

Задание 1 (BigData)

В коде указаны комментарии по поводу принятых решений в ходе написания программы. Также есть общий счетчик времени и регистратор процессов, отслеживающие время выполнения каждого действия.

Если коротко:

1. Читаем данных через Dask (Надстройка над Pandas)
2. Избавляемся от лишних данных
3. Индексируем по ключу
4. Разбиваем данные по 100 Мб
5. Переходим к Pandas DataFrame
6. Ищем индексы недвижимости, встречающейся несколько раз
7. Через `pandas.dataframe.loc()` ищем ответ
8. Формируем файл-ответ

Файл-ответ и скрипт программы находятся в папке task1.

Задание 2 (Найти ошибку)

Я вижу здесь классическую проблему data racing condition, которая возникает, когда два пользователя пытаются изменить одни и те же данные. Для ее устранения есть несколько подходов:

1. Ограничения на уровне БД. Но в задаче просят исправить ошибку именно на уровне кода
2. Check-Then-Act – проверка данных перед их изменением. В данной задаче этого не достаточно для решения проблемы, как минимум потому что она уже есть в коде (`req.status == 'opened' && req.state == 'ready'`)
3. Read-Modify-Write – обеспечение атомарности за счет введение счетчика операций и контроля его значения. В данной задаче функционал этого подхода уже частично реализован (`request.version += 1`), но необходимо добавить в код проверку версии счетчика непосредственно перед записью новых данных.

Третий способ подходит, но, к сожалению, в условиях задачи не сказано, проверяют ли этот счетчик другие скрипты или микросервисы и есть ли конкретные смысловые обозначения каждого состояния данного инкремента. Поэтому я предположил, что нет ни того, ни другого.

Модифицированный код находится в папке task2.

Задание 3 (Прислать в телеграмм последнюю новость с vc.ru)

Честно говоря, я так и не смог понять, как можно **ВООБЩЕ** не использовать токены в этом задании, потому что API телеграмма для личных переписок приватный, это значит, что нельзя использовать свой собственный аккаунт для отправки сообщений. Единственный известный мне способ – это создать бота, но создание бота подразумевает создание токена для управления им.

Может вы хотели, чтобы я вынес токен в отдельный файл и прочитал его в коде из файла? Может вы хотели, чтобы я не использовал в коде **ДРУГИХ** токенов, кроме токена бота? Может я, конечно, не знаю, что есть какой-то способ (прошу рассказать).

В любом случае, я решил эту задачу, создав бота и используя один токен для него. И так как на странице vs.ru отображаются не последние новости, а рекомендуемые, я использовал страницу vs.ru/new.

Токен бота активен, но у меня, к сожалению, нет сервера, на котором я могу запустить процесс с ботом. Поэтому вы можете сами это сделать на локальной машине и написать боту: «привет», далее он вас проинструктирует.

Код для бота в папке `task3`.

Ссылка на бота в тг: https://t.me/the_korok_bot