Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: «Естественно-языковой интерфейс ИС»

# Тема: «Реализация алгоритмов автоматического морфологического и лексико-грамматического анализа текста естественного языка»

Выполнил:

Студент 3 курса

Группы ИИ-21

Романко Н. А.

Проверила:

Якимук А. В.

Брест 2024

**Цель:** закрепить знания морфологического и лексико-грамматического анализа текста.

**Ход работы:**

|  |  |  |
| --- | --- | --- |
| **Номер**  **варианта** | **Язык текста** | **Формат документа** |
| 4 | Арабский | Html |

**Задание 1.** Создать программную реализацию решения задачи о редакционном расстоянии методом динамического программирования между словами, подаваемыми на вход системы и словами текста.

**Задание 2.** Спроектируйте внешний интерфейс автоматизированной системы осуществляющей решения задачи о редакционном расстоянии между словами, подаваемыми на вход системы и словами текста.

**Задание 3.** Разработайте алгоритмы сервисных функций, предоставляемых Вашей системой.

**Задание 4.** Реализуйте разработанные алгоритмы в виде программ на языке программирования.

**Задание 5.** Протестируйте разработанную систему, используя различные входные тексты.

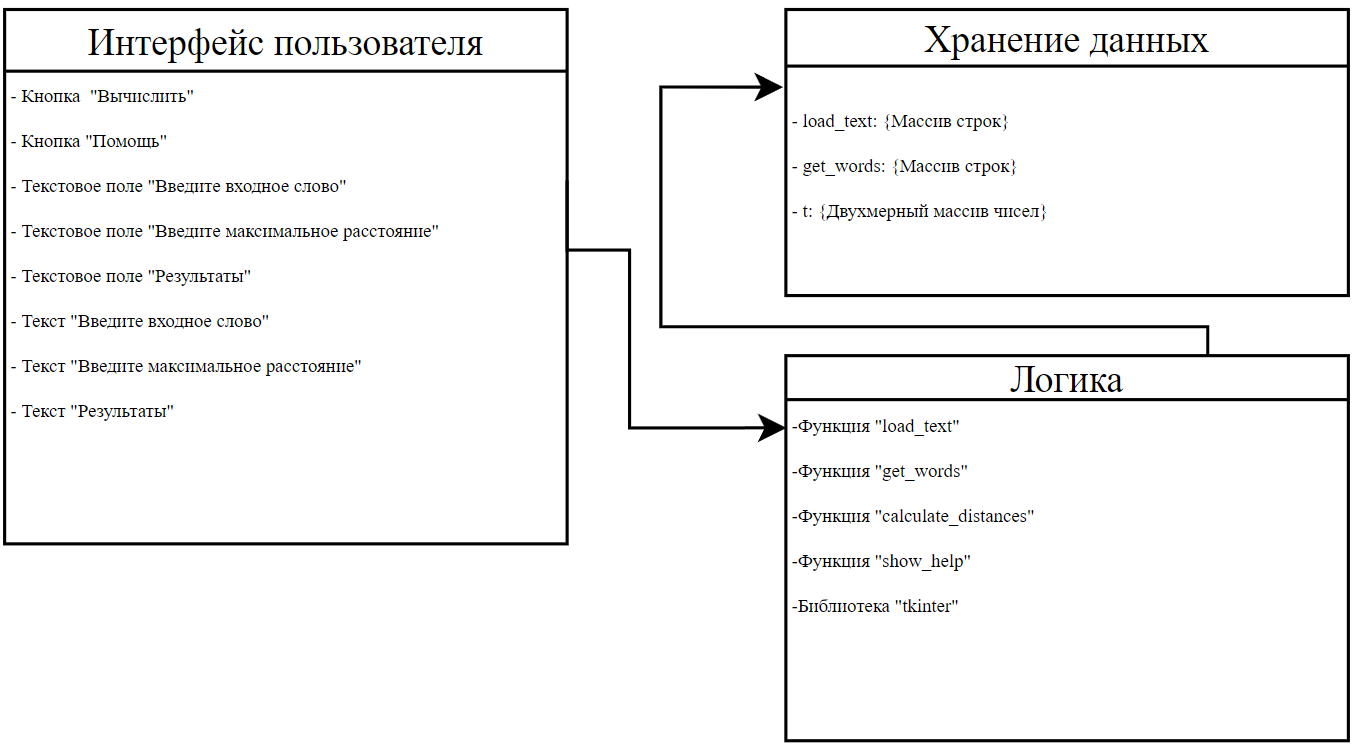
**Задание 6.** Разработайте и внедрите в систему средства помощи пользователю.

**Задание 7.** Протестируйте систему, включающую средства помощи пользователю.

**Методические указания:**

* на входе – слова естественно-языкового текста содержащие орфографические ошибки;
* на выходе – перечень слов, расстояние до которых от входного слова не превышает некоторого заданного Вами значения, упорядоченный по убыванию величины расстояния редактирования;
* интерфейс системы должен быть предельно простым и доступным для пользователей любого уровня, содержать понятный набор инструментов и средств, а также help-средства;
* наличие средств распечатки полученной на выходе информации.

**Структурно-функциональная схема:**



**Листинг:**

import re

import html

from collections import defaultdict

from Levenshtein import distance

import tkinter as tk

from tkinter import ttk, filedialog

def load\_html\_file(file\_path):

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

return file.read()

def preprocess\_text(text):

cleaned\_text = re.sub('<[^<]+?>', '', text)

cleaned\_text = html.unescape(cleaned\_text)

return cleaned\_text

def find\_similar\_words(input\_word, text, max\_distance):

similar\_words = defaultdict(list)

words = text.split()

for word in words:

dist = distance(input\_word, word)

if dist <= max\_distance:

similar\_words[dist].append(word)

return similar\_words

def choose\_file():

file\_path = filedialog.askopenfilename(filetypes=[("HTML files", "\*.html")])

if file\_path:

label\_file.config(text=file\_path)

def search():

input\_word = entry\_word.get()

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

html\_file\_path = label\_file.cget("text")

if not html\_file\_path:

tk.messagebox.showerror("Ошибка", "Выберите HTML файл")

return

html\_text = load\_html\_file(html\_file\_path)

cleaned\_text = preprocess\_text(html\_text)

similar\_words = find\_similar\_words(input\_word, cleaned\_text, max\_distance)

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

for dist in sorted(similar\_words.keys()):

output\_text.insert(tk.END, f"Редакционное расстояние: {dist}\n")

for word in similar\_words[dist]:

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

output\_text.insert(tk.END, "\n")

root = tk.Tk()

root.title("Поиск слов, или чет такое, 3-я лаба крч")

frame = ttk.Frame(root)

frame.pack(padx=10, pady=10)

label\_word = ttk.Label(frame, text="Введите слово:")

label\_word.grid(row=0, column=0, sticky="w")

entry\_word = ttk.Entry(frame, width=30)

entry\_word.grid(row=0, column=1, padx=5, pady=5)

label\_distance = ttk.Label(frame, text="Максимальное расстояние:")

label\_distance.grid(row=1, column=0, sticky="w")

entry\_distance = ttk.Entry(frame, width=10)

entry\_distance.grid(row=1, column=1, padx=5, pady=5)

button\_choose\_file = ttk.Button(frame, text="Выбрать файл", command=choose\_file)

button\_choose\_file.grid(row=2, column=0, columnspan=2, pady=5)

label\_file = ttk.Label(frame, text="Выбранный файл: ")

label\_file.grid(row=3, column=0, columnspan=2, sticky="w")

button\_search = ttk.Button(frame, text="Поиск", command=search)

button\_search.grid(row=4, column=0, columnspan=2, pady=10)

output\_text = tk.Text(frame, width=50, height=20)

output\_text.grid(row=5, column=0, columnspan=2, padx=5, pady=5)

root.mainloop()

# Результат:

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

**Вывод:** в ходе выполнения лабораторной работы закрепил знания морфологического и лексико-грамматического анализа текста.