Lab 13. Парсинг.

Скачать данные по разным районам Алматы. Сделать короткий анализ кода построчно.

Также используя данный код, получить данные из сайтов OLX.kz, kolesa.kz

Lab 13

Показать результат (скаченные файлы должны содержать не менее 1000 данных о квартирах).

Попытаться удалить некоторые избыточные данные (столбцы/ атрибуты) .

Код СРСП:

```
import requests
from bs4 import BeautifulSoup
from time import sleep
from datetime import datetime
import pandas as pd
from tqdm import tqdm



class parsing:
    def __init__(self, url_path='/prodazha/kvartiry/almaty-medeuskij-gornyj-gigant/?page=', count=3):
        self.page_url = 'https://krisha.kz'
        result = []
        for url in tqdm(list(set(self.__get_urls_posts(count, url_path)))):
            result.append(self.__get_places_information(url))
            sleep(1)
        pd.DataFrame(result).to_csv('dataset_{0}.csv'.format(datetime.now().strftime("%Y-%m-%d_%H-%M-%S")))
        print('Successfully parsed and saved!')


    def __get_urls_posts(self, count=3, url_path='/prodazha/kvartiry/almaty-medeuskij-gornyj-gigant/?page='):
        results = []
        for i in tqdm(range(2, count)):
            page = requests.get(self.page_url + url_path + str(i) + '/')
```

```
            soup = BeautifulSoup(page.content, 'html.parser')
            soup_find = [self.page_url+page.find(class_='a-card__title').get('href') for page in
soup.find_all(class_='a-card__header-left')]
            sleep(0.7)
            results += soup_find
        return results


    def __get_places_information(self, url):
        row = {}
        page = requests.get(url)
        soup = BeautifulSoup(page.content, 'html.parser')
        try:
            row['title'] = soup.find(class_='offer__advert-title').find('h1').text
        except:
            row['title'] = 'NaN'
        try:
            row['price'] = soup.find(class_='offer__price').text.strip().replace(u'\xa0', '')
        except:
            row['price'] = 'NaN'
        try:
            for data in soup.find_all(class_='offer__info-item'):
                try:
                    row[data.find(class_='offer__info-title').text] = data.find(class_='offer__advert-short-info').text
                except:
                    pass
        except:
            pass
        try:
            for data in soup.find(class_='offer__parameters').find_all('dl'):
                try:
```

```
            row[data.find('dt').text] = data.find('dd').text
        except:
            pass
    except:
        pass
    try:
        row['description'] = soup.find(class_='a-text a-text-white-spaces').text
    except:
        row['description'] = 'NaN'

    return row


def __write_to_csv(filename='dataset.csv', row=[]):
    with open(filename,'a') as file:
        file.write(';'.join(row))
        file.write('\n')
```