

При создании автоматизированной информационной системы использовался язык программирования C# с платформой .Net 6. Для реализации выбран фреймворк Blazor в редакции Server. Для взаимодействия с данными в БД используется СУБД MSSQL Server 2017 и Entity-Framework 6.

Разработанная система упростит сбор различного рода информации руководителями отделов УП «ИВЦ Минфина» и предоставление этой информации региональными представителями.

Р. С. Калинин

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. С. А. Лукашевич, ст. преподаватель

ВЕБ-ПАРСИНГ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА PYTHON ДЛЯ СБОРА ДАННЫХ С ЦЕЛЬЮ ОЦЕНКИ ФИНАНСОВОГО СОСТОЯНИЯ ОРГАНИЗАЦИИ

Для оценки финансового состояния организации нам необходимо собрать большой объем информации с веб страниц. Например, получить большое количество бухгалтерских показателей. Самое простое что можно сделать – это скопировать данную информацию и вставить в нашу программу. Однако так как нам необходимо получить большое количество данных для различных организаций то единственный и наилучший выход для этого это прибегнуть к парсингу.

Парсинг это программный сбор информации с различных веб сайтов. Такой метод не требует повторного ввода данных или копипастинга, а это значит, что парсеры экономят время, собирают большие объемы данных и группируют их в нужную форму.

В этой статье мы рассмотрим парсинг с использованием таких библиотек как Beautiful Soup и requests в Python.

Requests – это лицензированная HTTP библиотека Apache2 которая позволяет отправлять HTTP запросы с использованием Python. В этой простой в использовании библиотеке присутствует множество функций, начиная от передачи параметров и заканчивая отправкой пользовательских заголовков и проверкой SSL. Прежде всего, requests это внешний модуль, а значит, перед тем как что-то делать, его нужно будет установить.

Модуль запросов Python имеет несколько встроенных методов для выполнения HTTP-запросов к указанному URL и использованием за-

просов POST, GET, PATCH, PUTM, HEAD. HTTP-запрос предназначен либо для получения данных из указанного URL, либо для передачи данных на сервер. Он действует как протокол запроса-ответа между клиентом и сервером. Для этого будет использоваться GET-запрос.

Метод GET используется для получения информации из определенного ресурса.

```
1 import requests
2 # Creating a get request
3 r = requests.get('https://ru.tradingview.com/symbols/MOEX-GAZP/financials-statistics-and-ratios/')
4 # Checking the response from the server
5 print(r)
6 # Printing the request content
7 print(r.content)
```

Рисунок 1 – Создание GET запроса

```
<Response [200]>
b'\n<!DOCTYPE html>\n<html lang="ru" dir="ltr"\n
class="is-not-authenticated is-not-pro is-not-trial
">\n<head>\n\t<script nonce="2ReuEWB410fGfWugK4EI9w==">var environment
= "battle";\n\t\twindow.WS_HOST_PING_REQUIRED = true;\n\t\twindow
.BUILD_TIME = "2022_03_09-13_00";\n\t\twindow.WEBSOCKET_HOST =
"data.tradingview.com";\n\t\twindow.WEBSOCKET_PRO_HOST = "prodata
.tradingview.com";\n\t\twindow.WEBSOCKET_HOST_FOR_DEEP_BACKTESTING =
```

Рисунок 2 – Вывод

Получив HTML страницы необходимо преобразовать этот необработанный HTML в полезную информацию. В первую очередь в этом поможет библиотека BeautifulSoup. Данная библиотека построена поверх библиотек разбора HTML, таких как html5lib, lxml, html.parser и т.д. Таким образом BeautifulSoup и библиотеку синтаксического анализа можно использовать одновременно.

```
1 import requests
2 from bs4 import BeautifulSoup
3 # Creating a get request
4 r = requests.get('https://ru.tradingview.com/symbols/MOEX-GAZP/financials-statistics-and-ratios/')
5 # Checking the response from the server
6 print(r)
7 # Parsing the HTML
8 soup = BeautifulSoup(r.content, 'html.parser')
9 print(soup.prettify())
```

Рисунок 3 – Разбор содержимого HTML

```
<Response [200]>
<!DOCTYPE html>
<html class="is-not-authenticated is-not-pro is-not-trial" dir="ltr" lang="ru">
<head>
<script nonce="8qxAZVHy+NWM00Wsmn9HoA==">
var environment = "battle";
window.WS_HOST_PING_REQUIRED = true;
window.BUILD_TIME = "2022_03_09-13_00";
window.WEBSOCKET_HOST = "data.tradingview.com";
window.WEBSOCKET_HOST PRO = "prodata.tradingview.com";
```

Рисунок 4 – Вывод

Теперь можно извлекать полезные данные из содержимого HTML. Объект `soup` содержит все данные во вложенной структуре, которые могут быть извлечены программно.

П. В. Клименко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Семченко**, канд. физ.-мат. наук, доцент

КОМПЬЮТЕРНОЕ ОБНАРУЖЕНИЕ И МОРФОЛОГИЧЕСКИЙ АНАЛИЗ НАНОТРУБОК С ИСПОЛЬЗОВАНИЕМ СНИМКОВ СКАНИРУЮЩЕГО ЭЛЕКТРОННОГО МИКРОСКОПА

Анализ изображений – область, изучающая методы извлечения полезной информации из изображений. Анализ изображений тесно связан с их обработкой и включает такие задачи, как поиск объектов заданной формы, выделение границ, улучшение изображений, сегментацию и трекинг, количественное описание областей изображения. Эти задачи встречаются в микробиологии, медицине, астрономии, робототехнике, системах безопасности и слежения, а также других сферах науки и техники, имеющих дело с цифровыми изображениями. До недавнего времени анализ и обработка изображений были невыполнимы в полной мере из-за нехватки вычислительных мощностей. Только значительное усовершенствование компьютеров дало возможность появлению современных алгоритмов обработки, позволяющих в полной мере получить информацию о изображении. Даже сегодня построение качественной программы, работающей с изображениями, одна из самых трудоемких и времязатратных задач в сфере компьютерных наук.

Сканирующая электронная микроскопия высокого разрешения (HR-SEM) – один из самых распространенных инструментов, используемых для исследования морфологии нанотрубок.