# **КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені**

# **ТАРАСА ШЕВЧЕНКА**



## **ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №2**

# **з курсу**

**«Системний аналіз та теорія прийняття рішень»**

*Студента 3 курсу*

*групи ПП-31 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

Момотюка Михайла Тарасовича

*Викладач:*

Плескач В.Л.

Білий Р.О

## **Київ – 2023**

**Тема роботи:** Метод аналізу ієрархій.

**Мета роботи:** Закріпити теоретичні знання та отримати практичні навички здійснення експертної оцінки пріоритетності альтернатив методом аналізу ієрархії Сааті.

**Завдання:**

1. Побудувати декомпозицію задачі ієрархії за обраною предметною областю.

2. Визначити пріоритетність критеріїв.

3. Визначити пріоритетність альтернатив за кожним з критеріїв.

4. Визначити глобальний пріоритет альтернатив.

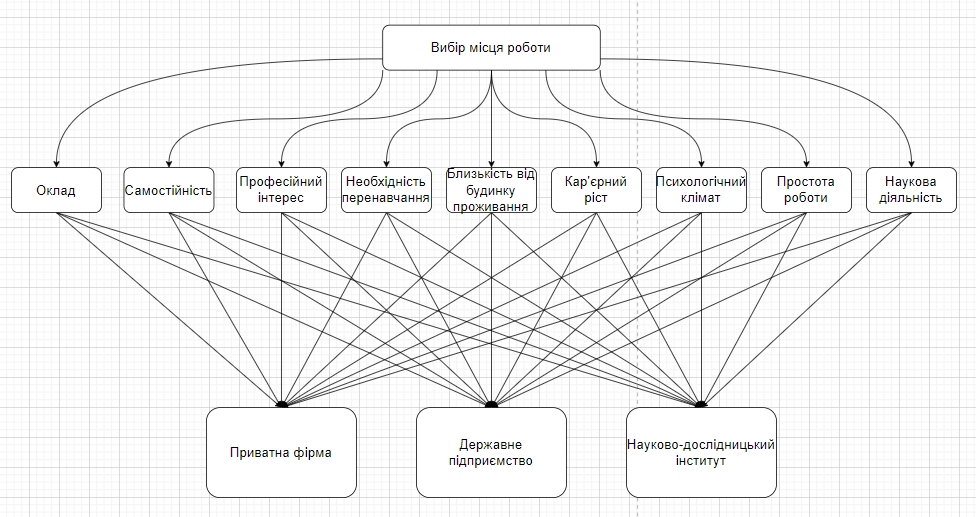
5. Оцінити рівень узгодженості думок експертів.

6. Зробити необхідні висновки.

13. Тато здійснює вибір місця роботи;варіанти: приватна фірма; державне підприємство; науково-дослідний інститут. Критерії: оклад; самостійність; професійний інтерес; необхідність перенавчання; близькість від будинку проживання; кар’єрний ріст; психологічний клімат; простота роботи; наукова діяльність.

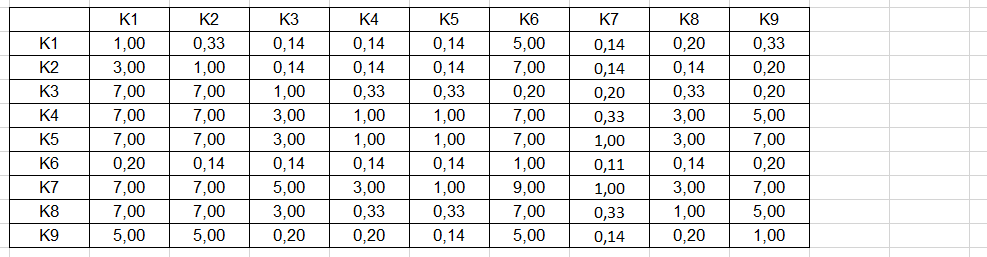
**Хід роботи**

Декомпозиція задачі:

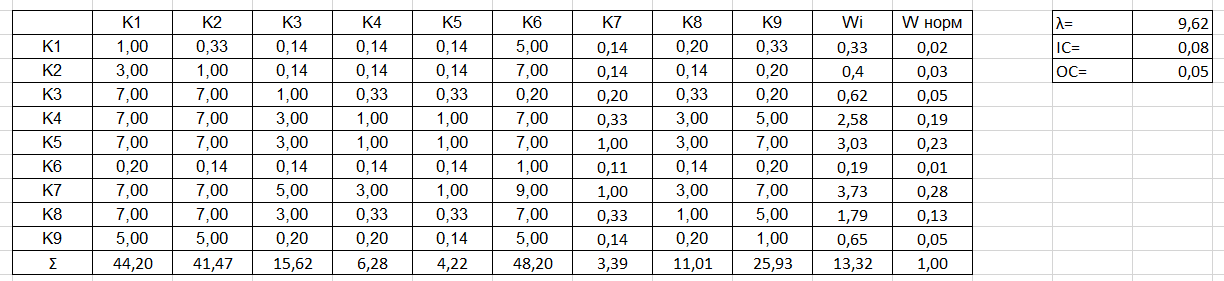


Завдання 1:

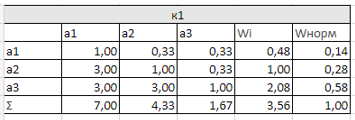
Вводимо матрицю попарних порівнянь важливості критеріїв

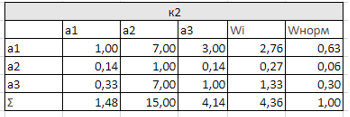


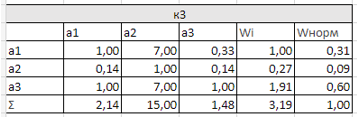
Значення індексу узгодженості менше 0,2 та відносної узгодженості менше 0,1, що свідчить про узгоженість думок експертів.

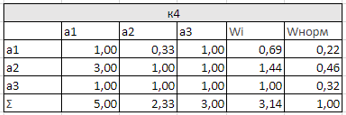


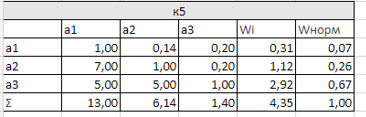
Визначаємо пріоритети альтернатив за кожним з критеріїв

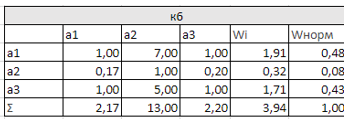


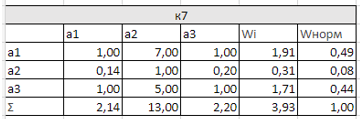


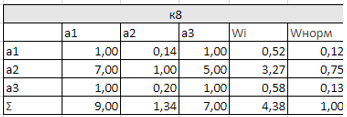


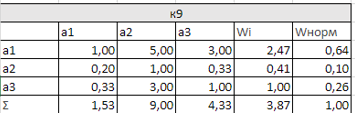




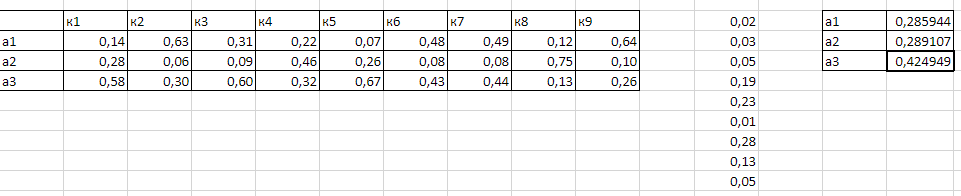








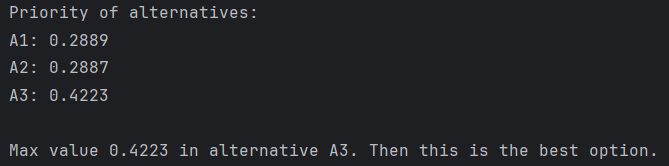
Розраховуємо пріоритети альтернатив



**Лістинг програми:**

import openpyxl  
import numpy as np  
  
  
def get\_matrix\_from\_sheet(sheet):  
 matrix = []  
  
 for row in sheet.iter\_rows(values\_only=True):  
 matrix.append(row)  
  
 return np.array(matrix)  
  
  
def calculate\_priority\_matrix(matrix):  
 number\_of\_rows = matrix.shape[0]  
 priorities\_matrix = np.zeros((number\_of\_rows, 1))  
 priorities\_sum = 0  
  
 for row in range(number\_of\_rows):  
 priority = np.prod(matrix[row])  
 priority = priority \*\* (1 / number\_of\_rows)  
 priorities\_sum += priority  
 priorities\_matrix[row, 0] = priority  
  
 priorities\_matrix /= priorities\_sum  
 return priorities\_matrix  
  
  
def create\_criteria\_priority\_matrix(workbook):  
 matrix = get\_matrix\_from\_sheet(workbook.worksheets[0])  
 matrix = calculate\_priority\_matrix(matrix)  
  
 return matrix  
  
  
def create\_alternatives\_priority\_matrix(workbook):  
 sheet\_names = workbook.sheetnames  
 number\_of\_sheets = len(sheet\_names)  
  
 tmp\_matrix1 = get\_matrix\_from\_sheet(workbook.worksheets[0])  
 tmp\_matrix2 = get\_matrix\_from\_sheet(workbook.worksheets[1])  
  
 number\_of\_columns = tmp\_matrix1.shape[1]  
 number\_of\_rows = tmp\_matrix2.shape[0]  
  
 alternative\_priority\_matrix = np.zeros((number\_of\_rows, number\_of\_columns))  
  
 for i in range(1, number\_of\_sheets):  
 matrix = get\_matrix\_from\_sheet(workbook.worksheets[i])  
 alternative\_priority\_matrix\_column = calculate\_priority\_matrix(matrix)  
  
 for j in range(number\_of\_rows):  
 alternative\_priority\_matrix[j][i - 1] = alternative\_priority\_matrix\_column[j][0]  
  
 return alternative\_priority\_matrix  
  
  
def display\_results\_matrix(matrix, text):  
 for i, row in enumerate(matrix, start=1):  
 for value in row:  
 print(f"{text}{i}: {round(value, 4)}")  
  
  
def find\_and\_display\_result(matrix1, matrix2):  
 results\_matrix = np.dot(matrix1, matrix2)  
  
 max\_result\_index = np.unravel\_index(np.argmax(results\_matrix), results\_matrix.shape)  
  
 result = round(results\_matrix[max\_result\_index], 4)  
  
 print('Priority of alternatives:')  
 display\_results\_matrix(results\_matrix, 'A')  
 print()  
 print(f'Max value {result} in alternative A{max\_result\_index[0] + 1}. '  
 f'Then this is the best option.')  
  
  
file\_path = 'lab2.xlsx'  
  
workbook = openpyxl.load\_workbook(file\_path)  
  
criteria\_priority\_matrix = create\_criteria\_priority\_matrix(workbook)  
alternative\_priority\_matrix = create\_alternatives\_priority\_matrix(workbook)  
  
find\_and\_display\_result(alternative\_priority\_matrix, criteria\_priority\_matrix)  
  
workbook.close()

Результат:



**Висновок:** У ході лабораторної роботи я закріпив теоретичні знання та отримав практичні навички здійснення експертної оцінки пріоритетності альтернатив методом аналізу ієрархії Сааті.