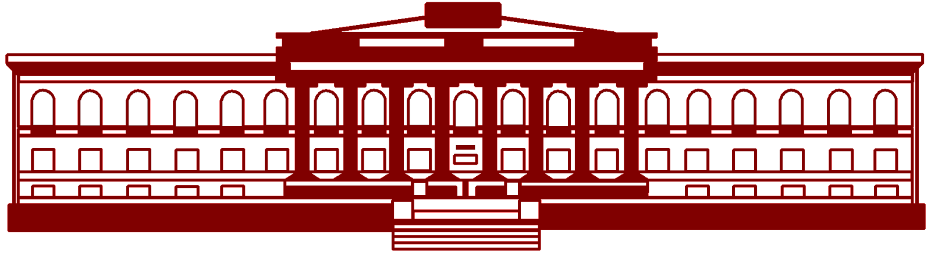
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**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-32*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Чалого Єгора Олеговича

*Викладач:*

*Білий Р.О.*

**Київ – 2023**

**Тема:** 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.

**Індивідуальні завдання:**

**Варіант 16**

**Завдання 9:**

Пекарня пече хліб на продаж магазинам. Собівартість однієї булки становить 30 пенсів, її продають за 40 пенсів. У таблиці наведено дані про попит за останні 50 днів:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Попит на день, тис. шт. | 10 | 12 | 14 | 16 | 18 |
| Число днів | 5 | 10 | 15 | 15 | 5 |

Якщо булка спечена, але не продана, то збитки складуть 20 пенсів за штуку. Визначте, скільки булок потрібно випікати в день.

**Завдання 12:**

Компанія «Brownhill Manufacturing Company» збирається виробляти новий товар, для чого потрібно буде побудувати новий завод. Після розгляду декількох варіантів були залишені три основні.

А. Побудувати завод вартістю 600 000 ф. ст. При цьому варіанті можливі: великий попит з імовірністю 0,7 і низький попит з імовірністю 0,3. Якщо попит буде великим, то очікується річний дохід в розмірі 250 000 ф. ст. протягом наступних п'яти років; якщо попит низький, то щорічні збитки через великі капіталовкладення складуть 50000 ф. ст.

Б. Побудувати маленький завод вартістю 350 000 ф. ст. Тут також можливі великий попит з імовірністю 0,7 і низький попит з імовірністю 0,3. У разі великого попиту щорічний дохід протягом п'яти років складе 150 000 ф. ст., при низькому попиті - 25000 ф. ст.

В. Відразу завод не будувати, а відкласти вирішення цього питання на один рік для збору додаткової інформації, яка може бути позитивною або негативною з ймовірностями 0,8 і 0,2 відповідно. Через рік, якщо інформація виявиться позитивною, можна побудувати великий або маленький завод за вказаними вище цінами. Керівництво компанії може вирішити взагалі ніякого заводу не будувати, якщо інформація буде негативною. Незалежно від типу заводу ймовірності великого та низького попиту змінюються на 0,9 і 0,1 відповідно, якщо буде отримана позитивна інформація. Доходи на наступні чотири роки залишаються такими ж, якими вони були у варіантах А і Б.

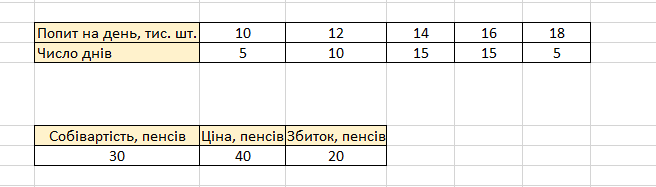
Всі витрати виражені в поточній вартості і не повинні дисконтуватися.

1. Намалюйте «дерево», що охоплює всі можливості, що відкриваються перед компанією.

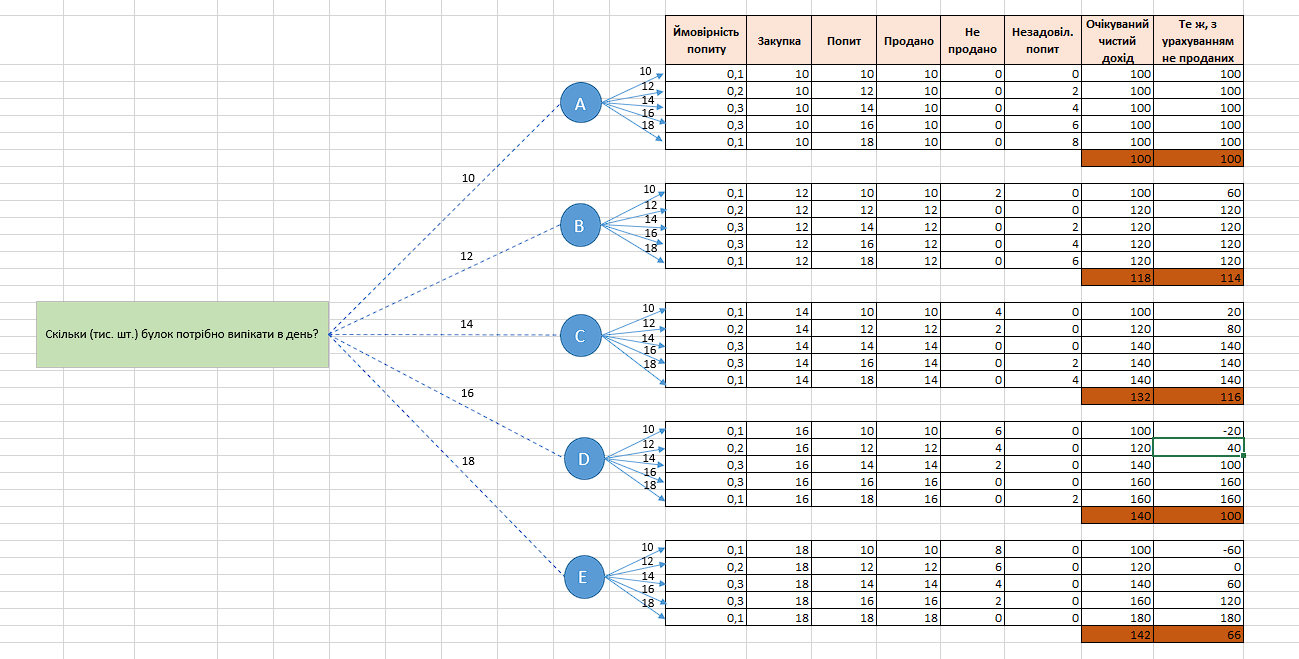
2. Визначте найбільш ефективну послідовність дій керівництва фірми, ґрунтуючись на очікуваних доходах кожного варіанта.

**Хід роботи:**

**Завдання 9**



Побудував дерево прийняття рішень:



Робимо висновок, що рекомендовано випікати 14 тис. булок в день, якщо ОПР готовий зважати на всі ризики та середній очікуваний прибуток становитиме 116 тис. пенсів. В випадку нехтування ризиками, варто випікати 18 тис. булок з очікуваним доходом 142 тис. пенсів.

Написав код, який виявляє оптимальну кількість булок, які потрібно випікати:

from typing import List, Tuple

DemandStats = Tuple[float, int]

Stats = List[DemandStats]

def calculate\_profit\_and\_decision(cost\_of\_production: float, selling\_price: float, sale\_price: float, demand\_stats: Stats) -> Tuple[float, int]:

    net\_profit\_per\_unit = selling\_price - cost\_of\_production

    net\_loss\_per\_unit = sale\_price - cost\_of\_production

    expected\_profits = []

    for prob, demand\_quantity in demand\_stats:

        total\_available = demand\_quantity

        expected\_profit = 0

        for inner\_prob, inner\_demand in demand\_stats:

            sold\_units = min(total\_available, inner\_demand)

            unsold\_units = total\_available - sold\_units

            expected\_profit\_by\_demand = sold\_units \* net\_profit\_per\_unit + unsold\_units \* net\_loss\_per\_unit

            expected\_profit += expected\_profit\_by\_demand \* inner\_prob

        expected\_profits.append(expected\_profit)

    max\_profit = max(expected\_profits)

    recommended\_quantity = demand\_stats[expected\_profits.index(max\_profit)][1]

    return max\_profit, recommended\_quantity

demand\_statistics: Stats = [

    (0.1, 10),

    (0.2, 12),

    (0.3, 14),

    (0.3, 16),

    (0.1, 18)

]

production\_cost: float = 30

selling\_price\_per\_unit: float = 40

sale\_price\_per\_unit: float = 10

def main():

    print("Завдання 9:\n")

    for prob, demand in demand\_statistics:

        print(f"Ймовірність попиту: {prob}, попит: {demand} тис.шт.")

    print("Собівартість виробництва: " + str(production\_cost))

    print("Ціна продажу: " + str(selling\_price\_per\_unit))

    print("Прибуток: " + str(selling\_price\_per\_unit - production\_cost))

    print("Втрати на непродану одиницю товару: " + str(sale\_price\_per\_unit - production\_cost) + "\n")

    net\_profit\_per\_unit = selling\_price\_per\_unit - production\_cost

    net\_loss\_per\_unit = sale\_price\_per\_unit - production\_cost

    expected\_profits = []

    for prob, demand\_quantity in demand\_statistics:

        total\_available = demand\_quantity

        expected\_profit = 0

        for inner\_prob, inner\_demand in demand\_statistics:

            sold\_units = min(total\_available, inner\_demand)

            unsold\_units = total\_available - sold\_units

            expected\_profit\_by\_demand = sold\_units \* net\_profit\_per\_unit + unsold\_units \* net\_loss\_per\_unit

            expected\_profit += expected\_profit\_by\_demand \* inner\_prob

        expected\_profits.append(expected\_profit)

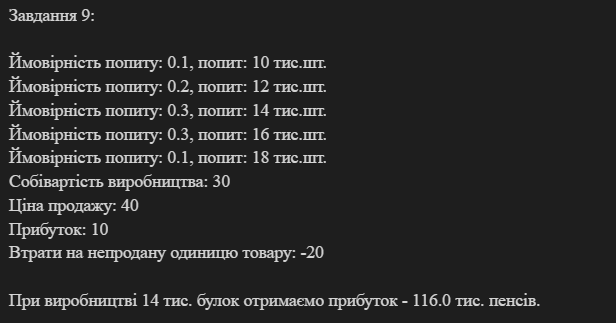
    result = calculate\_profit\_and\_decision(production\_cost, selling\_price\_per\_unit, sale\_price\_per\_unit, demand\_statistics)

    print(f"При виробництві {result[1]} тис. булок отримаємо прибуток - {result[0]} тис. пенсів.")

if \_\_name\_\_ == "\_\_main\_\_":

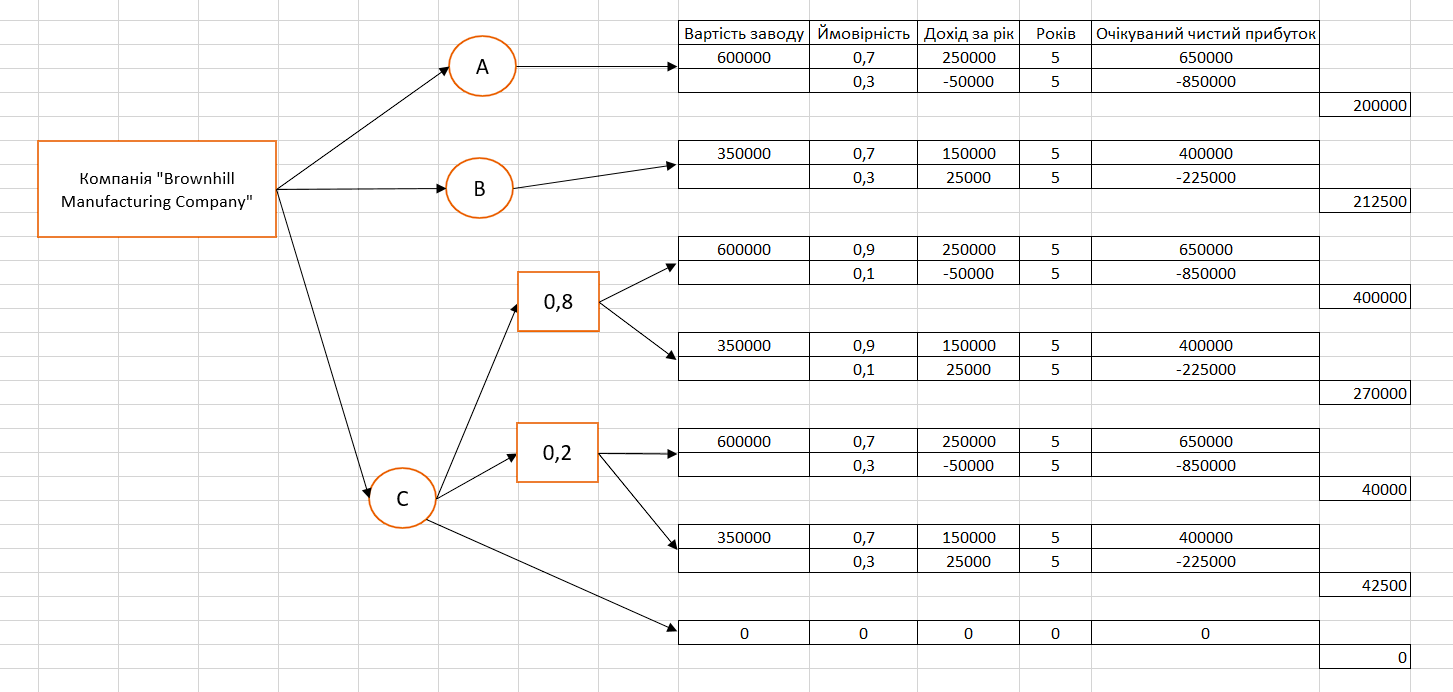
    main()

Результат:



**Завдання 12**

Намалював «дерево», що охоплює всі можливості, які відкриваються перед компанією «Brownhill Manufacturing Company»:



Судячи з дерева для ОПР є три варіанти:  
1) Якщо немає часу чекати і потрібен дохід, тоді починати будувати маленький завод, оскільки вага даної гілки найбільша.

2) Якщо є час чекати, тоді збирати додаткову інформацію, а через рік шанс невдачі зменшиться до 0,8% і у винятку позитивної інформації, почати будувати великий завод, оскільки вага даної гілки найбільша.

3) Якщо через рік інформація виявиться негативною, не будувати завод, оскільки будуть великі ризики понести збитки.

Написав код обчислення ваги кожної гілки:

import numpy as np

option\_A = [600000, [0.7, 0.3], [250000, -50000]]

option\_B = [350000, [0.7, 0.3], [150000, 25000]]

option\_C1 = [600000, [0.9 \* 0.8, 0.1 \* 0.8], [250000, -50000]]

option\_C2 = [350000, [0.9 \* 0.8, 0.1 \* 0.8], [150000, 25000]]

option\_C3 = [600000, [0.9 \* 0.2, 0.1 \* 0.2], [250000, -50000]]

option\_C4 = [350000, [0.9 \* 0.2, 0.1 \* 0.2], [150000, 25000]]

tree = [option\_A, option\_B, option\_C1, option\_C2, option\_C3, option\_C4]

def calc\_tables(result\_incomes: list):

    results = []

    for i in range(len(tree)):

        temp = []

        branch = tree[i]

        for j in range(2):

            price = branch[0]

            profit = branch[2][j]

            probability = branch[1][j]

            temp.append([

                branch[0],

                probability,

                profit,

                profit \* 5 - price,

                probability \* (profit \* 5 - price),

            ])

        expected\_income\_full = np.sum(temp, axis=0)[4]

        result\_incomes.append(expected\_income\_full)

        results.append(temp)

    return results

def print\_tables(results, result\_incomes):

    for i in range(len(results)):

        print("Гілка №" + str(i + 1))

        for row in results[i]:

            print("Вартість:", row[0])

            print("Ймовірність попиту:", row[1])

            print("Дохід за рік:", row[2])

            print("Чистий дохід за 5 років:", row[3])

            print("Очікуваний чистий дохід за 5 років (враховуючи ймовірність):", row[4])

            print()

        print("Очікуваний чистий дохід за весь період:", round(result\_incomes[i], 2))

        print("\n")

    print("Гілка №" + str(len(results) + 1))

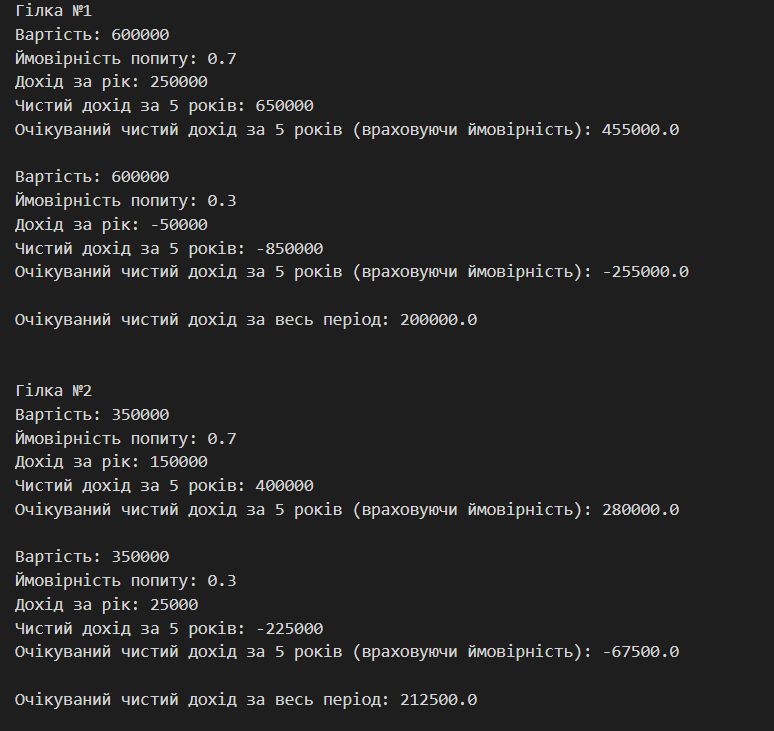
    print("Зовсім не будувати завод.")

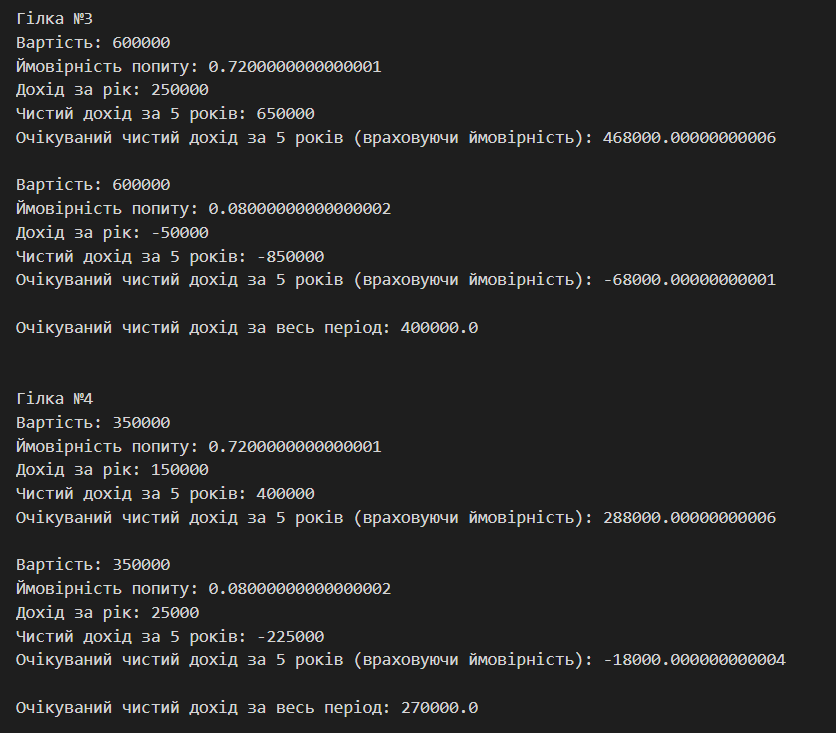
result\_incomes = []

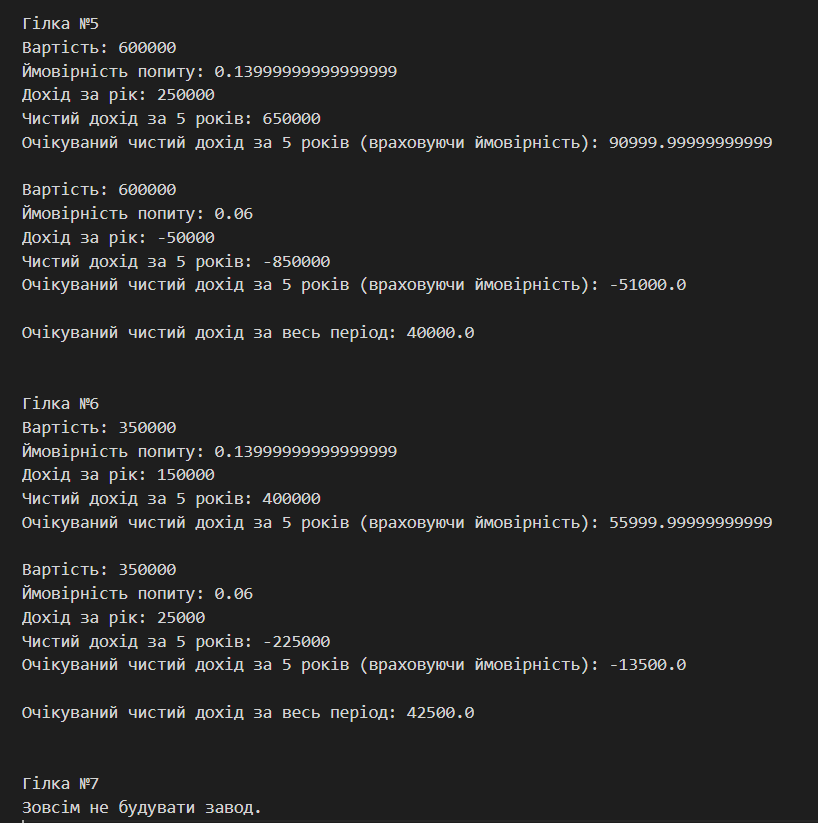
result = calc\_tables(result\_incomes)

print\_tables(result, result\_incomes)

Результат:







**Висновок:**

В ході виконання лабораторної роботи дослідив методи прийняття рішення в умовах ризиків і напрацював навички пошуку раціональних рішень в умовах ризику з використанням пакета MS Excel, а також запрограмував рішення на мові програмування Python.