PYTHON · АВТОМАТИЗАЦИЯ РУТИНЫ

Сверка прайс-листов на Python без pandas

Разбор задачи, рабочий скрипт на стандартной библиотеке и ловушки реальных Excel-файлов заказчика

ПРАЙС A	ПРАЙС B	ИТОГ	
Молоко 3.2	74.90	74.90	
Хлеб белый	41.20	39.80	PRICE
Сахар 1кг	95.00	95.00	
Масло сл.	312.00	—	MISS

```
$ python reconcile.py a.xlsx b.xlsx
diffs = 7   ->   result.xlsx
```

ЛИС практическая серия

Шадрин Данил

Сверка прайс-листов на Python без pandas

https://litres.ru/74697583
SelfPub; 2026

Аннотация

Реальная задача небольшой торговой компании: прайсы от трёх-пяти поставщиков, и их сводят глазами в Excel. В книге показано, как заменить это скриптом на чистом Python — без pandas и openpyxl, только стандартная библиотека. Разобрано: почему .xlsx — это zip с XML и как читать его самому, определение колонок по подсказкам в шапке, нормализация чисел с неразрывными пробелами (U+00A0 и U+202F), сравнение с отчётом PRICE_DIFF / MISSING / OUT_OF_STOCK / DUPLICATE и выдача результата в xlsx или CSV. Внутри полный исходник reconcile.py, таблица архитектуры, разбор ловушек реальных файлов заказчика и расчёт экономии времени. Полезно бухгалтеру, менеджеру, товароведу и начинающему Python-разработчику, который хочет зарабатывать на автоматизации рутины.

Содержание

Автоматизация сверки прайс-листов поставщиков на Python без сторонних тяжёлых библиотек 4

1. Постановка задачи 5
2. Почему выбран Python и стандартная библиотека 7

Конец ознакомительного фрагмента. 8

Шадрин Данил
Сверка прайс-листов на Python без pandas

Автоматизация сверки прайс-листов поставщиков на Python без сторонних тяжёлых библиотек

Учебно-практическая работа с исходным кодом. Тема: как заменить ручное сравнение

таблиц скриптом, если у заказчика «голый» Python и ставить pandas/openpyxl нельзя.

1. Постановка задачи

Небольшая торговая компания получает от трёх-пяти поставщиков файлы с прайсами.

Каждый поставщик присылает свой формат: у одного Артикул;Наименование;Цена,

у другого номер товара;товар;цена, руб;остаток на складе, у третьего — Excel

с шапкой в три строки. Внутри один и тот же артикул может встречаться дважды,

цены пишутся как 1 950,00 #, а часть товара заканчивается.

Сотрудник каждый день открывает таблицы, ищет глазами:

чего нет у поставщика, хотя есть в нашем прайсе;

где цена ушла;

где позиция закончилась;

где дубль артикула (обычно ошибка выгрузки);

что у поставщика появилось новое.

На 500 позиций это 40–60 минут, и ошибки неизбежны. Задача — сделать скрипт,

который делает это за секунды и выдаёт готовый отчёт.

Требования заказчика к решению:

Работает на обычном Python 3.8+, без pandas, numpy, openpyxl.

Понимает и CSV (, или ;, UTF-8/CP1251), и Excel .xlsx с разными шапками.

Сам определяет, какая колонка что значит, без ручного редактирования кода.

Результат — таблица расхождений и сводка по каждому файлу.

2. Почему выбран Python и стандартная библиотека

Установка pandas тянет за собой десятки мегабайт и часто ломается на корпоративных

компьютерах с 32-битным Office или старыми версиями Excel. Для задачи, где нужно

просто прочитать строки и сравнить числа, хватает модулей csv, zipfile,

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.