Министерство просвещения РФ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики
Кафедра информатики, информационных технологий
и методики обучения информатике

Система формирования документов на основе базы данных и шаблонов

Выпускная квалификационная работа

Допущено к защите
Зав. кафедрой Сардак Л. В.
«22» мая 2023 г. _____

Исполнитель: Чарыев Багтыяр
обучающийся группы ИСиТ-1931
_____

Руководитель: Кудрявцев А. В.
кандидат педагогических наук,
доцент кафедры ИИТ и МОИ

Екатеринбург – 2023

Оглавление

Введение

Современные технологии помогают не только усовершенствовать процессы на производстве, но и значительно облегчить общую деятельность компании и сотрудников. При работе с документами часто стоит задача обработки данных, полученных с web-страниц. Обычно данные обрабатываются в ручном режиме, что требует определенных затрат времени, а также не исключает появление ошибок. В настоящее время подобные процессы стараются автоматизировать, однако остается еще достаточное количество операций, выполняемых вручную. Одним из таких процессов является сбор публикаций автора. Система формирования документов на основе базы данных снижает время на обработку данных, повышает надежность результат, что делает данный процесс более эффективным. Следовательно, представленная нами работа является актуальной системой обработки данных. Разработанное нами программное обеспечение можно адаптировать и для других задач подобного рода.

Цель: разработать систему формирования документов на основе базы данных и шаблонов.

Задачи:

1. Проанализировать существующие системы, позволяющие передать информацию с web-страниц в базу данных.
2. Разработать программу ввода данных с web-страниц в базу данных;
3. Произвести анализ список публикаций авторов официальных сайтов;
4. Подготовить техническую документацию.

Глава 1.Теоретические основы автоматизации создания списка публикаций автора на основе сведений официальных сайтов.

1.1 Обоснование необходимости автоматизации создания списка публикаций автора

Часто при приеме на работу преподавателя или продлении контракта, а также для получения степени или звания требуют предоставить ряд документов, в том числе и список его публикаций. Данный список можно найти на различных сайтах, например, на elibrary.ru. Конечно, можно просто скопировать этот список, но гораздо удобнее сохранить его в базе данных, для дальнейших манипуляций с ним. Для этого нам необходимо собрать определенные данные с сайта, такой процесс называется «парсинг».

Понятие парсинга широкое. Этот термин был заимствован из английского языка «to parse», что означает «считывать». Если дать общее определение, парсинг — это процесс сбора интернет-данных и последующая их обработка и анализ. Парсерами называют программы, которые помогают собирать и систематизировать данные. Информацию можно брать как со своего веб-ресурса, так и с других сайтов.

При разработке информационной системы используется большое количество различных методологий и процессов

Методология представляет собой набор некоторых принципов и методик, используя которые можно достичь определенных целей и результатов.

Процессы в свою очередь представляют формальное описание всех бизнес-процессов разработки информационной системы.

В настоящие время наибольшее распространение в практике все больше находят так называемые гибкие методологии разработки [Error: Reference source not found, с. 26 – 27], которые состоят из семейства подходов к разработке программного обеспечения, которые ориентированы:

1. на применение интерактивной разработки;

2. динамическое формирование требований;

3. обеспечение реализации требований.

Примеры гибких методологий ARIS, Scrum, DSDM, RAD, SADT и другие.

Стоит рассмотреть и основные свойства информационных систем [Error: Reference source not found, с. 87 – 96].

В первую очередь любая информационная система предназначена для сбора, хранения и обработки информации. Поэтому в ее базисе лежит среда по работе с данными, которая должна обеспечивать должный уровень надежности хранения и обработки, а также эффективный доступ к ней. Стоит учесть, что в обычных вычислительных системах присутствие данной среды не является обязательным параметром.

Во вторую очередь организация любой информационной системы, ее функции должны соответствовать установленным критериям.

Наконец в третью очередь функционирование информационной системы должно подвергаться контролю со стороны персонала, а он в свою очередь должен понимать и использовать систему согласно описанным инструкциям и документам. Интерфейс для конечного пользователя информационной системы должен быть простым, удобным и легко осваиваемым, а также предоставлять весь описанный функционал, и в то же время, не иметь возможности, которые могли бы нанести вред информационной системе [Error: Reference source not found, с. 15 – 27].

В современном мире прогресс в различных областях, включающих: мощность и производительность компьютерных систем, сетевые технологии и широкие возможности интеграций компьютерной техники с различным оборудованием, позволяет постоянно расширять функциональные возможности информационных систем. Также параллельно с этим на протяжении последних лет происходит постоянный поиск новых, более удобных и универсальных, методов их программно-технологической реализации.

Применение информационных систем в текущие время позволяет [Error: Reference source not found]:

1. своевременно применять решения по менеджменту компании;
2. значительно повышать степень обоснованности всех принимаемых решений непосредственно за счет выполнения оперативного сбора, обработки и передачи информации;
3. повысить объемы обрабатываемой и хранимой информации из-за замены бумажные носители информации на электронные, также это позволяет более перерабатывать информацию и снижать объемы документов на бумаге;
4. увеличить эффективность управления и коммуникации за счет своевременного предоставления информации;
5. уменьшать трудозатраты или полностью освобождать работников от монотонной работы за счет ее автоматизации;
6. совершенствовать структуры потоков информации, ее анализа и системы документооборота в компании;
7. улучшать другие аспекты бизнес-процессов компании [Error: Reference source not found, с. 7].

Цель информационной технологии – производство информации для ее анализа человеком и принятия на основе его решения по выполнению какого-либо действия. [4]

Самым распространенным источником информации в современном мире является интернет.

Интернет (internet) – это глобальная сеть передачи данных, связывающая информационные системы и сети связи различных стран посредством глобального адресного пространства, основанном на использовании стека протоколов TCP/IP.

Особенности Интернета определяют его огромный потенциал и возрастающую роль в современном мире. Главной отличительной особенностью Интернета является своевременное обновление контента, а главное требование общества к сети – наличие актуальной информации на данный момент времени. [3]

Любой сайт, интернет-магазин или другой онлайн-ресурс состоит из определенных элементов, одним из которых является Web-страница.

Web-страница (web-страница, интернет-страница) – это одна из составных частей Web-сайта, интернет-магазина, портала или блога в интернет-пространстве. [3]

Использование возможностей Интернета в разных сферах человеческой деятельности, например, в сфере торговли имеет множество преимуществ для потребителей при просмотре всего спектра товаров. В виду того, что количество информации непрерывно растет, найти нужную или уникальную достаточно сложно. Появляется необходимость внедрения специальных информационных систем для поиска нужной информации.

Информационная система представляет собой программный комплекс, который обеспечивает автоматизированный поиск и отбор необходимых данных (извлечение информации) на основе информационно-поискового языка и необходимых правил поиска.

К таким информационно-поисковым программам относятся поисковики и парсеры.

Парсинг позволяет автоматизировать сбор информации с Web страниц и передать данные для дальнейшей обработки. Наиболее часто данные передаются в файл. В нашем случае данные необходимо не только записать в файл, но и преобразовать в базу данных, для того чтобы, например, можно было сформировать список литературы в любом требуемом формате.

Таким образом, автоматизация таких процессов, как сбор информации с сайтов, преобразование и запись в базу, формирование на основе базы данных новых документов позволяет существенно сэкономить время сотрудников, что повысит эффективность их труда.

1.2. Технологии формирования документов на основе данных, полученных с WEB-страниц

Информацию, которую собирает парсер можно собрать и вручную, но это долго. Программы автоматизируют сбор информации и помогают её интерпретировать. Можно представить, как человек открывает браузер, ходит по сайтам и копирует с них данные. Парсинг — то же самое, только ходит не человек, а робот.

Для парсинга можно использовать:

специальные готовые программы. Их очень много. Функционал некоторых программ ограничен, и они могут решить только одну задачу. Есть многофункциональные, которые могут собирать разного рода информацию из разных источников;

самостоятельно написанные программы. Парсер можно создать на практически любом языке программирования, например, PHP, C++ и Python.

Большой объём данных непросто систематизировать вручную. Парсинг данных помогает:

- заполнить карточки товаров на новом сайте — на заполнение вручную уйдёт много времени;
- привести сайт в порядок — парсинг поможет найти страницы с ошибками, карточки товаров с неправильным описанием, повторы, ошибки в информации об оставшихся товарах на складе;
- оценить среднюю стоимость продукта, собрать информацию по другим компаниям на рынке;
- регулярно следить за изменениями — например, повышением цен или нововведениями у прямых конкурентов;
- собрать тексты с зарубежных сайтов и перевести их автоматически.

Приведём несколько примеров программ для парсинга, которые подойдут для разных задач.

Screaming Frog SEO Spider. Сервис специализируется на работе с SEO-данными. Программа требует немного практики, но у неё большие возможности:

- ищет нерабочие ссылки,
- может просматривать robots.txt,
- обнаруживает дубликаты страниц,
- просматривает Sitemap.

Netpeak Spider. Тоже работает с SEO-показателями сайтов, а именно:

- сканирует слабые места в оптимизации сайта,
- помогает создавать карты сайта,
- готовит комплексный анализ структуры.

Xenu's Link Sleuth. Программа предназначена только для парсинга битых ссылок. Так как программа выполняет одну задачу, интерфейс лёгкий и понятный.

Церебро Таргет. Делает парсинг данных аудитории в ВК. Можно узнать, в каких сообществах состоит ваша аудитория. Анализирует фотографии вашей аудитории. Рисует портрет комментаторов. Если вы планируете использовать ВК как площадку для продвижения, определённо стоит воспользоваться этим парсером.

Segmento Target. Собрат Церебро Таргет, но он работает не только с ВК, но и с Instagram и Одноклассниками. Благодаря этому вы можете ещё подробнее изучить аудиторию и попробовать продвигаться на нескольких площадках.

Xmlatafeed. Узкоспециализированная программа. Позволяет постоянно мониторить ассортимент и цены на товары конкурентов. Все данные структурируются в таблицу со всеми ссылками на товары и датами обновлений страниц.

Компании стремятся автоматизировать все свои процессы. Это ускоряет работу и экономит деньги. Парсеры могут значительно сократить расходы на анализ рынка и увеличить продуктивность сайта. Если ни одна из готовых программ не подходит под ваш проект, вы всегда можете нанять программистов, которые создадут вам новую программу.

Таким образом, исходя из вышеизложенного, можно прийти к выводу о том, что необходимость создания документов на основе данных, полученных с Web-страниц, обуславливается своим удобством и пользой для современного человека, а именно экономией времени на поиск нужной информации из сети, сохранением данных в удобном формате для последующего использования.

Перед разработкой информационной системы, требуется рассмотреть и проанализировать существующие технологии по автоматизированному сбору и преобразованию данных из Web-ресурсов, обосновать выбранную технологию, как наиболее эффективную и полезную в использовании.

Технологией называется совокупность методов и инструментов для достижения желаемого результата; в широком смысле - применение научного знания для решения практических задач.

Рассмотрим существующие технологии автоматизированного поиска информации.

Сбор данных по средствам API

API – интерфейс взаимодействий между сайтом и сторонними программами и серверами, набор готовых классов, процедур, функций, структур, констант, предоставляемых приложением (сервисом, библиотекой) или операционной системой для использования во внешних программных продуктах. [3]

Вход и регистрация на разных онлайн-сервисах или платформах, осуществляемый через аккаунты в социальных сетях, является использованием API. В данном случае приложения используют базы данных социальных сетей. При этом сервис может получать информацию о пользователе и манипулировать ею в своих целях.

Open API – система для разработчиков сторонних сайтов, которая предоставляет возможность авторизовать пользователей социальных сетей на сайте.

Несмотря на все удобство использования API, существует одно ограничение – социальная сеть не может отдавать все данные, которые видны пользователям в интерфейсе. Эти ограничения имеют две причины:

- Социальные сети стараются сохранять приватность своих пользователей;
- Некоторые функции слишком сильно нагружают серверную часть приложения.

Для того, чтобы преодолеть эти ограничения, следует использовать такую технологию как парсинг Web-сайта.

API Google Analytics позволяет создавать персональные отчеты, собственные инструменты анализа, выгружать данные статистики в большом объеме, создавать виджеты с сайта, использующего аналитику.

Структура запроса включает в себя обязательные параметры:

- ids – уникальный id профиля в Google Analytics;
- start-date – дата начала сбора данных;
- end-date – дата окончания сбора данных;
- metrics – запрашиваемые показатели.

Необязательными параметрами являются:

- dimensions – сегментация по критериям;
- filters - фильтры;
- max-results – максимальный результат.

Инструменты, которые можно использовать при работе с API:

- Google Query Explorer + Excel;

К плюсам при использовании можно отнести возвращение до 10000 строк, выгрузку в TSV, использование до 10 параметров и 7 показателей в одном запросе.

- Различные надстройки для Excel;
- Коммерческие сервисы;
- Google Spreadsheet + Google API Script. [26]

1. Сбор данных с помощью средств копирования (эмуляции) поведения пользователя в браузере.

Одним из наиболее популярных средств эмуляции поведения пользователя в браузере является Selenium.

Selenium – это проект, в рамках которого разрабатывается серия программных продуктов с открытым исходным кодом:

- Selenium WebDriver;
- Selenium RC;
- Selenium Server;
- Selenium Grid;
- Selenium IDE. [3]

Selenium WebDriver – это программная библиотека для управления браузерами. Более короткое название при использовании WebDriver.

Это целое семейство драйверов для различных браузеров, а также набор клиентских библиотек на разных языках, позволяющими работать с этими драйверами.

Предназначается для организации распределенной сети, позволяющей параллельно запускать много браузеров на большом количестве машин. Selenium Grid имеет топологию «звезда», то есть в его составе имеется выделенный сервер, который носит название «хаб» или «коммутатор», а остальные сервера называются «ноды» или «узлы». На рисунке 3 представлена архитектура Selenium Grid.

Сеть может быть гетерогенной, то есть коммутатор и узлы могут работать под управлением разных операционных систем, на них могут быть установлены разные браузеры.

Одна из задач Selenium Grid заключается в том, чтобы «подбирать» подходящий узел, когда во время старта браузера указываются требования к нему – тип браузера, операционная система, версия, архитектура процессора и ряд других атрибутов.

Selenium IDE – плагин к браузеру Firefox, который может записывать действия пользователя, воспроизводить их, а также генерировать код для WebDriver или Selenium RC, в котором выполняются те же самые действия. [5]

Для парсинга html наиболее распространены следующие варианты:

Регулярные выражения. Являются наиболее универсальным и настраиваемым средством синематического разбора, но использовать исключительно их – сложная задача для разработчика системы в виду того, что потребуется сильно специализировать каждое регулярное выражение, которые создают дополнительную нагрузку на операционную систему.

BeautifulSoup, lxml – наиболее распространенные библиотеки для парсинга html-страниц и выбор одной из них обусловлен личными предпочтениями разработчика. Данные библиотеки тесно связаны тем, что BeautifulSoup стал использовать lxml в качестве внутреннего парсера для ускорения, а в lxml добавлен модуль Soupparser. [20]

Парсинг может быть осуществлен на любом языке программирования, работающем с веб-контентом. Веб-приложения для парсинга обычно пишут на C++, Delphi, PHP, Python. Проще всего писать парсер на языках высокого уровня, которые содержат библиотеки для работы с веб-контентом.

В книге Б. Бенгфорт, Р. Билбро, Т. Охедо «Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка» дается четкое понимание технологии парсинга, рассказывается для чего нужны парсеры и какие что представляют собой входные параметры. [8]

В книге приведены листинги различных вариантов написания парсера для наглядного представления.

Для лучшего понимания технологии парсинга была проанализирована статья «Методы парсинга сайтов», в которой описываются основные методы парсинга и методология парсинга в целом. Рассмотрены основные этапы работы парсера. [8]

Парсинг html-страницы представляет собой процесс, который можно разбить на три этапа:

1 этап. Получение исходного кода Web-страницы.

В разных языках программирования для этого предусмотрены различные способы.

Например, в языке программирования Python3 чаще всего используется библиотека «reqests», которая сохраняет «дерево» сайта в переменную. [7]

2 этап. Извлечение html-кода необходимых данных.

После получения исходно кода Web-страницы, требуется ее обработать следующими действиями:

- отделить обычный текст от гипертекстовой разметки;
- построить иерархическое «дерево» элементов документа;
- корректно среагировать на неправильный код;

сделать выборку нужной информации с Web-страницы. [7]

3 этап. Фиксация результата.

Обработав данные на странице, требуется их сохранить в необходимом виде для последующей обработки.

Существует несколько способов сохранения полученной информации:

- построить иерархические JSON-структуры;

записать в CSV-файл;

внести в базу данных;

сконвертировать в Excel-таблицу.

Программы-парсеры часто используют в таких областях, где происходит копирование данных с Web-ресурсов с целью дальнейшего размещения на своих Web-ресурсах, где информация за короткий промежуток времени становится неактуальной, а также где копирование информации вручную потребует значительных затрат человеческих ресурсов (например, в сфере инвестиций для сбора данных о курсе валют, стоимости ценных бумаг). [7]

Учитывая вышеизложенное, можно выделить основные области применения технологии парсинга:

анализ общественного мнения;

мониторинг СМИ в реальном времени;

маркетинговые исследования;

сбор информации о погоде;

обновление новостных порталов;

сбор и обработка спортивной статистики;

разработка мобильных приложений;

парсингу подвергаются сайты с кино, рецептами, книгами и т.д.

построение списка потенциальных пользователей на базе информации о пользователей ресурсов конкурентов; автоматическое ценообразование на базе анализа цен конкурентов. [7]

Изучив технологии формирования данных, полученных с Web-страниц и проведя анализ существующих технологий по автоматизированному сбору информации, можно прийти к выводу, что парсинг сайтов оказался самой оптимальной и эффективной технологией по сбору и обработке информации, так как парсинг сайтов совмещает в себе достоинства всех имеющихся технологий.

Создание собственной программы для парсинга помогает сэкономить значительное количество времени и средств.

Для создания информационной системы требуется провести сравнительный анализ технологий, необходимых для реализации приложения. В рамках ВКР проведен сравнительный анализ технологий:

- СУБД
- Пользовательский графический интерфейс
- Технологии и алгоритмы проверки документов

Определение выбора метода разработки стоит начать с выбора СУБД. СУБД – комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД [25].

В качестве СУБД будем рассматривать реляционные БД, поскольку для решения нашей задачи необходима согласованность данных [27]. Данный тип СУБД присутствует практически в любой организации.

Рассмотрим некоторые из них:

«SQLite» – компактная встраиваемая СУБД. Слово «встраиваемый» означает, что SQLite не использует парадигмы клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа.

Достоинства:

1) Компактная

2) Бесплатная

3) Простота

Недостатки:

1) Отсутствует парадигма клиент-сервер
2) Отсутствие возможности работы между разными сервисами и информационными системами

«MySQL» – свободная реляционная система управления базами данных. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Достоинства:

1) Популярная
2) Свободно распространяемая СУБД
3) Легкость администрирования
4) Простота

Недостатки:

1) Присутствуют некоторые проблемы с выполнением сложных запросов и построением плана запроса
2) Менее гибкий, по сравнению с PostgreSQL
3) Менее строгий в плане разработки

«PostgreSQL» Свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Достоинства:

1) Свободно распространяемая СУБД
2) Гибкая
3) Широкий функционал
4) Мощность

Недостатки:

1) Сложнее администрирование, чем в MySQL
2) Множественное чтение запросов медленнее, чем в MySQL

«Oracle DB» –объектно-реляционная система управления базами данных компании Oracle. Является одним из самых надежных и широко используемых реляционных СУБД. Система построена вокруг реляционной базы данных, в которой объекты данных могут напрямую обращаться к пользователям через структурированный язык запросов. Oracle Database –это коммерческий продукт, который стоит весьма дорого

Достоинства:

1) Мощность
2) Широкий функционал
3) Самый лучший вариант для OLTP – обработка транзакций в реальном времени

Недостатки:

1) Необходима лицензия
2) Дорогое масштабирование

На основе проведенного анализа было принято решение использовать СУБД PostgreSQL за ее мощность, бесплатность и популярность среди организаций [15].

Пользовательский интерфейс – способ и средства взаимодействия пользователя с программами. Он определяет взаимодействие человека с операционной системой (ОС) и прикладными программами (приложениями), работающими под её управлением. Наиболее распространёнными аппаратными средствами реализации интерфейса пользователя служат клавиатура, мышь, стилус, джойстик, экран монитора или компьютерного устройства (смартфона, цифровой камеры и др. В большинстве ОС применяется графический интерфейс пользователя. При этом для экранного отображения ввода и вывода команд пользователя и данных используются окна– области экрана, каждая из которых относится к одной из работающих программ. Элементы управления программой изображаются графически внутри окон (в виде меню, кнопок, полей ввода и др.). Выбор пользователем одного из элементов может быть сделан с помощью мыши, клавиатуры, джойстика или прикосновения к экрану (если экран сенсорный). Программа может производить вывод результатов обработки данных (например, на экран монитора) в виде текста, гипертекста, таблиц, диаграмм, видео и др. Стандартность графических элементов управления облегчает процесс освоения пользователем новых программ. [29]

Другим видом интерфейса пользователя является интерфейса командной строки: текстовые команды вводятся пользователем с клавиатуры в окне специальной программы (например, команда lsof, введённая в командной строке утилиты «Терминал», работающей под управлением OS X, позволяет вывести список всех файлов, открытых в данный момент).

При анализе реализаций интерфейсов предпочтение было отдано графическому интерфейсу.

Основными плюсами данного выбора являются:

1) Скорость работы с интерфейсом – намного быстрее использовать не только клавиатуру, но и манипулятор-мышь, что дает преимущество нашему интерфейсу перед стандартным командным интерфейсом.

2) Удобство использования – Большинство программ имеют именно данный вид интерфейса, что позволяет нам сделать вывод, что большинство пользователей уже работали с таким видом интерфейса и быстро поймут основные его принципы

Из минусов же можно выделить:

1) Ресурсоемкость – в сравнении с интерфейсами командной строки графические интерфейсы требуют больше вычислительных ресурсов пользователя. Для современных компьютеров это не является большой проблемой, так как нынешнее поколение компьютеров достаточно мощное, чтобы комфортно их поддерживать.

Преимущества перевешивают минусы, что позволяет нам убедиться в правильности нашего выбора, так как он на порядок удобнее реализации в командной строке и позволяет использующему данный интерфейс получить наилучший «пользовательский опыт».

Рассмотрим языки программирования для реализации пользовательского графического интерфейса:

«Java» – строго типизированный объектно-ориентированный язык программирования общего назначения. Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины [11]. Занимает высокие места в рейтингах популярности языков программирования.

Достоинства:

1) Отличная поддержка языка
2) Обширная сфера использования
3) Кроссплатформенность
4) Строго типизированный язык

Недостатки:

1) Требует большое количество памяти

2) Низкая скорость по сравнению с C и C++

«Python» – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами.

Достоинства:

1) Низкий порог вхождения
2) Динамическая типизация
3) Кроссплатформенность
4) Обширная сфера использования

Недостатки:

1) Медленная скорость работы
2) Требуют больше памяти

«PHP» – C-подобный скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Достоинства:

1) Низкий порог вхождения
2) Проверенные инструменты разработки

Недостатки:

1) Невозможно создать десктопное приложение
2) Медленная скорость работы при вызове функций

«JavaScript» – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Достоинства:

1) Доступность - изучив основы JavaScript, вы сможете без труда понять большинство из них и всегда повысить свою квалификацию
2) Кроссплатформенность
3) Полная интеграция с браузером

Недостатки:

1) Отсутствие чтения и загрузки файлов.
2) Доступен для злоумышленников: весьма легко встроить какой-либо вредоносный код, который может нанести большой урон.
3) Нет возможности выявить ошибки заранее, только на этапе работы.

Исходы из вышеописанного сравнения было принято решение использовать язык программирования PHP для написания приложения. Язык PHP был выбран за кроссплатформенность, за обширную документацию и скорость. В качестве среды разработки будем использовать OpenServer за его простоту и быстрое создание макета для интерфейса.

1.3 Техническое задание на разработку технологии

Составлено на основе ГОСТ 34.602-89 «Техническое задание на создание системы формирования документов на основе базы данных и шаблонов». [1]

1. Общие сведения.

1.1. Название организации-заказчика.

ФГБОУ ВО «УрГПУ».

1.2. Название продукта разработки (проектирования).

«Парсер списка публикаций».

1.3. Назначение продукта.

Разработанная информационная система предназначена создания списка публикаций автора на основе сведений официальных сайтов с целью последующей записи в базу данных с возможностью преобразования в документ.

1.4. Плановые сроки начала и окончания работ.

В соответствие с планом выполнения ВКР (01.09.2023 – 05.06.2023).

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Данный продукт разработки предлагается заказчику для автоматизированного сбора и систематизации данных из различных интернет-источников.

Среди основных сфер применения такой технологии можно выделить: данные размещенные на специальных ресурсах, сайты, специализирующиеся на публикации научных трудов.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Разработчик должен обладать следующими качествами:

- Умение работать с ПК на уровне уверенного пользователя,

- Умение разрабатывать программы на языке программирования Python, знание инструментов разработки.

  Пользователь должен обладать следующими качествами:
- Базовые навыки работы с ПК;
- Умение работать в интернет-браузере (Яндекс браузер, Chrome, Mozilla Firefox, Internet Explorer 11);
- Умение работать с программным обеспечением для чтения файлов Word, PDF.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Созданный продукт должен считывать Web-страницу в файл, находить в ней ключевые слова (или значения), вставлять их в готовый шаблон и записывать в документ.

3.2. Аппаратные требования.

- Процессор: Intel Core i3/i5;
- Разрядность: x86 (32-bit)/x64 (64-bit);
- Оперативная память: 2Гб и более;
- Свободное место на жестком диске: около 1Гб;
- Видеокарта: 128 Мб;
- Интернет – широкополосный доступ;
- Монитор с разрешением экрана от 1028x768 пикселей;
- Клавиатура, мышь.

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

- Операционная система: Windows 7, Windows 8, Windows 8.1, Windows 10 или более поздней версии;
- Браузер на усмотрение пользователя (например, Яндекс браузер, Chrome, Mozilla Firefox, Internet Explorer 11);
- Установленное программное обеспечение Python 3.5.1.

3.4. Указание программного обеспечения, используемого для реализации.

«Парсер списка публикаций» - программа для формирования документов на основе данных, полученных с Web-страниц, выполняющая основные текстовые операции.

Данная программа применяется для быстрого обхода интернет-страниц, определения технической информации, отбора нужной информации и отбрасывания ненужной, вывода конечных данных в необходимом виде.

Microsoft PDF – специальный формат электронных документов, который не зависит от выбранной ОС, программы просмотра электронных документов или еще чего-нибудь.

3.5. Для сетевых систем – особенности реализации серверной и клиентской частей.

Не предусмотрено.

3.6. Форматы входных и выходных данных

Входные данные парсера – текстовые данные с сайта.

Выходные данные парсера – база данных, содержащая список источников.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Не предусмотрено.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

3.9. Меры защиты информации.

Не предусмотрено.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

4.2. Размещение информации на экране, дизайн экрана.

4.3. Особенности ввода информации пользователем, представление выходных данных.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

5.2. Требования к содержанию отдельных документов.

6. Порядок сдачи-приемки продукта.

В соответствие с планом выполнения ВКР.

Глава 2. Разработка технологии формирования документов на основе данных, полученных с WEB-страниц

2.1. Функциональная модель по формированию документов на основе данных, полученных с WEB-страниц

На основании информации, полученной по результатам написания первой главы, в ходе изучения и анализа технологий формирования документов на основе данных, полученных с Web-страниц, для облегчения понимания и визуализации всех процессов, перед началом проектирования информационной системы необходимо построить модель бизнес-процессов.

Данная модель используется для описания технологии, а также установления связей между системой и внешними факторами.

Выберем технологию концептуального моделирования. Для этого проведем сравнительный анализ трех наиболее известных нотаций концептуального моделирования информационной системы таких как UML, ARIS, IDEF0.

UML (Unified Modeling Language) – унифицированный язык моделирования, который представляет собой графическую нотацию, предназначенную для моделирования и описания всех процессов, протекающих при разработке проекта [23].

ARIS (Architecture of Integrated Information Systems) – методология, а так же комплекс средств, который формализует информацию, анализирует и оптимизирует деятельность предприятия и представляет ее в виде графических моделей [23].

IDEF0 (Integration Definition for Function Modeling) – нотация, которая описывает и формализует бизнес-процессы, основанная на методологии и стандартах функционального моделирования [23].

На () представлена контекстная диаграмма процесса по формированию документов на основе данных, полученных с Web-страниц.

На (Рис 2) – декомпозиция этой диаграммы.

Рис 1 Контекстная диаграмма процесса по формированию документов на основе данных, полученных с Web-страниц

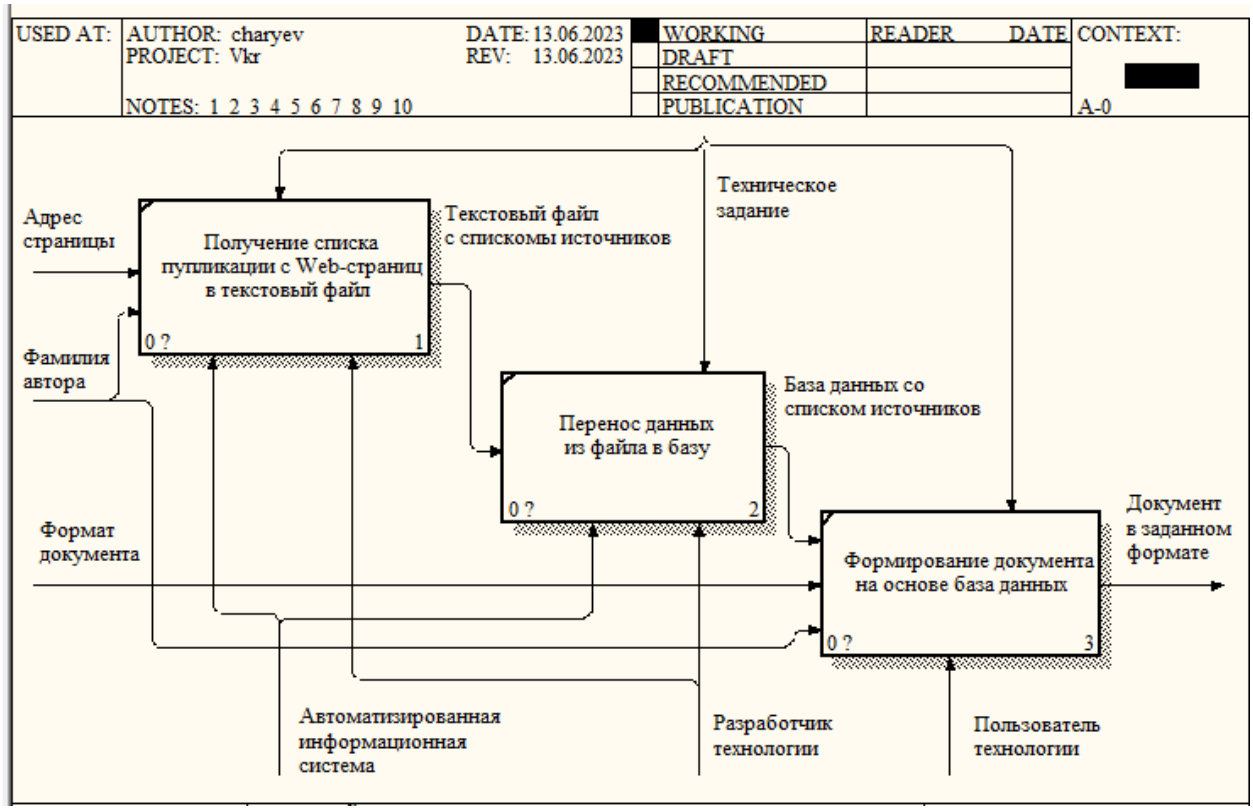


Рис 2 Декомпозиция диаграммы формированию документов на основе данных, полученных с Web-страниц

Представленная информационная система состоит из трех основных процессов:

1. Получение списка публикаций с Web-страниц в текстовый файл.
2. Перевод данных из текстового вида в поля базы данных
3. Формирование документа на основе базы данных.

Представленная модель определяет основные виды действий по преобразованию исходных данных в виде web-страницы в базу данных, а в дальнейшем в документ в произвольном формате.

Заказчик подает заявку на парсинг данных с сайта, содержащего публикации определенного автора, что подразумевает под собой получение данных с Web-страниц в тестовом формате с целью дальнейшего использования.

Таким образом, нами разработана концепция автоматизации преобразования источников публикации в документ любого вида.

2.2. Разработка информационной системы

Для считывания данных с Web-страниц разработчик создает программный продукт. Считывание данных берет свое начало с получения HTML-кода страницы. После проделанного анализа процесса «Технология формирования документов на основе данных, полученных с Web-страниц», перейдем к разработке информационной системы.

Для получения списка источников с Web-страниц, нами разработана программа на языке php. Рассмотрим ее работу:

Устанавливаем шрифт utf-8 для корректного отображения русских букв.

```
<meta charset="utf-8">
<?php
Берём контент из сохранённой веб страницы
$stri = file_get_contents('CharyevHTML.html');
Создаём текстовый файл для текста с тегами
```

```
$filename2 = __DIR__ . '/havetags.txt';
Создаём текстовый файл для текста без тегов
$filename = __DIR__ . '/notags.txt';
Сохраняем текст в файл с тегами
file_put_contents($filename2, $stri);
Убираем теги
$strip = strip_tags($stri);
Сохраняем текст страницы в файл без тегов
file_put_contents($filename, $strip);
Приведем последовательность команд для форматирования файла:
Находим начало списка источников
$stb = strpos($strip, 'Цит.');
Находим окончание списка
$ste = strpos($strip, 'Возможные действия');
Вычисляем длину списка
$ste = $ste - $stb;
Вырезаем список из страницы
$string = file_get_contents('./notags.txt', false, null, $stb, $ste);
Сохраняем в файл
file_put_contents($filename, $string);
```

Рассмотрим процесс преобразования текста, полученного файла со списком источников в базу данных. Создадим форму взаимодействия пользователя с системой (см. приложение 1). Форма запрашивает от пользователя имя файла с источниками литературы и имя таблицы базы данных. Далее пишем программу для записи содержимого файла в базу.

Устанавливаем шрифт utf-8 для корректного отображения русских букв.

```
mb_internal_encoding("UTF-8");
```

Задаем имя локального хоста, имя пользователя, пароль и имя базы данных.

$db_host = "localhost";

$db_name = "info";

$db_user = "root";

$db_pass = "root";

Устанавливаем соединение с сервером MySQL

$con = @mysqli_connect ($db_host,$db_user,$db_pass, $db_name)

or die("Ошибка при подключении к базе данных”);

Получаем имя таблицы от формы.

$table_name = $_POST['table_name'];

Пишем текст запроса для создания таблицы с полями: Идентификатор, название статьи, ФИО автора, указатель, текст (название сборника), город, год, число страниц.

$zapros0 = "CREATE TABLE $table_name

(id_statia INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,

name_statia TEXT, fio_avtor TEXT, ukazatel TEXT, text TEXT, gorod TEXT, goda TEXT, straniz TEXT, prim TEXT);";

Выполняем запрос.

$itog0 = mysqli_query($con, $zapros0);

Берем имя файла из формы

$file_name = $_FILES['file_name']['tmp_name'];

Далее преобразовываем данные из файла разбивая их на соответствующие поля и записываем в базу. Полный текст программы приведен в приложении 2.

2.3. Результаты работы информационной системы

Результатом работы информационной системы является «Программа-парсер», которая преобразует данные с WEB-страницы в тестовый файл, программа, реализующая интерфейс с пользователем и выполняющая перевод в данных с подробным описанием этапов разработки программного продукта, прошедшего апробацию в практическом применении, с приложением подробных скриншотов хода процесса разработки.

Создаем файл в текстовом редакторе (в блокноте) для того, чтобы записать в него полученные данные с Web-страниц.

Запускаем Web-сервер и SQL-сервер, например пакет программ OpenServer. Открываем Web-браузер, загружаем страницу elibrary.ru. Регистрируемся под именем автора и выводим список публикаций. Далее запускаем файл index.html, получаем текстовый файл notags.txt, содержащий список публикаций, на экране отображается «DONE», что означает действие выполнено.

Открываем файл index_000.html, выбираем созданный файл и указываем имя таблицы в базе (Рис 3).

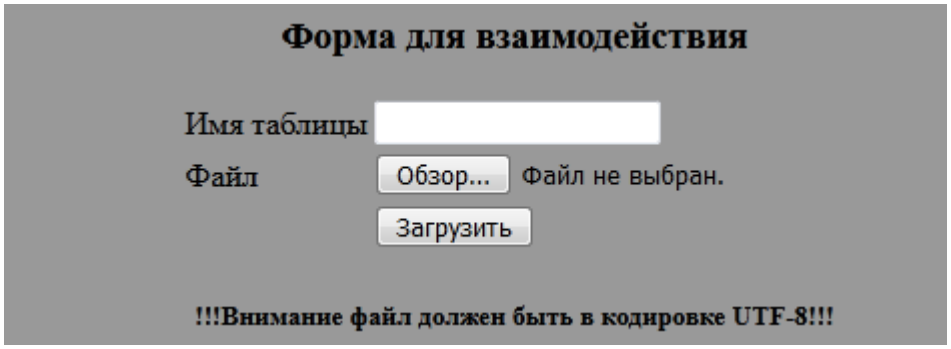


Рис 3 Форма для ввода данных

Нажимаем «Загрузить». Будет создана база данных «info», если таковой не было и в ней таблица с указанным именем (Рис 4).

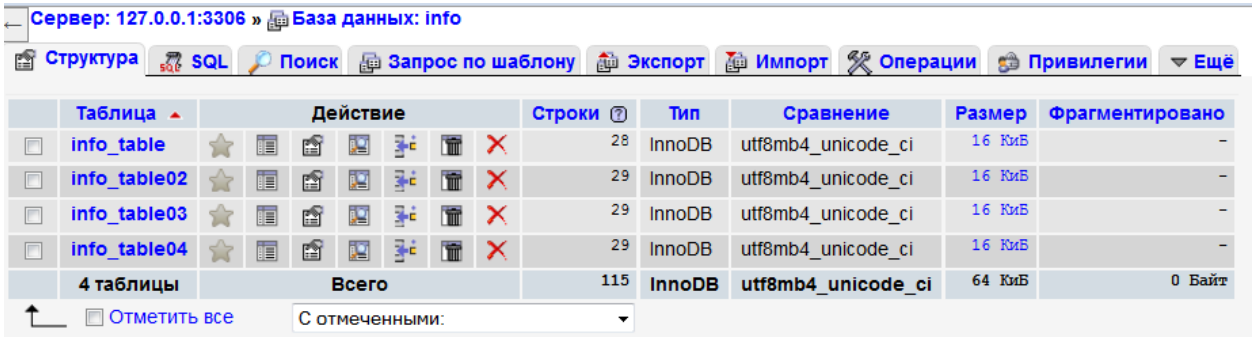


Рис 4 База данных с созданными таблицами

Каждая таблица будет иметь следующую структуру (Рис 5).

Рис 5 Структура таблица в базе

На (Рис 6) представлено содержание таблицы.

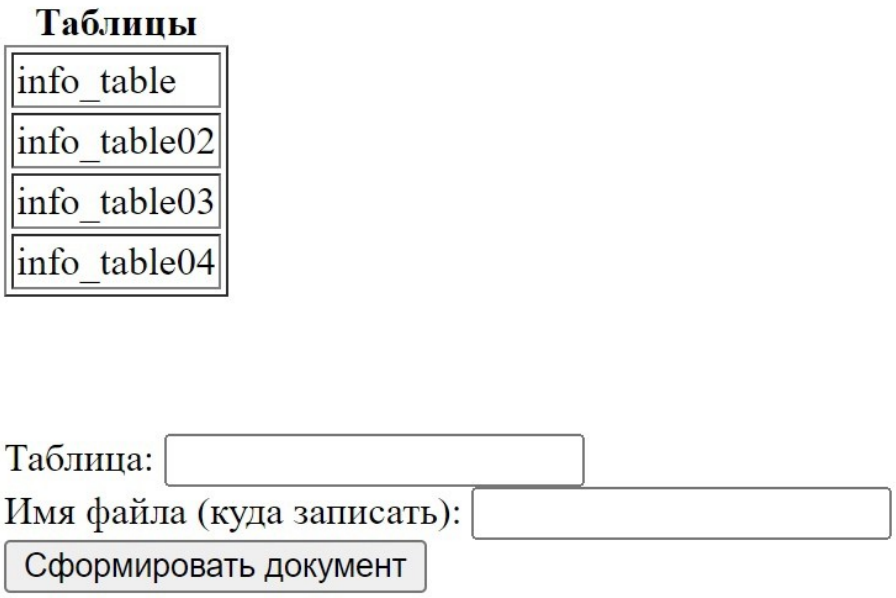
id_statia	name_statia	fio_avtor	ukazatel	text	gorod	goda	straniz
1	ИЗУЧЕНИЕ ДИСЦИПЛИНЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ» МЕТОД...	Кудрявцев А.В.	В книге			2021	163-175
2	ФОРМИРОВАНИЕ У СТУДЕНТОВ ПРЕДСТАВЛЕНИЯ РАБОТЫ АЛГ...	Кудрявцев А.В., Алексеевский П.И.	В сборнике			2021	233-238
3	ПРИЛОЖЕНИЕ 1 - ВЫПИСКА	Кудрявцев А.В.	В книге			2021	521
4	РАЗВИТИЕ СОВРЕМЕННОГО ВУЗА: НОВЫЕ МЕТОДЫ И ТЕХНОЛ...	Нагорнова А.Ю., Виниченко М.А., Ли Н., Алексеева И...		Коллективная монография	Ульяновск	2021	
5	ЦИФРОВОЙ ПЛАНЕТАРИЙ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА AR...	Кудрявцев А.В.				2021	57-62
6	ПРАКТИКА ВНЕДРЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ "ЦИФ...	Матвеева Т.В., Машкова Н.В., Ткачева О.Н., Кудрявц...				2021	66-70

Рис 6 Таблица с данными в базе

Таким, образом, данные из текстового файла корректно переведены в формат базы данных и распределены по соответствующим полям, что позволяет в дальнейшем на основе этих данных сформировать документ любого формата.

В данной программе необходимо ввести имя таблицы из предложенных, обработанных запросом SHOW TABLES из базы данных, и имя файла, куда будет выводиться готовая таблица, обязательно в формате Microsoft Word, например .doc или .docx

По нажатию кнопки "Сформировать Документ" пользователю необходимо лишь сохранить файл в удобное место и открыть, чтобы удостовериться, что данные действительно сохранились



Заключение

Результатом выполнения выпускной квалификационной работы стала программа «Парсер списка публикаций». Данный программный продукт позволяет считывать данные с Web-страниц с целью последующего формирования документа в удобном формате.

Реализована возможность сбора данных об источниках публикаций для заданных авторов и запись данной информации в текстовый файл с последующим преобразованием в базу данных и дальнейшем выводом результата в заданном формате.

В процессе выполнения данной работы была достигнута ее основная цель и решены все задачи, поставленные во введении, а именно:

Изучены технологии формирования документов на основе данных, полученных с Web-страниц;

Проанализированы существующие технологии формирования документов на основе данных, полученных с Web-страниц;

Разработана информационная система по формированию документов на основе данных, полученных с Web-страниц;

Проверен результат работы информационной системы.

В конце проделанной работы был проанализирован показатель эффективности работы программы «Парсер списка публикаций». Для этого было проведено тестирование парсера с целью проверки всех поставленных перед ним задач. По результатам тестирования стало ясно, что разработанный «Парсер списка публикаций» работает корректно.

Программа выполняет все предъявленные к ней требования. В процессе работы не возникает ошибок.

На основании вышеизложенного, следует считать, что результаты разработки соответствуют требованиям, указанным в техническом задании, поставленная цель достигнута.

Разработанный программный продукт может быть применен для сбора информации с различных сайтов источников литературы.

Список информационных источников

1. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: межгосударственный стандарт: утв. и введ. в действие Постановлением Государственного комитета СССР по стандартам от 24 марта 1989 г. No. 661: дата введ. 1990-01-01: переиздание июнь 2009 г. / разработан и внесен Государственным комитетом СССР по стандартам, Министерством приборостроения, средств автоматизации и систем управления СССР. – М.: Стандартинформ, 2009. – Текст: электронный // Электронный фонд правовых и нормативно-технических документов [сайт]. – URL: https://docs.cntd.ru/document/1200006924 (дата обращения 20.09.2022);
2. ГОСТ Р 7.0.100-2018. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления: национальный стандарт РФ: утв. и введ. в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 г. No. 1050-ст: дата введ. 2019-07-01: внесена поправка 2020 г. / разработан Федеральным государственным унитарным предприятием «Информационное телеграфное агентство России (ИТАР-ТАСС)», филиал «Российская книжная палата», Федеральным государственным бюджетным учреждением «Российская государственная библиотека», Федеральным государственным бюджетным учреждением «Российская национальная библиотека»: внесен Техническим комитетом по стандартизации ТК 191 «Научно-техническая информация, библиотечное и издательское дело». – Текст: электронный // Электронный фонд правовых и нормативно-технических документов [сайт]. – URL: https://docs.cntd.ru/document/1200161674 (дата обращения 29.10.2022);

3. Хохлова, Ю. Глоссарий по информационному обществу / Хохлова Ю.Е.,

Бунчук М.А. // Институт развития информационного общества. - 2009. - 160 с. (дата обращения 25.09.2022);
4. Портал: Компьютерные технологии. [сайт]. – URL: http://ru.wikipedia.org/wiki/Портал:Компьютерные_технологии (дата обращения 30.09.2022);
5. Википедия. Selenium. [сайт]. – URL: https://ru.wikipedia.org/wiki/Selenium/(дата обращения 30.09.2022);
6. Суханов, А.А., Маратканов, А.С. Анализ способов сбора социальных данных из сети интернет / Суханов, А.А., Маратканов, А.С. //International Scientific Review. -2017. -Вып. 1. С.22-25 (дата обращения 30.09.2022);
7. Парсинг html-сайтов с помощью PHP, Ruby. [сайт]. – URL: http://parsing.valemak.com/ru/what-why-how/stages-of-parsing/ (дата обращения 07.10.2022);
8. 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. (дата обращения 07.10.2022);
9. Википедия. Python. [сайт]. – URL: https://ru.wikipedia.org/wiki/Python (дата обращения 09.10.2022);
10. Погода в «Яндексе»: от виджета до собственной технологии. [сайт]. – URL:https://vc.ru/yandex/302991-pogoda-v-yandekse-ot-vidzheta-do-sobstvennoy-tehnologii (дата обращения 22.10.2022);
11. Requests: HTTP for Humans. [сайт]. – URL: http://docs.python-requests.org/en/master/# (дата обращения 22.10.2022);
12. Википедия. Технология. [сайт]. – URL: https://ru.wikipedia.org/wiki/Технология (дата обращения 22.09.2022);

13. Цхошвили Д.З., Иванова Н.А. Примеры Использования Технологии Парсинга/ Цхошвили Д.З., Иванова Н.А. // Статья в сборнике трудов конференции Язык:русский Год издания. 2017. С. 135-140 (дата обращения 15.10.2022);
14. lxml - XML and HTML with Python. [сайт]. – URL: http://lxml.de/index.html (дата обращения 10.01.2023);
15. Python. [сайт]. – URL: https://ru.wikipedia.org/wiki/Python (дата обращения 09.01.2023);
16. Топ-5 форматов для сбора данных с парсинга. [сайт]. – URL: https://markedata.io/kak-vybrat-format-polucheniya-dannyh-parsinga-dlya-internet-magazina/ (дата обращения 15.01.2023);
17. Статьи по написанию парсера на Python. [сайт]. – URL: https://parsemachine.com/articles (дата обращения 17.01.2023);
18. Составление технического задания (ТЗ) на разработку парсера. [сайт]. – URL: http://parsing-and-i.blogspot.com/2010/02/tehnicheskoe-zadanie-na-parser.html (дата обращения 29.12.2022);
19. Извлечение данных с Web-страниц с помощью кода на языке Python. [сайт]. – URL: https://baguzin.ru/wp/izvlechenie-dannyh-s-web-stranits-s-pomoshh/ (дата обращения 20.01.2023);
20. Интернет: особенности и возможности. [сайт]. – URL: https://studme.org/50396/menedzhment/internet_osobennosti_vozmozhnosti (дата обращения 16.12.2022);
21. Саркисян А.А. Влияние информационных технологий на жизнь человека в современных условиях // Молодежный научный форум: Технические и математические науки: электр. сб. ст. по мат. XXVI междунар. студ. науч.- практ. конф. № 7 (26);
22. Пасинг. Что это такое и где используется. [сайт]. – URL: https://ipipe.ru/info/parsing (дата обращения 29.01.2023);
23. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб. пособие / С. Ю. Золотов ; Томский гос.

ун-т систем управления и радиоэлектроники. - Томск : Эль Учебное пособие Контент, 2013. - 86 с. - ISBN 978-5-4332-0083-8. 69.

24. Строки. Функции и методы строк. [сайт]. – URL: https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html (дата обращения 03.02.2023);
25. Массив в Python [сайт]. – URL: https://pythonist.ru/massiv-v-python (дата обращения 30.01.2023);
26. Веб-аналитика для решения нестандартных задач [сайт]. – URL: https://www.seonews.ru/reviews/imetrics-2012-web-analitika-dlya-resheniya-nestandartnih-zadach/ (дата обращения 31.01.2023).
27.

Приложения

Приложение 1

HTML форма для получения данных от пользователя

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Начальная форма</title>
<style>
body {background-color: #999; }
</style>
</head>
<body>
<center>
<table>
<h3>Форма для взаимодействия</h3>
    <form action="000.php" method="POST" enctype="multipart/form-data">
<tr><td>Имя таблицы</td><td><input name="table_name" type="text"></td></tr> <tr><td>Файл</td><td><input type="file" name="file_name"></td></tr> <tr><td></td><td><input type="submit" value="Загрузить"></td></tr>
</form>
</table>
 <h5>!!!Внимание файл должен быть в кодировке UTF-8!!!</h5>
</center>
</body>
</html>
```

Приложение 2

Программа преобразования данных из текстового файла в базу

```
<?php
mb_internal_encoding("UTF-8");
if(!empty($_POST['table_name']))
{
        if (isset($_FILES['file_name']) && $_FILES['file_name']['error'] === UPLOAD_ERR_OK)
{
$db_host = "localhost";
$db_name = "info";
$db_user = "root";
$db_pass = "root";
$con = @mysqli_connect($db_host,$db_user,$db_pass, $db_name)
 or die("Ошибка при подключении к базе данных -> ".mysqli_connect_error());
$table_name = $_POST['table_name'];
$zapros0 = "
CREATE TABLE $table_name
(
id_statia INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
name_statia TEXT,
fio_avtor TEXT,
ukazatel TEXT,
text TEXT,
gorod TEXT,
goda TEXT,
straniz TEXT,
prim TEXT
);";
```

```
$itog0 = mysqli_query($con, $zapros0);
$file_name = $_FILES['file_name']['tmp_name'];
//echo $file_name;
$text_arr = file("$file_name");
//print_r($text_arr);
for($i=0; $i<count($text_arr); $i++)
$name = "";
$avtor = "";
$ukazatel = "";
$text = "";
$gorod = "";
$god = "";
$stroka = "";
$prim = "";
if (preg_match('/^[0-9]+\./', $text_arr[$i]))
{
if (preg_match('/^\[(.*)\]/', $text_arr[$i+1]))
{
$prim = $text_arr[$i+1];
$name = $text_arr[$i+2];
$subject2 = preg_split("/[\s.]+/", $text_arr[$i+3]);
if (preg_match('/^[A-Я]+$/', $subject2[1]) && preg_match('/^[A-Я]+$/',
$subject2[2]))
{
$avtor = $text_arr[$i+3];
$str = $text_arr[$i+4];
}
Else
{
$str = $text_arr[$i+3];
```

```
}
$keyw = preg_split("/[\s,]+/", $str);
for($j=0; $j<count($keyw); $j++)
{
$keyw1 = $keyw[$j]." ".$keyw[$j+1];
if (preg_match('/В книге:/', $keyw1) || preg_match('/В сборнике:/', $keyw1))
{
$ukazatel = preg_replace('/:/',",$keyw1);
}
else
{
if (preg_match('/Свидетельство о/', $keyw1))
{
$ukazatel = "Свидетельство";
}
}
if (preg_match('/\/+$/', $keyw[$j]))
{
$gorod = preg_replace('/\./',",$keyw[$j+1]);
}
$keyw2 = preg_replace('/\./',",$keyw[$j]);
if (preg_match('/^[0-9]{4}+$/', $keyw2))
{
$god = $keyw2;
}
if (preg_match('/C\.+$/', $keyw[$j]))
{
$stroka = preg_replace('/\./',",$keyw[$j+1]);
}
}
```

```
$text = strstr($str, '.', true);
$text = preg_replace('/В сборнике: /','',$text);
text = preg_replace('/В книге: /','',$text);
}
else
{
$name = $text_arr[$i+1];
$subject2 = preg_split('/[\s.]+/', $text_arr[$i+2]);
if (preg_match('/^[А-Я]+$/', $subject2[1]) && preg_match('/^[А-Я]+$/', $subject2[2]))
{
$avtor = $text_arr[$i+2];
$str = $text_arr[$i+3];
}
else
{
$str = $text_arr[$i+2];
}
$keyw = preg_split('/[\s,]+/', $str);
for($j=0; $j<count($keyw); $j++)
{
$keyw1 = $keyw[$j]." ".$keyw[$j+1];
if (preg_match('/В книге:/', $keyw1) || preg_match('/В сборнике:/', $keyw1))
{
$ukazatel = preg_replace('/:/','',$keyw1);
}
else
{
if (preg_match('/Свидетельство о:/', $keyw1))
{
```

```
$ukazatel = "Свидетельство";
}
}
if (preg_match('/\/+$/', $keyw[$j]))
{
$gorod = preg_replace('/\./',",$keyw[$j+1]);
}
$keyw2 = preg_replace('/\./',",$keyw[$j]);
if (preg_match('/^[0-9]{4}+$/', $keyw2))
{
$god = $keyw2;
}
if (preg_match('/C\.+$/', $keyw[$j]))
{
$stroka = preg_replace('/\./',",$keyw[$j+1]);
}
}
$text = strstr($str, '.', true);
$text = preg_replace('/В сборнике: /',",$text);
$text = preg_replace('/В книге: /',",$text);
}
/*
echo "<br>".$name."<br>";
echo $avtor."<br>";
echo $ukazatel."<br>";
echo $text."<br>";
echo $gorod."<br>";
echo $god."<br>";
echo $stroka."<br>";
echo $prim."<br>";
```

```
*/
$zapros = "INSERT INTO `$table_name`
 (`name_statia`,`fio_avtor`,`ukazatel`,`text`,`gorod`,`goda`,`straniz`,`prim`)
VALUES
('$name','$avtor','$ukazatel','$text','$gorod','$god','$stroka','$prim');    ";
$itog = mysqli_query($con, $zapros);
}
}
echo "Таблица успешно создана. Данные из файла добавлены.";
mysqli_close($con);
}
else
{
echo "Произошла ошибка при загрузке файла";
}
}
else
{
echo "Вы не указали имя таблицы";
}
?>
```