|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное автономное образовательное учреждение высшего образования  «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ» |

Кафедра компьютерных технологий и программной инженерии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ОТЧЁТ ПО ПРАКТИКЕ  ЗАЩИЩЁН С ОЦЕНКОЙ  Руководитель |  | | | |
| Ст. преп. |  |  |  | М.Д. Поляк |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

ОТЧЁТ ПО ПРАКТИКЕ

|  |  |  |
| --- | --- | --- |
| вид практики | производственная | |
| тип практики |  | |
| на тему индивидуального задания | | Парсинг сайта резерва членов избирательной комиссии Санкт-Петербурга |
|  | | | |
|  | | | |

|  |  |
| --- | --- |
| выполнен | Щетенюк Денисом Анатольевичем |
| фамилия, имя, отчество обучающегося в творительном падеже | |

|  |  |  |  |
| --- | --- | --- | --- |
| по направлению подготовки | 09.03.04 |  | Программная инженерия |
|  | код |  | наименование направления |
|  | | | |
| наименование направления | | | |
| направленности |  |  | Проектирование программных систем |
|  | код |  | наименование направленности |
|  | | | |
| наименование направленности | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Обучающийся группы № | 4032 |  |  |  | Д.А.Щетенюк |
|  | номер |  | подпись, дата |  | инициалы, фамилия |

Санкт–Петербург 2023

**Оглавление**

[1. Цель работы 3](#_Toc144804450)

[2. Исходные данные и их описание 3](#_Toc144804451)

[3. Практический раздел 3](#_Toc144804452)

[4. Результат 3](#_Toc144804453)

[5. Выводы 3](#_Toc144804454)

[Список литературы 4](#_Toc144804455)

[Приложение 4](#_Toc144804456)

# Цель работы

Реализовать парсинг сайта резерва избирательной комиссии Санкт-Петербурга.

# Исходные данные и их описание

Начальной точкой старта программы является ссылка на список избирательной комиссии.

# Практический раздел

Парсинг сайта избирательной комиссии реализован на Python.

Исходный код программы приведен в приложении.

Результат работы программы сохраняется в Google таблицу. https://docs.google.com/spreadsheets/d/1vRd3CxUpSzuYBujvHLHdtI2EL8nQ6FdqKTRuUP8JUX0/edit#gid=0

# Результат

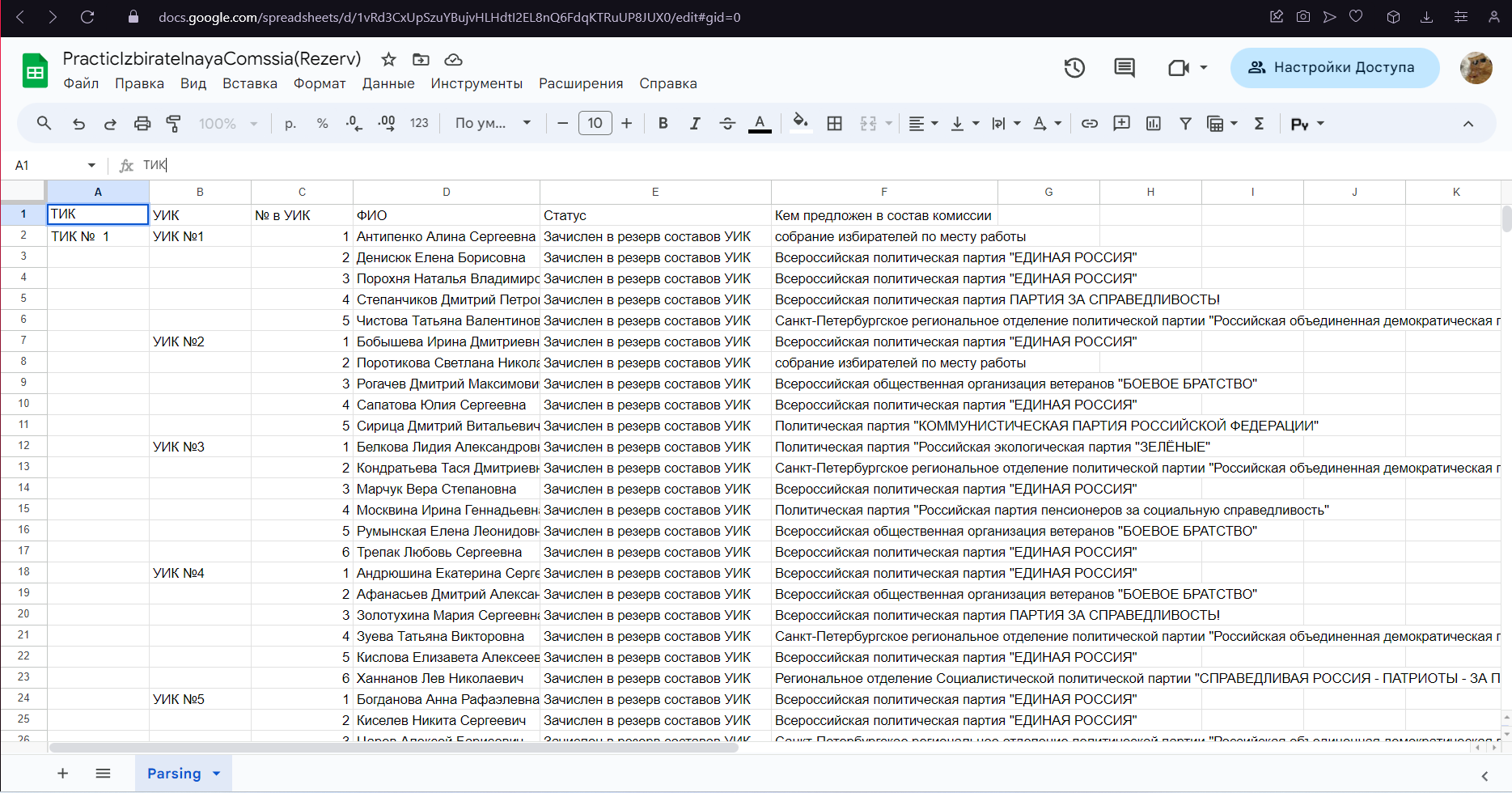


Рис 4.1. Результат работы программы

# Выводы

В результате выполнения практической работы была реализована программа, собирающая данные о резерве членов избирательной комиссии Санкт-Петербурга.

Программа имеет недостаток, связанный с записью данных в гугл таблицу. Из-за ограничения по количеству записей в минуту, было необходимо расставить паузы между записями, что существенно замедлило выполнение программы.

# Список литературы

1. Сайт избирательной комиссии Санкт-Петербурга. – URL: http://www.st-petersburg.vybory.izbirkom.ru/st-petersburg/ik\_r/
2. Python Quikstart (Вызов API Google Workspace) – URL: <https://developers.google.com/sheets/api/quickstart/python?hl=ru>
3. Гугл таблица – URL: https://docs.google.com/spreadsheets/d/1vRd3CxUpSzuYBujvHLHdtI2EL8nQ6FdqKTRuUP8JUX0/edit#gid=0
4. Beautiful Soup Documentation – URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
5. Requests Documentation – URL: https://requests.readthedocs.io/en/latest/

# Приложение

from \_\_future\_\_ import print\_function  
  
import os.path  
import requests  
from bs4 import BeautifulSoup  
import fake\_useragent  
import json  
import gspread  
import pickle  
import time  
  
from google.auth.transport.requests import Request  
from google.oauth2.credentials import Credentials  
from google\_auth\_oauthlib.flow import InstalledAppFlow  
from googleapiclient.discovery import build  
from googleapiclient.errors import HttpError  
  
# If modifying these scopes, delete the file token.json.  
SCOPES = ['https://www.googleapis.com/auth/spreadsheets']  
  
# The ID and range of a sample spreadsheet.  
SAMPLE\_SPREADSHEET\_ID = '1vRd3CxUpSzuYBujvHLHdtI2EL8nQ6FdqKTRuUP8JUX0'  
SAMPLE\_RANGE\_NAME = 'Parsing!A1:G20000'  
  
data\_base={  
 'first': 1,  
 'id2':278200090968,  
 'id': '#'  
}  
  
data\_tiks={  
 'id': 1  
}  
  
zapros='http://www.st-petersburg.vybory.izbirkom.ru/st-petersburg/ik\_r\_tree/278200090968?first=1&id2=278200090968&id=%23)'  
  
class GoogleSheet:  
 SPREADSHEET\_ID = '1vRd3CxUpSzuYBujvHLHdtI2EL8nQ6FdqKTRuUP8JUX0'  
 SCOPES = ['https://www.googleapis.com/auth/spreadsheets']  
 service = None  
  
 def \_\_init\_\_(self):  
 creds = None  
 if os.path.exists('token.pickle'):  
 with open('token.pickle', 'rb') as token:  
 creds = pickle.load(token)  
  
 if not creds or not creds.valid:  
 if creds and creds.expired and creds.refresh\_token:  
 creds.refresh(Request())  
 else:  
 print('flow')  
 flow = InstalledAppFlow.from\_client\_secrets\_file(  
 'credentials.json', self.SCOPES)  
 creds = flow.run\_local\_server(port=0)  
 with open('token.pickle', 'wb') as token:  
 pickle.dump(creds, token)  
  
 self.service = build('sheets', 'v4', credentials=creds)  
  
 def updateRangeValues(self, range, values):  
 data = [{  
 'range': range,  
 'values': values  
 }]  
 body = {  
 'valueInputOption': 'USER\_ENTERED',  
 'data': data  
 }  
 result = self.service.spreadsheets().values().batchUpdate(spreadsheetId=self.SPREADSHEET\_ID, body=body).execute()  
 print('{0} cells updated.'.format(result.get('totalUpdatedCells')))  
  
def main():  
 gs=GoogleSheet()  
  
 session=requests.Session()  
 base\_link='http://www.st-petersburg.vybory.izbirkom.ru/st-petersburg/ik\_r/278200090968'  
 user=fake\_useragent.UserAgent().random  
 header = {'user-agent': user}  
  
 response = session.get(zapros, headers=header, params=data\_base)  
 json\_part=response.json()[0]  
 child=json\_part['children']# список Тиков  
 iteration=2  
 for item in child:  
 Tik\_value=[  
 [item['text']]  
 ]  
 Tik\_Range=f'Parsing!A{iteration}:A{iteration}'  
 gs.updateRangeValues(Tik\_Range, Tik\_value)  
 time.sleep(5)  
 data\_tiks['id']=item['id']  
 r=requests.get(f"http://www.st-petersburg.vybory.izbirkom.ru/st-petersburg/ik\_r\_tree/?id={item['id']}", headers=header, params=data\_tiks)  
 json\_tiks=r.json()  
 for yik in json\_tiks:  
 time.sleep(5)  
 Yik\_value = [  
 [yik['text']]  
 ]  
 yik\_Range = f'Parsing!B{iteration}:B{iteration}'  
 gs.updateRangeValues(yik\_Range, Yik\_value)  
 yik\_info=requests.get(f"http://www.st-petersburg.vybory.izbirkom.ru/st-petersburg/ik\_r/{yik['id']}", headers=header).content  
 soup=BeautifulSoup(yik\_info, 'lxml')  
 block = soup.find\_all("tr")  
 for call in block:  
 if call.find("td"):  
 table\_value = []  
 table\_Range = f'Parsing!C{iteration}:F{iteration}'  
 for i in call.find\_all("td"):  
 thing=str(i.text).replace("\t", '').replace("\n", '')  
 table\_value.append(thing)  
 time.sleep(0.5)  
 gs.updateRangeValues(table\_Range, [table\_value])  
 iteration = iteration + 1  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()