**main.py**

Необхідні бібліотеки

import numpy as np  
import tkinter as tk  
from tkinter import messagebox

`numpy` -- для числових обчислень

`tkinter` -- для створення графічного інтерфейсу

Визначення трапецієподібної функції належності

# Трапецієвидна функцій належності  
def trapmf(x, a, b, c, d):  
 if a < x < b:  
 return (x - a) / (b - a)  
 elif b <= x <= c:  
 return 1  
 elif c < x < d:  
 return (x - d) / (c - d)  
 else:  
 return 0

Ця функція визначає трапецієподібну функцію належності, яка використовується для фазифікації вхідних значень. Вона повертає ступінь належності вхідного значення `x` до нечіткої множини, визначеної параметрами `a`, `b`, `c` і `d`.

Визначення функцій належності для різних показників

# Визначення функцій належності для систолічного тиску  
def systolic\_func(value):  
 data = {'low': trapmf(value, a=80, b=80, c=90, d=110),  
 'normal': trapmf(value, a=90, b=120, c=120, d=140),  
 'high': trapmf(value, a=130, b=160, c=180, d=180)}  
 return data  
  
# Визначення функцій належності для діастолічного тиску  
def diastolic\_func(value):  
 data = {'low': trapmf(value, a=50, b=50, c=60, d=80),  
 'normal': trapmf(value, a=60, b=80, c=80, d=100),  
 'high': trapmf(value, a=80, b=100, c=120, d=120)}  
 return data  
  
# Визначення функцій належності для пульсу  
def pulse\_func(value):  
 data = {'low': trapmf(value, a=40, b=40, c=50, d=70),  
 'normal': trapmf(value, a=50, b=70, c=70, d=90),  
 'high': trapmf(value, a=80, b=120, c=180, d=180)}  
 return data  
  
# Визначення функцій належності для оцінки артеріального тиску  
def blood\_pressure\_func(value):  
 data = {'low': trapmf(value, a=0, b=0, c=20, d=50),  
 'normal': trapmf(value, a=20, b=50, c=50, d=80),  
 'high': trapmf(value, a=50, b=80, c=100, d=100)}  
 return data

Ці функції використовують трапецієподібну функцію належності для визначення ступеня належності значення до категорій "low", "normal" і "high" для систолічного тиску, діастолічного тиску, пульсу та загальної оцінки артеріального тиску.

Визначення правил нечіткої системи

# Визначення правил нечіткої системи  
def calculate\_rules(systolic, diastolic, pulse):  
 return [  
 # Правило 1: Якщо (систолічний = high і діастолічний = high) або пульс = high, то оцінка = high  
 ('high', max(min(systolic['high'], diastolic['high']), pulse['high'])),  
  
 # Правило 2: Якщо систолічний = normal і діастолічний = normal і пульс = normal, то оцінка = normal  
 ('normal', min(systolic['normal'], diastolic['normal'], pulse['normal'])),  
  
 # Правило 3: Якщо (систолічний = low і діастолічний = low) або пульс = low, то оцінка = low.  
 ('low', max(min(systolic['low'], diastolic['low']), pulse['low'])),  
  
 # Правило 4: Якщо систолічний = normal і діастолічний = normal і пульс = high, то оцінка = normal  
 ('normal', min(systolic['normal'], diastolic['normal'], pulse['high'])),  
 ]

Ця функція визначає правила нечіткої логіки для оцінки артеріального тиску на основі фазифікованих значень систолічного тиску, діастолічного тиску та пульсу.

Імплікація та агрегація правил

# Імплікації відповідних правил (Мамдані)  
def calculate\_implication(rules):  
 num = 100  
 blood\_pressure\_values = np.linspace(start=0, stop=100, num=num)  
 blood\_pressure\_fuzzy = {'low': np.zeros(num), 'normal': np.zeros(num), 'high': np.zeros(num)}  
  
 for i in range(num):  
 fuzzy\_output = blood\_pressure\_func(value=blood\_pressure\_values[i])  
 blood\_pressure\_fuzzy['low'][i] = fuzzy\_output['low']  
 blood\_pressure\_fuzzy['normal'][i] = fuzzy\_output['normal']  
 blood\_pressure\_fuzzy['high'][i] = fuzzy\_output['high']  
  
 result = {'low': np.zeros(num), 'normal': np.zeros(num), 'high': np.zeros(num)}  
 counter = {'low': 0, 'normal': 0, 'high': 0}  
 for name, value in rules:  
 result[name] += np.minimum(value, blood\_pressure\_fuzzy[name])  
 counter[name] += 1  
  
 for key in result.keys():  
 result[key] /= counter[key]  
 return result

Ця функція обчислює імплікацію правил нечіткої логіки за Мамдані, створюючи нечіткі множини для значень артеріального тиску та обчислюючи результати імплікації.

Агрегація результатів імплікації

# Агрегації результатів імплікації правил  
def calculate\_aggregation(implication):  
 result = np.zeros\_like(implication[list(implication.keys())[0]])  
 for name, value in implication.items():  
 result = np.maximum(result, value)  
 return result

Ця функція об'єднує результати імплікації, обираючи максимальне значення для кожного значення артеріального тиску.

Дефазифікація результатів

# Проведення дефазифікації результатів  
def calculate\_defuzzification(aggregation):  
 num = len(aggregation)  
 blood\_pressure\_values = np.linspace(start=0, stop=100, num=num)  
  
 a, b = 0, 0  
 for i in range(num):  
 if aggregation[i] > 0:  
 a += aggregation[i] \* blood\_pressure\_values[i]  
 b += aggregation[i]  
 return a / b if not b == 0 else None

Ця функція проводить дефазифікацію, обчислюючи центроїд нечіткої множини для визначення остаточного значення артеріального тиску.

Функція для обчислення оцінки артеріального тиску

# Функція для обчислення оцінки артеріального тиску  
def calculate\_blood\_pressure(systolic\_value, diastolic\_value, pulse\_value):  
 # Проведення фазифікації вхідних параметрів  
 systolic\_fuzzy = systolic\_func(value=systolic\_value)  
 diastolic\_fuzzy = diastolic\_func(value=diastolic\_value)  
 pulse\_fuzzy = pulse\_func(value=pulse\_value)  
  
 # Обчислення відповідних правил  
 rules = calculate\_rules(systolic=systolic\_fuzzy, diastolic=diastolic\_fuzzy, pulse=pulse\_fuzzy)  
  
 # Обчислення імплікації за Мамдані  
 implication = calculate\_implication(rules=rules)  
  
 # Проведення агрегації вілповідних результатів  
 aggregation = calculate\_aggregation(implication=implication)  
  
 # Дефазивікація ортиманої оцінки кровяного тиску  
 blood\_pressure\_score = calculate\_defuzzification(aggregation=aggregation)  
  
 if blood\_pressure\_score is None:  
 return (100 - 0) / 2  
 return blood\_pressure\_score

Ця функція об'єднує всі попередні етапи для обчислення оцінки артеріального тиску на основі вхідних значень систолічного тиску, діастолічного тиску та пульсу.

Графічний інтерфейс користувача

def interface():  
 try:  
 systolic\_value = float(entry\_systolic.get())  
 diastolic\_value = float(entry\_diastolic.get())  
 pulse\_value = float(entry\_pulse.get())  
  
 if not (80 <= systolic\_value <= 180):  
 messagebox.showerror('Помилка', 'Межі систолічного тиску: 80 - 180')  
  
 if not (50 <= diastolic\_value <= 120):  
 messagebox.showerror('Помилка', 'Межі діастолічного тиску: 50 - 120')  
  
 if not (40 <= pulse\_value <= 180):  
 messagebox.showerror('Помилка', 'Межі пульсу: 40 - 180')  
  
 result = calculate\_blood\_pressure(  
 systolic\_value=systolic\_value,  
 diastolic\_value=diastolic\_value,  
 pulse\_value=pulse\_value  
 )  
 messagebox.showinfo('Результат', f'Оцінка артеріального тиску: {result:.2f}')  
 except ValueError:  
 messagebox.showerror('Помилка', 'Будь ласка, введіть числові значення для всіх полів.')

Ця функція реалізує логіку взаємодії користувача з графічним інтерфейсом. Вона зчитує вхідні значення, перевіряє їх на коректність та обчислює оцінку артеріального тиску.

Головна функція (точка входу)

if \_\_name\_\_ == '\_\_main\_\_':  
 # Створення графічного інтерфейсу користувача  
 root = tk.Tk()  
 root.title('Оцінка артеріального тиску')  
  
 tk.Label(root, text='Систолічний тиск').grid(row=0, column=0)  
 entry\_systolic = tk.Entry(root)  
 entry\_systolic.grid(row=0, column=1)  
  
 tk.Label(root, text='Діастолічний тиск').grid(row=1, column=0)  
 entry\_diastolic = tk.Entry(root)  
 entry\_diastolic.grid(row=1, column=1)  
  
 tk.Label(root, text='Пульс').grid(row=2, column=0)  
 entry\_pulse = tk.Entry(root)  
 entry\_pulse.grid(row=2, column=1)  
  
 tk.Button(root, text='Обчислити', command=interface).grid(row=3, column=0, columnspan=2)  
 root.mainloop()

Цей блок коду створює графічний інтерфейс, розміщуючи відповідні елементи (поля введення та кнопки) та запускає головний цикл програми.

**test.py**

Необхідні бібліотеки

import numpy as np  
from matplotlib import pyplot as plt  
from main import systolic\_func, diastolic\_func, pulse\_func, calculate\_blood\_pressure

`numpy` -- для числових обчислень

`matplotlib` -- для створення графічного інтерфейсу

`main` -- це власний файл (main.py), це для отримання функції з нього

Встановлення початкових умов

np.random.seed(seed=2024)  
USE\_NOISE = 1 # 1 = використання шуму, 0 = відсутність шуму

Цей блок встановлює початкове значення для генератора випадкових чисел, щоб результати були відтворюваними, та визначає, чи буде використовуватися шум у даних.

Генерація даних

# Визначення часового проміжку  
num = 100  
time = np.linspace(start=0, stop=1, num=num)  
  
# Зміна систолічного тиску з часом + шум  
systolic = (180 - 80) \* 0.8 \* time + 90 + 3 \* np.random.randn(num) \* USE\_NOISE  
  
# Зміна діастолічного тиску з часом + шум  
diastolic = (120 - 50) \* 0.8 \* time + 60 + 2 \* np.random.randn(num) \* USE\_NOISE  
  
# Зміна пульсу з часом + шум  
pulse = (180 - 40) \* 0.8 \* time + 50 + 3 \* np.random.randn(num) \* USE\_NOISE

Цей блок коду генерує дані для систолічного тиску, діастолічного тиску та пульсу. Значення змінюються з часом та включають шум, який моделюється випадковими числами з нормального розподілу, помноженими на USE\_NOISE.

Обчислення оцінки артеріального тиску

# Обчислення оцінки артеріального тиску у часі  
blood\_pressure = [calculate\_blood\_pressure(systolic\_value=systolic[t],  
 diastolic\_value=diastolic[t],  
 pulse\_value=pulse[t]) for t in range(num)]

Цей блок обчислює оцінку артеріального тиску для кожного моменту часу, використовуючи функцію `calculate\_blood\_pressure` та згенеровані дані для систолічного тиску, діастолічного тиску та пульсу.

Відображення графіку

# Відображення графіку  
plt.figure(figsize=(10, 8))  
  
plt.subplot(5, 1, 1)  
plt.plot(systolic, color="tab:blue")  
plt.ylabel('Систолічний')  
plt.xticks([])  
  
plt.subplot(5, 1, 2)  
plt.plot(diastolic, color="tab:orange")  
plt.ylabel('Діастолічний')  
plt.xticks([])  
  
plt.subplot(5, 1, 3)  
plt.plot(pulse, color="tab:green")  
plt.ylabel('Пульс')  
plt.xticks([])  
  
plt.subplot(5, 1, (4, 5))  
plt.plot(blood\_pressure, color="tab:purple")  
plt.title('Оцінка артеріального тиску')  
plt.xticks([])  
  
plt.tight\_layout()  
plt.show()

Цей блок коду створює графік, що складається з чотирьох підграфіків:

1. Систолічний тиск за часом.
2. Діастолічний тиск за часом.
3. Пульс за часом.
4. Оцінка артеріального тиску за часом.

Використовуючи функцію plt.subplot, кожен підграфік розташовується один під одним. А plt.show() відображає графік.

# Plotting the membership functions for systolic, diastolic, and pulse as in slide 4  
x\_systolic = np.linspace(60, 200, 400)  
x\_diastolic = np.linspace(40, 140, 400)  
x\_pulse = np.linspace(30, 190, 400)  
  
systolic\_low = [systolic\_func(value)['low'] for value in x\_systolic]  
systolic\_normal = [systolic\_func(value)['normal'] for value in x\_systolic]  
systolic\_high = [systolic\_func(value)['high'] for value in x\_systolic]  
  
diastolic\_low = [diastolic\_func(value)['low'] for value in x\_diastolic]  
diastolic\_normal = [diastolic\_func(value)['normal'] for value in x\_diastolic]  
diastolic\_high = [diastolic\_func(value)['high'] for value in x\_diastolic]  
  
pulse\_low = [pulse\_func(value)['low'] for value in x\_pulse]  
pulse\_normal = [pulse\_func(value)['normal'] for value in x\_pulse]  
pulse\_high = [pulse\_func(value)['high'] for value in x\_pulse]

В цьому блоці йде генерація значень для показу значень в функціях належності.

plt.figure(figsize=(15, 10))  
  
plt.subplot(3, 1, 1)  
plt.plot(x\_systolic, systolic\_low, label='Low', color='tab:blue')  
plt.plot(x\_systolic, systolic\_normal, label='Normal', color='tab:green')  
plt.plot(x\_systolic, systolic\_high, label='High', color='tab:red')  
plt.axhline(0, color='black', linewidth=3)  
plt.title('Систолічний тиск')  
plt.xlabel('Значення')  
plt.ylabel('Належність')  
plt.legend()  
plt.xlim(80, 180)  
  
plt.subplot(3, 1, 2)  
plt.plot(x\_diastolic, diastolic\_low, label='Low', color='tab:blue')  
plt.plot(x\_diastolic, diastolic\_normal, label='Normal', color='tab:green')  
plt.plot(x\_diastolic, diastolic\_high, label='High', color='tab:red')  
plt.axhline(0, color='black', linewidth=3)  
plt.title('Діастолічний тиск')  
plt.xlabel('Значення')  
plt.ylabel('Належність')  
plt.legend()  
plt.xlim(50, 120)  
  
plt.subplot(3, 1, 3)  
plt.plot(x\_pulse, pulse\_low, label='Low', color='tab:blue')  
plt.plot(x\_pulse, pulse\_normal, label='Normal', color='tab:green')  
plt.plot(x\_pulse, pulse\_high, label='High', color='tab:red')  
plt.axhline(0, color='black', linewidth=3)  
plt.title('Пульс')  
plt.xlabel('Значення')  
plt.ylabel('Належність')  
plt.legend()  
plt.xlim(40, 180)  
  
plt.tight\_layout()  
plt.show()

Цей блок коду створює графіки функцій належності для кожного нашого параметру.