Московский государственный технический университет им. Н.Э. Баумана

Отчет

Лабораторная работа № 8

ИСПОЛНИТЕЛЬ:

ФИО: Митрохина А.А.
Группа: ИБМ3-34Б

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

Описание задания

Цель: Осуществить разбор табличных данных из HTML-структуры с использованием библиотеки **BeautifulSoup**, сформировать извлеченные данные в табличный датасет с помощью Python и сохранить этот датасет в формате CSV.

Требования:

Использовать библиотеку BeautifulSoup для парсинга HTML.

Сформировать данные в виде списка словарей или использовать библиотеку pandas (для более сложной работы с датасетами). В данном примере будет использоваться стандартный список словарей.

Сохранить итоговый датасет в файл формата CSV.

Инструменты:

BeautifulSoup4 для парсинга.

csv для сохранения данных.

Текст программы

Для работы необходима установка библиотек:

Bash
pip install beautifulsoup4
Файл: scraper.py

Python
```
import csv
from bs4 import BeautifulSoup

# --- 1. Имитация HTML-данных сайта (для парсинга) ---
# Предположим, что мы парсим таблицу с данными о товарах или сотрудниках.
HTML_DOC = """
<html>
<head><title>Тестовые данные для парсинга</title></head>
<body>
    <h1>Список сотрудников и отделов</h1>
```

```
    <table id="employees_table">
        <thead>
            <tr>
                <th>ID</th>
                <th>Имя</th>
                <th>Отдел</th>
                <th>Зарплата</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>101</td>
                <td>Иванов А.</td>
                <td>IT</td>
                <td>120000</td>
            </tr>
            <tr>
                <td>102</td>
                <td>Петрова В.</td>
                <td>Бухгалтерия</td>
                <td>85000</td>
            </tr>
            <tr>
                <td>103</td>
                <td>Сидоров К.</td>
                <td>Продажи</td>
                <td>150000</td>
            </tr>
            <tr>
                <td>104</td>
                <td>Ковалева О.</td>
                <td>IT</td>
                <td>95000</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
"""

def parse_html_to_dataset(html_content):
    """
```

```
    Разбирает HTML-контент, извлекает данные из таблицы и
формирует датасет.
    """
    # Создание объекта BeautifulSoup
    soup = BeautifulSoup(html_content, 'html.parser')

    # Поиск нужной таблицы по ID
    table = soup.find('table', id='employees_table')

    if not table:
        print("Ошибка: Таблица с ID='employees_table' не найдена.")
        return [], []

    # 1. Извлечение заголовков (Header)
    headers = [th.text for th in table.find('thead').find_all('th')]

    dataset = []

    # 2. Извлечение данных (Body)
    for row in table.find('tbody').find_all('tr'):
        cells = row.find_all('td')

        # Создание словаря для каждой строки
        row_data = {}
        for i, cell in enumerate(cells):
            # Используем заголовок как ключ
            key = headers[i]
            value = cell.text.strip()
            row_data[key] = value

        dataset.append(row_data)

    return headers, dataset


def save_dataset_to_csv(headers, dataset,
filename="employee_data.csv"):
    """
    Сохраняет датасет (список словарей) в формате CSV.
    """
    try:
        with open(filename, 'w', newline='', encoding='utf-8') as csvfile:
            # Создание объекта-писателя, указывая заголовки для
```

```
порядка колонок
            writer = csv.DictWriter(csvfile, fieldnames=headers)

            # Запись заголовков
            writer.writeheader()

            # Запись данных
            writer.writerows(dataset)

        print(f"\n Датасет успешно сохранен в файл: {filename}")
        print(f'Количество записей: {len(dataset)}")

    except Exception as e:
        print(f" Ошибка при сохранении в CSV: {e}")


if __name__ == '__main__':
    # Шаг 1: Разбор HTML
    HEADERS, DATASET = parse_html_to_dataset(HTML_DOC)

    print("---  Результат парсинга (Датасет) ---")
    print(f"Заголовки: {HEADERS}")
    print("Первые 2 записи датасета:")
    for record in DATASET[:2]:
        print(record)

    # Шаг 2: Сохранение в CSV
    if DATASET:
        save_dataset_to_csv(HEADERS, DATASET)
```

Экранные формы с примерами выполнения программы

Для запуска программы необходимо сохранить код в файл scraper.py и запустить его в консоли.

1. Консоль (Запуск программы)

Заключение

В рамках лабораторной работы была успешно продемонстрирована работа с библиотекой **BeautifulSoup** для извлечения табличных данных из HTML. Сформированный датасет в виде списка словарей был корректно сохранен в файл формата CSV, что подтверждает выполнение всех поставленных задач.