Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: «Естественно-языковой интерфейс ИС»

# Тема: «Разработка автоматизированной системы диалогового взаимодействия с пользователем на естественном языке»

Выполнил:

Студент 3 курса

Группы ИИ-21

Карагодин Д. Л.

Проверила:

Якимук А. В.

Брест 2024

**Цель:** освоить принципы разработки диалоговых систем с поддержкой естественного языка.

**Ход работы:**

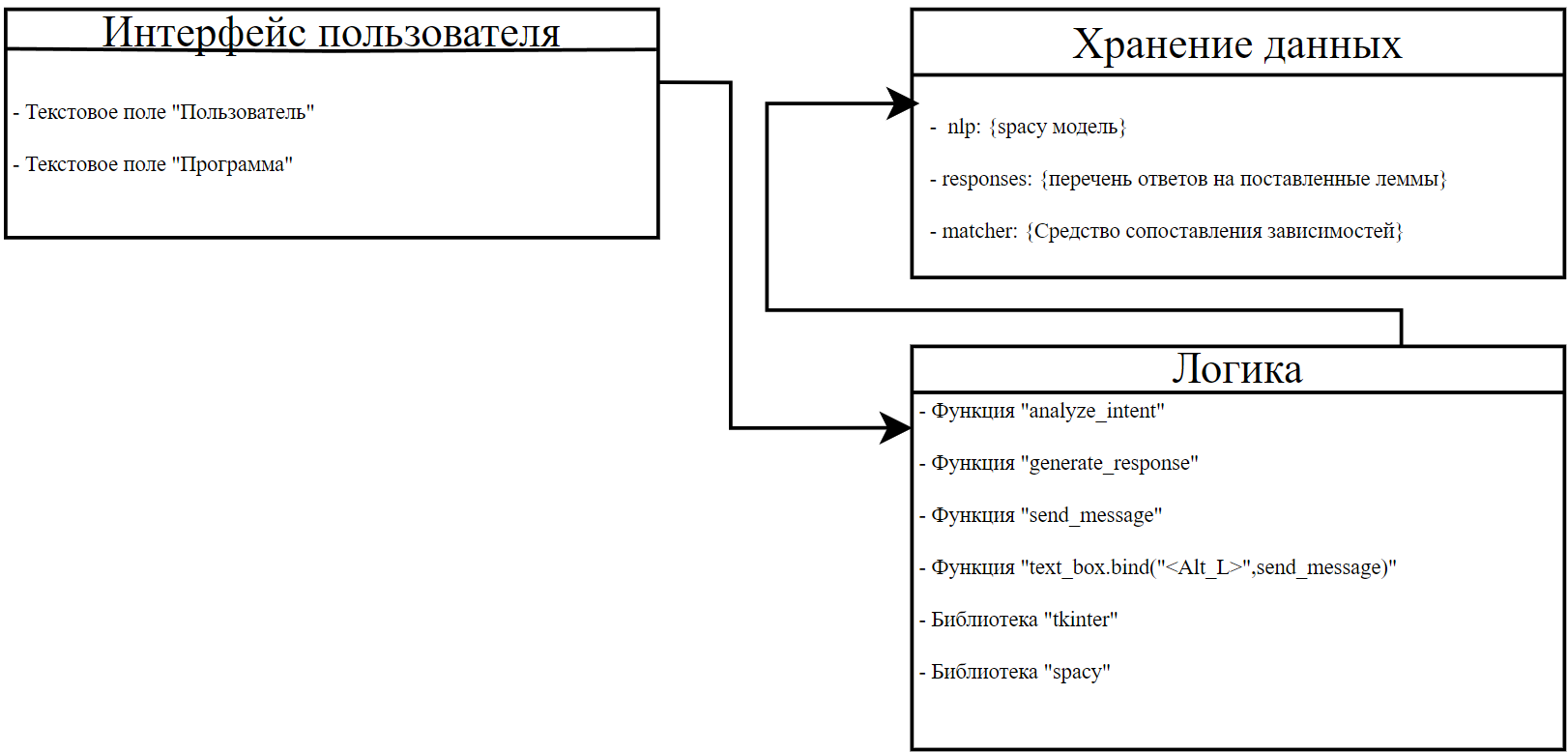
|  |  |  |
| --- | --- | --- |
| **Номер**  **варианта** | **Язык текста** | **Предметная область** |
| 11 | Русский | Услуги |

**Методические указания:**

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

* входные данные – текстовое сообщение на заданном естественном языке;
* выходные данные – автоматическая реакция системы на входное сообщение на естественном языке путем формирования ответного сообщения согласно варианту задания;
* взаимодействие с пользователем посредством графического интерфейса (интерфейс должен быть интуитивно понятным и дружественным пользователю);
* наличие системы средств помощи пользователю;
* обеспечение возможности ведения диалога с пользователем на естественном языке, сохранения, просмотра, редактирования истории диалога либо заданной его части.

**Структурно-функциональная схема:**



**Листинг:**

import tkinter as tk

from tkinter import filedialog

import spacy

from spacy.matcher import Matcher

# Загрузить модель spaCy для русского языка

nlp = spacy.load("ru\_core\_news\_md")

# Создать сопоставитель для поиска намерений пользователя

matcher = Matcher(nlp.vocab)

responses = {

    "услуга": "Я могу предложить вам следующие услуги:",

    "консультация": "Я могу проконсультировать вас по следующим вопросам:",

    "консультировать": "Я могу проконсультировать вас по следующим вопросам:",

    "жалоба": "Я сожалею, что у вас возникли проблемы. Я могу помочь вам подать жалобу.",

    "жаловать": "Я сожалею, что у вас возникли проблемы. Я могу помочь вам подать жалобу.",

    "предложение": "Спасибо за ваше предложение. Я передам его соответствующему отделу.",

    "предложить": "Спасибо за ваше предложение. Я передам его соответствующему отделу.",

    "благодарность": "Я рад, что смог вам помочь!",

    "благодарить": "Я рад, что смог вам помочь!",

    "отблагодарить": "Я рад, что смог вам помочь!",

    "поблагодарить": "Я рад, что смог вам помочь!",

}

patterns= [{"LEMMA": {"IN": [\*responses.keys()]}}]

matcher.add("lemmas", [patterns])

def analyze\_intent(text):

    # Проанализировать текст

    doc = nlp(text)

    print(doc)

    for token in doc:

        print(token.text,token.pos\_,token.dep\_,token.lemma\_)

    # Найти намерение пользователя

    matches = matcher(doc)

    print(matches)

    # Получить намерение из первого совпадения (если оно есть)

    intent = ""

    if len(matches) > 0 and len(matches[0])>2:

        intent = doc[matches[0][1]].lemma\_

        print(intent)

    return intent

def generate\_response(intent):

    # Получить ответ из словаря ответов на основе намерения

    response = responses.get(intent, "Я не понимаю вашего запроса.")

    return response

def send\_message(event):

    # Получить сообщение пользователя

    user\_message = text\_box.get("1.0", "end-1c")

    # Проанализировать намерение пользователя

    intent = analyze\_intent(user\_message)

    # Сгенерировать ответ системы

    response = generate\_response(intent)

    # Очистить поле ввода

    text\_box.delete("1.0", "end")

    # Отобразить ответ системы

    chat\_history.insert("end", f"\*\*Система:\*\* {response}\n")

# Создать основное окно GUI

root = tk.Tk()

root.title("Автоматический ответчик на естественном языке")

# Создать поле ввода для сообщения пользователя

text\_box = tk.Text(root)

text\_box.bind("<Alt\_L>", send\_message)

text\_box.pack()

# Создать поле вывода для чата

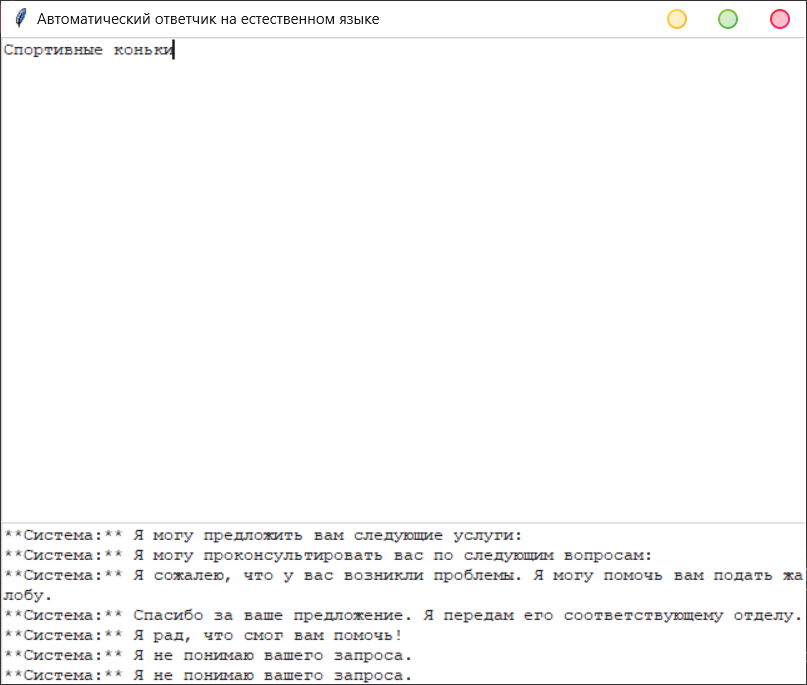
chat\_history = tk.Text(root)

chat\_history.pack()

# Запустить основное окно GUI

root.mainloop()

# Результат:



**Вывод:** в ходе выполнения лабораторной работы освоил принципы разработки диалоговых систем с поддержкой естественного языка.