Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине: «Естественно-языковой интерфейс ИС»

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

Выполнил:

Студент 3 курса

Группы ИИ-21

Романко Н. А.

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:**

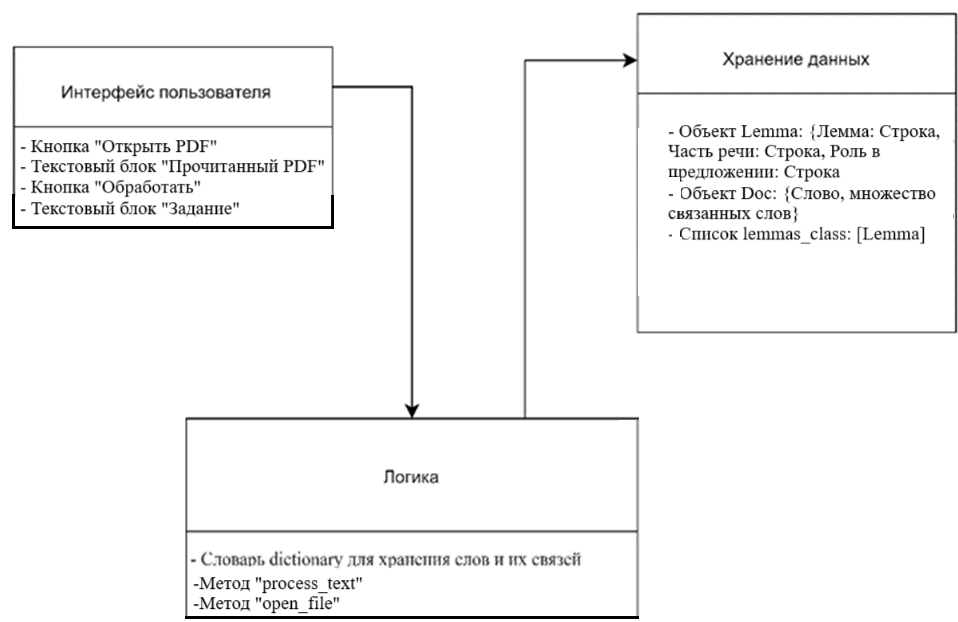
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Язык текста** | **Формат входного документа** | **Вариант задания** |
| 14 | Английский | PDF | Задание 1 |

**Задание 4.** Список слов, упорядоченный по алфавиту и включающий как лексемы, так и словоформы, с указанием частоты встречаемости каждой из форм. Для словоформ пользователю должна быть предоставлена возможность вводить дополнительную морфологическую информацию, а именно, отнесение слова к соответствующей части речи, указание рода, числа, падежа и т.п. При этом морфологическая информация может быть оформлена как отдельная неформатированная запись, т.е. это просто текст, который пользователь может оформлять произвольным образом.

**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 заданию;
* 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:**

import tkinter as tk

from tkinter import filedialog, ttk

import PyPDF2

from nltk.tokenize import word\_tokenize

from nltk.probability import FreqDist

from nltk import pos\_tag

from nltk.corpus import stopwords

# Перевод на русский

pos\_translation = {

...

}

class PDFAnalyzerApp:

    def \_\_init\_\_(self, master):

        self.master = master

        self.master.title("KnowYourPDF")

        self.label = ttk.Label(master, text="KnowYourPDF", font=("Helvetica", 16))

        self.label.pack(pady=10)

        self.open\_button = ttk.Button(master, text="Выбрать PDF файл", command=self.open\_pdf)

        self.open\_button.pack(pady=5)

        self.filename\_label = ttk.Label(master, text="", font=("Helvetica", 10))

        self.filename\_label.pack(pady=5)

        self.frame = ttk.Frame(master)

        self.frame.pack(fill=tk.BOTH, expand=True)

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

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

        self.scrollbar = ttk.Scrollbar(self.frame, orient="vertical", command=self.text\_area.yview)

        self.scrollbar.pack(side=tk.RIGHT, fill="y")

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

    def open\_pdf(self):

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

        if file\_path:

            self.filename\_label.config(text=f"Выбранный файл: {file\_path}")

            self.read\_pdf(file\_path)

    def read\_pdf(self, file\_path):

        with open(file\_path, "rb") as f:

            pdf\_reader = PyPDF2.PdfReader(f)

            text = ""

            for page in pdf\_reader.pages:

                text += page.extract\_text()

        self.analyze\_text(text)

    def analyze\_text(self, text):

        words = word\_tokenize(text)

        stop\_words = set(stopwords.words("english"))

        filtered\_words = [word.lower() for word in words if word.isalnum() and word.lower() not in stop\_words]

        fdist = FreqDist(filtered\_words)

        sorted\_fdist = sorted(fdist.items())

        self.text\_area.delete(1.0, tk.END)

        for word, freq in sorted\_fdist:

            tagged\_word = pos\_tag([word])[0]

            self.text\_area.insert(tk.END, f"{freq}: {word} - {tagged\_word[1]}\n")

def main():

    root = tk.Tk()

    app = PDFAnalyzerApp(root)

    root.mainloop()

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

    main()

# Результат:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

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