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:

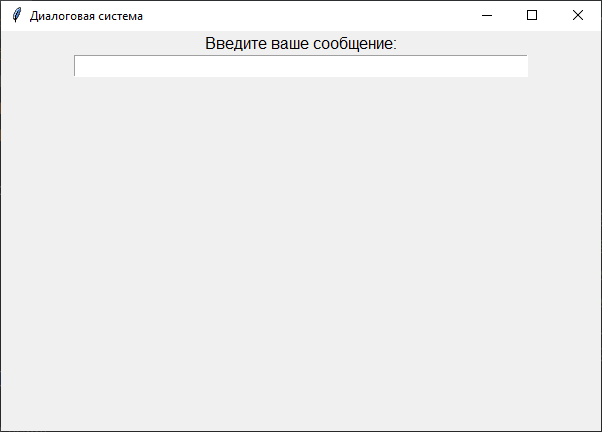
Булей Е. В.

Брест – 2024

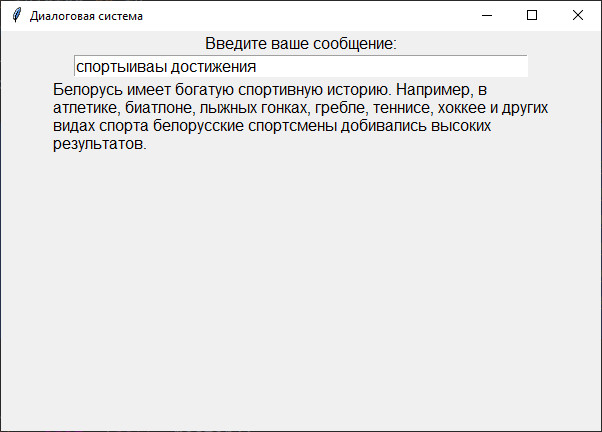
**Ход работы**

**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 Белоруси. Национальная хоккейная команда Белоруси активно выступает на международных турнирах.']],

['популярные спортивные мероприятия Беларуси', ['В Белоруси 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 Белоруси 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()