**Веб-краулер на Python, который собирает стихи с сайта stihi.ru:**

***код для запуска с комментариями:***

***-------------------------------------------------------------------------------------***

import requests  
from bs4 import BeautifulSoup  
import abc

# Определяем интерфейс для стратегии скачивания страницы  
class PageFetcherStrategy(metaclass=abc.ABCMeta):  
    @abc.abstractmethod  
    def fetch(self, url: str) -> str:  
        pass

# Реализация стратегии скачивания страницы напрямую  
class DirectPageFetcher(PageFetcherStrategy):  
    def fetch(self, url: str) -> str:  
        response = requests.get(url)  
        return response.text

# Реализация стратегии скачивания страницы через прокси  
class ProxyPageFetcher(PageFetcherStrategy):  
    def fetch(self, url: str) -> str:  
        proxies = {"http": "http://your\_proxy\_here", "https": "https://your\_proxy\_here"}  
        response = requests.get(url, proxies=proxies)  
        return response.text

# Класс краулера  
class Crawler:  
    def \_\_init\_\_(self, fetcher\_strategy: PageFetcherStrategy):  
        self.fetcher\_strategy = fetcher\_strategy

    def crawl(self, url: str) -> None:  
        page\_content = self.fetcher\_strategy.fetch(url)  
        self.parse\_page(page\_content)

    def parse\_page(self, page\_content: str) -> None:  
        soup = BeautifulSoup(page\_content, 'html.parser')  
        poems = soup.find\_all('div', class\_='text')  
        for idx, poem in enumerate(poems):  
            print(f"Poem {idx+1}:\n{poem.text}\n{'-'\*20}")

# Алгоритм запуска краулера  
if \_\_name\_\_ == "\_\_main\_\_":  
    url = "https://stihi.ru/poemslist.html"  # URL для сбора данных

    # Выбор стратегии скачивания: DirectPageFetcher или ProxyPageFetcher  
    fetcher\_strategy = DirectPageFetcher()

    # Создаем экземпляр краулера и запускаем его  
    crawler = Crawler(fetcher\_strategy)  
    crawler.crawl(url)

------------------------------------------------------------------------------------------------------------

2. Алгоритм запуска краулера:

* - Определить интерфейс `PageFetcherStrategy` для стратегии скачивания страницы.
* - Реализовать различные стратегии скачивания страницы, например, `DirectPageFetcher` и `ProxyPageFetcher`.
* - Определить класс `Crawler`, который будет использовать одну из стратегий скачивания для получения и последующего анализа страницы.
* - В методе `crawl` класса `Crawler`, вызвать метод `fetch` выбранной стратегии для получения содержимого страницы, затем вызвать `parse\_page` для анализа и извлечения необходимых данных.
* - Создать экземпляр `Crawler`, передав в конструктор выбранную стратегию.
* - Запустить краулер, вызвав метод `crawl` с URL-адресом, который нужно обработать.

**Убедитесь, что у вас установлены библиотеки `requests` и `beautifulsoup4` перед запуском кода.**

**Вы можете установить их с помощью команды `pip install requests beautifulsoup4`.**