Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Системы искусственного интеллекта**

(название дисциплины)

**Лабораторная работа № 2**

**Вариант № 7**

Выполнил

Студент группы \_\_\_ИВТАПбд-41\_\_\_\_

\_\_\_\_\_\_\_\_\_\_Карпова А.С.\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

Проверил(а):

ст. преподаватель кафедры «ВТ»

(должность)

\_\_\_\_\_\_\_\_\_Хайруллин И.Д.\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

Ульяновск

2024

**Оглавление**

[Задание 3](#_Toc185885973)

[Ход работы 3](#_Toc185885974)

[Тестирование 5](#_Toc185885975)

[Вывод 6](#_Toc185885976)

[Приложение. Исходный код 7](#_Toc185885977)

# Задание

Необходимо разработать программу на языке python, которая реализует предложенное вариантом задание. Предметную область можно выбрать из предложенного списка, либо выбрать свою.

Вариант:

На языке Python разработайте скрипт, позволяющий выполнить операцию дополнения, заданного пользователем нечеткого множества с треугольной функцией принадлежности. Входными данными будут параметры функции принадлежности и четкие объекты множества. Выходными – дополнение нечеткого множества.

# Ход работы

#### 1. Программа определяет треугольную функцию принадлежности с помощью функции triangular\_membership. Эта функция принимает значение xx и три параметра (a,b,c)(a, b, c), задающих границы и вершину треугольной функции.

#### 2. Программа использует заранее заданные категории:

* Для BMI:
  + **Недостаточный вес:** (15, 35, 60)
  + **Нормальный вес:** (55, 65, 75)
  + **Избыточный вес:** (70, 80, 90)
  + **Ожирение:** (80, 90, 100)
* Для уровня активности:
  + **Малоподвижный:** (0, 1, 2)
  + **Умеренный:** (1, 3, 4)
  + **Активный:** (3, 5, 6)
  + **Очень активный:** (5, 7, 10)

#### 3. Программа использует следующие функции:

* bmi\_membership:
  + Рассчитывает степени принадлежности заданного значения BMI ко всем категориям.
  + Для вычислений применяет треугольную функцию принадлежности.
* activity\_level\_membership:
  + Аналогично вычисляет принадлежности для уровня физической активности.

#### 4. Для визуализации функций принадлежности используется функция plot\_membership. Она принимает диапазон значений xx, параметры категорий, функцию принадлежности и заголовок. Для каждой категории она вычисляет значения функции принадлежности и строит графики, которые показывают, как значение xx относится к различным категориям.

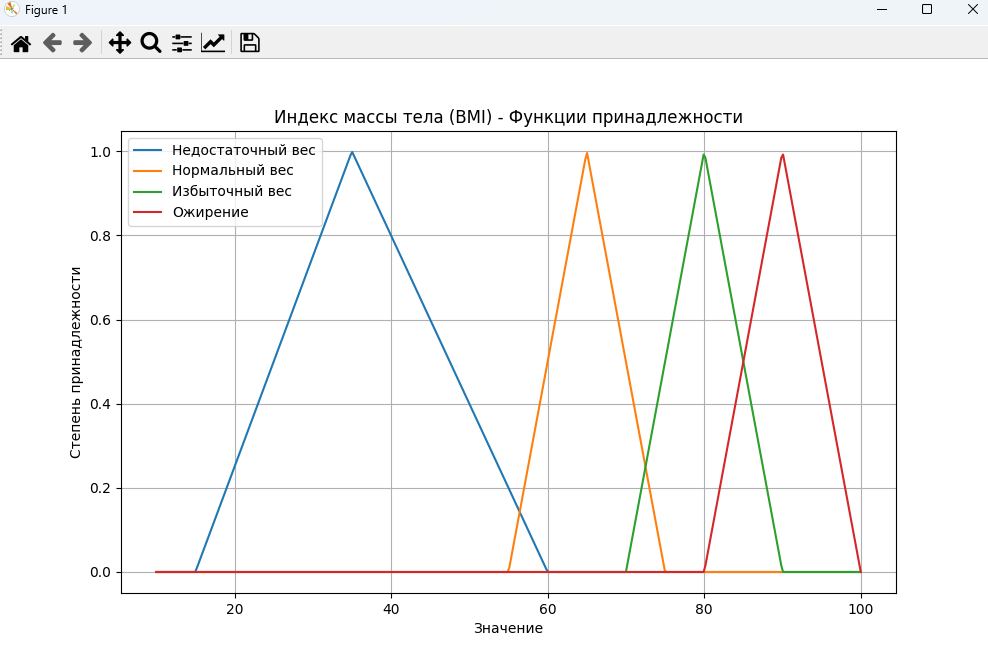
#### 5. Работа программы включает следующие шаги:

1. Пользователь вводит значение индекса массы тела (BMI) и уровня физической активности.
2. Программа задаёт диапазоны (рис. 1):
   * Для BMI: от 10 до 100.
   * Для уровня активности: от 0 до 10.
3. На основе введённых данных строятся графики функций принадлежности:
   * Для категорий BMI (рис. 2).
   * Для категорий уровня активности (рис. 3).

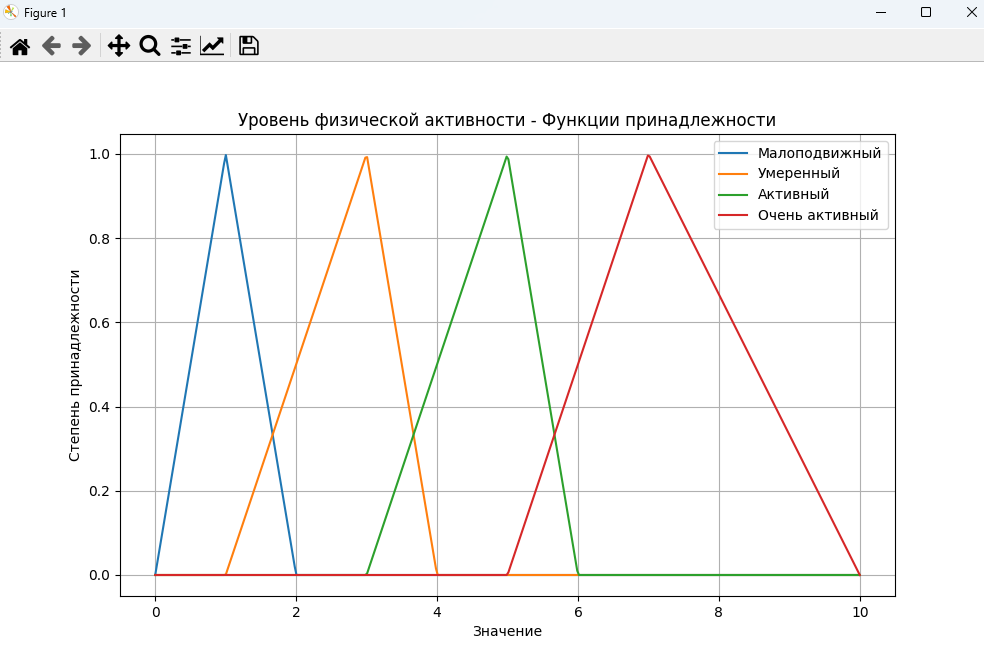
# Тестирование



*Рис. 1 Значения параметров*



*Рис. 2 График функций принадлежности для BMI*



*Рис. 3 График функций принадлежности для категорий уровня активности*

# Вывод

В данной лабораторной работе был разработан скрипт, который выполнял операцию дополнения заданных пользователем нечетких множеств с треугольными функциями принадлежности.

# Приложение. Исходный код

|  |
| --- |
| import matplotlib.pyplot as plt import numpy as np   def triangular\_membership(x, a, b, c):  #Функция для вычисления значения треугольной функции принадлежности  if x <= a or x >= c:  return 0  elif a < x <= b:  return (x - a) / (b - a)  elif b < x <= c:  return (c - x) / (c - b)   def bmi\_membership(bmi):  #Функция для вычисления степени принадлежности индекса массы тела (BMI)  categories = {  "Недостаточный вес": (15, 35, 60),  "Нормальный вес": (55, 65, 75),  "Избыточный вес": (70, 80, 90),  "Ожирение": (80, 90, 100),  }  memberships = {}  for category, (a, b, c) in categories.items():  memberships[category] = triangular\_membership(bmi, a, b, c)  return memberships   def activity\_level\_membership(activity\_level):  #Функция для вычисления степени принадлежности уровня физической активности  categories = {  "Малоподвижный": (0, 1, 2),  "Умеренный": (1, 3, 4),  "Активный": (3, 5, 6),  "Очень активный": (5, 7, 10),  }  memberships = {}  for category, (a, b, c) in categories.items():  memberships[category] = triangular\_membership(activity\_level, a, b, c)  return memberships   def plot\_membership(x\_vals, categories, membership\_func, title):  #Функция для построения графиков функции принадлежности  # Вычисление значений функции принадлежности для всех категорий  membership\_values = {category: [membership\_func(x, \*params) for x in x\_vals]  for category, params in categories.items()}   # Создаем графики  plt.figure(figsize=(10, 6))   # Отображаем функции принадлежности  for category, values in membership\_values.items():  plt.plot(x\_vals, values, label=category)   plt.title(f'{title} - Функции принадлежности')  plt.xlabel('Значение')  plt.ylabel('Степень принадлежности')  plt.legend()  plt.grid(True)  plt.show()   def main():  # Ввод данных от пользователя  bmi = float(input("Введите индекс массы тела (BMI): "))  activity\_level = float(input("Введите уровень физической активности (0-10): "))   # Диапазоны значений для построения графиков  bmi\_vals = np.linspace(10, 100, 400)  activity\_vals = np.linspace(0, 10, 400)   # Определение категорий для BMI и уровня активности  bmi\_categories = {  "Недостаточный вес": (15, 35, 60),  "Нормальный вес": (55, 65, 75),  "Избыточный вес": (70, 80, 90),  "Ожирение": (80, 90, 100),  }   activity\_categories = {  "Малоподвижный": (0, 1, 2),  "Умеренный": (1, 3, 4),  "Активный": (3, 5, 6),  "Очень активный": (5, 7, 10),  }   # Построение графиков для BMI и уровня физической активности  plot\_membership(bmi\_vals, bmi\_categories, triangular\_membership, 'Индекс массы тела (BMI)')  plot\_membership(activity\_vals, activity\_categories, triangular\_membership, 'Уровень физической активности')   if \_\_name\_\_ == "\_\_main\_\_":  main() |