Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «Естественно-языковой интерфейс ИС»

# Тема: «Разработка автоматизированной системы синтаксического анализа текста естественного языка»

Выполнил:

Студент 3 курса

Группы ИИ-21

Ливтинюк Т. В.

Проверила:

Якимук А. В.

Брест 2024

**Цель:** Освоить принципы разработки прикладных сервисных программ для решения задачи анализа текста естественного языка.

**Ход работы:**

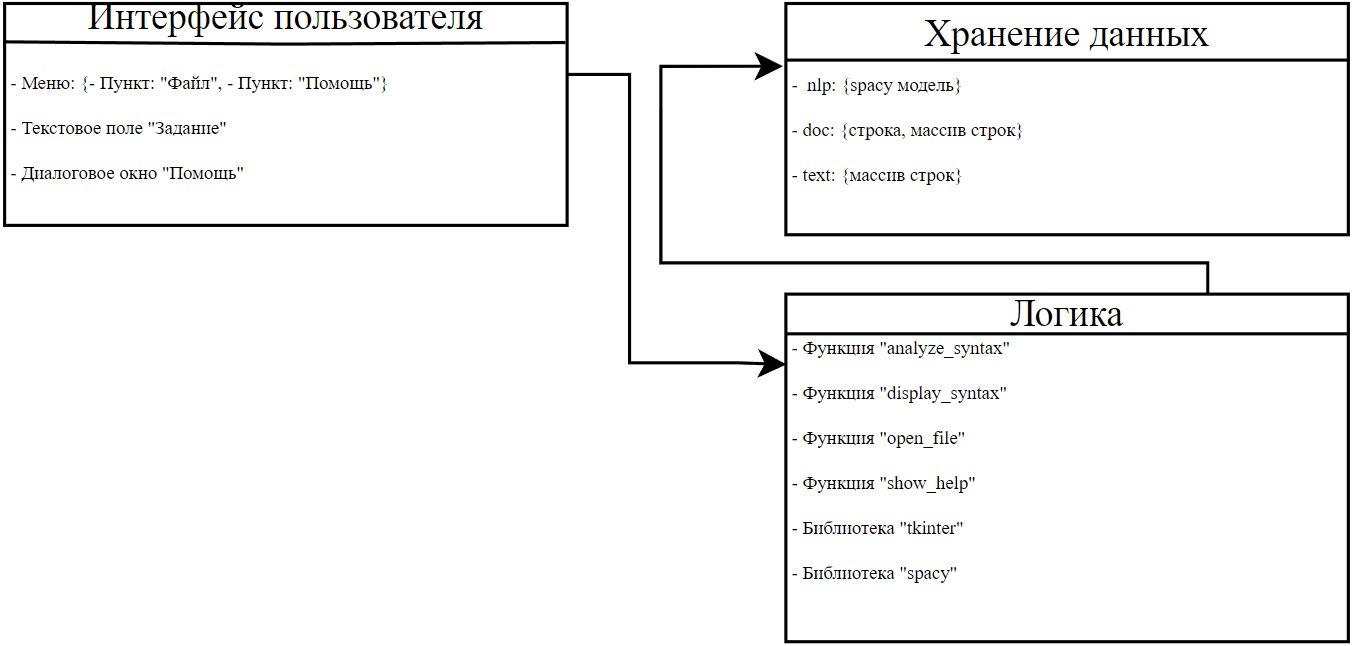
|  |  |  |
| --- | --- | --- |
| **№** | **Язык текста** | **Формат входного документа** |
| 7 | Английский | DOCX |

**Методические указания:**

Требуется спроектировать и программно реализовать структуры хранения данных, алгоритмы их обработки (в виде подключаемой библиотеки функций), необходимые в рамках следующих базовых требований к разрабатываемому приложению:

* входные данные – текст заданного естественного языка;
* выходные данные – структуры, полученные при проведении автоматического синтаксического анализа предложений входного текста согласно варианту 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, документирования автоматически получаемого результата либо заданной его части;
* поддержка различных форматов представления входных данных (TXT, RTF, PDF, HTML, DOC, DOCX).

**Структурно-функциональная схема:**



**Листинг:**

import tkinter as tk

import docx2txt

# Функция вычисления редакционного расстояния

def edit\_distance(str1, str2):

m = len(str1) + 1

n = len(str2) + 1

t = [[i + j for j in range(n)] for i in range(m)]

for i in range(1, m):

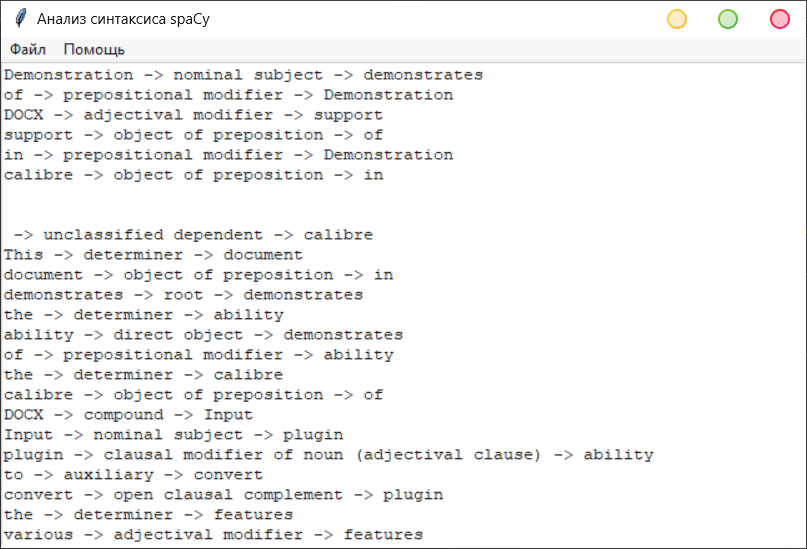
c = i - 1

for j in range(1, n):

d = j - 1

t[i][j] = min(t[c][j] + 1, t[i][d] + 1, t[c][d] + (str1[c] != str2[d]))

return t[m - 1][n - 1]



root = tk.Tk()

root.title("Редакционное расстояние")

def load\_text(filename):

text = docx2txt.process(filename)

return text

def get\_words(text):

words = text.split()

return words

text = load\_text('EAI/LAB3/text.docx')

words = get\_words(text)

def calculate\_distances():

input\_word = input\_entry.get()

max\_distance = int(max\_distance\_entry.get())

results = []

for word in words:

distance = edit\_distance(input\_word, word)

if distance <= max\_distance:

results.append((word, distance))

results.sort(key=lambda x: x[1], reverse=True)

output\_text.delete("1.0", tk.END)

for word, distance in results:

output\_text.insert(tk.END, f"{word} ({distance})\n")

# Интерфейс

input\_label = tk.Label(root, text="Введите входное слово:")

input\_entry = tk.Entry(root, width=100)

max\_distance\_label = tk.Label(root, text="Введите максимальное расстояние:")

max\_distance\_entry = tk.Entry(root, width=100)

calculate\_button = tk.Button(root, text="Вычислить", command=calculate\_distances)

output\_label = tk.Label(root, text="Результаты:")

output\_text = tk.Text(root, height=10)

input\_label.pack()

input\_entry.pack()

max\_distance\_label.pack()

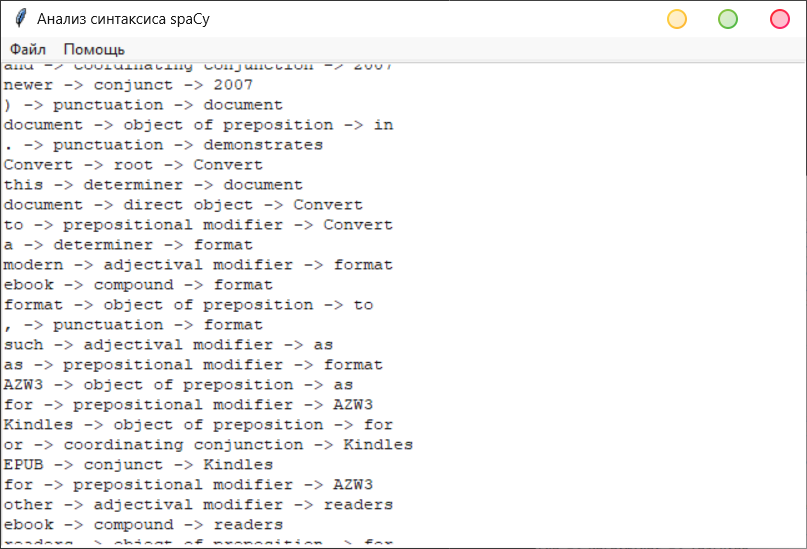
max\_distance\_entry.pack()

calculate\_button.pack()

output\_label.pack()

output\_text.pack()

root.mainloop()



**Вывод:** в ходе выполнения лабораторной работы освоил принципы разработки прикладных сервисных программ для решения задачи анализа текста естественного языка.