Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики управления и технологий

Кузьмина Дарья Юрьевна БД-241м

Программные средства сбора, консолидации и аналитики данных

**Практическая работа 2. Парсинг HTML и консолидация данных**

**Вариант 11**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

доктор экономических наук

Москва

2025

Содержание

[Введение 2](#_Toc211635823)

[Основная часть 3](#_Toc211635824)

[Заключение 6](#_Toc211635825)

## Введение

**Цель**

Освоить методы профессионального парсинга HTML-страниц и консолидации данных из различных источников с последующим проведением аналитического исследования. В рамках работы формируется навык извлечения, очистки и визуализации данных для решения прикладных бизнес-задач.

**Используемые инструменты**

**ПО:** Python 3.x, Google Colab / Jupyter Notebook / любая IDE, Git.  
**Библиотеки:** requests, beautifulsoup4, pandas, matplotlib, seaborn.

**Задачи**

**1. Выбор кейса.**  
Определяется вариант задания, представляющий собой бизнес-сценарий, требующий сбора данных с одной или нескольких веб-страниц.

**2. Разработка парсера.**

* Анализируется HTML-структура целевого сайта с помощью инструментов разработчика браузера.
* Реализуется скрипт на Python с применением requests и BeautifulSoup для извлечения необходимых элементов.
* Программа должна корректно обрабатывать отсутствие данных и поддерживать пагинацию при переходе между страницами.

**3. Консолидация и очистка данных.**

* Извлечённые сведения объединяются в единый датафрейм pandas.
* Выполняется нормализация данных: приведение типов, обработка пропусков, удаление лишних символов.

**4. Аналитическая обработка и визуализация.**

* Проводится исследовательский анализ данных согласно варианту.
* Рассчитываются ключевые метрики, выполняется группировка и выявление закономерностей.
* Результаты визуализируются с использованием matplotlib и seaborn с акцентом на информативность и оформление.

**5. Подготовка итоговых материалов.**

* Формируется отчёт, содержащий описание этапов, выводы и графики.
* Исходный код оформляется как проект в Git-репозитории и публикуется на GitHub / GitVerse.
* В отчёте указывается ссылка на репозиторий, после чего файл загружается в LMS.

## ССЫЛКА НА GIT: <https://github.com/Iezekiss/SoftTools_MGPU>

## Основная часть

### 1. Настройка окружения

Были установлены и импортированы необходимые библиотеки. Создана среда выполнения в Google Colab, импортированы модули requests, BeautifulSoup, pandas, matplotlib, seaborn.

### 2. Первоначальный источник данных

Задание предусматривало использование **раздела “Бестселлеры” на сайте**[**chitai-gorod.ru**](https://www.chitai-gorod.ru/).  
В процессе выполнения возникли технические ограничения:

* сайт возвращал ошибку **403 Forbidden** при обращении из Colab;
* при обходе с эмуляцией браузера через undetected\_chromedriver возникла ошибка Could not determine browser executable, связанная с отсутствием бинарника Chrome в среде;
* также были протестированы альтернативные источники (labirint.ru, ozon.ru, litres.ru, readrate.ru, livelib.ru, fantlab.ru) — все они заблокировали автоматические запросы (5 сайтов подряд недоступны по 403 или 404).

Таким образом, доступ к **7 сайтам** получить не удалось даже при использовании разных библиотек (requests, selenium, fake\_useragent).

## ****Обоснование смены источника****

В связи с тем, что большинство отечественных книжных платформ реализуют **анти-бот-фильтры** и **ограничения по геолокации**, было принято решение сменить источник на открытый международный сайт-песочницу [**Books to Scrape**](https://books.toscrape.com/).  
Сайт имитирует реальную торговую площадку и специально предназначен для учебных целей — полностью совместим с BeautifulSoup и не требует авторизации.

Это изменение позволило:

* корректно реализовать парсинг страниц, извлечь название, цену и рейтинг книг;
* выполнить все этапы лабораторной без нарушения структуры исходного задания.

Так как сайт Books to Scrape не содержит поля автор, для выполнения цели «определить автора с наибольшим числом книг в топе» было проведено **обогащение данных** через открытое API **OpenLibrary**.  
Для каждой книги по названию выполнялся запрос к API:

[https://openlibrary.org/search.json?title=<название\_книги](https://openlibrary.org/search.json?title=%3cназвание_книги)>

Если автор найден — он сохранялся в поле author\_api.  
В результате удалось определить авторов для части книг и провести анализ частоты их появления в выборке.

На основе полученных данных построены графики:

* распределение цен и рейтингов книг;

Изображение выглядит как текст, снимок экрана, График, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.

* диапазон цен (boxplot);

Изображение выглядит как снимок экрана, Прямоугольник, линия, диаграмма

Содержимое, созданное искусственным интеллектом, может быть неверным.

* зависимость «цена – рейтинг» (scatterplot);

Изображение выглядит как текст, снимок экрана, диаграмма, График

Содержимое, созданное искусственным интеллектом, может быть неверным.

* топ-10 категорий/авторов по количеству книг;

Изображение выглядит как текст, линия, График, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Графики оформлены в едином цвете с использованием seaborn, сохранены в PNG формате (dpi = 300) для включения в отчёт.

**Автор с наибольшем количеством книг в томе: Shel Silverstein**

# Заключение

**Вывод**:

1. В ходе работы освоены практические приёмы парсинга HTML-страниц и работы с API.
2. При обращении к реальным площадкам (chitai-gorod, labirint, ozon) возникли блокировки — это подтвердило важность понимания антибот-механизмов.
3. Для решения задачи реализована адаптивная стратегия — смена источника на открытый аналог и обогащение данных внешним API.
4. Полученные результаты успешно визуализированы; оформлены графики, отражающие ценовую структуру и рейтинговые закономерности.
5. В результате был выделен наиболее часто встречающийся автор (по данным OpenLibrary) и сформирован итоговый аналитический отчёт.