**базовый краулер для загрузки изображений с фиктивного imageboard.**

import requests  
from bs4 import BeautifulSoup  
from abc import ABC, abstractmethod  
import os

# Стратегия для выполнения HTTP-запросов  
class HttpRequestStrategy(ABC):

    @abstractmethod  
    def fetch(self, url: str) -> bytes:  
        pass

class DirectRequest(HttpRequestStrategy):

    def fetch(self, url: str) -> bytes:  
        response = requests.get(url)  
        response.raise\_for\_status()  
        return response.content

class ProxyRequest(HttpRequestStrategy):

    def \_\_init\_\_(self, proxy: str):  
        self.proxy = proxy

    def fetch(self, url: str) -> bytes:  
        proxies = {  
            "http": self.proxy,  
            "https": self.proxy  
        }  
        response = requests.get(url, proxies=proxies)  
        response.raise\_for\_status()  
        return response.content

class ImageboardCrawler:

    def \_\_init\_\_(self, strategy: HttpRequestStrategy, save\_path='./images'):  
        self.strategy = strategy  
        self.save\_path = save\_path  
        if not os.path.exists(save\_path):  
            os.makedirs(save\_path)

    def fetch\_images(self, board\_url: str):  
        html\_content = self.strategy.fetch(board\_url)  
        soup = BeautifulSoup(html\_content, 'html.parser')

        # Предполагая, что изображения находятся в тегах img с классом '.imageboard-img'  
        image\_tags = soup.select('.imageboard-img')  
         
        for img\_tag in image\_tags:  
            img\_url = img\_tag['src']  
            img\_name = os.path.basename(img\_url)  
             
            # Скачиваем изображение  
            img\_content = self.strategy.fetch(img\_url)  
            with open(os.path.join(self.save\_path, img\_name), 'wb') as img\_file:  
                img\_file.write(img\_content)  
            print(f"Downloaded {img\_name}!")

# Пример использования  
try:  
    crawler = ImageboardCrawler(DirectRequest())  
    crawler.fetch\_images("https://example-imageboard.com/boardname")

except Exception as e:  
    print(f"Ошибка: {e}")

* **Метод `fetch` возвращает `bytes` (бинарный контент), так как изображения обычно представлены в бинарном формате.**
* **Добавлена логика для сохранения изображений на диск.**

**- В качестве примера используется фиктивный URL `https://example-imageboard.com/boardname`. Вы должны заменить его на реальный URL доски**