и отправляют данные обратно клиентам.

Интерфейс с функционалом стандартного и пользовательского режима. Этот блок представляет собой интерфейс, который предоставляет два режима: стандартный и пользовательский.

Функционал:

- стандартный режим: Общие функции и возможности, доступные всем пользователям;
- пользовательский режим: Персонализированные настройки и функции, связанные с программой для парсинга.

Компоненты:

- пользовательские интерфейсы для каждого режима;
- настройки и персонализация.

Программа для парсинга. Программа, написанная с использованием модульной структуры, где каждый модуль отвечает за свою часть функционала.

Функционал:

- обработка данных и парсинг информации;
- взаимодействие модулей (например, settings, pipelines, items, XpathParsing, CSSParsing, middlewares).

Компоненты: модули, каждый из которых имеет четкую задачу и взаимодействует через методы и классы.

Веб-сервер - этот блок отвечает за связь между сервером и программой, обеспечивает доступ к программе через интернет.

Функционал:

- обработка запросов от пользователей и передачу их к парсинговой программе;
- управление данными и соединениями.

Компоненты:

- веб-сервер (например, Apache, Nginx);
- серверные скрипты и соединения с программой для парсинга

Для реализации модуля, как и указывали ранее используем Python 3.10, и создаем в PyCharm venv, а чтобы скачать и работать над scrapy, нам необходимо написать команду в консоле, чтобы начать процесс установки:

```
-pip install scrapy
```

Надо отметить, фреймворк Scrapy очень схож с веб фреймворком Django, где также идет работа через консоль команды.

Проект называем debox и вводим команду в консоль:

```
- scrapy startproject debox
```

Автоматически создаются Items, middlewares, pipelines, settings и spiders.

Самая важный файл – это debox, котором находятся внутри папки spiders, где и начинаем разрабатывать парсинг программы (рисунок 6).

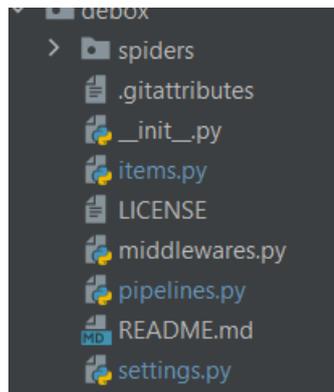


Рисунок 6– Скриншот проекта парсинга данных

Также, создаем дополнительный файл `deboxXpath`, так как планируется иметь 2 версии. Первая версия работает через `css`, а вторая - другая через `xpath`.

`Css` – будем использовать в основном для заказов;

`Xpath` – для сбора нужных нам данных.

Рассмотрим подключение библиотек и `url`

После создания есть класс `DeboxSpider` (рисунок 7)

```
class DeboxSpider(scrapy.Spider):
 name = 'deboxparce'
 allowed_domains = ['books.toscrape.com'] #books.toscrape.com
 start_urls = ['http://books.toscrape.com/'] #http://books.toscrape.com/
```

Рисунок 7 – Класс для подключения к URL

В переменной `Name`, указываем название команды, в котором будем использовать для запуска парсинг. К примеру, для `css` назовем “`deboxparcer`” и для `xpath` назовем “`deboxxparcer`”

Переменная `start_urls` – указывает что паук начнет свою работу по ссылке сайта, в котором нам необходимо спарсить.

Также, добавим ограничитель доменов переменную `allowed_domains` для `css`.

## Настройка выгрузка данных

Следующий шаг, это настройка выгрузка данных, в нашем случае экспортируем данные в data.xml, если планируем спарсить данные на русском языке, то используем {'data.xml': {'format': 'csv', 'encoding': 'cp1251'}}

Также, можем экспортировать помимо xml на json и csv (рисунок 8).

```
custom_settings = {
 'FEEDS': {'data.xml': {'format': 'xml', 'overwrite': True}}
}
```

Рисунок 8 – Настройка формата для баз данных

### Чтение и запись

Данный часть модуля отвечает за чтение и запись модуля.

Определение метода: Метод parse принимает два аргумента: self и response.

Извлечение сайтов: Переменная sites присваивает результат вызова response.css('article.product\_pod'). Этот вызов использует CSS селекторы для поиска всех элементов на странице, соответствующих селектору 'article.product\_pod'.

Цикл по сайтам: Код затем перебирает каждый элемент book в sites.

Извлечение относительного URL: Для каждого book он извлекает относительный URL с помощью book.css('h3 a ::attr(href)').get(). Эта строка находит атрибут href в теге a, который находится внутри тега h3, и получает его значение.

Конструирование полного URL:

Если относительный URL содержит подстроку 'catalogue/', он создает полный URL, объединяя 'https://books.toscrape.com/' с relative\_url.

В противном случае, он создает полный URL, объединяя 'https://books.toscrape.com/catalogue/' с relative\_url.

Переход по URL: В конце концов, он использует `response.follow(book_url, callback=self.parse_site_page)` для перехода по построенному URL и вызывает метод `parse_site_page` для обработки ответа (рисунок 9).

```
def parse(self, response):
 sites = response.css('article.product_pod')
 for book in sites:
 relative_url = book.css('h3 a ::attr(href)').get()

 if 'catalogue/' in relative_url:
 book_url = 'https://books.toscrape.com/' + relative_url
 else:
 book_url = 'https://books.toscrape.com/catalogue/' + relative_url
 yield response.follow(book_url, callback=self.parse_site_page)
```

Рисунок 9 – Модуль чтение и запись текста

### Чтение и запись рисунков

В этом модуле из URL через xpath тегов выводим URL ссылку на рисунки и парсим все что есть на сайте (рисунок 10).

```
def parse(self, response):
 items = ImagesItem()
 image_urls = response.xpath('//img/@src').extract()
 image_urls = ['https:' + url for url in image_urls]
 items['image_urls'] = image_urls
 return items
```

Рисунок 10 – Модуль чтение и сохранение рисунков

### Автоматический парсинг страниц

Эта часть модуля отвечает на автоматический переход на следующую страницу, если она имеется и парсит (рисунок 11).

```

next_page = response.css('li.next a ::attr(href)').get()
if next_page is not None:
 if 'catalogue/' in next_page:
 next_page_url = 'https://books.toscrape.com/' + next_page
 else:
 next_page_url = 'https://books.toscrape.com/catalogue/' + next_page
yield response.follow(next_page_url, callback=self.parse)

```

Рисунок 11 – Модуль перемещение к следующим страницам

### Структура данных и запрос

В самом пауке `debox.py` есть функция структура данных `parse_site_page` предназначена для парсинга страницы и извлечения ключевой информации, такой как название, цена, наличие, рейтинг и описание книги. Этот процесс начинается с получения HTML-кода страницы и извлечения строк таблицы, содержащих необходимые данные (рисунок 12).

```

def parse_site_page(self, response):

 table_rows = response.css("table tr")
 website_item = ExamItem()

 website_item['url'] = response.url,
 website_item['title'] = response.css('.product_main h1::text').get(),
 website_item['upc'] = table_rows[0].css("td ::text").get()
 website_item['product_type'] = table_rows[1].css("td ::text").get(),
 website_item['price_excl_tax'] = table_rows[2].css("td ::text").get(),
 website_item['price_incl_tax'] = table_rows[3].css("td ::text").get(),
 website_item['tax'] = table_rows[4].css("td ::text").get(),
 website_item['availability'] = table_rows[5].css("td ::text").get(),
 website_item['num_reviews'] = table_rows[6].css("td ::text").get(),
 website_item['stars'] = response.css("p.star-rating").attrib['class'],
 website_item['category'] = response.xpath("//ul[@class='breadcrumb']/li[@class='active']/preceding-sibling::li[1]/a/text()").get(),
 website_item['description'] = response.xpath("//div[@id='product_description']/following-sibling::p/text()").get(),
 website_item['price'] = response.css('p.price_color ::text').get(),

 yield website_item

```

Рисунок 12 – Модуль структуры данных

Класс `ExamItem` используется для определения структуры данных,

которые будут извлекаться с веб-страниц. Этот класс наследуется от scrapy.Item и содержит несколько полей, каждое из которых представляет определенный атрибут книги (рисунок 13).

```
new *
24 class ExamItem(scrapy.Item):
25 url = scrapy.Field()
26 title = scrapy.Field()
27 upc = scrapy.Field()
28 product_type = scrapy.Field()
29 price_excl_tax = scrapy.Field()
30 price_incl_tax = scrapy.Field()
31 tax = scrapy.Field()
32 availability = scrapy.Field()
33 num_reviews = scrapy.Field()
34 stars = scrapy.Field()
35 category = scrapy.Field()
36 description = scrapy.Field()
37 price = scrapy.Field()
38
```

Рисунок 13— класс для переменных данных

Конечный результат показан на рисунках 14 и 15.

```
- <product_type>
 <value>Books</value>
</product_type>
- <price_excl_tax>
 <value>£54.23</value>
</price_excl_tax>
- <price_incl_tax>
 <value>£54.23</value>
</price_incl_tax>
- <tax>
 <value>£0.00</value>
</tax>
- <availability>
 <value>In stock (20 available)</value>
</availability>
- <num_reviews>
 <value>0</value>
</num_reviews>
- <stars>
 <value>star-rating Five</value>
</stars>
- <category>
 <value>History</value>
</category>
- <description>
 <value>From a renowned historian comes a groundbreaking narrative of humanity's creation and evolution—a #1 international bestseller—that explores the ways in which biology and history have defined us and enhanced our understanding of what it means to be "human."One hundred thousand years ago, at least six different species of humans inhabited Earth. Yet today there is only one—h From a renowned historian comes a groundbreaking narrative of humanity's creation and evolution—a #1 international bestseller—that explores the ways in which biology and history have defined us and enhanced our understanding of what it means to be "human."One hundred thousand years ago, at least six different species of humans inhabited Earth. Yet today there is only one—homo sapiens. What happened to the others? And what may happen to us?Most books about the history of humanity pursue either a historical or a biological approach, but Dr. Yuval Noah Harari breaks the mold with this highly original book that begins about 70,000 years ago with the appearance of modern cognition. From examining the role evolving humans have played in the global ecosystem to charting the rise of empires, Sapiens integrates history and science to reconsider accepted narratives, connect past developments with contemporary concerns, and examine specific events within the context of larger ideas.Dr. Harari also compels us to look ahead, because over the last few decades humans have begun to bend laws of natural selection that have governed life for the past four billion years. We are acquiring the ability to design not only the world around us, but also ourselves. Where is this leading us, and what do we want to become?Featuring 27 photographs, 6 maps, and 25 illustrations/diagrams, this provocative and insightful work is sure to spark debate and is essential reading for aficionados of Jared Diamond, James Gleick, Matt Ridley, Robert Wright, and Sharon Moalem. ...more</value>
</description>
- <price>
 <value>£54.23</value>
</price>
</item>
</item>
```

```

- <item>
- <url>
<value>https://books.toscrape.com/catalogue/the-coming-woman-a-novel-based-on-the-life-of-the-infamous-feminist-victoria-woodhull_993/index.html</value>
</url>
- <title>
<value>The Coming Woman: A Novel Based on the Life of the Infamous Feminist, Victoria Woodhull</value>
</title>
<upc>e72a5dfc7e9267b2</upc>
- <product_type>
<value>Books</value>
</product_type>
- <price_excl_tax>
<value>£17.93</value>
</price_excl_tax>
- <price_incl_tax>
<value>£17.93</value>
</price_incl_tax>
- <tax>
<value>£0.00</value>
</tax>
- <availability>
<value>In stock (19 available)</value>
</availability>
- <num_reviews>
<value>0</value>
</num_reviews>
- <stars>
<value>star-rating Three</value>
</stars>
- <category>
<value>Default</value>
</category>
- <description>
<value>"If you have a heart, if you have a soul, Karen Hicks' The Coming Woman will make you fall in love with Victoria Woodhull."-Kinky Friedman, author and Governor of the Heart of Texas "What kind of confidence would it take for a woman to buck the old boy's club of politics in 1872? More than 140 years pre-Hillary, there was Victoria Woodhull. This book takes you back with a "If you have a heart, if you have a soul, Karen Hicks' The Coming Woman will make you fall in love with Victoria Woodhull."-Kinky Friedman, author and Governor of the Heart of Texas "What kind of confidence would it take for a woman to buck the old boy's club of politics in 1872? More than 140 years pre-Hillary, there was Victoria Woodhull. This book takes you back with a breathtaking, present-tense bird's eye view into a time when women's liberation was primarily confined to one woman's very capable, independent mind. I couldn't put it down."-Ruth Buzzi, Golden Globe Award winner and Television Hall of Fame inductee"Sadly, too many Americans have never heard of Victoria Woodhull, let alone learned of her story: her revolutionary campaign for the presidency at a time when women weren't even allowed to vote, her support for worker's rights, or her feminist commitment to equality, a century before the official battle over the Equal Rights Amendment. But in The Coming Woman, Karen Hicks brings Woodhull's efforts to life, and reminds us that some of our nation's greatest figures aren't always featured in the history books. It is a riveting account of an amazing woman and her struggle for justice and human dignity, told in an engaging and eminently readable style."-Tim Wise, author, "White Like Me: Reflections on Race from a Privileged Son""The Coming Woman" is a novel based on the life of feminist Victoria Woodhull, the first woman to run for U.S. President, 50 years before women could even vote!Running for President wasn't

```

Рисунок 14 - Полученный результат в XML

```

82 In stock (16 available),Young Adult,"Starting over sucks.When we moved to West Virginia right before my senior year, I'd pretty much resigned myself to thick accents, dodgy internet access, and a whole lot of boring, until I spotted my hot neighbor,
83 In stock (16 available),History,"Just in time for the presidential election of 2016 comes Political Suicide, a history of the best and most interesting missteps, peccadilloes, bad calls, back room hijinks, sordid pasts, rotten breaks, and just plain dumb mistakes in
84 In stock (16 available),Self Help,"If You Want People to Read Your Book, Writing It Is Only the Beginning There has truly never been a better time to be an author. For the first time, authors have direct access to the public via the Internet and can create a con
85 In stock (16 available),Sequential Art,"NEW HORROR SERIES FROM THE WALKING DEAD CREATOR ROBERT KIRKMAN! Kyle Barnes has been plagued by demonic possession all his life and now he needs answers. Unfortunately, what he uncovers along the wa
86 In stock (16 available),Add a comment,"It's Christmas Eve and private investigator Adam Fraley has lost his way while traveling a remote rural road in the foothills of the Colorado Rockies during a heavy snowstorm. At a loss as to where to turn next, he happ
87 In stock (16 available),Sequential Art,"Patience is a psychedelic science-fiction love story, veering with uncanny precision from violent destruction to deeply personal tenderness in a way that is both quintessentially and utterly unique in th
88 In stock (16 available),Sequential Art,"A Plea From the FutureOn the day that Naho begins 11th grade, she receives a letter from herself ten years in the future. At first, she writes it off as a prank, but as the letters' predictions come true one by one Naho
89 In stock (1 a homeles and a band of youths pursue him. Dalrymple renders this gang of eccentrics and mystics in a loose but realistic pen and ink style highly dynamic and tailor-made for depicting New York's teeming city streets. He has a flair for the drama
90 In stock (16 available),Fantasy,"Just as Annie and Liam are busy making plans to travel the world, a witch shows up and gives them a collection of postcards from the Magic Marketplace. Each postcard gives Annie and Liam the opportunity to travel to exotic
91 In stock (16 available),Sequential Art,"THE LONG-AWAITED STORY OF FANGIRLS TAKING ON TOKYO!Special large-size 2-in-1 edition of over 400 pages!""One of the best anime and manga for beginners. Enthusiasm - geeky and otherwise - is power in Princes
92 In stock (16 available),Science Fiction,"What if you could live multiple lives simultaneously, have constant, perfect companionship, and never die? That's the promise of Join, a revolutionary technology that allows small groups of minds to unite, forming i
93 In stock (16 available),Nonfiction,"The star of Orange is the New Black and Jane the Virgin presents her personal story of the real plight of undocumented immigrants in this countryDiane Guerrero, the television actress from the megahit Orange is the New B
94 In stock (16 available),Add a comment,"Displaying the most impressive throws, compelling armbars, dominant pins, and decisive chokes, judo is a particularly powerful sport and martial art that can bring opponents into submission in highly effective ways. W
95 In stock (16 available),Food and Drink,"It's time to venture beyond vanilla and chocolate and take your baking skills up a notch. Web's talking layers two, three, four, or more! Create sky-high, bakery-quality treats at home with Tessa Huff's 15
96 In stock (16 available),Sequential Art,"This New York Times Bestselling series continues with Jo, April, Mal, Molly, and Ripley as they take on everything that goes bump in the night. From scary stories to magical portals that lead to a land untouched by time, i
97 In stock (16 available),Sequential Art,"What a mystery!Jo, April, Mal, Molly, and Ripley are not your average campers and Miss Quinzella Thistwin Penniquiquil Thistle Crumpe's Camp for Hardcore Lady-Types is not your average summer camp. Between the r
98 In stock (16 available),Sequential Art,"FRIENDSHIP TO THE MAX!At Miss Quinzella Thistwin Penniquiquil Thistle Crumpe's camp for hard-core lady-types, things are not what they seem. Three-eyed foxes. Secret caves. Anagrams. Luckily, Jo, April, Mal, Molly,
99 In stock (16 available),Fantasy,"The year is 1779, and Carlo Morelli, the most renowned castrato singer in Europe, has been invited as an honored guest to Essterhazy Palace. With Carlo in Prince Nikolaus Esterhazy's carriage, ride a Prussian spy and one of
100 In stock (16 available),Food and Drink,"Cecilia Granata grew up cooking with her family in Italy. As a vegan, she learned to adapt her favorite recipes from around the country to be animal free while retaining the flavor and feeling of true Italian home cookin
101 In stock (16 available),Food and Drink,"A collection of stories and 100 sweet and savory French-inspired recipes from popular food blogger David Levovitz, reflecting the way Parisians eat today and featuring lush photography taken around Paris and in David
102 In stock (16 available),Sequential Art,"Susan, Esther, and Daisy started at university three weeks ago and became fast friends. Now, away from home for the first time, all three want to reinvent themselves. But in the face of hand-wringing boys, perso
103 In stock (16 available),Romance,"Caleb Stone was raised on the Upper East Side, where wealth and lineage reigns, and alternative lifestyles are hidden. It took him years to come out to his family, but he's still stuck in the stranglehold of their ex
104 In stock (16 available),Sequential Art,"The zombie apocalypse has never been more surreal! A mentally unhinged manga artist witnesses the beginning of a zombie outbreak in Tokyo, and he's certain of only two things: he's destined to be the city's hero, and
105 In stock (16 available),Romance,"One little white lie leads to more than she planned!Straight-laced lawyer Kate Matthews always plays by the rules. But when her ex gets engaged and a big promotion is on the line at work, she blurts out that she has a ne
106 In stock (16 available),Historical Fiction,"A Proper Historical Romance seriesThe first book in a new series which will focus on the lives of real historical figures who have intriguing stories of courtship that are recreated in fictionalized accounts based on real
107 In stock (16 available),Nonfiction,"While everyone is talking about the truth is that understanding the stats that underlie newspaper headlines, stock reports, weather forecasts, and so on is what helps you make
108 In stock (16 available),Romance,"Daunted by the singular sexual tastes and dark secrets of the beautiful, tormented young entrepreneur Christian Grey, Anastasia Steele has broken off their relationship to start a new career with a Seattle publishing house.B
109 In stock (16 available),Self Help,"In How to Be Miserable, psychologist Randy Paterson outlines 40 specific behaviors and habits, which if followed are sure to lead to a lifetime of unhappiness. On the other hand, if you do the opposite, you may yet joi
110 In stock (16 available),Science,"Around Christmas of 1882, while peering through a microscope at starfish larvae in which he had inserted tiny thorns, Russian zoologist Elie Metchnikoff had a brilliant insight: what if the mobile cells he saw gathering around t

```

Рисунок 15 - Полученный результат в csv

Итак, в нашем интерфейсе предусмотрено два режима: базовый и пользовательский.

Базовый режим: Этот режим предоставляет основные функциональные возможности, предназначенные главным образом для парсинга всех данных с

URL страницы. В базовый функционал входят:

Настройка path: Определение местоположения для сохранения данных.

URL: Поле для ввода URL ссылки.

Data: Выбор формата данных из трех вариантов: csv, xml, или json.

Text: если необходимо парсить текст с указанного URL, следует нажать соответствующую кнопку для активации этой функции.

Image: если нужно парсить изображения с указанного URL, следует нажать кнопку для активации этой функции.

Start: Кнопка запуска программы.

Close: Кнопка сброса всех настроек.

Эти возможности помогают настроить процесс парсинга максимально эффективно.

Вывод по третьей главе

Для реализации модуля использовался язык программирования Python, который предлагает множество мощных библиотек для веб парсинга. Множество библиотек, упрощают задачи, которые могли бы быть сложными в других языках. В нашем случае, для разработки и работы с парсингом использовали Pycharm .

Для реализации модуля использовался PyCharm, который предлагает обширные возможности для разработки на Python, включая:

- умное автодополнение помогает ускорить процесс написания кода и снижает вероятность ошибок;
- удобная подсветка синтаксиса делает код более читаемым.

PyCharm автоматически анализирует код на наличие ошибок и предлагает рекомендации по его улучшению.

Для Python был выбран фреймворк Scrapy, который выполняет все условия для 1С. Кроме того, поддерживает автоматизацию повторяющихся задач и управление парсингом большого количества страниц, включая планирование и ротацию IP-адресов.

Для загрузки веб-страницы браузер отправляет запрос на сервер, в пространстве памяти которого хранятся все документы. В текущей работе используется веб-сервер Apache. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и так далее.

Также, для выполнения HTTP-запросов и обработки ответов использовали Scrapy, который предоставляет мощные инструменты. Он создает и отправляет HTTP-запросы к целевым URL, обрабатывает ответы и извлекает интересные данные.

Результатом внедрения разработанной программы является значительное сокращение временных затрат и трудозатрат на обработку запросов, что приведет к экономии.

## Глава 4 Обоснование экономической эффективности проекта

Рассмотрим время, которое будет затрачено на данное мероприятие в таблице 6.

Таблица 6 — Временные фактические затраты

Этапы разработки	Фактические затраты (час)
1. Подготовка с описанием задач выполнения	8
2. Рассмотрение алгоритма по представленным задачам	10
3. Алгоритм решения задачи	20
4. Разработка программы по готовой блок-схеме	20
5. Автономное тестирование программы на ПК	82
6. Подготовка документов к внедрению	12
6.1. Согласование официальных документов и их утверждение	12
Итого	152

Для расчета коэффициента, учитывающего сложность разработки и затраты на корректировку исходного модуля, можно использовать следующую формулу:

$$K_{cl} = c \cdot (1 + p) \quad (1)$$

где  $c$  — коэффициент сложности программы;

$p$  — коэффициент коррекции программы в процессе разработки.

Эта формула помогает учитывать дополнительные ресурсы, необходимые для адаптации и улучшения модуля.

Следующая таблица коэффициентов по категориям, которая представлена в таблице 7.

Таблица 7 — Коэффициенты по категориям

Наименование коэффициента	Категории			
	1	2	3	4
Коэффициент сложности программы	1,25	1,5	1,6	2,0
Коэффициент коррекции программы	0,09	0,1	0,5	1,0

$$K_{сл} = 1,25 \cdot (1+0,1) = 1,375$$

Время, необходимое для разработки программы, с учётом её сложности, определяется путём умножения фактических временных затрат на соответствующий коэффициент сложности.

$$t_{раз_{сч}} = t_{раз_{ф}} \cdot K_{сл}, \quad (2)$$

$$t_{раз_{сч}} = 152 \cdot 1,375 = 209 \text{ часов}$$

Для определения средней часовой оплаты труда программиста важно сначала вычислить его годовой фонд заработной платы, принимая во внимание обязательные взносы в социальные фонды, которые являются значительной частью общих расходов на персонал. Эти взносы включают обязательные отчисления в Пенсионный фонд, Фонд обязательного медицинского страхования, а также Фонд социального страхования, что в совокупности увеличивает общую финансовую нагрузку на работодателя.

В ООО «Виват Интеллект» месячный оклад программиста устанавливается с учетом действующих на предприятии правил, что подразумевает использование различных надбавок. В частности, премия в размере 20% от оклада и районный коэффициент в размере 15% существенно влияют на итоговую сумму заработной платы. Такие подходы позволяют не только стимулировать сотрудников, но и учитывать особенности жизни и работы в определенных регионах, что является важным фактором для повышения конкурентоспособности на рынке труда.

Важно отметить, что в 2024 году общий фонд заработной платы увеличивается на 30% за счет социальных взносов. Распределение этих отчислений следующим образом: 22% идут в Пенсионный фонд РФ, 2,9% — в Фонд социального страхования РФ и 5,1% — в Федеральный фонд обязательного медицинского страхования. Эти индивидуальные ставки взносов становятся важным аспектом финансового планирования предприятия, оказывая влияние на его чистую прибыль и денежные потоки.

После определения общего фонда заработной платы рассчитывается месячный оклад программиста с учетом его квалификации и всех возможных надбавок. Этим шагом мы подходим к определению средней часовой оплаты труда, которая является конечным показателем для анализа затрат на труд и планирования бюджета компании. Такой подход к расчёту помогает экономистам и финансистам глубже понять структуру расходов на заработную плату и оптимизировать трудовые затраты, что в конечном итоге способствует устойчивому развитию бизнеса.

$$ЗП_M = ЗП_{ОСН} \cdot (1+K_{ДОП}) \cdot (1+K_{УР}) \cdot (1+K_{СН}), \quad (3)$$

где  $ЗП_{ОСН}$  — заработная плата специалиста;

$K_{ДОП}$  — коэффициент надбавки;

$K_{УР}$  — коэффициент региона Урал;

$K_{СН}$  — коэффициент взноса в социальный фонд.

$$З_{ПМ} = 20230 \cdot (1+0,2) \cdot (1+0,15) \cdot (1+0,30) = 36\,292,62 \text{ руб.}$$

Годовой фонд заработной платы с учетом вышеуказанных вычетов.

$$\Phi_{ЗРГ} = З_{ПМ} \cdot 12, \quad (4)$$

где  $З_{ПМ}$  — заработная плата в месяц;

12 — количество месяцев в году.

$$\Phi_{ЗРГ} = 36\,292,62 \cdot 12 = 435\,511,44 \text{ руб.}$$

Количество рабочих часов в году устанавливается в соответствии с производственным календарем на 2024 год.

$$n_p = (N - N_{ПВ}) \cdot N_{СМ} - N_{ПП} \cdot I, \quad (5)$$

где:  $N$  — общее количество дней в году;

$N_{ПВ}$  — количество праздничных и выходных дней в году;

$N_{ПП}$  — количество предпраздничных дней в году;

$N_{СМ}$  — продолжительность смены;

$I$  — величина сокращений предпраздничных рабочих дней.

Согласно производственному календарю на 2024 год, продолжительность рабочего времени составляет 1 970 часов при 40-часовой рабочей неделе. Средняя часовая оплата программиста определяется соотношением:

$$C_{разр} = \frac{\Phi_{ЗРГ}}{n_p}, \quad (6)$$

где  $\Phi_{ЗРГ}$  — годовой фонд заработной платы с учетом отчислений;

$n_p$  — число рабочих часов в году.

$$C = \frac{435\,511,44}{1\,970} = 221,07 \text{ руб./час.}$$

Затраты на оплату труда разработчика программы рассчитываются с использованием следующей формулы:

$$З_{разр} = t_{разр.сл} \cdot C_{разр}, \quad (7)$$

где  $t_{разр.сл}$  — трудоемкость создания программы, с учетом сложности программы, выраженная в часах;

$C_{разр}$  — средняя часовая оплата труда инженера-программиста.

$$З_{разр} = 209 \cdot 221,07 = 46\,203,63 \text{ руб.}$$

Определение годового фонда времени работы на персональном компьютере (ПК):

Для оценки себестоимости часа машинного времени, необходимо определить действительный годовой фонд времени работы ПК в часах. Время на профилактику: ежедневно — 0,5 часа, ежемесячно — 2 часа, ежегодно — 16 часов.

$$n_{ПК} = n_p - N_{РЕМ}, \quad (8)$$

где:  $N$  — общее количество дней в году;

$N_{ПВ}$  — количество праздничных и выходных дней в году;

$N_{ПП}$  — количество предпраздничных дней в году;

$N_{СМ}$  — продолжительность смены;

1 — величина сокращений предпраздничных рабочих дней.

$N_{РЕМ}$  — время на проведение профилактических мероприятий.

$$n_{ПК} = 1\,970 - 153,5 = 1\,816,5 \text{ часов.}$$

$$N_{РЕМ} = (N - N_{П} - N_{В}) \cdot K_{Д} + K_{М} \cdot 12 + K_{Г}, \quad (9)$$

где  $K_{Д}$  — коэффициент ежедневных профилактик; 0.5;  $K_{М}$  — коэффициент ежемесячных профилактик; (2);

12 — количество месяцев в году;

$K_{Г}$  — коэффициент ежегодных профилактик (6).

$$N_{РЕМ} = (365 - 118) \cdot 0,5 + 2 \cdot 12 + 6 = 153,5 \text{ часов.}$$

Годовые отчисления на амортизацию персонального компьютера

Балансовая стоимость ПК

$$C_{ПК} = C_P \cdot (1 + K_{УН}), \quad (10)$$

где  $C_P$  — рыночная стоимость ПК (определяется по прайсу);

$K_{УН}$  — коэффициент, учитывающий затраты на установку и наладку.

$$C_{ПК} = 32\,600 (1+0,1) = 35\,860 \text{ руб.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$Z_{ГАМ} = C_{ПК} \cdot H_A, \quad (11)$$

где  $C_{ПК}$  — балансовая стоимость ПК;

$H_A$  — норма амортизационных отчислений за год.

$$Z_{ГАМ} = 35\,860 \cdot 0,2 = 7\,172 \text{ руб.}$$

$$H_a = \frac{1}{T \frac{ПК}{ЭКС}} \cdot 100, \quad (12)$$

Где  $T \frac{ПК}{ЭКС}$  - полезный срок действия ПК

$$H_a = 1/5 \cdot 100 = 20\%$$

Затраты на электроэнергию

$$Z_{ГЭЛ} = P_{ЧПК} \cdot T_{ГПК} \cdot C_{ЭЛ} \cdot K_{ИНТ}, \quad (13)$$

где  $P_{ЧПК}$  — установочная мощность ПК;

$T_{ГПК}$  — годовой фонд полезного времени работы машины ( $n_{ПК}$ );

$C_{ЭЛ}$  — стоимость 1 кВт/ч. электроэнергии ( $C_{ЭЛ}=3,7$  руб./кВт/ч);  $K_{ИНТ}$  — коэффициент интенсивного использования ПК (0,9).

$$Z_{ГЭЛ} = 0,35 \cdot 1\,816,5 \cdot 3,7 \cdot 0,9 = 2\,117 \text{ руб.}$$

Текущие затраты на эксплуатацию ПК рассчитываются по формуле:

$$Z_{ГПК} = Z_{ГАМ} + Z_{ГЭЛ}, \quad (14)$$

где  $Z_{ГАМ}$  — годовые отчисления на амортизацию;

$Z_{ГЭЛ}$  — годовые затраты на электроэнергию для компьютера.

$$Z_{ГПК} = 7\,172 + 2\,117 = 9\,289 \text{ руб.}$$

Себестоимость часа работы на компьютере:

$$C_{ПК} = \frac{Z_{ГПК}}{n_{РПК}}, \quad (15)$$

где  $Z_{ГПК}$  — годовые затраты на ПК;

$n_{РПК}$  — годовой фонд полезного времени работы машины.

$$C_{ПК} = 9\,289 / 1\,816,5 = 5,11 \text{ руб./час}$$

Затраты на оплату машинного времени

Затраты на оплату машинного времени рассчитываются по формуле:

$$Z_{МАШ} = t_{МАШ} \cdot C_{ПК}, \quad (17)$$

$$Z_{МАШ} = 157 \cdot 5,11 = 802,27 \text{ руб.}$$

Общие затраты на создание программы

Общие затраты на создание программы определяем как сумму затрат на разработку программы и затрат на оплату машинного времени.

$$Z_{ОБЩ} = Z_{РАЗР} + Z_{МАШ}, \quad (18)$$

$$Z_{ОБЩ} = 46\,203,63 + 802,27 = 47\,009,90 \text{ руб.}$$

Итоговые затраты на расходные материалы

При определении совокупных расходов на создание программного обеспечения, помимо затрат на оплату труда разработчиков и использование машинного времени, следует учитывать и расходы на расходные материалы и комплектующие, используемые в процессе проектирования, разработки и внедрения. В таблице 8 приведены статьи затрат на расходные материалы с указанием их стоимости.

Таблица 8 — Расходные материалы

Статьи затрат	Стоимость за единицу	Количество	Итоговая стоимость
1. Пользование ресурсами Интернет	0,64 руб./час	30 ч.	19,20 руб.
2. Бумага	0,51 руб./л.	35 л.	17,85 руб.
Итого: (ЗРМ)			37,09 руб.

Общие затраты на разработку программного продукта приведены в таблице 9.

Таблица 9 — Общие затраты на разработку программного продукта

Статьи затрат	Условное обозначение	Итоговая стоимость
1. Общие затраты на заработную плату	ЗОбщ	47 009,90 руб.
2. Расходные материалы	ЗРМ	37,09 руб.
Итого: (СОбщ)		47 042,95 руб.

Предполагаемая цена программного продукта с учетом нормы прибыли

Вычислим предполагаемую цену продукта с учетом нормы прибыли:

$$Ц = C_{\text{Общ}} \cdot (1+N), \quad (19)$$

где  $C_{\text{Общ}}$  — общие затраты на разработку программного продукта;

$N$  — норма ожидаемой прибыли.

$$Ц = 47\,042,95 (1+0,2) = 56\,451,54 \text{ руб.}$$

Предполагаемая стоимость данного продукта, принимая во внимание ожидаемую норму прибыли, составляет 57 000 рублей.

Внедрение данного программного продукта в организацию приведет к значительному сокращению как временных, так и трудовых затрат, связанных с обслуживанием заявок сотрудников.

Для более детального анализа эффективности внедрения программного обеспечения в таблице 6 представлены ключевые показатели, которые способствуют расчету потенциальной выгоды от его использования. Эти показатели позволят выявить, насколько рационально и выгодно данное инвестиционное решение для организации, а также оценить влияние на общую продуктивность и оптимизацию рабочих процессов.

Таким образом, рассматривая данные показатели, можно провести комплексный анализ и сделать обоснованные выводы о финансовых преимуществах, которые принесет данное программное обеспечение.

В таблице 10 указаны показатели, помогающие рассчитать эффективность от внедрения данного программного продукта.

Таблица 10 — Расчет экономической эффективности

Показатели	Единица измерения	До внедрения программы	После внедрения программы
Время на обработки различной информации	мин.	120	10
Среднее количество обработок в месяц	шт.	45	30
Временные затраты в месяц	мин.	5 400	300
Сэкономленное время	час.	-	85
Средняя заработная плата сотрудников в час	руб.	200	200
Стоимость сэкономленных временных затрат в месяц	руб.	-	17 000
Время на формирование различных отчетов по предприятию	час.	20	0,2
Сэкономленное время	час.	-	19,8
Средняя заработная плата обслуживающего персонала в час	руб.	176	176
Стоимость сэкономленных временных затрат в месяц	руб.	-	3 484,80
Общая стоимость сэкономленных временных затрат в месяц	руб.		20 484,80

Следовательно, учитывая, что предполагаемая стоимость программного продукта составляет 57000 рублей, найдем срок его окупаемости.

$$T_{ок} = \frac{Ц}{Э} \quad (20)$$

где Ц — предполагаемая цена разрабатываемого продукта;

Э — экономическая эффективность.

$$T_{ок} = 57000/20484,80=3 \text{ мес.}$$

Полученный программный продукт обладает достаточно быстрой окупаемостью.

Сэкономленное свободное время специалиста организации позволит более эффективно работать с документами и отчетами.

Вывод: результатом внедрения разработанной программы является значительное сокращение временных затрат и трудозатрат на обработку запросов, что приведет к экономии.

Предполагаемая цена программного продукта. На основе всех затрат и ожидаемой прибыли мы можем рассчитать предполагаемую цену программного продукта, что составляет: 56451,54 руб.

#### Выводы по четвертой главе

Таким образом, окончательная цена программного продукта, включая прибыль, установлена в размере 57 000 руб., что обеспечивает быструю окупаемость 3 месяца и хорошие перспективы для компании. Этот продукт не только улучшит эффективность работы сотрудников, но и повысит качество предоставляемых услуг. Кроме того, использование этой программы позволит компании сосредоточиться на стратегических задачах, оставляя рутинные процессы за автоматизацией.

## Заключение

В результате выполнения производственной (преддипломной) практики была разработана первая рабочая версия модуля для парсинга, работающая с 1С. Парсинг в 1С является важной темой в контексте автоматизации бизнес-процессов и повышения эффективности обработки данных.

В работе рассматривалась деятельность ООО «Виват Интеллект».

Представлены процессные блок-схемы «Как есть» и «Как должно быть» для данной компании.

Преимущества предлагаемой новой модели:

- автоматизация процессов парсинга значительно уменьшит время, необходимое для сбора и обработки данных;
- устранение ручного ввода данных, что снижает вероятность ошибок;
- специалисты могут сосредоточиться на анализе данных и принятии бизнес-решений, вместо рутинной работы по сбору информации;
- возможность легко добавлять новые источники данных и менять правила парсинга.

Для успешного внедрения системы парсинга в ООО «Виват Интеллект» компания должна будет уделить внимание выбору технологии парсинга, обучению сотрудников и созданию инфраструктуры для обработки данных. Проведение регулярных мониторингов и доработок системы также требуется для обеспечения максимальной эффективности и повышения качества получаемых данных.

Для реализации модуля использовался язык программирования Python, который предлагает множество мощных библиотек для веб парсинга. Множество библиотек, упрощают задачи, которые могли бы быть сложными в других языках. В нашем случае, для разработки и работы с парсингом использовали Pycharm .

Для реализации модуля использовался PyCharm, который предлагает обширные возможности для разработки на Python, включая:

- умное автодополнение помогает ускорить процесс написания кода и снижает вероятность ошибок;
- удобная подсветка синтаксиса делает код более читаемым.

PyCharm автоматически анализирует код на наличие ошибок и предлагает рекомендации по его улучшению.

Для Python был выбран фреймворк Scrapy, который выполняет все условия для 1С. Кроме того, поддерживает автоматизацию повторяющихся задач и управление парсингом большого количества страниц, включая планирование и ротацию IP-адресов.

Для загрузки веб-страницы браузер отправляет запрос на сервер, в пространстве памяти которого хранятся все документы. В текущей работе используется веб-сервер Apache. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и так далее.

Также, для выполнения HTTP-запросов и обработки ответов использовали Scrapy, который предоставляет мощные инструменты. Он создает и отправляет HTTP-запросы к целевым URL, обрабатывает ответы и извлекает интересные данные.

Результатом внедрения разработанной программы является значительное сокращение временных затрат и трудозатрат на обработку запросов, что приведет к экономии.

Предполагаемая цена программного продукта. На основе всех затрат и ожидаемой прибыли мы можем рассчитать предполагаемую цену программного продукта, что составляет: 56451,54 руб.

Таким образом, окончательная цена программного продукта, включая прибыль, установлена в размере 57 000 руб., что обеспечивает быструю окупаемость 3 месяца и хорошие перспективы для компании.

## Список используемых источников

1. Бенгфорт Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка [Текст] / Б. Бенгфорт, Р. Билбро, Т. Охедо. — Санкт-Петербург: «Питер», 2018. — 367 с.
2. Бурков М. Python для парсинга: практическое руководство для программистов [Текст] / М. Бурков. — Санкт-Петербург: Питер, 2022. — 256 с.
3. Веб-скрейпинг на R: практическое руководство [Текст] / А. Сидоров. — Москва: Инфра-М, 2021. — 192 с.
4. Веб-парсинг на JavaScript: от основ к продвинутым техникам [Текст] / М. Кузнецов. — Санкт-Петербург: БХВ-Петербург, 2023. — 276 с.
5. ГЛОНАСС в логистике [Электронный ресурс]. — Режим доступа: [http://space-team.com/pressa/detail/ghonass\\_v\\_logistike/](http://space-team.com/pressa/detail/ghonass_v_logistike/) (дата обращения: 28.09.2024).
6. Ефремов Д. Парсинг для аналитиков: как извлекать данные для анализа [Текст] / Д. Ефремов. — Москва: Альпина Паблишер, 2020. — 280 с.
7. Как выполняется SWOT-анализ [Электронный ресурс]. — Режим доступа: <https://ivan-shamaev.ru/doing-swot-analysis/http://www.einsa.ru/strahovanie-gruzov.html> (дата обращения: 24.09.2024).
8. Как Майкрософт помогает разработчикам программного обеспечения добиться успеха [Электронный ресурс]. — Режим доступа: <https://msdn.microsoft.com/ru-ru/hh126360.aspx> (дата обращения: 28.09.2024).
9. Крамаров С.О., Овсянников В.А., Сахарова Л.В. и др. Автоматизированный сбор данных ключевых финансовых показателей предприятий IT-отрасли региона. Вестник кибернетики. 2022;3(47):39–45.
10. Князева Г. В. Мониторинг действий пользователя как часть системы комплексной безопасности компьютерных систем [Текст] / Г. В. Князева // Вестник Волжского университета имени В. Н. Татищева. 2015. — №1. — С. 6–

9.

11. Коваленко И. Веб-скрейпинг: эффективные методы извлечения данных [Текст] / И. Коваленко. — Москва: Цифровое будущее, 2020. — 300 с.

12. Контроль реализации управленческих решений [Электронный ресурс]. — Режим доступа: [https://studme.org/1299101011602/menedzhment/kontrol\\_realizatsii\\_upravlencheskih\\_resheniy](https://studme.org/1299101011602/menedzhment/kontrol_realizatsii_upravlencheskih_resheniy) (дата обращения: 28.09.2024).

13. Контроль спецтехники [Электронный ресурс]. — Режим доступа: <http://biz-navi.ru/solutions/special-machinery/> (дата обращения: 27.09.2024).

14. Контроль экскаваторов [Электронный ресурс]. — Режим доступа: <https://scout-gps.ru/resheniya/kontrol-spetstekhniki/control-dump-trucks/> (дата обращения: 29.09.2024).

15. Кулагин П. Продвинутый парсинг сайтов с помощью Scrapy [Текст] / П. Кулагин. — Казань: Казанский университет, 2021. — 350 с.

16. Кузнецов А. Веб-парсинг на Java: от простого к сложному [Текст] / А. Кузнецов. — Москва: Вильямс, 2022. — 304 с.

17. Курс по парсингу Python: от новичка до профессионала [Электронный ресурс]. — Режим доступа: <https://www.coursera.org/learn/web-scraping> (дата обращения: 05.10.2024).

18. Ларин С. Инструменты парсинга: от простого к сложному [Текст] / С. Ларин. — Новосибирск: Сибирское университетское издательство, 2023. — 232 с.

19. Методы парсинга сайтов [Электронный ресурс]. — Режим доступа: <http://seodrom.ru/parsing-saitov> (дата обращения: 28.09.2024).

20. Митрохин В. Эффективный парсинг: технологии и практики [Текст] / В. Митрохин. — Казань: Издательство Казанского университета, 2019. — 450 с.

21. Мониторинг транспорта [Электронный ресурс]. — Режим доступа: <https://scout-gps.ru/resheniya/monitoring-transporta/> (дата обращения: 28.09.2024).

22. Носов А. Парсинг данных для начинающих: полное руководство по

сбору информации из интернета [Текст] / А. Носов. — Санкт-Петербург: БХВ-Петербург, 2021. — 256 с.

23. Описание среды разработки Microsoft Visual Studio [Электронный ресурс]. — Режим доступа: <https://open-file.ru/programs/microsoft-visual-studio> (дата обращения: 01.10.2024).

24. Организация ООО «Автоконтейнер» [Электронный ресурс]. — Режим доступа: <http://www.list-org.com/company/6607807> (дата обращения: 22.09.2024).

25. Организация перевозок товаров автомобильным транспортом [Электронный ресурс]. — Режим доступа: <https://scicenter.online/promyis-hlennos-rossii-scicenter/organizatsiya-perevozok-tovarov-avtomobilnyim.html> (дата обращения: 28.09.2024).

26. Парсинг [Электронный ресурс]. — Режим <https://www.seonews.ru/glossary/parsing/> (дата обращения: 28.09.2024).

27. Парсинг данных с сайта [Электронный ресурс]. — Режим доступа: <https://semantica.in/blog/chto-takoe-parsing.html> (дата обращения: 10.10.2024).

28. Подчернина А.М., Коновалова А.И., Коновалов Е.Ю. и др. Автоматизированный сбор и анализ медицинских статистических данных. Московская медицина. 2019;5(33):86–88.

29. 30+ парсеров для сбора данных с любого сайта. Хабр. URL: <https://habr.com/ru/companies/click/articles/494020/> (дата обращения: 25.10.2024).

30. Романенко С. Технологии парсинга: секреты и советы [Текст] / С. Романенко. — Екатеринбург: УрФУ, 2021. — 200 с.

31. Python для анализа данных: парсинг веб-страниц и работа с API [Текст] / М. Гусев. — Москва: ДМК Пресс, 2023. — 288 с.

32. Смирнов Р. Парсинг данных в условиях большого объема информации [Текст] / Р. Смирнов. — Екатеринбург: УрГЭУ, 2020. — 240 с.

33. Суханов Н. Эффективный парсинг с использованием BeautifulSoup и Selenium [Текст] / Н. Суханов. — Москва: МФТИ, 2022. — 198 с.

34. Сбор данных с помощью Scrapy [Электронный ресурс]. — Режим доступа: <https://scrapy.org/> (дата обращения: 15.10.2024).
35. Степная Елена. Как выбрать решение для парсинга сайтов: классификация и большой обзор программ, сервисов и фреймворков. Хабр. URL: <https://habr.com/ru/articles/521646/> (дата обращения: 03.10.2024).
36. Система мониторинга грузового транспорта [Электронный ресурс]. — Режим доступа: [http://space-team.com/monitoring\\_transporta/industry\\_solutions/transportation/#system1](http://space-team.com/monitoring_transporta/industry_solutions/transportation/#system1) (дата обращения: 03.10.2024).
37. Страхование грузов и грузоперевозок [Электронный ресурс]. — Режим доступа: <http://www.einsa.ru/strahovanie-gruzov.html> (дата обращения: 08.10.2024).
38. Транспортные тарифы [Электронный ресурс]. — Режим доступа: <https://znaytovar.ru/new2652.html> (дата обращения: 02.10.2024).
39. Хартман О. Основы веб-парсинга на Python [Текст] / О. Хартман. — Киев: Издательство Ника-Центр, 2021. — 224 с.
40. Черемисинов А. Парсинг с Python: практическое руководство по автоматическому извлечению данных из веба [Текст] / А. Черемисинов. — Москва: Издательство Манн, Иванов и Фербер, 2020. — 320 с.
41. Чтение и запись текстовых файлов. StreamReader и StreamWriter [Электронный ресурс]. — Режим доступа: <https://metanit.com/sharp/tutorial/5.5.php> (дата обращения: 06.09.2024).
42. Шевчук Д. Стратегический менеджмент: конспект лекций [Текст] / Д. Шевчук. — Москва: Litres, 2017. — 1008 с.
43. Noah Kalson. Сбор данных в 2022 году: все, что вам нужно знать. URL: <https://ru-brightdata.com/blog/whybrightdata-ru/web-data-collection-2022> (дата обращения: 20.10.2024)
44. ЭКГ-20 начал работать на Сахалине [Электронный ресурс]. — Режим доступа: <https://www.uralmash.ru/press-center/news/5679/> (дата обращения: 22.09.2024).
45. Язык программирования C# [Электронный ресурс]. — Режим до-

стуга: <https://learn-code.ru/yazyki-programmirovaniya/c-sharp> (дата обращения: 21.09.2024).

46. 8 известных структур данных, о которых спросят на собеседовании [Электронный ресурс]. — Режим доступа: <https://proglib.io/p/8-data-structures/> (дата обращения: 29.09.2024).

47. WPWIN [Электронный ресурс]. — Режим доступа: <http://kpms.ru/Automatization/WPwin.htm> (дата обращения: 22.09.2024).

48. C# язык программирования [Электронный ресурс]. — Режим доступа: <http://progopedia.ru/language/csharp/> (дата обращения: 21.09.2024).

49. EPPlus [Электронный ресурс]. — Режим доступа: <https://github.com>

50. /JanKallman/EPPlus (дата обращения: 26.09.2024).

51. EPPlus [Электронный ресурс]. — Режим доступа: <https://riptutorial.com/ru/epplus> (дата обращения: 26.09.2024).

52. MaterialSkin [Электронный ресурс]. — Режим доступа:

53. <https://www.nuget.org/packages/MaterialSkin/> (дата обращения: 28.09.2024).

54. MaterialSkin для .NET WinForms [Электронный ресурс]. — Режим доступа: <https://github.com/IgnaceMaes/MaterialSkin> (дата обращения: 28.09.2024).

55. StreamWriter Class [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.io.streamwriter?view=netframework-4.8> (дата обращения: 02.09.2024).