

# Задание 1

## Основное задание

### Расположение

src/processors/tag\_processor.py

### Метод решения

Программа получает строку с тегами, которые могут быть представлены в различных стилях (например, snake\_case, kebab-case, CamelCase). Каждый тег преобразуется в стандартный формат без разделителей и в нижнем регистре. Это упрощает дальнейшее сравнение тегов, независимо от их исходного стиля написания. Программа проверяет каждый тег на соответствие правилам из заранее подготовленного списка правил. Для каждого тега проверяется, является ли он корректным (входит ли в список разрешенных или является синонимом). Если тег корректен, он добавляется в результаты. Некоторые теги могут быть составными и включать несколько разрешенных подстрок. Программа проверяет такие теги, разбивая их на части и извлекая разрешенные подстроки в виде отдельных тегов. Если в составном теге присутствует некорректный тег, то такой тег считается некорректным. Некорректные теги, которые не прошли проверку, либо сразу удаляются, либо (если включен режим отложенного удаления) записываются во временный кеш с датой истечения.

Для сохранения исходного порядка тегов применяется свойство словаря python: ключи в нем отсортированы по времени добавления

### Тесты

Основной тест был вынесен из main в директорию tests, туда же были добавлены еще тесты крайних случаев

## Дополнительное задание 1

### Расположение

src/parsers/html\_parser.py

### Метод решения

Модуль открывает HTML-файл, содержащий таблицу с правилами для тегов, и считывает его содержимое. Реализованный класс парсера позволяет искать теги HTML в кусках кода, наподобие BeautifulSoup. Трансформер ищет строки таблицы и извлекает данные из ячеек. Каждая строка таблицы содержит определенное правило, а заголовки таблицы указывают, к какому типу данных относится каждая ячейка (например, "Тэг", "Синонимы", "Оставить как есть"). После этого он проверяет, что заголовки таблицы соответствуют ожидаемому формату, предъявленному в условии. Из каждой строки таблицы создается объект правила. Парсер возвращает кортеж всех правил, который затем используется в основной задаче.

## Дополнительное задание 2

### Расположение

src/cache/cache.py

### Метод решения

При первом запуске программы создается файл для хранения кеша, с названием, содержащим уникальный ID задачи. Кеш сохраняется в специальной папке (.cache) в виде файла. Когда обнаруживается некорректный тег, программа добавляет его в кеш с текущей датой и временем истечения (через две недели). Если такой тег уже присутствует в кеше, дата истечения не обновляется. При каждом запуске программа проверяет, истек ли срок действия некорректных тегов. Если срок истек, тег удаляется из кеша и не включается в выходной результат. Если режим отложенного удаления включен, некорректные теги, срок хранения которых еще не истек, остаются в выходном результате. Если режим выключен, некорректные теги удаляются сразу, и кеш не используется.

## Запуск программы

### Взаимодействие с программой

Для запуска программы используйте следующую команду:

```
python main.py --path-to-rules-table=<path_to_table.html> --input=
<path_to_input_file> --output=<path_to_output_file>
```

Флаги:

--input=<path\_to\_input\_file> (обязательный): Путь к файлу с исходными тегами, которые будут обработаны. Входной файл должен содержать теги, разделенные символом ;. В качестве значения можно использовать stdin, если вы хотите ввести данные вручную через стандартный ввод.

--output=<path\_to\_output\_file> (обязательный): Путь к файлу, куда будут записаны результаты обработки. В выходном файле будут содержаться обработанные теги, отформатированные в соответствии с правилами. Можно использовать значение stdout, если вы хотите вывести результат на экран.

--task-id=<int> (опциональный): Идентификатор задачи. Этот флаг является числовым значением, которое используется для создания уникального кеша для каждого набора тегов. Необходим, если включен флаг --delayed-clean.

--delayed-clean (опциональный): Включает режим отложенного удаления некорректных тегов. Если этот флаг установлен, некорректные теги не будут немедленно удалены из выходных данных. Вместо этого они будут добавлены в кеш с датой истечения срока (например, через две недели). Если срок действия тега в кеше истек, тег будет удален автоматически.

-h: Выводит справочную информацию о программе и доступных флагах. Если используется -h, все остальные флаги становятся необязательными, и программа не будет выполнять обработку тегов.

### Тестирование

```
python3 -m unittest discover -s tests
```