Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

**ОТЧЁТ**

по дисциплине «Естественно-языковой интерфейс интеллектуальных систем»

Лабораторная работа №4

Тема: Семантико-синтаксический анализ текстов естественного языка

Выполнили: Демидовец Д.В.

Козырев Д.А.

гр. 221703

Проверил: Крапивин Ю.Б.

Минск 2025

**Цель работы:**

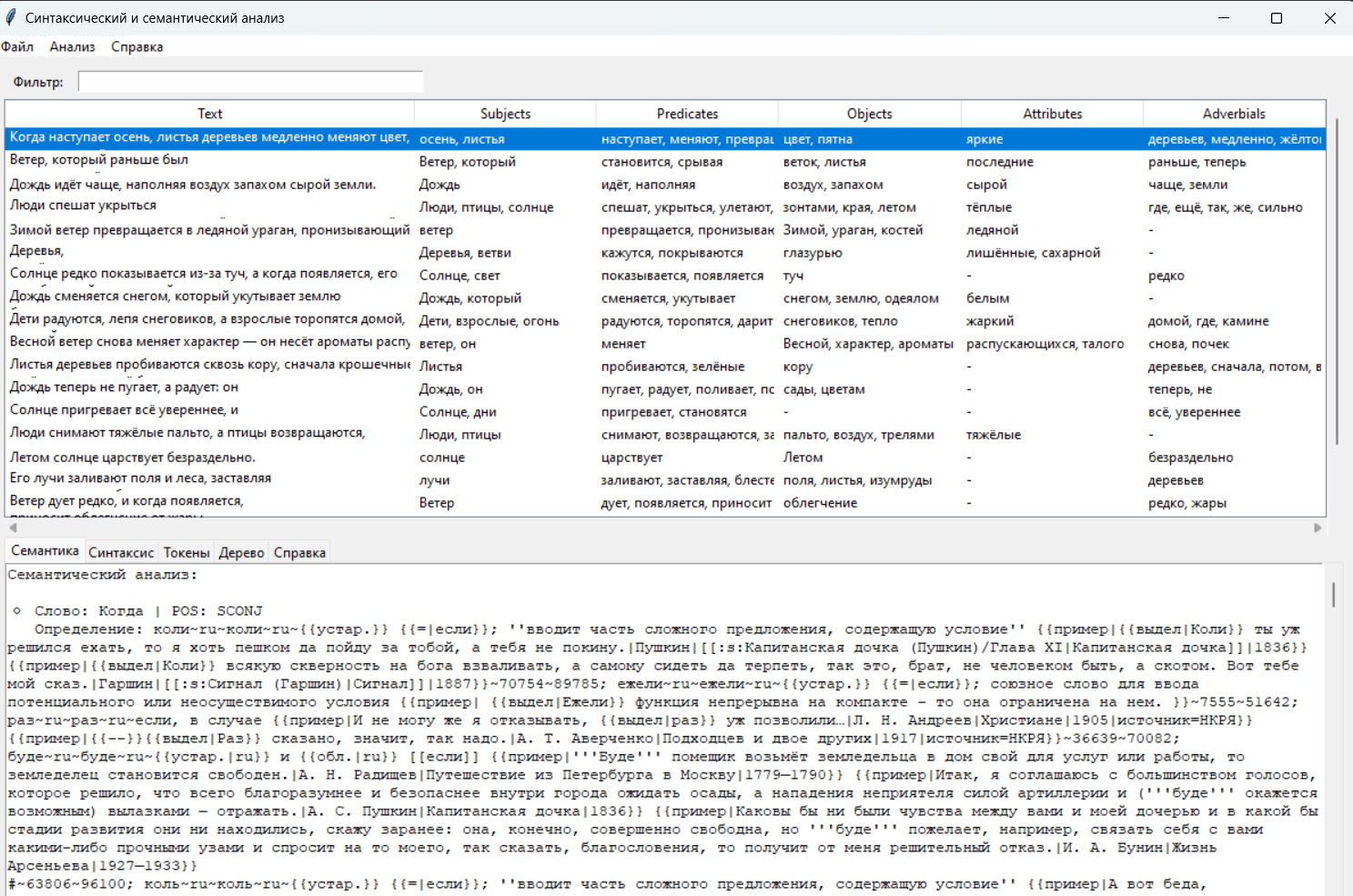
Освоить принципы разработки прикладных сервисных программ для решения задачи автоматического семантико-синтаксического анализа текста естественного языка.

**Задание:**

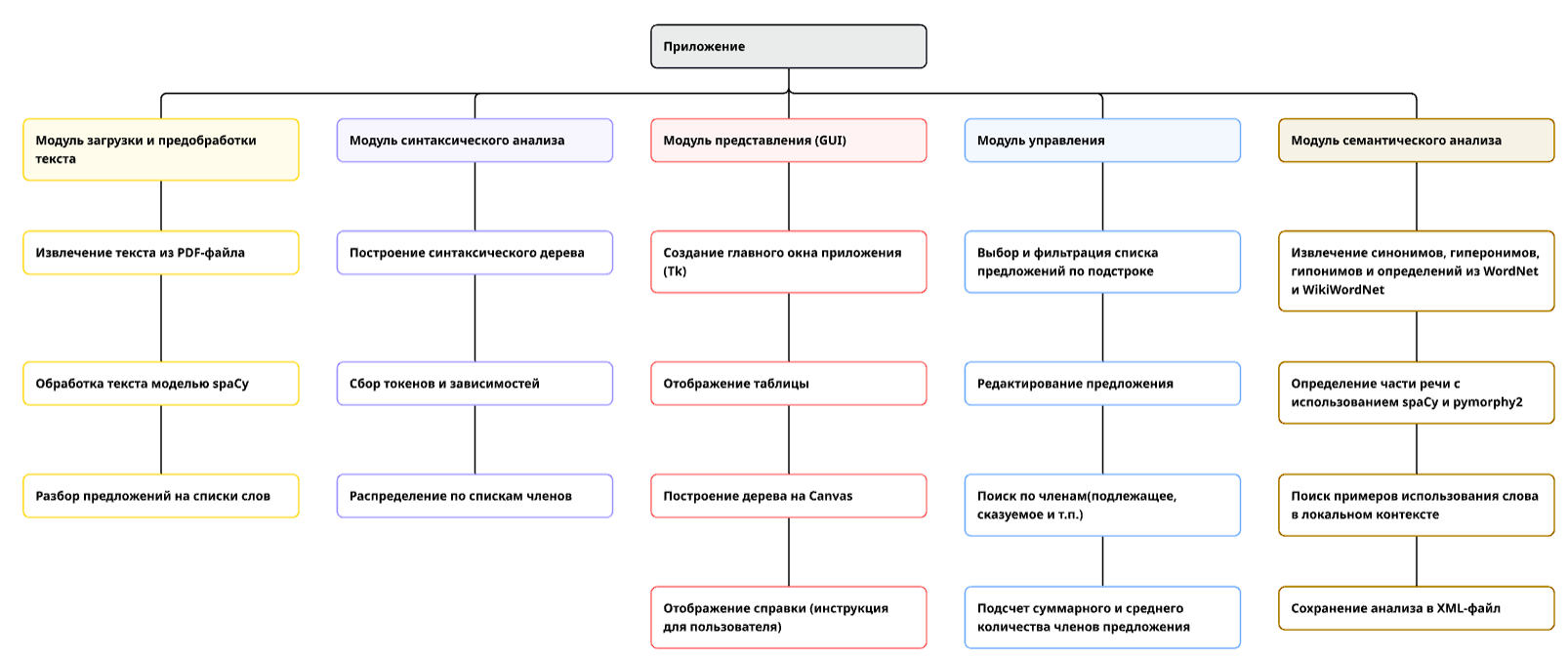
1. Познакомиться с назначением, структурой и функциональностью, предоставляемой базовым ЛП для решения задачи автоматического семантико-синтаксического анализа ТЕЯ.

2. Закрепить навыки программирования при решении задач автоматической обработки ТЕЯ.

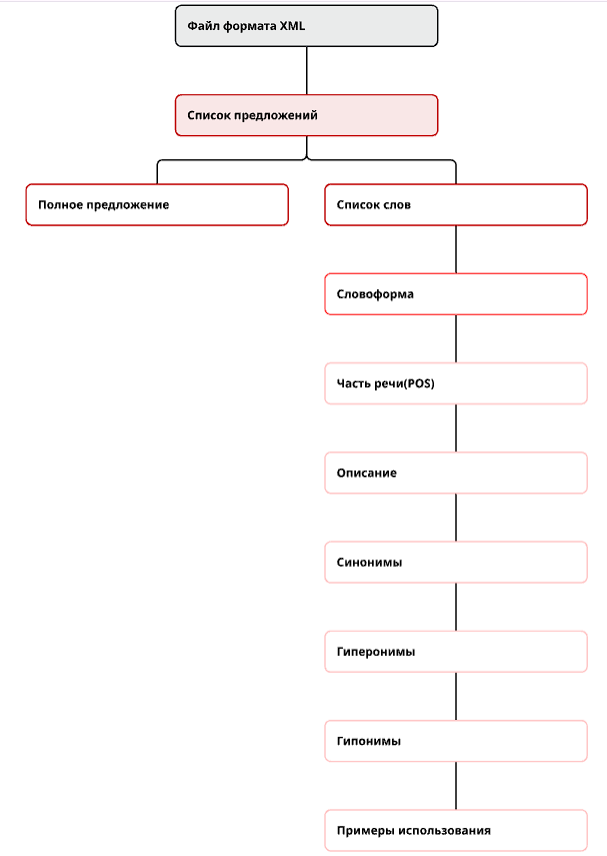
**Интерфейс:**

****

**Структурная схема приложения:**

****

**Структура хранения данных:**

****

**Использованные библиотеки:**

* Библиотека **re** используется для работы с регулярными выражениями — это позволяет фильтровать или проверять текстовые данные, например, отсеивать слова, содержащие знаки препинания или пробелы.
* Библиотека **tkinter** и её модули (ttk, filedialog, messagebox, scrolledtext) используются для построения графического интерфейса программы: создания окон, диалогов выбора файлов, вывода сообщений, прокручиваемых текстовых полей и интерфейсных элементов на основе темы оформления.
* Модуль **xml.etree.ElementTree** (ET) применяется для создания и сохранения результатов семантического анализа в формате XML. Он используется для построения дерева элементов, добавления вложенных тегов и записи готового XML-документа в файл.
* Модуль **pdfminer**.high\_level используется для извлечения текста из PDF-файлов, что позволяет загружать и обрабатывать документы без предварительного копирования их содержимого вручную.
* Библиотека **pymorphy3** используется для морфологического анализа слов русского языка. С её помощью определяются часть речи и грамматические характеристики, что важно для дальнейшей семантической обработки.
* Библиотека **nltk** применяется для лингвистических задач: токенизации, построения моделей на основе WordNet .
* Библиотека **spaCy** используется для токенизации текста, определения частей речи и получения базовых лингвистических характеристик, что необходимо для семантического анализа каждого слова в контексте предложения.
* Библиотека **ruwordnet** (обёртка над русским WordNet) применяется для получения синонимов, гиперонимов и гипонимов из стандартной версии русской WordNet.
* Библиотека **wiki\_ru\_wordnet** расширяет возможности семантического анализа за счёт доступа к более богатой базе данных — WikiWordNet, созданной на основе Википедии, из которой также извлекаются синонимы, гиперонимы, гипонимы и определения слов.
* Библиотека **BeautifulSoup** используется для парсинга HTML-страниц, извлечения нужных элементов из веб-контента при загрузке текстов или определений с сайтов.

**Описание алгоритмов:**

**1. Обработка текста из файла:**

Основной алгоритм для извлечения текста из PDF реализован в функции extract\_info:

1. Начало.
2. Пользователь выбирает PDF через диалоговое окно (tkinter.filedialog).
3. Функция extract\_text() из pdfminer.high\_level считывает текстовое содержимое файла.
4. Текст разбивается на строки, удаляются пустые и лишние пробелы.
5. Очищенный текст передаётся в NLP-пайплайн spaCy для дальнейшей обработки.
6. Конец.

**2. Синтаксическая NLP-обработка текста:**

1. Начало.
2. Очищенный текст (уже извлечённый из PDF) передаётся в языковую модель spaCy.
3. Создаётся объект doc, содержащий информацию о всём тексте.
4. С помощью встроенного механизма doc.sents текст разбивается на отдельные предложения.
5. Для каждого предложения выполняется:
   1. Токенизация (разделение на отдельные слова).
   2. Лемматизация (приведение слова к начальной форме).
   3. Определение частей речи.
   4. Анализ морфологических признаков (число, род, падеж и т.д.).
   5. Построение синтаксических зависимостей между словами.
6. Из каждого предложения выделяются подлежащее, сказуемое, дополнения и обстоятельства на основе синтаксических связей.
7. Сохранённые данные используются для отображения в таблице и дальнейшего анализа.
8. Конец.

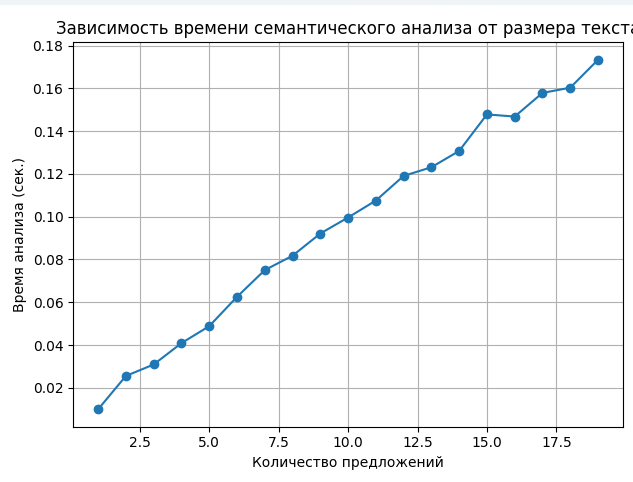
**3. Семантический анализ текста:**

1. Начало.
2. Пользователь выбирает предложение для анализа из общего списка.
3. Для каждого слова в выбранном предложении выполняются следующие действия:  
   a. Определяется лемма слова с помощью морфологического анализатора pymorphy3.  
   b. С помощью spaCy определяется часть речи слова.  
   c. Выполняется извлечение синонимов из RuWordNet и WikiWordNet, исключая совпадения с самим словом.  
   d. Извлекаются гиперонимы (слова более общего значения) из обеих версий WordNet.  
   e. Извлекаются гипонимы (слова более узкого значения) из обеих версий WordNet.  
   f. Получаются определения слова из WikiWordNet.  
   g. Производится поиск примеров употребления слова в локальном контексте — среди уже загруженных предложений.
4. Полученная семантическая информация (синонимы, гиперонимы, гипонимы, часть речи, определение, примеры) сохраняется в структуру словаря.
5. Формируется XML-структура (ElementTree), в которой создаётся корневой элемент <SemanticAnalysis> с вложенными тегами <Word> для каждого слова, содержащими всю семантическую информацию.
6. XML-структура сохраняется в атрибут объекта для возможного экспорта.
7. Конец.

**4. Редактирование предложения**

1. Начало.
2. Пользователь двойным щелчком выбирает предложение в таблице.
3. Открывается новое окно tkinter, где текст можно изменить вручную.
4. Пользователь вносит изменения и нажимает "Сохранить".
5. Обновлённое предложение сохраняется в основной список.
6. Повторно выполняется синтаксическая обработка редактированного предложения.
7. В таблице и на холсте отображаются актуальные данные.
8. Закрытие окна редактирования.
9. Конец.

**Тест производительности:**

****

В ходе эксперимента было проведено измерение времени выполнения семантического анализа при обработке текстов с различным количеством предложений. Для тестирования использовался один исходный текст размером на 20 предложений.

Полученный график демонстрирует линейную зависимость: с ростом количества предложений увеличивается и общее время обработки текста. Такая тенденция объясняется тем, что для каждого предложения выполняется последовательная обработка — токенизация, определение частей речи, описания, синонимов, гиперонимов и прочего. Эти операции имеют линейную временную сложность, что приводит к пропорциональному увеличению времени анализа при увеличении объёма текста.

**Вывод:**

В ходе выполнения лабораторной работы было разработано приложение для семантико-синтаксического анализа текста на русском языке, позволяющее обрабатывать документ формата .pdf и автоматически извлекать информацию о каждом слове. В результате работы программа формирует словарь, содержащий предложения и слова, из которых они состоят, а также основную информацию о словах: часть речи, описание, место в предложении и другие характеристики. Разработанная структура хранения данных обеспечивает наглядное и удобное представление информации, что значительно упрощает последующую обработку и анализ текста. В перспективе разработанный функционал можно расширять и адаптировать для других языков, что увеличит область применения в образовательной, исследовательской и редакторской практике.