

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**  
**“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”**

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**ОТЧЁТ**

По лабораторной работе № 6

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

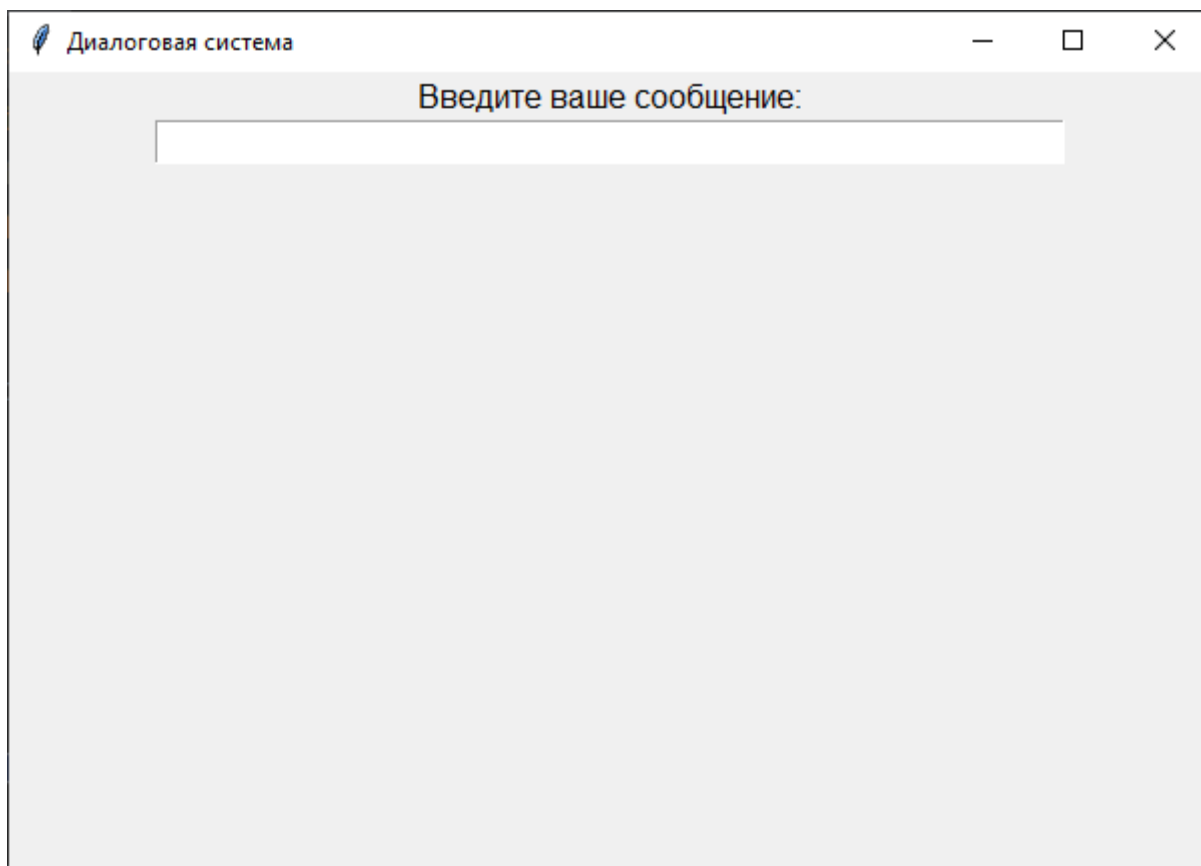
Проверил:

Булей Е. В.

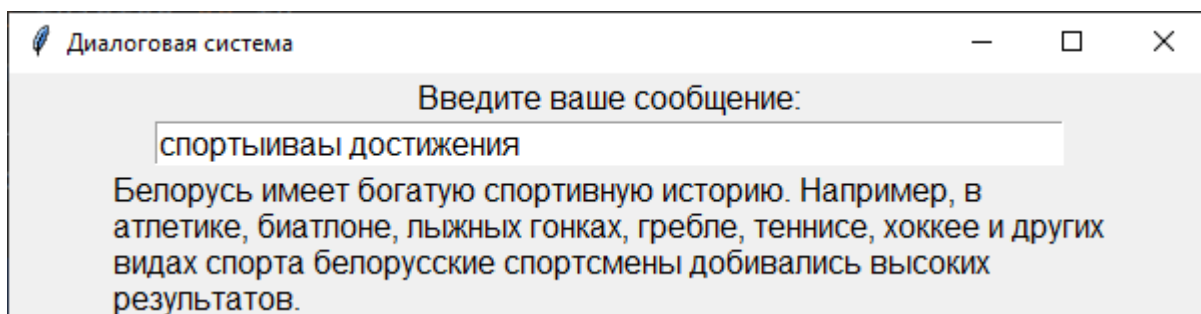
## Ход работы

### Задание:

1. *Входные данные* – текстовое сообщение на заданном естественном языке
2. *Выходные данные* – автоматическая реакция системы на входное сообщение на естественном языке путем формирования ответного сообщения согласно варианту задания
3. Взаимодействие с пользователем посредством графического интерфейса



*Приложение-чат про белорусский спорт*



Система знает ответы на следующие вопросы и ищет максимально подходящий ответ по двум нормализованным **метрикам** с весовыми коэффициентами:

1. Расстояние Левенштейна (0.6)
2. Вхождения слов (0.4)

**Вопросы:** 'белорусские спортивные достижения', 'знаменитые спортсмены Беларуси', 'белорусский хоккей', 'популярные спортивные мероприятия Беларуси', 'лучшие виды спорта Беларуси', 'спортивные клубы Беларуси'

## Код программы:

```
import tkinter as tk
from nltk.chat.util import reflections
from Levenshtein import distance
from collections import Counter

pairs = [
    ['белорусские спортивные достижения', ['Белорусь имеет богатую спортивную историю. Например, в атлетике, биатлоне, лыжных гонках, гребле, теннисе, хоккее и других видах спорта белорусские спортсмены добивались высоких результатов.']],
    ['знаменитые спортсмены Беларуси', ['Один из наиболее известных белорусских спортсменов - Дарья Домрачева, она стала чемпионкой мира и Олимпийской чемпионкой в биатлоне.']],
    ['белорусский хоккей', ['Хоккей - один из самых популярных видов спорта в Беларуси. Национальная хоккейная команда Беларуси активно выступает на международных турнирах.']],
    ['популярные спортивные мероприятия Беларуси', ['В Беларуси проводятся различные спортивные мероприятия, такие как международные соревнования по лыжным гонкам, биатлону, теннису и многим другим видам спорта.']],
    ['лучшие виды спорта Беларуси', ['Белорусские спортсмены часто достигают высоких результатов в зимних видах спорта, таких как биатлон, лыжные гонки и фигурное катание.']],
    ['спортивные клубы Беларуси', ['В Беларуси существуют множество популярных спортивных клубов, представляющих различные виды спорта. Например, "Динамо" - известный футбольный клуб, а также клубы по хоккею, баскетболу и другим видам спорта.']],
]

class ChatApplication:
    def __init__(self, master):
        self.master = master
        master.title("Диалоговая система")
        self.label = tk.Label(master, text="Введите ваше сообщение:", font=("Arial", 12))
        self.label.pack()
        self.entry = tk.Entry(master, font=("Arial", 12), width=50)
        self.entry.pack()
        self.response_label = tk.Label(master, text="", font=("Arial", 12), wraplength=500, justify="left")
        self.response_label.pack()
        self.entry.bind("<Return>", self.send_message)

    def calculate_similarity(self, user_message):
        closest_pair = None
        max_similarity = 0
        user_message_words = user_message.lower().split()
        user_message_counter = Counter(user_message_words)
        user_message_length = len(user_message_words)

        for pair in pairs:
            question = pair[0]
            question_words = question.lower().split()
            d = distance(question.lower(), user_message.lower())
            common_words = sum((user_message_counter & Counter(question_words)).values())
            normalized_distance = d / max(len(question), len(user_message))
            normalized_common_words = common_words / user_message_length

            similarity = 0.6 * (1 - normalized_distance) + 0.4 * normalized_common_words
            print(question, similarity)
            if similarity > max_similarity:
                max_similarity = similarity
                closest_pair = pair
        return closest_pair

    def send_message(self, event):
        user_message = self.entry.get()
        self.entry.delete(0, tk.END)

        closest_pair = self.calculate_similarity(user_message)
        print(closest_pair)
        if closest_pair:
            response = closest_pair[1][0]
            self.response_label.config(text=response)
        else:
            self.response_label.config(text="Извините, не могу найти подходящий ответ.")

def main():
    root = tk.Tk()
    app = ChatApplication(root)
    root.geometry("600x400")
    root.mainloop()

if __name__ == "__main__":
    main()
```