**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

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

**ОТЧЕТ**

**по лабораторной работе №3**

**по дисциплине «Функциональное программирование»**

**на тему**

**«Распределенная обработка и анализ больших данных»**

**Студент гр. 23Б15-пу**

**Черевко М.E.**

**Преподаватель**

**Киямов Ж. У.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](#_Toc185819753)
2. [Описание задачи (формализация задачи) 4](#_Toc185819754)
3. [Теоретическая часть 7](#_Toc185819755)
4. [Основные шаги программы 12](#_Toc185819756)
5. [Описание программы 15](#_Toc185819757)
6. [Рекомендации пользователя 17](#_Toc185819758)
7. [Рекомендации программиста 18](#_Toc185819759)
8. [Исходный код программы 18](#_Toc185819760)
9. [Контрольный пример 19](#_Toc185819761)
10. [Вывод 21](#_Toc185819762)
11. [Источники 22](#_Toc185819763)

# Цель работы

Целью является создание удобного инструмента с графическим интерфейсом на Python с использованием библиотеки Tkinter, который позволяет:

* Собирать данные с помощью API VK и Reddit.
* Создать программу, которая параллельно соберет данные из всех источников. Каждый источник должен быть обработан в отдельном потоке или процессе.
* Проводить обработку текста (удаление ссылок, знаков пунктуации, упоминаний и стоп-слов).
* Отследить наиболее частотные слова и хэштеги.
* Сохранять результаты анализа и исходные данные в формате JSON для дальнейшего использования.

Разработанное приложение ориентировано на использование в маркетинговых исследованиях, анализе пользовательских предпочтений, мониторинге активности в социальных сетях и других задачах, связанных с анализом больших текстовых данных.

# Описание задачи (формализация задачи)

Целью проекта является создание программы для парсинга и анализа данных из социальных сетей VK и Reddit, а также дальнейшего анализа текста на основе частоты слов и хештегов. Программа будет извлекать и обрабатывать посты из указанных пользователем групп VK и сабреддитов Reddit, удалять ненужные элементы (например, стоп-слова, знаки препинания, ссылки), анализировать частотность слов и хештегов, а затем сохранять результаты в формате JSON.

Задача состоит из следующих этапов:

1. **Извлечение данных**

Необходимо реализовать два вида извлечения данных:

* **Из VK**: Программа должна отправлять запросы к VK API для получения постов из указанных групп. Для этого используется метод wall.get, принимающий параметры:
  + access\_token (токен доступа к VK API).
  + v (версия API).
  + domain (короткое имя группы).
  + count (количество постов для извлечения).
  + filter (фильтрация по типу постов, например, только авторские).
* **Из Reddit**: Программа должна отправлять запросы к Reddit API для получения постов из указанных сабреддитов. Для этого используется библиотека praw, которая взаимодействует с API Reddit для получения горячих постов из заданных сабреддитов. Программа должна извлекать данные о заголовке поста, тексте, оценке, количестве комментариев и URL.

1. **Предобработка текста**

После извлечения данных необходимо провести предобработку текста:

* **Удаление ссылок**: Из текста удаляются все URL-адреса.
* **Удаление упоминаний (@)**: Убираются все упоминания пользователей, начинающиеся с "@".
* **Удаление знаков препинания**: Текст очищается от всех знаков препинания, чтобы остались только слова.
* **Удаление стоп-слов**: Для каждого поста удаляются слова, которые являются стоп-словами на русском или английском языке (в зависимости от языка текста).
* **Оставление только буквенных слов**: Убираются все слова, содержащие цифры или другие символы.

1. **Определение языка**

Для каждого поста необходимо определить язык текста (русский или английский). Это делается путем подсчета количества букв в каждом языке (русских и латинских), и текст с преобладанием определенного типа букв будет классифицирован на этот язык.

1. **Анализ данных**

После предобработки текста необходимо провести анализ:

* **Частотность слов**: Необходимо посчитать, как часто встречаются каждое слово в выбранных постах.
* **Частотность хештегов**: Необходимо выделить хештеги (слова, начинающиеся с символа "#") и подсчитать их частоту.

Для этого используется метод Counter, который подсчитывает количество уникальных элементов в списке.

1. **Сохранение результатов**

После того как анализ будет завершен, программа должна предоставить пользователю возможность сохранить результаты:

* **Парсинг данных**: Все извлеченные данные (посты с VK и Reddit) должны быть сохранены в JSON файл.
* **Результаты анализа**: Анализ частоты слов и хештегов должен быть также сохранен в JSON файл.

Программа должна поддерживать опцию сохранения всех полученных данных (постов) в отдельный файл, если пользователь выбрал эту опцию.

1. **Параллельная обработка данных**

Чтобы ускорить процесс парсинга и анализа данных, используется многозадачность. Для этого применяется ThreadPoolExecutor из библиотеки concurrent.futures, который позволяет параллельно отправлять запросы к VK и Reddit, а также обрабатывать и анализировать данные в многозадачном режиме.

1. **Пользовательский интерфейс (GUI)**

Программа предоставляет интерфейс с использованием библиотеки Tkinter, через который пользователь может:

* Вводить группы VK и сабреддиты Reddit, из которых нужно извлечь посты.
* Указывать количество постов для извлечения из каждой социальной сети.
* Задавать количество популярных слов и хештегов для анализа.
* Выбирать опцию для сохранения извлеченных данных в файл JSON.
* Запускать процесс парсинга и анализа данных.

1. **Ошибки и исключения**

Программа должна обрабатывать возможные ошибки:

* **Ошибка при запросе VK API**: Например, если неверен токен доступа или если группа не существует.
* **Ошибка при запросе Reddit API**: Если неправильные клиентский ID, секретный ключ или пользовательский агент.
* **Ошибка обработки данных**: Например, если данные с VK или Reddit не соответствуют ожидаемому формату.

Таким образом, задача сводится к извлечению данных из VK и Reddit, их обработке и анализу, с возможностью дальнейшего сохранения и использования этих данных в будущем.

# Теоретическая часть

В данной задаче разрабатывается программа для парсинга данных из социальных сетей VK и Reddit, обработки текста постов и проведения анализа с целью выявления популярных слов и хештегов. Теоретическая основа задачи охватывает несколько ключевых аспектов: парсинг данных из API, обработка текстовой информации, анализ частотности слов и хештегов, а также многозадачность и создание пользовательского интерфейса. Рассмотрим эти аспекты более подробно.

* 1. **Парсинг данных из социальных сетей**
  2. **API социальных сетей**

Парсинг данных из социальных сетей VK и Reddit осуществляется через их публичные API (программные интерфейсы). API позволяют программам взаимодействовать с сервером социальной сети для получения информации, такой как посты, комментарии, хештеги и т.д. Основные этапы работы с API:

* **Авторизация**: Для работы с API VK и Reddit требуется регистрация приложения и получение ключей доступа (токенов) для авторизации. Эти токены позволяют получать доступ к данным социальной сети от имени приложения.
* **Запросы**: Чтобы извлечь данные, нужно отправить HTTP-запрос с параметрами, которые определяют, какие данные и в каком количестве необходимо получить. Запросы могут быть различных типов (GET, POST) в зависимости от того, что требуется сделать (например, получить посты или отправить комментарии).
* **Обработка ответа**: Ответ от API приходит в виде JSON-объекта. JSON (JavaScript Object Notation) — это легкий формат обмена данными, который легко воспринимается человеком и машинами. Программа анализирует ответ в формате JSON, извлекает необходимые данные (например, текст поста, заголовок и т.д.).
  1. **Парсинг данных VK**

Для извлечения постов из групп VK используется метод wall.get VK API. Этот метод позволяет получить список постов с конкретной стены группы, с фильтрацией по типу постов (например, только авторские посты). Ответ от API содержит информацию о каждом посте, такую как текст, количество лайков, комментариев и репостов.

VK API ограничивает количество постов, которые можно получить за один запрос, поэтому для получения больших объемов данных потребуется выполнить несколько последовательных запросов.

* 1. **Парсинг данных Reddit**

Для работы с Reddit используется библиотека praw (Python Reddit API Wrapper), которая предоставляет удобный интерфейс для взаимодействия с API Reddit. Через API можно получать посты из различных сабреддитов, сортировать их по популярности и извлекать текстовые данные.

* 1. **Типы запросов и ограничения**
* **Ограничения по запросам**: Оба API (VK и Reddit) имеют ограничения на количество запросов в единицу времени (rate limiting), что означает, что необходимо учитывать паузы между запросами, чтобы избежать блокировки.
* **Формат данных**: Важно правильно обрабатывать формат данных, полученных от API, так как разные социальные сети могут возвращать информацию в разных структурах.

1. **Обработка текста**
   1. **Предобработка текста**

Перед анализом данных текст из постов необходимо предварительно обработать, чтобы исключить влияние ненужных элементов. Предобработка текста включает несколько шагов:

* **Удаление ссылок**: В тексте могут быть ссылки на другие страницы, которые не несут полезной информации для анализа. Для их удаления используется регулярное выражение, которое находит все строки, начинающиеся с http, и заменяет их на пустые строки.
* **Удаление упоминаний**: В тексте могут встречаться упоминания пользователей социальной сети (например, @username). Эти элементы также не являются частью анализа и удаляются.
* **Удаление знаков препинания**: Пунктуация не имеет значения при анализе частоты слов, поэтому все знаки препинания (например, точки, запятые, восклицательные знаки) удаляются из текста.
* **Удаление стоп-слов**: Стоп-слова — это слова, которые часто встречаются в языке, но не несут существенного значения (например, "и", "в", "на" для русского языка или "the", "a", "of" для английского). Стоп-слова удаляются для улучшения точности анализа.
* **Оставление только буквенных слов**: Чтобы исключить числовые данные и специальные символы, оставляются только слова, состоящие из букв.
  1. **Определение языка**

Чтобы правильно обработать текст, необходимо определить, на каком языке он написан. Для этого используются простые методы, такие как подсчет числа символов, принадлежащих к определенному алфавиту (русскому или латинскому). Текст с преобладанием букв русского алфавита будет отнесен к русскому языку, а с преобладанием латинских букв — к английскому.

1. **Анализ данных**
   1. **Частотный анализ**

Частотный анализ заключается в подсчете, как часто встречаются различные слова или хештеги в тексте. Для этого используется структура данных Counter из библиотеки collections, которая позволяет подсчитать количество вхождений каждого элемента в коллекции (в данном случае — слов и хештегов).

* **Частотность слов**: Программа подсчитывает, сколько раз каждое слово встречается в анализируемых постах. Для этого слова из текста помещаются в список и затем применяется метод Counter, который подсчитывает их количество.
* **Частотность хештегов**: Хештеги — это слова, начинающиеся с символа "#". Для их выделения используется регулярное выражение, и затем анализируется частота их появления.
  1. **Подсчет хештегов**

Хештеги часто используются в социальных сетях для категоризации постов или выделения ключевых тем. Подсчет частоты хештегов помогает понять, какие темы наиболее популярны среди пользователей.

1. **Многозадачность**
   1. **Параллельная обработка данных**

Для ускорения процесса парсинга и обработки данных используется многозадачность с помощью библиотеки concurrent.futures. Эта библиотека позволяет запускать несколько задач одновременно, что существенно сокращает время обработки данных. В данном случае используются потоки (ThreadPoolExecutor), чтобы параллельно отправлять запросы к API VK и Reddit, а также параллельно обрабатывать и анализировать полученные данные.

* 1. **Преимущества многозадачности**

Многозадачность позволяет ускорить обработку больших объемов данных, минимизировать время ожидания между запросами к API и улучшить отзывчивость программы. В данном контексте многозадачность особенно полезна для работы с большими объемами данных, когда одновременный доступ к нескольким источникам данных (VK и Reddit) может значительно ускорить процесс.

1. **Пользовательский интерфейс (GUI)**
   1. **Использование Tkinter**

Для взаимодействия с пользователем используется библиотека Tkinter, которая является стандартной для Python и позволяет создавать графические интерфейсы. GUI включает элементы управления, такие как текстовые поля (для ввода групп VK и сабреддитов Reddit), кнопки (для начала парсинга), флажки (для выбора сохранения данных), и метки (для отображения инструкций).

* 1. **Взаимодействие с пользователем**

Пользователь может ввести параметры для парсинга, такие как список групп VK, сабреддитов Reddit, количество постов и количество популярных слов и хештегов для анализа. После нажатия на кнопку "Запустить парсинг" программа начинает выполнение всех этапов задачи, а затем сообщает пользователю о завершении работы.

1. **Ошибки и исключения**
   1. **Обработка ошибок API**

Необходимо учитывать возможность ошибок при запросах к API. Это могут быть ошибки авторизации (например, неверный токен), ошибки сети (например, отсутствие соединения с интернетом) или ошибки на стороне сервера (например, API временно недоступен). В таких случаях программа должна корректно обрабатывать исключения и информировать пользователя об ошибках.

* 1. **Обработка ошибок в обработке данных**

Программа должна быть устойчива к ошибкам при обработке текста (например, если текст не содержит слов или имеет странную структуру) и корректно реагировать на такие ситуации.

# Основные шаги программы

Программа состоит из нескольких ключевых шагов, начиная от получения данных с социальных платформ VK и Reddit, их обработки, анализа и заканчивая сохранением результатов в файлы. Рассмотрим эти шаги более подробно:

* 1. Инициализация и настройка внешних библиотек

Программа начинается с импорта необходимых библиотек, которые обеспечивают функциональность для работы с API (например, requests, praw), обработки текста (например, nltk, re), создания графического интерфейса (через tkinter) и многозадачности (с использованием ThreadPoolExecutor из библиотеки concurrent.futures). Также загружаются стоп-слова для обработки текста, такие как "и", "на", "the" и другие, которые не несут существенной смысловой нагрузки при анализе.

* 1. **Подготовка данных для работы с API**

Для работы с API платформ VK и Reddit необходимо указать уникальные данные, такие как токены доступа и идентификаторы клиентов, которые будут использоваться при отправке запросов к этим платформам. VK требует токен доступа и версию API, а Reddit — клиентский ID, секретный ключ и пользовательский агент.

* 1. **Получение данных из VK**

Программа отправляет запросы к API VK для получения постов из указанных групп. Запросы отправляются с помощью библиотеки requests, и возвращаются данные в формате JSON. Если запрос успешен, программа извлекает необходимые данные, такие как текст поста, количество лайков, комментариев, репостов и просмотров.

* 1. **Получение данных из Reddit**

Для получения данных с Reddit используется библиотека praw. Она подключается к Reddit API с помощью предоставленных идентификаторов клиента и пользовательского агента, извлекает посты из указанных сабреддитов и получает информацию о каждом посте, включая заголовок, текст, рейтинг, количество комментариев и ссылку на пост.

* 1. Предобработка текста

Перед анализом текста программа выполняет его очистку:

* Приводит текст к нижнему регистру.
* Удаляет все URL-ссылки и упоминания пользователей (например, через символ "@" в Reddit).
* Убирает пунктуацию и оставляет только слова.
* Очищает текст от стоп-слов, таких как предлоги и союзы, которые не несут смысловой нагрузки.

Этот этап позволяет избавиться от лишней информации и подготовить данные для дальнейшего анализа.

* 1. **Анализ данных**

После предобработки текста программа выполняет анализ частоты слов и хештегов. Сначала программа анализирует все слова в постах и подсчитывает, как часто каждое слово встречается. Затем она отдельно анализирует хештеги, извлекая те слова, которые начинаются с символа "#". Результаты анализа (самые частотные слова и хештеги) сортируются и выводятся пользователю.

* 1. **Сохранение результатов**

После завершения анализа программа сохраняет результаты в два JSON-файла:

* **all\_parsed\_data.json**: содержит данные о постах, полученные из VK и Reddit, если выбрана опция сохранения парсинга.
* **analysis\_results.json**: содержит результаты анализа частотности слов и хештегов, полученных из всех постов.

Кроме того, пользователю выводится уведомление о завершении работы программы.

* 1. **Графический интерфейс**

Программа использует графический интерфейс, реализованный с помощью библиотеки tkinter, который позволяет пользователю:

* Вводить группы VK и сабреддиты Reddit для парсинга.
* Указывать количество постов, которые необходимо получить с каждой платформы.
* Настроить количество популярных слов и хештегов для анализа.
* Выбрать, сохранять ли данные о постах в JSON.

После ввода необходимых данных, пользователь нажимает кнопку для запуска парсинга, и программа выполняет все шаги, описанные выше.

* 1. **Многозадачность и параллельная обработка**

Для повышения производительности программа использует многозадачность через ThreadPoolExecutor. Это позволяет параллельно отправлять запросы к API VK и Reddit, а также обрабатывать полученные данные, что значительно ускоряет выполнение программы, особенно при работе с большими объемами данных.

# Описание программы

Программная реализация выполнена на языке Python 3.12.7 с использованием библиотек requests и praw для взаимодействия с API социальных сетей VK и Reddit, а также tkinter для разработки графического интерфейса пользователя.

Программа выполняет парсинг данных из групп VK и сабреддитов Reddit, извлекает информацию о постах, очищает текст от ссылок, упоминаний и стоп-слов, а также анализирует частотность слов и хештегов. Все полученные данные сохраняются в JSON-формате. Для повышения производительности используется многозадачность с помощью библиотеки concurrent.futures, что позволяет параллельно обрабатывать данные с обеих платформ. Графический интерфейс позволяет пользователю настроить параметры парсинга и анализа данных, а также сохранить результаты.

Таблица 1. Parsing\_func.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Описание | Возвращаемое значение | Параметры функции (описание) |
| fetch\_vk\_posts | Получает посты из указанной группы VK с помощью API VK | JSON-ответ от API или None | token (str): токен доступа к API VK, version (str): версия API, domain (str): домен группы VK, count (int): количество постов для получения |
| fetch\_reddit\_posts | Получает посты из указанного сабреддита Reddit с помощью API Reddit | Список словарей с постами | client\_id (str): ID клиента Reddit, client\_secret (str): секретный ключ Reddit, user\_agent (str): пользовательский агент, subreddit\_name (str): название сабреддита, post\_count (int): количество постов для получения |
| preprocess\_text | Обрабатывает текст: удаляет стоп-слова, пунктуацию, ссылки и прочее | Список слов | |  | | --- | |  |   text (str): текст для обработки, language (str): язык текста ('ru' или 'en')   |  | | --- | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | analyze\_data |  |  | | --- | |  | | Анализирует данные: подсчитывает частотность слов и хештегов | Кортеж с двумя списками | posts (list): список постов для анализа, top\_n (int): количество наиболее популярных слов/хештегов для анализа |
| detect\_language | Определяет язык текста (русский или английский) | Строка 'ru' или 'en' | text (str): текст для анализа |
| save\_results | Сохраняет результаты в JSON-файлы | None | vk\_data (list): данные о постах VK, reddit\_data (list): данные о постах Reddit, analysis\_results (dict): результаты анализа, save\_parsed\_data (bool): флаг для сохранения данных парсинга |
| collect\_data | Собирает данные с VK и Reddit, выполняет анализ и сохраняет результаты | None | vk\_groups (list): список групп VK, reddit\_subreddits (list): список сабреддитов Reddit, count\_vk (int): количество постов для VK, count\_reddit (int): количество постов для Reddit, top\_n (int): количество популярных слов/хештегов, save\_parsed\_data (bool): флаг для сохранения данных |
| start\_parsing | Запускает процесс парсинга данных с VK и Reddit, инициализируя все необходимые шаги | None | None |

# Рекомендации пользователя

1. **Указание групп и сабреддитов:**  
   При вводе групп VK и сабреддитов Reddit в соответствующие поля разделяйте их запятыми без пробелов. Например:
   * Для VK: group1,group2,group3
   * Для Reddit: subreddit1,subreddit2,subreddit3
2. **Настройка количества постов:**  
   Укажите количество постов для парсинга в полях "Количество постов VK" и "Количество постов Reddit". Для достижения баланса между объемом данных и скоростью выполнения рекомендуется указывать значения от 10 до 100.
3. **Число популярных слов/хештегов:**  
   Укажите число наиболее часто встречающихся слов и хештегов, которые будут включены в результаты анализа. Оптимальное значение — 10-20.
4. **Сохранение данных:**  
   Если вы хотите сохранить сырые данные парсинга (включая текст постов), отметьте опцию "Сохранять данные парсинга в JSON". Это создаст файл all\_parsed\_data.json.
5. **Форматирование результатов:**  
   Результаты анализа (частотность слов и хештегов) будут автоматически сохранены в файле analysis\_results.json для последующего изучения.
6. **Обработка ошибок:**
   * Если API VK или Reddit недоступно, программа может вывести сообщение об ошибке в консоль. Убедитесь, что вы используете правильные токены и параметры.
   * Для работы с Reddit убедитесь, что вы настроили приложение в Reddit и ввели корректные client\_id, client\_secret и user\_agent.
7. **Многозадачность:**  
   Программа использует многопоточный сбор данных для ускорения работы. Убедитесь, что ваш компьютер обладает достаточной производительностью для обработки больших объемов данных.
8. **Определение языка текста:**  
   Программа автоматически определяет язык текста (русский или английский) на основе символов. Убедитесь, что данные в группах или сабреддитах соответствуют этим языкам.

# Рекомендации программиста

1. **Работа с токенами и ключами:**
   * Храните токены VK и Reddit (VK\_TOKEN, REDDIT\_CLIENT\_ID, REDDIT\_CLIENT\_SECRET) в защищенных конфигурационных файлах или используйте переменные окружения. Это уменьшит риск утечки данных.
   * Убедитесь, что токены имеют актуальный срок действия и необходимый набор разрешений для доступа к API.
   * .
2. **Обновление зависимостей:**
   * Регулярно обновляйте используемые библиотеки (pandas, requests, praw, nltk) до актуальных версий.
   * Используйте команду: **pip install наименование**
   * Проверьте обратную совместимость функций после обновления, чтобы избежать сбоев.
3. **Совместимость и переносимость:**
   * Убедитесь, что программа совместима с разными версиями Python (например, 3.8 и выше). Для этого можно использовать pyenv или виртуальные окружения.
   * Для работы на других системах (например, Windows, Linux) протестируйте пути сохранения файлов и другие системные функции.
4. **Форматы данных:**

* Убедитесь, что программа совместима с разными версиями Python (например, 3.8 и выше). Для этого можно использовать
* Убедитесь, что сохраняемые файлы JSON имеют строгую структуру. Это облегчит их обработку в других программах.
* Добавьте поддержку сохранения результатов в других форматах, например, CSV или Excel, для удобства анализа.

Следуя этим рекомендациям, вы сможете улучшить стабильность, производительность и расширяемость программы, а также облегчить её поддержку.

# Исходный код программы

<https://github.com/FraaaM/Trend_analysis_via_social_media_parsing>

# Контрольный пример

1. Запустите программу **parsing.py**, сделать это можно с помощью VS code или через cmd если установлен python. В результате должно появиться следующее окно (Рис. 1):

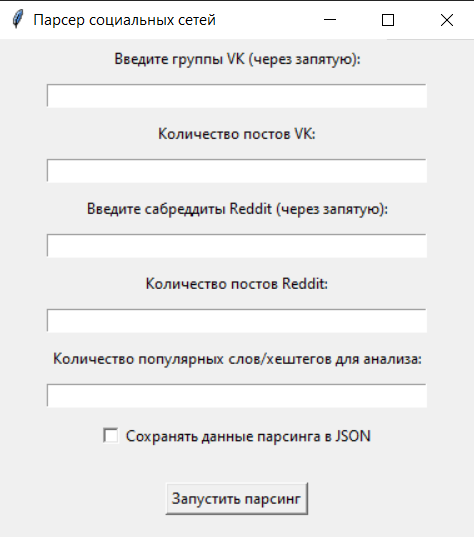


Рис. 1 – Начальное окно.

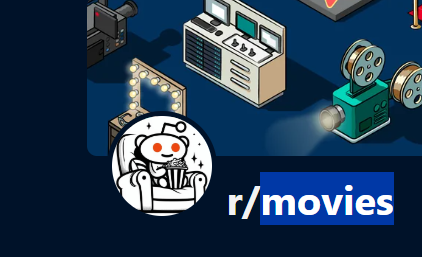
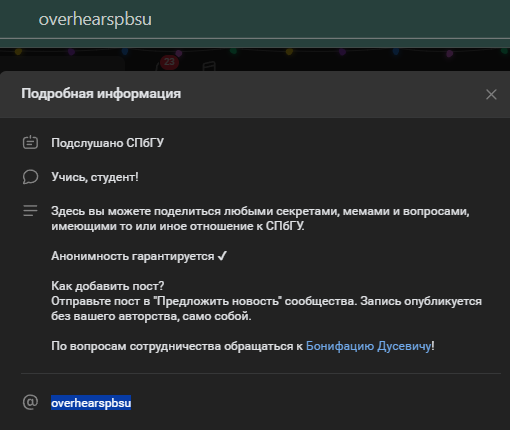
1. Для исправной работы вы должны ввести в соответствующие строчки названия групп. Для VK необходимо короткое имя группы (Рис. 2), для Reddit необходимо название сабреддита (Рис 3), также можно задать количество постов для парсинга и дополнительно хранить все результаты.

Рис. 3 – Сабреддит Рис. 2 - Короткое имя сообщества VK

1. Что бы без ошибок запустить программу, заполните поля аналогично как показано на рисунке (Рис. 4) и нажмите кнопку «Запустить парсинг».

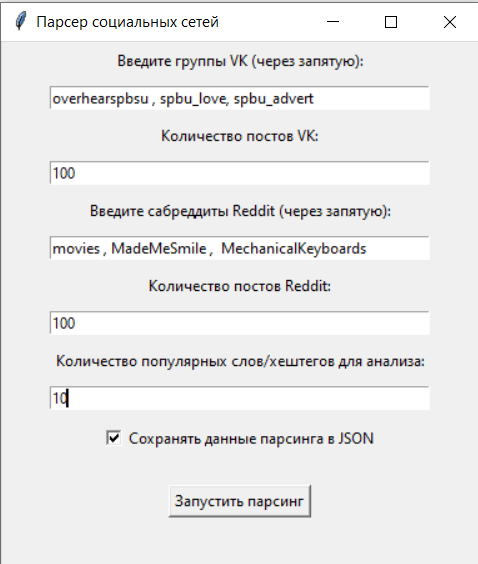


Рис. 4 – Заполненные поля.

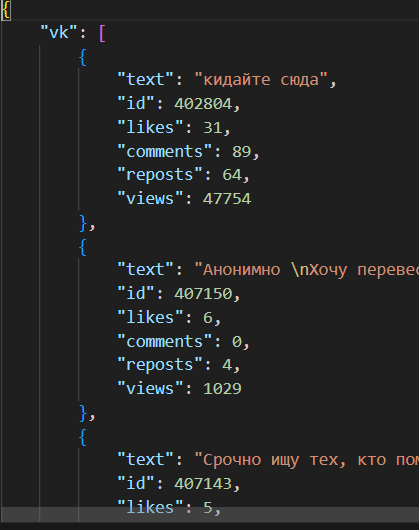
1. При штатной работы программы на выход вы получите 1 или 2 файла: analysis\_results.json (Рис. 5) и all\_parsed\_data.json (Рис. 6).

Рис. 5 - analysis\_results.json Рис. 6 - all\_parsed\_data.json

# Вывод

Проект представляет собой инструмент для парсинга и анализа данных из социальных сетей VK и Reddit, реализованный на языке Python. Основной целью разработки было создание удобного и функционального приложения, позволяющего извлекать, обрабатывать и анализировать данные из социальных сетей, предоставляя пользователю результаты в удобном формате.

Программа позволяет подключаться к API VK и Reddit, загружать заданное количество постов, обрабатывать их текстовую информацию, исключая стоп-слова, символы пунктуации, ссылки и ненужные элементы, а также проводить анализ для выявления наиболее популярных слов и хештегов. Это особенно полезно для мониторинга контента, анализа трендов или проведения маркетинговых исследований.

Код программы отличается модульной структурой, где каждая функция выполняет строго определённую задачу. Это упрощает как чтение кода, так и его модификацию или добавление новых функций. Использование многопоточности через ThreadPoolExecutor обеспечивает высокую производительность при одновременной работе с несколькими источниками данных, что делает приложение подходящим для обработки больших объёмов информации.

Графический интерфейс программы создан с использованием библиотеки Tkinter, что делает приложение интуитивно понятным и доступным даже для пользователей без технической подготовки. В интерфейсе реализована возможность указания групп VK и сабреддитов Reddit, настройки количества загружаемых постов, выбора числа популярных слов и хештегов для анализа, а также сохранения данных в JSON-файлы для дальнейшего использования.

Проект достиг поставленных целей, демонстрируя высокую функциональность и практическую пользу. Тем не менее, для улучшения программы можно рассмотреть возможность добавления новых функций, таких как визуализация результатов анализа, поддержка других социальных сетей или предоставление API для интеграции с внешними системами.

# Источники

* **tkinter**  
  <https://docs.python.org/3/library/tkinter.html>  
  Дата обращения: (23.12.2024)
* **nltk (Natural Language Toolkit)**  
  <https://www.nltk.org/>  
  Дата обращения: (23.12.2024)
* **pandas**  
  <https://pandas.pydata.org/>  
  Дата обращения: (23.12.2024)
* **requests**  
  <https://requests.readthedocs.io/>  
  Дата обращения: (23.12.2024)
* **praw (Python Reddit API Wrapper)**  
  <https://praw.readthedocs.io/>  
  Дата обращения: (23.12.2024)
* **concurrent.futures**  
  <https://docs.python.org/3/library/concurrent.futures.html>  
  Дата обращения: (23.12.2024)
* **stopwords (NLTK)**  
  <https://www.nltk.org/nltk_data/>  
  Дата обращения: (23.12.2024)
* **VK API Documentation**  
  <https://dev.vk.com/api>Дата обращения: (23.12.2024)
* **Reddit API Documentation**  
  <https://www.reddit.com/dev/api>  
  Дата обращения: (23.12.2024)