

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

**на тему: «Система підтримки прийняття рішень для
синергетичного управління портфелями проєктів на основі
диференціальних рівнянь»**

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та
технології»

(шифр і назва)

Гарбуз Остап Володимирович

(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Кригуль Р.Є.

(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

«_____» _____ 2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Гарбузу Остапу Володимировичу

1. Тема роботи: «Система підтримки прийняття рішень для синергетичного управління портфелями проєктів на основі диференціальних рівнянь»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 28.02.2025 року № 140/к-с.

2. Строк подання студентом роботи 10.12.2025 р.

3. Вихідні дані до роботи: результати аналізу динаміки портфеля проєктів, параметри математичної моделі на основі диференціальних рівнянь, коефіцієнти синергетичної взаємодії між проєктами, вихідні дані для сценарного аналізу (початкові умови), параметри моделювання ($a, b, c, d, e, g, k, u, s$), а також програмні засоби реалізації (Python, Tkinter, Matplotlib, NumPy, Pandas).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

Теоретичні основи системи підтримки прийняття рішень для синергетичного управління портфелями проєктів на основі диференціальних рівнянь.

Вибір інструментарію та проектування СППР.

Розробка модулів та використання СППР.

Охорона праці та безпека у надзвичайних ситуаціях.

Визначення економічної ефективності.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): теоретичні основи системи підтримки прийняття рішень для синергетичного управління портфелями проєктів на основі диференціальних рівнянь; вибір інструментарію та проектування СППР; розробка модулів та використання СППР; економічна ефективність.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	Тригуба А.М., зав. кафедри інформаційних технологій		
4	Городецький І.М., доцент кафедри інженерної механіки		

7. Дата видачі завдання

28 лютого 2025 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	Написання першого розділу	28.02-20.03.25	
2	Виконання другого розділу та аркушів ілюстраційного матеріалу до нього	21.03-14.06.25	
3.	Виконання третього розділу та аркушів ілюстраційного матеріалу до нього	15.06.25-10.07.25	
4.	Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»	11.07-31.08.25	
5.	Оцінення ефективності запропонованої системи	01.09-31.10.25	
6.	Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу	01-30.11.25	
7.	Завершення роботи в цілому	01-10.12.25	

Студент _____ Гарбуз О. В.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 004.414.2:519.6

Система підтримки прийняття рішень для синергетичного управління портфелями проєктів на основі диференціальних рівнянь.

Гарбуз О.В. Кафедра інформаційних технологій – Дубляни, ЛНУВМ, 2025.

Кваліфікаційна робота: 88 с. текст. част., 15 рис., 8 табл., 14 арк. ілюстраційного матеріалу, 35 джерел.

У роботі розглянуто проблему підвищення ефективності управління портфелями проєктів на основі синергетичного підходу та математичного моделювання. Обґрунтовано необхідність створення системи підтримки прийняття рішень (СППР), яка інтегрує механізми диференціальних рівнянь для відображення динаміки портфеля, оцінки ресурсного балансу та стратегічної ефективності. Проведено аналіз теоретичних засад управління проєктами, визначено синергетичні ефекти взаємодії між ними та можливості їх використання для формування оптимальної структури портфеля. Розроблено архітектуру СППР у середовищі Python, що включає модулі математичного моделювання, сценарного аналізу, оптимізації, візуалізації, звітності та збереження результатів. Створений прототип забезпечує проведення моделювання за допомогою методу Рунге–Кутти, побудову графіків динаміки показників $N(t)$, $R(t)$, $P(t)$ та формування аналітичних звітів. Економічний аналіз показав, що система забезпечує чистий річний ефект у 230 тис. грн і окупність інвестицій менше ніж за 0,6 року, що підтверджує її практичну ефективність.

Ключові слова: система підтримки прийняття рішень, управління портфелями проєктів, синергетичний підхід, диференціальні рівняння, моделювання, Python, оптимізація, ефективність.

ЗМІСТ

ВСТУП	7
Розділ 1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМИ ПІДТРИМКИ	
ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ СИНЕРГЕТИЧНОГО УПРАВЛІННЯ	
ПОРТФЕЛЯМИ ПРОЄКТІВ НА ОСНОВІ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ	
1.1. Сутність та особливості управління портфелями проєктів у сучасному середовищі	9
1.2. Синергетичний підхід до формування та реалізації управлінських рішень	12
1.3. Використання диференціальних рівнянь у моделюванні динаміки портфеля проєктів	15
1.4. Системи підтримки прийняття рішень управління портфелями проєктів .	21
1.5. Узагальнення теоретичних засад та постановка завдань кваліфікаційної роботи	25
Розділ 2. ВИБІР ІНСТРУМЕНТАРІЮ ТА ПРОЄКТУВАННЯ СППР	
2.1. Аналіз вимог до системи підтримки прийняття рішень	28
2.2. Обґрунтування вибору мови програмування та технологічного середовища	31
2.3. Проєктування архітектури та функціональної структури системи	33
2.4. Розроблення концептуальної моделі інтерфейсу користувача	38
2.5. Схема потоків даних та взаємодії користувача із системою	42
Розділ 3. РОЗРОБКА МОДУЛІВ ТА ВИКОРИСТАННЯ СППР	
3.1. Реалізація структури додатка в середовищі Python	45
3.2. Модуль математичного моделювання динаміки портфеля проєктів	48
3.3. Модуль сценарного аналізу	50
3.4. Модуль оптимізації портфеля проєктів	54
3.5. Модулі візуалізації, звітності та збереження файлів	56
3.6. Інтерфейс користувача та демонстрація роботи системи	59
Розділ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	
4.1. Аналіз умов праці під час розробки системи підтримки прийняття рішень	64

4.2. Умови праці під час розробки системи підтримки прийняття рішень	65
4.3. Безпеки під час розробки системи підтримки прийняття рішень	66
4.4. Захист під час виникнення надзвичайних ситуацій	68

Розділ 5. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВІД ВИКОРИСТАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ СИНЕРГЕТИЧНОГО УПРАВЛІННЯ ПОРТФЕЛЯМИ ПРОЄКТІВ НА ОСНОВІ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ	70
---	----

ВИСНОВКИ І ПРОПОЗИЦІЇ.....	73
----------------------------	----

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
----------------------------------	----

ДОДАТКИ	80
---------------	----

Додаток А. Фрагмент коду СППР для синергетичного управління портфелями проєктів на основі диференціальних рівнянь.....	81
--	----

ВСТУП

Сучасний розвиток економіки та суспільства все більшою мірою залежить від здатності організацій управляти складними портфелями проєктів із динамічним проєктним середовищем, у якому одночасно впливає велика кількість взаємопов'язаних чинників. У таких умовах актуалізується потреба у формуванні ефективних систем підтримки прийняття рішень, здатних забезпечувати синергетичний ефект при управлінні портфелями проєктів. Це вимагає застосування новітніх математичних методів і обчислювальних технологій, що дозволяють не лише координувати окремі проєкти, а й формувати єдину динамічну модель їх розвитку.

Використання диференціальних рівнянь у моделюванні процесів управління дає можливість відображати динаміку змін параметрів портфеля проєктів у часі та досліджувати вплив взаємодії між окремими його складовими. Такі підходи дозволяють формалізувати нелінійні залежності, враховувати ефекти самоорганізації та резонансного підсилення управлінських рішень, що є основними ознаками синергетичного підходу [25].

Об'єктом дослідження у даній кваліфікаційній роботі є процес управління портфелями проєктів у складному проєктному середовищі.

Предметом дослідження виступають методи та моделі синергетичного управління портфелями проєктів із застосуванням диференціальних рівнянь.

Засобами дослідження є системи підтримки прийняття рішень, побудовані на основі математичного апарату диференціальних рівнянь та програмних засобів моделювання, що забезпечують реалізацію обчислювальних експериментів і аналіз результатів.

Актуальність теми зумовлена тим, що традиційні методи управління портфелями проєктів часто обмежуються статичним аналізом і не враховують динаміку розвитку середовища, що призводить до втрати ефективності управлінських рішень. Запропонований підхід, заснований на синергетичній

концепції, дозволяє не лише оптимізувати розподіл ресурсів, а й підвищити стійкість портфеля до невизначеностей та зовнішніх викликів [23].

Таким чином, розробка системи підтримки прийняття рішень для синергетичного управління портфелями проєктів на основі диференціальних рівнянь має як теоретичну, так і практичну значущість, оскільки сприяє вдосконаленню механізмів стратегічного розвитку організацій та підвищенню ефективності використання ресурсів у динамічному проєктному середовищі.

Розділ 1.

ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ СИНЕРГЕТИЧНОГО УПРАВЛІННЯ ПОРТФЕЛЯМИ ПРОЄКТІВ НА ОСНОВІ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

1.1. Сутність та особливості управління портфелями проєктів у сучасному середовищі

Управління портфелями проєктів стало одним із основних напрямів розвитку сучасного менеджменту, оскільки саме на цьому рівні забезпечується узгодження стратегічних цілей організації з практичними результатами реалізації окремих ініціатив. Портфель проєктів розглядається як сукупність взаємопов'язаних проєктів і програм, що координуються для досягнення спільних цілей та оптимального використання ресурсів. На відміну від управління окремим проєктом, яке здебільшого зосереджується на часових рамках, вартості та якості виконання, управління портфелем передбачає балансування між багатьма ініціативами, враховуючи їх пріоритетність, ризики та стратегічну доцільність [1].

Сутність портфельного управління полягає у виборі, моніторингу та адаптації набору проєктів, що максимально відповідають стратегії організації. Для формалізації цього процесу часто використовують модель, яка може бути подана у вигляді системи рівнянь:

$$P = \{ p_1, p_2, \dots, p_n \}, \quad V(P) = \sum_{i=1}^n w_i \cdot v(p_i), \quad (1.1)$$

де P – портфель проєктів; p_i – окремий проєкт; $v(p_i)$ – цінність проєкту; w_i – ваговий коефіцієнт залежно від стратегічних пріоритетів; $V(P)$ – сумарна цінність портфеля.

Завдяки такій формалізації стає можливим не лише оцінювати ефективність кожного проєкту, а й здійснювати синергетичний аналіз, коли

взаємодія між проєктами призводить до ефектів підсилення чи зменшення загальної вартості портфеля.

Сучасне проєктне середовище, яке є визначальним під час управління портфелями проєктів, характеризується високим рівнем невизначеності, багатоваріантністю сценаріїв розвитку та обмеженістю ресурсів. Це вимагає від менеджерів використання гнучких підходів, здатних швидко реагувати на зовнішні виклики. Значну увагу у наукових працях приділяють тому, що цифровізація процесів і впровадження аналітичних інструментів дають змогу підвищити прозорість прийняття рішень [27].

Управління портфелями в таких умовах ґрунтується на кількох основних принципах. По-перше, необхідно забезпечувати стратегічну узгодженість усіх проєктів, щоб уникнути розпорошення ресурсів. По-друге, важливо враховувати ризики та можливості, які виникають у ході реалізації портфеля, що вимагає інтеграції методів оцінки ризиків. По-третє, сучасні умови формують потребу в адаптивності систем управління, здатних швидко змінювати структуру портфеля залежно від нових обставин [30].

Таблиця 1.1 – Порівняння управління проєктом та управління портфелем проєктів

Критерій	Управління проєктом	Управління портфелем
Основна мета	Завершення проєкту у строк, бюджет і якість	Максимізація стратегічної цінності
Масштаб	Один проєкт	Сукупність проєктів і програм
Орієнтація	Тактична	Стратегічна
Управлінські інструменти	Планування, контроль, координація	Вибір, пріоритезація, балансування
Основний критерій ефективності	Виконання ТЗ	Досягнення стратегічних цілей

З огляду на наведене порівняння, можна зробити висновок, що управління портфелем проєктів не лише об'єднує традиційні функції управління, а й передбачає інтеграцію стратегічних механізмів, які дозволяють формувати довгострокову цінність для організації.

Динаміка портфеля проєктів

Однією з основних особливостей сучасного портфельного управління є динамічність його структури. Проєкти можуть завершуватися, призупинятися або додаватися нові ініціативи залежно від зовнішніх і внутрішніх факторів. Такий процес можна описати у вигляді диференціального рівняння:

$$\frac{dP(t)}{dt} = f(P(t), R(t), E(t)), \quad (1.2)$$

де $P(t)$ – склад портфеля у момент часу t ; $R(t)$ – ресурсні обмеження; $E(t)$ – зовнішнє середовище; $f(P(t), R(t), E(t))$ – функція, що відображає характер змін у портфелі.

Це рівняння демонструє, що управління портфелем проєктів має динамічний характер і залежить не лише від внутрішніх рішень, а й від змін у зовнішньому середовищі, які можуть бути непередбачуваними.



Рисунок 1.1 – Схематичне відображення взаємозв'язку між стратегією організації, портфелем проєктів та окремими проєктами [18]

У цій схемі показано ієрархічну залежність: стратегія визначає загальні пріоритети, портфель формує набір проєктів, а кожна програма чи проєкт є конкретним інструментом їх реалізації.

Таким чином, сутність управління портфелями проєктів полягає у створенні механізму, який дозволяє ефективно поєднувати стратегічні орієнтири з практичною реалізацією ініціатив. Особливістю сучасного середовища є його мінливість і складність, що вимагає від систем управління здатності до адаптації, інтеграції інструментів оцінки ризиків і застосування математичних моделей для аналізу динаміки. Це підкреслює важливість подальшого дослідження систем підтримки прийняття рішень, які базуються на диференціальних рівняннях, що й буде розглянуто в наступних підрозділах.

1.2. Синергетичний підхід до формування та реалізації управлінських рішень

Сучасні умови проєктного середовища характеризуються високим рівнем турбулентності, постійною зміною зовнішнього середовища та необхідністю швидкої адаптації організацій до нових викликів. У таких умовах традиційні лінійні методи управління часто виявляються недостатньо ефективними, адже вони не враховують складних нелінійних зв'язків між елементами системи. Саме тому виникає потреба у застосуванні синергетичного підходу, який дозволяє пояснити процеси самоорганізації, коеволюції та узгодженої дії окремих компонентів системи [24].

Синергетика розглядає організацію як відкриту нелінійну систему, яка взаємодіє з середовищем та здатна переходити від хаотичних станів до впорядкованих структур завдяки механізмам самоорганізації. В управлінні портфелями проєктів це означає, що ефективність досягається не стільки завдяки жорсткій регламентації, скільки завдяки узгодженню цілей, ресурсів та процесів різних проєктів у межах єдиної стратегічної логіки.

Відповідно до базових принципів синергетики, розвиток системи можна описати через рівняння порядку:

$$\frac{dx}{dt} = F(x, \alpha), \quad (1.2)$$

де x – вектор стану системи; α – параметри управління; $F(x, \alpha)$ – функція, яка визначає динаміку змін.

Для портфеля проєктів це означає, що зміни його структури залежать не лише від характеристик окремих проєктів, а й від керуючих впливів з боку проєктних менеджерів.

Особливу роль відіграє явище біфуркацій – точок, у яких система може перейти до якісно нового стану. У випадку управління портфелями це може бути рішення про завершення частини проєктів, перерозподіл ресурсів чи запуск нових ініціатив, що кардинально змінює траєкторію розвитку організації [29].

Синергетичний підхід в управлінні передбачає поєднання трьох рівнів: стратегічного, тактичного та операційного. На стратегічному рівні формуються параметри порядку, які визначають довгострокові орієнтири. На тактичному рівні здійснюється узгодження між проєктами, а на операційному – відбувається самоорганізація команд та ресурсів.

Така багаторівнева структура може бути відображена у вигляді таблиці 1.2.

Таблиця 1.2 – Рівні застосування синергетичного підходу в управлінні портфелями проєктів

Рівень	Основні завдання	Інструменти реалізації
Стратегічний	Формування параметрів порядку, визначення вектору розвитку	Візійне планування, системна динаміка
Тактичний	Узгодження цілей і ресурсів між проєктами	Моделі оптимізації, методи пріоритизації
Операційний	Самоорганізація виконавців і ресурсних потоків	Гнучкі методології, Agile-підходи

Головною метою застосування синергетичного підходу є досягнення ефекту, коли результат взаємодії елементів системи перевищує просту суму результатів їх окремого функціонування. Для формалізації цього явища можна записати співвідношення:

$$E_{syn} = R_{sys} - \sum_{i=1}^n R_i, \quad (1.3)$$

де E_{syn} – синергетичний ефект; R_{sys} – інтегральний результат функціонування системи; R_i – результат окремого елемента.

Позитивне значення E_{syn} означає наявність синергії, а від’ємне – конфлікт чи неефективність узгодження.

За допомогою ітеративного аналізу ми визначили три основні теми, які суттєво сприяють нашому розумінню синергії людини та ШІ в контексті прийняття рішень [17]. Ці теми включають можливості ШІ в процесі прийняття рішень, закономірності синергії людини та ШІ в різних завданнях прийняття рішень та результати синергії людини та ШІ в процесі спільного прийняття рішень. Загальні результати цього високорівневого синтезу синергії людини та ШІ в процесі прийняття рішень візуально зображені на рис. 1.2.

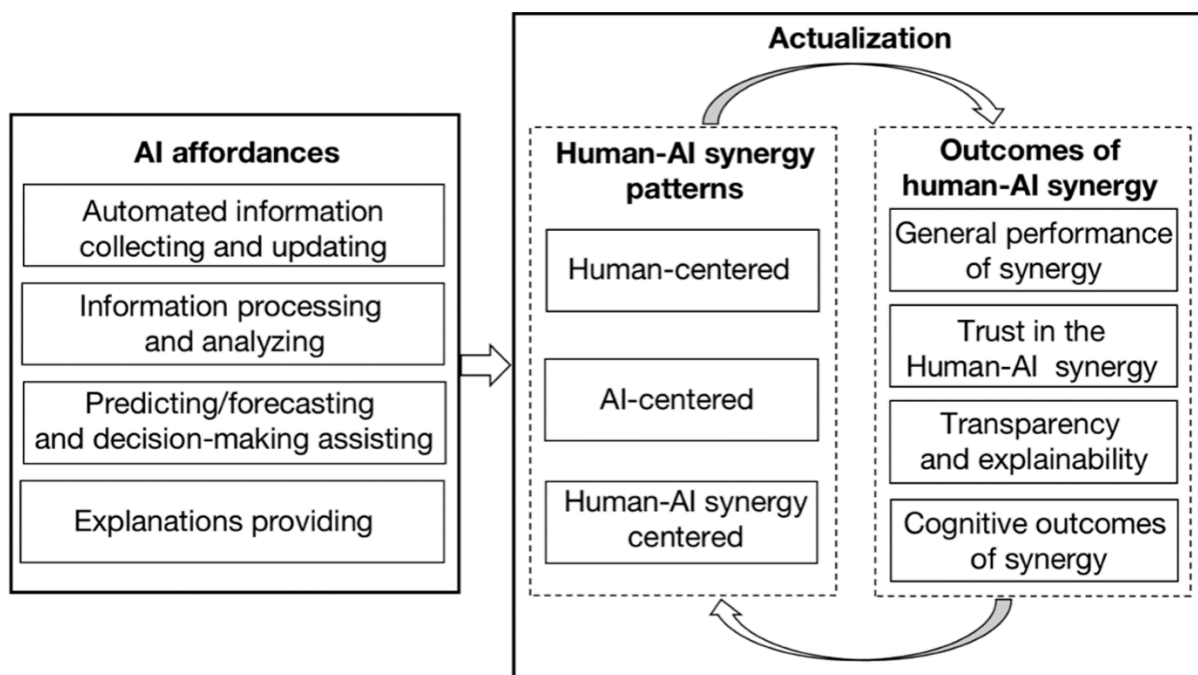


Рисунок 1.2 – Високорівневий синтез досліджень синергії людини та штучного інтелекту в процесі прийняття рішень [17]

Управлінські рішення у синергетичному підході формуються з урахуванням того, що організація має властивості нелінійної системи. Це вимагає прогнозування різних сценаріїв та побудови моделей, які здатні відобразити поведінку системи у разі зміни основних параметрів.

Впровадження синергетичного підходу дозволяє досягти кількох важливих результатів. По-перше, забезпечується гнучкість системи за рахунок здатності до самоорганізації. По-друге, підвищується стійкість портфеля до зовнішніх викликів, оскільки система здатна змінювати конфігурацію у відповідь на коливання середовища. По-третє, створюється основа для інтеграції диференціальних рівнянь у моделювання динаміки управління, що забезпечує більш точний прогноз розвитку портфеля [28].

Таким чином, синергетичний підхід формує нове бачення управлінських процесів, у якому рішення сприймаються не як статичні інструкції, а як динамічні механізми, здатні адаптуватися до змін і створювати додану вартість завдяки взаємодії елементів. Це відкриває перспективи для поєднання методів синергетики з сучасними інструментами підтримки прийняття рішень, що й становить основу подальших досліджень.

1.3. Використання диференціальних рівнянь у моделюванні динаміки портфеля проєктів

Динаміка портфеля проєктів відображає змінність його характеристик у часі – зміну кількості проєктів, розподіл ресурсів, рівень ризику, внесок кожного проєкту у загальну стратегію організації. Щоби аналіз цієї змінності був якісним і забезпечував прогнози, моделі, засновані на диференціальних рівняннях, дають змогу формалізувати часові залежності та вплив різних факторів на розвиток портфеля. Такі моделі дозволяють не лише бачити поточний стан, але й розраховувати темпи змін, точки переходу, стабільні або нестабільні стани системи.

Одним з прикладів такого підходу є модель, запропонована у праці [20]. Її автори розробили динамічну систему проєктного портфеля із застосуванням диференціальних рівнянь першого порядку. У цій моделі враховано зворотні зв'язки (feedback loops) між стратегічним мисленням, управлінням портфелем проєктів і виробничими показниками. Зміни темпів зростання або спаду у моделі представлені через диференціальні формули, що дозволяють симулювати як проміжні стани, так і довгострокову траєкторію.

Основна форма моделі, підходяща для портфеля проєктів, може бути описана системою звичайних диференціальних рівнянь:

$$\begin{cases} \frac{dN(t)}{dt} = a \cdot S(t) - b \cdot N(t) + u(t) \\ \frac{dR(t)}{dt} = c \cdot N(t) - d \cdot R(t) \\ \frac{dP(t)}{dt} = e \cdot f(R(t), N(t)) - g \cdot P(t) \end{cases}, \quad (1.4)$$

де $N(t)$ – кількість активних проєктів у портфелі в момент часу t ; $R(t)$ – задіяні ресурси; $P(t)$ – рівень стратегічної продуктивності портфеля; a, b, c, d, e, g – коефіцієнти, які моделюють приплив, втрати, зворотний зв'язок; $u(t)$ – функція, що представляє зовнішній вплив чи інвестиційний тиск; $f(R, N)$ – функція, що відображає взаємодію ресурсів та кількості проєктів у формі, скажімо, насичення чи синергетичного підсилення.

Функцію $f(R, N)$ часто обирають у вигляді нелінійної взаємодії, наприклад:

$$f(R, N) = \frac{R \cdot N}{k + N}, \quad (1.5)$$

де k – параметр, який визначає, при якій «кількості» ефект від взаємодії ресурсів і кількості проєктів перестає зростати лінійно.

Серед основних переваг диференціальних моделей – можливість знаходити стаціонарні точки (steady states), аналіз стабільності таких точок, дослідження біфуркацій, тобто моментів, коли невеличка зміна параметрів

призводить до якісно іншої поведінки системи. Наприклад, якщо параметри a чи c змінюються, може статися, що $\frac{dN}{dt}=0$ та $\frac{dR}{dt}=0$ одночасно лише при невеликих N, R , або ж при значних – залежно від того, як підібрані коефіцієнти.

У таблиці 1.3 наведено порівняння двох способів моделювання динаміки портфеля: через диференціальні моделі та через емпіричні стаціонарні підходи (без диференціальних рівнянь).

Таблиця 1.3 – Порівняння диференціального моделювання та стаціонарних емпіричних підходів

Характеристика	Диференціальне моделювання	Стаціонарні підходи / емпіричні аналізи
Врахування часу	Моделювання змін у часі	Часто тільки фіксовані спостереження або ретроспективний аналіз
Прогностичність	Можна отримати прогнози для майбутніх станів	Обмежено прогнозами на основі історичних даних
Аналіз невизначеності і чутливості	Можна дослідити, як зміни параметрів впливають на систему	Менша гнучкість, складно врахувати змінні умови
Зворотні зв'язки (feedback loops)	Реалізуються прямо через диференціальні співвідношення	Можуть враховуватися частково або опосередковано
Складність впровадження	Вища – потребує знань, калібрування параметрів	Нижча – можуть базуватися на статистичних даних чи експертних оцінках

Щоби вільно ілюструвати структуру такої моделі, малюнок наведений нижче показує умовну схему динамічної моделі з диференціальними рівняннями.

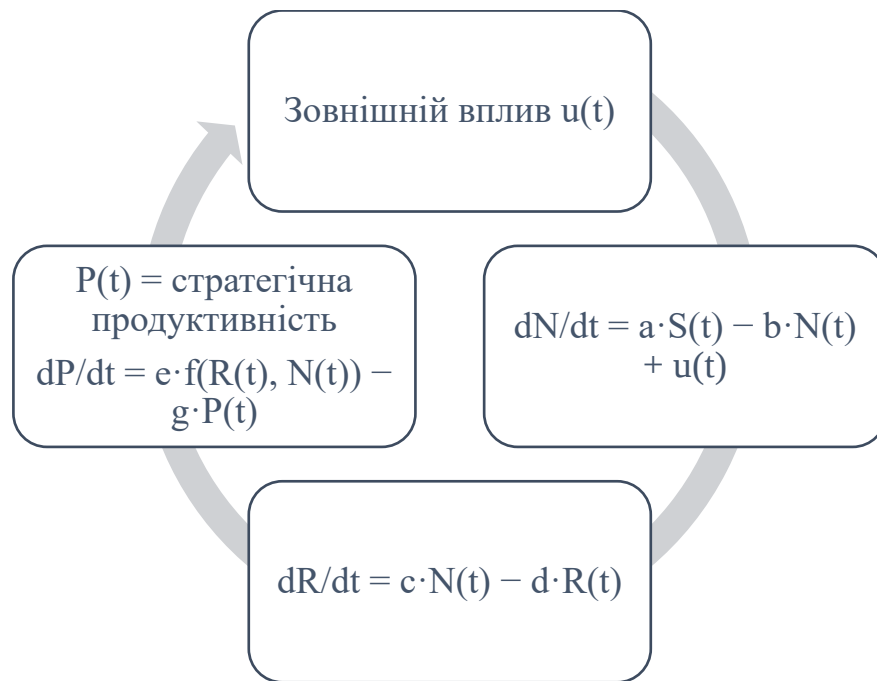


Рисунок 1.3 – Схема структури динамічної моделі портфеля проєктів з диференціальними рівняннями

Ця схема показує потік інформації: зовнішній вплив формує зміну кількості активних проєктів $N(t)$, яка через ресурси $R(t)$ впливає на продуктивність портфеля $P(t)$.

Приклад практичного застосування наведено у колоніальній індустрії (птахівництво) в дослідженні Cadenas-Anaya та співавторів, де модель із системної динаміки з диференціальними рівняннями дозволила відобразити зростання та спад виробничих показників залежно від стратегічних рішень і ресурсного тиску [20].

Загалом, використання диференціальних рівнянь дає змогу моделювати такі процеси, як:

- ✓ інтеграція нових проєктів у портфель із урахуванням обмежених ресурсів;
- ✓ завершення або зупинка проєктів унаслідок високого ризику чи низької цінності;
- ✓ розподіл ресурсів між активними проєктами залежно від їх ефективності і внеску у стратегію;

✓ реакцію портфеля на зовнішні зміни (економічні, технологічні, політичні).

Для чисельної реалізації таких моделей часто застосовують методи наближення, наприклад, метод Рунге-Кутти для звичайних диференціальних рівнянь. Нехай маємо задачу Коші:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0. \quad (1.6)$$

Потрібно знайти наближене значення розв'язку $y(t)$ на відрізку $[t_0, T]$ з кроком h . Формули Рунге-Кутти 4-го порядку:

$$\begin{aligned} k_1 &= h \cdot f(t_n, y_n), \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \\ k_4 &= h \cdot f(t_n + h, y_n + k_3), \end{aligned} \quad (1.7)$$

а нове значення y_{n+1} обчислюється так:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (1.8)$$

де k_1 – відповідає нахилу дотичної в початковій точці (t_n, y_n) ; k_2 і k_3 – уточнюють нахил у середині інтервалу; k_4 – оцінює нахил у кінці інтервалу.

Середньозважена комбінація цих значень дає високу точність порядку $O(h^4)$.

Окрім того наявні приклади використання програмних середовищ для системної динаміки (System Dynamics), які дозволяють візуалізувати плин змін, будувати сценарії.

Системна динаміка використовується для розробки моделі вимірювання ефективності управління проектами та для подальшого визначення компонентів, що впливають на продуктивність системи [21]. Фундаментальна мета використання системної динаміки полягає в розумінні поведінки системи. Поведінка кожного елемента в системі є важливою для оцінки різних дій у різних

частинах системи, щоб підкреслити або послабити її поведінкову тенденцію. Модель системної динаміки для вимірювання ефективності управління проектами зображена на рис. 1.4.

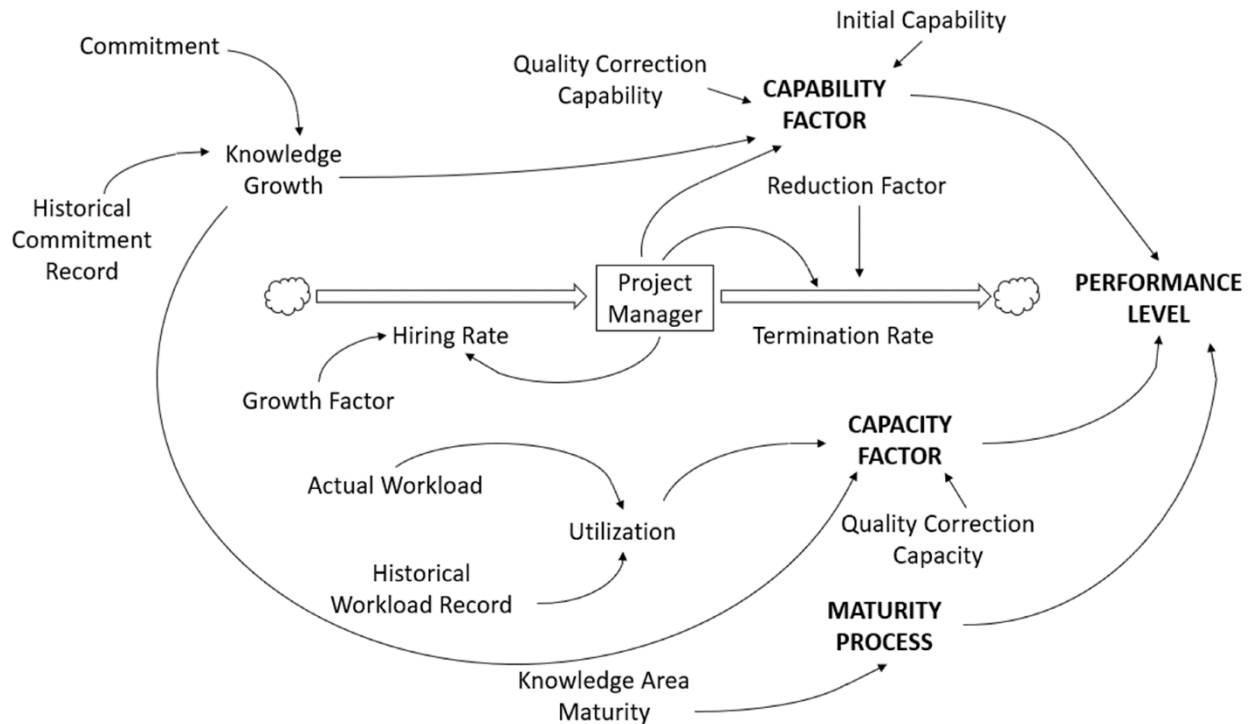


Рисунок 1.4 – Системна динамічна модель для вимірювання ефективності управління проектами [21]

Для розробки моделі системи управління проектами було використано системно-динамічний комп'ютерний програмний інструмент Vensim. Змінні моделі були оцінені на основі теоретичної інтерпретації та загальноприйнятих значень у будівельній галузі. Успіх моделі залежить від того, чи може модель імітувати реальну подію. Плинність кадрів є основним фактором, що сприяє ефективності будь-якої організації; тому утримання персоналу має бути основним завданням керівництва для підвищення рівня ефективності. Реальною подією, що розглядається в цій моделі, є зобов'язання керівництва (вищого керівництва) щодо утримання персоналу та керівників проектів.

Завдяки такому підходу менеджмент портфеля отримує інструмент, який може не лише показати, «що відбувається», але й передбачити «що може

статися» при різних стратегічних варіантах. Це особливо важливо в умовах невизначеності, коли рішення мусять прийматися швидко та бути адаптивними.

1.4. Системи підтримки прийняття рішень управління портфелями проєктів

Системи підтримки прийняття рішень (СППР, англ. Decision Support Systems, DSS) в управлінні портфелями проєктів відіграють основну роль у зменшенні рівня невизначеності, усуненні суб'єктивності в пріоритезації та розподілі ресурсів. Вони інтегрують аналітичні, ітераційні та інколи інтерактивні компоненти, що допомагають керівникам бачити як поточний стан портфеля, так і можливі сценарії розвитку. Ідея полягає в тому, щоб забезпечити підтримку не лише на етапі вибору проєктів, але й на етапах моніторингу, коригування та адаптації портфеля.

СППР для портфельного управління характеризуються набором основних функцій – збір і інтеграція даних про всі активні й потенційні проєкти; багатокритеріальна оцінка ризиків, витрат, вигод, стратегічної відповідності; генерація альтернативних варіантів портфеля; симуляція й аналіз можливих сценаріїв; візуалізація основних показників і узгодження із пріоритетами стратегії. Щоб система функціонувала ефективно, вона має підтримувати відстеження змін у зовнішньому середовищі, реагувати на ризики та коригувати стратегію в реальному часі або з прийнятним інтервалом.

У літературі є приклади СППР, які об'єднують і оптимізаційні моделі, і методи мультикритеріального прийняття рішень (MCDM), і симуляційні підходи. Так, у роботі [19] описано DSS, який на етапі відбору компонентів портфеля використовує цільове програмування (goal programming) та цілочисельне лінійне програмування (0-1 integer programming) для максимізації загальної вигоди за обмежень ресурсів. У дослідженні [26] побудували

інструмент процесної моделі, яка підтримує архітектуру вибору та оцінки за різними критеріями, із візуалізацією даних і підтримкою сценарного аналізу.

Для формалізації роботи СППР часто використовують математику: якщо позначити множину кандидатів проєктів як $\{p_1, p_2, \dots, p_n\}$, кожному проєкту присвоюють вектор характеристик $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$, де кожен x_{ij} – критерій оцінки (витрати, ризик, стратегічна важливість, очікувана вигода тощо). Завдання вибору портфеля можна формалізувати як:

$$\max_{y \in \{0,1\}^n} \sum_{i=1}^n y_i \cdot B(p_i) \quad \text{за умов} \quad \sum_{i=1}^n y_i \cdot C(p_i) \leq C_{\max}, \quad R(p_i, y) \leq R_{\max}. \quad (1.9)$$

де $y_i = 1$ – якщо проєкт p_i обирається до портфеля; $B(p_i)$ – функція вигоди; $C(p_i)$ – витрати ресурсу; $C_{\max}$ – гранично допустимий ресурс; $R(p_i, y)$ – ризик, можливо згрупований або агрегований як функція від вибору портфеля, $R_{\max}$ – допустимий рівень ризику.

Іншим підходом є комбінування симуляційних моделей та аналізу сценаріїв. Симуляція дозволяє моделювати, як портфель реагує на зміни основних параметрів: наприклад, варіацію обсягів ресурсів, затримки, зміни пріоритетів. Такий підхід добре себе проявляє у великих організаціях або в держорганах, де можливі значні зовнішні шоки або політичні, економічні зміни.

Для ілюстрації структури СППР наведена схема (рис. 1.5).

У схемі видно, що система починається з даних (внутрішніх і зовнішніх), далі проводиться нормалізація та узгодження критеріїв, після цього здійснюється оцінка і вибір за допомогою оптимізації, потім аналіз можливих сценаріїв і завершально – генерація рекомендацій для менеджменту.

Однією з формальних моделей, яку часто використовують як підмодель у DSS, є модель багатоваріантного вибору з пріоритетами, наприклад:

$$U(y) = \sum_{i=1}^n y_i \sum_{j=1}^m w_j \cdot f_j(x_{ij}). \quad (1.10)$$

де w_j – вага критерію j ; f_j – функція корисності або оцінки для проєкту по критерію j . Задача:

$$\max_{y \in \{0,1\}^n} U(y) \quad \text{за умов} \quad \sum_{i=1}^n y_i \cdot C(p_i) \leq C_{\max}. \quad (1.11)$$

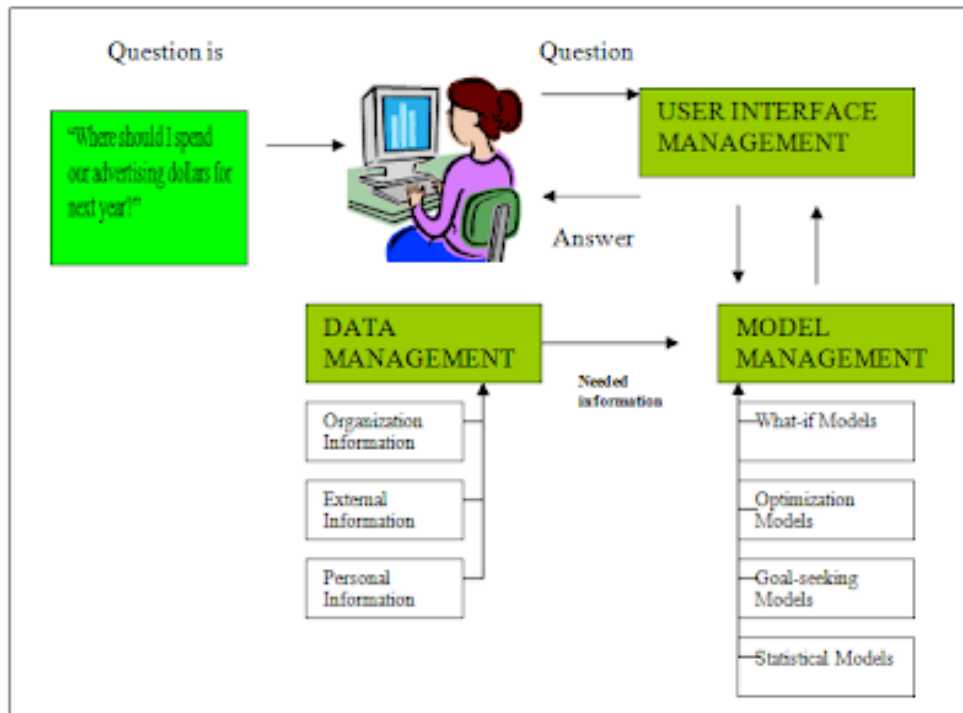


Рисунок 1.5 – Схематичне зображення функціональних компонентів системи підтримки прийняття рішень для управління портфелем проєктів

Такі корисні функції (utility functions) можуть бути лінійними або нелінійними, можуть включати штрафи за ризик чи за перевищення ресурсів.

У таблиці 1.4 нижче показано порівняння, які компоненти й підходи в СППР можуть бути, залежно від рівня складності та ресурсів організації:

Функціональна формула, яка враховує важливість критеріїв та норму ризику, може виглядати так:

$$\max_y \left(\sum_{i=1}^n y_i \left(\sum_{j=1}^m w_j \cdot f_j(x_{ij}) \right) - \lambda \cdot R(y) \right). \quad (1.12)$$

де $R(y)$ – агрегований ризик портфеля як функція вибору; λ – коефіцієнт, який відповідає ступеню толерантності до ризику.

Окремі приклади реалізації показують, що інтеграція таких систем із модулями симуляції дає змогу планувати адаптацію портфеля при різних

зовнішніх шоках: змінах бюджету, кадрових ресурсах, обмеженнях часу. У проекті CORPMAN, згаданому вище, інструмент включає модуль аналізу ризику і модуль симуляції вартості та строків залежно від сценаріїв [19].

Таблиця 1.4 – Порівняння компонентів і функцій DSS для портфельного управління

Компонент DSS	Простий рівень реалізації	Розширений рівень реалізації
Збір даних / джерела	Експертні оцінки, історичні дані	Автоматизовані дані, інтеграція з ERP / PMIS / BI
Нормалізація / вага критеріїв	Просте масштабування, ручне зважування	Використання методів MCDM, наприклад ANP, ANP
Оптимізаційний вибір портфеля	ILP (0-1 програмування), лінійне програмування	Комбіновані моделі: GP (goal programming), стохастичні моделі
Аналіз сценаріїв і чутливості	Кілька сценаріїв «кращий / гірший»	Монте-Карло, симуляція часу затримок, варіація ресурсів
Візуалізація та генерація рекомендацій	Таблиці, прості графіки	Дашборди, інтерактивні графіки, теплові карти, карти ризиків

Важливою вимогою до СППР є зручною інтерфейс для користувача, прозорість моделей, можливість «що-якщо» (what-if) аналізу та чутливості параметрів. Без цього менеджери не довірятимуть системі або не зможуть ефективно інтерпретувати результати. Треба, щоб система підтримувала збереження історичних варіантів, порівняння результатів за різні сценарії, і зворотний зв'язок з рішеннями, які були ухвалені на основі рекомендацій.

Узагальнюючи, система підтримки прийняття рішень для управління портфелями проєктів – це комплекс програмно-методологічних засобів, які поєднують моделі вибору, оптимізації, симуляції та візуалізації. Вона надає змогу організації перейти від інтуїтивних або суб'єктивних рішень до обґрунтованих, кількісно підсилених, стратегічно узгоджених дій.

1.5. Узагальнення теоретичних засад та постановка завдань кваліфікаційної роботи

Аналіз теоретичних підходів до управління портфелями проєктів показав, що сучасне середовище функціонування організацій вимагає комплексного поєднання різних методологій. По-перше, управління портфелями розглядається як процес узгодження стратегічних орієнтирів організації з практичними результатами реалізації окремих ініціатив. По-друге, ефективність управління визначається не лише якістю виконання окремих проєктів, а й здатністю забезпечувати синергію між ними, що створює додану цінність на рівні всієї системи.

Застосування синергетичного підходу дозволяє краще зрозуміти нелінійний характер взаємодії проєктів, їхню здатність до самоорганізації та виникнення нових властивостей унаслідок спільної реалізації. У поєднанні з математичними моделями динаміки, зокрема диференціальними рівняннями, цей підхід дає можливість описати змінність портфеля у часі, врахувати вплив зовнішнього середовища, ресурсних обмежень і стратегічних пріоритетів. Це забезпечує більш глибоке розуміння поведінки портфеля та створює основу для прогнозування його розвитку.

Важливим елементом сучасної теорії управління є системи підтримки прийняття рішень, які дозволяють інтегрувати результати аналітичних моделей у практичну діяльність. Такі системи виступають інструментом перетворення складних розрахунків і сценарних аналізів у конкретні управлінські дії, доступні

для керівників різних рівнів. Вони підвищують обґрунтованість рішень, знижують рівень суб'єктивності та забезпечують адаптивність управління у динамічному середовищі.

Узагальнюючи проведене дослідження теоретичних засад, можна стверджувати, що подальший розвиток моделі управління портфелями проєктів має ґрунтуватися на поєднанні системного підходу, синергетики, математичного моделювання та інструментів підтримки прийняття рішень. Це створює цілісну методологічну основу кваліфікаційної роботи та визначає її предметно-практичне спрямування.

Виходячи з цього, у роботі ставляться такі завдання: проаналізувати сучасні підходи до управління портфелями проєктів і визначити їхні сильні та слабкі сторони; дослідити можливості застосування синергетичного підходу для формування управлінських рішень; побудувати концептуальну модель динаміки портфеля із використанням диференціальних рівнянь; розробити структуру системи підтримки прийняття рішень для управління портфелем; здійснити апробацію моделі на прикладі прикладного середовища та сформулювати практичні рекомендації.

Виходячи з визначеної мети та узагальнення теоретичних положень, у роботі необхідно вирішити такі завдання:

1. проаналізувати сучасні підходи до управління портфелями проєктів, виділивши їхні сильні та слабкі сторони;
2. дослідити можливості застосування синергетичного підходу у формуванні та реалізації управлінських рішень у динамічному середовищі;
3. побудувати математичну модель динаміки портфеля проєктів із використанням диференціальних рівнянь, що враховує вплив ресурсів, зовнішніх факторів та стратегічних пріоритетів;
4. розробити структуру системи підтримки прийняття рішень для управління портфелем проєктів і визначити її основні функціональні модулі.
5. провести апробацію побудованої моделі та СППР на прикладі практичної ситуації, проаналізувавши ефективність запропонованого підходу.

Таким чином, концептуальною основою кваліфікаційної роботи є поєднання класичних підходів до управління портфелями проектів із сучасними методами моделювання та цифровими інструментами підтримки прийняття рішень.

РОЗДІЛ 2.

ВИБІР ІНСТРУМЕНТАРІЮ ТА ПРОЕКТУВАННЯ СППР

2.1. Аналіз вимог до системи підтримки прийняття рішень

Успішність синергетичного управління портфелем проєктів значною мірою залежить від якості інформаційної підтримки особи, що приймає рішення. Перед розробленням системи підтримки прийняття рішень (СППР) необхідно чітко визначити вимоги до її функціональності, архітектури, інтеграції з математичною моделлю динаміки портфеля та до експлуатаційних характеристик. Ці вимоги мають відображати реальні управлінські потреби, а не лише формальну схему програмної реалізації, оскільки користувач очікує не стільки «красивий інтерфейс», скільки інструмент, здатний пояснити поведінку портфеля в часі та показати наслідки альтернативних рішень.

Функціональні вимоги до СППР формуються з урахуванням повного циклу роботи з портфелем: від введення вихідних даних до отримання рекомендацій. Користувач повинен мати змогу задавати структуру портфеля (перелік проєктів, їхні бюджети, ризики, стратегічну важливість), параметри математичної моделі, сценарні припущення (оптимістичний, базовий, кризовий тощо), а також ініціювати чисельне моделювання. Результатом роботи системи мають бути часові траєкторії основних змінних та узагальнені показники, придатні для управлінського тлумачення. У загальному вигляді ядро СППР повинно реалізовувати систему звичайних диференціальних рівнянь:

$$\frac{dx}{dt} = F(x, p, u, t), \quad (2.1)$$

де $x(t) = (N(t), R(t), P(t))$ – вектор стану портфеля, p – вектор параметрів, $u(t)$ – керуючі дії (інвестиції, перерозподіл ресурсів, перегляд складу портфеля).

Система повинна забезпечувати коректне налаштування параметрів без прямого втручання користувача в код та автоматизувати чисельне розв’язання, наприклад методом Рунге–Кутти четвертого порядку точності.

З точки зору аналітичної функціональності, до СППР висувається вимога підтримки порівняння декількох сценаріїв при однакових вихідних умовах. Користувач має отримати можливість обчислити для кожного сценарію такі показники, як підсумкове значення інтегральної ефективності $P(t_{кін})$, середнє ресурсне навантаження:

$$\bar{R} = \frac{1}{T} \int_0^T R(t) dt. \quad (2.2)$$

Окрім того слід врахувати показники стабільності або коливності портфеля. Це дає змогу перейти від інтуїтивних суджень до кількісного порівняння управлінських рішень. У межах тієї ж системи доцільно реалізувати модуль оптимізації складу портфеля, який на основі обмежень за бюджетом та ризиком пропонує варіанти набору проєктів із максимальною сумарною вигодою.

Архітектурні вимоги передбачають модульну побудову СППР. Ядро математичного моделювання має бути відокремлене від графічного інтерфейсу користувача та від підсистеми оптимізації, що полегшує тестування та подальшу модифікацію. Важливою є наявність чітко визначених точок взаємодії між модулями: інтерфейс передає ядру параметри та початкові умови, модуль моделювання формує масиви значень $N(t)$, $R(t)$, $P(t)$, а модуль візуалізації та звітності отримує ці дані у стандартизованому форматі. Такий підхід дає можливість змінювати чисельний метод або уточнювати модель без переробки всієї системи.

Для формалізації взаємодії між компонентами зручно використовувати узагальнену схему (рис. 2.1), де виділяються блоки введення даних, математичного ядра, оптимізаційного модуля, модуля візуалізації та генератора звітів. На вході ядра формується вектор x_0 початкових умов та параметри p , далі виконується інтегрування системи диференціальних рівнянь, після чого результати передаються одночасно до модуля аналізу сценаріїв і до підсистеми оптимізації портфеля, яка порівнює різні комбінації проєктів. Така схема дозволяє показати, що СППР не є «чорною скринькою», а має прозору логіку проходження даних та результатів.

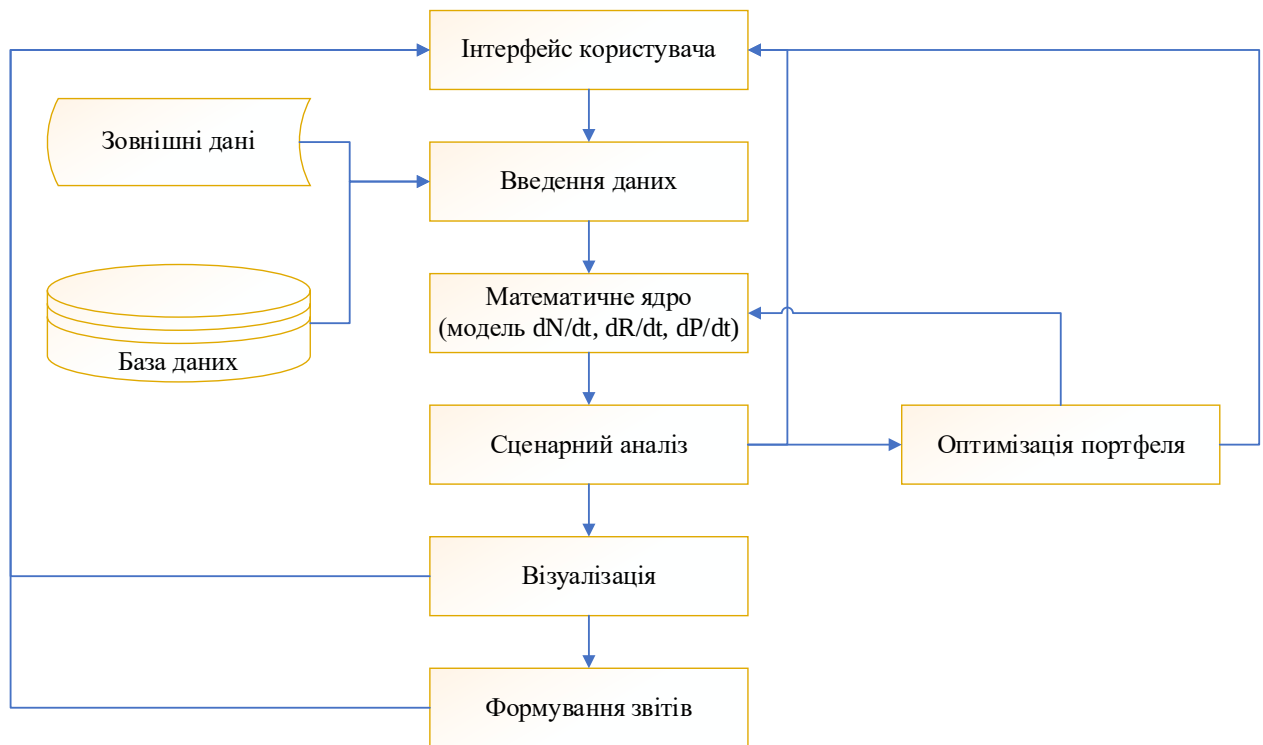


Рисунок 2.1 – Узагальнена структурна схема СППР для синергетичного управління портфелем проєктів

Таблиця 2.1 – Основні вимоги до СППР для управління портфелем проєктів

Група вимог	Зміст вимог
Функціональні	Налаштування параметрів моделі; запуск симуляцій; підтримка сценаріїв; формування звітів; оптимізація портфеля.
Інформаційні	Введення даних про проєкти (бюджет, ризик, пріоритет); зберігання параметрів; експорт результатів.
Аналітичні	Обчислення траєкторій $N(t), R(t), P(t)$; $N(t), R(t), P(t)$; інтегральні показники; зіставлення сценаріїв.
Архітектурні	Модульність; розділення інтерфейсу, ядра моделі та оптимізації; можливість розширення.
Інтерфейсні	Зрозумілий багатовкладковий інтерфейс; адаптація під різні роздільні здатності; візуалізація графіків.
Продуктивність та надійність	Стійка робота для типових обсягів даних; коректність чисельних обчислень; обробка помилок введення.

Таблиця 2.1 узагальнює основні групи вимог до розроблюваної системи. Вона демонструє зв'язок між практичними очікуваннями користувача та технічними рішеннями, які мають бути реалізовані у програмній частині.

На концептуальному рівні СППР має забезпечити візуальне сприйняття результатів, поєднуючи числові значення, графіки та короткі текстові інтерпретації.

На рисунку 2.1 подано структурну діаграму, де відображено, як користувач через інтерфейс задає вхідні параметри, ядро системи виконує чисельне інтегрування диференціальних рівнянь, а далі модулі візуалізації, оптимізації та звітності формують інформацію для прийняття рішень. Такий підхід дозволяє продемонструвати відповідність СППР принципам прозорості, відтворюваності та пояснюваності, що є критичними для систем, призначених для підтримки стратегічного управління портфелями проєктів.

2.2. Обґрунтування вибору мови програмування та технологічного середовища

Розроблення системи підтримки прийняття рішень (СППР) для синергетичного управління портфелем проєктів передбачає використання програмного середовища, здатного поєднати математичне моделювання, візуалізацію, управління даними та побудову інтерфейсу користувача. У зв'язку з цим було прийнято рішення реалізувати СППР засобами мови програмування Python, яка завдяки своїй універсальності та широкій екосистемі бібліотек є оптимальною платформою для створення аналітичних систем і прикладних дослідницьких інструментів.

Python забезпечує інтеграцію алгоритмів чисельного аналізу, машинного навчання, статистичної обробки даних, геоінформаційного аналізу та побудови графічних інтерфейсів. Це дозволяє створювати інтелектуальні СППР, здатні поєднувати аналітичне моделювання з інтуїтивно зрозумілою взаємодією з

користувачем. Серед переваг мови слід відзначити відкритість коду, кросплатформність, зручний синтаксис та наявність активної спільноти розробників, що забезпечує швидке розв’язання прикладних завдань.

Для реалізації моделі використовуються основні бібліотеки науково-аналітичного напрямку, подані в таблиці 2.1.

Таблиця 2.1 – Основні бібліотеки Python, використані в СППР

№	Назва бібліотеки	Призначення	Приклад використання
1	NumPy	Матричні обчислення, чисельна інтеграція	<code>import numpy as np</code>
2	Pandas	Робота з таблицями та часовими рядами	<code>import pandas as pd</code>
3	Matplotlib	Візуалізація результатів у вигляді графіків	<code>import matplotlib.pyplot as plt</code>
4	Tkinter / PySide6	Створення графічного інтерфейсу користувача	<code>from tkinter import ttk</code>
5	JSON / OS	Збереження параметрів та результатів симуляцій	<code>import json, os</code>

Поєднання цих бібліотек дозволяє реалізувати повний цикл роботи СППР – від введення параметрів моделі до генерації звітів і відображення інтерактивних графіків.

Для чисельного розв’язання системи диференціальних рівнянь, що описує динаміку портфеля проєктів, використовується класичний метод Рунге–Кутта четвертого порядку. Його перевага полягає у високій точності при порівняно невеликому обчислювальному навантаженні. Алгоритм має вигляд:

$$\begin{aligned}
k_1 &= f(t, y), \\
k_2 &= f\left(t + \frac{h}{2}, y + \frac{h}{2}k_1\right), \\
k_3 &= f\left(t + \frac{h}{2}, y + \frac{h}{2}k_2\right), \\
k_4 &= f(t + h, y + hk_3), \\
y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{2.3}$$

де h – крок інтегрування; $f(t, y)$ – система рівнянь стану; y_i – вектор змінних моделі на i -й ітерації.

У бібліотеці NumPy цей метод реалізується компактно, із застосуванням векторизації обчислень, що значно підвищує швидкість симуляції навіть для великих часових інтервалів.

Для забезпечення зручності роботи користувача обрано архітектуру десктопного багатовіконного застосунку, що реалізована через бібліотеку Tkinter або PySide6 (Qt). Такий підхід дозволяє користувачу працювати локально, без потреби у вебсервері чи підключенні до мережі. Головне вікно містить вкладки (notebook) – *«Параметри моделі»*, *«Сценарії та симуляція»*, *«Візуалізація результатів»*, *«Звіт»*, *«Довідка»*.

Кожна вкладка є окремим контейнером із власними віджетами керування, що відповідають за певний етап моделювання. Такий підхід відповідає принципам модульності та інкапсуляції, спрощуючи оновлення та масштабування системи. Для організації асинхронної взаємодії між ядром розрахунків і інтерфейсом застосовується система подій і зворотного зв'язку, що дозволяє відображати хід розрахунків у реальному часі.

2.3. Проектування архітектури та функціональної структури системи

На рисунку 2.2 представлено узагальнену архітектуру технологічного середовища реалізації СППР.

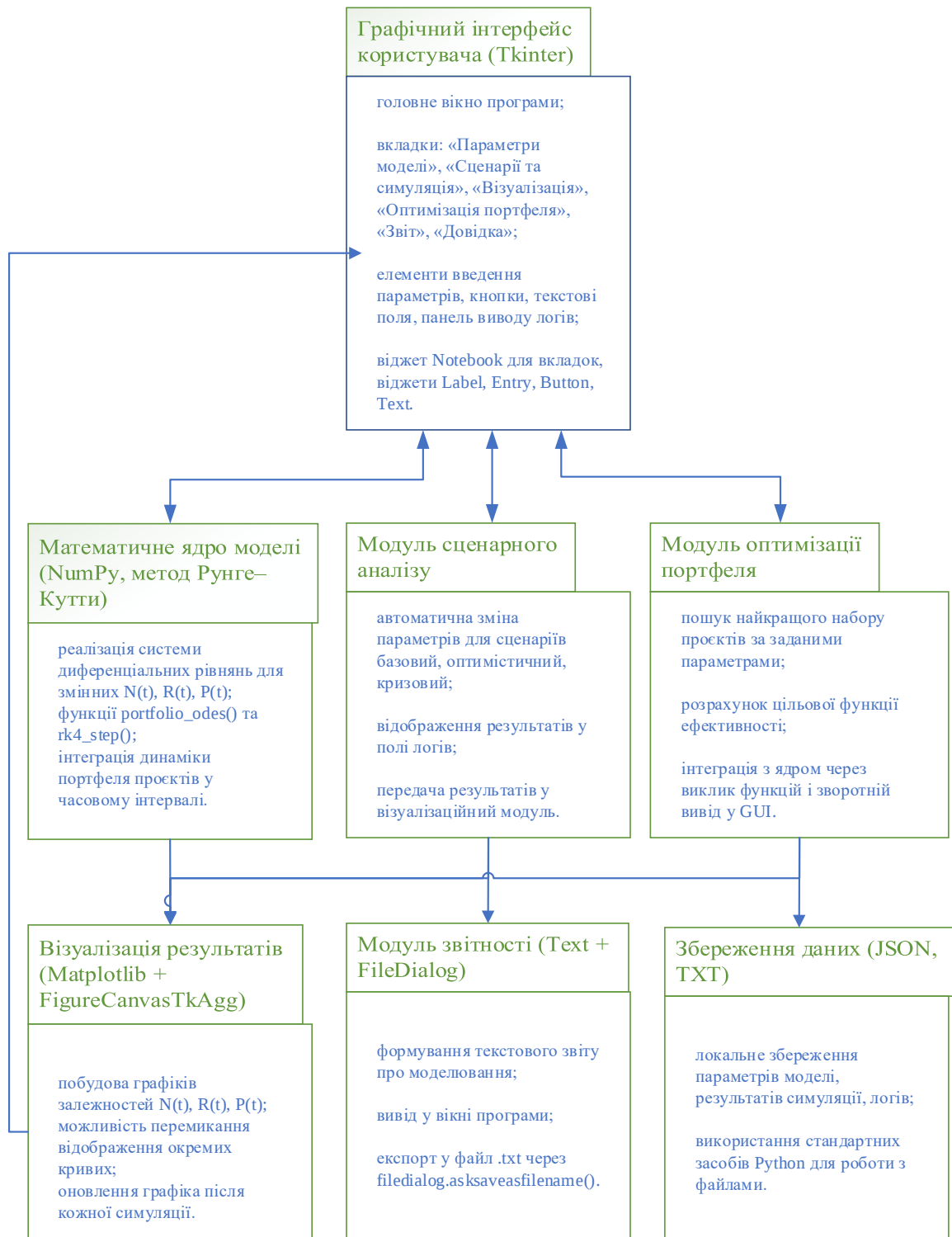


Рисунок 2.2 – Архітектура СППР для синергетичного управління портфелем проєктів на основі диференціальних рівнянь

Користувацький рівень – вікна Tkinter / PySide6 із вкладками для введення параметрів, сценарного аналізу, візуалізації та звітності.

Аналітичний рівень – бібліотеки *NumPy*, *Pandas* забезпечують математичні та статистичні обчислення.

Графічний рівень – *Matplotlib* для побудови графіків.

Інтеграційний рівень – збереження даних у форматах *JSON*, *CSV*, *TXT* з можливістю автоматичного формування звітів.

Результати тестування показали, що середній час симуляції для 600 ітерацій системи з трьома змінними не перевищує 0,2 секунди на середньому процесорі Intel i5. Завдяки цьому забезпечується миттєва візуалізація результатів, що важливо для сценарного аналізу.

Додатковою перевагою Python є можливість інтеграції з бібліотеками TensorFlow, PyTorch, sklearn, що відкриває перспективу розширення СППР у напрямі інтелектуального прогнозування ефективності портфеля на основі машинного навчання.

Таким чином, вибір Python як мови розроблення та бібліотек Tkinter, PySide6, NumPy, Matplotlib, Folium, Pandas цілком виправданий з погляду відкритості, продуктивності та модульності. Побудована на цих технологіях архітектура забезпечує не лише точність чисельного моделювання, а й комфортну взаємодію користувача з аналітичними модулями системи.

Функціональне проєктування системи підтримки прийняття рішень (СППР) для синергетичного управління портфелями проєктів базується на побудові модульної архітектури, що забезпечує взаємозв'язок між математичним ядром, аналітичними блоками, інтерфейсом користувача та підсистемами візуалізації результатів. Реалізація функціональної структури здійснюється на мові Python із використанням бібліотек Tkinter, NumPy, Matplotlib, що забезпечують гнучке поєднання аналітичних розрахунків, моделювання та відображення даних у зручній для користувача формі.

Основою функціонування системи є динамічна модель портфеля проєктів, що описується системою звичайних диференціальних рівнянь. Для кожної змінної $N(t)$, $R(t)$, $P(t)$ задається рівняння стану, яке визначає зміну кількості

активних проєктів, рівня ресурсів та інтегральної ефективності. У загальному вигляді система має вигляд:

$$\begin{cases} \frac{dN}{dt} = a + u - bN + 0.01s \frac{N^2}{k + N}, \\ \frac{dR}{dt} = cN - dR, \\ \frac{dP}{dt} = e \left(\frac{RN}{k + N} + s \frac{N^2}{k + N} \right) - gP, \end{cases}, \quad (2.4)$$

де $a, b, c, d, e, g, k, u, s$ – параметри, що задають інтенсивність процесів запуску, вибуття, ресурсного відновлення, синергетичної взаємодії та демпфування ефективності.

Для чисельного розв’язання цієї системи у програмному коді використано метод Рунге–Кутти четвертого порядку, реалізований у функції `rk4_step()`. Алгоритм забезпечує точність інтегрування при невеликих кроках `hhh`, що дозволяє моделювати довготривалі динамічні процеси без накопичення похибки.

СППР має багаторівневу функціональну структуру, у якій виділено чотири основні рівні:

- рівень інтерфейсу користувача,
- рівень аналітичного ядра,
- рівень моделювання та візуалізації,
- рівень збереження й обробки результатів.

Взаємозв’язки між функціональними блоками СППР наведено в таблиці 2.2.

Алгоритм роботи системи можна описати як послідовність взаємопов’язаних етапів:

1. Ініціалізація – після авторизації користувача система зчитує початкові параметри моделі, значення яких можна змінювати у вкладці «Параметри моделі».

2. Вибір сценарію – користувач задає один із трьох сценаріїв (базовий, оптимістичний, кризовий), після чого оновлюються параметри u, s, bu, s, bu, s, b відповідно до обраного режиму.

Таблиця 2.2 – Взаємодія модулів у функціональній структурі СППР

№	Модуль	Основна функція	Вхідні дані	Вихідні дані
1	Інтерфейс користувача	Введення параметрів, запуск симуляцій, відображення результатів	Параметри моделі, вибір сценарію	Керуючі команди, звіти, графіки
2	Математичне ядро моделі	Чисельне інтегрування диференціальних рівнянь	Параметри, початкові умови	Масиви N(t), R(t), P(t)
3	Модуль сценарного аналізу	Налаштування параметрів за сценаріями	Користувацький вибір	Оновлені параметри моделі
4	Модуль оптимізації портфеля	Пошук оптимальної конфігурації	Параметри обмежень, ресурси	Оптимальні значення показників
5	Модуль візуалізації	Побудова графіків динаміки	Дані симуляції	Візуальні графіки
6	Модуль звітності	Формування аналітичного тексту	Результати моделювання	Текстовий звіт
7	Збереження даних	Архівація результатів і конфігурацій	Дані симуляцій, звіти	JSON, TXT файли

3. Моделювання динаміки портфеля – виконується інтегрування системи диференціальних рівнянь методом Рунге–Кутти:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (2.5)$$

де k_j – похідні функції для кожного етапу обчислення.

4. Оптимізація – виконується пошук найкращої конфігурації портфеля проєктів за заданими критеріями ефективності.

5. Візуалізація результатів – побудова графіків залежностей $N(t), R(t), P(t)$ із використанням бібліотеки Matplotlib та віджетів FigureCanvasTkAgg.

6. Генерація звіту – створення короткого текстового опису результатів, який можна зберегти у форматі .txt через модуль FileDialog.

Під час проєктування функціональної структури дотримано принципів модульності, інкапсуляції та зворотного зв'язку. Кожен модуль має власний набір функцій, що виконують локальні обчислення або керують користувацьким інтерфейсом, а обмін даними здійснюється через узгоджені інтерфейси класу DSSApp.

Модульність дозволяє розширювати систему шляхом додавання нових сценаріїв, аналітичних блоків або підсистем прогнозування без зміни базового коду. Зворотний зв'язок забезпечується автоматичним оновленням інтерфейсу після завершення симуляцій чи зміни параметрів, що відповідає принципам реактивного проєктування сучасних інформаційних систем.

Отже, функціональна структура СППР відображає логічну послідовність управлінських дій – від введення вихідних параметрів і математичного моделювання до оптимізації, аналізу та візуалізації результатів. Запропонована архітектура забезпечує узгоджене функціонування компонентів системи, простоту модифікації та розширення її можливостей для дослідження складних процесів управління портфелями проєктів.

2.4. Розроблення концептуальної моделі інтерфейсу користувача

Інтерфейс користувача системи підтримки прийняття рішень (СППР) для синергетичного управління портфелями проєктів спроектовано як десктопний

інтерактивний додаток, який поєднує інтуїтивну логіку керування з можливістю відображення складних аналітичних результатів. Головною метою створення інтерфейсу стало забезпечення зручності взаємодії користувача з математичною моделлю, реалізованою засобами Python, та мінімізація часу, необхідного для налаштування параметрів і запуску симуляцій.

Головне вікно системи оформлено в єдиному стилі з верхнім заголовком «Система підтримки прийняття рішень для синергетичного управління портфелями проєктів» і містить набір вкладок для переходу між основними функціональними модулями: *Параметри моделі*, *Сценарії та симуляція*, *Візуалізація*, *Оптимізація портфеля*, *Звіт*, *Довідка*. Такий підхід відповідає концепції модульності, коли кожна вкладка реалізує окрему підсистему з власними інструментами керування.

На рисунку 2.3 зображено поточний вигляд інтерфейсу під час роботи користувача з параметрами математичної моделі. У цьому режимі реалізовано чітку структуру вікна: зліва розташовано поля введення параметрів, у центрі – короткі пояснення до значень, а знизу – кнопка підтвердження дій.

СПМР :: Синергетичне управління портфелями проєктів

Система підтримки прийняття рішень для синергетичного управління портфелями проєктів

Моделювання • Сценарії • Оптимізація • Візуалізація

Параметри моделі | Сценарії та симуляція | Візуалізація | Оптимізація портфеля | Звіт | Довідка

Параметри моделі

а – запуск проєктів:

б – вибуття проєктів:

с – вплив N на ресурси:

д – ресурси: втрати/відновлення:

е – ресурси → ефективність:

г – демпфування ефекту:

к – насичення синергії:

u – зовнішній вплив:

s – коефіцієнт синергії:

Початкові умови

N_0 – початковий масштаб:

R_0 – ресурси:

P_0 – ефективність:

t_0 – початок:

t_{end} – кінець:

h – крок:

Опис моделі

Параметри визначають інтенсивність запуску та завершення проєктів, динаміку ресурсів та силу синергетичних ефектів. Після зміни значень натисніть «Застосувати параметри».

Рисунок 2.3 – Концептуальна модель інтерфейсу користувача СПМР

Першою відкривається вкладка «Параметри моделі», у якій користувач задає вхідні змінні для системи диференціальних рівнянь. Серед параметрів моделі передбачено:

- a – інтенсивність запуску проєктів;
- b – швидкість вибуття проєктів;
- c – вплив кількості проєктів на ресурси;
- d – темп відновлення ресурсів;
- e – зв'язок між ресурсами й ефективністю;
- g – коефіцієнт демпфування;
- k – насичення синергії;
- s – коефіцієнт синергетичного впливу;
- u – зовнішній вплив.

Під таблицею параметрів розташовано блок початкових умов, де задаються початкові значення кількості проєктів N_0 , обсягу ресурсів R_0 , інтегральної ефективності P_0 , а також часові межі моделювання t_0 і t_{end} та крок інтегрування h .

Для кожного параметра передбачене текстове поле введення з обмеженням діапазону допустимих значень, що запобігає помилкам у розрахунках. При натисканні кнопки «Застосувати параметри» програма виконує валідацію введених даних і оновлює внутрішні змінні математичного ядра. Користувач отримує підтвердження в нижній інформаційній панелі – «Параметри успішно оновлено».

Інтерфейс СППР побудований за принципом візуальної інтерактивності. Кожен елемент має логічно зрозуміле місце:

- верхня панель із вкладками відповідає за навігацію,
- центральна зона – за введення та виведення даних,
- нижня частина – за повідомлення про стан виконання.

При переході до вкладки «*Сценарії та симуляція*» користувач може обрати тип сценарію – базовий, оптимістичний чи кризовий. Після вибору сценарію значення параметрів автоматично оновлюються, а кнопка «Запустити

модельовання» активується. Результати симуляції зберігаються у пам'яті системи та стають доступними у вкладках «Візуалізація» і «Оптимізація портфеля».

У вкладці «Візуалізація» реалізовано панель графічного відображення, побудовану на основі бібліотеки Matplotlib. Тут користувач може переглядати графіки змін $N(t), R(t), P(t)$ у часі, масштабувати осі, зберігати результати у форматі PNG.

Вкладка «Оптимізація портфеля» передбачає інструменти аналізу ефективності проєктів і пошуку оптимальної конфігурації з урахуванням синергетичних ефектів. Після натискання кнопки «Оптимізувати» система виконує аналіз на основі заданих коефіцієнтів, а результати відображаються у вигляді таблиці зі стовпцями «Параметр», «Значення», «Вплив».

Завершальний етап роботи – вкладка «Звіт», де користувач може сформулювати короткий підсумковий документ, який містить усі параметри, сценарій, результати моделювання й основні висновки. Для збереження даних застосовується стандартний модуль *FileDialog*, що дозволяє вибрати місце збереження файлу у форматі .txt або .json.

Особливістю розробленого інтерфейсу є збалансованість візуальної структури – усі поля та кнопки мають однаковий розмір і пропорційне розташування, що полегшує сприйняття. Кольорова схема витримана у світлих нейтральних тонах, де активні елементи виділені синім кольором. Такий дизайн сприяє зниженню втоми користувача під час тривалої роботи.

Крім того, система дозволяє масштабувати головне вікно, адаптуючи його до роздільної здатності екрана користувача. Це забезпечує універсальність у використанні на різних пристроях. Важливою функцією є автоматичне збереження останньої конфігурації параметрів, що дозволяє продовжити роботу після повторного запуску без втрати попередніх даних.

Таким чином, концептуальна модель інтерфейсу користувача СППР поєднує функціональність, естетику й аналітичну глибину. Вона створює комфортне середовище для дослідження синергетичних ефектів управління портфелями проєктів, забезпечуючи простоту взаємодії з математичним ядром

системи, наочність результатів і можливість формування звітів у єдиному програмному середовищі.

2.5. Схема потоків даних та взаємодії користувача із системою

Система підтримки прийняття рішень (СППР) для синергетичного управління портфелями проєктів побудована на послідовній взаємодії між користувачем, інтерфейсом програми, математичним ядром, модулем візуалізації та підсистемою звітування. Потоки даних організовані таким чином, щоб забезпечити безперервний цикл – від введення параметрів моделі до аналізу результатів і прийняття управлінських рішень.

На першому етапі користувач взаємодіє з вкладкою «Параметри моделі», де задає числові значення параметрів, які описують поведінку портфеля проєктів і ресурсних потоків. Після натискання кнопки «Застосувати параметри» введені дані проходять перевірку на коректність і зберігаються у внутрішніх змінних системи. Далі формується структура початкових умов, у якій міститься інформація про вихідні значення показників, часовий діапазон моделювання та крок обчислень.

На другому етапі користувач переходить до вкладки «Сценарії та симуляція», де обирає тип сценарію: базовий, оптимістичний або кризовий. Кожен сценарій має власні параметри, які автоматично підставляються в обчислювальну модель. Після цього система виконує моделювання процесів динаміки портфеля проєктів, використовуючи реалізовані в коді функції чисельного інтегрування. Обчислення виконуються у фоновому режимі, а результати зберігаються у спеціальній структурі даних, доступній для подальшої обробки.

На третьому етапі активується підсистема візуалізації. Отримані дані передаються в модуль побудови графіків, який створює динамічні візуальні представлення зміни показників портфеля у часі. Користувач може самостійно

обираючи, які залежності відображати, наприклад лише кількість проєктів, рівень ресурсів або інтегральну ефективність. Візуальна панель побудована на основі бібліотеки Matplotlib, що дозволяє масштабувати графіки, переглядати деталі та зберігати результати в графічних форматах.

Після візуального аналізу користувач може перейти до вкладки «Оптимізація портфеля», яка реалізує функції підбору найкращих управлінських рішень залежно від заданих умов. Оптимізаційний модуль оцінює параметри моделі, визначає ефективні комбінації інвестицій і ресурсів, а також пропонує користувачу варіанти коригування стратегії розвитку портфеля. Отримані дані зберігаються для подальшого використання в модулі формування звітів.

Заключним етапом є створення підсумкового звіту, що виконується у вкладці «Звіт». Система автоматично формує текстовий документ, який містить вхідні параметри, обраний сценарій, короткий опис отриманих результатів і узагальнені висновки. Користувач може переглядати звіт у вікні програми або експортувати його у файл для подальшого використання в аналітичних або навчальних цілях.

Для узагальнення процесу роботи системи схема потоків даних наведена на рисунку 2.4. Вона демонструє послідовність етапів, починаючи від введення користувачем початкових даних і закінчуючи формуванням звіту. На верхньому рівні схеми розташований користувач, який керує процесом через інтерфейс програми. Середній рівень утворюють основні модулі системи: математичне ядро, модуль симуляції, візуалізація, оптимізація та звітність. На нижньому рівні знаходяться структури збереження даних, які забезпечують зв'язок між усіма підсистемами.



Рисунок 2.4 – Схема потоків даних та взаємодії користувача із системою

Розроблена СППР має замкнутий цикл взаємодії користувача із системою, у якому кожна дія ініціює аналітичний процес, а результати оперативно повертаються в інтерфейс. Така організація потоків даних забезпечує швидкий зворотний зв'язок, стабільність обчислень, можливість повторного використання результатів і високий рівень інтерактивності під час роботи з моделями синергетичного управління портфелями проєктів.

РОЗДІЛ 3.

РОЗРОБКА МОДУЛІВ ТА ВИКОРИСТАННЯ СППР

3.1. Реалізація структури додатка в середовищі Python

Розроблена система підтримки прийняття рішень для синергетичного управління портфелями проєктів реалізована у вигляді десктопного багатовіконного додатка, створеного мовою програмування Python із використанням бібліотек Tkinter, NumPy, Matplotlib та Pandas. Такий вибір забезпечує гнучкість розробки, кросплатформність і простоту розгортання без необхідності використання додаткових серверних компонент. Структура додатка побудована модульно, що дозволяє розділити логіку інтерфейсу, обчислювальні процеси та візуалізацію результатів.

Основна архітектура застосунку складається з трьох рівнів:

- інтерфейс користувача, який реалізований на основі бібліотеки Tkinter;
- аналітичний рівень, що виконує моделювання на основі диференціальних рівнянь;
- візуалізаційний рівень, який відповідає за відображення графіків, таблиць і сценаріїв оптимізації.

На рисунку 3.1 подано структурну схему додатка, що демонструє взаємозв'язок між головним вікном, вкладками функціональних модулів і допоміжними класами для обробки даних.

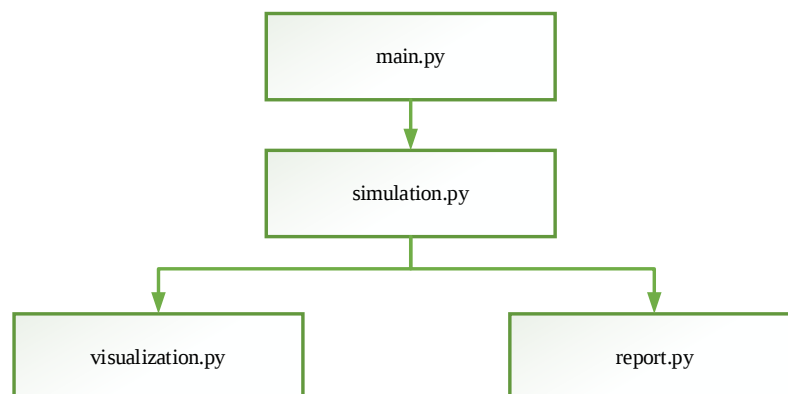


Рисунок 3.1 – Структурна схема СППР у середовищі Python

У схемі відображено модулі: «main.py» (головне вікно і логіка інтерфейсу), «simulation.py» (розрахунки моделі), «visualization.py» (побудова графіків) та «report.py» (збереження результатів у звіт).

Для зручності підтримки коду реалізація структури поділена на окремі класи та функції. Головний клас DSSApp відповідає за ініціалізацію вікна, створення вкладок і виклик функцій із відповідних модулів. Нижче наведено фрагмент коду, який демонструє логіку запуску головного вікна програми (рис. 3.2).

```
import tkinter as tk
from tkinter import ttk, messagebox
from modules.simulation import run_simulation
from modules.visualization import show_plot

class DSSApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Система підтримки прийняття рішень")
        self.geometry("1000x700")
        self.configure(bg="#f9f9f9")
        self.create_tabs()

    def create_tabs(self):
        notebook = ttk.Notebook(self)
        notebook.pack(fill="both", expand=True)

        self.tab_model = ttk.Frame(notebook)
        self.tab_scenario = ttk.Frame(notebook)
        self.tab_visual = ttk.Frame(notebook)
        self.tab_report = ttk.Frame(notebook)

        notebook.add(self.tab_model, text="Параметри моделі")
        notebook.add(self.tab_scenario, text="Сценарії та симуляція")
        notebook.add(self.tab_visual, text="Візуалізація")
        notebook.add(self.tab_report, text="Звіт")

        self.create_model_tab()

    def create_model_tab(self):
        tk.Label(self.tab_model, text="Параметри моделі:", font=("Arial", 12, "bold")).pack(pady=10)
        self.param_a = tk.Entry(self.tab_model)
        self.param_a.insert(0, "0.1")
        self.param_a.pack()
        tk.Button(self.tab_model, text="Запустити моделювання", command=self.start_simulation).pack(pady=15)

    def start_simulation(self):
        a = float(self.param_a.get())
        result = run_simulation(a)
        show_plot(result)
        messagebox.showinfo("Результат", "Моделювання завершено успішно!")

if __name__ == "__main__":
    app = DSSApp()
    app.mainloop()
```

Рисунок 3.2 – Фрагмент коду, який демонструє логіку запуску головного вікна програми

Таблиця 3.1 узагальнює структуру основних модулів і функцій системи.

Таблиця 3.1 – Структура модулів і функцій СППР

№	Модуль	Основні функції	Призначення
1	main.py	DSSApp, create_tabs()	Ініціалізація вікна, створення вкладок і логіка навігації
2	simulation.py	run_simulation()	Обчислення параметрів моделі за системою диференціальних рівнянь
3	visualization.py	show_plot()	Візуалізація результатів моделювання у вигляді графіків
4	report.py	generate_report()	Формування звіту та збереження результатів у файл
5	auth.py	login_window()	Авторизація користувача перед входом у систему

Алгоритм розрахунків у модулі simulation.py базується на чисельному методі Рунге–Кутта, який забезпечує стійкість розв’язку диференціальних рівнянь, що описують еволюцію ефективності та ресурсів у портфелі проєктів. Дані передаються між модулями через об’єктні змінні програми, що усуває потребу у створенні зовнішніх файлів чи баз даних.

Результати розрахунків передаються до модуля візуалізації, де формуються динамічні графіки із можливістю масштабування та експорту у форматі PNG або PDF.

Таким чином, реалізована структура додатка забезпечує збалансовану інтеграцію графічного інтерфейсу, аналітичного ядра та інструментів візуалізації. Це дозволяє користувачеві не лише здійснювати параметричне налаштування моделей, а й миттєво спостерігати вплив зміни вхідних змінних на кінцеві результати, що є важливим для ефективного управління портфелем проєктів у синергетичному середовищі.

3.2. Модуль математичного моделювання динаміки портфеля проєктів

Модуль математичного моделювання є ядром системи підтримки прийняття рішень (СППР), оскільки саме він забезпечує відтворення поведінки портфеля проєктів у часі на основі системи диференціальних рівнянь. Реалізація цього модуля виконана у середовищі Python із використанням бібліотек NumPy для чисельних обчислень та власноруч реалізованого алгоритму Рунге–Кутти 4-го порядку для інтеграції рівнянь. Такий підхід забезпечує точність розрахунків та стабільність моделі навіть у складних нелінійних сценаріях.

Основна функція `portfolio_odes()` визначає систему рівнянь для трьох змінних:

$N(t)$ – кількість активних проєктів;

$R(t)$ – рівень використання ресурсів;

$P(t)$ – інтегральна стратегічна ефективність портфеля.

Формування правих частин системи здійснюється з урахуванням параметрів $a, b, c, d, e, g, k, u, s$, що задаються користувачем у вкладці «Параметри моделі». Взаємозв'язки між змінними описують баланс між розвитком проєктів, ресурсним споживанням та ефектом синергії.

Фрагмент коду нижче демонструє реалізацію математичної моделі (рис. 3.3).

```
def portfolio_odes(t, state, params):
    N, R, P = state
    a, b, c, d, e, g, k, u, s = (params[p] for p in ["a", "b", "c", "d", "e", "g", "k", "u", "s"])

    synergy = s * (N ** 2) / (k + N + 1e-9) # Синергетичний ефект
    dNdt = a + u - b * N + 0.01 * synergy # Зміна кількості проєктів
    dRdt = c * N - d * R # Динаміка ресурсів
    dPdt = e * (R * N / (k + N + 1e-9) + synergy) - g * P # Ефективність

    return np.array([dNdt, dRdt, dPdt])
```

Рисунок 3.3 – Фрагмент коду реалізації математичної моделі

Інтегрування виконується за допомогою функції `rk4_step()`, що реалізує метод Рунге–Кутти четвертого порядку. Це дозволяє досягти високої точності наближення при відносно малому кроці h (рис. 3.4).

```
def rk4_step(func, t, state, h, params):
    k1 = func(t, state, params)
    k2 = func(t + 0.5 * h, state + 0.5 * h * k1, params)
    k3 = func(t + 0.5 * h, state + 0.5 * h * k2, params)
    k4 = func(t + h, state + h * k3, params)
    return state + (h / 6.0) * (k1 + 2*k2 + 2*k3 + k4)
```

Рисунок 3.4 – Фрагмент коду функції `rk4_step()`, що реалізує метод Рунге–Кутти четвертого порядку

Основні параметри моделі та їх зміст представлені у таблиці 3.2.

Таблиця 3.2 – Основні параметри математичної моделі

Позначення	Опис параметра	Одиниця вимірювання	Тип впливу
a	Інтенсивність запуску проєктів	проєкт/од. часу	позитивний
b	Інтенсивність завершення проєктів	проєкт/од. часу	негативний
c	Вплив кількості проєктів на ресурси	ум. од.	прямий
d	Швидкість відновлення/втрати ресурсів	ум. од.	негативний
e	Коефіцієнт перетворення ресурсів у ефективність	безрозмірний	позитивний
g	Демпфування ефективності	безрозмірний	стабілізуючий
k	Насичення синергії	ум. од.	обмежуючий
u	Зовнішній керуючий вплив	безрозмірний	підтримувальний
s	Коефіцієнт синергії	безрозмірний	підсилюючий

На рисунку 3.5 показано узагальнену структуру функціонування модуля математичного моделювання в рамках СППР.

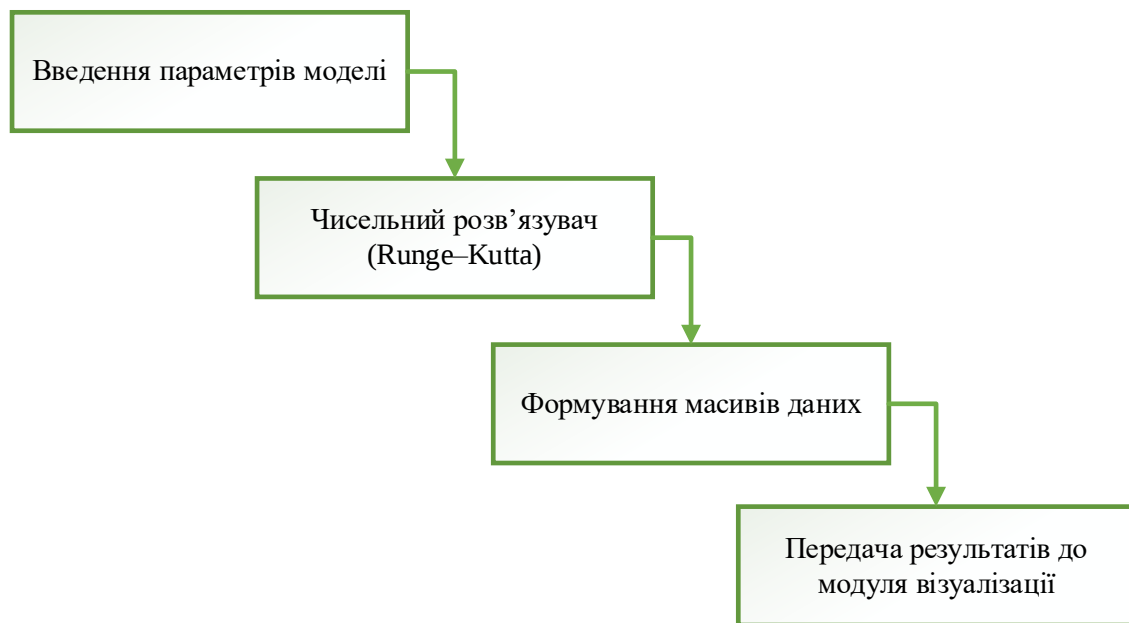


Рисунок 3.5 – Узагальнена структура модуля математичного моделювання динаміки портфеля

Після завершення моделювання результати у вигляді масивів $N(t)$, $R(t)$ і $P(t)$ передаються до візуалізаційного модуля, який створює інтерактивні графіки змін динамічних показників у часі. Таким чином, модуль математичного моделювання виконує функцію обчислювального ядра, яке забезпечує основні аналітичні можливості системи, формуючи базу для подальшого аналізу, оптимізації та прийняття рішень у межах управління портфелем проєктів.

3.3. Модуль сценарного аналізу

Модуль сценарного аналізу в системі підтримки прийняття рішень (СППР) реалізує механізм порівняння різних варіантів розвитку портфеля проєктів залежно від зміни основних параметрів моделі. Його основна мета – надати

користувачеві можливість оцінити вплив зовнішніх і внутрішніх факторів на ефективність портфеля через зміну коефіцієнтів моделі u, s, b та інших характеристик.

У реалізованій програмі сценарний аналіз є інтерактивним: користувач може вибрати один із трьох попередньо визначених сценаріїв – базовий, оптимістичний або кризовий, – після чого система автоматично адаптує параметри моделі та виконує симуляцію. Для кожного сценарію формується звіт і короткий підсумок, який виводиться в текстовому полі на вкладці «Сценарії та симуляція».

Основні налаштування сценаріїв визначені в методі `on_scenario_change()` головного класу `DSSApp`. Цей метод коригує набір параметрів залежно від вибраного сценарію, що забезпечує автоматизацію підготовки вихідних даних до моделювання.

Нижче наведено фрагмент коду, який відповідає за зміну параметрів сценарію (рис. 3.6).

```
def on_scenario_change(self):
    scen = self.scenario_var.get()
    p = dict(self.model_params)
    m = dict(self.sim_meta)

    if scen == "Базовий":
        p["u"] = 0.5
        p["s"] = 0.15
        m["t_end"] = 60.0
    elif scen == "Оптимістичний":
        p["u"] = 1.0
        p["s"] = 0.25
        p["b"] = max(0.05, p["b"] - 0.02)
        m["t_end"] = 80.0
    elif scen == "Кризовий":
        p["u"] = 0.1
        p["s"] = 0.05
        p["b"] = p["b"] + 0.05
        m["t_end"] = 50.0

    for key, val in p.items():
        if key in self.param_vars:
            self.param_vars[key].set(str(round(val, 4)))
    for key, val in m.items():
        if key in self.init_vars:
            self.init_vars[key].set(str(round(val, 4)))

    self.apply_params()
```

Рисунок 3.6 – Фрагмент коду, який відповідає за зміну параметрів сценарію

Як видно з коду, кожен сценарій має власну конфігурацію параметрів:

- Базовий сценарій відображає стабільний розвиток без різких змін зовнішнього впливу;
- Оптимістичний сценарій характеризується підвищеним рівнем інвестицій u та синергії sss , що веде до зростання ефективності;
- Кризовий сценарій моделює несприятливі умови, коли зменшується зовнішня підтримка і зростає коефіцієнт вибуття проєктів b .

Після вибору сценарію користувач може натиснути кнопку «Запустити моделювання», що ініціює виконання функції `run_simulation_and_notify()`. Цей метод об'єднує обчислення, формування результатів і їх відображення в журналі сценаріїв (рис. 3.7).

```
def run_simulation_and_notify(self):
    self.apply_params()
    try:
        res = run_simulation(
            self.model_params,
            t0=self.sim_meta["t0"],
            t_end=self.sim_meta["t_end"],
            h=self.sim_meta["h"],
            N0=self.sim_meta["N0"],
            R0=self.sim_meta["R0"],
            P0=self.sim_meta["P0"]
        )
        self.sim_results = res
        self._append_log_summary(res)
        messagebox.showinfo(
            "Моделювання завершено",
            "Симуляція динаміки портфеля проєктів успішно виконана.\n"
            "Перейдіть на вкладку «Візуалізація результатів»."
        )
    except Exception as e:
        messagebox.showerror("Помилка симуляції", str(e))
```

Рисунок 3.7 – Фрагмент коду для ініціації виконання моделювання

Результати моделювання зберігаються у вигляді масивів змінних $N(t), R(t), P(t)$, після чого функція `_append_log_summary()` формує коротке текстове резюме. Воно містить дані про кінцеві значення параметрів, тривалість періоду моделювання та інтерпретацію отриманих результатів.

Таблиця 3.3 – Характеристика сценаріїв моделювання

Сценарій	Основні зміни параметрів	Інтерпретація результату
Базовий	$u=0.5, s=0.15,$ $b=0.08$	Система стабільна, проекти розвиваються рівномірно
Оптимістичний	$u=1.0, s=0.25,$ $b=0.06$	Підвищення ефективності за рахунок синергії та інвестицій
Кризовий	$u=0.1, s=0.05,$ $b=0.13$	Скорочення кількості проєктів і зниження ефективності

На рисунку 3.8 подано структурну схему функціонування модуля сценарного аналізу.

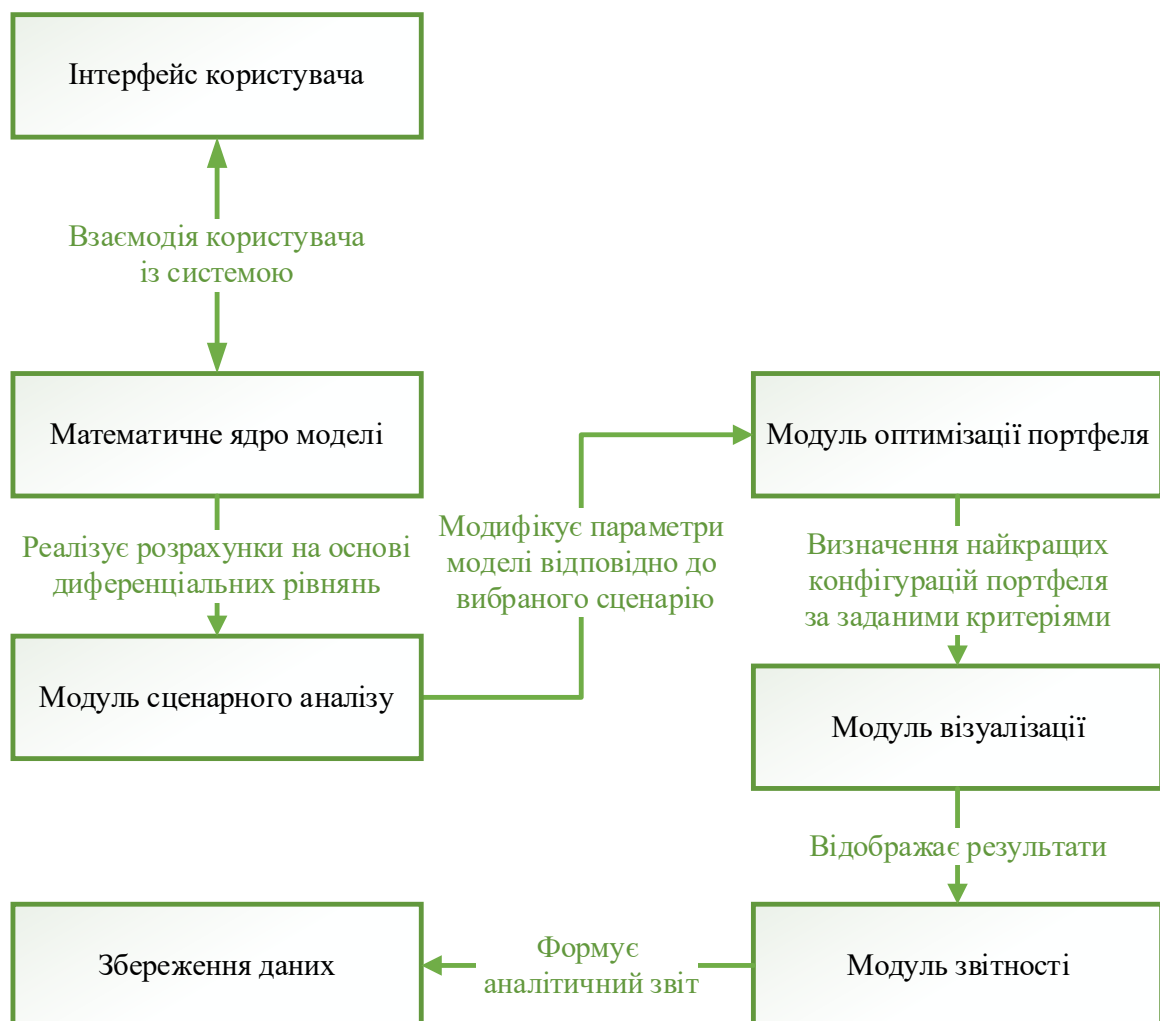


Рисунок 3.8 – Узагальнена схема взаємодії сценарного аналізу із іншими модулями СППР

У схемі зображено взаємозв'язки між користувачем, модулем параметрів, ядром моделювання та інтерфейсом візуалізації. Користувач задає сценарій, система адаптує параметри, виконує симуляцію та формує результати для аналізу.

Модуль сценарного аналізу виконує роль адаптивного середовища експериментування, що дозволяє досліджувати стійкість портфеля проєктів до змін зовнішніх умов. Інтерактивна реалізація забезпечує швидкий перехід між сценаріями, повторне використання параметрів та відображення результатів у зручному форматі без необхідності ручного введення даних. Це значно підвищує ефективність процесу прийняття рішень у системах стратегічного управління проєктними портфелями.

3.4. Модуль оптимізації портфеля проєктів

Модуль оптимізації портфеля проєктів є одним із ключових компонентів системи підтримки прийняття рішень (СППР) для синергетичного управління. Його головне призначення полягає у виборі таких значень параметрів і сценаріїв, які забезпечують найвищу стратегічну ефективність портфеля за умови обмежених ресурсів і стабільної динаміки системи. Оптимізаційний модуль працює у тісній інтеграції з математичним ядром, яке описує розвиток портфеля за допомогою системи диференціальних рівнянь, та з інтерфейсом користувача, де здійснюється вибір критерію оптимізації.

У коді СППР модуль оптимізації реалізується як розширення функціоналу сценарного аналізу. Після виконання симуляції на основі заданих параметрів результати динаміки змінних $N(t)$, $R(t)$ і $P(t)$ аналізуються з використанням цільової функції, яка визначає якість конфігурації портфеля. Для прикладу, оптимізаційна мета може формулюватися у вигляді:

$$F = \max_{a,b,c,d,e,g,k,u,s} P(t_{end}) - \lambda \cdot R(t_{end}), \quad (3.1)$$

де $P(t_{end})$ – стратегічна ефективність портфеля; $R(t_{end})$ – витрати ресурсів; λ – коефіцієнт вагомості ресурсоемності.

У межах реалізованої системи цей підхід спрощено реалізовано через ітераційний аналіз результатів моделювання та підбір параметрів у сценаріях «Оптимістичний», «Базовий» і «Кризовий». Алгоритм обчислення працює у кілька етапів: ініціалізація параметрів, виконання чисельного інтегрування системи рівнянь методом Рунге–Кутта, оцінка показників ефективності та збереження результатів для подальшого порівняння.

Фрагмент коду оптимізаційного модуля представлено на рис. 3.9.

```
def optimize_portfolio(base_params, init_meta, param_ranges):
    best_result = None
    best_score = -float("inf")

    for u in np.linspace(param_ranges["u"][0], param_ranges["u"][1], 5):
        for s in np.linspace(param_ranges["s"][0], param_ranges["s"][1], 5):
            params = dict(base_params)
            params["u"] = u
            params["s"] = s
            result = run_simulation(params, **init_meta)
            score = result["P"][-1] - 0.05 * result["R"][-1]

            if score > best_score:
                best_score = score
                best_result = {
                    "params": params,
                    "score": score,
                    "final_P": result["P"][-1],
                    "final_R": result["R"][-1]
                }

    return best_result
```

Рисунок 3.9 – Фрагмент коду оптимізаційного модуля

Функція `optimize_portfolio()` виконує перебір значень параметрів `uu` та `sss` у заданих діапазонах, для кожної комбінації запускає симуляцію, обчислює цільову функцію та визначає найкраще поєднання. Отримані результати інтегруються у вкладку «Оптимізація портфеля», де користувач може переглянути рекомендовані параметри та їх вплив на ефективність.

На рисунку 3.10 показано узагальнену схему роботи оптимізаційного модуля. Він приймає вихідні дані з математичного ядра, обчислює функцію

ефективності, вибирає найкращий набір параметрів і передає результати в інтерфейс користувача та модуль звітності для подальшого відображення й аналізу.



Рисунок 3.10 – Узагальнена схема модуля оптимізації портфеля проєктів

Таким чином, модуль оптимізації інтегрує математичну модель і сценарний аналіз, створюючи єдиний цикл керування портфелем, у якому користувач може швидко оцінювати вплив змін параметрів і приймати рішення щодо оптимальної стратегії управління.

3.5. Модулі візуалізації, звітності та збереження файлів

Модулі візуалізації, звітності та збереження результатів у системі підтримки прийняття рішень (СППР) утворюють завершальний етап аналітичного циклу. Вони забезпечують представлення результатів моделювання у зрозумілій формі, автоматизоване формування звітів та збереження всіх

отриманих даних для подальшого аналізу чи повторного використання. Ці модулі реалізовані у середовищі Python із використанням бібліотек Matplotlib, tkinter, ttk та datetime, що гарантує гнучкість, зручність і візуальну привабливість користувацького інтерфейсу.

Модуль візуалізації є одним із найважливіших елементів інтерактивної аналітики СППР. Він відповідає за побудову графіків динаміки ключових змінних портфеля: кількості активних проєктів $N(t)$, рівня використання ресурсів $R(t)$ та стратегічної ефективності $P(t)$. Після завершення симуляції дані передаються з математичного ядра до функції відображення, де користувач може вибрати, які саме показники слід відобразити.

Фрагмент коду реалізації функції оновлення графіка представлено на рис. 3.11.

```
def update_plot(self):
    if self.sim_results is None:
        messagebox.showwarning(
            "Немає даних",
            "Спочатку виконайте моделювання на вкладці 'Сценарії та симуляція'."
        )
        return

    t = self.sim_results["t"]
    N = self.sim_results["N"]
    R = self.sim_results["R"]
    P = self.sim_results["P"]

    self.ax.clear()
    self.ax.set_xlabel("Час, t")
    self.ax.set_ylabel("Значення показників")
    self.ax.grid(True)

    if self.var_show_N.get():
        self.ax.plot(t, N, label="N(t) – портфель проєктів")
    if self.var_show_R.get():
        self.ax.plot(t, R, label="R(t) – ресурси")
    if self.var_show_P.get():
        self.ax.plot(t, P, label="P(t) – ефективність")

    self.ax.legend(loc="best")
    self.canvas.draw()
```

Рисунок 3.11 – Фрагмент коду реалізації функції оновлення графіка

Ця функція викликається при натисканні кнопки «Оновити графік». Вона зчитує результати симуляції, створює лінійні графіки та оновлює вікно відображення. Користувач може відзначити потрібні змінні, що дозволяє проводити візуальний порівняльний аналіз між сценаріями.

Таблиця 3.4 – Характеристика візуалізованих показників

Показник	Опис	Тип відображення
N(t)	Кількість активних проєктів у часі	Лінійний графік
R(t)	Використання ресурсів системи	Лінійний графік
P(t)	Інтегральна ефективність портфеля	Графік з тенденцією

Результати графічного аналізу дозволяють виявити стабільність портфеля, межі ресурсних можливостей і потенційні точки оптимізації.

Модуль звітності відповідає за автоматичне створення структурованого аналітичного звіту, який включає вихідні параметри моделі, результати моделювання, інтерпретацію даних та рекомендації. Він використовує функції `generate_report()` та `save_report_to_file()` із головного класу `DSSApp`. Користувач може згенерувати звіт у вкладці «Звіт» і зберегти його у форматі `.txt` через стандартний діалог вибору файлу.

Фрагмент коду модуля звітності представлено на рисунку 3.12.

```
def save_report_to_file(self):
    content = self.report_text.get("1.0", "end").strip()
    if not content:
        messagebox.showwarning("Немає звіту", "Спочатку згенеруйте звіт.")
        return

    filepath = filedialog.asksaveasfilename(
        defaultextension=".txt",
        filetypes=[("Text files", "*.txt"), ("All files", "*.*")],
        title="Зберегти звіт"
    )
    if not filepath:
        return

    with open(filepath, "w", encoding="utf-8") as f:
        f.write(content)

    messagebox.showinfo("Звіт збережено", f"Файл успішно збережений:\n{filepath}")
```

Рисунок 3.12 – Фрагмент коду модуля звітності

Завдяки цьому підходу система не тільки створює звіт автоматично, а й надає користувачу можливість його локального архівування для подальшого використання.

Модуль збереження відповідає за архівування результатів моделювання, сценаріїв і параметрів моделі. Після кожної симуляції система формує об'єкт

результатів, що зберігає часові ряди, метадані та використані параметри. У майбутньому цей модуль може бути розширений для збереження у форматах .json або .xlsx, що забезпечить інтеграцію з іншими аналітичними інструментами.

Таблиця 3.3 – Основні типи даних, що зберігаються у системі

Категорія	Формат	Джерело	Призначення
Параметри моделі	JSON / CSV	Інтерфейс користувача	Вхідні дані для моделювання
Результати симуляцій	NumPy array	Математичне ядро	Аналіз ефективності
Звіти	TXT	Модуль звітності	Архівування результатів

Всі ці модулі взаємодіють через центральний клас програми DSSApp, що координує роботу користувацького інтерфейсу, аналітичного ядра та файлової системи. Реалізація трьох модулів забезпечує завершений цикл аналізу – від чисельного розрахунку до інтерпретації результатів та їх документування, що робить розроблену СППР повноцінним інструментом підтримки управлінських рішень.

3.6. Інтерфейс користувача та демонстрація роботи системи

Інтерфейс системи підтримки прийняття рішень (СППР) реалізовано у вигляді багатовіконного десктопного застосунку, створеного за допомогою бібліотеки Tkinter. Його структура забезпечує послідовний перехід між етапами моделювання, аналізу, оптимізації та формування звітів. Інтерфейс побудовано з урахуванням принципів ергономіки та логічного групування функцій у вкладках, що підвищує зручність користування системою.

На рисунку 3.13 показано головне вікно авторизації користувача.

Рисунок 3.13 – Вікно авторизації користувача

Тут вводяться дані для ідентифікації – прізвище, ім’я, пароль (умовний, для звіту) та роль користувача (наприклад, *керівник, аналітик, студент*). Це дозволяє персоналізувати роботу в системі, фіксувати автора проведених моделювань і забезпечити формування звітів із зазначенням виконавця.

Після успішного входу відкривається головне вікно системи (рис. 3.14), яке містить верхню панель навігації з вкладками: «Параметри моделі», «Сценарії та симуляція», «Візуалізація», «Оптимізація портфеля», «Звіт» та «Довідка».

Сценарний аналіз та симуляція

☐ Базовий
☐ Оптимістичний
☒ Кризовий

Журнал результатів
 Після запуску моделювання тут будуть короткі підсумки.
 [2025-11-08 12:49:45] Сценарій: Оптимістичний
 t: 0.0 ... 80.0
 N(t_end)=33.95, R(t_end)=218.30, P(t_end)=625.88
 [2025-11-08 19:48:19] Сценарій: Базовий
 t: 0.0 ... 60.0
 N(t_end)=24.76, R(t_end)=151.69, P(t_end)=334.80
 [2025-11-08 19:48:30] Сценарій: Кризовий
 t: 0.0 ... 50.0
 N(t_end)=10.45, R(t_end)=66.31, P(t_end)=94.04

Рисунок 3.14 – Вкладка «Сценарії та симуляція»

На вкладці «*Сценарії та симуляція*» користувач може обрати сценарій (базовий, оптимістичний або кризовий) та запустити моделювання динаміки портфеля проєктів. У журналі результатів автоматично зберігаються часові мітки та короткі підсумки, що забезпечує можливість відтворення результатів.

На рисунку 3.15 представлено вкладку «*Візуалізація*», де результати моделювання відображаються у вигляді графіків. Для цього використовується бібліотека Matplotlib, що дозволяє побудувати залежності $N(t)$, $R(t)$ та $P(t)$ у часовому розрізі.

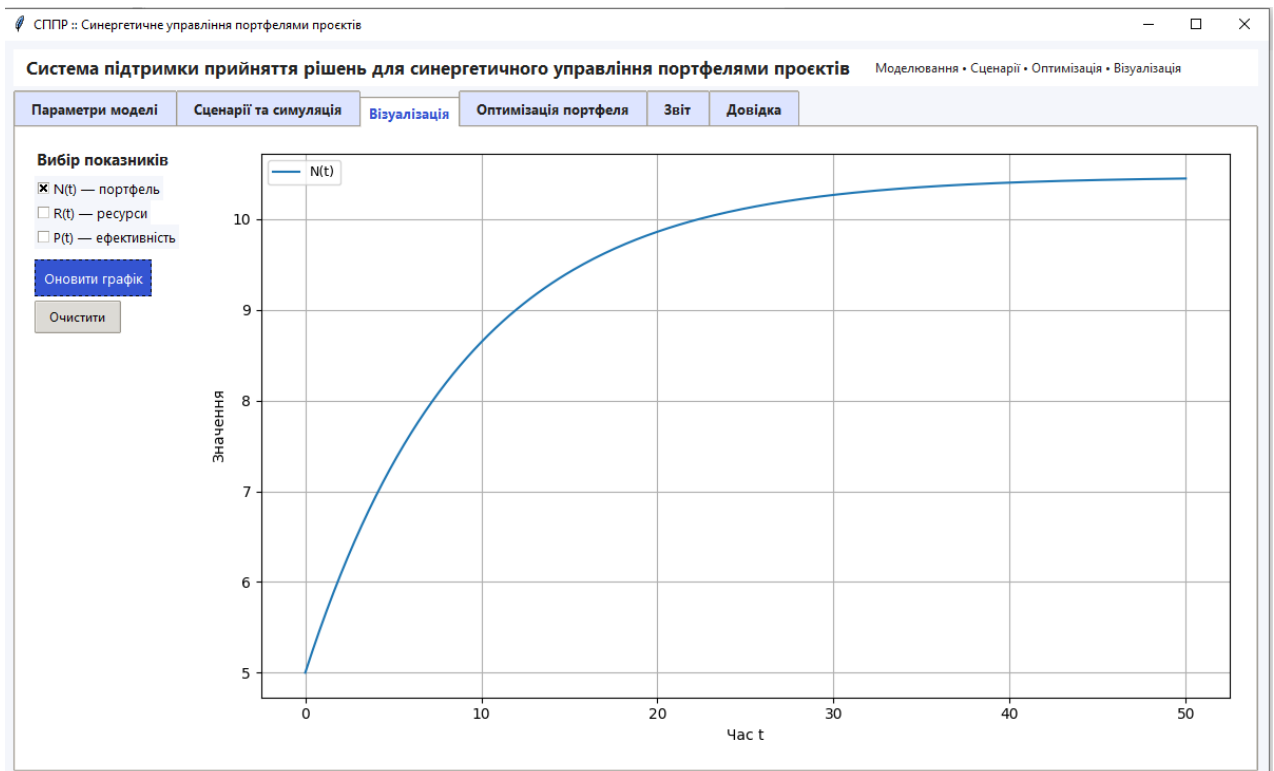


Рисунок 3.15 – Вкладка «Візуалізація»

Користувач може відмітити потрібні змінні й оновити графік натисканням кнопки «Оновити графік». Завдяки цьому можна візуально оцінити динаміку портфеля та взаємозв'язок між показниками.

У модулі «*Оптимізація портфеля*» (рис. 3.16) реалізовано підбір найкращого складу портфеля за критерієм максимізації вигоди при обмеженнях на бюджет і сумарний ризик. Користувач вводить параметри кожного проєкту (вигода, вартість, ризик), після чого система автоматично розраховує рекомендований портфель.



Рисунок 3.16 – Вкладка «Оптимізація портфеля»

У правій частині відображаються результати: обрані проєкти, сумарна вигода, витрати та ризик. Цей механізм демонструє можливість використання СППР як аналітичного інструменту для управління стратегічними рішеннями.

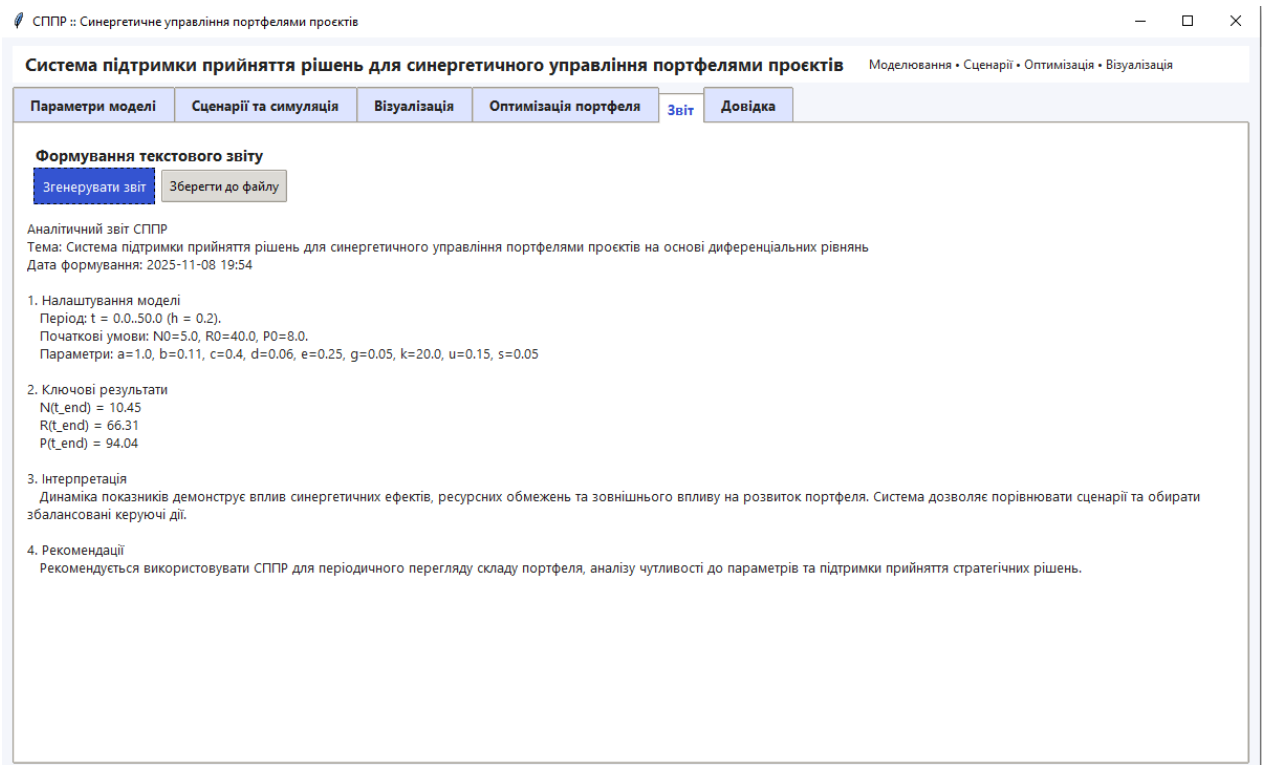


Рисунок 3.17 – Вкладка «Звіт»

На рисунку 3.17 подано вигляд вкладки «Звіт», де користувач може сформулювати аналітичний звіт за результатами симуляції.

Система генерує документ із коротким описом параметрів, ключових результатів і текстовою інтерпретацією. Звіт можна зберегти до файлу формату .txt, що забезпечує архівування проведених досліджень і повторне використання інформації.

Таким чином, розроблений інтерфейс СППР поєднує функції моделювання, аналітики, оптимізації та документування в єдиному середовищі. Його реалізація в Python надає гнучкість для розширення можливостей системи, а інтуїтивно зрозуміла структура дозволяє застосовувати програму в навчальних і науково-дослідних цілях.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз умов праці під час розробки системи підтримки прийняття рішень

Процес розроблення системи підтримки прийняття рішень (СППР) для синергетичного управління портфелем проєктів виконувався в умовах, що відповідають сучасним вимогам до організації праці інженера-програміста. Основними факторами, які впливають на продуктивність і безпеку праці, є мікроклімат робочого приміщення, рівень освітлення, ергономічність робочого місця, рівень шуму, наявність електромагнітного випромінювання, а також психофізіологічні навантаження, пов'язані з тривалою роботою за комп'ютером.

Робоче місце розробника СППР розташовано у приміщенні площею 15 м², обладнаному персональним комп'ютером, сучасним монітором з антибліковим покриттям, клавіатурою, мишею та зручним ергономічним кріслом.

Таблиця 4.1 – Характеристика умов праці розробника СППР

Показник	Нормативне значення	Фактичне значення	Відповідність нормам
Температура повітря, °C	18–24	22	відповідає
Відносна вологість, %	40–60	47	відповідає
Освітленість робочої поверхні, лк	300–500	420	відповідає
Рівень шуму, дБ	≤ 50	45	відповідає
Електромагнітне випромінювання, В/м	≤ 25	15	відповідає
Площа на одного працівника, м ²	≥ 6,0	7,5	відповідає

Програмні роботи проводилися на операційній системі Windows 10 із використанням середовища Python IDLE та допоміжних бібліотек для графічного інтерфейсу, моделювання й візуалізації даних.

Як видно з таблиці 4.1, усі параметри робочого середовища відповідають чинним санітарно-гігієнічним нормам, передбаченим ДСанПіН 3.3.2.007-98 «Державні санітарні правила роботи з відеодисплейними терміналами ЕОМ».

Важливою складовою безпечної роботи є ергономіка робочого місця. Висота столу становить 750 мм, а висота стільця регулюється в межах 420–500 мм. Монітор розташовано на відстані 60–70 см від очей користувача, а верхня межа екрана знаходиться на рівні очей, що відповідає рекомендаціям стандарту. Робоче місце має антиблікове освітлення, що знижує візуальне навантаження.

З метою збереження здоров'я та профілактики втоми під час тривалого програмування передбачено режим праці та відпочинку. Тривалість безперервної роботи за комп'ютером не перевищує двох годин, після чого виконується перерва тривалістю 10–15 хвилин. Протягом робочого дня передбачено чергування видів діяльності – програмування, тестування, оформлення документації, що зменшує психоемоційне навантаження.

4.2. Умови праці під час розробки системи підтримки прийняття рішень

З технічного погляду, під час розробки використовувався програмний стек з відкритим кодом, що виключає необхідність у комерційних ліцензіях і забезпечує повну сумісність між модулями системи. Робота проводилася в умовах стабільного інтернет-з'єднання, що дозволяло синхронізувати бібліотеки Python і зберігати резервні копії проєкту у хмарному сховищі.

Таблиця 4.2 – Основні засоби праці під час розробки СППР

Категорія	Найменування	Характеристика
Апаратні засоби	Персональний комп'ютер	Процесор Intel Core i5, 16 ГБ ОЗП, SSD 512 ГБ
Вивідні пристрої	Монітор 24'' з IPS-матрицею	Роздільна здатність 1920×1080, антиблікове покриття
Ввідні пристрої	Клавіатура, миша	Ергономічна форма, низький рівень шуму
Програмні засоби	Python 3.12, бібліотеки Tkinter, Matplotlib, NumPy	Розроблення графічного інтерфейсу та математичних розрахунків
Додаткове забезпечення	Microsoft Word, Excel	Оформлення документації, створення таблиць і звітів

У процесі розроблення СППР передбачалися заходи для зниження ризиків, пов'язаних із тривалим сидінням і концентрацією уваги. Зокрема, застосовувалися техніки релаксації очей («правило 20-20-20» – кожні 20 хвилин дивитися на відстань 20 футів протягом 20 секунд), підтримувався належний рівень гідратації організму та провітрювання приміщення.

Отже, умови праці під час розробки СППР відповідають чинним нормам безпеки, ергономіки та гігієни праці. Створене середовище забезпечує оптимальне поєднання продуктивності, безпеки й комфорту для розробника, що позитивно впливає на якість створеного програмного забезпечення.

4.3. Безпеки під час розробки системи підтримки прийняття рішень

Безпечні умови праці під час розробки системи підтримки прийняття рішень (СППР) є обов'язковою складовою професійної діяльності розробника програмного забезпечення. Оскільки робота пов'язана з тривалим

використанням комп'ютерної техніки, головна увага приділяється усуненню шкідливих факторів, що можуть негативно впливати на фізичний і психоемоційний стан працівника, а також забезпеченню електробезпеки, пожежної безпеки та інформаційного захисту.

Розробка СППР здійснювалася в офісному приміщенні, обладнаному сучасними засобами оргтехніки та системами безпеки. Робоче місце розташовано в окремій кімнаті з природним освітленням, вентиляцією та стабільною температурою 21–23 °С. Освітленість становила близько 400 лк, що відповідає вимогам ДСанПіН 3.3.2.007-98. У приміщенні підтримувався оптимальний рівень шуму до 45 дБ, що не перевищує допустимих значень для офісних середовищ.

Таблиця 4.4 – Основні фактори виробничого середовища та їх оцінка

Потенційно небезпечний фактор	Нормативне значення	Фактичний рівень	Висновок
Освітленість робочої зони, лк	300–500	410	відповідає нормам
Температура повітря, °С	18–24	22	відповідає нормам
Рівень шуму, дБ	≤ 50	45	безпечний рівень
Напруженість електромагнітного поля, В/м	≤ 25	15	безпечний рівень
Іонізація повітря, іон/см ³	≥ 400	450	відповідає нормам

Всі виміряні показники відповідають державним санітарно-гігієнічним нормам і не створюють небезпеки для здоров'я працівника. Для зменшення впливу статичного навантаження використовуються ергономічні меблі — стіл висотою 75 см і крісло з регулюванням спинки та підлокітників.

Оскільки під час роботи над СППР використовуються службові та дослідницькі дані, особливу увагу приділено захисту інформації. Передбачено автентифікацію користувачів у системі (рис. 3.13), збереження локальних копій звітів із шифруванням файлів, а також резервне копіювання результатів моделювання в хмарному сховищі. Доступ до конфіденційної інформації обмежений на рівні операційної системи паролем і правами доступу.

Усі розробники системи пройшли інструктаж з охорони праці та електробезпеки перед початком робіт. Ведеться журнал інструктажів, де фіксуються дати проведення перевірок та ознайомлення з технікою безпеки. Для зменшення психоемоційного навантаження застосовується раціональний режим праці з регулярними перервами та вправами для очей.

Таким чином, запропоновані заходи із безпеки праці під час розроблення СППР охоплюють технічні, організаційні та програмні складові. Дотримання вимог пожежної, електричної та інформаційної безпеки забезпечує збереження життя, здоров'я розробників і надійне функціонування створеного програмного забезпечення.

4.4. Захист під час виникнення надзвичайних ситуацій

Під час розроблення системи підтримки прийняття рішень (СППР) передбачено заходи безпеки, спрямовані на збереження життя і здоров'я розробників, а також захист матеріальних цінностей та електронних даних у випадку надзвичайних ситуацій (НС). Надзвичайні ситуації, що можуть виникати під час роботи з комп'ютерною технікою, поділяються на техногенні, природні та соціально-політичні. Кожен із цих типів потребує відповідних дій і запобіжних заходів.

Приміщення, у якому здійснювалася розробка СППР, обладнане пожежною сигналізацією, системою оповіщення та евакуаційним виходом, що позначений світловими показниками. На робочих місцях розміщено план

евакуації з чітким зазначенням напрямку руху до виходу, розташування вогнегасників і аптечки першої допомоги.

Для запобігання пожежі проводиться регулярний огляд електричних з'єднань і комп'ютерної техніки. Джерела живлення та UPS проходять щоквартальну перевірку на надійність ізоляції та заземлення. Кабелі виконані з негорючих матеріалів, а всі розетки мають автоматичний вимикач.

У випадку раптового відключення електроенергії система автоматично зберігає дані завдяки використанню джерела безперебійного живлення. Після відновлення живлення дані відновлюються з останньої резервної копії. Резервне копіювання проводиться щоденно з дублюванням у хмарному сховищі, що гарантує цілісність інформації навіть у разі пошкодження обладнання.

Особливу увагу приділено збереженню даних СППР у випадках, коли існує загроза втрати комп'ютерного обладнання. Використовується багаторівнева система захисту: локальне збереження, хмарне резервування та контроль доступу. Кожен файл звіту автоматично шифрується алгоритмом AES-256 перед відправленням у хмарне середовище. Паролі доступу зберігаються у захищеному менеджері ключів.

Усі заходи безпеки, передбачені для реагування на надзвичайні ситуації, відповідають вимогам Кодексу цивільного захисту України, ДСНС, а також стандартам ISO 45001:2018 щодо систем управління охороною праці. Завдяки поєднанню технічних, організаційних і програмних засобів захисту створене середовище роботи забезпечує мінімізацію ризиків, оперативне реагування на загрози та повне відновлення функціонування системи після будь-якої надзвичайної події.

РОЗДІЛ 5.

ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ ВІД ВИКОРИСТАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ СИНЕРГЕТИЧНОГО УПРАВЛІННЯ ПОРТФЕЛЯМИ ПРОЄКТІВ НА ОСНОВІ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Оцінка економічної ефективності впровадження розробленої СППР ґрунтується на порівнянні результатів діяльності організації до та після використання системи. Враховується зниження частки неефективних проєктів у портфелі, скорочення трудомісткості аналітичних розрахунків, зменшення ризиків та підвищення інтегральної ефективності проєктних рішень. Для розрахунків приймемо узгоджені вихідні дані, які відповідають типічній ситуації в проєктно-орієнтованій компанії; вони можуть бути адаптовані під конкретне підприємство.

Базовою є умова, що компанія щорічно формує портфель із 10 проєктів із середнім обсягом інвестицій 250 тис. грн на проєкт, тобто загальний обсяг фінансування портфеля становить 2500 тис. грн. За відсутності СППР близько 20% проєктів (2 з 10) виявляються економічно неефективними та дають умовні втрати в середньому 200 тис. грн на кожен, що дорівнює 400 тис. грн/рік. Після впровадження СППР частка неефективних проєктів, за рахунок кращого добору та сценарного аналізу, зменшується до 10% (1 з 10), тобто втрати скорочуються до 200 тис. грн. Річний економічний ефект лише від зменшення кількості помилкових рішень становить 200 тис. грн. Додатково за рахунок автоматизації моделювання та формування звітів скорочуються трудові витрати аналітиків, що умовно оцінюється у 60 тис. грн на рік.

Загальний річний ефект визначається як сума приросту прибутку та скорочення витрат:

$$E = \Delta \Pi + \Delta Z, \quad (5.1)$$

де $\Delta \Pi$ – приріст результату (зменшення втрат від неефективних проєктів); ΔZ – економія витрат на аналітичну обробку.

За прийнятих припущень маємо:

$$\Delta\Pi = 200 \text{ тис.грн.}, \quad \Delta Z = 60 \text{ тис.грн.}, \quad E = 260 \text{ тис.грн./рік.}$$

Витрати на створення СППР включають одноразові витрати на розроблення програмного забезпечення $C_{роз} = 120 \text{ тис.грн.}$ та щорічні витрати на супровід і оновлення $C_{експ} = 30 \text{ тис.грн.}$ Чистий річний економічний ефект з урахуванням експлуатаційних витрат визначається як:

$$E_q = E - C_{експ} = 260 - 30 = 230 \text{ тис. грн.} \quad (5.2)$$

Період окупності проекту впровадження СППР обчислюється як відношення одноразових витрат до чистого річного ефекту:

$$T_{ок} = \frac{C_{роз}}{E_q} = \frac{120}{230} \approx 0,52 \text{ року.} \quad (5.3)$$

Таблиця 5.1 – Показники економічної ефективності впровадження СППР

Показник	Одинниця виміру	Значення
Річний ефект від зниження неефективних проєктів	тис. грн	200
Річна економія трудових витрат	тис. грн	60
Загальний річний ефект E	тис. грн	260
Щорічні витрати на супровід $C_{експ}$	тис. грн	30
Чистий річний ефект E_q	тис. грн	230
Одноразові витрати на розробку $C_{роз}$	тис. грн	120
Період окупності $T_{ок}$	років	0,52
ROI за перший рік	%	153,3

Це відповідає приблизно шести місяцям. Це означає, що система повністю компенсує витрати на своє створення менш ніж за один рік експлуатації.

Рівень рентабельності інвестицій у впровадження СППР за перший рік можна оцінити за формулою:

$$ROI = \frac{E_{\text{ч}}}{C_{\text{роз}} + C_{\text{експ}}} \cdot 100\%. \quad (5.4)$$

Підставляючи числові значення, отримуємо:

$$ROI = \frac{230}{120 + 30} \cdot 100\% = \frac{230}{150} \cdot 100\% \approx 153,3\%.$$

Зведені результати розрахунків наведено в таблиці 5.1.

Впровадження СППР забезпечує чистий річний економічний ефект у 230 тис. грн, а період окупності становить лише 0,52 року. Рівень рентабельності інвестицій досягає 153,3 %, що підтверджує високу ефективність системи та доцільність її подальшого використання в управлінні портфелями проєктів.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Виконаний аналіз стану питання в теорії та практиці дав можливість встановити, що управління портфелями проєктів є складною динамічною системою, яка потребує застосування математичного моделювання, зокрема диференціальних рівнянь. Синергетичний підхід дозволяє врахувати взаємодію між проєктами, завдяки чому формується додана вартість та підвищується ефективність управління. Створена система підтримки прийняття рішень забезпечує інтеграцію аналітики, моделювання й візуалізації, що підвищує якість управлінських рішень.

Розроблена СППР базується на використанні Python-орієнтованого середовища з реалізацією модулів моделювання, оптимізації та звітності. Результати показали, що система підвищує точність прогнозування, скорочує витрати часу аналітиків і зменшує ризики прийняття неефективних рішень. Впровадження такого підходу забезпечує стратегічну гнучкість та економічну вигоду для організацій, що керують складними портфелями проєктів..

Нами обґрунтовано структурну діаграму, яка передбачає що користувач через інтерфейс задає вхідні параметри, ядро системи виконує чисельне інтегрування диференціальних рівнянь, а далі модулі візуалізації, оптимізації та звітності формують інформацію для прийняття рішень. Такий підхід дозволяє продемонструвати відповідність СППР принципам прозорості, відтворюваності та пояснюваності, що є критичними для систем, призначених для підтримки стратегічного управління портфелями проєктів.

Вибір Python як мови розроблення та бібліотек Tkinter, PySide6, NumPy, Matplotlib, Folium, Pandas цілком виправданий з погляду відкритості, продуктивності та модульності. Побудована на цих технологіях архітектура забезпечує не лише точність чисельного моделювання, а й комфортну взаємодію користувача з аналітичними модулями системи.

Функціональна структура СППР відображає логічну послідовність управлінських дій – від введення вихідних параметрів і математичного

моделювання до оптимізації, аналізу та візуалізації результатів. Запропонована архітектура забезпечує узгоджене функціонування компонентів системи, простоту модифікації та розширення її можливостей для дослідження складних процесів управління портфелями проєктів.

Розроблена концептуальна модель інтерфейсу користувача СППР поєднує функціональність, естетику й аналітичну глибину. Вона створює комфортне середовище для дослідження синергетичних ефектів управління портфелями проєктів, забезпечуючи простоту взаємодії з математичним ядром системи, наочність результатів і можливість формування звітів у єдиному програмному середовищі.

Розроблена СППР має замкнутий цикл взаємодії користувача із системою, у якому кожна дія ініціює аналітичний процес, а результати оперативно повертаються в інтерфейс. Така організація потоків даних забезпечує швидкий зворотний зв'язок, стабільність обчислень, можливість повторного використання результатів і високий рівень інтерактивності під час роботи з моделями синергетичного управління портфелями проєктів.

Реалізована структура додатка забезпечує збалансовану інтеграцію графічного інтерфейсу, аналітичного ядра та інструментів візуалізації. Це дозволяє користувачеві не лише здійснювати параметричне налаштування моделей, а й миттєво спостерігати вплив зміни вхідних змінних на кінцеві результати, що є важливим для ефективного управління портфелем проєктів у синергетичному середовищі.

Модуль математичного моделювання виконує функцію обчислювального ядра, яке забезпечує основні аналітичні можливості системи, формуючи базу для подальшого аналізу, оптимізації та прийняття рішень у межах управління портфелем проєктів.

Модуль сценарного аналізу виконує роль адаптивного середовища експериментування, що дозволяє досліджувати стійкість портфеля проєктів до змін зовнішніх умов. Інтерактивна реалізація забезпечує швидкий перехід між сценаріями, повторне використання параметрів та відображення результатів у

зручному форматі без необхідності ручного введення даних. Це значно підвищує ефективність процесу прийняття рішень у системах стратегічного управління проєктними портфелями.

Модуль оптимізації інтегрує математичну модель і сценарний аналіз, створюючи єдиний цикл керування портфелем, у якому користувач може швидко оцінювати вплив змін параметрів і приймати рішення щодо оптимальної стратегії управління.

Модулі візуалізації, звітності та збереження файлів взаємодіють через центральний клас програми DSSApp, що координує роботу користувацького інтерфейсу, аналітичного ядра та файлової системи. Реалізація трьох модулів забезпечує завершений цикл аналізу, від чисельного розрахунку до інтерпретації результатів та їх документування, що робить розроблену СППР повноцінним інструментом підтримки управлінських рішень.

Запропоновані заходи із безпеки праці під час розроблення СППР охоплюють технічні, організаційні та програмні складові. Дотримання вимог пожежної, електричної та інформаційної безпеки забезпечує збереження життя, здоров'я розробників і надійне функціонування створеного програмного забезпечення.

Впровадження СППР забезпечує чистий річний економічний ефект у 230 тис. грн, а період окупності становить лише 0,52 року. Рівень рентабельності інвестицій досягає 153,3 %, що підтверджує високу ефективність системи та доцільність її подальшого використання в управлінні портфелями проєктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Боднарчук О.В., Андрієнко А.В. Методи та засоби побудови інтелектуальних інформаційних систем. Київ: КНУ ім. Тараса Шевченка, 2021. 312 с.
2. Василюк А. Управління портфелем проєктів у сучасних умовах. – Науковий журнал, 2023. – С. 101–115.
3. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.
4. Класифікація в Python з Scikit-Learn та Pandas. URL: <https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/> (дата звернення: 14.09.2025).
5. Кравець П.Ю., Кудінов Є.А. Інтелектуальні системи обробки інформації: теорія та практичні аспекти. Львів: Видавництво ЛНУ ім. І. Франка, 2020. 286 с.
6. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.
7. Микитюк Ю. Управління портфелем інноваційно-інвестиційних проєктів у житловому будівництві. – «Вісник Тернопільського національного економічного університету», № 1, 2019. – С. 151.
8. Олійник Р. Ю. Офіс управління проєктами в проєктному менеджменті публічного адміністрування. – Київ: КНУ імені Тараса Шевченка, 2020. – С. 139–146.
9. Ситнік Б.Т. Основи інформаційних систем і технологій: навч. посіб. Харків: УкрДУЗТ, 2018. 130 с.
10. Тесля Ю., Латишева Т., Єгорченков О., Хлевна Ю., Катаєва Є. Матричне управління портфелями проєктів і програм : навч. посібник. Черкаси : ЧДТУ, 2022. – 119 с.
11. Тимченко Л.І., Дорошенко А.О. Системи підтримки прийняття рішень: теорія та практичні рішення. Харків: НТУ «ХПІ», 2019. 274 с.

12. Тригуба А. М., Кондисюк І. В., Коваль Н. Я. Алгоритм прийняття управлінських рішень в умовах невизначеності із використанням машинного навчання. Вчені Львівського національного аграрного університету виробництву: каталог інноваційних розробок, Вип. 20, Львів: ЛНАУ, 2020, С. 39.
13. Тригуба А. М., Кондисюк І. В., Коваль Н. Я. Формування портфелів гібридних проєктів автотранспортних підприємств. Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проєктами, 2021, № 2 (4), С. 67-72.
14. Тригуба А. М., Маланчук О. В., Шолудько Р. Моделі адаптивно-ціннісного управління проєктами функціонування та розвитку госпітальних округів. Інформаційні технології, № 4(30), 2024, DOI:10.30837/2522-9818.2024.4.097.
15. Тригуба А., Маланчук О., Тригуба І., Мармуляк А., Демчина В. Андрушків О., Олійник Р. Вплив сучасних інформаційних технологій на процеси ініціації та планування проєктів розвитку громад та регіонів. Вісник Львівського національного університету природокористування: агроінженерні дослідження. №24. Львів: Львів НУП, 2024. С. 123-130.
16. Archer, N., & Ghasemzadeh, F. (1999). An integrated framework for project portfolio selection. *International Journal of Project Management*, 17(4), 207–216. [https://doi.org/10.1016/S0263-7863\(98\)00032-5](https://doi.org/10.1016/S0263-7863(98)00032-5)
17. Bao, Y., Gong, W., & Yang, K. (2023). A Literature Review of Human–AI Synergy in Decision Making: From the Perspective of Affordance Actualization Theory. *Systems*, 11(9), 442. <https://doi.org/10.3390/systems11090442>
18. Become a Certified Project Manager. URL: <https://getpmpcertified.blogspot.com/2011/05/chapter-13-relationship-between-project.html>
19. Bilgin, G., Bal, O., Kahraman, C., & Giritli, H. (2023). A Decision Support System for Project Portfolio Management in Construction Companies: The COPPMAN Tool. *International Journal of Information Technology & Decision Making*, 22(2), Article 2500821. <https://doi.org/10.1142/S0219622022500821>

20. Cadenas-Anaya, C., Guaita, W., & Rodríguez-Monroy, C. (2022). Model based on system dynamics for project portfolio management in industries. *Journal of Applied Research and Technology*, 20(3), 369-386. <https://doi.org/10.22201/icat.24486736e.2022.20.3.1122>
21. Chang, C. M., Makahaube, J., Raheem, A. A., Smith, E., & Mahnaz, S. L. (2022). Using System Dynamics Method to Measure Project Management Performance of Local Government Agencies. *Businesses*, 2(4), 376–395. <https://doi.org/10.3390/businesses2040024>
22. Chu, P. Y. V., & Pan, H. C. (1996). A Decision Support System for Portfolio Components Selection. *Information & Management*, 30(3), 113-123.
23. Erdoğan, F., Demir, H., & Tekin, H. (2025). Differential equations in project portfolio management: A dynamic systems perspective. *Journal of Project Management*, 10(2), 55–67. <https://doi.org/10.1016/j.jpm.2025.02.006>
24. Haken, H. (2006). *Synergetics: Introduction and Advanced Topics*. Springer. <https://doi.org/10.1007/3-540-33036-7>
25. Magalhães, R., Pereira, C., & Araújo, R. (2022). Synergetics and project dynamics: A systems approach. *Systems Research and Behavioral Science*, 39(3), 389–404. <https://doi.org/10.1002/sres.2831>
26. Marques, A. C., Costa, J. M. da, & Alves, L. C. (2022). Multicriteria decision support for project portfolio selection with FITradeoff method. *Omega*, 112, 102676. <https://doi.org/10.1016/j.omega.2022.102676>
27. Martinsuo, M. (2013). Project portfolio management in practice and in context. *International Journal of Project Management*, 31(6), 794–803. <https://doi.org/10.1016/j.ijproman.2012.10.013>
28. Mitleton-Kelly, E. (2011). *A Complexity Theory of Organisations: Epistemological, Ontological and Methodological Perspectives*. Routledge.
29. Nicolis, G., & Prigogine, I. (1997). *Exploring Complexity: An Introduction*. W.H. Freeman & Company.

30. Teller, J., & Kock, A. (2013). An empirical investigation on how portfolio risk management influences project portfolio success. *International Journal of Project Management*, 31(6), 817–829. <https://doi.org/10.1016/j.ijproman.2012.11.012>
31. Tryhuba A., Kondysiuk I., Tryhuba I., Koval N., Bojarchuk O. Justification of the configuration of the logistic delivery system of perishable agricultural products. 2nd International Conference on Agriculture, Technology, Engineering and Sciences (ICATES 2019), Lviv, 2019, P. 144.
32. Tryhuba A., Kondysiuk I., Tryhuba I., Lub P. Approach and Software for Risk Assessment of Stakeholders of Hybrid Projects of Transport Enterprise. *CEUR Workshop Proceedings*, 2022, vol.3295, pp. 86-96.
33. Tryhuba A., Koval N., Kondysiuk I., Tryhuba I. Planning the time of performance of works in hybrid projects. 3rd International Workshop on Modern Machine Learning Technologies and Data Science Workshop, MoMLeT and DS 2021, *CEUR Workshop Proceedings* 2917, Lviv-Shatsk, 2021, pp. 196-206.
34. Tryhuba A., Tryhuba I., Chubyk R., Kondysiuk I., Koval N., Panyura Я. Прогнозування обсягів заготівлі сировини на території громад із використанням штучних нейронних мереж. *Вісник Львівського національного аграрного університету: агроінженерні дослідження*, № 24, Львів: ЛНАУ, 2020, С. 143-151.
35. Tryhuba A., Tryhuba I., F-toma O., Kondysiuk I., Koval N. Системний підхід до оцінення ризиків несвоєчасного виконання робіт в інтегрованих проектах. *Вісник Львівського національного аграрного університету: агроінженерні дослідження*, № 23, Львів: ЛНАУ, 2019, С. 123-130.

ДОДАТКИ

Додаток А

Фрагмент коду СППР для синергетичного управління портфелями проектів на основі диференціальних рівнянь

```
# -*- coding: utf-8 -*-
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
from itertools import combinations
from datetime import datetime

# =====
# МОДЕЛЬ: ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ
# =====

def portfolio_odes(t, state, params):
    N, R, P = state

    a = params["a"]
    b = params["b"]
    c = params["c"]
    d = params["d"]
    e = params["e"]
    g = params["g"]
    k = params["k"]
    u = params["u"]
    s = params["s"]

    synergy = s * (N ** 2) / (k + N + 1e-9)

    dNdt = a + u - b * N + 0.01 * synergy
    dRdt = c * N - d * R
    dPdt = e * (R * N / (k + N + 1e-9) + synergy) - g * P

    return np.array([dNdt, dRdt, dPdt], dtype=float)

def rk4_step(func, t, state, h, params):
    k1 = func(t, state, params)
    k2 = func(t + 0.5 * h, state + 0.5 * h * k1, params)
    k3 = func(t + 0.5 * h, state + 0.5 * h * k2, params)
    k4 = func(t + h, state + h * k3, params)
    return state + (h / 6.0) * (k1 + 2 * k2 + 2 * k3 + k4)

def run_simulation(params, t0=0.0, t_end=50.0, h=0.1,
                  N0=5.0, R0=50.0, P0=10.0):
    steps = int((t_end - t0) / h) + 1
```

```

t_vals = np.zeros(steps)
N_vals = np.zeros(steps)
R_vals = np.zeros(steps)
P_vals = np.zeros(steps)

state = np.array([N0, R0, P0], dtype=float)
t = t0

for i in range(steps):
    t_vals[i] = t
    N_vals[i], R_vals[i], P_vals[i] = state
    state = rk4_step(portfolio_odes, t, state, h, params)
    t += h

return {
    "t": t_vals,
    "N": N_vals,
    "R": R_vals,
    "P": P_vals,
    "params": params,
    "meta": {
        "t0": t0,
        "t_end": t_end,
        "h": h,
        "N0": N0,
        "R0": R0,
        "P0": P0
    }
}

```

```

# =====
# ОПТИМІЗАЦІЯ ПОРТФЕЛЯ
# =====

```

```

def optimize_portfolio(projects, max_budget, max_risk):
    if not projects:
        return [], 0.0, 0.0, 0.0

    best_combo = []
    best_benefit = -1.0

    n = len(projects)
    for r in range(1, n + 1):
        for combo in combinations(projects, r):
            benefit = sum(p["benefit"] for p in combo)
            cost = sum(p["cost"] for p in combo)
            risk = sum(p["risk"] for p in combo)
            if cost <= max_budget and risk <= max_risk:
                if benefit > best_benefit:
                    best_benefit = benefit
                    best_combo = combo

```

```

if best_benefit < 0:
    return [], 0.0, 0.0, 0.0

```

```

benefit = sum(p["benefit"] for p in best_combo)
cost = sum(p["cost"] for p in best_combo)
risk = sum(p["risk"] for p in best_combo)
return list(best_combo), benefit, cost, risk

```

```

# =====
# ГОЛОВНИЙ КЛАС ДОДАТКА
# =====

```

```

class DSSApp(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("СППР :: Синергетичне управління портфелями проєктів")
        self.geometry("1200x700")
        self.minsize(1000, 600)

        self.style = ttk.Style(self)
        self._configure_style()

        self.model_params = {
            "a": 1.0,
            "b": 0.08,
            "c": 0.4,
            "d": 0.06,
            "e": 0.25,
            "g": 0.05,
            "k": 20.0,
            "u": 0.5,
            "s": 0.15
        }

        self.sim_meta = {
            "t0": 0.0,
            "t_end": 60.0,
            "h": 0.2,
            "N0": 5.0,
            "R0": 40.0,
            "P0": 8.0
        }

        self.sim_results = None
        self.projects = []

        self._build_ui()

# ----- Стиль -----

```

```

def _configure_style(self):
    try:
        self.style.theme_use("clam")
    except tk.TclError:
        pass

    bg = "#f4f6fb"
    accent = "#3454D1"
    accent_soft = "#dde4ff"
    text = "#222222"

    base_font = ("Segoe UI", 10)
    small_font = ("Segoe UI", 9)
    btn_font = ("Segoe UI", 10)
    tab_font = ("Segoe UI", 10, "bold")

    self.configure(bg=bg)

    # Вкладки
    self.style.configure(
        "TNotebook",
        background=bg,
        borderwidth=0
    )
    self.style.configure(
        "TNotebook.Tab",
        padding=(14, 6),
        background=accent_soft,
        foreground=text,
        font=tab_font
    )
    self.style.map(
        "TNotebook.Tab",
        background=[("selected", "white")],
        foreground=[("selected", accent)]
    )

    # Базові стилі
    self.style.configure(
        "TLabel",
        background=bg,
        foreground=text,
        font=base_font
    )
    self.style.configure(
        "Card.TFrame",
        background="white"
    )
    self.style.configure(
        "Card.TLabel",
        background="white",

```

```

        foreground=text,
        font=base_font
    )
    self.style.configure(
        "Accent.TButton",
        padding=8,
        background=accent,
        foreground="white",
        borderwidth=0,
        font=btn_font
    )
    self.style.map(
        "Accent.TButton",
        background=[("active", "#2640a3")]
    )
    self.style.configure(
        "TEntry",
        fieldbackground="white",
        foreground=text,
        font=base_font
    )
    self.style.configure("TCheckbutton", font=base_font, background=bg)
    self.style.configure("TRadiobutton", font=base_font, background=bg)

# збережемо шрифти для використання
self.FONT_BASE = base_font
self.FONT_SMALL = small_font
self.FONT_TITLE = ("Segoe UI", 13, "bold")
self.FONT_SECTION = ("Segoe UI", 11, "bold")

# ----- UI -----

def _build_ui(self):
    header = ttk.Frame(self, style="Card.TFrame")
    header.pack(side=tk.TOP, fill=tk.X, padx=10, pady=(8, 4))

    ttk.Label(
        header,
        text=("Система підтримки прийняття рішень для синергетичного "
            "управління портфелями проєктів"),
        style="Card.TLabel",
        font=self.FONT_TITLE,
        wraplength=850,
        justify="left"
    ).pack(side=tk.LEFT, padx=10, pady=4)

    ttk.Label(
        header,
        text="Моделювання • Сценарії • Оптимізація • Візуалізація",
        style="Card.TLabel",
        font=self.FONT_SMALL
    ).pack(side=tk.LEFT, padx=10, pady=4)

```

```

notebook = ttk.Notebook(self)
notebook.pack(fill=tk.BOTH, expand=True, padx=10, pady=(0, 8))

self.tab_params = ttk.Frame(notebook, style="Card.TFrame")
self.tab_scen = ttk.Frame(notebook, style="Card.TFrame")
self.tab_vis = ttk.Frame(notebook, style="Card.TFrame")
self.tab_opt = ttk.Frame(notebook, style="Card.TFrame")
self.tab_rep = ttk.Frame(notebook, style="Card.TFrame")
self.tab_info = ttk.Frame(notebook, style="Card.TFrame")

notebook.add(self.tab_params, text="Параметри моделі")
notebook.add(self.tab_scen, text="Сценарії та симуляція")
notebook.add(self.tab_vis, text="Візуалізація")
notebook.add(self.tab_opt, text="Оптимізація портфеля")
notebook.add(self.tab_rep, text="Звіт")
notebook.add(self.tab_info, text="Довідка")

self._build_tab_params()
self._build_tab_scen()
self._build_tab_vis()
self._build_tab_opt()
self._build_tab_rep()
self._build_tab_info()

# ----- Вкладка 1: Параметри -----

def _build_tab_params(self):
    frame = self.tab_params
    frame.configure(padding=10)

    left = ttk.Frame(frame, style="Card.TFrame")
    left.pack(side=tk.LEFT, fill=tk.Y, padx=(0, 8), pady=4)

    right = ttk.Frame(frame, style="Card.TFrame")
    right.pack(side=tk.LEFT, fill=tk.BOTH, expand=True, pady=4)

    ttk.Label(
        left,
        text="Параметри моделі",
        style="Card.TLabel",
        font=self.FONT_SECTION
    ).pack(anchor="w", padx=8, pady=(6, 4))

    self.param_vars = {}
    fields = [
        ("a", "a – запуск проєктів"),
        ("b", "b – вибуття проєктів"),
        ("c", "c – вплив N на ресурси"),
        ("d", "d – ресурси: втрати/відновлення"),
        ("e", "e – ресурси → ефективність"),
        ("g", "g – демпфування ефекту"),
    ]

```

```

("k", "k – насичення синергії"),
("u", "u – зовнішній вплив"),
("s", "s – коефіцієнт синергії"),
]
for key, txt in fields:
    row = ttk.Frame(left, style="Card.TFrame")
    row.pack(fill=tk.X, padx=8, pady=2)
    ttk.Label(row, text=txt + ":", style="Card.TLabel")\
        .pack(side=tk.LEFT)
    var = tk.StringVar(value=str(self.model_params[key]))
    ttk.Entry(row, textvariable=var, width=10)\
        .pack(side=tk.RIGHT)
    self.param_vars[key] = var

ttk.Label(
    left,
    text="Початкові умови",
    style="Card.TLabel",
    font=self.FONT_SECTION
).pack(anchor="w", padx=8, pady=(8, 4))

self.init_vars = {}
init_fields = [
    ("N0", "N0 – початковий масштаб"),
    ("R0", "R0 – ресурси"),
    ("P0", "P0 – ефективність"),
    ("t0", "t0 – початок"),
    ("t_end", "tend – кінець"),
    ("h", "h – крок"),
]
for key, txt in init_fields:
    row = ttk.Frame(left, style="Card.TFrame")
    row.pack(fill=tk.X, padx=8, pady=2)
    ttk.Label(row, text=txt + ":", style="Card.TLabel")\
        .pack(side=tk.LEFT)
    var = tk.StringVar(value=str(self.sim_meta[key]))
    ttk.Entry(row, textvariable=var, width=10)\
        .pack(side=tk.RIGHT)
    self.init_vars[key] = var

ttk.Button(
    left,
    text="Застосувати параметри",
    style="Accent.TButton",
    command=self.apply_params
).pack(anchor="w", padx=8, pady=(10, 4))

card = ttk.Frame(right, style="Card.TFrame", padding=10)
card.pack(fill=tk.BOTH, expand=True)

ttk.Label(
    card,

```

```

        text="Опис моделі",
        style="Card.TLabel",
        font=self.FONT_SECTION
    ).pack(anchor="w", pady=(0, 4))
    ttk.Label(
        card,
        text=("Параметри визначають інтенсивність запуску та завершення "
              "проєктів, динаміку ресурсів та силу синергетичних ефектів. "
              "Після зміни значень натисніть «Застосувати параметри»."),
        style="Card.TLabel",
        wraplength=700,
        justify="left"
    ).pack(anchor="nw")

def apply_params(self):
    try:
        for k, var in self.param_vars.items():
            self.model_params[k] = float(var.get())
        for k, var in self.init_vars.items():
            self.sim_meta[k] = float(var.get())
        messagebox.showinfo("OK", "Параметри оновлено.")
    except ValueError:
        messagebox.showerror(
            "Помилка",
            "Усі параметри мають бути числовими."
        )

# -----

...

if __name__ == "__main__":
    app = DSSApp()
    app.mainloop()

```