МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра прикладной математики**

**КУРСОВАЯ РАБОТА**

**ПРИМЕНЕНИЕ ТЕХНОЛОГИИ СИНТАКСИЧЕСКОГО АНАЛИЗА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.М. Баутин

(подпись)

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) Технология программирования

Научный руководитель

канд. физ.-мат. наук, доц \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Калайдина

(подпись)

Нормоконтролер

канд. физ.-мат. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Калайдина

(подпись)

Краснодар

2018

**РЕФЕРАТ**

Курсовая работа содержит 26 страниц, 5 рисунков, 10 источников.

Ключевые слова:

ВЕБ-СКРЕЙПИНГ, КРАУЛИНГ, ПАРСИНГ, БАЗЫ ДАННЫХ, СБОР ИНФОРМАЦИИ

Цель курсовой работы – исследовать основные способы веб-скрейпинга для поиска информации и реализовать соответствующее приложение.

В курсовой работе изучены принципы технологии веб-скрейпинга, понятие поискового робота и способы применения синтаксического анализа.

В курсовой работе разработана программа, позволяющая получить всю необходимую информацию с новостного сайта lenta.ru и сохранить эти данные в реляционной базе данных.

Реализация выполнена на языке программирования Python с использованием библиотеки Scrapy.

СОДЕРЖАНИЕ

[Введение 4](#_Toc516320551)

[1 Основополагающие принципы технологии веб-скрейпинга 5](#_Toc516320552)

[1.1 Технологии веб-скрейпинга 5](#_Toc516320553)

[1.2 Краулинг 7](#_Toc516320553)

[1.2.1 Основные сведения о поисковых роботах 7](#_Toc516320554)

[1.2.2 Возможности библиотеки Scrapy 8](#_Toc516320555)

[1.2.3 Robots.txt 10](#_Toc516320555)

[1.3 Парсинг 12](#_Toc516320554)

[1.4 Способы хранения информации. 13](#_Toc516320554)

[1.4.1 Реляционная база данных 13](#_Toc516320554)

[1.4.2 Формат JSON 14](#_Toc516320554)

[2 Современные инструменты для веб-скрейпинга 16](#_Toc516320556)

[3 Ограничения на использование скрейперов 18](#_Toc516320570)

[4 Создание парсера на основе библиотеки Scrapy 19](#_Toc516320570)

[4.1 Проектирование приложения-парсера 19](#_Toc516320554)

[4.2 Программная реализация 21](#_Toc516320554)

[Заключение 25](#_Toc516320573)

[Список использованных источников 26](#_Toc516320574)

# ВВЕДЕНИЕ

Сегодня, в эпоху бурного развития информационных технологий, знания являются главными факторами успеха в обществе. Благодаря сети Интернет возможность овладеть знаниями есть у каждого. Хотя, формально Интернет – это глобальная сеть компьютерных ресурсов с коллективным доступом на основе использования единой стандартной схемы адресации, высокопроизводительной магистрали и высокоскоростных линий связи с главными сетевыми компьютерами. На сегодняшний день, Интернет нечто большее, чем просто способ коммуникации между людьми, это ещё и самый большой источник информации в мире. И этот источник растёт на глазах, ведь каждый день создаются многочисленные веб-сайты и интернет-порталы, где располагается важный и интересный контент. Однако овладеть таким количеством информации человеку не под силу, и тут на помощь приходит компьютер, в частности технология веб-скрейпинга.

Веб-скрейпинг – сравнительно недавнее изобретение, призванное значительно упростить жизнь всем, кто, так или иначе, сталкивается с необходимостью сбора данных в Интернете. Скрейпинг – это технология, использующая скрипты для захода на сайт под видом обычного пользователя и собирающая информацию по заранее установленным параметрам. Таким образом, можно получать, обрабатывать, систематизировать и сохранять в обычном текстовом формате данные тысяч веб-страниц за считанные минуты. Эта технология пользуется особым спросом в журналистике и статистике, что поддерживает её актуальность.

Целью курсовой работы является изучение технологии веб-скрейпинга на основе библиотеки Python Scrapy и поиск важной и актуальной информации.

Итог проделанной работы – создание приложения с использованием технологии веб-скрейпинга.

## 1 Основополагающие принципы технологии веб-скрейпинга

### Технологии веб-скрейпинга

Веб-скрейпинг – технология для синтаксического преобразования HTML-страниц в более удобные для потребления формы [1]. Технология веб-скрейпинга является узконаправленной частью более общего понятия web mining (WM).

WM – программное обеспечение, предназначенное для извлечения знаний из данных, обычно из документов и сопровождающих их гиперссылок, из сведений о пользователях и их активности, зафиксированных в различных журналах, в общем, из всего, что может быть доступно в Сети [5]. Первые публикации на тему WM датируются серединой 90-х годов, когда мирно сосуществовали два подхода: процессный, представляющий WM как последовательность выполняемых задач, и дата-центричный, привязанный к типам данных. В дальнейшем преимущественное распространение получил второй. Сегодня под WM понимают совокупность методов автоматического систематического обхода сети интернет с целью сбора требуемых данных о компаниях и людях для выработки информации и принятия решений.

Дата-центричность означает акцент WM на работе с тремя источниками информации, каждому из которых соответствует вид WM:

* Данные о действиях пользователей, от журналов серверов до отслеживания обращений к браузеру. Web Usage Mining(WUM) или нагрузочный WM.
* Веб-графы, описывающие прямые связи между страницами WWW. В веб-графе вершины – это страницы WWW, а дуги – гиперссылки между ними. По графу устанавливаются связи между страницами, людьми и любыми иными объектами. WCM стал областью активных исследований, и основные сложности здесь вызваны гетерогенностью веб-данных и их низкой структуризацией, затрудняющей выделение целевой информации. Кроме того, в WCM необходимо решать ряд специфических задач:
* извлечение структурированных данных из веб-страниц с использованием методов машинного обучения и нейронных сетей;
* формирование процедур унификации форматов представления данных и их интеграции из разных источников;
* выделение оценок продуктов и услуг в отзывах, размещаемых на форумах, в блогах и чатах.

В WCM почти всегда выполняется процедура перевода данных из формы, предназначенной для чтения человеком, в форму, удобную для обработки компьютером. Такая процедура называется data scraping. Спустя некоторое время эта технология модифицировалась и стала называться web scraping.

Полностью автоматизированная генерация возможна пока только на экспериментальном уровне, а высшим достижением веб-скрейпинга являются анализаторы веб-страниц с элементами искусственного интеллекта на базе систем компьютерного зрения и машинного обучения.

Веб-скрейпинг работает с тремя типами информации: структурированными, неструктурированными и квазиструктурированными.

Для работы со структурированными данными достаточно применить служебные процедуры сначала обхода страниц, затем генерации и исполнения упаковщика, а потом можно переходить к анализу содержимого страницы.

Сложнее с квазиструктурированными данными. Примером таких данных может быть граф посещений сайта. Для работы с квазиструктурированными данными предложены специальные языки, такие как NEXIR ELOG, предназначенные для программ-упаковщиков. С их помощью описывается процедура выделения данных, завершающаяся созданием объектной модели данных (OEM).

Неструктурированный DM призван облегчить восприятие пользователям больших массивов текстов. Выделяют несколько типов такого рода операций:

* отслеживание тематики – оценка области интересов пользователя;
* свертка – создание резюме документов;
* ранжирование – упорядочение документов и их распределение по заранее определенным категориям;
* кластеризация – объединение схожих документов в группы;
* визуализация данных – решение проблемы коммуникации пользователя с данными.

Около дюжины компаний производят сегодня инструменты для WCM в виде традиционных загружаемых коммерческих и свободно-распространяемых программ и облачных сервисов. Однако все инструменты имеют некоторую похожую структуру.

Так, любой инструмент для веб-скрейпинга включает в себя следующие элементы:

* краулинг;
* парсинг;
* хранение информации.

Рассмотрим все стадии веб-скрейпинга подробнее.

**1.2 Краулинг**

1.2.1 Основные сведения о поисковых роботах

Поисковый робот или краулер – это программа, реализующая технологию краулинга и разработанная для перехода по веб-страницам и для занесения информации о них в базу данных [2]. Краулинг позволяет автоматизировать такие функции, как сбор особых видов информации, например, адресов электронной почты.

Функционирование поискового робота строится по тем же принципам, по которым работает браузер. Паук посещает сайт, оценивает содержимое страницы, переносит их в базу поисковой системы, затем по ссылкам переходит на другие ресурсы, повторяя один и тот же алгоритм действий. Результат этих поисков – перебор веб-ресурсов в строгой последовательности. Попадая на веб-ресурс, паук находит предназначенный для него файл robots.txt, и дальнейшие поиски должен проводить согласно правилам, установленным в robots.txt.

При использовании технологии краулинга необходимо задавать следующие характеристики:

* глубина – указывает, на какое количество кликов от стартовой страницы вы хотите погрузиться;
* количество параллельных загрузок страниц;
* пауза между страницами для облегчения нагрузки на сервер, не редко владелец ресурса определяет значение этого параметра в файле robots.txt;

Краулер не должен сильно нагружать сервер сайта, иначе такой краулинг воспринимается как кибер атака и приводит к блокированию и сайта, и краулера [6].

1.2.2 Возможности библиотеки Scrapy

Scrapy – это прикладная среда для сканирования веб-сайтов и извлечения структурированных данных, которая может использоваться для широкого спектра полезных приложений, таких как анализ данных, обработка информации или архивирование данных [7].

Несмотря на то, что Scrapy изначально был разработан для просмотра веб-страниц, его также можно использовать для извлечения данных с помощью API (таких как Amazon Associates Web Services ) или в качестве веб-сканера общего назначения.

Scrapy предоставляет множество мощных функций для легкого и эффективного очищения, таких как:

– Встроенная поддержка выбора и извлечения данных из источников HTML / XML с использованием расширенных селекторов CSS и выражений XPath, а также вспомогательные методы с использованием регулярных выражений;

– Интерактивная оболочка консоли IPython для обработки CSS и XPath выражений;

– Встроенная поддержка генерации экспорта фидов в нескольких форматах (JSON, CSV, XML) и их хранения в нескольких бэкэндах (FTP, S3, локальная файловая система);

– Надежная поддержка кодирования и автоопределения для работы с внешними, нестандартными и неработающими объявлениями кодировки;

– Поддержка расширяемости, позволяющая вам подключать свои собственные функции, используя сигналы и четко определенный API (связующее ПО, расширения и конвейеры);

– Широкий спектр встроенных расширений и промежуточного программного обеспечения для обработки:

а) cookies и обработка сессии;

б) функции HTTP, такие как сжатие, аутентификация, кэширование;

в) подмена пользовательского агента;

г) robots.txt;

д) ограничение глубины сканирования.

Scraping состоит из двух этапов:

– Систематический поиск и загрузка веб-страниц;

– Извлечение данных с веб-страниц.

Создать поискового робота с нуля можно с помощью различных модулей и библиотек, которые предоставляет язык программирования, однако в дальнейшем – по мере роста программы – это может вызвать ряд проблем. К примеру, вам понадобится переформатировать извлечённые данные в CSV, XML или JSON. Также вы можете столкнуться с сайтами, для анализа которых необходимы специальные настройки и модели доступа.

Поэтому лучше сразу разработать робота на основе библиотеки, которая устраняет все эти потенциальные проблемы. Для этого используются Python и Scrapy.

Scrapy – одна из наиболее популярных и производительных библиотек Python для получения данных с веб-страниц, которая включает в себя большинство общих функциональных возможностей. Это значит, что вам не придётся самостоятельно прописывать многие функции. Scrapy позволяет быстро и без труда создать «веб-паука».

Пакет Scrapy (как и большинство других пакетов Python) можно найти в PyPI (Python Package Index, также известен как pip) – это поддерживаемый сообществом репозиторий для всех вышедших пакетов Python.

Поисковые роботы позволяют извлечь информацию о ряде продуктов, получить большой объём текстовых или количественных данных, извлечь данные с сайта без официального API и многое другое.

1.2.3 Robots.txt

Файл *robots.txt* – это текстовый файл, находящийся в корневой директории сайта, в котором записываются специальные инструкции для поисковых роботов, например, параметры индексирования сайта [5]. Эти инструкции могут запрещать к индексации некоторые разделы или страницы на сайте, рекомендовать поисковому роботу соблюдать определенный временной интервал между скачиванием документов с сервера, блокировать неважные изображения и скрипты. После изучения файла, робот посещает главную страницу, а с нее переходит по ссылкам, продвигаясь в глубину.

Под индексацией понимается анализ и занесение в базу поисковых систем страниц сайта, после которого сайт выдаётся в результатах поисковых запросов.

В первую очередь, при работе с robots.txt, роботы проверяют наличие записей, начинающихся с User-agent, такие записи заключают между собой инструкции для определённого робота. Однако существуют поисковые роботы, которые не прислушиваются к общим запрещающим правилам в robots.txt, поэтому требуется прописывать блоки команд для таких роботов отдельно.

После обращения к роботу, с помощью директивы user-agent, следуют команды, запрещающие или разрешающие обращение к определённым разделам сайта. Для запрета к просмотру используется директива Disallow, а для разрешения Allow, параметрами, которых является URL раздела или страницы сайта.

В основном директива Disallow применяется для запрета посещения пауком служебных разделов, конфиденциальной информации, а также для удаления дублей страниц. Под дублями страниц понимается исключение обращения к одной и той же странице по разным адресам.

Менее востребованная директива Allow в большей степени предназначается для роботов поисковых систем, она помогает правильно проиндексировать сайт. Используя директиву allow, вы обозначаете роботу страницы необходимые к индексации. Allow реже используется по сравнению с disallow, это связано с тем, что по умолчанию все страницы сайта должны индексироваться.

Очень важной и полезной директивой в robots.txt, при рассмотрении темы веб-скрейпинга, является crawl-delay, с её помощью владелец сайта может задать минимальный промежуток времени между окончанием загрузки одной страницы и началом загрузки следующей. Crawl-delay используется, когда сервер перегружен и не успевает обрабатывать запросы на загрузку страниц.

Так же существуют директивы host – указывает основное зеркало сайта, и sitemap – указывает URL карты сайта, которая отображает структуру построения сайта и расположение контента на нём. Зеркалом сайта называется его полный или частичный дубликат.

**1.3 Парсинг**

В общем смысле, парсинг – это линейное сопоставление последовательности слов с правилами языка. Для этого создается математическая модель сопоставления лексем с формальной грамматикой, описанная языком программирования [8].

Парсер – это программа, созданная для автоматизации постобработки полученной c веб-сайта информации. Другими словами, парсер – это программа по анализу и преобразованию текста с целью выделить из него определённые фрагменты или удалить лишнее. В отличие от человека, парсер быстро и безошибочно отделяет нужную и отбрасывает лишнюю информацию из html документа, и эффективно упаковывает результат в определённом формате.

Существует несколько методов извлечения данных:

* Анализ DOM дерева html страницы. DOM – это представление HTML/XML документа в виде дерева объектов, которое позволяет скриптам изменять содержимое и структуру документа. Данные в таком подходе получаются по атрибуту элемента дерева или при отсутствии таковых, спускаясь вниз по DOM дереву. Полученные данные могут быть любой структуры, а для получения значения элемента, достаточно знать его расположение. Однако скрипт, использующий данный метод нужно привязывать к движку, а при изменении расположения элемента теряется доступ к нему;
* Парсинг строк. Этот способ парсинга имеет узкую область применения, так как получение данных происходит путём парсинга отдельных строк, что, в свою очередь, возможно только в случае чётко фиксированного формата данных;
* Использование регулярных выражений. Этот метод, в основном, используют для решения небольших задач или для написания собственных процедур;
* XML парсинг. Еще одним подходом является рассмотрение HTML как XML данные. Причина в том, что HTML редко бывает валидным, под валидностью HTML, в данном случае, понимается соответствие XML стандартам. Библиотеки, реализовавшие такой подход, больше времени уделяли преобразованию HTML в XML, чем непосредственно парсингу данных;
* Визуальный подход. Суть подхода в том, чтобы пользователь мог без использования программного языка или API настроить систему для получения нужных данных любой сложности и вложенности. Однако такой подход находится на начальной стадии развития.

Проблемы при парсинге HTML данных:

* использование JavaScript / AJAX / асинхронных загрузок элементов страниц очень усложняют написание парсеров;
* различные движки для рендеринга HTML могут выдавать разные DOM деревья;
* большие объемы данных требуют писать распределенные парсеры, что влечет за собой дополнительные затраты на синхронизацию.

**1.4 Способы хранения информации**

1.4.1 Реляционная база данных

Реляционная база данных – это модель [базы данных,](https://techterms.com/definition/database) которая хранит данные в [таблицах](https://techterms.com/definition/table). Подавляющее большинство баз данных, используемых в современных приложениях, являются реляционными [10]. Аналогичным образом, большинство систем управления базами данных являются системами управления реляционными базами данных. Другие модели баз данных включают в себя [простые файловые](https://techterms.com/definition/flatfile) и иерархические базы данных, хотя они используются редко.

Каждая таблица в реляционной базе данных содержит [строки](https://techterms.com/definition/row) и [столбцы](https://techterms.com/definition/column). Таблицу можно представить в виде [матрицы](https://techterms.com/definition/matrix) строк и столбцов, где каждое пересечение строки и столбца содержит определенное значение.

Таблицы базы данных часто включают [первичный ключ](https://techterms.com/definition/primarykey), который предоставляет уникальный идентификатор для каждой строки в таблице. Ключ может быть назначен столбцу или он может состоять из нескольких столбцов, которые вместе образуют уникальную комбинацию значений. В любом случае первичный ключ обеспечивает эффективный способ [индексации](https://techterms.com/definition/index) данных и может использоваться для обмена значениями между таблицами в базе данных. Например, значение первичного ключа из одной таблицы может быть присвоено полю в строке другой таблицы. Значения, импортированные из других таблиц, называются внешними ключами.

Стандартный способ доступа к данным из реляционной базы данных – запрос [SQL](https://techterms.com/definition/sql). SQL-запросы могут использоваться для создания, изменения и удаления таблиц, а также для выбора, вставки и удаления данных из существующих таблиц.

1.4.2 Формат JSON

Формат данных не имеет каких-либо ограничений, но, чаще всего, используется JSON. JSON – текстовый формат, полностью независимый от языка реализации, признан простым способом хранить и передавать структурированные данные [9]. Несмотря на происхождение от JavaScript, данный формат может использоваться практически с любым языком программирования.

Почему его используют:

* компактен;
* предложения легко читаются и составляются как человеком, так и компьютером;
* легко преобразовать в структуру данных для большинства языков программирования (числа, строки, логические переменные, массивы и так далее);
* чтение и создание структур JSON возможно осуществлять с помощью большинства языков программирования;
* на его основе можно собрать базу данных, например, используя утилиту jl-sql, которая принимает на вход поток JSON-объектов, разделённых символом новой строки, и помещает в базу данных, используя SQL запросы. SQL – формальный непроцедурный язык программирования применяемый в реляционной базе данных, управляемой соответствующей системой управления базами данных [8];
* Наиболее частое распространенное использование JSON – пересылка данных по сети. Так, например, выполнив скрипт скрейпера на сервере и получив результат в формате JSON, его можно с лёгкостью отправить заказчику по интернету для дальнейшего использования.

Файл JSON имеет два способа представления как набор пар ключ-значение или как упорядоченный набор значений, где способ реализации представлений зависит от языка реализации.

Ещё одной сильной чертой JSON формата является его встроенность в большинство современных браузеров, что делает использование формата удобным и ускоряет обработку JSON данных.

**2 Современные инструменты для веб-скрейпинга**

Рассмотрение темы веб-скрейпинга невозможно без изучения положения дел на рынке готового ПО. К сожалению выбор инструментов не так широк, как хотелось бы, к тому же большинство инструментов предназначено для англоязычных пользователей, что делает затруднительным выбор программы для работы, однако всё же есть достойные предложения.

Для начала следует разделить две основные категории продуктов. К первой категории относятся продукты, позволяющие создать на своей базе полноценный скрейпер. Такие программы чаще всего являются фреймворками или расширениями для браузеров, реже веб-сервисами. В таких продуктах заранее определён язык программирования и существует ряд ограничений. Плюсами таких решений, является возможность использования различных библиотек и самое главное, качество скрейпинга зависит только от пользователя. Ко второй категории можно отнести полноценные программы, позволяющие пользователям скрейпить данные не прибегая к программированию. Существуют разные форматы таких программ. Веб-скрейпер может представлять из себя как простое расширение в браузере, так и онлайн сервис. Рассмотрим ведущие продукты для веб-скрейпинга.

Отличным ПО для веб-скрейпинга, на сегодня, является OutWit Hub от компании OutWit Technologies. Данная программа является расширением браузера Firefox. Такая программа может служить для скрейпинга небольших объёмов данных, скорее для домашнего использования. Программа является самой простой для использования среди своих конкурентов, однако даже её использование невозможно без знаний основ веб-программирования. В данной программе пользователь играет важную роль, так как он задаёт тэги, в которых заключена искомая информация, определяет критерии поиска и виды ссылок, по которым будет проходить скрейпер. Также пользователь определяет формат, в котором сохранится найденная информация. Для качественного скрейпинга необходимо приложить немало усилий в настройке программы.

Конкурентом, рассмотренного программного обеспечения, является продукт под названием OpenRefine от одноимённой компании. OpenRefine очень популярен среди журналистов и статистиков. О простоте использования говорить не приходится, поэтому понадобится не один день ознакомления. Понимая то, что основными пользователями продукта являются не программисты с опытом веб-разработки, компания решила выпустить полноценную книгу, являющуюся своеобразным гидом по программе. Ещё одним преимуществом инструмента является множество расширений, которые значительно увеличивают возможности программы. Возможно, эти факторы стали ключевыми в борьбе за столь большую аудиторию.

Для решения задач, связанных со скрейпингом данных нет универсального инструмента. Необходимо предварительно изучить возможности предлагаемых скрейперов и, сделав выбор инструмента, придётся потратить много времени на его освоение. Программы для веб-скрейпинга хоть и позволяют уйти от чистого программирования, но всё-таки мало пригодны для среднестатистических пользователей.

**3 Ограничения на использование скрейперов**

Веб-скрейпинг – это извлечение контента с веб-сайтов. В свою очередь у любого сайта есть свой правообладатель или администратор, который устанавливает правила пользования ресурсом. Поэтому необходимо взаимодействовать с владельцем сайта или контента прежде чем запускать скрейпер. У популярных сайтов сотни тысяч посетителей за сутки, поэтому невозможно каждому пользователю, желающему скачать информацию, связаться напрямую с владельцем домена. Однако пренебрегать правами автора нельзя, ведь нарушение авторских прав может влечь гражданско-правовую, административную и/или уголовную ответственность.

Владельцы сайтов, заинтересованные в защите своих прав, оповещают пользователей в следующем формате: знак копирайта © + имя правообладателя (автора) + год первого опубликования. Однако возможны и другие форматы. Например, © 2018 YouTube, LLC.

Так же встречаются сайты, на которых имеются Пользовательские соглашения либо заявления о соблюдении конфиденциальности – это, как правило, частные веб-сайты закрытого либо коммерческого характера. Такие сайты публикуют весьма объемные правила пользования своим ресурсом. Они оговаривают доступ к сайту, порядок его использования, здесь содержатся многочисленные предупреждения, цель которых – защита информационных технологий и защита товарного знака, разнообразная защита от копирования и вообще защита сайта в плане авторских прав и охраны интеллектуальной собственности.

Минус данных способов оповещения об ограничении использования контента заключается в том, что при автоматизированном копировании данных, например, с использованием скрейпера, пользователь будет оповещён только после совершения правонарушения, а если быть точнее, во время выполнения парсинга страницы. Для предотвращения таких случаев в корневой директории сайта вебмастера помещают файл robots.txt.

**4 Создание парсера на основе библиотеки Scrapy**

Интернет как наиболее стремительно развивающееся средство коммуникации приобретает все более важную роль в деятельности человека. В этой ситуации является актуальным разработка программных систем, позволяющих отбирать информационные ресурсы сети Интернет в соответствии с запросом пользователя. Эффективность этого процесса могут обеспечить, появившиеся с развитием интеллектуальных технологий, специальные программы, называемые интеллектуальными помощниками.

В рамках курсового проекта поставлена задача о разработке программы-парсера, которая может из простой структуры сайта lenta.ru получить всю нужную информацию, необходимую пользователю.

**4.1 Проектирование приложения-парсера**

Сбор информации в интернете – трудоемкая, рутинная, отнимающая много времени работа. Парсеры, способные в течение суток перебрать большую часть веб-ресурсов в поисках нужной информации, автоматизируют ее.

Наиболее активно в целях парсинга используется всемирная сеть поисковых систем. Роботы собирают информацию и в частных интересах. Данную технологию используют программы автоматической проверки уникальности текстовой информации, быстро сравнивая содержимое сотен веб-страниц с предложенным текстом.

Без программ парсинга владельцам интернет-магазинов, которым требуется множество однотипных описаний товаров, технических характеристик и другого контента, не являющегося интеллектуальной собственностью, было бы трудно вручную заполнять характеристики товаров.

Возможностью автоматического анализа чужого контента для наполнения своего сайта пользуются многие веб-мастера и администраторы сайтов. Это оправдано, если требуется часто изменять контент для представления текущих новостей или другой, быстро меняющейся информации.

Веб-скрейпинг – необходимость для организаторов спам-рассылок по электронной почте или каналам мобильной связи. Для этого им надо запустить бота путешествовать по социальным сетям и собирать телефоны, адреса, явки.

Ну и хозяева некоторых, особенно недавно организованных веб-ресурсов, любят наполнить свой сайт чужим контентом.

Конечно же, парсеры не читают тексты, они всего лишь сравнивают предложенный набор слов с тем, что обнаружили в интернете и действуют по заданной программе.

Рассмотрим функционал программы. Программа анализирует статьи сайта lenta.ru при указании временного интервала по одной из 11 рубрик, таких как, Россия, мир, бывший СССР, экономика, силовые структуры, наука и техника, культура, спорт, интернет и СМИ, путешествия и из жизни. То есть при выборе в программе по какой именно рубрике парсить статьи, программа заполняет таблицу lenta\_ru базы данных parsing\_sites.

Программа-парсер содержит семь модулей, таких как \_init\_, middlewares, pipelines, settings, lenta, db, items.

Рассмотрим некоторые из них поподробнее.

Модуль lenta\_ru состоит из одного класса WebSpider\_Lenta () и четырех функций:

а) log () – метод для удобного вывода результата в консоль;

б) start\_requests () – метод для запросов получения даты публикации, формирования ссылки статьи к пауку;

в) parse () – метод для парсинга статей lenta\_ru;

г) parseView () – метод для приведения записей в консоль к нормальному виду.

Также обратимся к модулю db программы. Он содержит также один класс DBHandler () и четыре функции, такие как, \_init\_ (), DBConnect (), addNews () и log (). Основным методом в этом классе является addNews, так как именно в нём происходит заполнение таблицы lenta\_ru базы данных parse\_sites.

Рассмотрим структуру таблицы lenta\_ru. Она состоит из семи полей, таких как:

– id – идентификатор каждой записи в таблице;

– title – заголовок каждой статьи;

– pubdate – дата публикации статьи;

– text – текст статьи;

– author – источник получения информации;

– url – ссылка на статью;

– rubric – название рубрики.

Разработанное приложение можно расширить функциями, связанными с синтаксическим анализом, такими как, поиск слов каждой статьи или подсчёт предложений всех новостных ресурсов конкретного сайта. Также можно написать интерфейс для приложения или сформировать файл в формате XML и JSON.

**4.2 Программная реализация**

Для реализации проекта приложения-парсера используется язык программирования Python с фреймворком Scrapy.

Выбор фреймворка Scrapy обоснован его открытой архитектурой, то есть возможностью расширения функций пакета собственными модулями.

Особенностью разработки приложения является возможность указания даты публикации статьи на сайте lenta.ru в программе (рис. 1).

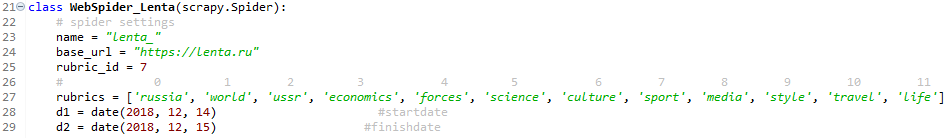


Рисунок 1 – Поля класса WebSpider\_Lenta

Поля класса d1 и d2 устанавливают дату начала и дату конца парсинга веб-ресурса. Время для анализа статей можно выбрать любое, в программе задано с 14 декабря 2018 года по 15 декабря 2018 года соответственно.

Выберем рубрику с rubric\_id = 7, заполним таблицу lenta\_ru базы данных рубрикой «спорт» за данный период (рис. 2).

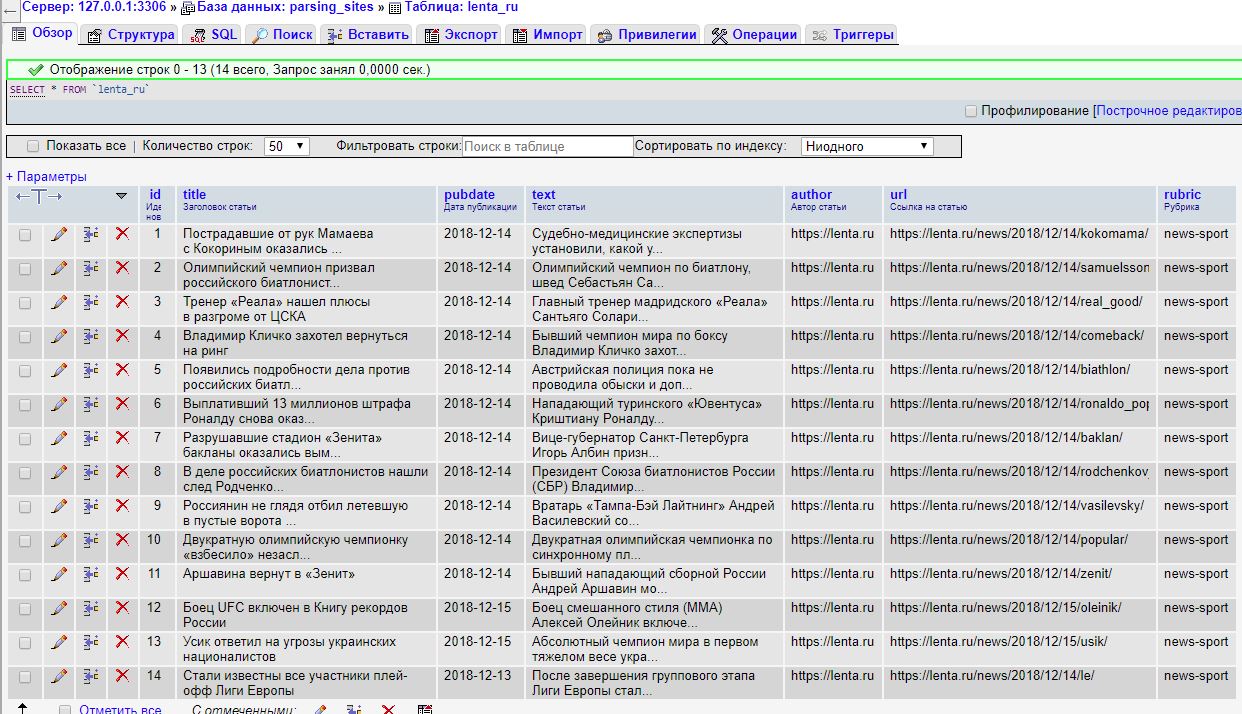


Рисунок 2 – База данных, заполненная рубрикой «спорт»

Рассмотрим результат работы программы (рис. 3).

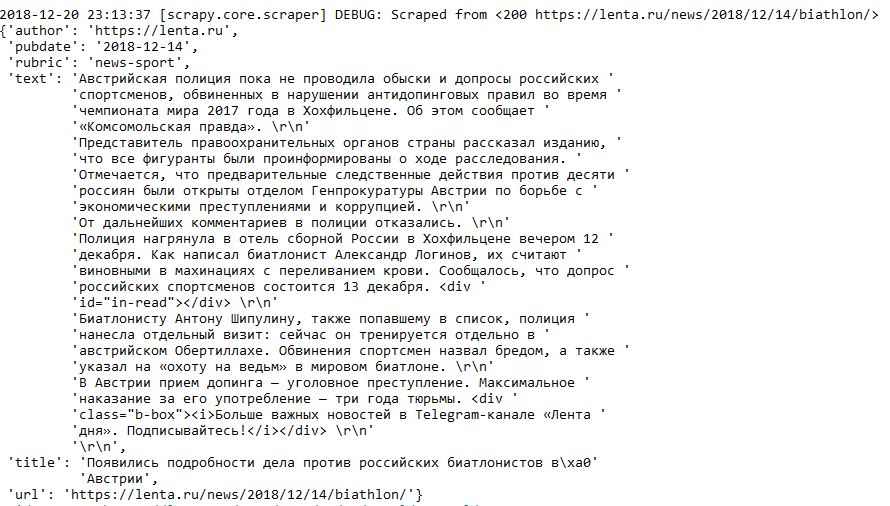


Рисунок 3 – Вывод полученных данных парсером в консоль



Рисунок 4 – Статья про оценку матча «Реал-Цска» из источника lenta.ru

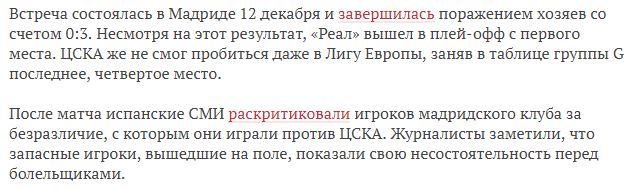


Рисунок 5 – Продолжение статьи про оценку матча «Реал-Цска»

В курсовой работе подробно представлены рисунки примера одной статьи веб-ресурса lenta.ru, однако база данных заполнена 14 статьями рубрики «спорт» за определённый период времени.

**ЗАКЛЮЧЕНИЕ**

Цель курсовой работы – исследовать основные способы веб-скрейпинга для поиска информации и реализовать соответствующее приложение – достигнута.

При подготовке к курсовой работе исследовались вопросы, связанные с изучением и применением технологии веб-скрейпинга, рассмотрены виды поисковых роботов и способы работы синтаксического анализа.

В курсовой работе разработано приложение, позволяющее найти всю нужную информацию с веб-ресурса lenta.ru и записать эти данные в реляционной базе данных простой структуры.

Разработанное приложение можно расширить функциями, связанными с синтаксическим анализом, такими как, поиск слов каждой статьи или подсчёт предложений всех новостных ресурсов конкретного сайта. Также можно написать интерфейс приложения и расширить проект парсингом других веб-ресурсов

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 Козлов А.М. Web scraping при помощи Node.js – URL: <https://habrahabr.ru/post/301426/> [25 ноября 2018].

2 Crawling и scraping. – URL: <https://ru.code-maven.com/adventures-in-crawling-and-scraping-the-world> [29 ноября 2018].

3 Парсинг. Что это? – URL: https://www.ipipe.ru/info/parsing.html   
[1 декабря 2018].

4 Использование robots.txt – URL https://yandex.ru/support/web/robots-txt.xml [4 декабря 2018].

5 Data Mining: учебное пособие. – URL: <https://habr.com/post/348028/>   
[3 декабря 2018].

6 Применение robots.txt – URL https://yandex.ru/methods/use/robots-txt.xml [6 декабря 2018].

7 Scrapy 1.5 documentation – URL <https://doc.scrapy.org/en/latest/>   
[8 декабря 2018].

8 Николаева И., Ландо Т. Прикладная и компьютерная лингвистика. Изд. 2-е. – М.: ЛЕНАНД, 2017. – 320 с.

9 Козлов А.М. Tress. – URL: https://github.com/astur/tress [13 декабря 2018].

10 Реляционные базы данных – URL: <https://yandex.ru/support/data> base [16 декабря 2018].