

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_\_» \_\_\_\_\_ 2025 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інженерія програмного  
забезпечення мультимедійних та інформаційно-пошукових систем»**

**спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Програмне забезпечення системи оцінки ефективності  
наукових та науково-технічних проєктів»**

Виконав:

студент IV курсу, групи КП-11

Регеша Олександр Миколайович

\_\_\_\_\_

Керівник:

доцент кафедри ПЗКС, к.ф.-м.н., доцент,

Нещадим Олександр Михайлович

\_\_\_\_\_

Консультант з нормоконтролю:

доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович

\_\_\_\_\_

Рецензент:

доцент кафедри СПіСКС, к.т.н., доцент,

Боярінова Юлія Євгенівна

\_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2025 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення  
мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

Регеші Олександр Миколайовичу

1. Тема проєкту «Програмне забезпечення системи оцінки ефективності наукових та науково-технічних проєктів», керівник проєкту Нещадим Олександр Михайлович, доцент кафедри ПЗКС, к.т.н., доцент, затверджені наказом по університету №1808-С від «29» травня 2025 р.
2. Термін подання студентом проєкту «13» червня 2025 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
  - Аналіз проблеми оцінки науково-технічних проєктів;
  - Модель оцінювання наукових проєктів;
  - Обґрунтування вибору засобів розроблення;
  - особливості проєктування та реалізації системи.
5. Перелік обов'язкового графічного матеріалу:
  - Схема функціонування системи (креслення);
  - діаграма основних процесів (креслення);
  - структура основних компонентів системи (плакат);
  - дерево проблем системи (плакат).

## 6. Консультанти розділів проєкту

| Розділ        | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|---------------|-------------------------------------------|----------------|------------------|
|               |                                           | завдання видав | завдання прийняв |
| Нормоконтроль | Онай М.В., доцент                         |                |                  |

## 7. Дата видачі завдання «31» жовтня 2024 р.

### Календарний план

| № з/п | Назва етапів виконання дипломного проєкту  | Термін виконання етапів проєкту | Примітка |
|-------|--------------------------------------------|---------------------------------|----------|
| 1.    | Вивчення літератури за тематикою проєкту   | 11.11.2024                      |          |
| 2.    | Аналітичний огляд                          | 25.11.2024                      |          |
| 3.    | Постановка задачі                          | 15.12.2024                      |          |
| 4.    | Модель оцінювання наукових проєктів        | 03.02.2025                      |          |
| 5.    | Проектування системи                       | 19.02.2025                      |          |
| 7.    | Програмна реалізація                       | 10.03.2025                      |          |
| 8.    | Тестовий приклад                           | 17.03.2025                      |          |
| 9.    | Оформлення документації дипломного проєкту | 30.04.2025                      |          |
| 10.   | Висновки                                   | 05.05.2025                      |          |

Студент

Олександр РЕГЕША

Керівник проєкту

Олександр НЕЩАДИМ

## АНОТАЦІЯ

Даний дипломний проєкт присвячений розробці системи оцінки ефективності наукових та науково-технічних проєктів.

Система реалізована у вигляді програмного забезпечення, яке дозволяє експертам проводити комплексне оцінювання проєктів за різними критеріями з використанням методу Харрінгтона. Завантаживши вхідні дані у форматі електронної таблиці, користувачі можуть отримати детальний аналіз кожного проєкту, включаючи узагальнені оцінки, рейтинги за окремими критеріями та візуалізацію результатів у вигляді діаграм.

Система надає можливість визначати як односторонні, так і двосторонні обмеження для критеріїв, враховує коефіцієнти важливості та дозволяє використовувати як числові, так і вербальні значення для оцінки. Для зручності користувачів реалізовано простий та інтуїтивно зрозумілий інтерфейс, який дозволяє переглядати різні форми представлення даних: узагальнені оцінки проєктів, оцінки за обраними критеріями, детальний аналіз окремого проєкту та порівняльні діаграми.

Метою даного дипломного проєкту є підвищення ефективності процесу відбору науково-технічних проєктів шляхом розробки програмного забезпечення, що забезпечує об'єктивну оцінку на основі багатокритеріального аналізу.

У даному дипломному проєкті розроблено: архітектуру системи, функції для розрахунку бажаностей критеріїв з одностороннім та двостороннім обмеженнями, алгоритм для обчислення узагальненого показника бажаності проєкту, інтерфейс користувача та модулі для візуалізації результатів оцінювання. Система успішно протестована на випадкових і граничних значеннях, а також на повноцінному тестовому прикладі, що імітує реальну практичну задачу з оцінки та вибору науково-технічних проєктів.

## **ABSTRACT**

This diploma project is dedicated to the development of a system for evaluating the effectiveness of scientific and technical projects.

The system is implemented as software that allows experts to conduct comprehensive evaluation of projects according to various criteria using the Harrington method. By uploading input data in spreadsheet format, users can obtain detailed analysis of each project, including generalized assessments, ratings by individual criteria, and visualization of results in the form of diagrams.

The system provides the ability to define both one-sided and two-sided constraints for criteria, takes into account importance coefficients, and allows the use of both numerical and verbal values for evaluation. For user convenience, a simple and intuitive interface is implemented, which allows viewing different forms of data presentation: generalized project assessments, evaluations by selected criteria, detailed analysis of individual projects, and comparative diagrams.

The purpose of this diploma project is to increase the efficiency of the scientific and technical project selection process by developing software that provides objective evaluation based on multi-criteria analysis.

In this diploma project, the following components have been developed: system architecture, functions for calculating the desirability of criteria with one-sided and two-sided constraints, an algorithm for calculating the generalized desirability index of a project, user interface, and modules for visualizing evaluation results. The system has been successfully tested on random and boundary values, as well as on a comprehensive test example that simulates a real practical task of evaluating and selecting scientific and technical projects.

ДП. 045420-01-90 Програмне забезпечення системи оцінки ефективності наукових і науково-технічних проєктів. Відомість проєкту

| Позначення      | Найменування                      | Кіл-ть | Примітка |
|-----------------|-----------------------------------|--------|----------|
|                 | Документація проєкту              |        |          |
|                 |                                   |        |          |
| ДП.045420-02-91 | Програмне забезпечення            | 5      |          |
|                 | системи оцінки ефективності       |        |          |
|                 | наукових та науково-технічних     |        |          |
|                 | проєктів. Технічне завдання       |        |          |
|                 |                                   |        |          |
| ДП.045420-03-81 | Програмне забезпечення            | 57     |          |
|                 | системи оцінки ефективності       |        |          |
|                 | наукових та науково-технічних     |        |          |
|                 | проєктів. Пояснювальна записка    |        |          |
|                 |                                   |        |          |
| ДП.045420-04-51 | Програмне забезпечення            | 4      |          |
|                 | системи оцінки ефективності       |        |          |
|                 | наукових та науково-технічних     |        |          |
|                 | проєктів. Програма та методика    |        |          |
|                 | тестування                        |        |          |
|                 |                                   |        |          |
| ДП.045420-05-34 | Програмне забезпечення            | 10     |          |
|                 | системи оцінки ефективності       |        |          |
|                 | наукових та науково-технічних     |        |          |
|                 | проєктів. Керівництво користувача |        |          |
|                 |                                   |        |          |
| ДП.045420-06-99 | Програмне забезпечення            | 1      |          |
|                 | системи оцінки ефективності       |        |          |
|                 | наукових та науково-технічних     |        |          |
|                 | проєктів. Схема функціонування    |        |          |
|                 | системи                           |        |          |
|                 |                                   |        |          |
|                 |                                   |        |          |
|                 |                                   |        |          |
|                 |                                   |        |          |

[illegible]

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

**“ЗАТВЕРДЖЕНО”**

Завідувач кафедри

\_\_\_\_\_Євгенія СУЛЕМА

“\_\_\_” \_\_\_\_\_2024 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНКИ**  
**ЕФЕКТИВНОСТІ НАУКОВИХ ТА НАУКОВО-ТЕХНІЧНИХ**  
**ПРОЄКТІВ**

**Технічне завдання**

**ДП.045420-02-91**

**“ПОГОДЖЕНО”**

Керівник проєкту:

\_\_\_\_\_Олександр НЕЩАДИМ

Нормоконтроль:

\_\_\_\_\_Микола ОНАЙ

Виконавець:

\_\_\_\_\_Олександр РЕГЕША



## ЗМІСТ

|                                              |   |
|----------------------------------------------|---|
| 1. Найменування та галузь застосування ..... | 3 |
| 2. Підстава для розроблення.....             | 3 |
| 3. Призначення розробки.....                 | 3 |
| 4. Вимоги до програмного продукту .....      | 3 |
| 5. Вимоги до проєктної документації.....     | 4 |
| 6. Етапи проєктування.....                   | 5 |
| 7. Порядок тестування розробки.....          | 5 |

## **1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ**

**Назва розробки:** програмне забезпечення системи оцінки ефективності наукових та науково-технічних проєктів.

**Галузь застосування:** інформаційні технології.

## **2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ**

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

## **3. ПРИЗНАЧЕННЯ РОЗРОБКИ**

Розробка призначена для використання науковими установами, дослідниками, керівниками наукових проєктів та грантодавцями для аналізу продуктивності наукових досліджень, оптимізації розподілу фінансування та підвищення прозорості оцінювання наукової діяльності.

## **4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ**

Система оцінки ефективності наукових та науково-технічних проєктів повинна забезпечувати такі основні функції:

- 1) можливість завантаження вхідних даних у форматі електронних таблиць, де кожен аркуш відповідає окремому проєкту, що оцінюється;
- 2) можливість вибору типу аналізу даних зі списку доступних опцій;
- 3) можливість формування рейтингу проєктів за різними критеріями та з урахуванням коефіцієнтів важливості;

- 4) можливість візуалізації результатів у вигляді різних типів діаграм (стовпчикові, кругові) для наочного представлення результатів оцінювання;
- 5) можливість збереження результатів аналізу у форматі текстових файлів або електронних таблиць.

Додаткові вимоги:

- 1) інтуїтивно зрозумілий інтерфейс, що не потребує спеціальної підготовки для роботи з системою;
- 2) обробка помилок, пов'язаних із некоректним форматом або змістом вхідних даних, з інформуванням користувача;
- 3) відображення бронювань у таблиці завантаженості кімнат відповідним кольором команди, що запрошена на зустріч;
- 4) реалізація як одностороннього, так і двостороннього обмеження при розрахунку бажаностей критеріїв.

## **5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ**

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
  - схема функціонування системи (креслення);
  - діаграма основних процесів (креслення).

## **6. ЕТАПИ ПРОЄКТУВАННЯ**

|                                               |            |
|-----------------------------------------------|------------|
| Вивчення літератури за тематикою роботи ..... | 11.11.2024 |
| Аналітичний огляд .....                       | 25.11.2024 |
| Постановка задачі.....                        | 15.12.2024 |
| Модель оцінювання наукових проєктів.....      | 03.02.2025 |
| Проектування системи.....                     | 19.02.2025 |
| Програмна реалізація .....                    | 10.03.2025 |
| Тестовий приклад.....                         | 17.03.2025 |
| Висновки .....                                | 15.05.2025 |
| Доповідь та презентація.....                  | 25.05.2025 |

## **7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ**

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

**“ЗАТВЕРДЖЕНО”**

Завідувач кафедри

\_\_\_\_\_Євгенія СУЛЕМА

“ \_\_\_\_ ” \_\_\_\_\_2025 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНКИ**  
**ЕФЕКТИВНОСТІ НАУКОВИХ ТА НАУКОВО-ТЕХНІЧНИХ**  
**ПРОЄКТІВ**

**Пояснювальна записка**

ДП.045420-03-81

**“ПОГОДЖЕНО”**

Керівник проєкту:

\_\_\_\_\_Олександр НЕЩАДИМ

Нормоконтроль:

\_\_\_\_\_Микола ОНАЙ

Виконавець:

\_\_\_\_\_Олександр РЕГЕША

2025

## ЗМІСТ

|                                                                        |    |
|------------------------------------------------------------------------|----|
| СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....                          | 4  |
| ВСТУП.....                                                             | 5  |
| 1. АНАЛІЗ ПРОБЛЕМИ ОЦІНКИ НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ ..                | 6  |
| 1.1. Значення науково-технічних проєктів у розробці технічних засобів. | 6  |
| 1.2. Оцінка та конкурсний вибір науково-технічних проєктів.....        | 7  |
| 1.3. Розгляд підходів до відбору науково-технічних проєктів .....      | 10 |
| 1.4. Інструменти відбору наукових проєктів.....                        | 17 |
| 1.5. Висновки до розділу 1.....                                        | 18 |
| 2. МОДЕЛЬ ОЦІНЮВАННЯ НАУКОВИХ ПРОЄКТІВ .....                           | 19 |
| 2.1. Огляд інструментальних засобів для відбору наукових розробок....  | 19 |
| 2.2. Критерії, для яких визначено одnobічне обмеження .....            | 21 |
| 2.3. Критерії, для яких визначено двобічне обмеження.....              | 23 |
| 2.4. Обчислення узагальненого показника бажаності .....                | 26 |
| 2.5. Добір критеріїв оцінювання .....                                  | 28 |
| 2.6. Висновки до розділу 2 .....                                       | 30 |
| 3. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ.....                       | 31 |
| 3.1. Вибір мови програмування для реалізації ПЗ .....                  | 31 |
| 3.2. Обґрунтування формату вхідних даних.....                          | 32 |
| 3.3. Висновки до розділу 3.....                                        | 33 |
| 4. ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЇ СИСТЕМИ.....                 | 34 |
| 4.1. Опис вимог до розроблюваної системи .....                         | 35 |
| 4.2. Аналіз проблематики та цілей вебзастосунку .....                  | 36 |
| 4.3. Опис архітектури системи.....                                     | 37 |
| 4.4. Особливості реалізації .....                                      | 41 |
| 4.5. Опис реалізованого користувацького інтерфейсу .....               | 44 |
| 4.6. Особливості проведеного тестування .....                          | 49 |
| 4.7. Висновки до розділу 4.....                                        | 54 |
| ВИСНОВКИ .....                                                         | 55 |
| СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ .....                          | 56 |
| ДОДАТКИ.....                                                           | 58 |

## **СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**

ПЗ – програмне забезпечення.

ІТ – Information Technology (інформаційні технології).

СЕЕНТП – система оцінки ефективності наукових та науково-технічних проєктів.

НТП – науково-технічний проєкт.

НАП – науково-аналітичний проєкт.

ФБП – функція бажаності по Харрінгтону.

МАІ – метод аналізу ієрархій (метод Сааті).

МБО – метод багатокритеріальної оптимізації.

ОПР – особа, що приймає рішення.

API – Application Programming Interface (інтерфейс прикладного програмування).

UI – User Interface (інтерфейс користувача).

UX – User Experience (користувацький досвід).

## **ВСТУП**

У сучасному світі науково-технічні проєкти відіграють фундаментальну роль у розвитку суспільства. Їх кількість постійно зростає, оскільки вони спрямовані на вирішення нагальних проблем людства, модернізацію існуючих технологій, підвищення добробуту населення та забезпечення принципів сталого розвитку.

Актуальність обраної теми зумовлена потребою у раціональному розподілі ресурсів через належне оцінювання ефективності науково-технічних проєктів. Цей процес дає змогу встановити, наскільки проєкт досягає поставлених цілей, а також визначити його потенційний вплив на наукові, технологічні, соціальні та економічні аспекти розвитку. Складність такого оцінювання полягає у необхідності інтегрального аналізу численних параметрів та індикаторів.

Головною метою даного дипломного проєкту є створення програмної забезпечення для комплексного оцінювання ефективності наукових та науково-технічних проєктів, що дозволить оптимізувати та автоматизувати процедури обчислень, аналізу та прийняття рішень.

Об'єктом дослідження виступають наукові та науково-технічні проєкти в їх різноманітності, тоді як предметом дослідження є методологія та критерії оцінювання й відбору таких проєктів.



# **1. АНАЛІЗ ПРОБЛЕМИ ОЦІНКИ НАУКОВИХ ТА НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ**

## **1.1. Значення науково-технічних проєктів у розробці технічних засобів**

Наукові та науково-технічні проєкти відіграють ключову роль у створенні нових технічних рішень. Вони являють собою комплекс дій, спрямованих на організацію і здійснення наукових досліджень з метою досягнення конкретного результату. Такий результат може бути представлений як нове знання, отримане під час проведення фундаментальних або прикладних досліджень, яке фіксується на відповідних носіях інформації. Також це можуть бути практичні розробки: нові або суттєво вдосконалені матеріали, продукти, технології, пристрої, процеси чи послуги, а також завершення важливих етапів науково-технічної діяльності, зокрема випробувань, впроваджень тощо.

Міжнародна та національна практика поділяє життєвий цикл продукції на п'ять основних етапів: дослідження, проєктування, виробництво, обіг, споживання та утилізація. Цей поділ є універсальним, хоча в окремих випадках може змінюватись. Незмінним залишається лише те, що початковим етапом завжди є дослідження та проєктування. Саме на цьому етапі визначаються оптимальні підходи, ресурси, методи, строки виконання та інші важливі параметри, що впливатимуть на весь подальший процес реалізації проєкту. Результатом дослідницької фази є проєктна модель майбутнього продукту, що включає аналіз ризиків, варіанти їх подолання та стратегічні рішення щодо реалізації. Важливість цього етапу полягає в тому, що будь-які помилки, допущені тут, можуть значною мірою вплинути на кінцевий результат, навіть якщо подальше виконання буде здійснене висококваліфікованими фахівцями. Невдалий вибір методології, матеріалів чи інструментів здатен знизити ефективність робіт, а недооцінка або переоцінка бюджету – вплинути на доцільність самого проєкту.

## **1.2. Оцінка та конкурсний вибір науково-технічних проєктів**

Оцінювання науково-технічних проєктів, особливо в умовах конкурсного відбору, часто ускладнене відсутністю уніфікованих кількісних характеристик. Об'єкти оцінювання мають різну природу та параметри, тому їх порівняння вимагає спеціальних підходів. Необхідно узгодити всі критерії, подати їх у спільній системі координат, що дозволяє здійснити адекватне зіставлення. Це досягається шляхом нормалізації – перетворення показників у безрозмірну форму. Завдяки цьому можна створити узагальнений показник, що охоплює технічні, економічні, соціальні та інші характеристики.

Необхідність обґрунтованої та неупередженої оцінки особливо актуальна у процесі відбору проєктів, які претендують на фінансування або інші види підтримки. Такі завдання мають багатокритеріальний характер і часто вирішуються в умовах неповної чи неточної інформації. В умовах обмежених ресурсів постає необхідність раціонального їх розподілу між численними запитами. Адже забезпечити фінансування всіх ініціатив неможливо, тому доводиться обирати ті, що демонструють найвищий потенціал – як з точки зору прикладної користі, так і з огляду на їхній внесок у майбутній науковий поступ.

Оптимальний вибір дозволяє зосередити зусилля на реалізації найбільш перспективних проєктів, що здатні принести вагомі результати. Зокрема, державні інституції, такі як МОН України, систематично організовують конкурси для відбору наукових ініціатив, що потребують фінансової підтримки або інших ресурсів. На підставі даних, оприлюднених на офіційному ресурсі міністерства, можна простежити динаміку фінансування у межах конкурсів для молодих науковців. Так було запроваджено конкурсну програму, орієнтовану на підтримку молодих вчених, у межах якої здійснювався відбір проєктів на фінансування. Подальші діаграми (рис. 1.1-1.3) ілюструють динаміку розвитку цієї ініціативи, підтверджуючи зростаючу увагу до підтримки досліджень.



Рис. 1.1. Динаміка фінансування проєктів за результатами конкурсу молодих учених.



Рис. 1.2. Кількість підтриманих проєктів за результатами конкурсу молодих учених.



Рис. 1.3. Візуалізація розподілу підтриманих проєктів за рівнем оцінки

Із аналізу графічних матеріалів, зокрема діаграми «Сума фінансування проєктів у 2016–2020 роках, млн грн», впливає, що починаючи з 2019 року спостерігається помітне зростання обсягів фінансової підтримки. Така тенденція свідчить про посилення уваги до розвитку наукового середовища та підтримки молодих дослідників. Попри загальне зменшення кількості проєктів, які отримали підтримку [3], спостерігається зростання частки високорівневих ініціатив. Це підтверджується даними, наведеними на діаграмі «Візуалізація розподілу підтриманих проєктів за рівнем оцінки» [4], де зафіксовано як абсолютне, так і відносне зростання таких проєктів. Завдяки цьому вони забезпечуються належним фінансуванням.

Серед учасників конкурсного відбору присутні не лише академічні чи державні установи, але й приватні структури. Залучення останніх може забезпечити додаткові фінансові переваги та сприяти ефективній реалізації наукових ініціатив. У зв'язку з цим для міністерства особливо актуальним є об'єктивне та комплексне оцінювання потенціалу кожного претендента. Зокрема, необхідно визначити ті проєкти, які можуть сприяти активізації

наукової діяльності та формуванню сприятливих умов для розвитку наукового середовища країни або забезпечити вагомі прикладні результати для окремих галузей.

У цьому контексті метою даного дослідження є розроблення системи для оцінювання ефективності наукових і науково-технічних проєктів. Такий підхід дозволить автоматизувати частину процесів, що, у свою чергу, спростить аналітичну роботу експертів, зменшить вплив суб'єктивного фактору та забезпечить вищу точність прийнятих рішень завдяки застосуванню формалізованих критеріїв. Запропонована система має універсальний характер і може бути адаптована для оцінювання явищ або елементів, що не піддаються однозначному кількісному вимірюванню вираженої кількісної характеристики. Окрім основної сфери застосування, така система здатна знайти застосування також і в інших галузях – наприклад, при оцінюванні студентських дослідницьких проєктів, дипломних робіт, конкурсів у сфері мистецтва та подібних ініціатив тощо.

### **1.3. Розгляд підходів до відбору наукових і науково-прикладних ініціатив**

В процесі застосування більшості методів багатокритеріального аналізу постають дві ключові задачі: по-перше, як здійснити оцінювання альтернатив за кожним окремим критерієм, і по-друге – яким чином перетворити ці оцінки у комплексний показник корисності кожної альтернативи. У типовій процедурі прийняття рішень експертна група або консультанти формують набір релевантних критеріїв, а також визначають спосіб вимірювання якісних характеристик за кожним із них. Як правило, застосовуються шкали оцінювання: шкала оцінювання може бути як десятибальною, так і приведеною до інтервалу від 0 до 1.

Наступним етапом є оцінювання альтернатив відповідно до обраної шкали. Якщо у процесі оцінювання задіяна група експертів, то індивідуальні оцінки підлягають агрегуванню в єдину узгоджену оцінку. Після цього стає

можливим перехід до інтегрального оцінювання альтернатив на основі формалізованої функції корисності. У випадках, коли кількість альтернатив або критеріїв є значною, доцільно використовувати програмні засоби, які автоматизують обчислення загальних оцінок на основі введених даних – зокрема, формул, вагових коефіцієнтів та оцінок.

Існує низка методів для розв'язання задач багатокритеріального вибору, які умовно поділяються на такі групи: прямі методи, методи порогів непорівняльності, методи компенсації та аксіоматичні підходи.

До прямих методів належить, зокрема, метод обчислення підсумкового балу на основі вагових коефіцієнтів для окремих критеріїв, у межах якого передбачено заздалегідь визначену залежність між оцінками відповідно до індивідуальних оцінок кожного з критеріїв, а також загальною корисністю альтернативи. Сутність методу полягає в призначенні кожному критерію певної ваги, що відображає його відносну важливість, після чого здійснюється множення цих ваг на відповідні оцінки альтернатив. Іншим прикладом прямого методу є ієрархічний підхід на основі дерева рішень, що дозволяє розгляд усіх можливих варіантів рішень із подальшим визначенням їх ймовірностей та очікуваної цінності, зазвичай у грошовому вираженні.

Алгоритм багатокритеріального аналізу в рамках прямих методів включає такі етапи:

- 1) визначення релевантних критеріїв оцінювання;
- 2) присвоєння ваг критеріям відповідно до їхньої важливості та, за потреби, виключення другорядних;
- 3) побудова шкал бажаності або функцій корисності щодо кожного з оцінюваних параметрів;
- 4) проведення порівняльного аналізу варіантів;
- 5) визначення інтегрального показника доцільності для кожного з розглянутих варіантів.

Інструменти прийняття рішень порогів непорівняльності передбачають використання спеціальних правил для порівняння альтернатив. Згідно з ними, альтернатива вважається кращою, якщо вона має вищі оцінки за більшістю критеріїв. На основі таких правил пари альтернатив класифікуються які можуть бути як взаємно зіставними (одна альтернатива переважає іншу або є рівнозначною), так і несумісними для прямого порівняння. Методологія була вперше представлена французьким дослідником Б. Руа і передбачає поетапну оптимізацію –спочатку за одним критерієм, потім за наступним, із встановленням відповідних порогових обмежень. На основі відношень переваги (бінарних відносин) формується ядро (множина Парето), що містить домінуючі та непорівнянні альтернативи. Після кількох ітерацій ядро звужується до остаточного набору пріоритетних альтернатив.

Методи компенсації ґрунтуються на прагненні врівноважити переваги одних альтернатив недоліками інших. У цьому випадку кожному з варіантів призначаються відповідні характеристики або значення критеріїв, які потім порівнюються попарно, що дозволяє виявити найбільш збалансовані рішення.

Аксіоматичні методи базуються на формулюванні сукупності властивостей, яким має відповідати функція корисності. Ці властивості перевіряються шляхом опитування експертів, після чого формується модель залежності загальної оцінки від значень відповідно до окремих параметрів, визначених у дослідженні [5].

Серед широкого спектра методів, застосовуваних для багатокритеріального вибору, варто виокремити підхід, відомий як метод "ідеального рішення" (також LINMAP) – Linear Programming Technique for Multidimensional Analysis of Preference). Його основою є принцип наближення до умовно найкращого варіанта, згідно з яким найбільш прийнятною є альтернатива, тобто визначення ступеня віддаленості кожного варіанту від еталонного рішення. Цей підхід передбачає

використання матриці розміром  $n \times m$ , де  $n$  – кількість альтернатив, а  $m$  – кількість критеріїв. Для кожного з показників необхідно встановити цільову функцію (монотонно зростаюча або спадна), а також ваговий коефіцієнт, який може бути отриманий або вручну (експертно), або шляхом застосування допоміжних алгоритмів чи методик.

Застосування такого підходу (LINMAP) є одним з ефективних підходів багатокритеріального оцінювання альтернатив. Його суть полягає у визначенні такої альтернативи, яка має мінімальне відхилення від умовно найкращого (ідеального) варіанта, сформованого на основі критеріїв оцінювання.

Першим етапом алгоритму є нормалізація матриці рішень, що передбачає усунення впливу різних одиниць вимірювання критеріїв. Така трансформація дозволяє привести всі значення до безрозмірної форми, що забезпечує можливість їх коректного порівняння. Одним з поширених способів нормалізації є поділ значення кожного елемента на норму відповідного вектору стовпця. Формула нормалізованого елемента матриці має вигляд:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^n x_{kj}^2}}. \quad (1.1)$$

Наступним кроком є визначення ідеальної альтернативи, яка у цьому методі розглядається як умовна альтернатива, що має найкращі значення кожного показника серед усіх доступних. Для її формування з кожного стовпця нормалізованої матриці вибирається екстремальне (максимальне або мінімальне залежно від напрямку оптимізації) значення, яке представляє найкраще досяжне значення критерію в межах розглянутих альтернатив.

Після побудови ідеального варіанту здійснюється розрахунок ступеня віддаленості кожної альтернативи від ідеальної. Для цього використовується евклідова метрика, що враховує вагові коефіцієнти критеріїв.



Кожна відстань ( $S_i$ ) відображає рівень близькості альтернативи  $i$  до умовно найкращої альтернативи:

$$S_i = \sqrt{\sum_{j=1}^m w_j (r_{ij} - r_j^*)^2}, \quad (1.2)$$

де:

- $w_j$  – коефіцієнт значущості, що відповідає конкретному показнику;
- $r_{ij}$  – отримане значення показника  $j(1, \dots, m)$  для альтернативи з індексом  $i$ ;
- $r_j^*$  – значення показника, прийняте для умовно найкращої (еталонної) альтернативи.

Заключним етапом процедури є порівняння отриманих відстаней  $S_i$  для всіх альтернатив. Оптимальною визнається та альтернатива, яка має мінімальне значення цього показника, що вказує на її максимальну близькість до ідеального варіанту [7, 8].

Ієрархічний підхід до прийняття рішень (так званий МАІ), запропонований американським науковцем Томасом Сааті, є ефективним інструментом для прийняття рішень у складних ситуаціях, де на результат впливають численні суперечливі чинники. Завдяки універсальності та простоті застосування, цей метод знайшов широке застосування у плануванні, управлінні, економічному аналізі та інших сферах, де необхідна систематизація переваг серед альтернатив.

Сутність методу полягає у наданні змогу розкласти складну задачу на послідовні рівні з простішими компонентами з подальшим використанням процедури попарного порівняння між ними. При цьому оцінюється не абсолютна величина, а відносна значущість кожного елементу відносно іншого. Наприклад, якщо розглядати три об'єкти – А, В і С – і вважати, що варіант А є кращим за В, а В, у свою чергу, переважає С, тоді логіка методу передбачає збереження цієї ієрархії в остаточному виборі. Аналогічний підхід застосовується і для оцінювання нематеріальних критеріїв.

Для кількісного вираження переваг використовується спеціальна шкала парних порівнянь, розроблена Сааті. Ця шкала є безрозмірною, що дозволяє уникнути труднощів, пов'язаних з неоднорідністю одиниць вимірювання. Її ефективність була багаторазово підтверджена як теоретично, так і практично. Практика показує, що під час оцінювання учасники найчастіше відповідають на запитання типу: «Який з варіантів має вищий пріоритет?» або «Котрий з них видається найбільш прийнятним/реалістичним?» тощо.

Першим кроком у застосуванні методу є структурування проблеми у вигляді ієрархії. Верхній рівень такої ієрархії представляє основну мету (наприклад, ухвалення управлінського рішення), проміжні рівні містять критерії або підцілі, що впливають на досягнення мети, а найнижчий рівень зазвичай складається з альтернатив, які необхідно оцінити. Ієрархія вважається завершеною тоді, коли кожен із компонентів верхнього рівня виступає орієнтиром для оцінки усіх елементів нижчого рівня. Можливе також формування підієрархій, що мають спільну головну ціль. Умовою коректного функціонування є дотримання принципу узгодженості, коли усі підпорядковані елементи підлягають оцінюванню відносно одного критерію вищого рівня, і цей процес триває до досягнення вершини ієрархії.

Наступним етапом є формування таблиць, у яких здійснюються попарні зіставлення для кожної категорії критеріїв одного рівня щодо відповідного елементу вищого рівня. Порівняння проводяться шляхом встановлення відносної переваги одного елемента над іншим. У разі, якщо перший з об'єктів має вищу перевагу над другим, у відповідну клітинку записується ціле число; у зворотному випадку – його обернене дробове значення. Самооцінка елементів дорівнює одиниці, отже, головна діагональ матриці завжди заповнена одиницями.

Після заповнення матриць переходять до перевірки узгодженості суджень. Це необхідно для зменшення впливу суб'єктивних похибок. Спеціальний коефіцієнт узгодженості дозволяє оцінити рівень

суперечливості у судженнях експерта. Вважається допустимим, якщо значення цього показника не перевищує 10%. У випадках, де абсолютна точність не є критичною, припустимим вважається рівень до 20%. Якщо межа перевищена, необхідно переглянути судження і, за потреби, скоригувати порівняння. Варто зазначити, що метод передбачає допустимий рівень відхилення або непослідовності, який є характерним при прийнятті колективних рішень.

Таким чином, ієрархічна методологія прийняття рішень надає змогу дозволяє зручно та логічно структурувати складну задачу і здійснити обґрунтований вибір серед альтернатив шляхом поетапного порівняння з урахуванням вагомості кожного критерію [12].

У середині 60-х років XX століття науковець зі США Едвін Харрінгтон запропонував метод узагальнення якісних показників продукції у вигляді єдиного індикатора – функції бажаності. Його підхід був зумовлений необхідністю адекватного відображення загальної якості об'єкта, яка, як правило, формується не одним, а кількома критеріями, що мають неоднорідну шкалу вимірювання та часто мають складні взаємозв'язки. Харрінгтон відзначав, що якість промислових виробів не може бути оцінена лише за однією характеристикою, адже у практичних умовах на результат впливає комплекс факторів.

Для подолання цієї складності він запропонував концепцію безрозмірної шкали – так званої шкали бажаності (позначається як  $d$ ), яка дозволяє перетворити різномірні критерії на єдину уніфіковану метрику. Застосування такого підходу відкриває можливість арифметичної обробки оцінок, що були отримані за різними параметрами, шляхом об'єднання їх у єдиний показник. Завдяки цьому функція бажаності може виступати не лише інструментом оцінки, а й критерієм оптимізації, забезпечуючи однозначне та кількісно обґрунтоване уявлення про рівень якості чи ефективності об'єкта.

Важливою характеристикою інтегральної функції бажаності є її здатність забезпечувати стабільну оцінку, яка зберігає інформативність навіть при зміні альтернатив або розширенні переліку критеріїв. Це вигідно відрізняє її від методів типу «ідеальної точки», де будь-яке доповнення новими варіантами змінює орієнтири оцінки, вимагаючи повторного аналізу. Окрім того, підхід Харрінгтона дає змогу варіювати вигляд функції залежно від вимог до зростання або спадання впливу критерію, тоді як у класичних методах цільові функції часто залишаються лінійними та строго монотонними.

На відміну від, наприклад, методу аналізу ієрархій, що більше орієнтований на визначення відносного пріоритету альтернатив через ранжування, функція бажаності Харрінгтона забезпечує числову інтегровану оцінку, що особливо важливо у випадках, коли потрібно не лише порівняти об'єкти, а й кількісно оцінити рівень досягнення мети, наприклад, при виборі науково-технічних проєктів.

Проведений аналіз свідчить про те, що застосування бажанісної функції за Харрінгтоном гарантує низку переваг над іншими підходами багатокритеріальної оцінки. Вона дозволяє обрати найкращі варіанти серед запропонованих альтернатив, розраховуючи об'єктивну узагальнену оцінку кожного з них.

#### **1.4. Інструменти відбору наукових проєктів**

Серед засобів, що використовуються для реалізації багатокритеріальних методів, варто згадати електронні таблиці (Microsoft Excel, Google Sheets), які дозволяють вводити та обробляти дані вручну. Однак такий інструмент не забезпечує автоматизованої підтримки методології – усі математичні моделі й принципи прийняття рішень стосовно обчислень користувач повинен формувати самостійно. Подібні обмеження притаманні й системам математичного моделювання, зокрема MATLAB, які хоч і є потужними інструментами для оптимізації, проте не

мають вбудованих рішень для конкретних моделей оцінювання.

Існують також спеціалізовані програмні продукти для підтримки для багатоаспектного аналізу використовуються програмні продукти на кшталт Decision Lab, Expert Choice, SuperDecisions, а також веб-платформи MCDA Web чи Expert Choice Online. Ці інструменти втілюють стандартні підходи, серед яких – ієрархічний метод аналізу, а також інші математичні алгоритми аналізу ієрархій, метод ідеального рішення та інші – однак реалізація функції бажаності за Харрінгтоном у таких системах, як правило, не передбачено.

Таким чином, беручи до уваги як аналітичну гнучкість, так і прикладну ефективність, можна зробити висновок, що застосування функції Харрінгтона є раціональним вибором для оцінки складних об'єктів, зокрема – науково-технічних проєктів. Це зумовлює актуальність завдання створення інформаційної системи, що використовує дану функцію як основний критерій для оцінювання та вибору у багатокритеріальних умовах.

### **1.5. Висновки до першого розділу**

У результаті виконання першого розділу дипломного проєкту було досягнуто наступних цілей:

- 1) проаналізовано сучасний стан процесів щодо дослідження й оцінювання наукових і техніко-наукових ініціатив;
- 2) детально проаналізовано основні методи багатокритеріального оцінювання, що застосовуються в подібних задачах;
- 3) детально проаналізовано особливості та переваги використання узагальненої функції бажаності Харрінгтона;
- 4) проведено порівняння цього методу з іншими підходами, зокрема методом ідеальної точки та методом аналізу ієрархій;
- 5) обґрунтовано доцільність використання функції бажаності Харрінгтона як математичної основи для майбутньої системи;
- 6) розглянуто технічні засоби, які можуть бути використані для

реалізації системи оцінювання, визначено їх переваги та обмеження.

На основі виконаного аналізу зроблено висновок, що створення прикладного програмного забезпечення для підтримки прийняття рішень щодо вибору науково-технічних проєктів з використанням узагальненої функції Харрінгтона є актуальним і обґрунтованим напрямом розробки.

## 2. МОДЕЛЬ ОЦІНЮВАННЯ НАУКОВИХ ПРОЄКТІВ

### 2.1. Огляд інструментальних засобів для відбору наукових і науково-технічних розробок

В рамках дослідження для реалізації системи було задіяно модель, засновану на узагальненій функції бажаності, яку запропонував Харрінгтон. Такий підхід дозволяє перевести різнорідні показники оцінювання в уніфіковану шкалу бажаності, що варіюється у діапазоні значень між 0 і 1. Значення, близьке до нуля, сигналізує про неприйнятність критерію, тоді як значення, наближене до одиниці, вказує на найвищу відповідність заданим очікуванням.

Побудова шкали ґрунтується на логарифмічному перетворенні, що зумовлює появу характерних контрольних точок: 0,37 і 0,63. Перше з цих значень відповідає пороговому рівню прийнятності (приблизне відношення  $1/e$ ), а друге – позначає межу задовільних характеристик ( $1-1/e$ ). Саме між цими значеннями формується зона середньої бажаності. Візуально шкала розбивається на п'ять інтервалів, кожен з яких має словесне тлумачення – від "дуже погано" до "дуже добре". Це дозволяє не лише кількісно, а й якісно інтерпретувати результати оцінки.

Таблиця 2.1

Шкала бажаності Харрінгтона

| Шкала «d» | Якісні характеристики шкали «d» |
|-----------|---------------------------------|
| 0.80-1.00 | Дуже добре                      |
| 0.63-0.80 | Добре                           |
| 0.37-0.63 | Задовільно                      |
| 0.20-0.37 | Погано                          |
| 0.00-0.20 | Дуже погано                     |

Існує кілька підходів до побудови функцій бажаності залежно від характеру критеріїв. Якщо значення критерію мають оптимум всередині інтервалу, використовується функція з двобічними обмеженнями – графік такої функції нагадує колоколоподібну криву. У випадку, коли потрібно максимізувати або мінімізувати значення, доцільно застосовувати односторонні обмеження. Ці варіанти дозволяють гнучко адаптувати модель під специфіку конкретних задач.

Разом із тим, базовий варіант функції Харрінгтона не враховує низку важливих аспектів, які виникають у реальних задачах прийняття рішень. Зокрема, вона не передбачає можливості задання несиметричних інтервалів для критеріїв із неоднаковою чутливістю до відхилення вгору чи вниз від ідеального значення. У практиці трапляються ситуації, коли допустиме зменшення параметра майже не впливає на результат, а його зростання – критичне, або навпаки. Щоб урахувати ці випадки, у даному розділі буде обґрунтовано та запропоновано модифіковану математичну модель з несиметричними двосторонніми обмеженнями.

Ще одне обмеження класичної формули – рівнозначне трактування всіх критеріїв. Відсутність механізму вагового коригування робить модель нечутливою до пріоритетності характеристик, що в ряді випадків може суттєво спотворювати підсумкову оцінку. Для вирішення цієї проблеми буде розглянуто методику інтеграції вагових коефіцієнтів до обчислення загальної бажаності, що забезпечить більш точну відповідність результатів реальним потребам користувача.

Таким чином, у цьому розділі особливу увагу буде приділено удосконаленню теоретичної бази моделі Харрінгтона – шляхом розширення її функціональності відповідно до вимог системи, яка розробляється.

## **2.2. Критерії, для яких визначено одnobічне обмеження**

У попередньому підрозділі розглянуто загальні принципи застосування функції бажаності Харрінгтона як математичної основи для



оцінювання проєктів. Залежно від характеру зміни значень критерію та впливу цієї зміни на загальну оцінку, функцію бажаності можна побудувати у двох варіантах: із односторонніми або двосторонніми обмеженнями.

Коли зростання значення критерію веде до покращення його оцінки, говорять про функцію з позитивним одностороннім обмеженням. І навпаки, у випадках, коли підвищення значення критерію є небажаним, застосовують спадну форму цієї функції. Такий підхід дає змогу враховувати як ті параметри, де максимізація є ціллю, так і ті, де слід прагнути мінімізації.

На графіку, що відображає шкалу бажаності для односторонніх залежностей, по осі абсцис відкладається перетворене значення оцінюваного показника, тоді як по осі ординат – розраховане значення бажаності у діапазоні значень між 0 і 1. Типова крива має сигмоїдальну форму, що відображає нелінійну залежність між значенням показника і відповідною якісною оцінкою. Зростання або спадання кривої залежить від того, чи показник є позитивним чи негативним чинником впливу.

Розрахунок значення бажаності для такого типу функції виконується за допомогою наступного математичного виразу:

$$d = e^{-(e^{-\hat{Y}})}, \quad (2.1)$$

де використовуються:

- стала  $e$ , що є основою натурального логарифму;
- перетворене значення змінної, що забезпечує масштабування та нормалізацію уніфікованої шкали.

Для приведення початкового значення критерію до стандартизованого вигляду використовується лінійне перетворення. Воно дозволяє перевести фактичне значення в уніфіковані межі, у межах яких функція бажаності поводить себе стабільно:

$$\hat{Y} = -2 + 7 * \frac{(Y - Y_{min})}{(Y_{max} - Y_{min})}, \quad (2.2)$$

де  $Y_{max}, Y_{min}$  – відповідно мінімально та максимально допустимі значення критерію, що вважаються задовільними для оцінки.

Візуалізація такої функції представлена на рисунку нижче (рис.2.1). Вона ілюструє, яким чином змінюється значення бажаності залежно від входу, який піддався лінійному перетворенню.

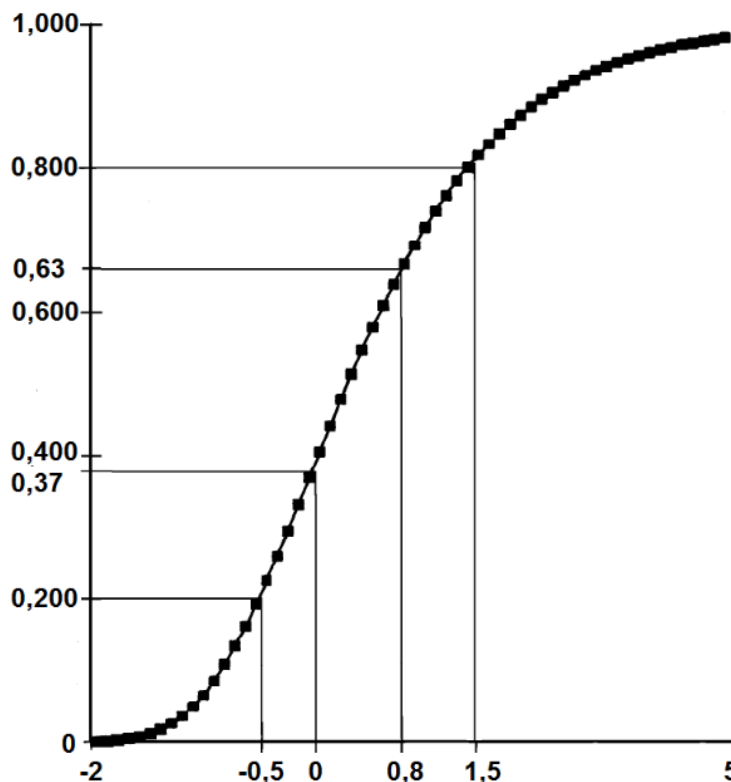


Рис. 2.1. Крива бажаності Харрінгтона (однобічне обмеження)

Якщо розглядати ситуацію, коли зменшення значення показника є позитивним фактором, змінюється порядок введення меж у формулу трансформації. Зокрема, менше значення подається як верхнє обмеження, а більше – як нижнє. Це дозволяє уникнути зміни самої функції та її форми, оскільки результат нормалізації все одно залишатиметься в межах від  $-2$  до  $5$ . Такий підхід уніфікує логіку обробки різних типів критеріїв і спрощує подальшу реалізацію обчислень у програмному середовищі.

### 2.3. Критерії, для яких визначено двобічне обмеження

У разі оцінювання параметрів, для яких важливим є знаходження в певному інтервалі, а не просто максимізація чи мінімізація значення, доцільно використовувати двосторонній підхід до побудови функції

бажаності. У класичному підході, запропонованому Д. Харрінгтоном, розглядається симетричний варіант, за якого ступінь зниження бажаності по обидва боки від найкращого значення однаковий. Така модель зручна, коли відхилення від оптимуму є рівнозначно небажаними в обох напрямках. Візуально ця функція має дзвоноподібну форму, як показано на наступному рисунку (рис.2.2).

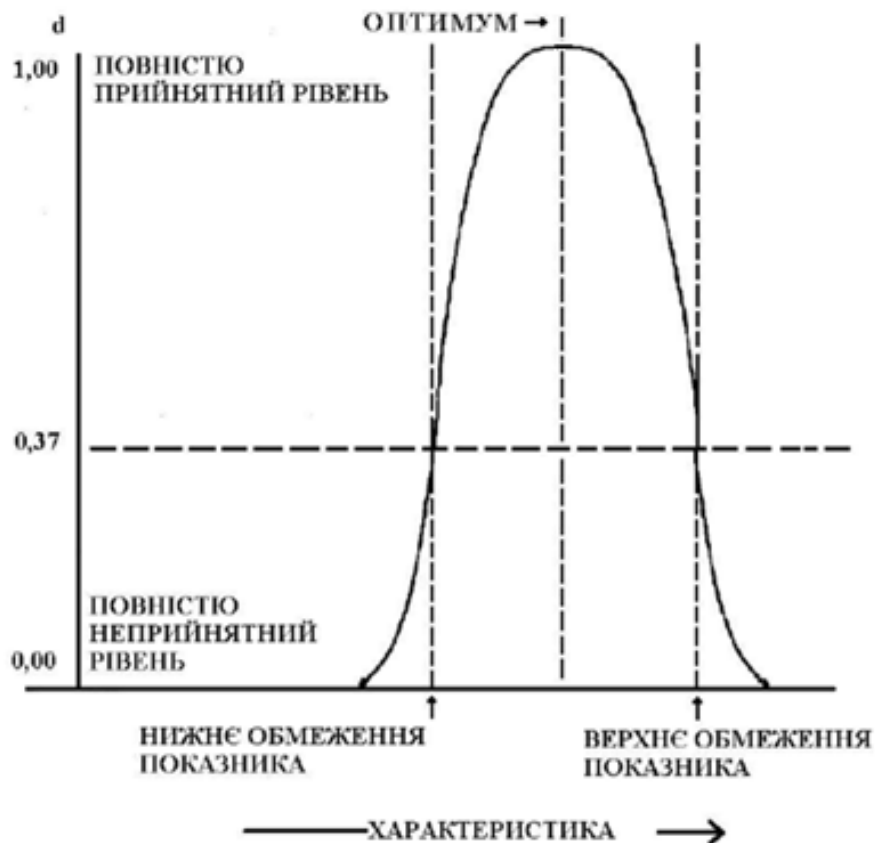


Рис. 2.2. Крива бажаності Харрінгтона (двостороннє обмеження)

Проте варто зауважити, що у практичних завданнях трапляються ситуації, коли симетричність не відповідає реальній природі оцінюваного показника. Наприклад, у випадку температури тіла, оптимумом є  $36,6^{\circ}\text{C}$ , але відхилення в менший бік можуть бути критичнішими, ніж у більший, або навпаки. Тому для більшої гнучкості у моделюванні доцільно використовувати несиметричні функції бажаності, в яких поведінка функції ліворуч і праворуч від оптимуму описується різними параметрами.

Загальна формула бажаності для симетричної двосторонньої моделі задається виразом:

$$d = e^{-(|\dot{Y}|)^n}, \quad (2.3)$$

де:

- $e$  – стала натурального логарифма ( $\approx 2.718$ );
- $n$  – позитивне число, що формує нахил функції;
- $\dot{Y}$  – результат перетворення значення змінної  $Y$ .

Перетворена величина, отримана шляхом нормалізації вхідного параметра відносно середнього значення діапазону:

$$\dot{Y} = \frac{2Y - (Y_{max} + Y_{min})}{Y_{max} - Y_{min}}, \quad (2.4)$$

де:

- $Y$  – значення показника;
- $Y_{max}, Y_{min}$  – мінімально та максимально припустимі значення оцінки; [10]

Параметр  $n$ , що входить до формули, регулює крутизну спадання функції і визначається залежно від обраної точки бажаності  $d$  та відповідного значення трансформованої змінної:

$$n = \frac{\ln \ln(\frac{1}{d})}{\ln |\dot{Y}|}. \quad (2.5)$$

Підбір параметра  $n$  виконується емпірично з урахуванням особливостей кожного конкретного завдання. Як правило, бажаність у контрольній точці встановлюється в межах 0,6–0,9 для забезпечення плавного переходу між областями «добре» та «задовільно».

У разі, якщо бажаність не знижується симетрично, вводиться комбінована функція, яка об'єднує два симетричні елементи з різними межами та параметрами кривизни. Центром такої функції є оптимальне значення критерію, у якому бажаність вважається максимальною й дорівнює 1. При цьому поведінка функції зліва і справа задається незалежно.

Формально така функція може бути записана у вигляді:

$$d = \begin{cases} e^{-(|\dot{Y}_1|)^{n_1}}, & \text{якщо } Y < Y^* \\ 1, & \text{якщо } Y = Y^* \\ e^{-(|\dot{Y}_2|)^{n_2}}, & \text{якщо } Y > Y^* \end{cases}, \quad (2.6)$$

де:

- $Y^*$  – оптимальне значення;
- $Y$  – реальне значення оцінюваного показника;
- $n_1, n_2$  – позитивні параметри (можуть бути дробовими), що впливають на крутість кривої біля нижньої та верхньої межі;
- $\dot{Y}_1, \dot{Y}_2$  – значення функції після лінійного перетворення змінної  $Y$  з обох сторін від оптимального рівня.

Даний вираз описує поведінку бажаності в залежності від того, чи значення критерію менше, дорівнює або перевищує оптимум  $Y^*$ . Кожна з частин використовує свою трансформовану змінну.

Зокрема, для ділянки, що розташована ліворуч від оптимального значення, нижню межу можна вивести, знаючи оптимум і фактичну нижню границю. Через симетрію лівої частини цей розрахунок виконується за такою формулою:

$$Y^* = \frac{Y_{min} + Y_{max}}{2} \Rightarrow Y_{max} = 2Y^* - Y_{min}. \quad (2.7)$$

Підставляючи отримане значення у формулу нормалізації, отримаємо:

$$\dot{Y}_1 = \frac{2Y - (2Y^* - Y_{min} + Y_{min})}{2Y^* - Y_{min} - Y_{min}} \Rightarrow \dot{Y}_1 = \frac{Y - Y^*}{Y^* - Y_{min}}. \quad (2.8)$$

Аналогічним чином визначається трансформація для правої частини функції, використовуючи праву межу та оптимум:

$$\dot{Y}_2 = \frac{Y - Y^*}{Y_{max} - Y^*}. \quad (2.9)$$

Графічне представлення такої несиметричної функції наведено нижче (рис.2.3).

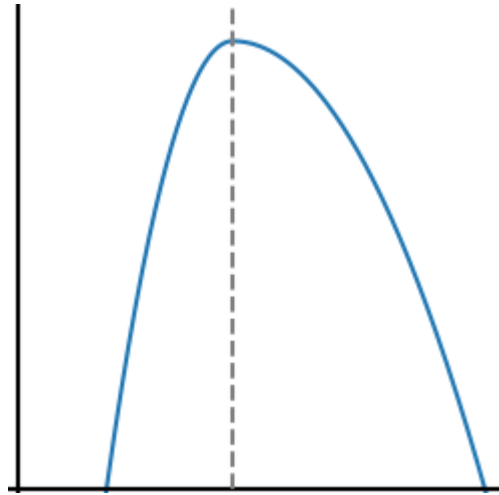


Рис. 2.3. Функція бажаності (несиметричне двобічне обмеження)

Завдяки поділу функції на два незалежні фрагменти, кожен з яких має власну параметризацію, стає можливим точніше моделювання тих ситуацій, де чутливість до відхилення змінюється залежно від напрямку. Такий підхід розширює можливості адаптації функції бажаності до реальних оцінювальних задач у системах підтримки прийняття рішень.

#### **2.4. Розрахунок інтегрального показника бажаності**

Для здійснення порівняльної оцінки альтернатив при наявності різних критеріїв, вводиться шкала бажаності, оскільки показники вимірюються в різних одиницях. Це дозволяє трансформувати початкову задачу з різнорідними критеріями у задачу, де всі оцінки подаються в уніфікованій формі. Наступним логічним кроком є об'єднання окремих функцій бажаності в єдиний узагальнений індикатор.

Один із підходів полягає у використанні геометричного середнього значення всіх часткових оцінок бажаності. Навіть якщо більшість критеріїв мають високі значення, наявність одного критичного (тобто нульового або дуже низького) показника знижує загальну придатність рішення до нуля. Такий принцип особливо доцільний у випадках, коли існують неприпустимі порогові значення. Наприклад, у процесі оцінювання проєктної пропозиції, наявність занадто високої вартості або надмірно тривалого терміну

виконання може зробити її повністю неприйнятною, незалежно від інших переваг. У таких випадках відповідний показник бажаності дорівнює нулю, і результатом стає повне нівелювання загальної оцінки.

Інтегральний показник у цьому випадку визначається наступним чином:

$$D = \sqrt[n]{d_1 * d_2 * ... * d_n}, \quad (2.10)$$

де:

- $d_i$  – значення функції бажаності для  $i$ -го критерію;
- $n$  – загальна кількість критеріїв.

Незважаючи на те, що наведений метод дозволяє об'єднати різноманітні критерії в одну оцінку, він не враховує їхню відносну значущість. У реальності деякі показники можуть мати більший вплив на кінцевий результат, ніж інші. Для врахування цього аспекту доцільно модифікувати базову формулу, ввівши вагові коефіцієнти, які будуть відображати важливість усіх критеріїв.

З огляду на використання геометричного середнього, вагові коефіцієнти не множаться на відповідні значення функцій бажаності, а виступають у ролі показників ступеня. Таким чином, більш значущі критерії отримають вагоміший вплив на кінцевий результат. Щоб забезпечити коректність розрахунків, необхідно нормалізувати коефіцієнти так, щоб їх сума дорівнювала кількості критеріїв  $n$ . В результаті модифікований варіант формули набуває наступного вигляду:

$$D = \sqrt[n]{d_1^{k_1} * d_2^{k_2} * ... * d_n^{k_n}}, \quad (2.11)$$

де:

- $k_i$  – коефіцієнт значущості для  $i$ -го критерію, інші позначення – аналогічні попередній формулі.

При цьому коефіцієнти важливості повинні задовольняти умову нормалізації:

$$\sum_{i=1}^n k_i = n. \quad (2.12)$$

Такий підхід дозволяє не тільки уніфікувати різнотипні критерії, але й врахувати їхній реальний вплив, що значно підвищує обґрунтованість прийнятого рішення.

## 2.5. Добір критеріїв оцінювання

Оцінювання проєктів за допомогою узагальненої функції Харрінгтона, як і при застосуванні інших підходів у задачах багатокритеріального аналізу, значною мірою залежить від достовірності експертних оцінок. Якщо початкові дані не будуть достатньо точними, обґрунтованими й узгодженими, жоден розрахунковий метод не забезпечить надійного результату. Саме тому для кожної конкретної ситуації необхідно ретельно визначити релевантні критерії, що дадуть змогу мінімізувати вплив суб'єктивного чинника. Хоча набір критеріїв може змінюватися залежно від проєкту, існують певні універсальні показники, що часто виявляються актуальними. У рамках даного проєкту обрано такі критерії, для яких експерти встановлюватимуть межі прийнятності, коефіцієнти значущості та поточні значення, необхідні для подальшого оцінювання проєкту.

Кошторис проєкту – один із базових індикаторів, який характеризує необхідний обсяг фінансування. Враховуючи його природу, доцільно застосовувати модель, де діє одностороннє обмеження, що, що передбачає мінімізацію значення цього показника.

Строк виконання – показує тривалість реалізації проєкту. Як і у випадку з кошторисом, метою є мінімізація часу, отже, також використовується функція з одностороннім обмеженням.



Новизна – відображає рівень оригінальності й наукової унікальності результатів, починаючи від добре відомих до повністю нових відкриттів. На відміну від попередніх критеріїв, новизна має якісну природу, тому її оцінювання здійснюється експертами, бажано в більшій кількості для досягнення об'єктивності. Щоб полегшити інтерпретацію, застосовується вербальна шкала з градаціями, від «незадовільного» до «ідеального». У цьому випадку використовують функцію максимізації з одностороннім обмеженням.

Значимість – визначає, наскільки вагомим буде вплив проєкту на розвиток відповідної наукової або прикладної галузі. Оцінювання доцільно проводити за допомогою вербальної шкали. Враховуючи бажання підвищити важливість показника, використовується однобічна модель, яка фокусується на максимізацію.

Об'єктивність – дає уявлення про ймовірність отримання достовірних результатів у процесі виконання проєкту, а також про наявність чи відсутність факторів, що можуть знизити точність. Використовується функція орієнтована на досягнення найкращого результату.

Професійний рівень учасників – відображає відповідність компетентності команди складності завдань, що мають бути реалізовані в межах проєкту. Використовується однобічна модель, яка фокусується на максимізацію, та вербальна шкала для фіксації оцінки.

Частка ресурсів, спрямованих на теоретичні дослідження, у деяких випадках викликає необхідність балансування між вкладенням у теоретичну й практичну частини дослідження. Надто низький або надто високий відсоток теоретичної складової може негативно вплинути на загальний успіх проєкту. Отже, доцільно використати модель з двобічним обмеженням, яка допомагає знайти оптимальне значення цього показника.

Залучення учасників освітнього процесу – оскільки фінансування багатьох науково-технічних ініціатив здійснюється за участю держави, участь представників освітнього середовища (викладачів, студентів тощо)

може вважатися позитивним чинником. Водночас надмірна участь осіб із недостатнім досвідом може знизити ефективність виконання проєкту. Тому для цього критерію доцільно використати функцію з двостороннім обмеженням та оцінювання за допомогою вербальної шкали.

## **2.6. Висновки до другого розділу**

У цьому розділі було досліджено підхід до оцінювання наукових та науково-технічних проєктів на основі функції бажаності Харрінгтона. Така модель виявилася зручною для уніфікації різнорідних критеріїв в єдину шкалу, що дозволяє порівнювати проєкти за якісними та кількісними характеристиками одночасно. Аналіз показав, що використання шкали бажаності спрощує інтерпретацію результатів, завдяки чіткому розподілу на рівні якості – від «дуже погано» до «дуже добре».

Залежно від характеру оцінюваного показника, було розглянуто побудову функцій із односторонніми та двосторонніми обмеженнями. У першому випадку модель дозволяє коректно враховувати бажаність підвищення або зниження параметра. У другому – підходить для тих ситуацій, коли найкращий результат розміщено всередині певного інтервалу, і відхилення в будь-який бік є небажаним. При цьому запропоновано варіанти як симетричної, так і несиметричної реалізації функції, що забезпечує більшу точність у відображенні реальних умов.

Також було звернено увагу на обмеження класичної моделі – зокрема, її нечутливість до вагомості критеріїв та однакове ставлення до відхилень у різні боки. Для усунення цих недоліків запропоновано вдосконалення, які дозволяють гнучко адаптувати модель під конкретні вимоги користувача.

Отже, модифікована модель оцінювання, що базується на розширеній функції бажаності Харрінгтона, є доцільною основою для побудови ефективної системи підтримки прийняття рішень у сфері науково-технічного аналізу проєктів.

### **3. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ**

У цьому розділі наведено обґрунтування вибору мови програмування, бібліотек, інструментів розробки та формату вхідних даних, що були використані під час створення програмного забезпечення для механізму прийняття рішень щодо НД проєктів. Було обрано засоби забезпечують ефективну реалізацію функціональності, зручність користувацької взаємодії та можливість подальшої масштабованості розробленої системи.

#### **3.1. Вибір мови програмування для реалізації програмного забезпечення**

Розробка моделі для оцінювання та добору науково-технічних ініціатив здійснювалася з використанням мови програмування Python. Це мова програмування широкого застосування високого рівня, що дозволяє використання широкої екосистеми бібліотек та високу гнучкість у розв'язанні прикладних задач. Основна філософія мови Python – зрозумілий та читабельний код, що досягається, зокрема, за допомогою значущих відступів (правило «off-side»), які формують структуру програми.

Python широко використовується в галузях аналізу даних, машинного навчання, наукових обчислень, автоматизації процесів та створення інтерфейсів користувача. Для реалізації програмного забезпечення в межах цього проєкту було обрано такі бібліотеки та модулі:

1. **Openpyxl** – інструмент для взаємодії з електронними таблицями Excel: відкриття, редагування, створення нових листів і налаштування зовнішнього вигляду документів. Є особливо важливим для обробки вхідних таблиць користувача.
2. **Math** – вбудований модуль для виконання базових та складних математичних обчислень. Зокрема, він дозволяє виконувати тригонометричні, логарифмічні обчислення, округлення та інші дії, необхідні для реалізації логіки розрахунку критеріїв оцінки.

3. Модулі `tkk`, `filedialog` і `messagebox` належать до вбудованого інструментарію для побудови графічного інтерфейсу користувача. Вона дозволяє створювати вікна, кнопки, поля введення, списки тощо. Розширення `tkk` додає сучасні стилі віджетів, `filedialog` використовується для відкриття а також збереження інформації у файл. `messagebox` – для виведення інформаційних повідомлень і попереджень.
4. `Matplotlib` – потужний інструмент який дозволяє малювати графіки та різнотипні візуалізації. Вона використовується для графічного представлення результатів оцінки, що дозволяє користувачам краще інтерпретувати дані та приймати обґрунтовані рішення.

Для написання та налагодження коду було обрано IDE `PyCharm`, що забезпечує зручне програмування, налагодження та запуск коду. Це дозволяє ефективно організовувати роботу над проєктом та забезпечувати високу якість програмного коду.

Таким чином, поєднання мови програмування `Python` з потужними бібліотеками та зручним середовищем розробки `PyCharm` дає можливість створити ефективне, гнучке та зручне у використанні програмне забезпечення.

### **3.2. Обґрунтування формату вхідних даних**

Для подання вхідних даних у розробленій системі було обрано формат електронних таблиць – зокрема, файли формату `Microsoft Excel (XLSX)`. Такий вибір є цілком обґрунтованим з кількох причин:

По-перше, електронні таблиці є звичним та зручним інструментом для користувачів різного рівня підготовки. Програми на кшталт `Microsoft Excel`, `Google Sheets` або `LibreOffice Calc` використовуються для створення, редагування та аналізу табличних даних у багатьох сферах, зокрема у науковій, освітній, інженерній та управлінській діяльності.

По-друге, таблична структура ідеально підходить для зберігання та організації вхідних даних, таких як перелік проєктів, значення критеріїв оцінки, вагові коефіцієнти, результати попередніх етапів тощо. Таблиці дозволяють логічно впорядкувати дані, що спрощує їх обробку та забезпечує цілісність структури при імпорті в програму.

По-третє, використання електронних таблиць дозволяє спростити процес інтеграції з іншими програмами та джерелами даних. Наприклад, результати, отримані за допомогою сторонніх систем або програм, можуть бути експортовані до Excel і використані в даній системі без потреби у додатковому форматуванні.

Крім того, бібліотека `openpyxl`, яка використовується у програмному забезпеченні, забезпечує надійний та ефективний інструментарій для зчитування даних з Excel – файлів, що дозволяє легко інтегрувати цей формат у програму. Програма автоматично зчитує таблицю, перевіряє її структуру, обробляє дані та виконує відповідні розрахунки.

Таким чином, формат Excel є логічним та технічно обґрунтованим вибором для подання вхідних даних у системі оцінки науково-технічних проєктів. Він забезпечує гнучкість, зручність та високу сумісність з іншими інструментами, що значно підвищує ефективність роботи кінцевих користувачів.

### **3.3. Висновки до третього розділу**

У результаті проведеного обґрунтування вибору засобів розробки програмного забезпечення можна зробити висновки, що вибір мови програмування Python є цілком виправданим. Висока популярність мови, її простота та гнучкість роблять Python ідеальним інструментом для створення систем, які потребують швидкої розробки, зручної роботи з даними та інтеграції з різними бібліотеками. Бібліотеки, такі як `openpyxl`, `math`, `tkinter`, `matplotlib`, значно розширюють функціональні можливості програми, дозволяючи ефективно обробляти таблиці, виконувати

математичні операції, створювати графічні інтерфейси та візуалізації даних. Вибір середовища розробки PyCharm також підтверджує свою ефективність завдяки широким можливостям для налагодження та організації роботи.

Що стосується формату вхідних даних, вибір електронних таблиць формату Microsoft Excel є обґрунтованим з кількох важливих причин. Цей формат є широко поширеним та зручним для кінцевих користувачів, що дозволяє значно спростити процес роботи з даними. Використання табличної структури забезпечує логічну організацію та зручність у зберіганні та обробці вхідних даних, таких як переліки проєктів і результати оцінки. Крім того, бібліотека `openpyxl` надає ефективні засоби для зчитування та обробки даних з Excel-файлів, що дозволяє забезпечити високу сумісність із різними джерелами інформації. Цей підхід дозволяє користувачам без труднощів інтегрувати результати з інших програм та систем і спрощує весь процес використання програмного забезпечення.

## **4. ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЇ СИСТЕМИ ОЦІНКИ ТА ВИБОРУ НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ**

У цьому розділі розглянуто ключові аспекти проєктування та реалізації системи, яка дозволяє здійснювати оцінку та порівняння наукових проєктів за визначеними критеріями. Система передбачає можливість інтерактивної роботи користувача з даними, автоматизованого розрахунку рівнів бажаності, а також зручного подання результатів у графічному й табличному вигляді.

У межах розробки розглядається архітектура майбутнього програмного забезпечення, основні його компоненти та взаємодія між ними. Для наочного відображення структури й логіки роботи системи використовуються різноманітні діаграми, які демонструють як загальний підхід до обробки вхідної інформації, так і окремі процеси, що відбуваються в межах кожного функціонального модуля.

Також у межах розділу формалізуються вимоги до функціональності системи, її взаємодії з користувачем, обробки вхідних даних та подання вихідних результатів. Важливим аспектом є забезпечення зрозумілого та інтуїтивного інтерфейсу, що дозволяє працювати з системою навіть користувачам без глибоких технічних знань. У результаті отримується інструмент, який сприяє підвищенню об'єктивності та ефективності процесу вибору серед альтернативних проєктних рішень.

### **4.1. Опис вимог до розроблюваної системи**

Розроблювана система повинна відповідати як функціональним, так і нефункціональним вимогам, щоб забезпечити повноцінну підтримку процесу оцінювання об'єктів за заданими критеріями.

Функціональні вимоги:

1. Система має забезпечувати можливість завантаження вхідної інформації у форматі електронних таблиць. Кожна таблиця повинна

містити дані про окремий об'єкт, що підлягає аналізу або оцінюванню.

2. Користувачу необхідно надати зручний інтерфейс, який містить перелік доступних функцій. Серед можливих дій – створення рейтингових списків з урахуванням обраних параметрів, обчислення інтегрального показника при включенні або виключенні певних критеріїв тощо.
3. Після вибору конкретної дії система має виконати її автоматично, обробивши дані відповідним чином.
4. Інформацію, отриману в результаті обробки, слід подавати в наочному вигляді. Окрім текстових пояснень, система повинна підтримувати графічні способи подачі даних – у вигляді гістограм, кругових діаграм, графіків тощо, залежно від контексту інформації.
5. Користувач має мати можливість зберегти результати у зручному форматі для подальшого використання.

#### Нефункціональні вимоги

1. Інтерфейс системи повинен бути інтуїтивно зрозумілим, щоб забезпечити можливість роботи з нею навіть для користувачів без спеціальної підготовки чи досвіду роботи з подібними системами.
2. Система повинна враховувати валідацію введеної інформації. У випадку виявлення помилки система має видавати інформативне повідомлення, що дозволяє користувачу її виправити.

## **4.2. Аналіз проблематики та цілей вебзастосунку**

Програмне забезпечення для оцінки ефективності наукових та науково-технічних проєктів має низку проблем, які можуть впливати на його ефективність та зручність використання. Перш за все, виникають труднощі з обробкою даних, зокрема з неправильною формою або форматом вхідних файлів, а також з невідповідністю структури даних вимогам системи. Це може призвести до значних помилок під час оцінки проєктів.



Додатково, проблеми з інтерфейсом користувача, такі як неясність у функціональності та незрозумілий, неінтуїтивно зрозумілий дизайн, ускладнюють взаємодію користувача із системою. Нарешті, обмежена кількість варіантів візуалізації результатів та повільна обробка великих обсягів даних можуть знижувати ефективність системи та ускладнювати аналіз даних. Для підвищення ефективності та зручності використання, важливо вирішити ці проблеми шляхом вдосконалення системи, покращення інтерфейсу, оптимізації обробки даних і забезпечення більшої гнучкості у візуалізації результатів.

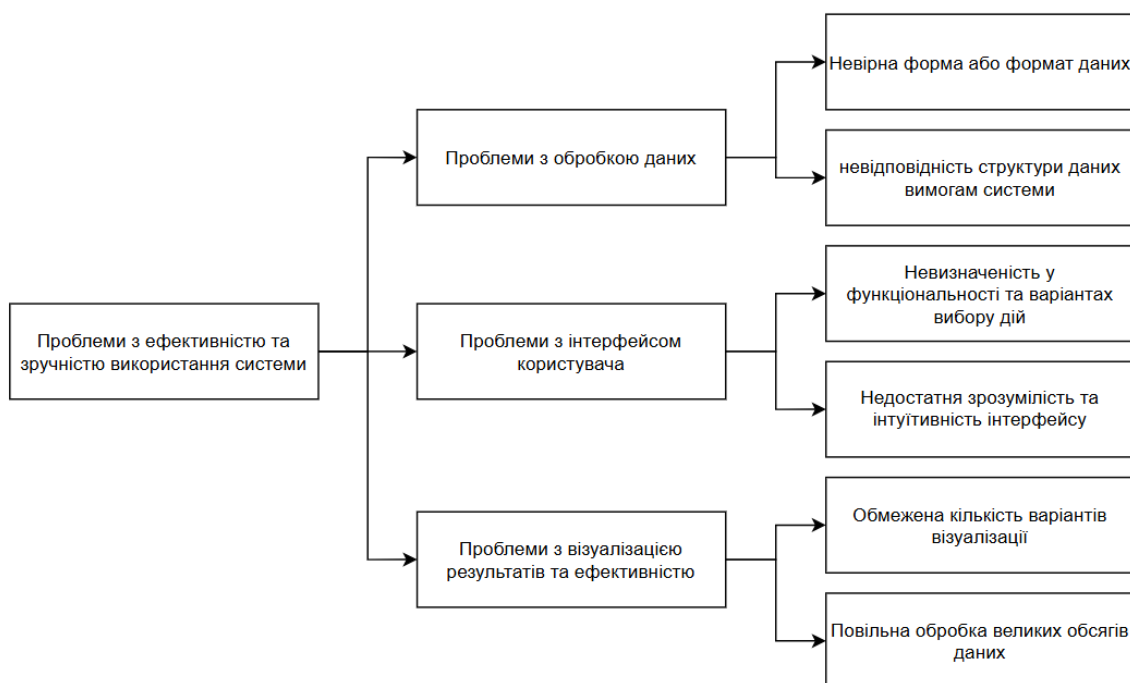


Рис. 4.1. Дерево проблем

### 4.3. Опис архітектури системи

Для реалізації ефективного та структурованого підходу до створення системи оцінювання науково-технічних ініціатив, необхідно окреслити її архітектурну модель. Це дозволяє визначити основні складові системи, їхні функції та взаємозв'язки між ними. Нижче представлено загальну структурну схему системи, де відображено компоненти клієнтської та серверної частин, а також логіку їхньої взаємодії.

Клієнтський рівень включає модуль інтерфейсу, через який здійснюється взаємодія користувача із системою, та блок обробки введених даних, що забезпечує їх первинну упаковку для подальшої передачі на сервер.

На серверному боці зосереджено основні функціональні процеси: аналіз значень, наданих користувачем; обчислення рейтингів за окремими критеріями; обробка сукупних оцінок проєктів; підготовка даних до виводу у зручному вигляді. Усі ці дії виконуються з метою надання об'єктивної та структурованої оцінки кожному проєкту.

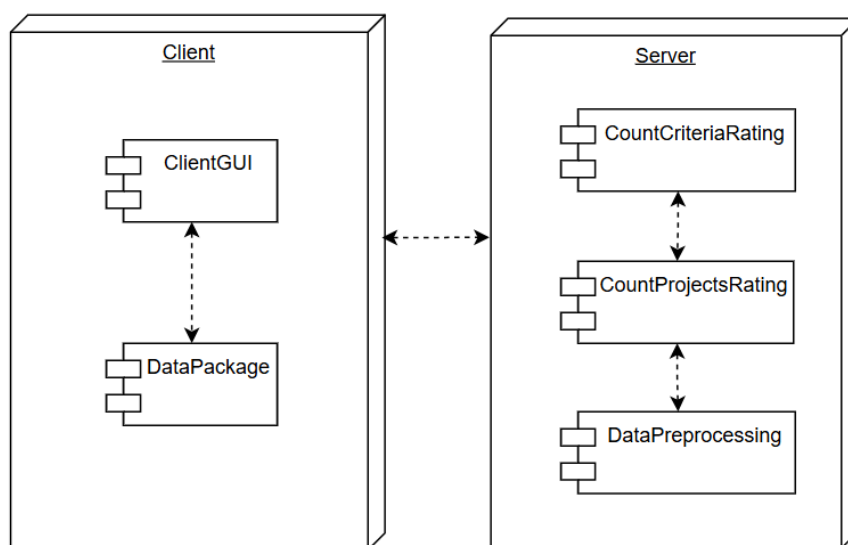


Рис. 4.2. Структура основних компонентів системи

З метою візуалізації логіки роботи системи та кроків, які виконуються під час аналізу й прийняття рішень, створюється набір діаграм, що описують ключові процеси. Такий підхід дозволяє побачити повну картину функціонування програмного забезпечення – від моменту введення даних до отримання кінцевого результату.

Одна з базових моделей, що описує загальну логіку функціонування системи – це схема у форматі IDEF0. Вона дає змогу чітко уявити, які вхідні параметри надходять до системи, які ресурси чи інструменти використовуються під час обробки, та який формується результат.



Рис. 4.3. Діаграма у форматі IDEF0

Для більш детального подання функціональності кожного етапу застосовується декомпозиція у вигляді DFD-діаграми, що демонструє ієрархічну структуру виконання завдань у межах інформаційної системи. У цьому проєкті розрізняються три основні функціональні блоки: обчислення індивідуальних оцінок за критеріями, формування зведених оцінок проєктів та генерація вихідної інформації. Користувач подає на вхід критерії оцінювання, бажані значення та відповідні межі, а також самі оцінки по проєктах. Після проходження зазначених етапів, система надає результати у вигляді інтегральних оцінок, окремих характеристик та графічних візуалізацій впливу критеріїв.

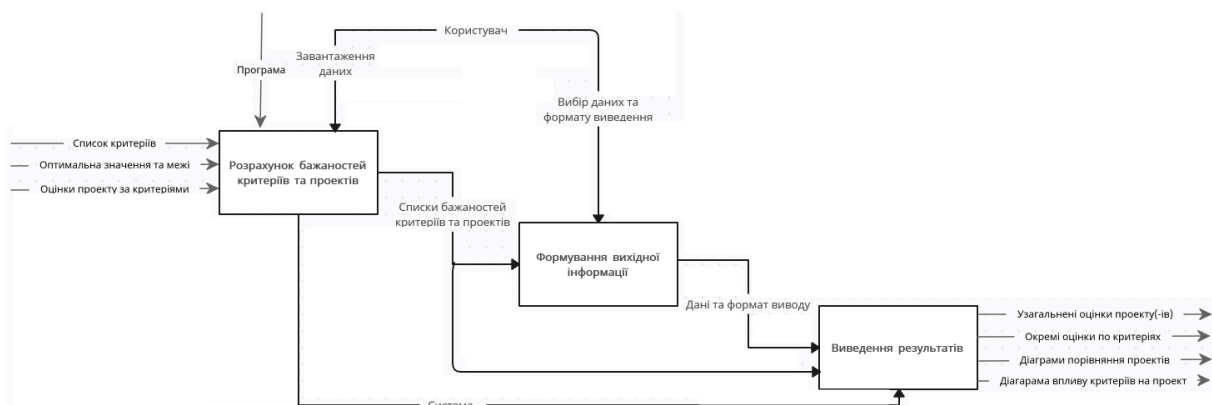


Рис. 4.4. DFD-схема функціональної структури

З метою кращого розуміння послідовності подій у межах роботи системи доцільно використати подієво-орієнтовану модель – EPC-діаграму. У такій діаграмі чітко простежується логіка переходів між станами, залежно від подій, що відбуваються, і дій, які запускаються в результаті.

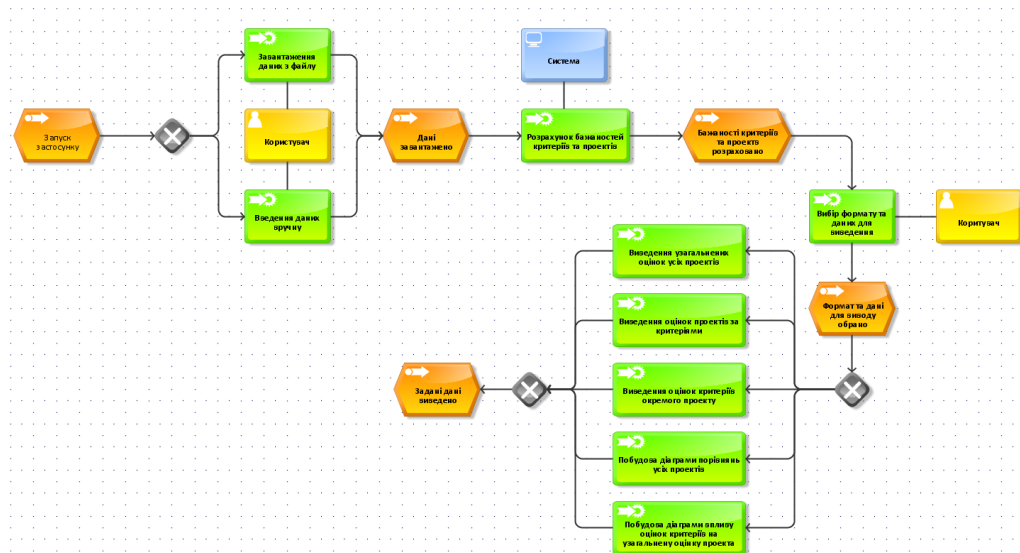


Рис. 4.5. EPC-діаграма основних процесів

Інструментом для моделювання архітектури є use-case діаграма, яка дозволяє виявити взаємозв'язок між кінцевими користувачами (акторами) та функціоналом системи. На такій схемі можна побачити, які саме дії доступні користувачеві та як вони реалізуються в межах ПЗ.

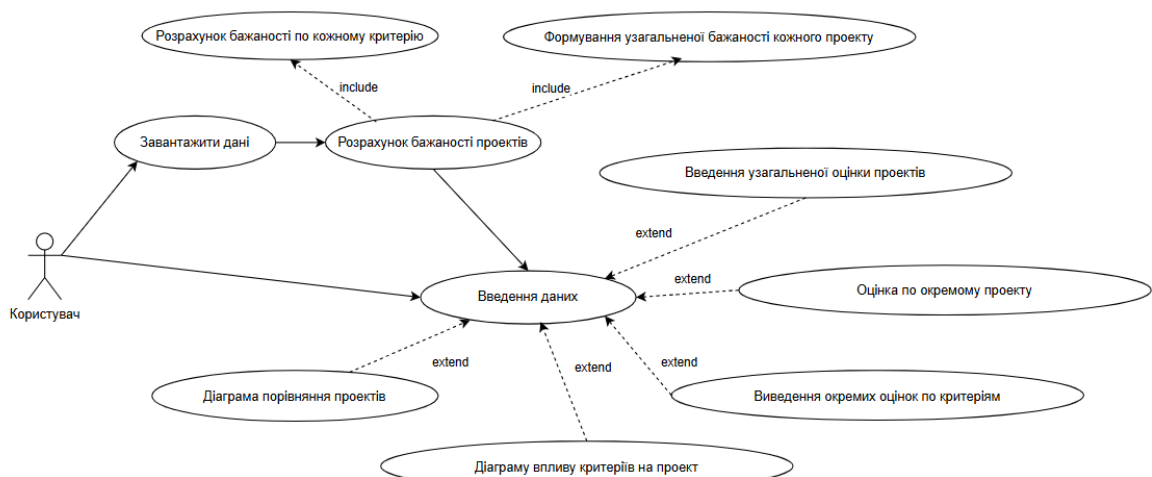


Рис. 4.6. Діаграма варіантів використання (use-case)

#### 4.4. Особливості реалізації

Перед тим як перейти до кодування, на основі попередніх досліджень була побудована структурна схема, що ілюструє основні етапи роботи системи (рис. 4.7).



Рис. 4.7. Схема функціонування системи

Для реалізації методики Харрінгтона у рамках цього підходу необхідно створити дві окремі функції на Python: одна для випадків із одностороннім обмеженням, інша – для двостороннього. Обидві функції приймають три вхідні параметри: нижню межу прийнятного значення ( $x_{low}$ ), верхню межу ( $x_{up}$ ) і конкретне значення показника ( $x$ ). На виході очікується значення бажаності для відповідного критерію.

При реалізації формул, поданих у другому розділі, використовуватиметься адаптований підхід. Для одностороннього обмеження застосовується модифікована формула лінійного перетворення, у якій крайні значення бажаності (від  $-2$  до  $5$ ) прив'язуються до меж  $x_{low}$  та  $x_{up}$ . Ці межі не є оптимальними значеннями у класичному сенсі, а радше вказують на порогові – тобто ті, після яких рівень бажаності різко падає або зростає.

У процесі обчислення слід враховувати можливість поділу на нуль – така ситуація виникає, якщо  $x_{low}$  дорівнює  $x_{up}$ . Щоб уникнути помилки, у такому випадку алгоритм задає заздалегідь визначене значення бажаності, базуючись на логічних припущеннях. Цей підхід актуальний як для одностороннього, так і для двостороннього обмеження, оскільки в обох ситуаціях це дозволяє уникнути виключних ситуацій.

У разі, якщо експерт вказав лише одне граничне значення (тобто  $x_{low} = x_{up}$ ), поведінка функцій буде наступною:

Для одностороннього обмеження:

- 1) якщо показник перевищує задане значення (або менше, якщо мета – мінімізація), бажаність дорівнює  $1$ ;
- 2) якщо він точно дорівнює пороговому значенню – бажаність встановлюється на рівні  $0.55$ ;
- 3) якщо показник гірший за межу (менший або більший, залежно від мети), бажаність становить  $0.1$ .

Для двостороннього обмеження:

- 1) якщо показник відповідає заданому значенню – бажаність максимальна і дорівнює 1;
- 2) в інших випадках – 0.1.

Використання значення 0.1 як мінімального рівня бажаності зумовлено потребою зберегти ненульовий загальний результат при обчисленні середнього геометричного. Це дозволяє уникнути повного занулення оцінки проєкту, що суперечило б оригінальній концепції функції бажаності Харрінгтона. До прикладу, у модифікаціях таких як підхід Деррінджера та С'юча допускається обнулення, але в класичному методі цього немає.

Значення 0.55 обране не випадково: воно відображає середнє між 1 та 0.1, а також близьке до результату, який можна отримати при обчисленні бажаності для значення, що лежить посередині між допустимими межами.

Наступним кроком є обчислення інтегральної оцінки бажаності проєкту. Для цього всі часткові значення об'єднуються за допомогою геометричного середнього з урахуванням вагових коефіцієнтів, як це описано в другому розділі.

Щоб підвищити зручність користування системою, передбачається підтримка обробки вербальних (словесних) значень. Це дозволяє користувачу задавати значення критеріїв у звичній формі – наприклад, "високий рівень" або "середній рівень", – які будуть автоматично перетворені на числові показники.

Розподіл чисел у шкалі (0, 1, 2, 3, 4, 5, 7) підібраний так, щоб відображати ефект уповільнення росту бажаності: при переході на вищі рівні зміни стають менш відчутними. Такий розподіл створює майже рівномірний крок між рівнями сприйняття.

Аналогічно можна застосовувати шкалу для інтерпретації кінцевого числового результату оцінки, зіставляючи отримане значення у діапазоні від 0 до 1 із відповідною вербальною характеристикою.

Для інтерпретації підсумкових результатів оцінювання проєктів застосовується вербальна шкала, яка відповідає певним числовим діапазнам. Зокрема, якщо підсумкова оцінка проєкту перебуває в межах від 0 до 0.5 (не включаючи 0.5), він вважається таким, що є небажаним для реалізації. Значення від 0.5 до 0.65 свідчать про низький пріоритет проєкту. Якщо оцінка знаходиться у проміжку від 0.65 до 0.8, проєкт класифікується як такий, що має середній пріоритет. І, нарешті, якщо результат перевищує 0.8 і не перевищує 1, проєкт оцінюється як високопріоритетний.

#### **4.5. Опис реалізованого користувацького інтерфейсу**

Опис користувацького інтерфейсу реалізованої системи зосереджений на зручності, простоті та логічній структурі взаємодії. При проєктуванні інтерфейсу було враховано вимоги до зручності використання навіть для користувачів без спеціальної технічної підготовки. Завдяки особливостям функціоналу вдалося уникнути перевантаження елементами управління – інтерфейс містить лише необхідні компоненти для забезпечення роботи системи.

У відповідності до логіки, закладеної у процесі моделювання, система складається з трьох логічних етапів: підготовка вхідних даних, вибір типу вихідної інформації, а також представлення результатів. Кожному з етапів відповідає окреме вікно в інтерфейсі.

На першому екрані користувачу доступна лише функція завантаження необхідних вхідних даних, без яких подальше використання системи є неможливим. Цей екран відображає мінімальну кількість елементів управління, що дозволяє сконцентруватися лише на підготовчому етапі. Інші функції стають доступними лише після успішного завантаження даних, що забезпечує послідовність і логіку роботи з системою.



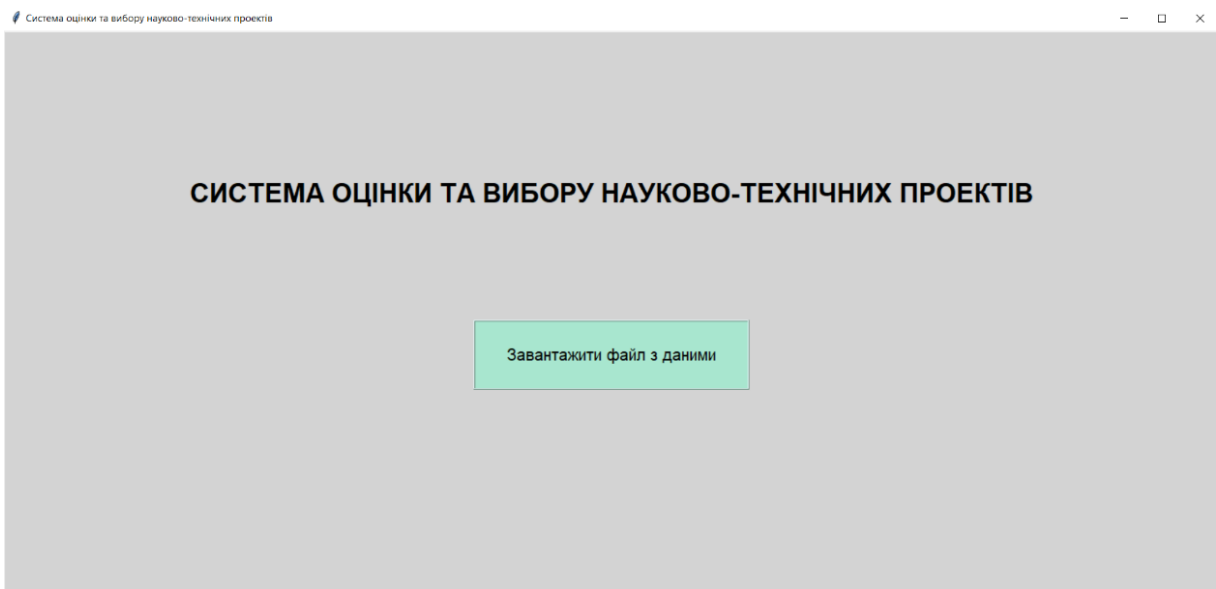


Рис. 4.8. Початковий екран

Після завантаження даних система переходить до другого етапу – формування запиту на виведення певного типу інформації. Користувач має змогу обрати один із кількох варіантів: загальні оцінки проєктів, результати за окремими критеріями, детальна інформація про конкретний проєкт, графічне порівняння результатів або ж аналіз впливу критеріїв на певний проєкт.



Рис. 4.9. Екран з варіантами вихідних даних

У разі вибору таких опцій, як аналіз за критеріями або деталізація одного проєкту, інтерфейс динамічно відображає додаткові елементи управління. Наприклад, для вибору кількох критеріїв використовуються чекбокси, що дозволяють користувачу одночасно позначити кілька варіантів. Якщо потрібно обрати лише один проєкт, система надає випадаючий список з доступними варіантами. Ці елементи зручно розміщено поруч із основною зоною вибору, що забезпечує швидкий доступ без відкриття нових вікон.

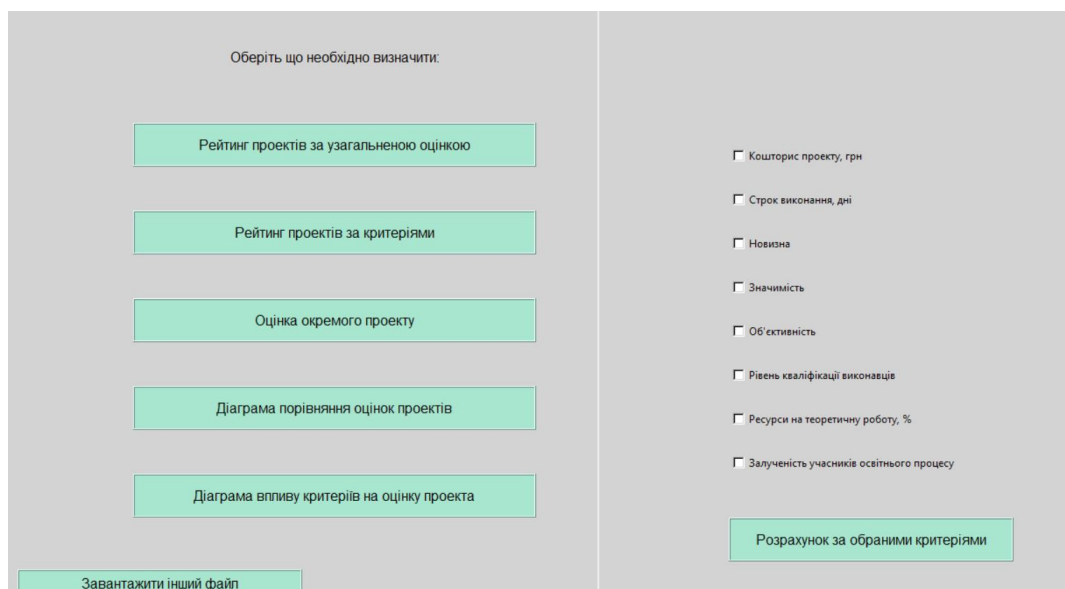


Рис. 4.10. Екран прикладом розміщення додаткових параметрів

Предбачено також можливість повторного завантаження нових вхідних даних без необхідності перезавантаження всієї програми, що робить роботу з системою більш гнучкою та зручною. Для цього в інтерфейсі реалізовано окрему кнопку, яка забезпечує швидкий перехід до початкового етапу.

Третій етап полягає у відображенні результатів, які залежать від вибраного типу виводу. В межах цього етапу формується новий візуальний блок (фрейм), який демонструє відповідну інформацію у зручному для користувача форматі. Незалежно від змісту, усі результати подаються у зрозумілій формі, доповненій можливістю повернутися до попереднього

вікна вибору, що забезпечує безперервність роботи з системою в рамках однієї сесії. Додатково реалізовано функцію сортування, яка дозволяє зручно переглядати рейтинги та швидко знаходити необхідні дані. Це підвищує ефективність аналізу отриманих результатів та сприяє прийняттю обґрунтованих рішень.

| Назва проекту                   | Рейтинг проекту    | Бажаність проекту             |
|---------------------------------|--------------------|-------------------------------|
| Збирання ІЧ-енергії             | 0.5544723264329096 | Проект з низьким пріоритетом  |
| Оптимізація сорго як сировини   | 0.3735675616366001 | Небажаний проект              |
| Контакти для сонячних елементів | 0.8499534308313909 | Проект з високим пріоритетом  |
| Вплив пожеж на вуглецевий цикл  | 0.3675953710065621 | Небажаний проект              |
| Збирання даних дист. зондування | 0.5520804891750446 | Проект з низьким пріоритетом  |
| Розміщення віднов. енергії      | 0.9325833658940142 | Проект з високим пріоритетом  |
| Влас. високоентропійного сплаву | 0.7135547521302785 | Проект з середнім пріоритетом |

Зберегти результати      Відсортувати      Повернутися

Рис. 4.11. Екран з прикладом результатів

Для збереження отриманої інформації користувач може скористатися кнопкою «Зберегти результати». При цьому йому пропонується ввести назву файлу та вибрати його формат – за замовчуванням використовується текстовий файл (\*.txt), однак система підтримує й інші формати, зокрема для електронних таблиць. У разі експорту до табличного файлу дані структуруються по клітинках, що підвищує зручність подальшої обробки.

\*test.txt – Блокнот

Файл Правка Формат Вид Справка

Узагальнена бажаність проектів:

|                                 |                    |                               |
|---------------------------------|--------------------|-------------------------------|
| Збирання ІЧ-енергії             | 0.5544723264329096 | Проект з низьким пріоритетом  |
| Оптимізація сорго як сировини   | 0.3735675616366001 | Небажаний проект              |
| Контакти для сонячних елементів | 0.8499534308313909 | Проект з високим пріоритетом  |
| Вплив пожеж на вуглецевий цикл  | 0.3675953710065621 | Небажаний проект              |
| Збирання даних дист. зондування | 0.5520804891750446 | Проект з низьким пріоритетом  |
| Розміщення віднов. енергії      | 0.9325833658940142 | Проект з високим пріоритетом  |
| Влас. високоентропійного сплаву | 0.7135547521302785 | Проект з середнім пріоритетом |

Рис. 4.12. Приклад збереженого результату

Таким чином, розроблений інтерфейс об'єднує простоту, функціональність та логічну послідовність, що дозволяє ефективно використовувати систему на кожному з її етапів.

#### 4.6. Особливості проведеного тестування

Один з найважливіших етапів у розробці програмного забезпечення – це перевірка його працездатності. У межах реалізації цієї системи для забезпечення коректності її функціонування було застосовано методику тестування за принципом «чорної скриньки». Основна увага приділялася генерації тестових даних випадковим чином і аналізу граничних значень, що дозволяє виявити помилки, які можуть проявитися в найбільш критичних умовах експлуатації системи.

Для реалізації стохастичного підходу не використовувалися зовнішні файли, натомість була створена функціональність, що забезпечує автоматичне заповнення необхідних масивів випадковими числами за допомогою вбудованого генератора. Це дозволило в короткий термін протестувати різноманітні варіанти поведінки системи. Результати показали, що під час тестування на випадкових даних жодних критичних помилок виявлено не було, що свідчить про базову стійкість системи до різного роду вхідних даних.

Було вручну сформовано тестову вибірку, що включала мінімальні й максимальні допустимі значення, нульові значення, неправильні або некоректно введені дані, а також значення, що можуть викликати логічні суперечності у роботі програми.

|   | A          | B                                        | C           | D           | E        | F                                                  | G                              | H                        | I                                             | J  |
|---|------------|------------------------------------------|-------------|-------------|----------|----------------------------------------------------|--------------------------------|--------------------------|-----------------------------------------------|----|
|   | № критерію | Назва критерію                           | значення    | min         | max      | Тип функції:<br>одностороння(1),<br>двостороння(2) | Оптимум (для<br>двосторонньої) | Коефіцієнт<br>важливості | Коефіцієнт<br>кривизни (для<br>двосторонньої) |    |
| 1 |            |                                          |             |             |          |                                                    |                                |                          |                                               |    |
| 2 | 1          | Кошторис проекту, грн                    | 0           | 0           | 0        | 1                                                  |                                | 0                        |                                               |    |
| 3 | 2          | Строк виконання, дні                     | 80          | 120         | 80       | 1                                                  |                                | -5                       |                                               |    |
| 4 | 3          | Новизна                                  | високий     | низький     | високий  | інше                                               |                                | 9                        |                                               |    |
| 5 | 4          | Значимість                               | ідеальний   | мінімальний | середній | 1                                                  |                                | 4                        |                                               |    |
| 6 | 5          | Об'єктивність                            | високий     | високий     | високий  | 1                                                  |                                |                          |                                               |    |
| 7 | 6          | Рівень кваліфікації виконавців           | високий     | низький     | високий  | 1                                                  |                                | 3                        |                                               |    |
| 8 | 7          | Ресурси на теоретичну роботу, %          | 11          | 10          | 10       | 2                                                  | 16                             |                          | 4                                             | -7 |
| 9 | 8          | Залученість учасників освітнього процесу | мінімальний | низький     | низький  | 2                                                  | невизначено                    | 3                        | 9                                             |    |

Рис. 4.13. Приклад вхідних даних для тестування

Під час цього тестування система продемонструвала здатність обробляти виняткові ситуації без аварійних завершень. Помилки, викликані некоректним введенням, були перехоплені, а користувачу відображалося повідомлення з поясненням і відповідна заміна неправильного значення на коректне.

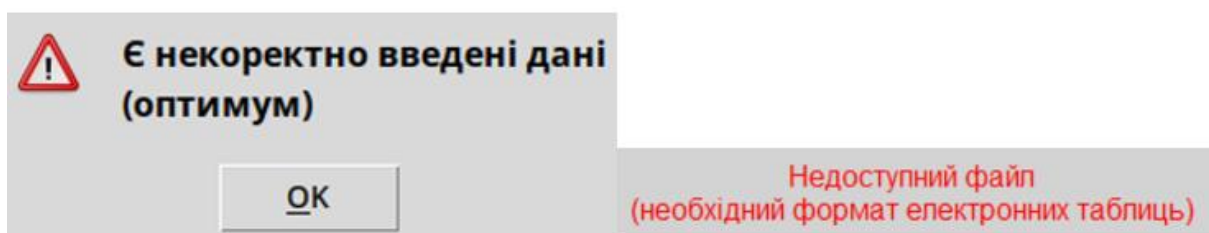


Рис. 4.14. Повідомлення про некоректні вхідні дані

Після завершення перевірки шляхом випадкового тестування та тестування граничних ситуацій було підготовлено повноцінний тестовий приклад. Він відображає можливість застосування системи до реального практичного завдання – наприклад, аналізу проєктів, поданих на конкурсний відбір. Умова передбачає оцінювання семи ініціатив, серед яких необхідно виявити ті, що мають найбільшу перспективу отримати фінансування. Експерти формують набір критеріїв, що використовуються для оцінки, а також встановлюють коефіцієнти вагомості кожного з них і межі бажаних значень. У випадках, коли присутнє двостороннє обмеження, допускається зазначення оптимуму, або він розраховується автоматично як середнє між мінімумом і максимумом.

На основі цих параметрів формується шаблон оцінювання, який адаптується під конкретні обставини. У таблиці заповнюються всі поля, окрім стовпця «значення», який заповнюється після завершення експертної оцінки.

| Назва проекту                   | Рейтинг проекту    | Важкість проекту              |
|---------------------------------|--------------------|-------------------------------|
| Розміщення віднов. енергії      | 0.9325833658940142 | Проект з високим пріоритетом  |
| Контакти для сонячних елементів | 0.8499534308313909 | Проект з високим пріоритетом  |
| Влас. високоентропійного сплаву | 0.7135547521302785 | Проект з середнім пріоритетом |
| Збирання ІЧ-енергії             | 0.5544723264329096 | Проект з низьким пріоритетом  |
| Збирання даних дист. зондування | 0.5520804891750446 | Проект з низьким пріоритетом  |
| Оптимізація сорго як сировини   | 0.3735675616366001 | Небажаний проект              |
| Вплив пожеж на вуглецевий цикл  | 0.3675953710065621 | Небажаний проект              |

Зберегти результати      Відсортувати      Повернутися

Рис. 4.15. Рейтинг проєктів за узагальненою оцінкою

Для кращої візуалізації результатів система надає кілька варіантів виведення інформації. Перший – загальний рейтинг проєктів. Другий – рейтинг за окремими критеріями, що особливо корисно в разі потреби підтримати не лише проєкти з найвищою загальною оцінкою, а й ті, що мають переваги за певними характеристиками (наприклад, рівень новизни або об’єктивності).

☐ Кошторис проекту, грн  
☐ Строк виконання, дні  
☒ Новизна  
☒ Значимість  
☒ Об’єктивність  
☐ Рівень кваліфікації виконавців  
☐ Ресурси на теоретичну роботу, %  
☐ Залученість учасників освітнього процесу

Розрахунок за обраними критеріями

Рис. 4.16. Вибір критеріїв для оцінювання

| Назва проекту                   | Рейтинг проекту     | Бажаність проекту             |
|---------------------------------|---------------------|-------------------------------|
| Розміщення віднов. енергії      | 0.9241612729779077  | Проект з високим пріоритетом  |
| Контакти для сонячних елементів | 0.797399856691261   | Проект з середнім пріоритетом |
| Влас. високоентропійного сплаву | 0.629394639258557   | Проект з низьким пріоритетом  |
| Збирання даних дист. зондування | 0.398282620594642   | Небажаний проект              |
| Збирання ІЧ-енергії             | 0.37115147731569037 | Небажаний проект              |
| Оптимізація сорго як сировини   | 0.23486565312504962 | Небажаний проект              |
| Вплив пожеж на вуглецевий цикл  | 0.23486565312504962 | Небажаний проект              |

Зберегти результати      Відсортувати      Повернутися

Рис. 4.17. Рейтинг проєктів за критеріями

Також доступна функція аналізу конкретного проєкту. Користувач може обрати будь-яку ініціативу та переглянути її оцінку за всіма критеріями, виявити, за рахунок яких факторів сформувався загальний бал. Наприклад, проєкт «Розміщування відновлювальної енергії» отримав високий рейтинг завдяки високій кваліфікації виконавців і залученості учасників, хоча його слабкими сторонами стали строки реалізації та відносно низька значимість.

Збирання ІЧ-енергії ▼

Розрахунок за обраним проєктом

Рис. 4.18. Приклад обраного проєкту для заповнення всіх показників

| Оцінка окремого проекту                       |                     |
|-----------------------------------------------|---------------------|
| Проект "Збирання ІЧ-енергії"                  |                     |
| Назва критерію                                | Рейтинг по критерію |
| Кошторис проекту, грн                         | 0.816076652651101   |
| Строк виконання, дні                          | 0.9879981898974657  |
| Новизна                                       | 0.1001610497561715  |
| Значимість                                    | 0.8000107130043536  |
| Об'єктивність                                 | 0.6380561665820186  |
| Рівень кваліфікації виконавців                | 0.9932847020678415  |
| Ресурси на теоретичну роботу, %               | 0.9769462779851399  |
| Залученість учасників освітнього процесу      | 1                   |
| Загальний рейтинг проекту: 0.5544723264329096 |                     |
| Зберегти результати                           | Повернутися         |

Рис. 4.19. Результати оцінювання окремого проекту

Задля зручного сприйняття даних передбачено побудову графіків. Наприклад, стовпчикова діаграма дозволяє наочно порівняти загальні оцінки всіх кандидатів.

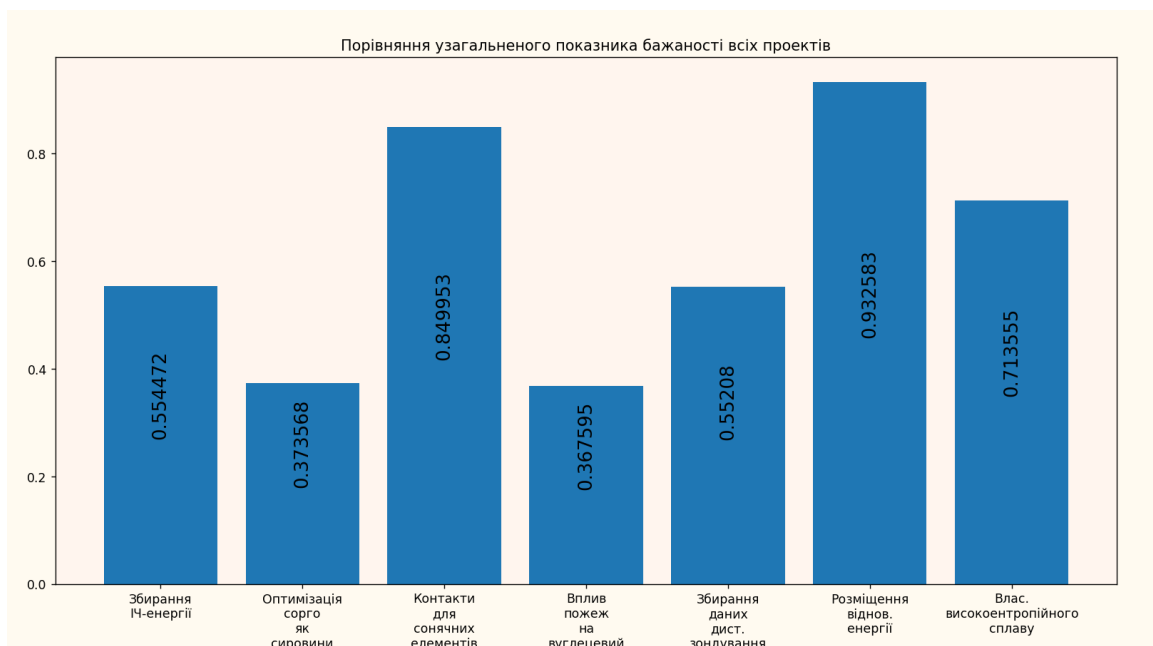


Рис. 4.20. Приклад діаграми зіставлення інтегральних оцінок

Видно, що найвищу оцінку отримав проект з відновлюваної енергетики, а найменшу – ініціатива, пов'язана із збиранням даних дистанційним зондуванням.



Додатковим корисним засобом аналізу виступає візуалізація впливу окремих критеріїв на загальний підсумок. Це допомагає швидко оцінити, які саме характеристики найбільше вплинули на остаточний бал.

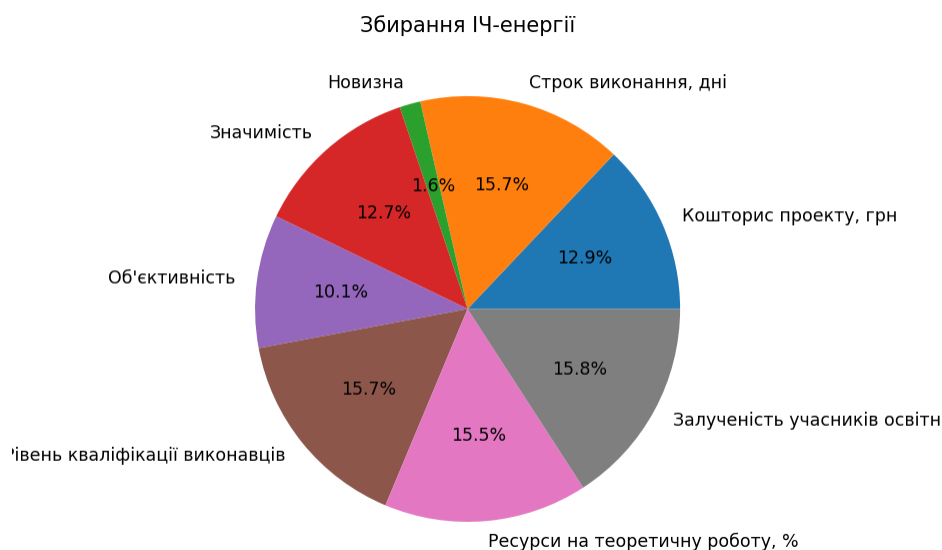


Рис. 4.21. Вплив кожного з факторів на результат окремого проекту

З аналізу видно, що проєкт «Розміщення відновлюваної енергії» отримав високий бал через високий рівень об'єктивності, збалансований бюджет й високу компетентність команди. Водночас, недоліками стали низька новизна і тривалі строки реалізації.

У підсумку, проведене тестування підтвердило надійність і практичну придатність системи. Комісія експертів змогла обрати найбільш ефективний проєкт, оцінити сильні й слабкі сторони кожної ініціативи, а також прослідкувати, які саме критерії найбільше вплинули на результати.

#### 4.7. Висновки до четвертого розділу

У ході проведення тестування розробленої системи було забезпечено всебічну перевірку її функціональності, надійності та стійкості до некоректних або граничних вхідних даних. Застосування методу чорної скриньки дало змогу зосередитися саме на результатах роботи системи при

різноманітних умовах, без заглиблення в її внутрішню структуру. Генерація випадкових вхідних даних, а також аналіз крайових значень, дозволили перевірити коректність обробки як типових сценаріїв, так і потенційно проблемних випадків.

Особливу увагу приділено роботі з неправильно введеними користувачем даними – система не лише виявляє такі помилки, але й пропонує зручні механізми для їх обробки та інформування користувача про необхідність виправлення. Важливо відзначити, що виявлені в процесі тестування проблеми стосувалися переважно обробки нештатних ситуацій, і всі вони були успішно вирішені в процесі доопрацювання програмного забезпечення.

Завдяки результатам тестування вдалося підтвердити, що система не лише технічно коректна, але й зручна для кінцевих користувачів, здатна підтримати процес аналізу, вибору та обґрунтування рішень на основі багатокритеріального оцінювання. Це свідчить про її готовність до реального застосування та подальшого впровадження.

## ВИСНОВКИ

Метою даного дипломного проєкту було створення програмного засобу для підтримки процесу оцінки та вибору науково-технічних проєктів шляхом реалізації математичної моделі на основі узагальненої функції бажаності Харрінгтона. У результаті виконання було досягнуто повну реалізацію поставлених завдань: проведено аналіз предметної області, обґрунтовано вибір методу багатокритеріальної оцінки, побудовано математичну модель і реалізовано програмне забезпечення, що відповідає функціональним та нефункціональним вимогам.

У процесі дослідження було виявлено, що наявні інструменти не забезпечують достатнього рівня гнучкості та зручності для комплексної оцінки наукових і науково-технічних проєктів за багатьма критеріями. Запропонована система, на відміну від аналогів, дозволяє враховувати як односторонні, так і двосторонні обмеження для окремих показників, а також вагомість кожного критерію в загальній оцінці. Це робить її адаптивною до реальних умов конкурсного відбору і здатною забезпечити об'єктивність та прозорість при прийнятті рішень.

У рамках реалізації системи створено зручний графічний інтерфейс, що дозволяє користувачеві без глибоких технічних знань вводити необхідні дані, запускати процес оцінювання та отримувати результат у доступній формі. Проведене тестування підтвердило коректність роботи системи, її стійкість до помилкових даних та здатність обробляти типові і граничні сценарії. Практична апробація на умовному прикладі підтвердила доцільність і ефективність використання розробленого програмного продукту.

Таким чином, результати дипломного проєкту демонструють, що поставлена мета була досягнута в повному обсязі. Розроблена система є актуальним рішенням для автоматизації процесів оцінки наукових проєктів.

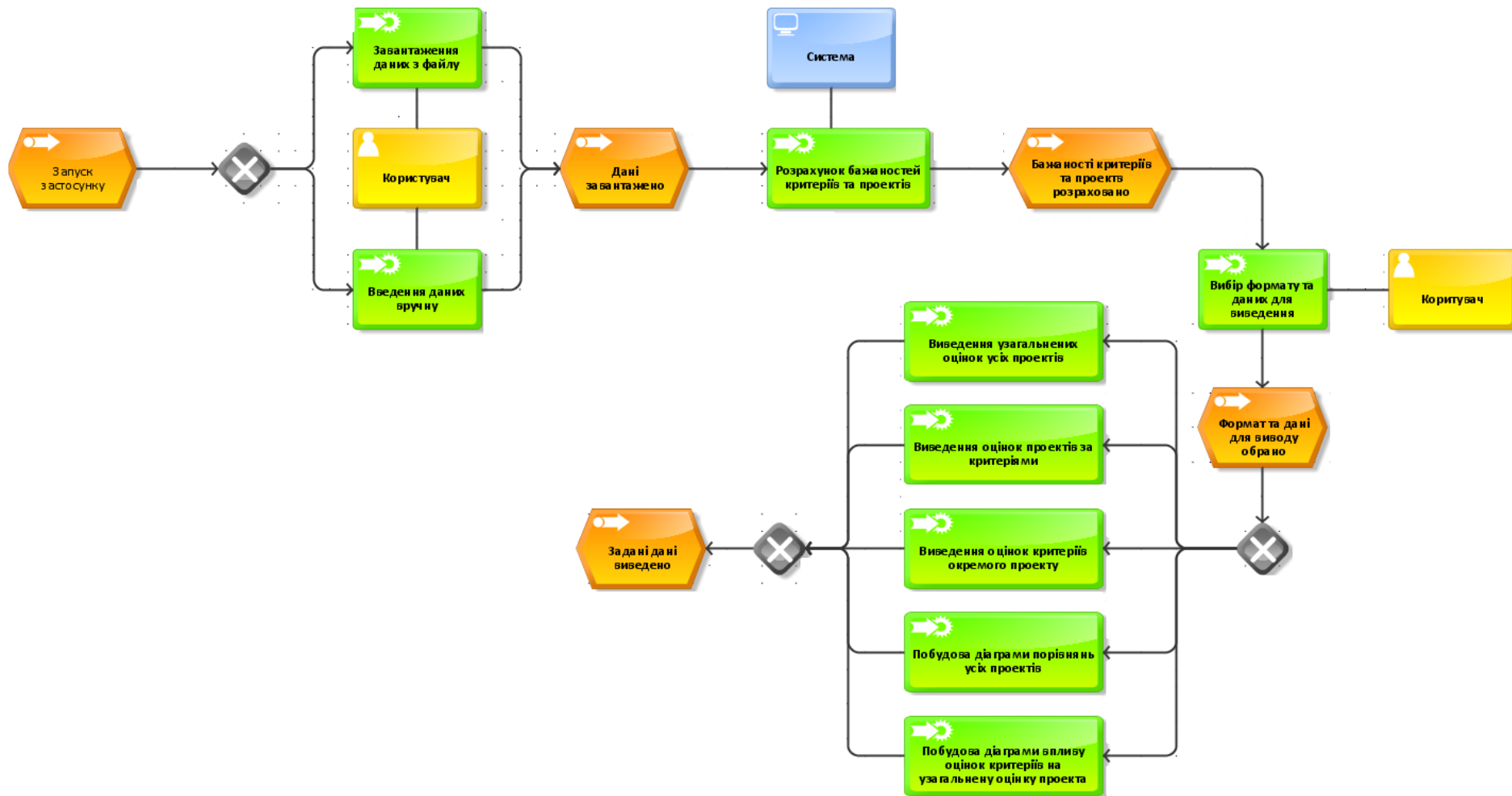
## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Проєкт закону України. Про внесення змін до Закону України «Про наукову і науково-технічну діяльність» [Електронний ресурс]. – Режим доступу: <https://www.uzhnu.edu.ua/en/infocentre/get/5863>. – Дата доступу: 12.05.2025
2. Самохвалов Ю. Я., Бурба О. І. Оцінка ефективності наукових та науково-технічних проєктів на основі узагальненої функції Харрінгтона // Системи управління, навігації та зв'язку. – 2018. – Т. 4, № 50.
3. Затверджено 54 проєкти робіт молодих вчених на 2021 рік [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/news/zatverdzheno-54-proekti-robit-molodikh-vchenikh-na-2021-rik>. – Дата доступу: 12.05.2025 р.
4. Наступного року МОН профінансує майже 200 проєктів молодих вчених на 100 млн грн – за 4 роки бюджет конкурсу збільшили в 8 разів [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/news/nastupnogo-roku-mon-profinansue-mayzhe-200-proektiv-molodikh-vchenikh-na-100-mln-grn-za-4-roki-byudzheth-konkursu-zbilshili-v-8-raziv>. – Дата доступу: 12.05.2025 р.
5. Методи багатокритеріальної оцінки. Онлайн-ресурс «Підручники для студентів онлайн» [Електронний ресурс]. – Режим доступу: [https://stud.com.ua/31899/menedzhment/metodi\\_bagatokriterialnoyi\\_otsinki](https://stud.com.ua/31899/menedzhment/metodi_bagatokriterialnoyi_otsinki). – Дата доступу: 12.05.2025 р.
6. Метод ідеальної точки. Файловий архів студентів StudFiles [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/2266953/>. – Дата доступу: 12.05.2025 р.
7. Srinivasan V., Shocker A. D. Linear programming techniques for multidimensional analysis of preference // Psychometrica. – 1973. – Vol. 38. – P. 337–342.
8. Harrington E. C. A Utility-Based Approach to Quality Assessment: The Desirability Function // Industrial Quality Control. – 1965. – Vol. 21(4). – P. 494–498.
9. Шутяк Ю. В. Використання функції бажаності для оцінки економічної

- безпеки підприємства // Наукові студії. – Вип. 7. – С. 147–154.
10. Навчальні матеріали онлайн [Електронний ресурс]. – Режим доступу: [https://pidru4niki.com/15660721/menedzhment/metod\\_analizu\\_iyerarhiy\\_instrument\\_dlya\\_priynyattya\\_rishen\\_pri\\_strategichnomu\\_planuvanni](https://pidru4niki.com/15660721/menedzhment/metod_analizu_iyerarhiy_instrument_dlya_priynyattya_rishen_pri_strategichnomu_planuvanni). – Дата доступу: 12.05.2025 р.
  11. Офіційний сайт Python [Електронний ресурс]. – Режим доступу: <https://www.python.org/doc/essays/blurb/>. – Дата доступу: 12.05.2025 р.
  12. Сайт Linode. The Pros and Cons of Python Programming [Електронний ресурс]. – Режим доступу: <https://www.linode.com/docs/guides/pros-and-cons-of-python/>. – Дата доступу: 12.05.2025 р.
  13. Нечипорук О. В. Оцінювання інвестиційної привабливості підприємств з використанням рейтингових підходів // Вісник технічного університету. – 2007. – № 75. – С. 275–284.
  14. Lutz M. Learning Python. – 5th ed. – O'Reilly Media, 2013. – 1648 р.

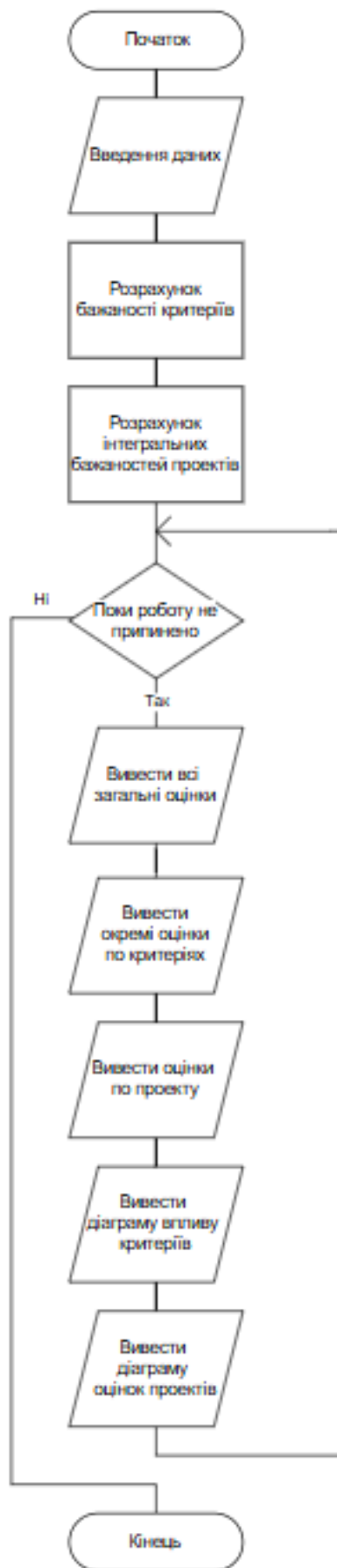
## **ДОДАТКИ**

**Додаток 1**  
**Копії графічних матеріалів**

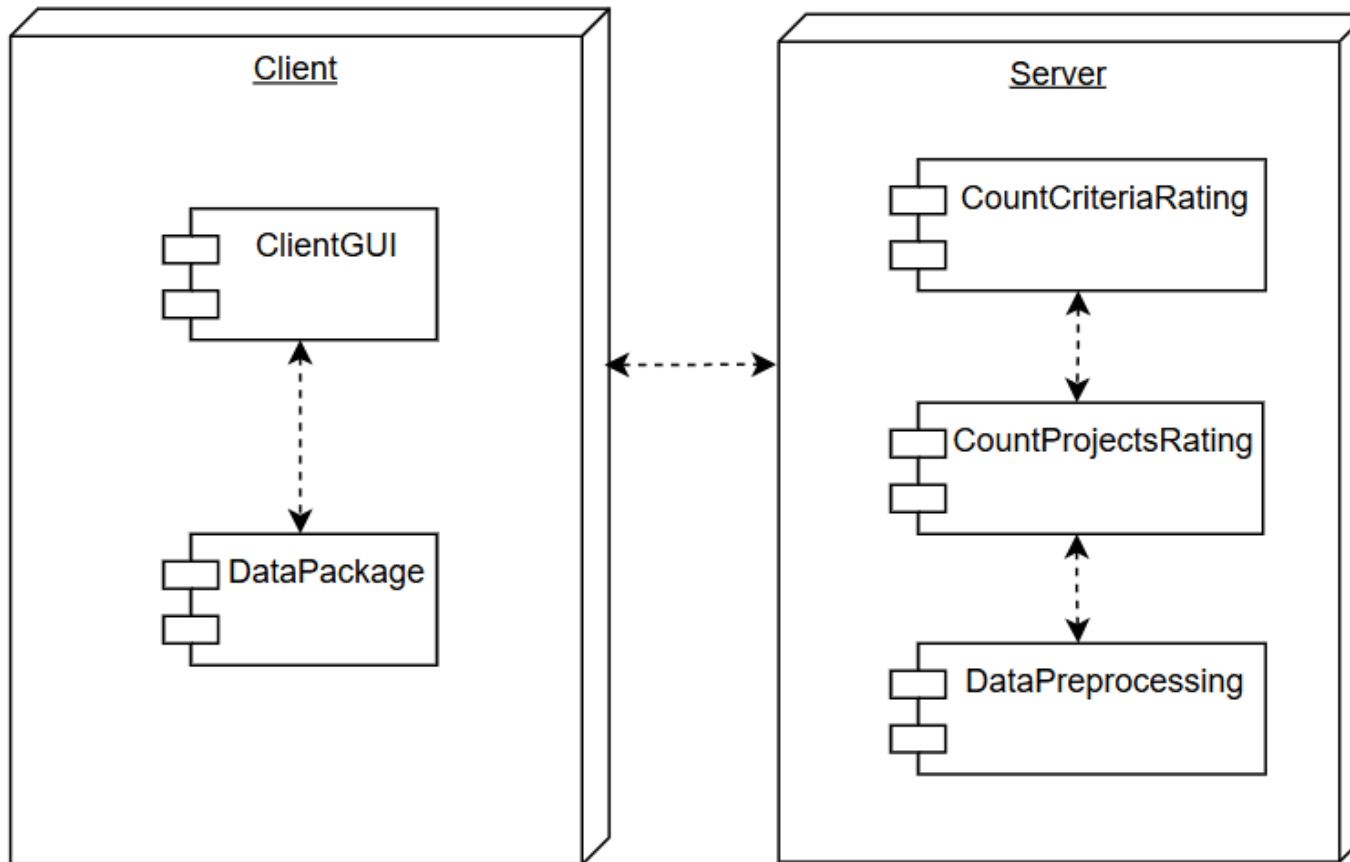


ДП.045420-06-99. Програмне забезпечення системи оцінки наукових та науково-технічних проектів. Схема функціонування системи. EDP діаграма

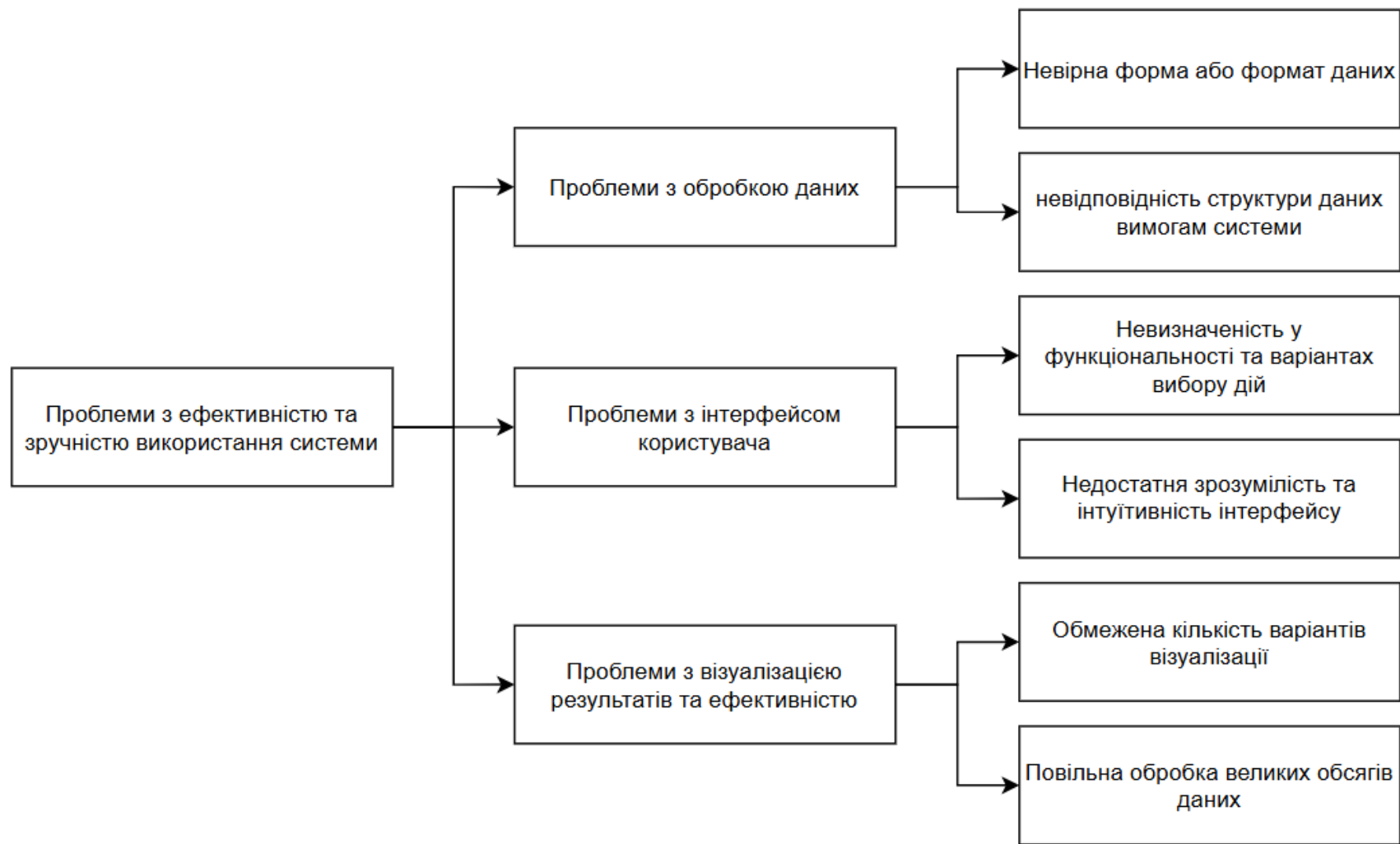




ДП.045420-07-99. Програмне забезпечення системи оцінки наукових та науково-технічних проєктів. Діаграма основних процесів. Блок схема алгоритму



Структура основних компонентів системи  
Регеша О. М., група КП-11



**Додаток 2**  
**Лістинг програми**

```

import tkinter as tk
from tkinter import ttk
from tkinter import filedialog
from tkinter import messagebox
from openpyxl import load_workbook
from math import exp, log
import matplotlib
import matplotlib.pyplot as plt

matplotlib.use('Qt5Agg')

root = tk.Tk()
root.title('Система оцінки та вибору
          науково-технічних проєктів')
root['bg'] = 'light grey'
root.geometry('1500x700')

root.resizable(height=True,
               width=True)

main = tk.Frame(root, bg='white')
main.place(x=0, y=0)

canvas_output_selection =
    tk.Canvas()
canvas_output_selection_buildup =
    tk.Canvas()

verbal_dict = {
    'незадовільний': 0,
    'мінімальний': 1,
    'низький': 2,
    'середній': 3,
    'високий': 4,
    'ідеальний': 7
}

def f_harrington_unilateral(x_low,
                             x_up, x, minimization=False):
    try:
        x_low = float(x_low)
        x_up = float(x_up)
        x = float(x)
    except:
        x_low =
        verbal_dict[x_low.lower()]
        x_up =
        verbal_dict[x_up.lower()]
        x = verbal_dict[x.lower()]
    finally:
        if x_low == x_up:
            if x == x_low:
                d = 0.55
            elif minimization:
                if x < x_low:
                    d = 1
            else:

```

```

        d = 0.1
    else:
        if x < x_low:
            d = 0.1
        else:
            d = 1
    else:
        y = -2 + 7 * (x -
x_low) / (x_up - x_low)
        d = exp(-exp(-y))
    return d

```

```

def f_harrington_bilateral(x_low,
x_up, x, optimum, n1, n2):
    try:
        x_low = float(x_low)
        x_up = float(x_up)
        x = float(x)
    except:
        x_low =
verbal_dict[x_low.lower()]
        x_up =
verbal_dict[x_up.lower()]
        x = verbal_dict[x.lower()]

    finally:
        try:
            optimum =
float(optimum)
        except:
            if optimum is None:
                optimum = (x_low +
x_up) / 2
            optimum =
float(optimum)
            elif optimum in
verbal_dict:
                optimum =
verbal_dict[optimum.lower()]
            else:
                optimum = (x_low +
x_up) / 2

    messagebox.showwarning('Попередж
ення', 'Є некоректно введені
дані\n(оптимум) ')

    if n1 and n2:
        try:
            n1 = float(n1)
            n2 = float(n2)
            n1 = log(log(1 /
n1)) / log(abs((2 * n2 - (x_up
+ x_low)) / (x_up - x_low)))
            n2 = n1
        except:
            n1 = n2 = 3
    else:

```

```

        n1 = n2 = 3

    if x_low == x_up:
        if x < x_low:
            d = 0.1
        elif x == x_low:
            d = 1
        else:
            d = 0.1
    else:
        if x < optimum:
            y = (x - optimum) /
(optimum - x_low)
            d = exp(-(abs(y)))
        ** n1)
        elif x > optimum:
            y = (x - optimum) /
(x_up - optimum)
            d = exp(-(abs(y)))
        ** n2)
        else:
            d = 1
    return d

```

```

def data_processing(filepath):
    wb = load_workbook(filepath)
    criteria_rating = []
    object_rating = []
    verbal_scale = []
    sheet_names = wb.sheetnames

    for i in sheet_names:
        sheet = wb[i]

        values_x_low = []
        values_x_up = []
        values_x = []
        values_function_type = []
        values_optimum = []
        values_importance = []
        n1 = []
        n2 = []

        j = 2
        while sheet[f'C{j}'].value
or sheet[f'D{j}'].value or
sheet[f'E{j}'].value or
sheet[f'F{j}'].value:

            values_x.append(sheet[f'C{j}'].
value)

            values_x_low.append(sheet[f'D{j}
'].value)

            values_x_up.append(sheet[f'E{j}
'].value)

            try:

```

```

values_function_type.append(int
(sheet[f'F{j}'].value))
    except:

values_function_type.append(1)

messagebox.showwarning('Попередж
ення', 'Є некоректно введені
дані\n(тип функції)')

values_optimum.append(sheet[f'G
{j}'].value)
    if
sheet[f'H{j}'].value:
        if
float(sheet[f'H{j}'].value) <
0:

values_importance.append(0)
    else:

values_importance.append(float(
sheet[f'H{j}'].value))
    else:

values_importance.append(1)

n1.append(sheet[f'I{j}'].value)

n2.append(sheet[f'J{j}'].value)

    j += 1

    d_array = []
    for j in
range(len(values_x)):
        if
values_function_type[j] == 1:

d_array.append(f_harrington_uni
lateral(values_x_low[j],
values_x_up[j], values_x[j]))
        elif
values_function_type[j] == -1:

d_array.append(f_harrington_uni
lateral(values_x_up[j],
values_x_low[j], values_x[j],
True))
        elif
values_function_type[j] == 2:

d_array.append(f_harrington_bil
ateral(values_x_low[j],
values_x_up[j], values_x[j],
values_optimum[j],

```



```

n1[j], n2[j]))

criteria_rating.append(d_array)

# Нормалізація коефіцієнтів
важливості
temp = 0
for k in values_importance:
    temp += k
temp =
len(values_importance) / temp
for j in
range(len(values_importance)):
    values_importance[j] =
values_importance[j] * temp

# Розрахунок узагальненого
показника бажаності
D = 1
for j in
range(len(d_array)):
    D = D * d_array[j] **
values_importance[j]
D = D ** (1 /
len(values_importance))
object_rating.append(D)

if D >= 0.8:

verbal_scale.append('Проект з
високим пріоритетом')
elif D >= 0.65:

verbal_scale.append('Проект з
середнім пріоритетом')
elif D >= 0.5:

verbal_scale.append('Проект з
низьким пріоритетом')
else:

verbal_scale.append('Небажаний
проект')

criteria_names = []
sheet = wb[sheet_names[0]]
j = 2
while sheet[f'B{j}'].value:

criteria_names.append(sheet[f'B
{j}'].value)
j += 1

return sheet_names,
criteria_rating, object_rating,
verbal_scale, criteria_names,
values_importance

```

```

def write_in_file(data, head=None):
    new_file =
    filedialog.asksaveasfilename(de
    faultextension='.txt')
    f = open(new_file, 'w')
    if head:
        f.write(f'{head}:\n\n')
    for i in data:
        for j in i:
            f.write(f'{j}\t')
        f.write('\n')

def summarized_rating(sheet_names,
    object_rating, verbal_scale,
    by_criteria=False):
    global
    canvas_output_selection_buildup
    try:

    canvas_output_selection_buildup
    .destroy()
    finally:
        pass

    canvas_result = tk.Canvas(root,
    width=1500, height=900,
    bg='light grey')
    canvas_result.place(x=0, y=0)

    if by_criteria:
        label_head =
        tk.Label(canvas_result,
        text='Рейтинг проєктів за
        критеріями', font=('helv36', 16),

    bg='light gray')
        label_head.place(x=750,
        y=100, anchor='center')
    else:
        label_head =
        tk.Label(canvas_result,
        text='Рейтинг проєктів за
        узагальненою оцінкою',
        font=('helv36', 16),

    bg='light gray')
        label_head.place(x=750,
        y=100, anchor='center')

    def build_tree(sort=False):
        style = ttk.Style()
        style.theme_use('clam')

    style.configure('Treeview.Headi
    ng', background='green3')

```

```

style.configure('style.Treeview', font=('Arial', 14),
rowheight=30)
    result_table =
ttk.Treeview(canvas_result,
height=15,
style='style.Treeview')
    result_table['columns'] =
('sheet_names',
'object_rating',
'verbal_scale')
    result_table.column('#0',
width=0, stretch=False)

result_table.column('sheet_names', anchor='center', width=400)

result_table.column('object_rating', anchor='center',
width=400)

result_table.column('verbal_scale', anchor='center',
width=400)

    result_table.heading('#0',
text='', anchor='center')

result_table.heading('sheet_names', text='Назва проекту',
anchor='center')

result_table.heading('object_rating', text='Рейтинг проекту',
anchor='center')

result_table.heading('verbal_scale', text='Бажаність проекту',
anchor='center')

    formatted_data = []
    for i in
range(len(sheet_names)):

formatted_data.append([sheet_names[i], object_rating[i],
verbal_scale[i]])

    if sort:

formatted_data.sort(reverse=True, key=lambda a: a[1])
    for i in
formatted_data:

result_table.insert('', 'end',
values=i)
    else:
    for i in

```

```

formatted_data:

result_table.insert('', 'end',
values=i)

        result_table.place(x=750,
y=300, anchor='center')
        return formatted_data

result_data = build_tree()

btn_sort_data =
tk.Button(canvas_result,
text='Зберегти результати',
font='helv36', width=20,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',

activeforeground='white',
relief='ridge')
btn_sort_data.place(x=300,
y=650, anchor='se')
btn_sort_data.bind('<Button-
1>', lambda event:
write_in_file(result_data,
'Узагальнена бажаність проєктів'))

def sort_tree():
    sub_result_data =
build_tree(True)
    sub_btn_sort_data =
tk.Button(canvas_result,
text='Зберегти результати',
font='helv36', width=20,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',

activeforeground='white',
relief='ridge')

sub_btn_sort_data.place(x=300,
y=650, anchor='se')

sub_btn_sort_data.bind('<Button
-1>', lambda event:
write_in_file(sub_result_data))

btn_sort_data =
tk.Button(canvas_result,
text='Відсортувати',
font='helv36', width=15,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',

```

```

activeforeground='white',
relief='ridge')
btn_sort_data.place(x=1000,
y=650, anchor='se')
btn_sort_data.bind('<Button-
1>', lambda event: sort_tree())

btn_return_to_output_selection
= tk.Button(canvas_result,
text='Повернутися',
font='helv36', width=15,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')

btn_return_to_output_selection.
place(x=1250, y=650,
anchor='se')

btn_return_to_output_selection.
bind('<Button-1>', lambda
event: canvas_result.destroy())

def rating_by_criteria(sheet_names,
criteria_rating,
chosen_criteria,
values_importance):
new_rating_by_criteria = []
new_verbal_scale_by_criteria =
[]
for i in criteria_rating:
D = 1
total_importance = 0
for m in range(len(i)):
if
chosen_criteria[m].get():
D = D * i[m] **
values_importance[m]
total_importance +=
values_importance[m]
D = D ** (1 /
total_importance)

new_rating_by_criteria.append(D
)

if D >= 0.8:

new_verbal_scale_by_criteria.ap
pend('Проект з високим
пріоритетом')
elif D >= 0.65:

new_verbal_scale_by_criteria.ap

```

```

pend('Проект з середнім
пріоритетом')
    elif D >= 0.5:

new_verbal_scale_by_criteria.ap
pend('Проект з низьким
пріоритетом')
    else:

new_verbal_scale_by_criteria.ap
pend('Небажаний проект')

summarized_rating(sheet_names,
new_rating_by_criteria,
new_verbal_scale_by_criteria,
True)

def single_rating(chosen_object,
criteria_names,
criteria_rating,
object_rating):
canvas_result = tk.Canvas(root,
width=1500, height=900,
bg='light grey')
canvas_result.place(x=0, y=0)

result_data = []

label_head =
tk.Label(canvas_result,
text='Оцінка окремого проекту',
font=('helv36', 16),
bg='light
gray')
label_head.place(x=750, y=100,
anchor='center')

label_head2 =
tk.Label(canvas_result,
text=f'Проект
\"{chosen_object}\"',
font=('helv36', 16),
bg='light
gray')
label_head2.place(x=750, y=150,
anchor='center')

style = ttk.Style()
style.theme_use('clam')

style.configure('Treeview.Headi
ng', background='green3')

style.configure('style.Treeview
', font=('Arial', 14),
rowheight=30)

result_table =

```

```

ttk.Treeview(canvas_result,
height=10,
style='style.Treeview')
result_table['columns'] =
('criteria_names',
'criteria_rating')
result_table.column('#0',
width=0, stretch=False)

result_table.column('criteria_n
ames', anchor='center',
width=500)

result_table.column('criteria_r
ating', anchor='center',
width=500)

result_table.heading('#0',
text='', anchor='center')

result_table.heading('criteria_
names', text='Назва критерію',
anchor='center')

result_table.heading('criteria_
rating', text='Рейтинг по
критерію', anchor='center')

for i in
range(len(criteria_names)):
    result_table.insert('',
'end',
values=[criteria_names[i],
criteria_rating[i]])

result_data.append([criteria_na
mes[i], criteria_rating[i]])

result_table.place(x=750,
y=400, anchor='center')

label_result =
tk.Label(canvas_result,
text=f'Загальний рейтинг проекту:
{object_rating}',
font=('helv36', 16),
bg='light
gray')
label_result.place(x=750,
y=600, anchor='center')

btn_sort_data =
tk.Button(canvas_result,
text='Зберегти результати',
font='helv36', width=20,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',

```

```

activeforeground='white',
relief='ridge')
btn_sort_data.place(x=300,
y=650, anchor='se')
btn_sort_data.bind('<Button-
1>', lambda event:
write_in_file(result_data,
chosen_object))

btn_return_to_output_selection
= tk.Button(canvas_result,
text='Повернутися',
font='helv36', width=15,
height=2,

bg='#a8e6cf', fg='black',
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')

btn_return_to_output_selection.
place(x=1450, y=650,
anchor='se')

btn_return_to_output_selection.
bind('<Button-1>', lambda
event: canvas_result.destroy())

```

```

def
summarized_rating_diagram(object
t_rating, sheet_names):
x = [i.replace(' ', '\n') for i
in sheet_names]
y = object_rating

fig, ax = plt.subplots()

ax.bar(x, y)

ax.set_facecolor('seashell')
ax.set_title('Порівняння
узагальненого показника бажаності всіх
проектів')

fig.set_facecolor('floralwhite'
)

for i in range(len(x)):
    plt.text(i, y[i] / 2,
round(y[i], 6), ha='center',
        fontsize=15,
rotation=90)

fig.set_figwidth(18)
fig.set_figheight(9)

plt.show()

```



```

plt.close()

def
    influence_by_criteria(chosen_ob
    ject, criteria_names,
    criteria_rating):
    y = criteria_rating

    plt.title(chosen_object)
    plt.pie(y,
    labels=criteria_names,
    autopct='%1.1f%%', )
    plt.show()

def start_window():
    def
        output_selection(sheet_names,
        criteria_rating, object_rating,
        verbal_scale, criteria_names,
        values_importance):
            canvas_start.destroy()
            global
            canvas_output_selection
            canvas_output_selection =
            tk.Canvas(root, width=1500,
            height=900, bg='light gray')

            canvas_output_selection.place(x
            =0, y=0)

            label_output_selection =
            tk.Label(canvas_output_selectio
            n, text='Оберіть що необхідно
            визначити:',

            font='helv36', bg='light gray')

            label_output_selection.place(x=
            500, y=100, anchor='center')

            btn_summarized_rating =
            tk.Button(canvas_output_selecti
            on, text='Рейтинг проектів за
            узагальненою оцінкою',

            font='20', width=50, height=2,
            bg='#a8e6cf', fg='black',

            activebackground='#e3ecfa',
            activeforeground='white',
            relief='ridge')

            btn_summarized_rating.place(x=5
            00, y=200, anchor='center')

            btn_summarized_rating.bind('<Bu
            tton-1>', lambda event:

```

```

summarized_rating(sheet_names,
object_rating, verbal_scale))

    def choose_criteria():
        global
canvas_output_selection_buildup
        try:

canvas_output_selection_buildup
        .destroy()
            finally:
                pass

                chosen_criteria = []
                checkbutton_array = []

canvas_output_selection_buildup
=
tk.Canvas(canvas_output_selecti
on, width=700, height=900,
bg='light grey')

canvas_output_selection_buildup
.place(x=800, y=0)
        for i in
range(len(criteria_rating[0])):

chosen_criteria.append(tk.Boole
anVar())

checkbutton_array.append(tk.Che
ckbutton(canvas_output_selectio
n_buildup,

text=f'{criteria_names[i]}',
variable=chosen_criteria[i],

onvalue=1, offvalue=0,
bg='light gray'))

checkbutton_array[i].place(x=15
0, y=200+50*i)

        def confirmation():
            for k in
chosen_criteria:
                if k.get():

canvas_output_selection_buildup
        .destroy()

rating_by_criteria(sheet_names,
criteria_rating,
chosen_criteria,
values_importance)
                break

        btn_confirm_selection =
tk.Button(canvas_output_selecti

```

```

on_buildup, text='Розрахунок за
обраними критеріями',

font='20', width=35, height=2,
bg='#a8e6cf', fg='black',

activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')

btn_confirm_selection.place(x=1
50,
y=250+50*len(checkbutton_array)
, anchor='w')

btn_confirm_selection.bind('<Bu
tton-1>', lambda event:
confirmation())

    # Buttons for outpost
    selection
        btn_rating_by_criteria =
tk.Button(canvas_output_selecti
on, text='Рейтинг проєктів за
критеріями',

font='20', width=50, height=2,
bg='#a8e6cf', fg='black',

activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')

btn_rating_by_criteria.place(x=
500, y=300, anchor='center')

btn_rating_by_criteria.bind('<B
utton-1>', lambda event:
choose_criteria())

    def
choose_object(diagram=False):
        global
canvas_output_selection_buildup
        try:

canvas_output_selection_buildup
.destroy()
            finally:
                pass

canvas_output_selection_buildup
=
tk.Canvas(canvas_output_selecti
on, width=700, height=900,
bg='light gray',

highlightthickness=0)

```

```

canvas_output_selection_buildup
.place(x=800, y=0)
        object_combobox =
ttk.Combobox(canvas_output_sele
ction_buildup, width=30,
font=20, state='readonly')
        object_combobox_values
= []
        for i in sheet_names:

object_combobox_values.append(i
)

object_combobox['values'] =
object_combobox_values

object_combobox.place(x=350,
y=360, anchor='center')

        def confirmation():
            if
object_combobox.get() != '':
                chosen_object =
object_combobox.get()

canvas_output_selection_buildup
.destroy()

                if diagram:

influence_by_criteria(chosen_ob
ject, criteria_names,

criteria_rating[sheet_names.ind
ex(chosen_object)])
                else:

single_rating(chosen_object,
criteria_names,
criteria_rating[sheet_names.ind
ex(chosen_object)],

object_rating[sheet_names.index
(chosen_object)])

        btn_confirm_selection =
tk.Button(canvas_output_selecti
on_buildup, text='Розрахунок за
обраним проектом',

font='20', width=30, height=2,
bg='#a8e6cf', fg='black',

activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')

btn_confirm_selection.place(x=3
50, y=440, anchor='center')

```

```
btn_confirm_selection.bind('<Button-1>', lambda event:
confirmation())
```

```
btn_single_rating =
tk.Button(canvas_output_selecti
on, text='Оцінка окремого проекту',
```

```
font='20', width=50, height=2,
bg='#a8e6cf', fg='black',
```

```
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')
```

```
btn_single_rating.place(x=500,
y=400, anchor='center')
```

```
btn_single_rating.bind('<Button
-1>', lambda event:
choose_object())
```

```
btn_summarized_rating_diagram =
tk.Button(canvas_output_selecti
on, text='Діаграма порівняння оцінок
проектів',
```

```
font='20', width=50, height=2,
bg='#a8e6cf', fg='black',
```

```
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')
```

```
btn_summarized_rating_diagram.p
lace(x=500, y=500,
anchor='center')
```

```
btn_summarized_rating_diagram.b
ind('<Button-1>', lambda event:
summarized_rating_diagram(objec
t_rating, sheet_names))
```

```
btn_influence_by_criteria =
tk.Button(canvas_output_selecti
on, text='Діаграма впливу критеріїв
на оцінку проекту',
```

```
font='20', width=50, height=2,
bg='#a8e6cf', fg='black',
```

```
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')
```

```
btn_influence_by_criteria.place
(x=500, y=600, anchor='center')
```

```

btn_influence_by_criteria.bind(
    '<Button-1>', lambda event:
        choose_object(True))

    btn_restart =
tk.Button(canvas_output_selection,

text='Завантажити інший файл',
font='20', width=35, height=1,
bg='#a8e6cf',

fg='black',
activebackground='#e3ecfa',
activeforeground='white',
relief='ridge')
    btn_restart.place(x=300,
y=700, anchor='center')
    btn_restart.bind('<Button-
1>', lambda event:
start_window())

def read_file():
    filepath =
filedialog.askopenfilename()
    if filepath[-5:] ==
'.xlsx':
        sheet_names,
criteria_rating, object_rating,
verbal_scale, criteria_names,
values_importance =
data_processing(filepath)

output_selection(sheet_names,
criteria_rating, object_rating,
verbal_scale, criteria_names,
values_importance)
    else:
        label_file_error =
tk.Label(canvas_start,
text='Недоступний файл\n(необхідний
формат електронних таблиць)',

font='helv36', bg='light gray',
fg='red')

label_file_error.place(x=750,
y=500, anchor='center')

canvas_start = tk.Canvas(root,
width=1500, height=900,
bg='light grey')
canvas_start.place(x=0, y=0)

start_label =
tk.Label(canvas_start,
text='СИСТЕМА ОЦІНКИ ТА
ВИБОРУ НАУКОВО-ТЕХНІЧНИХ

```

ПРОЕКТІВ',

```
font=('Arial', 25, 'bold'),  
bg='light gray')  
start_label.place(x=750, y=200,  
anchor='center')
```

```
btn_open_file =  
tk.Button(canvas_start,  
text='Завантажити файл з даними',  
font=('helv36', 15), width=30,  
height=3, bg='#a8e6cf',
```

```
fg='black',  
activebackground='#e3ecfa',  
activeforeground='white',  
relief='ridge')  
btn_open_file.place(x=750,  
y=400, anchor='center')  
btn_open_file.bind('<Button-  
1>', lambda event: read_file())
```

```
start_window()
```

```
root.mainloop()
```



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНКИ  
НАУКОВИХ ТА НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ**

Виконав: Регеша Олександр Миколайович

Керівник: доцент каф. ПЗКС ФПМ, к.ф.-м.н. доцент Нещадим Олександр  
Михайлович

Київ – 2021





# ПОСТАНОВКА ЗАДАЧІ



**Мета:** Автоматизувати процес оцінювання ефективності наукових та науково-технічних проєктів шляхом розробки програмного забезпечення, що забезпечує комплексний багатокритеріальний аналіз із використанням узагальненої функції бажаності Харрінгтона, для підвищення об'єктивності, прозорості та обґрунтованості рішень щодо підтримки або фінансування дослідницьких ініціатив.

## Завдання проєкту:

1. Проаналізувати існуючі методи багатокритеріального
2. Розробити архітектуру програмного забезпечення
3. Створити графічний інтерфейс
4. Протестувати функціональність програмного засобу



# АКТУАЛЬНІСТЬ



- Науково-технічні проєкти потребують ретельного аналізу на старті.
- Помилки в оцінці знижують якість і ефективність реалізації.
- Відбір проєктів потребує автоматизації, особливо у конкурсах МОН.



# ФУНКЦІОНАЛЬНІ ВИМОГИ



- Система приймає вхідні дані з електронних таблиць.
- Користувач обирає дії зі списку (фільтрація, обчислення, тощо).
- Система виконує обрані дії та візуалізує результати (діаграми, графіки).
- Результати можна зберігати у файл.



# НЕФУНКЦІОНАЛЬНІ ВИМОГИ



- Інтерфейс має бути зрозумілим для користувача без досвіду.
- Система повинна виявляти та обробляти помилки введення.



# ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ

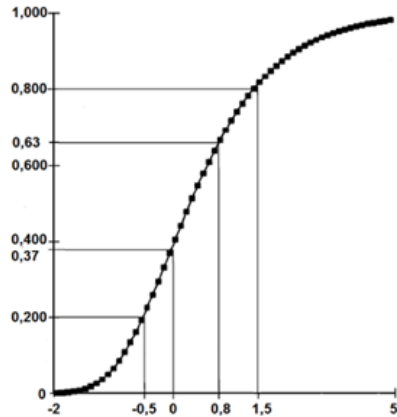


matplotlib



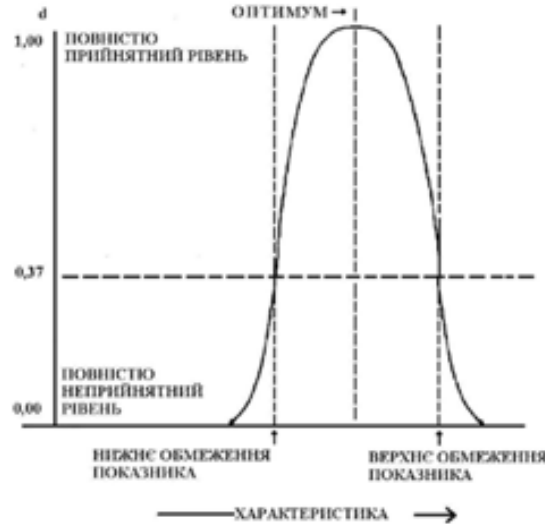
Tkinter

# ФУНКЦІЯ БАЖАНОСТІ ХАРРІНГТОНА



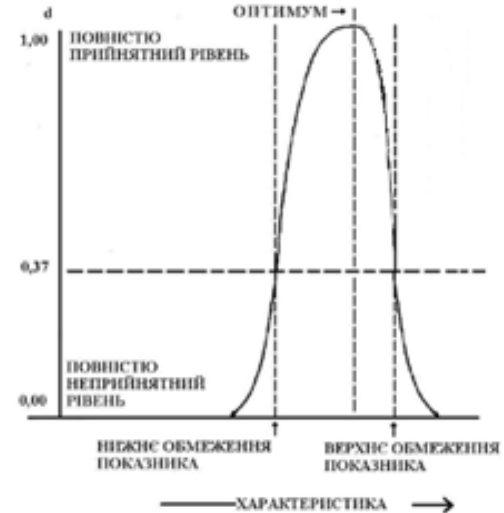
$$d = e^{-(e^{-\hat{Y}})}$$

$$\hat{Y} = -2 + 7 * \frac{(Y - Y_{min})}{(Y_{max} - Y_{min})}$$



$$d = e^{-(|\hat{Y}|)^n}$$

$$\hat{Y} = \frac{2Y - (Y_{max} + Y_{min})}{Y_{max} - Y_{min}}$$



$$d = \begin{cases} e^{-(|\hat{Y}_1|)^{n_1}}, & \text{якщо } Y < Y^* \\ 1, & \text{якщо } Y = Y^* \\ e^{-(|\hat{Y}_2|)^{n_2}}, & \text{якщо } Y > Y^* \end{cases}$$

$$\hat{Y}_1 = \frac{Y - Y^*}{Y^* - Y_{min}} \quad \hat{Y}_2 = \frac{Y - Y^*}{Y_{max} - Y^*}$$

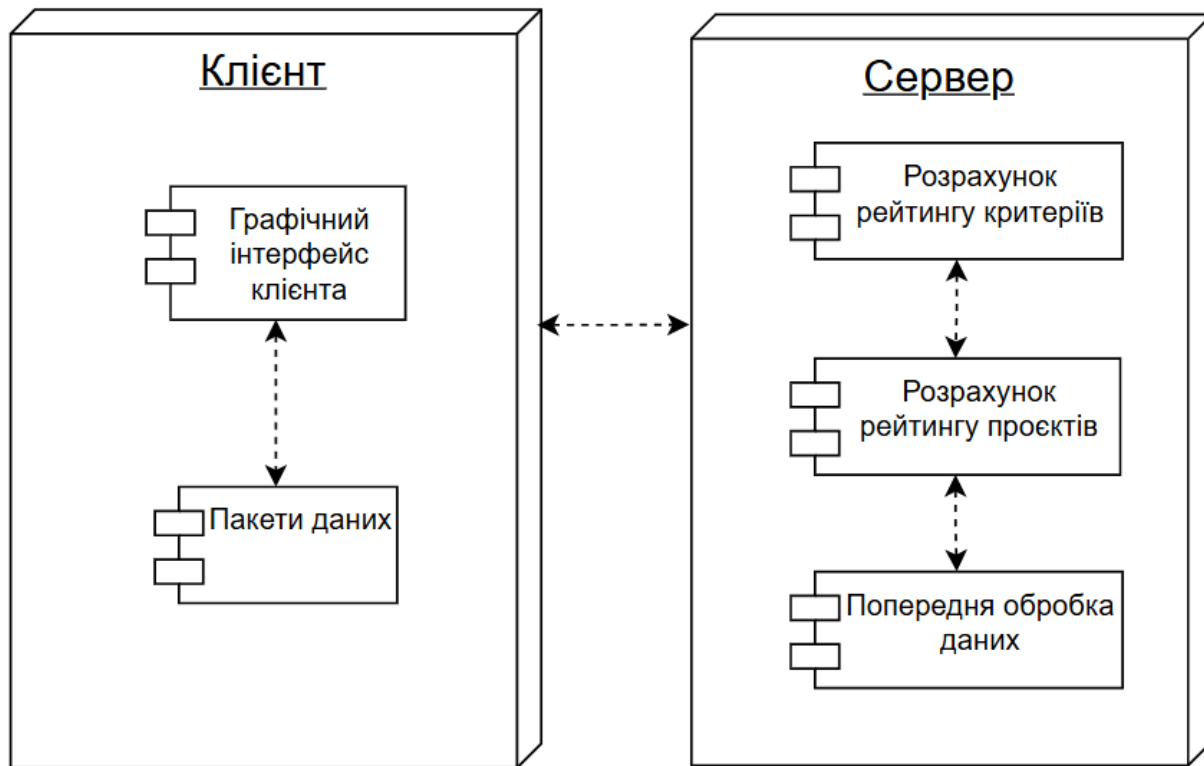


# ВИБІР КРИТЕРІЇВ ДЛЯ ОЦІНКИ ПРОЄКТІВ



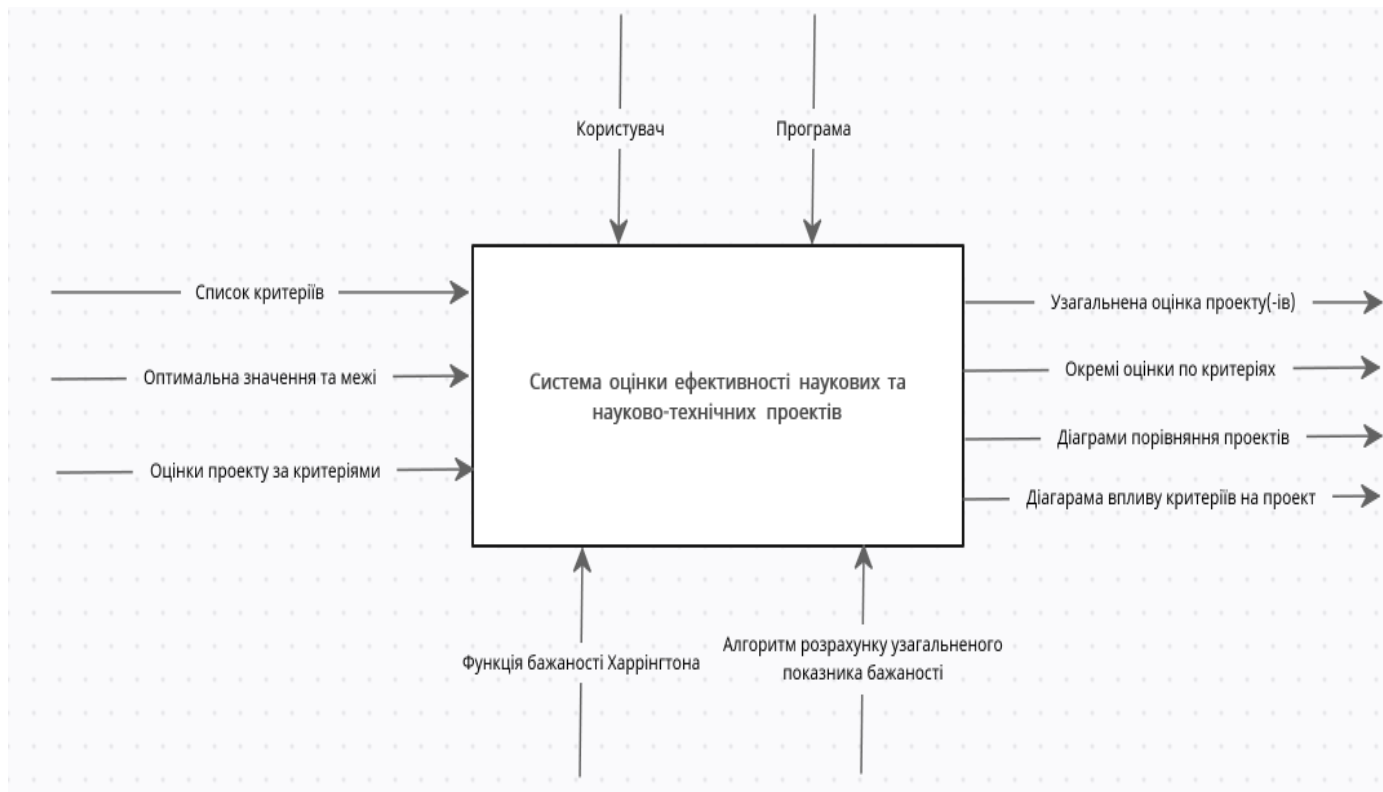
- Кошторис проекту
- Строк виконання
- Новизна
- Значимість
- Об'єктивність
- Рівень кваліфікації виконавців
- Відсоток ресурсів витрачених на теоретичну роботу
- Залучення учасників освітнього процесу

# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

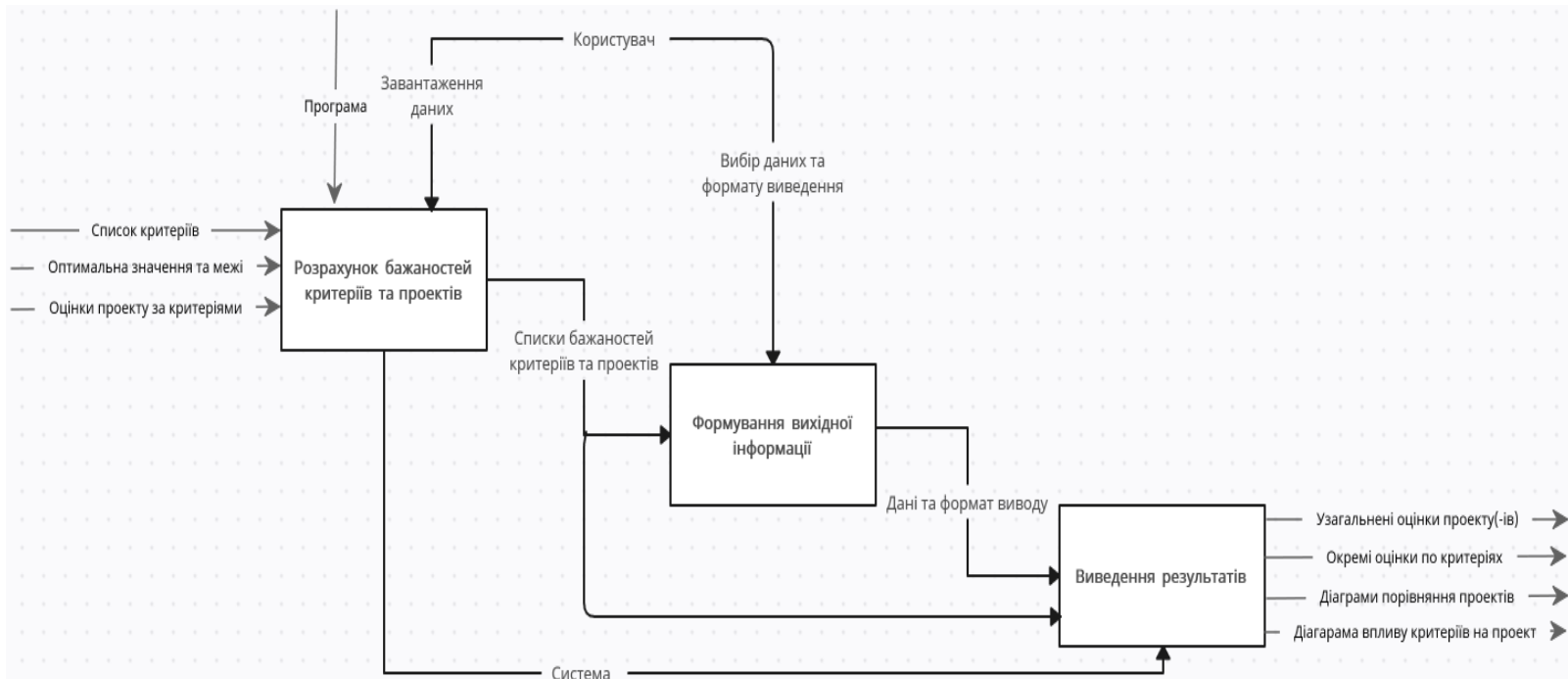




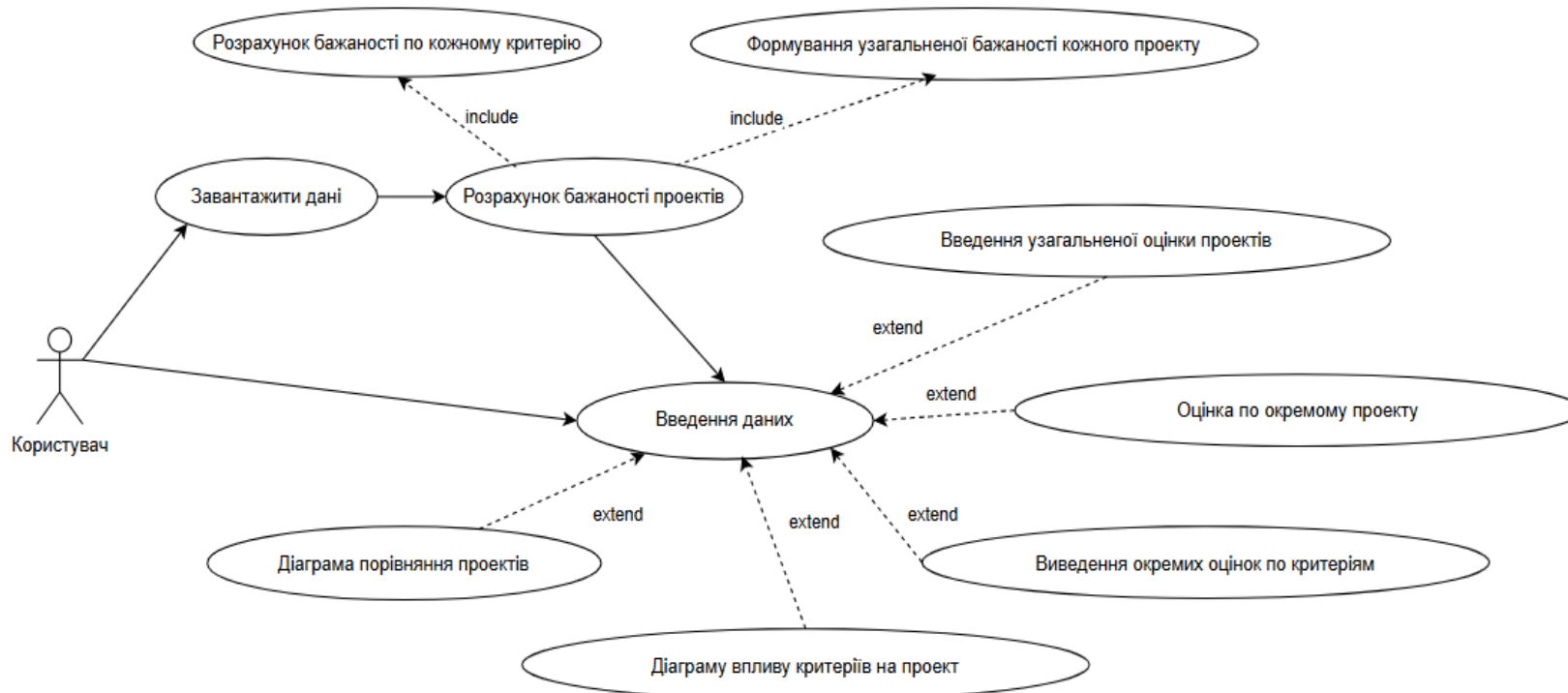
# РОЗРОБЛЕНЕ АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ



# РОЗРОБЛЕНЕ АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ



# РОЗРОБЛЕНЕ АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ





# ПОРІВНЯННЯ ПРОГРАМНОГО ПРОДУКТУ З АНАЛОГАМИ



- Метод ідеальної точки
- Метод Сааті (метод аналізу ієрархії)
- Узагальнена функція Харрінгтона





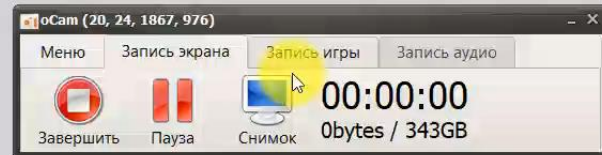
# ВИСНОВКИ



1. Створено програму для автоматизованої оцінки науково-технічних проєктів.
2. Використано функцію бажаності Харрінгтона для багатокритеріального аналізу.
3. Реалізовано зручний інтерфейс та обробку вхідних даних.
4. Тестування підтвердило коректність і надійність роботи.

# СИСТЕМА ОЦІНКИ ТА ВИБОРУ НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ

Завантажити файл з даними





**Дякую за увагу!**

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНКИ НАУКОВИХ  
ТА НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ**

**Програма та методика тестування**

ДП.045420-04-51

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олександр НЕЩАДИМ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Олександр РЕГЕША



## ЗМІСТ

|                                      |   |
|--------------------------------------|---|
| 1. Об'єкт випробувань .....          | 3 |
| 2. Мета тестування .....             | 3 |
| 3. Методи тестування.....            | 3 |
| 4. Засоби та порядок тестування..... | 4 |

## **1. ОБ'ЄКТ ВИПРОБУВАНЬ**

Об'єктом випробувань є програмна система для підтримки прийняття рішень щодо оцінки науково-технічних проєктів. Система розроблена мовою програмування Python з використанням бібліотек openpyxl, math, tkinter, matplotlib, а також забезпечує обробку вхідних Excel-файлів (XLSX), побудову графіків та інтерфейс користувача для взаємодії з системою.

## **2. МЕТА ТЕСТУВАННЯ**

Мета тестування полягає у перевірці:

1. Функціональної коректності всіх модулів системи: зчитування вхідних даних, обчислення оцінок, формування результатів, виведення графіків.
2. Сумісності з вхідними XLSX-файлами, включаючи відповідність структурі даних.
3. Точності математичних розрахунків, згідно з заданими критеріями оцінювання.
4. Стабільності роботи системи при обробці даних різного обсягу.
5. Зручності взаємодії з графічним інтерфейсом користувача.
6. Відповідності поведінки програми технічному завданню.

## **3. МЕТОДИ ТЕСТУВАННЯ**

У процесі перевірки використовувався метод "сірої скриньки" (Gray Box Testing), що передбачає часткове знання внутрішньої логіки програми, а також тестування на рівні системи. Застосовано такі методи:

1. Функціональне тестування: перевірка відповідності функціоналу очікуваним вимогам.
2. Критичний шлях (Critical Path Testing): проходження основних сценаріїв використання.

3. Тестування продуктивності.
4. Тестування стабільності (Stability Testing).
5. Тестування навантаження (Load Testing) із великим обсягом даних.
6. Тестування інтерфейсу користувача на коректність та логіку розміщення елементів.

#### **4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ**

Для тестування було використано такі засоби:

- 1) середовище PyCharm – для запуску та налагодження коду;
- 2) вбудовані засоби тестування Python (наприклад, unittest);
- 3) ручне динамічне тестування інтерфейсу користувача та введення даних;
- 4) перевірка граничних і некоректних значень у полях введення;
- 5) статичне тестування коду – перевірка відповідності стилю, пошук потенційних помилок;
- 6) тестування сумісності з XLSX-файлами різної структури;
- 7) оцінка зручності використання на основі емпіричних спостережень.

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОЦІНКИ НАУКОВИХ  
ТА НАУКОВО-ТЕХНІЧНИХ ПРОЄКТІВ**

**Керівництво користувача**

ДП.045420-05-34

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олександр НЕЩАДИМ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Олександр РЕГЕША

## ЗМІСТ

|                                      |          |
|--------------------------------------|----------|
| 1. Опис структури системи.....       | <u>3</u> |
| 2. Загальні можливості.....          | <u>4</u> |
| 3. Процедура завантаження даних..... | 5        |
| 4. Використання функціоналу.....     | 6        |
| 5. Візуалізація результатів.....     | 8        |
| 6. Збереження результатів .....      | 10       |

## **1. Опис структури системи**

Система розроблена з урахуванням принципів модульності, масштабованості та зручності використання. Основними складовими її структури є інтерфейс користувача, логіка обробки даних, модулі збереження інформації та візуалізації результатів. Кожен із цих компонентів виконує чітко визначену роль у забезпеченні ефективного функціонування всього програмного продукту.

Інтерфейс користувача реалізовано у вигляді графічного вікна, що забезпечує взаємодію з користувачем за допомогою елементів керування, таких як кнопки, текстові поля та списки. Він дозволяє здійснювати навігацію системою, запускати функціональні модулі та переглядати результати аналізу. У центральній частині інтерфейсу розміщується основне робоче поле, де відображається завантажена інформація або результати обчислень.

Логіка обробки даних зосереджена в окремому функціональному модулі, що відповідає за прийом вхідних даних, їхню валідацію, обробку та передавання у відповідні підсистеми. Цей модуль взаємодіє як із внутрішньою структурою даних, так і з зовнішніми джерелами, якщо це передбачено сценарієм використання.

Модуль збереження забезпечує запис і зчитування інформації до локального або зовнішнього сховища.

Окрему увагу приділено модулю візуалізації. Його завдання полягає в наочному представленні оброблених даних у вигляді графіків або діаграм. Це значно спрощує процес інтерпретації результатів та прийняття рішень на основі отриманої інформації.

Уся система реалізована із застосуванням об'єктно-орієнтованого підходу, що забезпечує гнучкість у зміні структури та розширенні функціоналу. Компоненти легко адаптуються до нових вимог, що робить систему придатною для довготривалого використання та розвитку.

## **2. Загальні можливості**

Система надає користувачеві широкий спектр функціональних можливостей, які охоплюють увесь цикл роботи з даними — від їхнього завантаження до збереження оброблених результатів. Вона призначена для автоматизації обробки інформації, аналітики, візуалізації та зручного управління даними, що робить її ефективним інструментом як для навчальних, так і для професійних цілей.

Однією з ключових можливостей є інтерактивна взаємодія з даними. Користувач може вільно завантажувати інформацію з локального носія або зовнішніх джерел, редагувати її безпосередньо в інтерфейсі та запускати обробку за допомогою інтуїтивно зрозумілих елементів керування. Завдяки модульному підходу система підтримує гнучке налаштування параметрів обробки, що дозволяє адаптувати її до різних сценаріїв використання.

Особливістю системи є реалізація алгоритмів автоматичного аналізу даних, які дозволяють виявляти закономірності, будувати аналітичні моделі або візуальні уявлення на основі заданих критеріїв. Візуалізація результатів відіграє важливу роль у процесі аналізу, адже наочне подання значно полегшує сприйняття складної інформації.

Крім того, система дозволяє зберігати результати у різних форматах, що забезпечує зручність під час подальшого використання отриманих даних у звітах, презентаціях або інших додатках. Передбачено також функцію експорту даних, яка дає змогу інтегрувати систему з іншими програмними засобами або платформами.

Інтерфейс системи орієнтований на користувача і побудований таким чином, щоб забезпечити максимальну простоту у користуванні навіть для осіб без спеціальної підготовки. Завдяки цьому вона може бути успішно застосована в освітньому процесі, для виконання лабораторних робіт, досліджень або аналітичних звітів.

### 3. Процедура завантаження даних

Процес завантаження даних у систему є першим і одним із найважливіших етапів роботи користувача. Від якості та коректності завантаженої інформації залежить точність подальшого аналізу та результатів обробки. Інтерфейс передбачає просту та зручну форму імпорту, що дозволяє швидко підключати необхідні файли без зайвих технічних налаштувань.

Після запуску системи користувач переходить до відповідного розділу, де може вибрати джерело даних — локальний файл або зовнішній ресурс. Система підтримує поширені формати файлів, такі як CSV, TXT, XLSX, що забезпечує сумісність з більшістю табличних редакторів і джерел інформації. Дані можуть містити як числові значення, так і текстову інформацію, яка надалі буде оброблятися або аналізуватися за заданими алгоритмами.

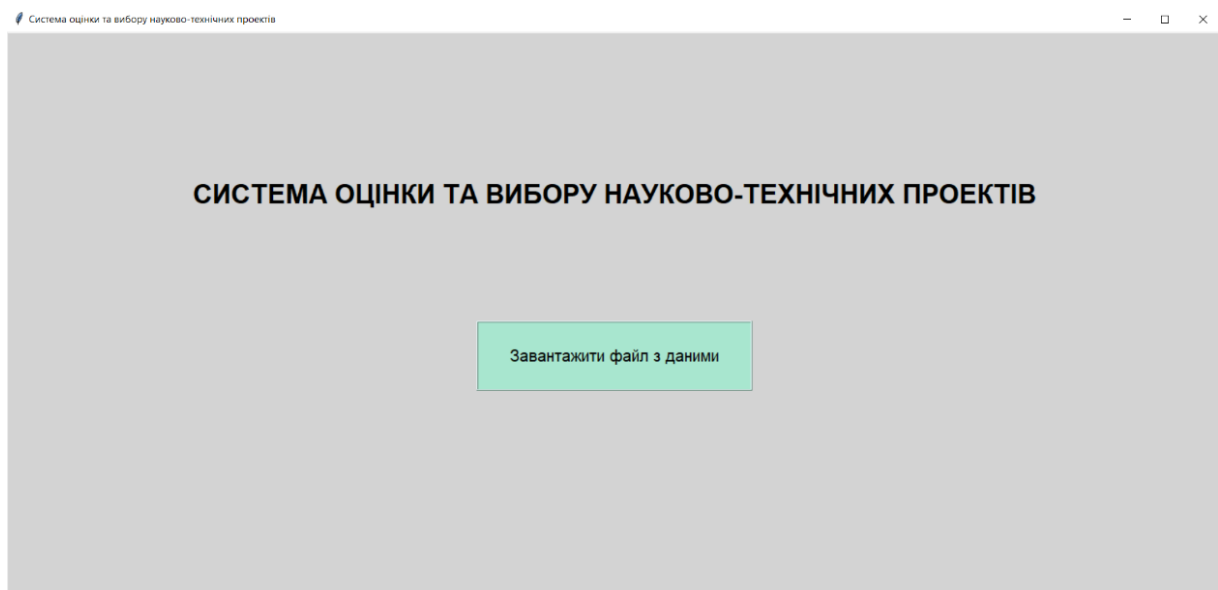


Рис. 3.1. Екран завантаження даних

Після вибору файлу система виконує попередню перевірку структури даних, що дозволяє уникнути помилок, пов'язаних із некоректним форматом або відсутністю обов'язкових полів. У разі виявлення проблем користувач отримує повідомлення з підказками щодо усунення помилок.



Якщо дані завантажено успішно, вони відображаються у таблиці, де можна переглянути або змінити окремі значення.

| № критерію | Назва критерію                           | значення | min           | max       | Тип функції:<br>одностороння(1),<br>двостороння(2) | Оптимум (для<br>двосторонньої) | Коефіцієнт<br>важливості | Коефіцієнт<br>кривизни |
|------------|------------------------------------------|----------|---------------|-----------|----------------------------------------------------|--------------------------------|--------------------------|------------------------|
| 1          | Кошторис проекту, грн                    | 150000   | 200000        | 50000     | 1                                                  |                                | 1                        |                        |
| 2          | Строк виконання, дні                     | 150      | 200           | 80        | 1                                                  |                                | 1                        |                        |
| 3          | Новизна                                  | низький  | мінімальний   | ідеальний | 1                                                  |                                | 2                        |                        |
| 4          | Значимість                               | високий  | мінімальний   | ідеальний | 1                                                  |                                | 2                        |                        |
| 5          | Об'єктивність                            | середній | низький       | ідеальний | 1                                                  |                                | 2                        |                        |
| 6          | Рівень кваліфікації виконавців           | високий  | мінімальний   | високий   | 1                                                  |                                | 1                        |                        |
| 7          | Ресурси на теоретичну роботу, %          | 20       | 10            | 30        | 2                                                  | 16                             | 1                        |                        |
| 8          | Залученість учасників освітнього процесу | низький  | незадовільний | низький   | 2                                                  | мінімальний                    | 0.5                      |                        |

Рис. 3.2. Приклад вхідних даних

Завдяки передбаченій перевірці типів даних та автоматичному розпізнаванню колонок система допомагає зберігати правильну структуру інформації, що особливо важливо для подальших обчислень або побудови візуалізацій. На цьому етапі також можна задати параметри попередньої обробки — наприклад, фільтрувати зайві рядки, замінювати пропущені значення або нормалізувати числові дані.

Таким чином, модуль завантаження даних забезпечує простий, швидкий і безпечний старт роботи в системі, надаючи користувачу інструменти для контролю та підготовки вхідної інформації відповідно до поставлених завдань.

#### 4. Використання функціональності

Після успішного завантаження даних користувач отримує доступ до основного функціоналу системи, який реалізує ключові можливості аналізу, обробки та взаємодії з інформацією. Інтерфейс системи побудований таким чином, щоб забезпечити інтуїтивну навігацію між різними модулями, дозволяючи легко переходити від одного етапу обробки до іншого.

Функціонал охоплює широкий спектр операцій: від базових дій, таких як пошук, сортування та фільтрація даних, до більш складних алгоритмічних процесів, включаючи класифікацію, обчислення статистичних показників, а також виконання специфічних перетворень. Всі

операції виконуються в межах інтерактивного середовища, де результати одразу відображаються у візуальній або табличній формі.



Рис. 4.9. Екран з варіантами вихідних даних

Система дозволяє користувачу гнучко налаштовувати параметри обробки, вибираючи потрібні методи залежно від задачі. Наприклад, у випадку роботи з числовими даними доступні функції обчислення середнього значення, медіани, дисперсії, а також побудова графіків на основі обраних параметрів. При роботі з текстовими масивами доступні інструменти лінгвістичного аналізу, підрахунок частоти слів та пошук ключових елементів.

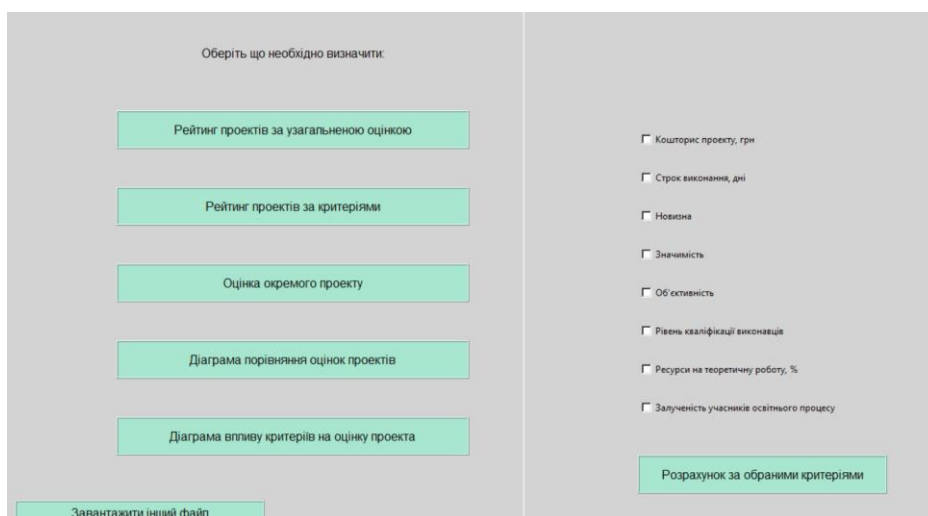


Рис. 4.10. Екран прикладом розміщення додаткових параметрів

Кожен функціональний модуль супроводжується підказками та короткими інструкціями, що значно полегшує роботу для користувачів з різним рівнем підготовки. Результати виконання операцій можуть бути переглянуті у реальному часі, а також підлягають збереженню або подальшому редагуванню. Це дає змогу не лише проводити експрес-аналіз, а й створювати глибокі дослідження з можливістю повернення до попередніх етапів.

Таким чином, використання функціоналу системи забезпечує гнучкість, швидкість і точність при роботі з даними, відповідаючи на актуальні потреби як науковців, так і фахівців прикладних галузей.

## 5. Візуалізація результатів

Візуалізація є одним із ключових етапів роботи з даними, оскільки вона дозволяє користувачу не лише інтерпретувати результати, але й виявити приховані залежності, тенденції та аномалії, які важко помітити у табличному поданні. У системі реалізовані інструменти побудови графіків, діаграм та інших візуальних елементів, які автоматично формуються на основі оброблених даних.

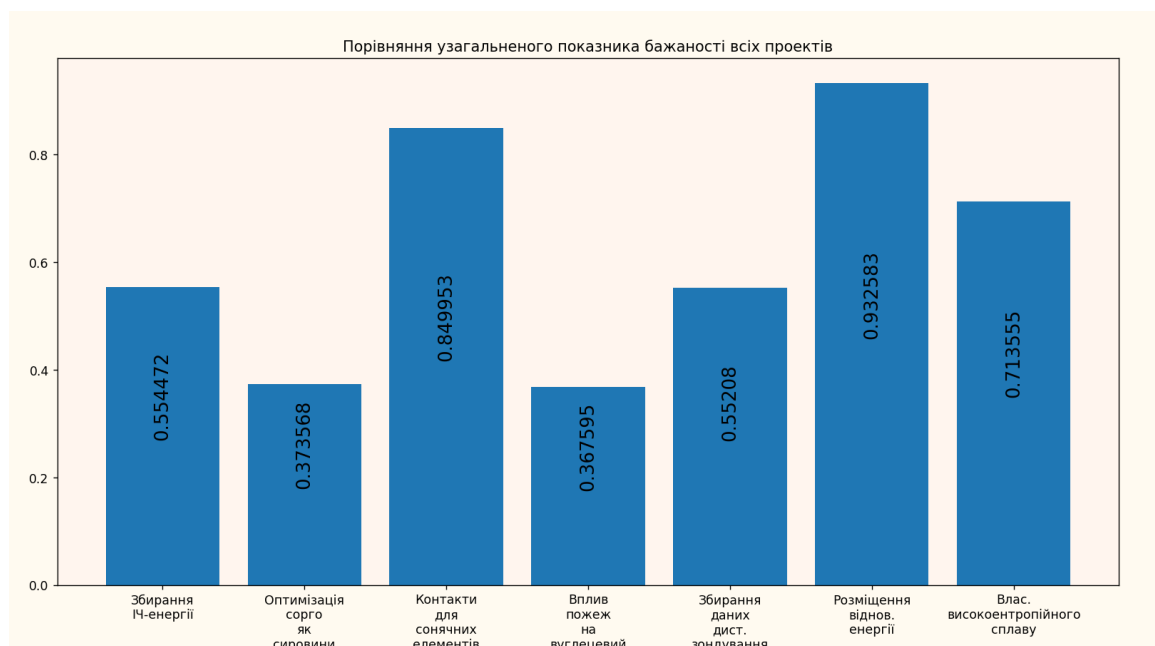


Рис. 5.1. Приклад діаграми зіставлення інтегральних оцінок

Інтерфейс дозволяє змінювати параметри виводу, включаючи кольорову гаму, тип графіка (лінійний, стовпчиковий, круговий тощо), масштабування осей та підписування елементів. Завдяки цьому користувач може адаптувати візуалізацію під свої аналітичні потреби, зробивши її максимально інформативною та зручною для подальшого використання у звітах чи презентаціях.

Кожен графічний елемент синхронізований із джерельними даними, тому будь-яка зміна у вибірці автоматично відображається на візуальному рівні. Це особливо корисно при порівнянні різних наборів даних або при послідовному аналізі результатів за певними критеріями.



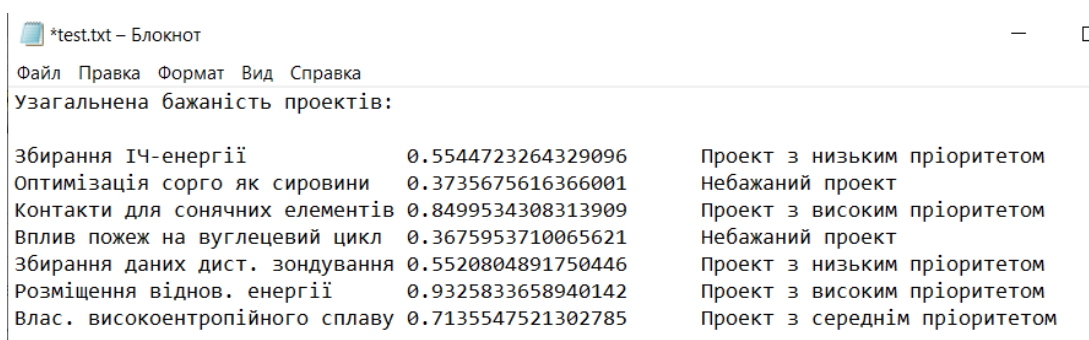
Рис. 5.2. Вплив кожного з факторів на результат окремого проекту

Система також підтримує інтерактивність у візуалізаціях: наведення курсору на певний сегмент графіка відкриває додаткову інформацію, а можливість масштабування дозволяє зосередитись на конкретних ділянках графіка. Такий підхід значно розширює аналітичний потенціал системи, оскільки забезпечує як загальний огляд, так і глибоке занурення в деталі.

Таким чином, засоби візуалізації результатів, реалізовані у системі, сприяють кращому розумінню даних, прийняттю обґрунтованих рішень та ефективному представленню аналітичної інформації.

## 6. Збереження результатів

Функціональність збереження результатів реалізована в системі з урахуванням потреб користувачів у подальшому аналізі, документуванні та презентації отриманих даних. Після завершення обробки, користувач має можливість зберегти результати в одному або кількох форматах, зокрема як текстовий звіт, таблицю, графік або повноцінний звіт із комбінованим вмістом.



|                                 |                    |                               |
|---------------------------------|--------------------|-------------------------------|
| Узагальнена бажаність проектів: |                    |                               |
| Збирання ІЧ-енергії             | 0.5544723264329096 | Проект з низьким пріоритетом  |
| Оптимізація сорго як сировини   | 0.3735675616366001 | Небажаний проект              |
| Контакти для сонячних елементів | 0.8499534308313909 | Проект з високим пріоритетом  |
| Вплив пожеж на вуглецевий цикл  | 0.3675953710065621 | Небажаний проект              |
| Збирання даних дист. зондування | 0.5520804891750446 | Проект з низьким пріоритетом  |
| Розміщення віднов. енергії      | 0.9325833658940142 | Проект з високим пріоритетом  |
| Влас. високоентропійного сплаву | 0.7135547521302785 | Проект з середнім пріоритетом |

Рис. 4.12. Приклад збереженого результату

Збереження виконується як у локальну файлову систему, так і, за потреби, у хмарне сховище або у вигляді бази даних для подальшого використання в інших модулях системи. Передбачено налаштування, які дозволяють обрати структуру та деталізацію звіту: наприклад, включити лише підсумкові значення або розгорнуту інформацію з усіма проміжними результатами.

Таким чином, система забезпечує надійне, зручне та гнучке збереження результатів, що дозволяє користувачам зберігати повний контроль над створеними даними і результатами аналізу, а також легко інтегрувати їх у зовнішні процеси обробки чи звітування.