

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчёт
по курсу «**Естественно-языковой интерфейс интеллектуальных
систем**»

Лабораторная работа №4
«Семантико-синтаксический анализ текстов естественного языка»

Выполнили студенты группы 121701:	Чвилёв И.А. Воронцов Р.Г. Силибин С.
Проверил:	Крапивин Ю.Б.

Минск 2024

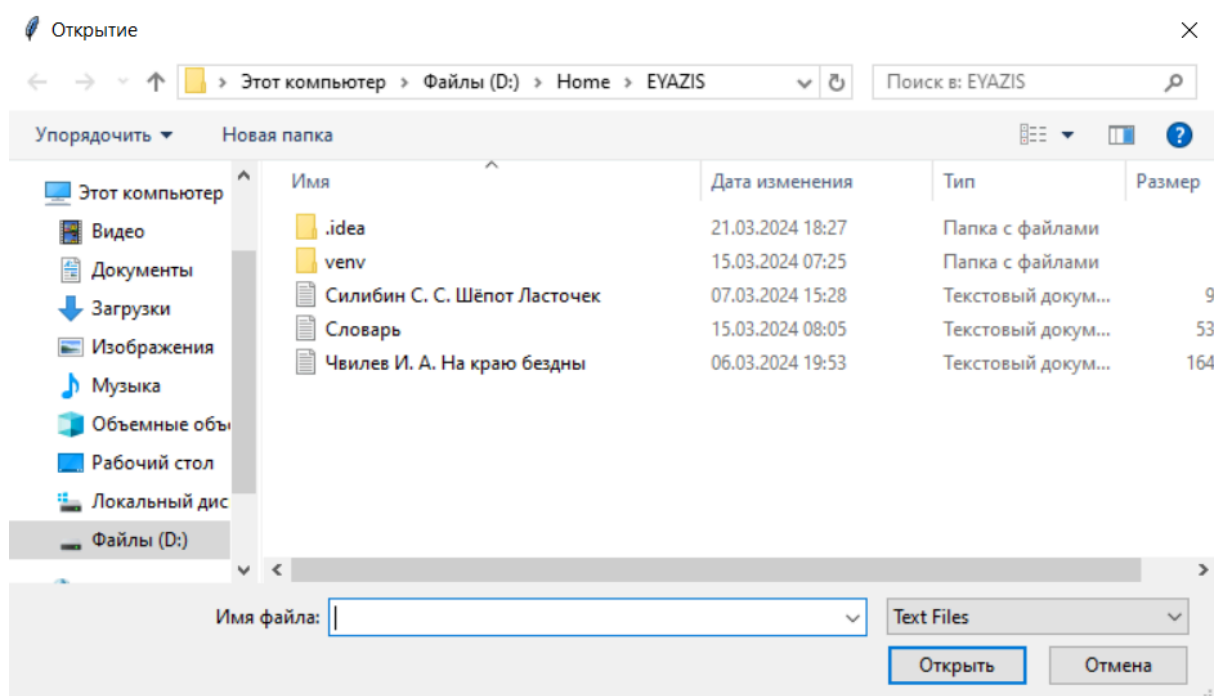
Цель работы: Освоить принципы разработки прикладных сервисных программ для решения задачи автоматического семантико-синтаксического анализа текста естественного языка.

Задание:

1. Познакомиться с назначением, структурой и функциональностью, предоставляемой базовым ЛП для решения задачи автоматического семантико-синтаксического анализа ТЕЯ.
2. Закрепить навыки программирования при решении задач автоматической обработки ТЕЯ.

Алгоритм работы программы:

- 1) Открывается диалог выбора файла на диске
 - а) Пользователь выбирает текстовый файл, переход к шагу 2



- 2) Программа проходит по всему тексту, получает все слова из файла, подсчитывает их частоту, производит морфологический анализ и создает описание для каждого слова.

```

def parse_text(self, file_paths):
    words = []
    for file_path in file_paths:
        with open(file_path, "r", encoding="utf-8") as file:
            content = file.read()
            file_words = re.findall(r"\w+", content.lower())
            words.extend(file_words)

    self.word_freq = Counter(words)

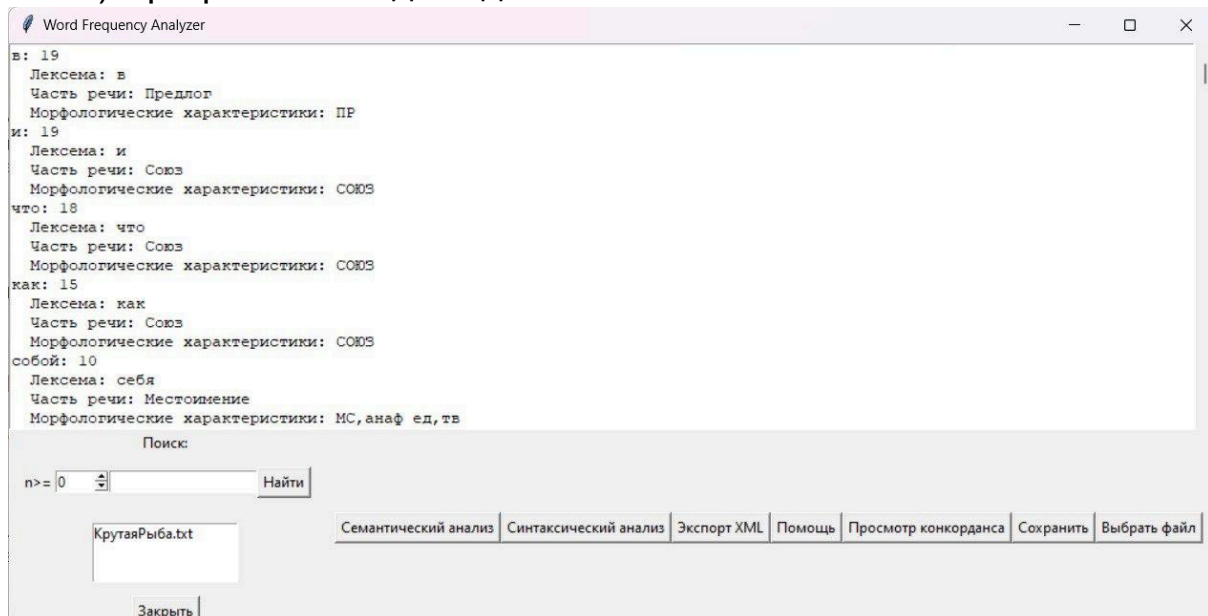
    for word in self.word_freq:
        parsed_word = self.morph.parse(word)[0]
        word_desc = {
            "wordform": word,
            "lexeme": parsed_word.normal_form,
            "pos": convert_tags_to_russian(parsed_word.tag.POS),
            "morphological_properties": parsed_word.tag.cyr_repr,
        }
        self.word_desc[word] = word_desc

    self.word_freq = dict(sorted(self.word_freq.items(), key=lambda item: item[1], reverse=True))
    self.update_text_box()

```

3) Программа формирует основное окно интерфейса и выводит сформированный словарь. Формируются поля для поиска, фильтрации по длине слова, просмотра конкорданса, синтаксического анализа, выбора файла (файлов), завершения работы, вкладка помощи, сохранения текста, экспорта.

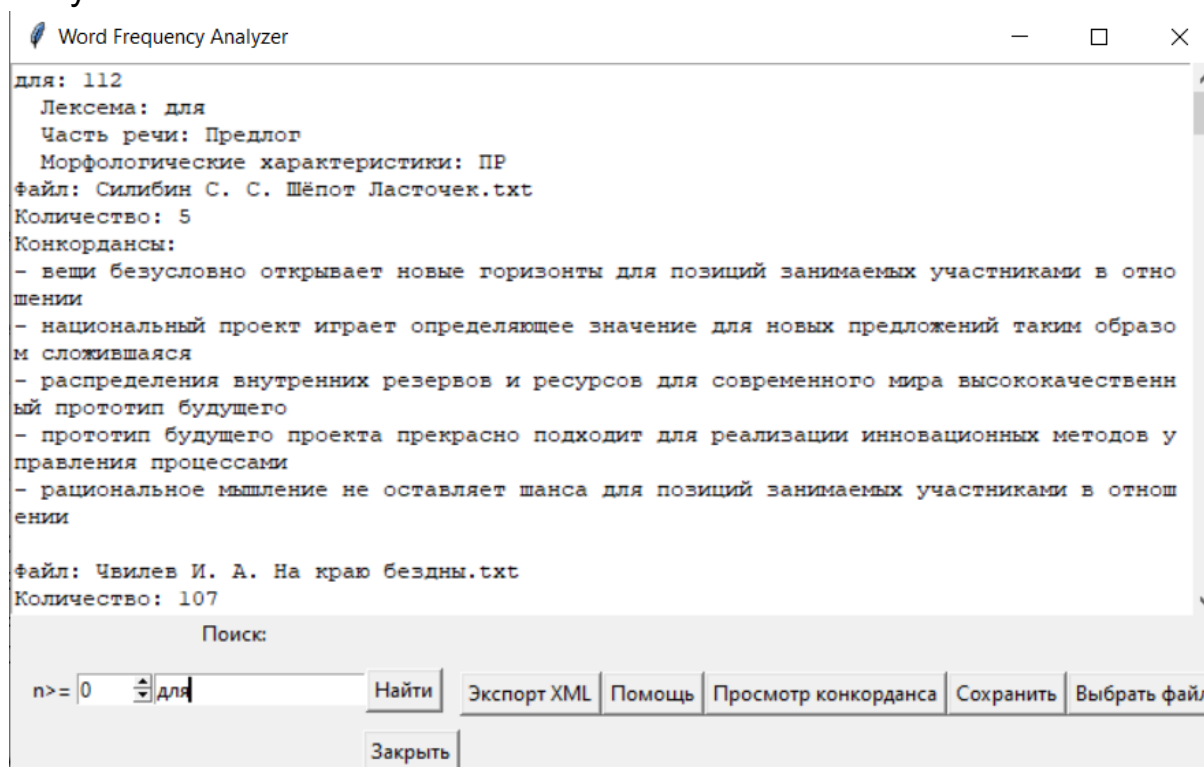
3.1) Программа ожидает действия пользователя.



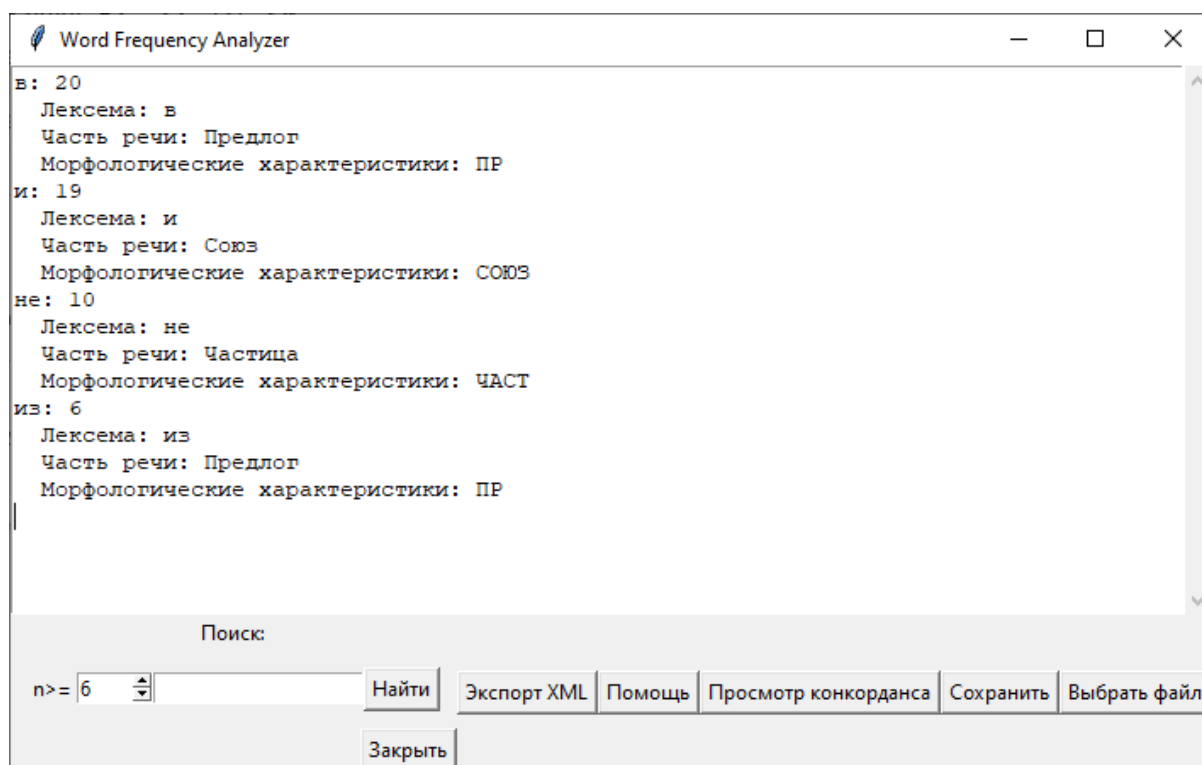
а) Пользователь вводит запрос в строку поиска и нажимает “Найти”. Программа переходит к шагу 4.

- б) Пользователь выбирает минимальное значение n , введя число в поле слева от строки поиска и нажимает “Найти”. Программа переходит к шагу 5.
- г) Пользователь нажимает “Сохранить”. Программа переходит к шагу 6.
- д) Пользователь нажимает на кнопку “Помощь”. Программа переходит к шагу 7.
- е) Пользователь нажимает на кнопку “Экспорт XML”. Программа переходит к шагу 8.
- ж) Пользователь нажимает на кнопку “Синтаксический анализ”. Программа переходит к шагу 9.
- з) Пользователь нажимает на кнопку “Семантический анализ”. Программа переходит к шагу 10.
- и) Пользователь закрывает программу. Программа переходит к шагу 11.

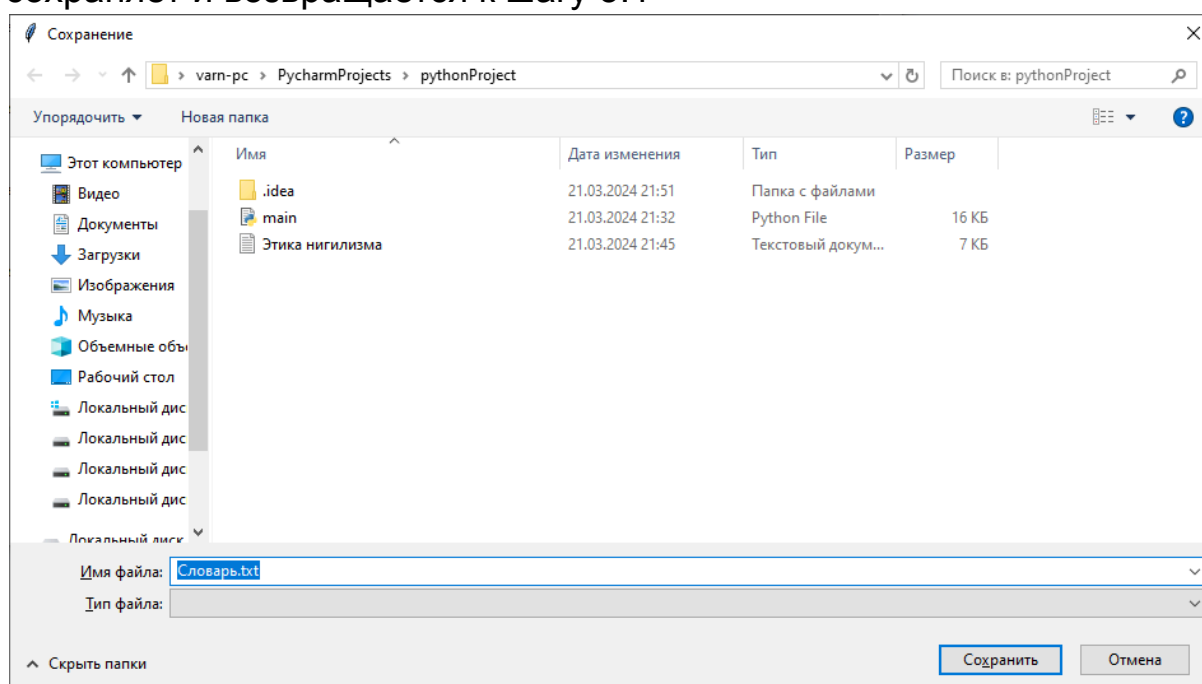
4) Список выведенных слов обновляется, включая только те, которые содержат значение поиска. Программа возвращается к шагу 3.1.



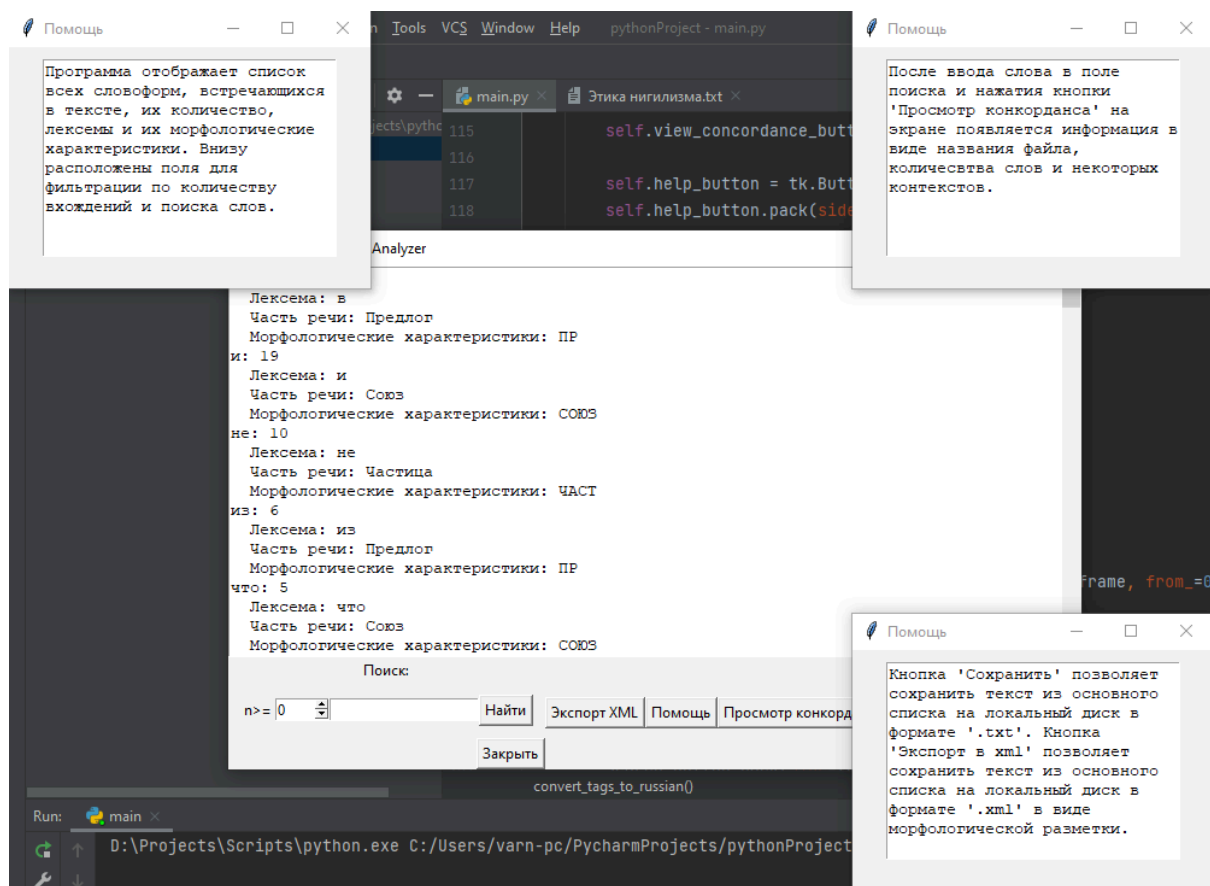
5) Список выведенных слов обновляется, включая только те, число которых больше или равно значению в поле “ $n \geq$ ”. Программа возвращается к шагу 3.1



6) Программа предлагает пользователю выбрать имя файла и расположение на диске, куда этот файл будет сохранён. Затем сохраняет и возвращается к шагу 3.1



7) Открываются 3 окна-подсказки, объясняющие пользователю что, где и для чего расположено. Заккрыть их можно повторным нажатием на "Помощь". Программа возвращается к шагу 3.1



8) Результаты работы программы экспортируются в файл формата XML.

```
<?xml version='1.0' encoding='windows-1251'?>
<text>
<w>банальные<ana lemma="банальный" pos="Прилагательное (полное)" gram="ПРИЛ, кач мн, им" />
</w>
<w>но<ana lemma="но" pos="Союз" gram="СОЮЗ" />
</w>
<w>неопровержимые<ana lemma="неопровержимый" pos="Прилагательное (полное)" gram="ПРИЛ, кач мн, им" />
</w>
<w>выводы<ana lemma="вывод" pos="Существительное" gram="СУЩ, неод, мр мн, вн" />
</w>
<w>а<ana lemma="а" pos="Союз" gram="СОЮЗ" />
</w>
<w>также<ana lemma="также" pos="Частица" gram="ЧАСТ" />
</w>
<w>акционеры<ana lemma="акционер" pos="Существительное" gram="СУЩ, од, мр мн, им" />
</w>
<w>крупнейших<ana lemma="крупный" pos="Прилагательное (полное)" gram="ПРИЛ, превосх, кач мн, рд" />
</w>
```

9) Программа формирует окно с синтаксическим анализом. Здесь же можно вводить запрос в строку поиска в соответствующее поле,

и, нажав на кнопку “Найти”, синтаксический анализ обновится.

Синтаксический анализ

Поиск: Найти

Предложение:
Равным образом, социально-экономическое развитие представляет собой интересный эксперимент проверки укрепления моральных ценностей.

Синтаксическое дерево:

→ Равным	Прилагательное-определение
→ образом	Обстоятельство
→ ,	Знак препинания
→ социально-экономическое	Прилагательное-определение
→ развитие	Субъект
→ представляет	фиксированное выражение
→ собой	Прилагательное-определение
→ интересный	Объект
→ эксперимент	Существительное-модификатор
→ проверки	Существительное-модификатор
→ укрепления	Прилагательное-определение
→ моральных	Существительное-модификатор
→ ценностей	Знак препинания
→ .	

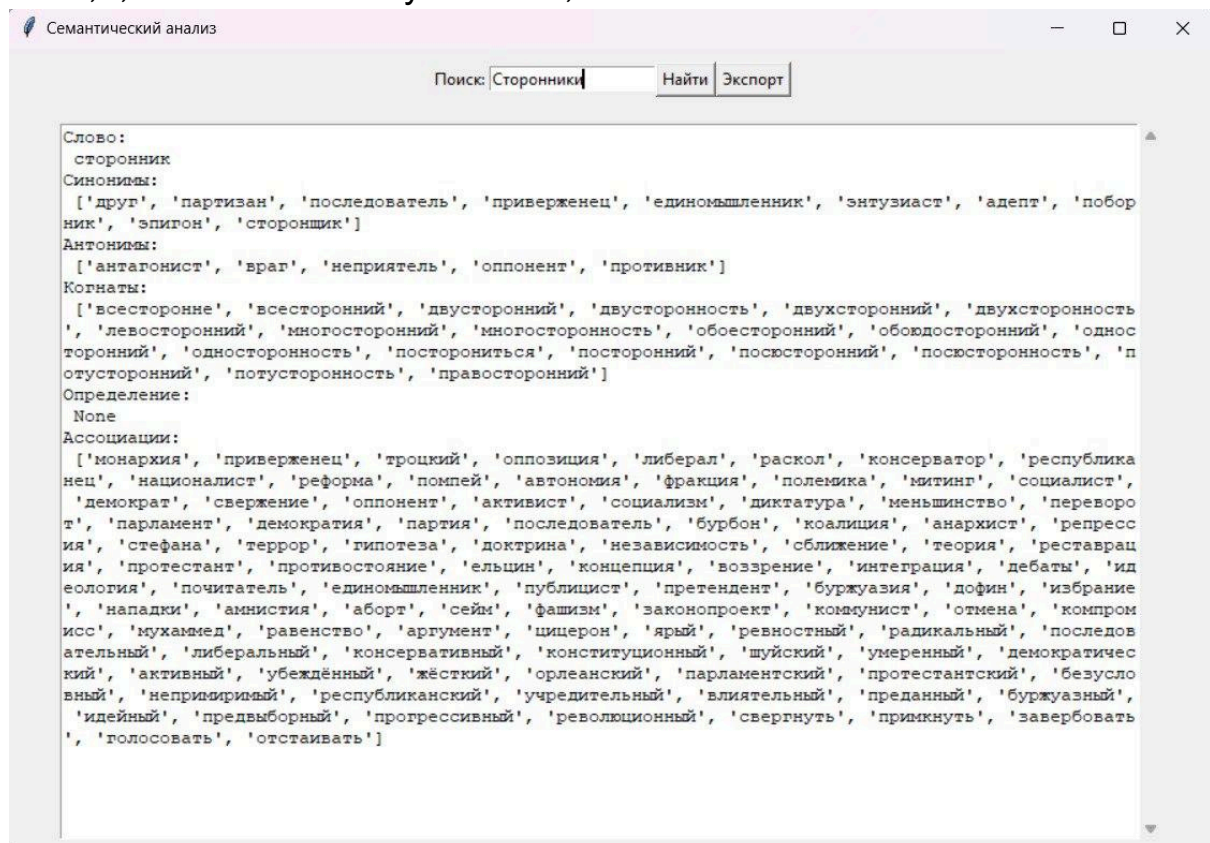
Предложение:
А ещё сторонники тоталитаризма в науке, инициированные исключительно синтетически, своевременно верифицированы.

Синтаксическое дерево:

→ А	Координационный союз
→ ещё	
→ сторонники	Субъект
→ тоталитаризма	Существительное-модификатор
→ в	Падежная маркировка
→ науке	Существительное-модификатор
→ ,	Знак препинания
→ инициированные	Прилагательное-модификатор
→ исключительно	Наречие-модификатор

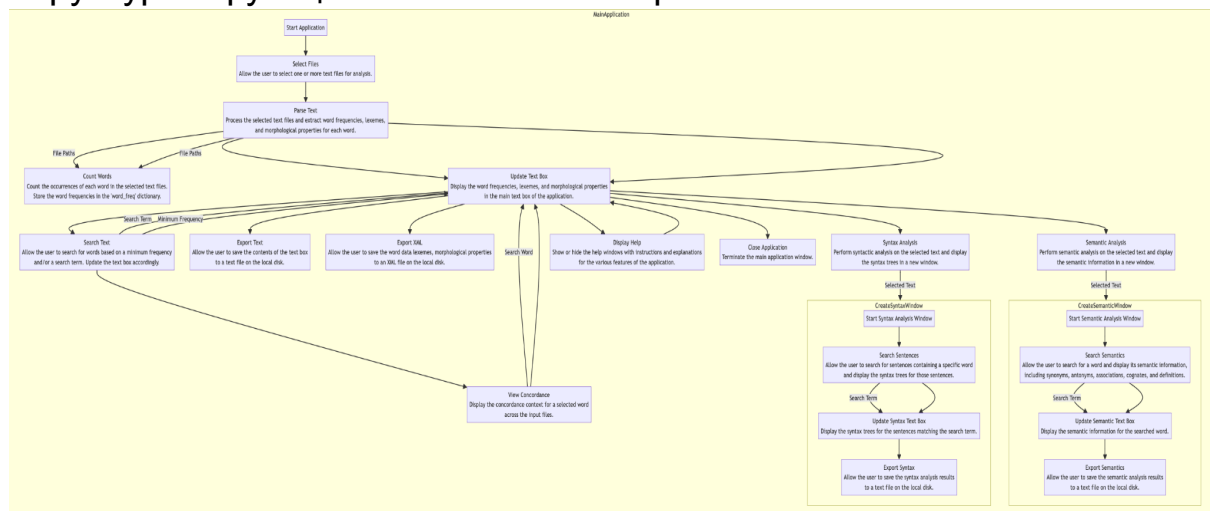
10) Программа формирует окно с семантическим анализом. Здесь же можно вводить запрос в строку поиска в соответствующее

поле, и, нажав на кнопку “Найти”, семантический анализ обновится.



11) Программа завершает работу.

Структурно-функциональная схема приложения:



Использованные структуры хранения данных:

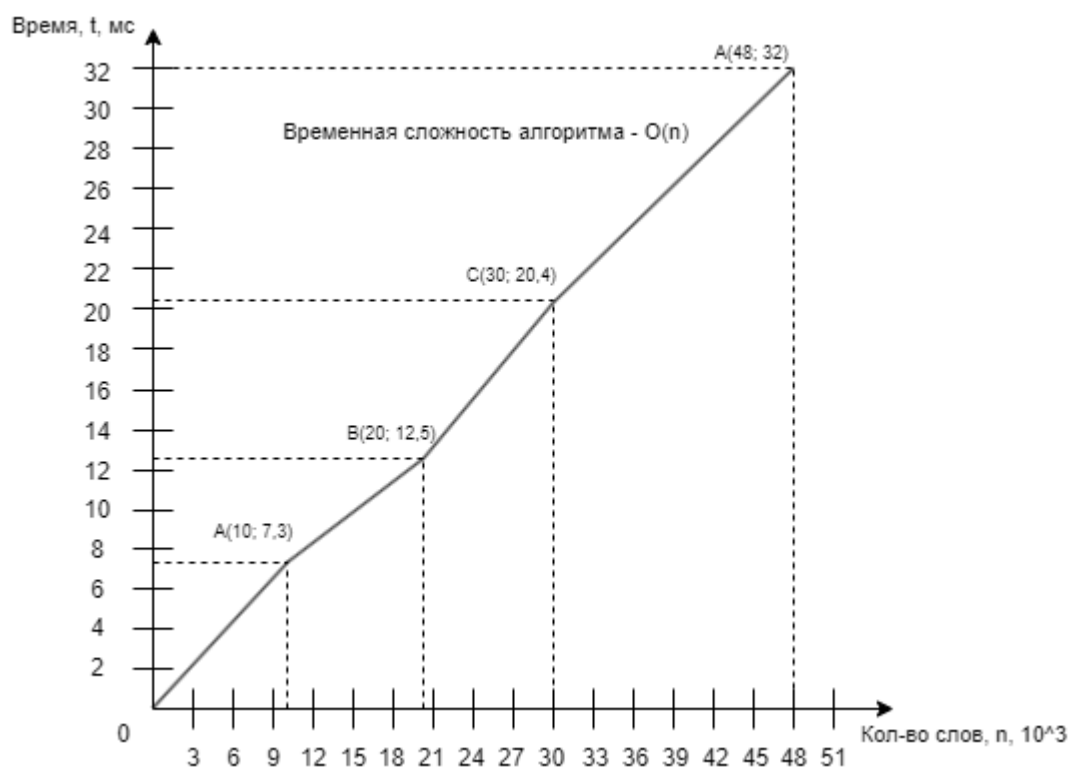
Выбрав функцию “Семантический анализ” программа, используя библиотеку Ruslingua, отправляет запросы для получения различной лингвистической информации о словах, а именно для получения антонимов, синонимов, словесных ассоциаций, родственных слов и определений. Для работы с семантическим анализом необходим доступ к интернету.

```
def get_semantic_data(self, word):
    ruslingua = RusLingua()
    synonyms = ruslingua.get_synonyms(word)
    antonyms = ruslingua.get_antonyms(word)
    associations = ruslingua.get_associations(word)
    cognates = ruslingua.get_cognate_words(word)
    definitions = self.get_word_definition(word)

    return synonyms, antonyms, associations, cognates, definitions

def search_sentences():
    search_text = search_entry.get().strip().lower()
    text_area.delete('1.0', tk.END)
    if (search_text != ''):
        words = self.lemmatize_sentence(text, search_text)
        for word in words:
            synonyms, antonyms, associations, cognates, definitions = self.get_semantic_data(word)
            text_area.insert(tk.END, f'Слово:\n {word}\n')
            text_area.insert(tk.END, f'Синонимы:\n {synonyms}\n')
            text_area.insert(tk.END, f'Антонимы:\n {antonyms}\n')
            text_area.insert(tk.END, f'Когнаты:\n {cognates}\n')
            text_area.insert(tk.END, f'Определение:\n {definitions}\n')
            text_area.insert(tk.END, f'Ассоциации:\n {associations}\n')
            text_area.insert(tk.END, '\n')
    else:
        text_area.insert(tk.END, text)
```

Оценка быстродействия приложения:



Оценка была проведена путем вывода времени программы после ее завершения. В зависимости от количества слов на график было нанесено несколько точек, а эти точки были соединены прямыми линиями.

Вывод: Были изучены принципы разработки прикладных сервисных программ для решения задачи автоматического семантико-синтаксического анализа текста естественного языка (ТЕЯ), в частности познакомились с назначением, структурой и функциональностью, предоставляемой базовым лингвистическим процессором (ЛП), а также закрепили навыки программирования при решении задач автоматической обработки текста естественного языка (ТЕЯ). Разработанное приложение может быть полезно для исследования языка, обучения языковым навыкам, понимании сложных текстов и оптимизации систем обработки языка.