Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

За шестой семестр

По дисциплине: «Естественно-языковой интерфейс ИС»

# Тема: «Разработка автоматизированной системы формирования словаря естественного языка»

Выполнил:

Студент 3 курса

Группы ИИ-21(II)

Литвинюк Т. В.

Проверила:

Якимук А. В.

Брест 2024

**Цель:** освоить принципы разработки прикладных сервисных программ для решения задачи автоматического лексического и лексико-грамматического анализа текста естественного языка.

**Ход работы:**

**Вариант 8**

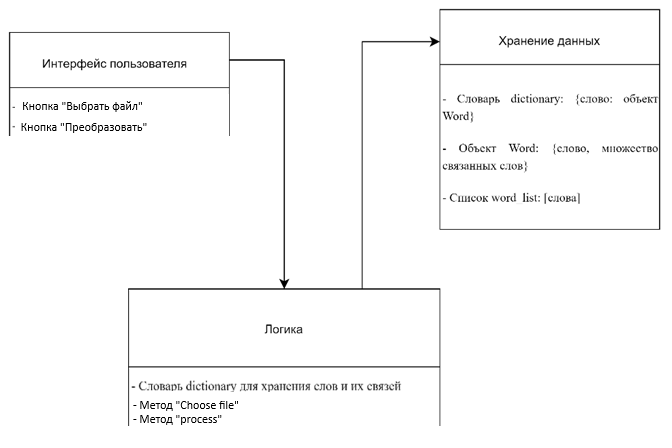


**Задание 3.** Список слов, упорядоченный по алфавиту и включающий только лексемы с дополнительно оформленными записями для образования словосочетаний. В этих записях должны храниться слова, с которыми данное слово может сочетаться. При этом возможно обеспечение автоматизированного извлечения из исходных текстов типовых словосочетаний с их последующей обработкой. \*Можно добавить средства синтеза словосочетаний, но при этом в словарь необходимо будет добавить правила формирования словоформ (см. задание 2).

**Методические указания:**

Требуется спроектировать и программно реализовать структуры хранения данных, алгоритмы их обработки, необходимые в рамках следующих базовых требований к разрабатываемому приложению:

* входные данные – текст заданного естественного языка;
* выходные данные – перечень лексем с дополнительной информацией согласно задания;
* взаимодействие с пользователем посредствам графического интерфейса (интерфейс должен быть интуитивно-понятным и дружественным пользователю);
* наличие системы средств помощи пользователю;
* обеспечение возможности построения, сохранения, просмотра, редактирования, пополнения, фильтрации и поиска по заданному условию, документирования автоматически получаемого словаря либо заданной его части;
* поддержка различных форматов представления входных данных (TXT, RTF, PDF, DOC, DOCX).



Листинг:

from nltk.tokenize import word\_tokenize

from nltk.tag import pos\_tag

from collections import defaultdict

from striprtf.striprtf import rtf\_to\_text

import tkinter as tk

from tkinter import ttk, filedialog

def analyze\_text(*input\_text*):

    words = word\_tokenize(input\_text.lower())

    # Определяем части речи слов

    tagged\_words = pos\_tag(words)

    word\_combinations = defaultdict(list)

    for i in range(len(tagged\_words) - 1):

        word, pos = tagged\_words[i]

        next\_word, next\_pos = tagged\_words[i+1]

        # Учитываем только существительные и глаголы

        if pos.startswith('N') and next\_pos.startswith(('N', 'V')):

            word\_combinations[word].append(next\_word)

    result = ""

    for word, combinations in sorted(word\_combinations.items()):

        result += f"{word}: {', '.join(combinations)}\n"

    return result

def display\_scrollable\_text(*text*):

    # Создаем главное окно tkinter

    root = tk.Tk()

    root.title("Scrollable Text")

    # Создаем текстовое поле

    text\_area = tk.Text(root, *wrap*="word", *height*=20, *width*=60)

    text\_area.insert(tk.END, text)

    text\_area.pack(*side*=tk.LEFT, *fill*=tk.BOTH, *expand*=True)

    # Создаем вертикальную полосу прокрутки

    scrollbar = tk. Scrollbar(root, *command*=text\_area.yview)

    scrollbar.pack(*side*=tk.RIGHT, *fill*=tk.Y)

    text\_area.config(*yscrollcommand*=scrollbar.set)

    # Запускаем главный цикл обработки событий

    root.mainloop()

def main():

    root = tk.Tk()

    file\_path = tk.StringVar(root)

    def ask\_file\_path():

        path = filedialog.askopenfilename()

        if path:

            file\_path.set(path)

    file\_path\_btn = tk.Button(root, *text*="Выбрать файл", *command*=ask\_file\_path)

    file\_path\_btn.pack()

    def convert():

        if file\_path.get():

            try:

                with open(file\_path.get(), 'r', *encoding*='utf-8') as file:

                    text = file.read()

            except FileNotFoundError:

                print("Файл не найден.")

            if file\_path.get().endswith('.rtf'):

                text = rtf\_to\_text(text)

            analysis\_result = analyze\_text(text)

            display\_scrollable\_text(analysis\_result)

    convert\_btn = tk.Button(root, *text*="Преобразовать", *command*=convert)

    convert\_btn.pack()

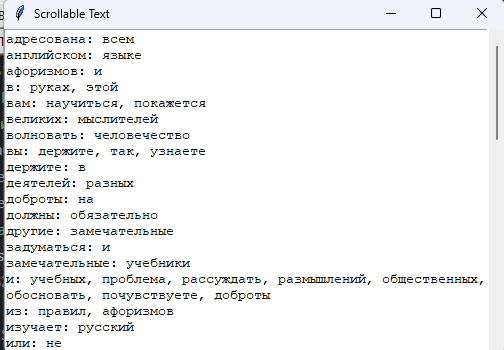
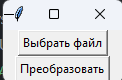
    root.mainloop()

    # file\_path = 'EAI/LAB1/file.txt'  #input("Введите путь к файлу с текстом: ")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Результат:



**Вывод:** в ходе выполнения лабораторной работы освоил принципы разработки прикладных сервисных программ для решения задачи автоматического лексического и лексико-грамматического анализа текста естественного языка.