# **КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені**

# **ТАРАСА ШЕВЧЕНКА**



## **ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №3**

# **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

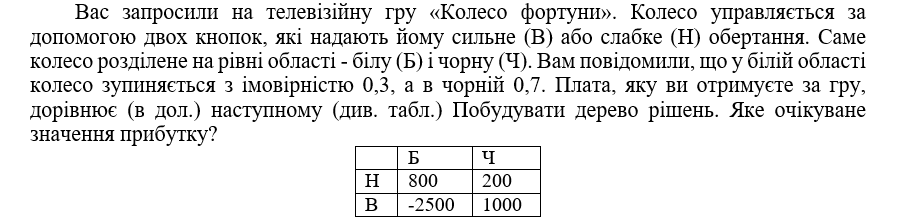
## **Київ – 2023**

**Тема роботи:** Прийняття рішення в умовах ризиків (дерево прийняття рішень).

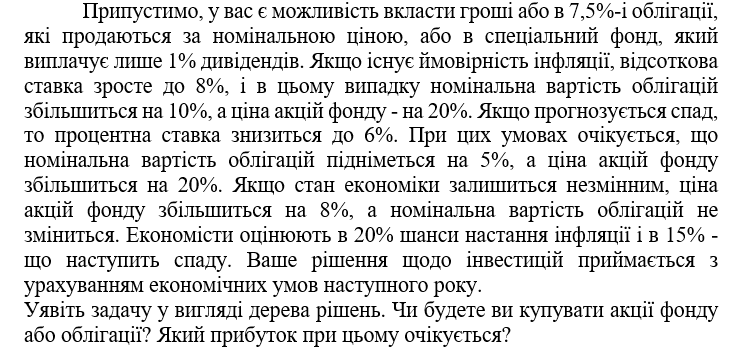
**Мета роботи:** Дослідити методи прийняття рішення в умовах ризиків і напрацювати навички пошуку раціональних рішень в умовах ризику з використанням пакета MS Excel, вміти запрограмувати рішення.

**Завдання:**

Task\_5:

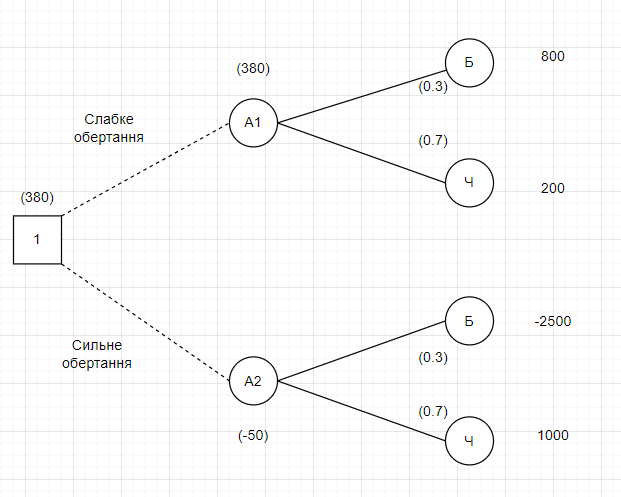


Task\_12:



**Хід роботи**

Task\_5:



Чистий дохід, що отримується в разі вибору альтернативи A1:  
M(A1) = 0.3 \* 800 + 0.7 \* 200 = 380

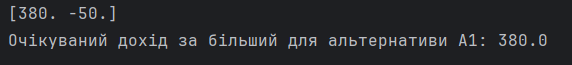
Чистий дохід, що отримується в разі вибору альтернативи A2:  
M(A2) = 0.3 \* - 2500 + 0.7 \* 100 = -50

В альтернативі А1 завжди перемагаємо, тому приймаємо рішення слабо крутити колесо.

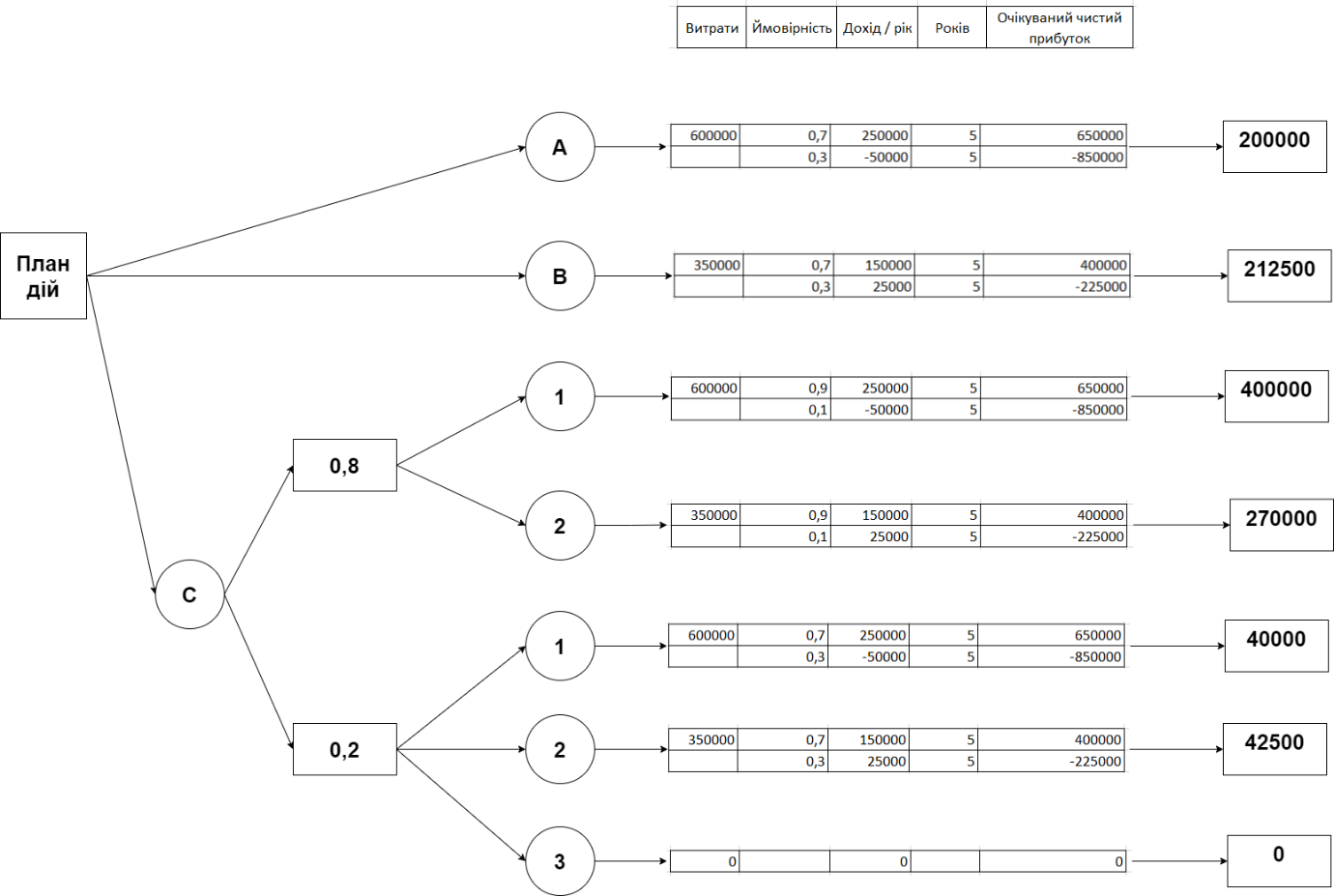
**Лістинг програми:**

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 find\_alternative\_values(workbook):  
 awards\_matrix = get\_matrix\_from\_sheet(workbook.worksheets[0])  
 chances\_matrix = get\_matrix\_from\_sheet(workbook.worksheets[1])  
  
 num\_rows = awards\_matrix.shape[0]  
 num\_columns = awards\_matrix.shape[1]  
  
 alternative\_values = np.zeros(num\_rows)  
  
 for row in range(num\_rows):  
 total\_sum = 0  
  
 for column in range(num\_columns):  
 total\_sum += awards\_matrix[row, column] \* chances\_matrix[0, column]  
  
 alternative\_values[row] = total\_sum  
  
 return alternative\_values  
  
  
file\_path = 'lab3\_5.xlsx'  
  
try:  
 workbook = openpyxl.load\_workbook(file\_path)  
except FileNotFoundError:  
 print(f"Файл '{file\_path}' не знайдено.")  
 exit()  
alternative\_values = find\_alternative\_values(workbook)  
max\_index = np.argmax(alternative\_values)  
  
print(alternative\_values)  
print(f'Очікуваний дохід за більший для альтернативи A{max\_index + 1}: {alternative\_values[max\_index]}')  
  
workbook.close()

Результат:



Task\_12:



Варіанти розвитку подій:

1. Немає потреби у додатковому доході (є можливість зачекати)

Зачекати 1 рік, що збільшить шанс успіху для будівництва заводів на `(0.7 - 0.8 \* 0.9) \* 100% = 2%` та знизить шанс невдачі до `0.8%`

1) Якщо інформація буде позитивною, то будувати великий завод, оскільки вага цієї альтернативи найбільша

2) Якщо інформація буде негативною, то не будувати завод взагалі, оскільки ризики будівництва будуть надто великими

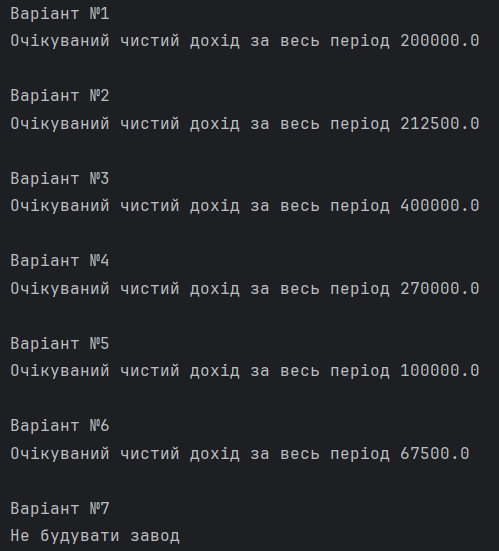
2. Є потреба у доході (немає можливості чекати)

Одразу будувати маленький завод, оскільки вага цієї гілки дерева найбільша (саме у випадку якщо треба будувати одразу).

**Лістинг програми:**

import numpy as np  
  
  
# Варіанти будівництва заводів  
# [Ціна будівництва, ймовірність попиту, дохід за рік]  
A = [600000, [0.7, 0.3], [250000, -50000]]  
B = [350000, [0.7, 0.3], [150000, 25000]]  
C1 = [600000, [0.9 \* 0.8, 0.1 \* 0.8], [250000, -50000]]  
C2 = [350000, [0.9 \* 0.8, 0.1 \* 0.8], [150000, 25000]]  
C3 = [600000, [0.9 \* 0.2, 0.1 \* 0.2], [250000, -50000]]  
C4 = [350000, [0.9 \* 0.2, 0.1 \* 0.2], [150000, 25000]]  
  
tree = [A, B, C1, C2, C3, C4]  
  
  
def calculate\_expected\_incomes(tree):  
 result\_incomes = []  
  
 results = []  
 for i in range(len(tree)):  
 temp\_results = []  
 branch = tree[i]  
  
 for j in range(2):  
 price = branch[0]  
 profit = branch[2][j]  
 probability = branch[1][j]  
  
 temp\_results.append([  
 branch[0],  
 probability,  
 profit,  
 profit \* 5 - price,  
 probability \* (profit \* 5 - price),  
 ])  
  
 expected\_income\_full = np.sum(temp\_results, axis=0)[4]  
 result\_incomes.append(expected\_income\_full)  
  
 results.append(temp\_results)  
  
 return results, result\_incomes  
  
  
def display\_results(results, result\_incomes):  
 for i in range(len(results)):  
 print("\nВаріант №" + str(i + 1))  
 print(f"Очікуваний чистий дохід за весь період {round(result\_incomes[i], 2)}")  
  
 print("\nВаріант №" + str(len(results) + 1))  
 print("Не будувати завод")  
  
  
results, result\_incomes = calculate\_expected\_incomes(tree)  
  
display\_results(results, result\_incomes)

Результат:



**Висновок:** У ході лабораторної роботи я опанував методи прийняття рішення в умовах ризиків і отримав навички пошуку раціональних рішень в умовах ризику з використанням пакета MS Excel та реалізував дані обчислення програмно.