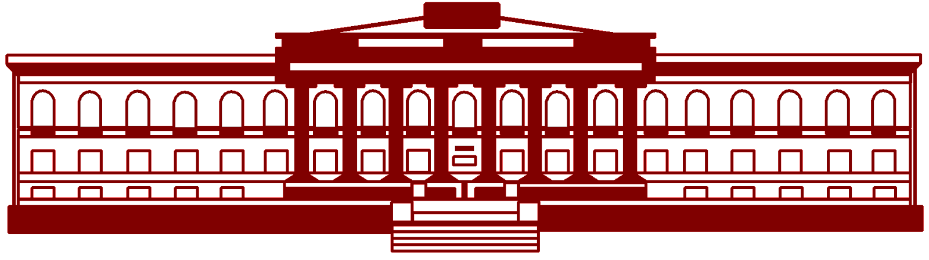
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №1**

**з курсу**

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

*студента 3 курсу*

*групи ПП-32*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Лоя Максима Юрійовича

*Викладач:*

Білий Р.О.

**Київ – 2024**

**Тема:** Прийняття рішення в умовах повної визначеності.

**Мета роботи:** Дослідити методи прийняття рішення в умовах повної визначеності.

**Порядок виконання роботи:**

1. Вирішити задачі засобами MS Excel.
2. Запрограмувати розв’язок задачі.
3. Скласти звіт з лабораторної роботи.
4. Захистити роботу.

Хід роботи:

***ЗАВДАННЯ 1.***

ОПР вибирає адвоката для представлення його інтересів у суді. Як альтернатив у нього є адвокати А1, А2, А3 і А4. В якості критеріїв виступають: Вартість (К1), Авторитет (К2), Репутація (К3), Спеціалізація (К4). Оцінки показників привабливості кожного адвоката (альтернативи) за кожним критерієм, а також ваги критеріїв за десятибальною системою представлені матрицею:

Скопіював першу таблицю у ексель файл

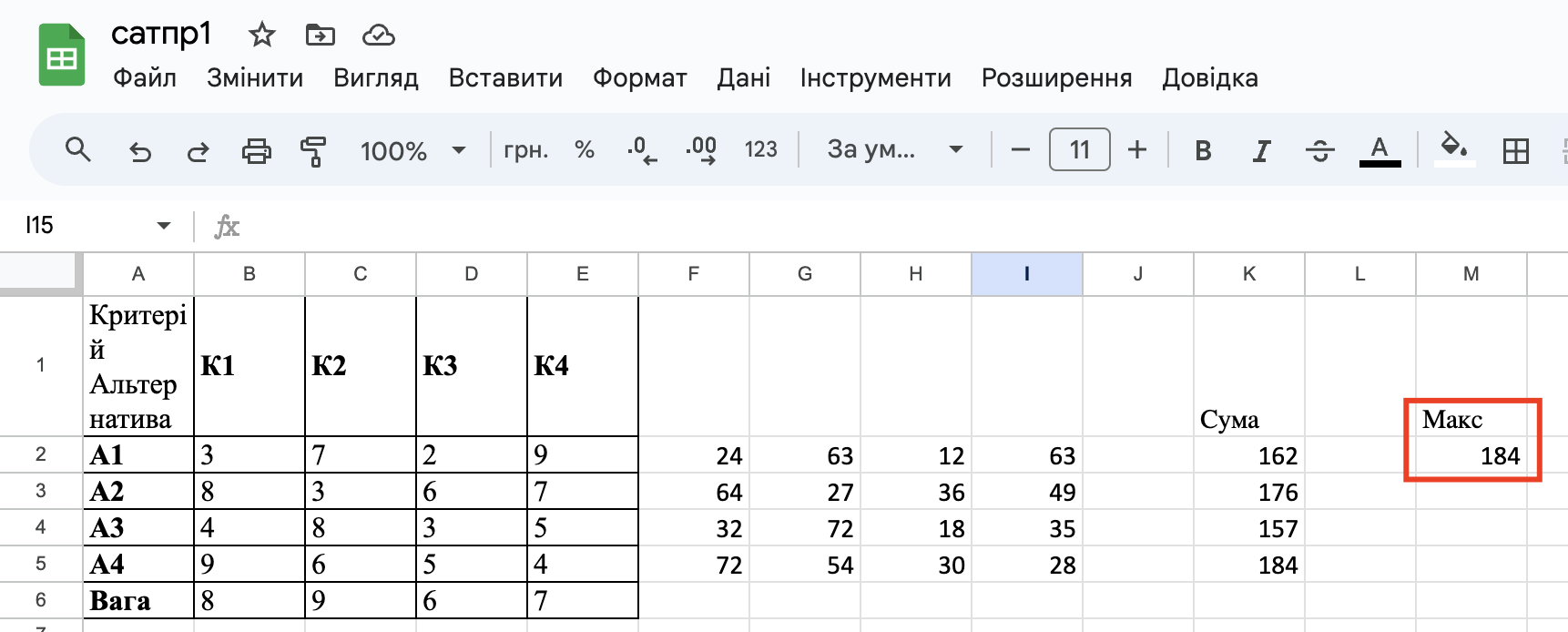
| Критерій Альтернатива | **К1** | **К2** | **К3** | **К4** |
| --- | --- | --- | --- | --- |
| **А1** | 3 | 7 | 2 | 9 |
| **А2** | 8 | 3 | 6 | 7 |
| **А3** | 4 | 8 | 3 | 5 |
| **А4** | 9 | 6 | 5 | 4 |
| **Вага** | 8 | 9 | 6 | 7 |

Далі розрахував оцінки за допомогою формул

| 24 | 63 | 12 | 63 |
| --- | --- | --- | --- |
| 64 | 27 | 36 | 49 |
| 32 | 72 | 18 | 35 |
| 72 | 54 | 30 | 28 |

Далі підрахував сумарні оцінки та знайшов максимальну

| Сума |  | Макс |
| --- | --- | --- |
| 162 |  | 184 |
| 176 |  |  |
| 157 |  |  |
| 184 |  |  |



***ЗАВДАННЯ 2.***

Директор приватного підприємства має намір прийняти на посаду юриста одного фахівця. Є п'ять кандидатів на цю посаду: А1, А2, А3, А4, А5. В якості критеріїв виступають: Освіта (100 бальна система, максимізується, К1), Запитувана зарплата (тис. грн. на місяць, К2); Стаж роботи на юридичній посаді (років, К3); Частка виграних справ у суді; Характеристики з місць робіт, авторитет (10 бальна система, максимізується). Оцінки альтернатив за всіма критерієм, а також ваги критеріїв наведені в таблиці.

Скопіював другу таблицю у ексель файл. Також для себе відмітив, які критерії необхідно максимізувати, а які мінімізувати. Мінімізувати необхідно тільки другий критерій.

| Критерий Альтернатива | **К1** | **К2** | **К3** | **К4** | **К5** |
| --- | --- | --- | --- | --- | --- |
| **А1** | 85 | 30 | 22 | 0,65 | 6 |
| **А2** | 60 | 20 | 10 | 0,6 | 7 |
| **А3** | 30 | 12 | 5 | 0,45 | 5 |
| **А4** | 75 | 24 | 13 | 0,7 | 8 |
| **А5** | 40 | 15 | 7 | 0,55 | 7 |
| **Вага** | 7 | 5 | 6 | 8 | 6 |
|  | макс | мін | макс | макс | макс |
|  | освіта | зарплата | стаж | вигр. спр | авторитет |

Нормалізував оцінки за прикладом

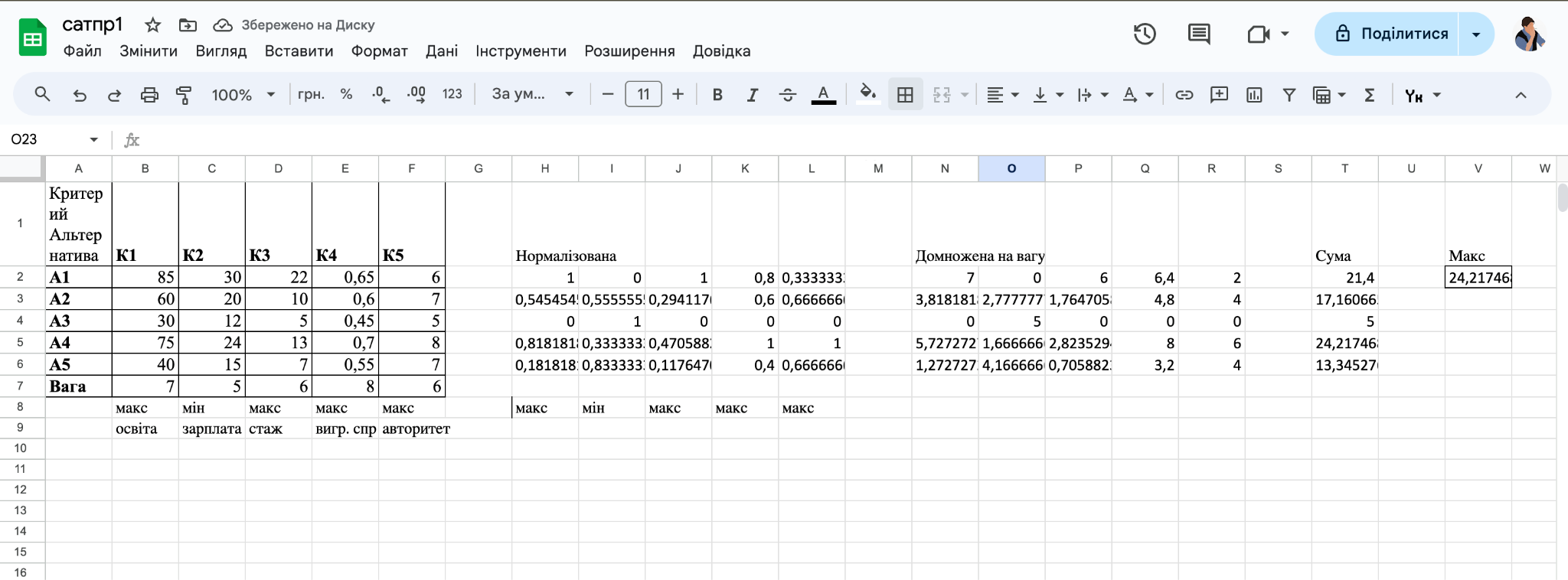
| 1 | 0 | 1 | 0,8 | 0,3333333333 |
| --- | --- | --- | --- | --- |
| 0,5454545455 | 0,5555555556 | 0,2941176471 | 0,6 | 0,6666666667 |
| 0 | 1 | 0 | 0 | 0 |
| 0,8181818182 | 0,3333333333 | 0,4705882353 | 1 | 1 |
| 0,1818181818 | 0,8333333333 | 0,1176470588 | 0,4 | 0,6666666667 |
|  |  |  |  |  |
| макс | мін | макс | макс | макс |

Домножив оцінки на вагу

| 7 | 0 | 6 | 6,4 | 2 |
| --- | --- | --- | --- | --- |
| 3,818181818 | 2,777777778 | 1,764705882 | 4,8 | 4 |
| 0 | 5 | 0 | 0 | 0 |
| 5,727272727 | 1,666666667 | 2,823529412 | 8 | 6 |
| 1,272727273 | 4,166666667 | 0,7058823529 | 3,2 | 4 |

Далі підрахував сумарні оцінки та знайшов максимальну

| Сума |  | Макс |
| --- | --- | --- |
| 21,4 |  | 24,21746881 |
| 17,16066548 |  |  |
| 5 |  |  |
| 24,21746881 |  |  |
| 13,34527629 |  |  |



Далі вирішив перше завдання за допомогою Python.

# Оцінювання претендентів за різними критеріями

applicants = [

[3, 7, 2, 9],

[8, 3, 6, 7],

[4, 8, 3, 5],

[9, 6, 5, 4]

]

# Коефіцієнти важливості критеріїв

criteria\_weights = [8, 9, 6, 7]

# Розрахунок сукупних балів для кожного претендента

applicants\_total\_scores = []

for applicant in applicants:

total = 0

for i in range(len(applicant)):

total += applicant[i] \* criteria\_weights[i]

applicants\_total\_scores.append(total)

applicant\_names = ['A1', 'A2', 'A3', 'A4']

# Визначення претендента з найвищим балом

highest\_score = max(applicants\_total\_scores)

best\_applicant\_index = applicants\_total\_scores.index(highest\_score)

top\_applicant = applicant\_names[best\_applicant\_index]

print(f"Кращий претендент: {top\_applicant}")

print(f"Найвищий бал: {highest\_score}")

****

Реалізував друге завдання.

# Оцінка продуктивності юристів за показниками

lawyers = [

[85, 30, 22, 0.65, 6],

[60, 20, 10, 0.6, 7],

[30, 12, 5, 0.45, 5],

[75, 24, 13, 0.7, 8],

[40, 15, 7, 0.55, 7]

]

# Вагові значення для кожного параметра

criteria\_weights = [7, 5, 6, 8, 6]

# Функція для пошуку мінімуму та максимуму по кожному критерію

def find\_min\_max(data):

transposed\_data = list(map(list, zip(\*data)))

mins = [min(column) for column in transposed\_data]

maxs = [max(column) for column in transposed\_data]

return mins, maxs

# Нормалізація даних

def normalize\_data(dataset, min\_vals, max\_vals):

normalized\_data = []

for entry in dataset:

norm\_entry = []

for idx, value in enumerate(entry):

if max\_vals[idx] == min\_vals[idx]:

norm\_entry.append(0)

else:

norm\_entry.append((value - min\_vals[idx]) / (max\_vals[idx] - min\_vals[idx]))

normalized\_data.append(norm\_entry)

return normalized\_data

# Спеціальна нормалізація для другого критерію

def normalize\_second\_criterion(normalized\_dataset, original\_dataset, max\_vals, min\_vals):

for i in range(len(normalized\_dataset)):

normalized\_dataset[i][1] = (max\_vals[1] - original\_dataset[i][1]) / (max\_vals[1] - min\_vals[1])

return normalized\_dataset

# Обчислення загальних балів

def calculate\_scores(normalized\_data, weights):

scores = []

for entry in normalized\_data:

total\_score = sum([entry[i] \* weights[i] for i in range(len(weights))])

scores.append(total\_score)

return scores

# Імена юристів

names = ['A1', 'A2', 'A3', 'A4', 'A5']

# Основна логіка

min\_values, max\_values = find\_min\_max(lawyers)

normalized\_lawyers = normalize\_data(lawyers, min\_values, max\_values)

normalized\_lawyers = normalize\_second\_criterion(normalized\_lawyers, lawyers, max\_values, min\_values)

scores = calculate\_scores(normalized\_lawyers, criteria\_weights)

# Визначення юриста з найвищим балом

best\_index = scores.index(max(scores))

print(f"Найкращий адвокат: {names[best\_index]}")

print(f"Підсумковий бал: {scores[best\_index]}")



**Висновки**

У цій лабораторній я розглянув прийняття рішення в умовах повної визначеності з використанням Excel та реалізував прийняття рішення програмно.