**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики - процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Функциональное программирование»**

**на тему «Парсер социальных сетей»**

**Студент гр. 23Б15-пу**

**Кубякин Н.А.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

***Оглавление***

[1. Цель работы 3](#_Toc178797423)

[2. Описание задачи (формализация задачи) 3](#_Toc178797424)

[3. Теоретическая часть 4](#_Toc178797431)

[4. Основные шаги программы 4](#_Toc178797442)

[5. Описание программы 5](#_Toc178797442)

[6. Рекомендации пользователя 6](#_Toc178797443)

[7. Рекомендации программиста 7](#_Toc178797444)

[8. Исходный код программы 7](#_Toc178797445)

[9. Контрольный пример 8](#_Toc178797447)

[10. Вывод 10](#_Toc178797447)

[11. Источники 10](#_Toc178797448)

# Цель работы

# Создать программу для сбора статистики количества повторов слов на различных страницах в социальных сетях/мессенджерах. В данном случае рассматриваются «ВКонтакте» и «Телеграм»

# Описание задачи (формализация задачи)

Программа должна:

1. Принимать от пользователя предоставленные ключи и токены.
2. Используя токены собирать текстовую информацию с различных страниц.
3. Осуществлять обработку собранной информации.
4. Выводить информацию в графический интерфейс.

# Теоретическая часть

# Парсинг и сбор информации:

# Токен или ключ доступа — это случайно сгенерированная строка, состоящая из цифр, латинских букв и специальных символов. Токен возвращается в качестве результата авторизации пользователя.

# В «ВКонтакте» токен создается при входе пользователя в соцсеть. Токен

# содержит информацию о том, кто именно обращается к серверу и какие права у него есть.

# В «Телеграм» токен – это уникальный идентификатор, который позволяет взаимодействовать с API Telegram.

# Токены необходимы для обеспечения доступа к API, а так же идентификации и безопасности пользователя.

# Для работы с API и токенизации используются библиотеки spacy, nltk, vk\_api и telethon.

# Обработка информации:

# Для обработки файлов html и json используются библиотеки bs4 и json.

# Основные шаги программы

#### 1. **Сбор информации в html и json форматах с сайтов/приложений, используя данные токены. Интересующие страницы задаются пользователем.**

#### 2**. Обработка полученной информации с целью извлечения обычного текста.**

#### 3. **Поиск самых часто встречающихся слов и вычисления количества их появлений на странице.**

#### 4. **Удаление стоп-слов.**

#### 5. **Вывод результатов в графический интерфейс**

# Описание программы

Описание функций программы представлено в таблице 5.

*Таблица 5. Описание функций*

|  |  |  |
| --- | --- | --- |
| Имя | Параметры | Описание |
| vk\_get\_user\_id | username: Строка содержащая имя пользователя. | Метод для получения id пользователя. |
| vk\_get\_wall | owner\_id: Строка, содержащая id пользователя.  count: Количество постов | Метод для сбора информации со стены ВК. |
| parse\_vk | groups: Группы для парсинга.  data\_queue: Очередь для собранной информации.  num\_posts: Количество постов для парсинга. | Метод для парсинга заданных групп ВК. |
| parse\_tg | group\_names: Группы для парсинга.  data\_queue: Очередь для собранной информации.  num\_messages: Количество сообщений для парсинга. | Метод для парсинга заданных чатов/каналов Телеграм. |
| preprocess\_data | data\_queue: Собранная информация.  preprocessed\_data: Переменная для хранения текста.  stop\_words: Стоп-слова. | Метод для подготовки информации к анализу. |
| show\_data | prefix: Флаг соцсети.  textarea: Место для показа текста.  text: Текст для вывода.  stop\_words: Стоп-слова | Вспомогательный метод для вывода текста в интерфейс |
| vk\_analyze | num\_threads: Количество параллелей для анализа.  num\_vk\_posts: Количество постов.  stop\_words: Стоп-слова. | Метод для финального анализа данных ВК. |
| tg\_analyze | num\_threads: Количество параллелей для анализа.  num\_telegram\_posts: Количество постов.  stop\_words: Стоп-слова. | Метод для финального анализа данных ТГ. |
| choice | data\_queue: Собранная информация.  preprocessed\_data: Переменная для хранения текста. | Вспомогательная функция для анализа текста. |

# Рекомендации пользователя

Для использования программы:

1. Получить токены ВК и Телеграм. Инструкции можно найти в официальной документации.
2. Открыть файл с кодом и ввести токены в соответствующие места в коде (переменные после объявления библиотек).
3. Сохранить код и запустить программу.
4. Ввести id чатов/групп в соответствующие поля.
5. Нажать кнопку “Analyze”
6. В интерфейсе выведется информация по самым частым словам в введённых группах/чатах.

# Рекомендации программиста

# Для запуска программы необходим Python, а также 64-битная операционная система Windows, или Linux, или macOS.

# Библиотеки: bs4 [[1]](#bs4), codecs [[2]](#cod), json [[3]](#jso), threading [[4]](#thr), nltk [[5]](#nlt), spacy [[6]](#spa), vk\_api [[7]](#vka), telethon [[8]](#tel), asyncio [[9]](#asy), tkinter [[10]](#tki).

# Исходный код программы

# Исходный код программы и необходимые текстовые файлы доступны по ссылке: <https://github.com/NikiTaku1/spbu_funcprog/tree/main/lab2>

# Контрольный пример

В данном разделе представлены контрольные примеры работы программы.

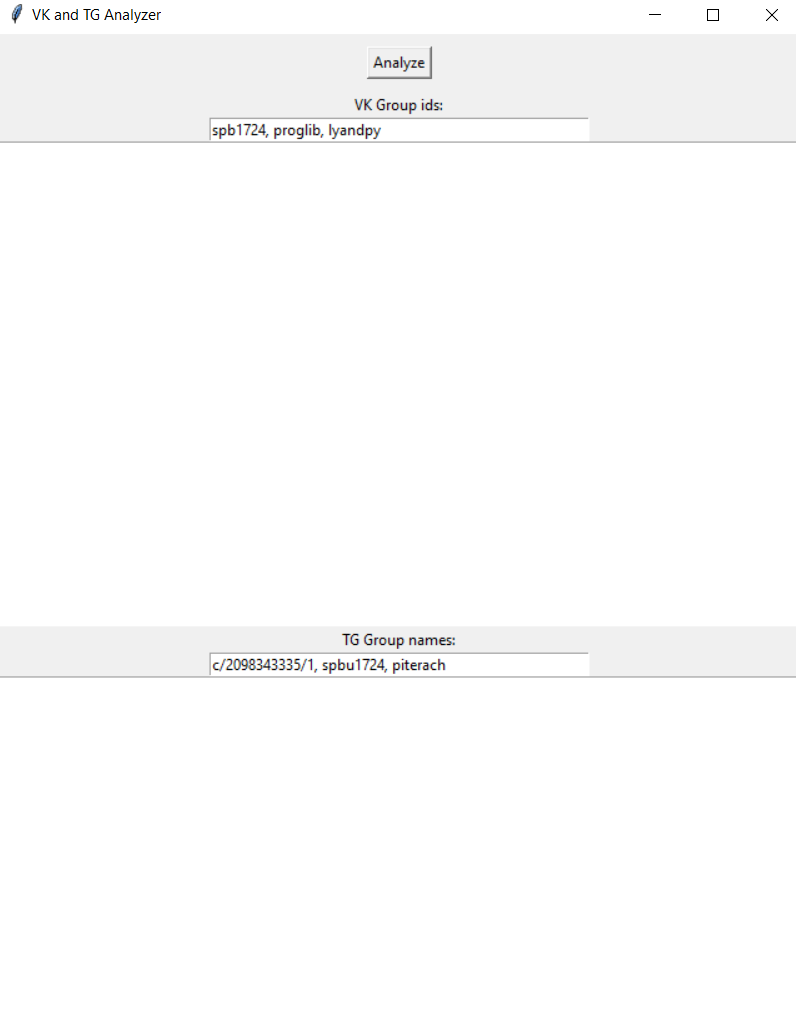


Рис. 9.1 Окно интерфейса

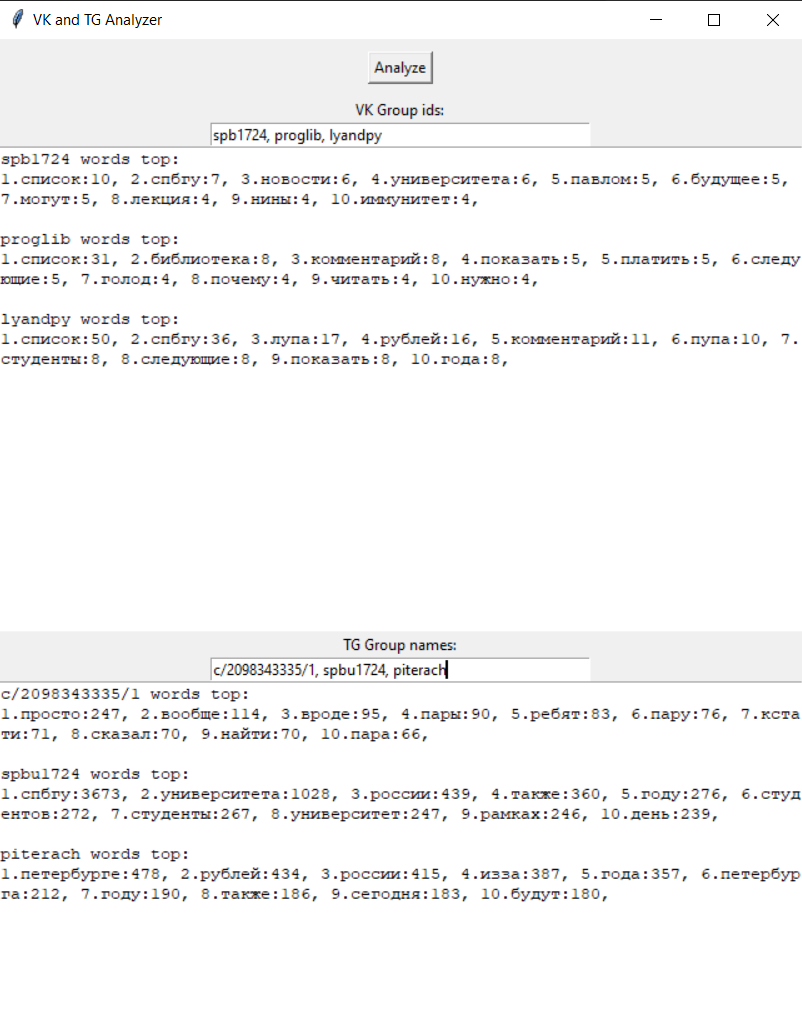


Рис. 9.2 Вывод результатов

# Вывод

# В ходе разработки был успешно создан парсер социальных сетей «ВКонтакте» и «Телеграм». Программа имеет графический интерфейс и умеет считать количество самых часто встречающихся слов на странице.

# Источники

# 1. bs4 — bs4 documentation // Documentation URL: <https://beautiful-soup-4.readthedocs.io/en/latest/> (дата обращения: 15.12.2024).

# 2. codecs — codecs documentation // Documentation URL: [https://docs.python.org/3/library/codecs.html](https://docs.python.org/3/library/codecs.html%20) (дата обращения: 15.12.2024).

# 3. json — json documentation // Documentation URL: <https://docs.python.org/3/library/json.html> (дата обращения: 15.12.2024).

# 4. threading — threading documentation // Documentation URL: <https://docs.python.org/3/library/threading.html> (дата обращения: 15.12.2024).

# 5. nltk — nltk documentation // Documentation URL: <https://nltk.readthedocs.io/en/latest/> (дата обращения: 15.12.2024).

# 6. spacy — spacy documentation // Documentation URL: [https://spacy.io/api/doc](https://spacy.io/api/doc%20) (дата обращения: 15.12.2024).

# 7. vk\_api — vk\_api documentation // Documentation URL: <https://vk-api.readthedocs.io/en/latest/index.html> (дата обращения: 15.12.2024).

# 8. telethon — telethon documentation // Documentation URL: [https://docs.telethon.dev/en/stable/](https://docs.telethon.dev/en/stable/%20) (дата обращения: 15.12.2024).

# 9. asyncio — asyncio documentation // Documentation URL: <https://docs.python.org/3/library/asyncio.html> (дата обращения: 15.12.2024).

# 10. tkinter — tkinter documentation // Documentation URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 15.12.2024).