Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

МОРСКОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

и информационных технологий

ОТЧЕТ

о выполнении учебной практике по дисциплине

«Базы данных»

на тему

«Установление принципа работы сайтов интересных для поддержания информативности

базы данных»

ИСПОЛНИТЕЛЬ

Студент группы 20390 Сагиян Г.Г.

РУКОВОДИТЕЛЬ

Заместитель декана ФЦПТ Померанец Л. К.

Санкт-Петербург

2021

**Оглавление**

[Предварительные результаты разработки парсера для сайта “Закупки” 3](#_Toc75911449)

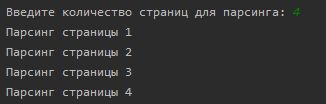
# Предварительные результаты разработки парсера для сайта “Закупки”

**Исходный код программы:**

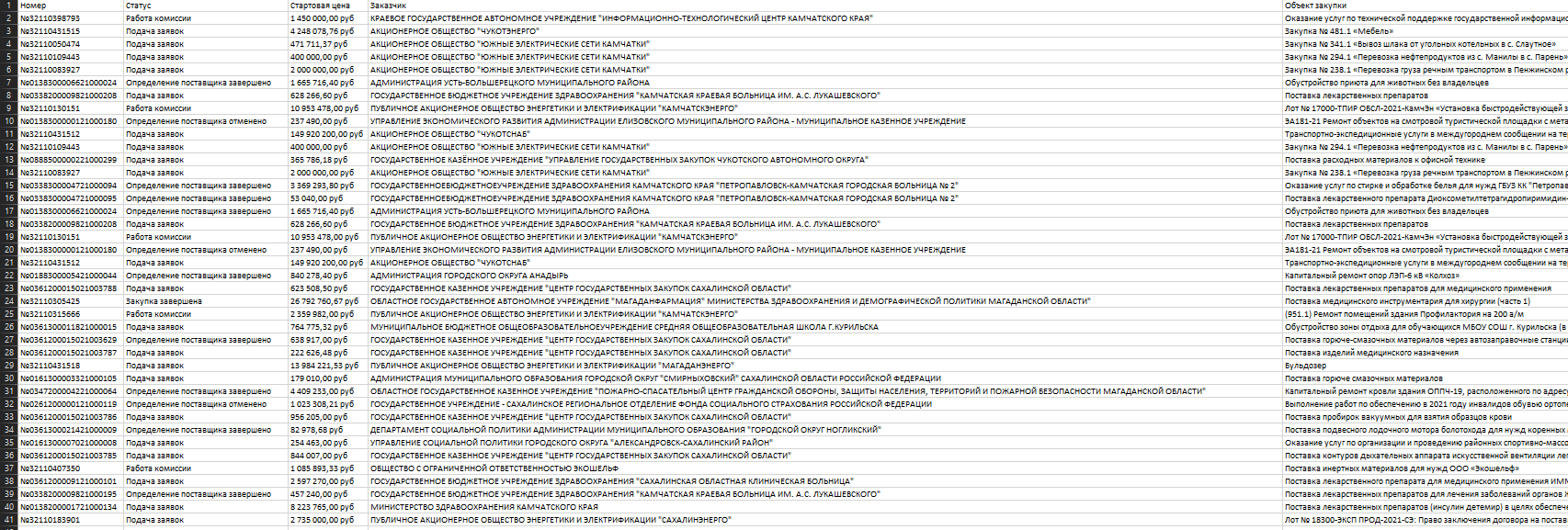
import requests  
from bs4 import BeautifulSoup  
import csv  
  
# Сделать ссылку по пользовательскому вводу  
FILE = 'parsing\_results.csv'  
URL = 'https://zakupki.gov.ru/epz/order/extendedsearch/results.html' # далее может быть задана пользователем,  
# хотя пока нет нужды, парсятся-то только страницы закупок  
HEADERS = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '  
 '(KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36',  
 'accept': '\*/\*'}  
  
  
def get\_html(url, params=None):  
 r = requests.get(url, headers = HEADERS, params = params)  
 return r  
  
  
def get\_content(html):  
 soup = BeautifulSoup(html, 'html.parser')  
 items = soup.find\_all('div', class\_='row no-gutters registry-entry\_\_form mr-0')  
 positions = []  
 for item in items: # Тут могут быть любе критерии для выбора записей(номера, объекты закупки, заказчики и т.д.)  
 positions.append({  
 'number': item.find('div', class\_='registry-entry\_\_header-mid\_\_number').get\_text().replace('\r\n', '').replace('\n', '').replace(' ', ''),  
 'status': item.find('div', class\_='registry-entry\_\_header-mid\_\_title').get\_text().replace('\r\n', '').replace('\n', '').replace(' ', ''),  
 'start\_price': item.find('div', class\_='price-block\_\_value').get\_text().replace('\r\n', '').replace('\n', '').replace(' ', '').replace('₽', 'руб'),  
 'customer': item.find('div', class\_='registry-entry\_\_body-href').get\_text().replace('\r\n', '').replace('\n', '').replace(' ', ''),  
 'procurement\_object': item.find('div', class\_='registry-entry\_\_body-value').get\_text().replace('\r\n', '').replace('\n', '').replace(' ', ''),  
 'last\_update': item.find('div', class\_='data-block\_\_value').get\_text()  
 })  
 return positions  
  
  
def save\_file(items, path):  
 *"""Сохранение результатов парсинга в файл"""* with open(path, 'w', newline = '') as file:  
 writer = csv.writer(file, delimiter=';')  
 writer.writerow(['Номер', 'Статус', 'Стартовая цена', 'Заказчик', 'Объект закупки', 'Обновлено'])  
 for item in items:  
 writer.writerow([item['number'], item['status'], item['start\_price'], item['customer'],  
 item['procurement\_object'], item['last\_update']])  
  
  
def parse():  
 html = get\_html(URL)  
 result = []  
 if html.status\_code == 200:  
 user\_req\_page = int(input("Введите количество страниц для парсинга: "))  
 page\_number = 1  
 while page\_number <= user\_req\_page:  
 html = get\_html(URL, params = {'pageNumber': page\_number})  
 result.extend(get\_content(html.text))  
 print(f"Парсинг страницы {page\_number}")  
 page\_number += 1  
 get\_content(html.text)  
 else:  
 print("Error")  
  
 print(result)  
 save\_file(result, FILE)  
 print(f'Найдено закупок: {len(result)}')  
  
  
parse()

**Возможности и результаты работы:**

* Запуск



* Результат



**Перспективы дальнейшего развития:**

1. Параметризация поиска записей
2. Автоматический парсинг с заданной периодичностью и уведомлениями об изменении статуса заказа