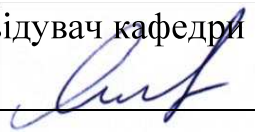
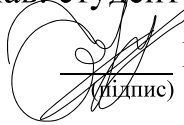
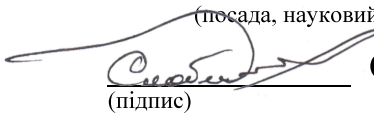


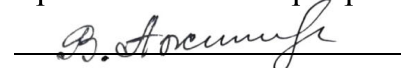
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**Національний університет кораблебудування імені адмірала Макарова****Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій****"Допущений до захисту"****Завідувач кафедри ІУСТ****к.т.н, доц. Михелєв І.Л.****" ____ " _____ 2025 р.*****КВАЛІФІКАЦІЙНА РОБОТА*****на здобуття ступеня вищої освіти "магістр"****на тему:****Дослідження методів підтримки прийняття рішень на основі нечітких
логічних правил для розробки інформаційної системи структуризації
повідомлення результату****Виконав: студент групи 6141 м****Горун І.П.****(підпис)****Керівник роботи:****к.т.н., проф****(посада, науковий ступінь вчене звання)****Слободян С.О****(підпис)****м. Миколаїв – 2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
 Навчально-науковий інститут комп'ютерних наук та управління проєктами
 Кафедра інформаційних управляючих систем та технологій
 Спеціальність 122 "Комп'ютерні науки"
 Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми



к.т.н, доц. Ажищев В.Ф.

"___" _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Горуну Ігору Петровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів підтримки прийняття рішень на основі нечітких логічних правил для розробки інформаційної системи структуризації повідомлення результату

Керівник роботи Слободян Сергій Олегович

Затверджені наказом ректора № 1355-уч. від "14" листопада 2025 року.

2. Термін подання роботи: 15 грудня 2025 р.

3. Вихідні дані до проєкту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

Вступ, • Аналіз предметної області і постановка задачі, • Проведення досліджень за темою магістерської роботи, • Реалізація проєкту інформаційної системи (технічна документація по проєкту), • Результати розробки, • Організаційно-економічний розділ, • Розділ з охорони праці • Висновки, • Список використаної літератури, • Додатки Технічне завдання, текст програми, інструкція користувачу

5. Перелік презентаційних матеріалів

1 Варіанти обробки даних 2 Недоліки існуючої системи 3 Концепція розробляємої системи

4 Модель варіантів використання 6 Модель взаємодії системи із зовнішніми сутностями

7 Концептуальна модель даних 8 Фізична модель даних 9 Діаграма класів ПЗ

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	К.т.н. проф. Слободян С.О.	05.09.2024	22.12.2024
2	К.т.н. проф. Слободян С.О.	23.12.2024	15.06.2025
3	К.т.н. проф. Слободян С.О.	15.06.2025	25.09.2025
4	К.т.н. проф. Слободян С.О.	25.09.2025	14.11.2025
5	К.т.н. проф. Слободян С.О.	25.09.2025	14.11.2025
6	К.т.н. проф. Слободян С.О.	25.09.2025	14.11.2025

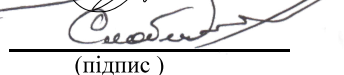
7. Дата видачі завдання 05 вересня 2024 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2025	
3	Розділ з розробки проєктних рішень по системі (та її частинах)	15.06.2025	
4	Реалізація проєкту та розробка робочої документації	25.09.2025	
5	Обчислення економічної ефективності	14.11.2025	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2025	
7	Оформлення пояснювальної записки	14.11.2025	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2025	
9	Подання магістерської роботи рецензенту	20.11.2025	
10	Подання магістерської роботи на державний захист	15.12.2025	

Студент

Керівник роботи


(підпис)


(підпис)

Горун І.П.
(ПІБ)

Слободян С.О.
(ПІБ)

АНОТАЦІЯ

Магістерська дипломна робота на тему «Дослідження методів підтримки прийняття рішень на основі нечітких логічних правил для розробки інформаційної системи структуризації повідомлення результату».

Магістерська дипломна робота виконана на 98 аркушах, містить 7 таблиць, 11 рисунків, 9 додатків та список використаної літератури з 15 найменувань., які містять технічне завдання, текст програмного коду та інструкцію користувача.

Метою роботи є дослідити застосування теорії нечітких множин шляхом використання критеріїв якісного характеру, які описуються і аналізуються в рамках цієї теорії, та реалізувати на практиці на її основі системи для підтримки прийняття рішень..

Об'єкт і предмет дослідження:

- **Об'єкт дослідження** - процеси проектування ІС.
- **Предмет дослідження** - моделі і технології розробки ІС на основі нечітких логічних правил.

Для досягнення поставленої цілі необхідно вирішити такі задачі:

- 1) проаналізувати та порівняти методи підтримки прийняття рішень на основі нечіткої математики, зробити висновки й обрати найбільш доцільний для реалізації метод;
- 2) виконати постановку задачі на розробку інформаційної системи реалізації;
- 3) розробити інформаційну систему.

Ключові слова: інформаційна система, прийняття рішень, нечітка логіка, структуризація повідомлень.

ABSTRACT

Master's thesis on the topic "Research on decision support methods based on fuzzy logical rules for the development of an information system for structuring the result message."

The master's thesis is completed on 98 sheets, contains 7 tables, 11 figures, 9 appendices and a list of used literature of 15 titles, which contain the technical specifications, the text of the program code and the user manual.

Object and subject of research:

Object of research - IS design processes.

Subject of research - IS development models and technologies based on fuzzy logical rules.

To achieve the goal, it is necessary to solve the following tasks:

- 1) analyze and compare methods of decision support based on fuzzy mathematics, draw conclusions and choose the most appropriate method for implementation;
- 2) formulate the task for the development of an information system for implementation;
- 3) develop an information system.

Keywords: nformation system, decision making, fuzzy logic, message structuring.

ЗМІСТ

ВСТУП

1 ПОСТАНОВКА ЗАДАЧІ Й ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

1.1 Основні елементи теорії нечітких множин та підтримки прийняття рішень.

1.2 Опис методів прийняття рішень на основі нечіткої математики.

1.2.1 Багатокритеріальний вибір альтернатив на основі перетину нечітких множин.

1.2.2 Багатокритеріальний вибір альтернатив на основі нечіткого відношення переваги.

1.2.3 Багатокритеріальний вибір альтернатив на основі нечіткого логічного виводу.

1.2.4 Багатокритеріальний вибір альтернатив на основі згортки.

1.2.5 Суб'єктивне визначення пріоритетів. Метод аналізу ієрархій.

1.2.6 Порівняльний аналіз методів.

1.2.7 Аналіз результатів.

1.3 Постановка задачі.

1.3.1 Формулювання цілей та задач проекту.

1.3.2 Вимоги до програмного забезпечення.

2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 2.1 Загальносистемні рішення.

2.1.1 Розробка варіантів використання.

2.1.2 Інформаційна модель.

2.1.3 Проект інтерфейсу користувача.

2.2 Рішення з інформаційного забезпечення.

2.2.1 Статична модель даних.

2.2.2 Динамічна модель системи.

2.3 Рішення з програмного забезпечення.

2.3.1 Вибір мови програмування.

2.3.2 Компоненти системи.

2.3.3 Розміщення.

2.3.4 Реалізація та тестування.

3 РЕЗУЛЬТАТИ РОБОТИ

4 РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ
ПРОГРАМНОГО ПРОДУКТУ

5 ОХОРОНА ПРАЦІ

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Висновки

Список використаної літератури

Додатки

Вступ

Актуальність: в призначені для збору, збереження та обробки значної кількості інформації. Обробка інформації виконується із залученням досвіду фахівців, які базуючись на власному досвіді та за допомогою спрощених математичних методів проводять оцінку загального стану системи. Для оцінювання таких систем часто використовують математичні методи нечітких множин, які пропонують нам гнучкість для міркувань, де можна розглянути усі невизначеності. Для цього системи нечіткого виводу повинні містити базу правил нечітких продукцій і реалізувати нечіткий висновок на основі умов, представлених у формі нечітких лінгвістичних висловлювань.

Об'єкт роботи: процеси, які відбуваються в системах під час надання сервісів та послуг кінцевим користувачам.

Предмет роботи: математичні методи та моделі побудови нечітких логічних правил, теорії алгебри, логіки, методи підтримки прийняття рішень і методи нечіткої логіки.

Мета роботи: підвищити ефективність роботи телекомунікаційної системи за рахунок побудови бази нечітких знань з статистичних наборів даних у вигляді логічних структурованих правил.

Для досягнення мети дослідження було поставлено та вирішено такі основні задачі:

1. Проаналізувати методи побудови баз знань.
2. Проаналізувати можливість подання баз знань, використовуючи теорію нечіткої логіки для подальшої обробки даних у форматі правил у форматі
3. Проаналізувати методи візуалізації бази знань за допомогою графів з перевіркою їх на коректність.
4. Модифікувати алгоритм узгодження та перевірки на коректність правил бази нечітких знань

1 ПОСТАНОВКА ЗАДАЧІ Й ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

1.1 Основні елементи теорії нечітких множин та підтримки прийняття рішень

Задача прийняття рішень – одна з найбільш поширених у будь-якій предметній області. Її розв’язок зводиться до вибору однієї або декількох кращих альтернатив з деякого набору. Для того, щоб зробити такий вибір, необхідно чітко визначити ціль та критерії (показники якості), за якими будуть проводитися оцінки деякого набору альтернативних варіантів. Вибір методу рішення такої задачі залежить від кількості і якості наявної інформації. Дані, необхідні для здійснення обґрунтованого вибору, можна поділити на чотири категорії: інформація про альтернативні варіанти, інформація про критерії вибору, інформація про уподобання та інформація про оточення задач.

У процесі прийняття рішень виділяють такі етапи:

- а) попередній аналіз проблеми;
- б) постановка задачі;
- в) отримання вихідних даних;
- г) розв’язок задачі прийняття рішень;
- д) аналіз та інтерпретація отриманих результатів.

Серед задач прийняття рішень виділяють задачі в умовах визначеності, задачі в умовах ризику та невизначеності.

До задач прийняття рішень в умовах визначеності відносяться задачі, для рішення яких є достатня й достовірна кількісна інформація. В даному випадку з успіхом застосовуються методи математичного програмування, суть яких складається в знаходженні оптимальних рішень на базі математичної моделі реального об’єкту.

У тих випадках, коли можливі результати можна описати за допомогою деякого імовірнісного розподілу, виникають задачі прийняття рішень в умовах ризику. Звичайно для розв’язку задач цього типу застосовують методи теорії корисності.

Задачі прийняття рішень в умовах невизначеності мають місце тоді, коли інформація, необхідна для прийняття рішень, є неточною, неповною, не кількісною, а формальні моделі досліджуваної системи або занадто складні, або відсутні. У таких випадках для рішення задачі звичайно використовуються знання експертів. На відміну від підходу, прийнятого в експертних системах, для розв'язку задачі прийняття рішення знання експертів звичайно виражені у виді деяких кількісних даних, які називаються уподобаннями.

Однією з умов існування задачі прийняття рішень є наявність декількох допустимих альтернатив, з яких слід обрати в деякому сенсі найкращу.

Задача прийняття рішень називається тривіальною, якщо вона характеризується виключно одним критерієм і усім альтернативам приписані конкретні числові оцінки у відповідності до значення вказаного критерію.

Однак на практиці часто необхідно шуканим рішенням задовольнити одразу декілька критеріїв. Причому деякі з них можуть потребувати мінімізації, а інші – максимізації. Такі задачі називаються багатокритеріальними. Дуже часто критерії протирічать один одному, наприклад, строки побудови деякого об'єкту та об'єми використаних при цьому коштів. У результаті наближення до оптимального значення по одному критерію призводить до віддалення від оптимума за другим критерієм. У зв'язку з цим ставиться задача знаходження компромісного рішення, пошук якого може бути здійснений із застосуванням методів на основі нечіткої математики. В даному випадку застосовується теорія нечітких множин шляхом використання критеріїв якісного характеру, які описуються і аналізуються в рамках цієї теорії [9, 10, 11]. За думкою основоположника теорії нечітких множин Л.А. Заде, це крок на шляху до зближення точності класичної математики й усепроникаючої неточності реального світу. Неточність реального світу тут розуміється як його лише приблизна відповідність формальним моделям, що використовуються для опису. Причина цієї неточності криється не тільки в обмеженості сучасних знань про світ, але й у відсутності повної адекватності використовуваних для опису світу понять і моделей. Реальні явища, як правило, характеризуються нечітким описом

і розмитими межами, а це означає, що й моделі і поняття, які їх відображують, теж повинні бути нечіткими. Людське мислення враховує цей факт, використовуючи у своїх судженнях якісні нечіткі поняття: «великий», «малий», «добрий», «поганий» тощо, а також припускаючи паралельні думки (альтернативні лінії мислення). Однак у розглянутих формальних моделях, що використовуються при класичних підходах, це не знаходить свого відображення. Розглянемо її основні положення.

У традиційній прикладній математиці множина розуміється як сукупність елементів, яким присутня певна спільна властивість.

Поняття нечіткої множини – це варіант математичної формалізації нечіткої інформації. У основі цього поняття лежить представлення про те, що складові для даної множини елементи, які мають спільну властивість, можуть мати цю властивість у різному ступені і, як наслідок, належати до даної множини у різному ступені.

Нехай X - звичайна множина елементів.

Нечіткою множиною C в X називається сукупність пар виду $(x, \mu_C(x))$, де $x \in X$, а μ_C - функція, яка задана на множині X і приймає значення з інтервалу $[0, 1]$, тобто $\mu_C : X \rightarrow [0, 1]$. Функція μ_C називається функцією приналежності нечіткої множини C . Значення цієї функції для конкретного x називається ступенем приналежності цього елементу нечіткій множині C .

Звичайно функція μ_C назначається експертом. Те, що будь-яка нечітка множина C завжди є підмножиною деякої звичайної множини X , стало причиною для того, що іноді нечітку множину називають нечіткою підмножиною. Якщо $\mu_C(x) = 1$ для будь-якого $x \in X$, то C - звичайна множина, яка співпадає з X .

Нечітка множина задається переліченням пар $(x | \mu_C(x))$.

Розглянемо звичайну скінченну множину $X : X = \{a, b, c, d, e, f\}$. Задамо на ній нечітку підмножину $C = \{(a | 0), (b | 1), (c | 0,5), (d | 0), (e | 0,5), (f | 0)\}$.

Іноді проглядає подібність між функцією приналежності та імовірністю. Ця подібність у значній мірі зовнішня і обмежується спільністю області значень.

Дійсно, трактувати як імовірність ступінь приналежності, наприклад, конкретної людини зростом 180см до групи високих людей некоректно, так як факт приналежності не є випадковою подією. Як наслідок, немає необхідності в умові нормування для значень функції приналежності (в теорії ймовірностей для подій, що складають повну групу, сума ймовірностей повинна бути рівною одиниці). Наприклад, нехай N - множина натуральних чисел. Задамо нечітку множину C «невеликих» натуральних чисел: $C = \{(0 | 1), (1 | 0,8), (2 | 0,6), (3 | 0,4), \dots\}$. Нечітка множина може бути описаною графічно у вигляді графіка функції приналежності (рис. 1.1).

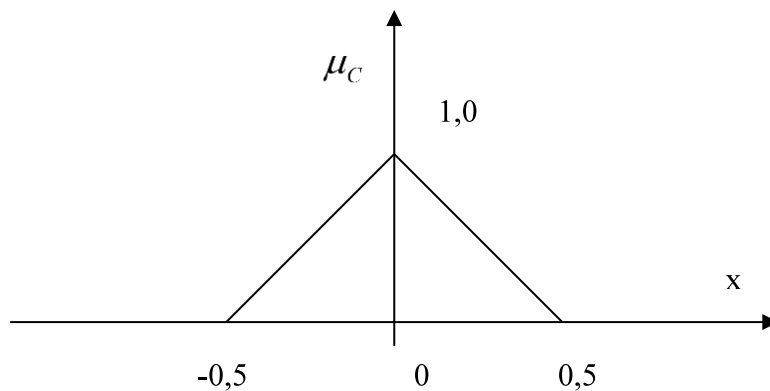


Рисунок 1.1 – Графічне зображення множини дійсних чисел, близьких до нуля. Вид функції приналежності відображає судження експерта, який її назначає, контекст, в якому вона використовується. Так, наприклад, у звичайному житті чоловік зростом 1,8м буде признаний високим, а серед баскетболістів – низьким. Якщо A та B - нечіткі множини в X , а $\mu_A(x)$ та $\mu_B(x)$ - їх функції приналежності відповідно, то кажуть, що A включає в себе B ($B \subseteq A$), якщо для будь-якого $x \in X$ виконується нерівність $\mu_B(x) \leq \mu_A(x)$.

Об'єднанням нечітких множин A та B в X називається нечітка множина $A \cup B$ з функцією приналежності виду

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad (1.7)$$

де $x \in X$.

Пересіченням нечітких множин A та B в X називається нечітка множина $A \cap B$ з функцією приналежності виду

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (1.8)$$

де $x \in X$.

Нечітким відношенням R на множині X називається нечітка підмножина декартового добутку $X \times X$, яка характеризується функцією приналежності $\mu_R : X \times X \rightarrow [0,1]$.

Часним випадком відношення є функціональне відношення або функція.

З поняттям «нечітка множина» тісно зв'язане поняття «лінгвістична змінна», тобто змінна, яка приймає «лінгвістичні» значення, тобто виражені на природній мові. Людина у своїй мові дуже часто використовує лінгвістичні змінні, наприклад, «учень – слабкий» або «учень – сильний». У цих виразах лінгвістична змінна «учень» набуває лінгвістичних значень «слабкий» та «сильний». Однак у понятті «лінгвістичної змінної» додатково з кожним її лінгвістичним значенням зв'язується нечітка множина, ступінь приналежності до якої задається на деякій базовій множині $V = \{v\}$.

Одним з методів, де використовуються ці поняття, є метод нечітких множин.

Нехай з m експертів n_1 на питання про приналежність елемента $x \in X = \{x_1, x_2, \dots, x_m\}$ відповідає позитивно, а інша частина експертів ($n_2 = m - 1$) – негативно. Тоді ФП визначається, як

$$\mu(x) = \frac{n_1}{(n_1 + n_2)} \quad (1.1)$$

де n_1 - кількість експертів, які відповіли на питання позитивно, n_2 - кількість експертів, які відповіли на питання негативно.

Будується функція приналежності, яка дозволяє приймати рішення про вибір кількості робітників.

1.2 Опис методів прийняття рішень на основі нечіткої математики

1.2.1 Багатокритеріальний вибір альтернатив на основі перетину нечітких множин

Нехай маємо m альтернатив (стратегій) $A_1, A_2, \dots, A_m$, серед яких потрібно здійснити вибір, і множину критеріїв $C_1, C_2, \dots, C_n$. При цьому оцінки альтернатив по кожному j -му критерію представлені нечіткою множиною $C_j = \{A_1 | \mu_{C_j}(A_1), A_2 | \mu_{C_j}(A_2), \dots, A_m | \mu_{C_j}(A_m)\}$.

Правило вибору найкращої альтернативи можна представити як перетин нечітких множин, які відповідають усім критеріям:

$$D = C_1 \cap C_2 \cap \dots \cap C_n \quad (1.9)$$

При цьому кожна альтернатива A_i буде належати до нечіткої множини D зі ступенем приналежності

$$\mu_D(A_i) = \min_j \mu_{C_j}(A_i) \quad (1.10)$$

де $i = 1, 2, \dots, m$.

Найкращою вважається альтернатива A^* , яка має найбільше значення функції приналежності, тобто:

$$\mu_D(A^*) = \max_i \mu_D(A_i) \quad (1.11)$$

Якщо критерії C_j мають різну важливість, то їх внесок у загальне рішення можна представити як зважений перетин:

$$D = C_1^{\alpha_1} \cap C_2^{\alpha_2} \cap \dots \cap C_n^{\alpha_n} \quad (1.12)$$

де α_j - вагові коефіцієнти, які повинні задовольняти наступним умовам:

$$\alpha_j \geq 0; j = 1, 2, \dots, n; \frac{1}{n} \sum_{j=1}^n \alpha_j = 1 \quad (1.13)$$

1.2.2 Багатокритеріальний вибір альтернатив на основі нечіткого відношення переваги

Нехай задано множину альтернатив A і кожна альтернатива характеризується декількома критеріями якості з номерами $j = 1, \dots, m$. Інформація про попарне порівняння альтернатив за кожним критерієм якості j представлена у формі відношення переваги R_j на множині A . Потрібно обрати найкращу альтернативу з множини $\{A, R_1, \dots, R_m\}$.

Метод багатокритеріального вибору альтернатив на основі нечіткої переваги заснований на ряді означень.

Нечітким відношенням переваги на множині A називається будь-яке задане на цій множині рефлексивне нечітке відношення, функція приналежності якого обчислюється наступним шляхом:

$$\mu_{R^x}(a, b) = \begin{cases} \mu_R(a, b) - \mu_R(b, a), & \mu_R(a, b) \geq \mu_R(b, a); \\ 0, & \mu_R(a, b) < \mu_R(b, a) \end{cases} \quad (1.14)$$

Нехай A - множина альтернатив і μ_R - задане на ній нечітке відношення переваги. Нечітка підмножина невідомінованих альтернатив множини (A, μ_R) описується функцією приналежності (1.15).

$$\mu_R^{HD} - 1 - \sup_{a, b \in A} (\mu_{R^N}(b, a) - \mu_{R^N}(a, b)), a \in A \quad (1.15)$$

Чітко невідомінованими називаються альтернативи, для яких $\mu_R^{HD}(a) = 1$, а множина таких альтернатив

$$A^{HD} = \{a \mid a \in A, \mu_R^{HD}(a) = 1\} \quad (1.16)$$

Носієм нечіткої множини B з функцією приналежності $\mu_B(a) \in$ множина $\{a \mid a \in A, \mu_B > 0\}$.

Процедура рішення задачі вибору виконується у декілька етапів:

- будується нечітке відношення Q_1 , яке є пересічення вихідних відношень переваг:

$$\mu_{Q_1}(a, b) = \min(\mu_{R_1}(a, b), \dots, \mu_{R_m}(a, b)) \quad (1.17)$$

та визначається нечітка підмножина невідомінованих альтернатив у множині (A, μ_{Q_1}) :

$$\mu_{Q_1}^{HD}(a) = 1 - \sup_{b \in A} (\mu_{Q_1}(b, a) - \mu_{Q_1}(a, b)) \quad (1.18)$$

- будується нечітке відношення Q_2 :

$$\mu_{Q_2}(a, b) = \sum_{j=1}^m w_j \mu_{R_j}(a, b) \quad (1.19)$$

і визначається нечітка підмножина невідомінованих альтернатив у множині (A, μ_{Q_2}) :

$$\mu_{Q_2}^{HD}(a) = 1 - \sup_{b \in A} (\mu_{Q_2}(b, a) - \mu_{Q_2}(a, b)) \quad (1.20)$$

Дана функція впорядковує альтернативи за ступенем їх недомінованості. Числа w_j у наведеній згортці представляють собою коефіцієнти відносної важливості критеріїв, що розглядаються, для яких виконуються наступні умови:

$$\sum_{j=1}^m w_j = 1, w_j \geq 0, j = \overline{1, m} \quad (1.21)$$

- відшукується пересічення множин $\mu_{Q_1}^{HD}$ і $\mu_{Q_2}^{HD}$:

$$\mu^{HD}(a) = \min(\mu_{Q_1}^{HD}(a), \mu_{Q_2}^{HD}(a)) \quad (1.22)$$

- раціональним вважається вибір альтернатив із множини

$$A^{HD} = \{a' | a' \in A, \mu^{HD}(a') = \sup_{a \in A} \mu^{HD}(a)\} \quad (1.23)$$

Найбільш раціональною альтернативою із множини A^{HD} є та, яка має максимальну ступінь недомінованості.

1.2.3 Багатокритеріальний вибір альтернатив на основі нечіткого логічного виводу

На поняттях нечіткої множини та лінгвістичної змінної заснована процедура нечіткого логічного виводу, яка використовується для прийняття рішень у багатокритеріальних задачах, коли інформація про комплексний компромісний критерій задана у вигляді нечітких правил. Подібні процедури реалізуються звичайно на основі технології експертних систем (ЕС).

Нечітким правилом називають запис вигляду «Якщо «умова», То «висновок»», де «умова» та «висновок» виражені за допомогою лінгвістичних змінних, наприклад «Якщо «ціна велика» і «попит малий», То «оборот малий»».

При цьому з кожним лінгвістичним значенням змінних «ціна», «попит» та «оборот» зіставляється відповідна нечітка множина зі своєю функцією приналежності, заданою на своїй базовій множині. Нечіткі правила формують базу знань. Важливо, що в нечіткій ЕС на відміну від традиційної при будь-яких вихідних даних працюють усі правила одночасно. Це відбувається тому, що усі значення кожної лінгвістичної змінної характеризуються в загальному вигляді

ненульовим ступенем приналежності. Однак ступінь впливу різних правил на результат може бути різною. Принцип обчислення суперпозиції багатьох впливів на кінцевий результат лежить у основі нечітких ЕС.

Процес обробки нечітких правил виводу в ЕС складається з чотирьох етапів:

- обчислення ступеня істиності лівих частин правил – визначення ступеня приналежності вхідних значень нечітким множинам, вказаним в умовах;
- модифікація нечітких множин, вказаних у правих частинах правил, у відповідності до значень істиності, отриманих на першому етапі;
- об'єднання (суперпозиція) модифікованих множин;
- скаляризація результату суперпозиції – перехід від нечітких множин до скалярних значень.

1.2.4 Багатокритеріальний вибір альтернатив на основі згортки

У цьому методі експертні уподобання представлені за допомогою нечітких чисел, які мають функції приналежності трикутного вигляду (рис. 1.2).

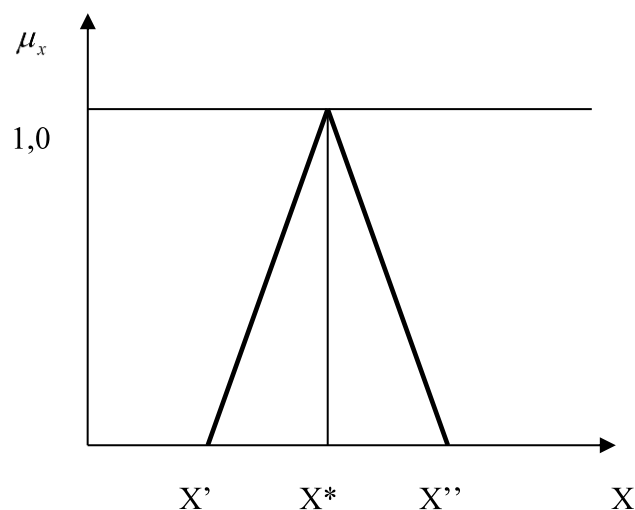


Рисунок 1.2 – Трикутне представлення нечіткого числа

Нехай маємо підмножину альтернатив $A = \{a_1, a_2, \dots, a_m\}$ і множину критеріїв $C = \{c_1, c_2, \dots, c_n\}$, при цьому оцінка j -ї альтернативи по i -му критерію представлена нечітким числом R_{ij} , а відносна важливість i -го критерію задається

коефіцієнтом $a_i = 1, 2, \dots, n$. Якщо коефіцієнти a нормовані, то зважена оцінка j -ї альтернативи обчислюється за формулою

$$R_j = \sum_{i=1}^n a_i R_{ij} \quad (1.24)$$

Якщо функції приналежності $\mu_{R_{ij}}(r_{ij})$ та $\mu_{a_{ij}}(a_i)$ мають трикутний вигляд, то для них, як і для нечіткого числа X , вершина X' , а також ліва X^* і права X'' границі визначаються наступними співвідношеннями:

$$\begin{aligned} \forall \delta : \mu(X' - \delta) = 0; \mu(X' + \delta) \neq 0; \\ \forall \delta : \mu(X'' - \delta) \neq 0; \mu(X'' + \delta) = 0; \mu(X^*) = 1 \end{aligned} \quad (1.25)$$

Зважена оцінка j -ї альтернативи R_j є результатом лінійної комбінації нечітких чисел і також буде мати функцію приналежності трикутного вигляду. Вершину і границі нечіткого числа $Z = X'Y$, отриманого в результаті операцій складання або множення (символ $'$ означає узагальнюючу операцію), можна обчислити наступним шляхом:

$$Z' = X'Y'; Z'' = X''Y''; Z^* = X^*Y^* \quad (1.26)$$

Ранжування альтернатив з використанням отриманих зважених оцінок можливе на основі їх нечіткої композиції:

$$\mu_j(j) = \sup_{r_1, r_2, \dots, r_m : r_2 \geq r_j} \min_{j=1, \dots, m} \mu_{R_j}(r_j) \quad (1.27)$$

де $\mu_j(j)$ - нечітка множина альтернатив, які відповідають поняттю «краща альтернатива». Кращою вважається альтернатива, яка має найбільше значення $\mu_j(j)$.

Пріоритет кожної альтернативи обчислюється шляхом вибору мінімуму серед точок перетину правої границі відповідного їй нечіткого числа R_j з межами нечітких чисел, які представляють зважені оцінки альтернатив, що розміщуються праворуч на числовій вісі (задовольняють умові $r_k > r_j$). При цьому мається на увазі, що права границя області визначення нечітких чисел відповідає тим оцінкам, що мають найбільшу перевагу, а ліва – найгіршим.

1.2.5 Суб'єктивне визначення пріоритетів. Метод аналізу ієрархій

Як правило, критерії, які використовуються для прийняття рішень у багатокритеріальних задачах, характеризуються різним ступенем важливості в контексті розглядуваної задачі. Для врахування цього факту кожному з критеріїв назначається ступінь пріоритетності. Звичайно це роблять експерти на основі свого особистого досвіду і, звичайно, їх судження в певній мірі виявляється суб'єктивним. Не слід забувати, що оптимальні рішення, вибрані з використанням суб'єктивних оцінок, також будуть суб'єктивними. Ступінь суб'єктивності рішення можна зменшити, якщо замість суб'єктивних оцінок, назначених однією особою, увести середнє з таких оцінок, назначених незалежно один від одного групою експертів. Метод опитування експертів широко застосовується у сучасній науці. Досвід застосування подібних методів вчить, що часто оцінки експертів виявляються далеко не такими різними, як це могло передбачатися. Вивести з них деякі передумови для прийняття розумного рішення досить можливо. Одним із таких підходів є метод аналізу ієрархій.

Метод аналізу мереж дозволяє вирішувати широкий клас задач, працювати з багатокритеріальними проблемами з взаємним впливом критеріїв і альтернатив, що погано формалізуються. Він дозволяє отримати оцінки пріоритетності усіх елементів сітьової структури відносно заданої цілі при наявності зв'язків. Елементи задач прогнозування в даному підході об'єднуються в кластери, між якими можливі довільні зв'язки. Формування кластерів і зв'язків між ними є неформальною процедурою і здійснюється на основі конкретних знань про специфіку вирішуваної задачі. Об'єднання елементів у кластери дозволяє знизити розмірність задачі й покращити узгодженість суджень [4]. Допускаються різниці в думках і конфлікти, як це трапляється в реальному світі.

З цих причин цей метод було обрано для поглибленого дослідження і використання в роботі.

Важливим моментом в методі аналізу мереж є формулювання головної цілі задачі й питань, які слід задати експертам при заповненні матриць попарних порівнянь. Для оцінки впливу кластерів один на одного також будуються матриці попарних порівнянь, а отримані після їх обробки пріоритети після цього в якості

вагових коефіцієнтів при розрахунку пріоритетів впливу елементів кластерів. Елементи кластерів порівнюються за всіма елементами кластерів, що впливають на них. Отримані вектори пріоритетів записуються у суперматрицю. Зазначимо поняття, необхідні для описання алгоритму рішення задач, що мають мережеву структуру.

Квадратна матриця називається такою, що приводиться, якщо вона може бути перетворена шляхом перестановок до діагонального або трикутного виду; в іншому випадку матриця є такою, що не приводиться.

Невід'ємна матриця W , що не приводиться, називається примітивною тоді й тільки тоді, коли існує ціле число $m \geq 1$, при якому $W^m > 0$. Примітивна матриця має єдиний головний власний вектор.

Суперматриця називається власною, якщо в нормальній формі її елементи представляють собою примітивні матриці [4].

Для розв'язання задачі будується ієрархія – система, в якій рівні розміщені й пронумеровані так, що найнижчий рівень містить альтернативи, що рейтинуються, й вузли рівнів з більшими номерами можуть домінувати над вузлами з меншими номерами. Таким чином, в ієрархії зв'язки визначають шляхи від вершини до альтернатив через проміжні рівні, які складаються з вузлів-факторів.

Формування ієрархії є складним процесом, який може призводити до різних результатів. Цей процес потребує знань і розгляду множини варіантів. Для забезпечення такої можливості необхідні комп'ютерна та інформаційна підтримки.

Визначаються пріоритети вузлів – позитивні числа, які слугують для кількісного вираження важливості (ваги, значущості, пріоритетності тощо) даного вузла в кластері відносно інших вузлів кластера по відношенню до критерію, розміщеного у вершині кластера. Сума усіх пріоритетів вузлів кластера дорівнює одиниці. Тому часто пріоритети можна трактувати як імовірності, долі цілого ресурсу тощо. Для задання пріоритетів використовуються парні порівняння – оцінки пріоритета одного вузла до пріоритета іншого. Для вираження оцінки

попарних порівнянь використовується шкала порівнянь, яка може бути якісною або кількісною.

По пріоритетах будується матриця порівнянь – таблиця числових значень парних порівнянь.

Із групи матриць парних порівнянь формується набір локальних пріоритетів, які виражають відносний вплив множини елементів на елемент вищого рівня. Знаходиться відносна цінність кожного об'єкту через «рішення» матриць, кожна з яких має оберненосиметричні властивості. Для цього необхідно розрахувати множину власних векторів для кожної матриці, а після цього нормалізувати результат до одиниці, отримуючи тим самим вектор пріоритетів.

Одним із найкращих шляхів розрахунку пріоритетів є геометричне середнє:

$$\begin{array}{c|ccccccc}
 & A_1 & A_2 & A_3 & A_4 & & & \\
\hline
A_1 & \frac{w_1}{w_1} & \frac{w_1}{w_2} & \frac{w_1}{w_3} & \frac{w_1}{w_4} & \rightarrow & \sqrt[4]{\frac{w_1}{w_1} \times \frac{w_1}{w_2} \times \frac{w_1}{w_3} \times \frac{w_1}{w_4}} & = a \quad \frac{a}{\sum} = x_1 \\
A_2 & \frac{w_2}{w_1} & \frac{w_2}{w_2} & \frac{w_2}{w_3} & \frac{w_2}{w_4} & \rightarrow & \sqrt[4]{\frac{w_2}{w_1} \times \frac{w_2}{w_2} \times \frac{w_2}{w_3} \times \frac{w_2}{w_4}} & = b \quad \frac{b}{\sum} = x_2 \\
A_3 & \frac{w_3}{w_1} & \frac{w_3}{w_2} & \frac{w_3}{w_3} & \frac{w_3}{w_4} & \rightarrow & \sqrt[4]{\frac{w_3}{w_1} \times \frac{w_3}{w_2} \times \frac{w_3}{w_3} \times \frac{w_3}{w_4}} & = c \quad \frac{c}{\sum} = x_3 \\
A_4 & \frac{w_4}{w_1} & \frac{w_4}{w_2} & \frac{w_4}{w_3} & \frac{w_4}{w_4} & \rightarrow & \sqrt[4]{\frac{w_4}{w_1} \times \frac{w_4}{w_2} \times \frac{w_4}{w_3} \times \frac{w_4}{w_4}} & = d \quad \frac{d}{\sum} = x_4
\end{array} \quad (1.2)$$

де A_i - альтернативи, $\frac{w_i}{w_j}$ - вираження попарних оцінок, x - вектор пріоритетів.

У розрахунку локальних пріоритетів обчислюють індекс узгодженості (ІУ), який дає інформацію про ступінь порушення чисельної (кардинальної, $a_{ij}a_{jk} = a_{ik}$) і транзитивної (порядкової) узгодженості. Для покращення узгодженості можна рекомендувати пошук додаткової інформації і перегляд даних, використаних при побудові шкали. Відсутність узгодженості може бути серйозним обмежуючим фактором для дослідження деяких проблем, але не бути таким для інших. Однак абсолютної узгодженості навіть із найбільш точними інструментами важко досягти на практиці. Слід зазначити, що між достовірністю та непротиріччям порівнянь немає ніякого зв'язку. Протиріччя в порівняннях виникають через

суб'єктивні оцінки експертів. Індекс узгодженості не залежить від шкал порівнянь, але залежить від кількості парних порівнянь. Індекс узгодженості – позитивне число, чим менше протиріч у порівняннях, тим він менший.

Для обчислення індексу узгодженості в кожній матриці спочатку сумується кожний стовбець суджень, потім сума кожного i -го стовпця помножується на відповідну величину нормалізованого вектора пріоритетів x_i , після чого отримані значення сумуються. Таким чином отримується величина, яка позначається $\lambda_{\max}$. Для індексу узгодженості маємо

$$IY = \frac{\lambda_{\max} - n}{n - 1}, \quad (1.3)$$

де n - число елементів, що порівнюються. Для оберненосиметричної матриці завжди $\lambda_{\max} \geq n$.

Потім ця величина порівнюється з тією, яка була б отримана при випадковому виборі кількісних суджень зі шкали $1/9, 1/8, 1/7, \dots, 1, 2, \dots, 9$, але створенні оберненосиметричної матриці (табл. 1.3).

Таблиця 1.3 – Середні узгодженості для випадкових матриць різного порядку

Розмір матриці	1	2	3	4	5	6	7	8	9	10
Випадкова узгодженість	0	0	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

Після цього може бути обчислене відношення узгодженості

$$BY = \frac{IY}{BunY}, \quad (1.4)$$

де $BunY$ – випадкова узгодженість.

Величина ВУ повинна бути порядку 10% або менше, щоб бути прийнятною. В деяких випадках можна допустити 20%, але не більше. Якщо ВУ виходить з цих меж, то учасникам треба дослідити задачу та перевірити свої судження.

Можна зробити висновок, що для проведення обґрунтованих обчислень треба порівнювати не більше, ніж 7 ± 2 елементи. В такому випадку маленька похибка

в кожній відносній величині змінює її не дуже значно. Якщо виникає необхідність порівняти більше елементів, то це слід робити шляхом ієрархічної декомпозиції [8].

1.2.6 Порівняльний аналіз методів

Розглянемо підходи до розв'язання однієї задачі багатокритеріального вибору в умовах невизначеності з використанням різних методів. Для цього будемо використовувати вихідну інформацію, отриману від одного й того ж самого кваліфікованого експерта [13].

Вирішується задача вибору інвестором одного з трьох інвестиційних проектів: a_1 – проект технологічного комплексу для термінового знешкодження й переробки відходів; a_2 – проект заводу з виробництва аерозольних вогнегасників третього покоління; a_3 – проект створення інвестиційно-фінансової компанії.

В даному випадку головною ціллю ОПР є вибір раціонального інвестиційного проекту. При цьому об'єм фінансових ресурсів інвестора обмежений, він не схильний до великого ризику і має не найгірші надії на майбутнє. Для вибору сформований такий набір критеріїв: c_1 – рентабельність; c_2 – оцінка можливих ринків збуту; c_3 – початкові засоби, c_4 – виробничий ризик; c_5 – інвестиційний ризик. Раціональний вибір пов'язаний з прагненням отримати рішення, яке характеризується прийнятними оцінками за всіма критеріями, хоча їх значимість для ОПР може бути різною.

Вирішимо задачу методом максимінної згортки. Оцінки альтернатив за заданими критеріями представлені наступними нечіткими множинами:

$$\begin{aligned}\mu_{c_1}(a) &= \{0,5/a_1; 0,1/a_2; 1/a_3\}; \\ \mu_{c_2}(a) &= \{0,7/a_1; 0,5/a_2; 0,2/a_3\}; \\ \mu_{c_3}(a) &= \{0,3/a_1; 0,5/a_2; 1/a_3\}; \\ \mu_{c_4}(a) &= \{0,5/a_1; 0,3/a_2; 0,9/a_3\}; \\ \mu_{c_5}(a) &= \{0,6/a_1; 0,4/a_2; 0,2/a_3\};\end{aligned}\quad (1.5)$$

Вагові коефіцієнти важливості критеріїв, що розглядаються, визначені з використанням процедури попарного порівняння і мають наступні значення:

$$b = \{1,78; 0,68; 0,67; 0,22; 1,65\} \quad (1.6)$$

Найбільш важливими критеріями для ОПР є рентабельність та інвестиційний ризик, значно більш важливим – розмір вкладень і оцінка ринків, найменшою вагою наділений виробничий ризик.

Найкраща альтернатива визначається наступним способом:

$$\begin{aligned} \mu_D(a^*) &= \max\{\min\{0,5^{1,78}; 0,7^{0,68}; 0,3^{0,67}; 0,5^{0,22}; 0,6^{1,65}\}; \\ &\min\{0,1^{1,78}; 0,5^{0,68}; 0,5^{0,67}; 0,3^{0,22}; 0,4^{1,65}\} \\ &\min\{1^{1,78}; 0,2^{0,68}; 1^{0,67}; 0,9^{0,22}; 0,2^{1,65}\}\} \end{aligned} \quad (1.7)$$

Множина оптимальних альтернатив має вид:

$$\mu_D(a) = \{0,781/a_1; 0,432/a_2; 0,588/a_3\} \quad (1.8)$$

Максимальне значення належить a_1 , на другому місці знаходиться a_3 , найгіршою альтернативою буде a_2 .

Розв'яжемо задачу з використанням методу відношень уподобань. На основі функцій приналежності побудовані наступні відношення уподобань на множині альтернатив (1.9).

$$\begin{aligned} \mu_{R_1} &= \begin{array}{c|ccc} c_1 & a_1 & a_2 & a_3 \\ \hline a_1 & 1 & 0,4 & 0 \\ a_2 & 0 & 1 & 0 \\ a_3 & 0,5 & 0,9 & 1 \end{array}; & \mu_{R_2} &= \begin{array}{c|ccc} c_2 & a_1 & a_2 & a_3 \\ \hline a_1 & 1 & 0,2 & 0,5 \\ a_2 & 0 & 1 & 0,3 \\ a_3 & 0 & 0 & 1 \end{array} \\ \mu_{R_3} &= \begin{array}{c|ccc} c_3 & a_1 & a_2 & a_3 \\ \hline a_1 & 1 & 0 & 0 \\ a_2 & 0,2 & 1 & 0 \\ a_3 & 0,7 & 0,5 & 1 \end{array}; & \mu_{R_4} &= \begin{array}{c|ccc} c_4 & a_1 & a_2 & a_3 \\ \hline a_1 & 1 & 0,2 & 0 \\ a_2 & 0 & 1 & 0 \\ a_3 & 0,4 & 0,6 & 1 \end{array} \\ \mu_{R_5} &= \begin{array}{c|ccc} c_5 & a_1 & a_2 & a_3 \\ \hline a_1 & 1 & 0,2 & 0,4 \\ a_2 & 0 & 1 & 0,2 \\ a_3 & 0 & 0 & 1 \end{array} \end{aligned} \quad (1.9)$$

Множина недомінуємих альтернатив $\mu^{HD}_{Q_2} = \|111\|$. Значення нормованих на одиницю вагових коефіцієнтів критеріїв задані вектором $w = \{0,36; 0,14; 0,13; 0,05; 0,32\}$. Обчислимо нечітке відношення Q_2 :

$$\mu_{Q_2}(a_i, a_j) = \begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline a_2 & 1 & 0,246 & 0,121 \\ a_2 & 0,026 & 1 & 0,106 \\ a_3 & 0,291 & 0,419 & 1 \end{array} \quad (1.10)$$

Знаходимо підмножину невідомісних альтернатив множини $\{A, \mu_{Q_2}\} : \mu^{HD}_{Q_2}(a_i) = \|0,83; 0,59; 1\|$. Результируюча множина невідомісних альтернатив – це перетин множин $\mu_{Q_2} \cap \mu^{HD}_{Q_2}$:

$$\mu^{HD}_{Q_1} \cap \mu^{HD}_{Q_2} = \{(1;1;1) \cap (0,83;0,69;1)\} = \{(0,83;0,69;1)\}$$

Отже, раціональним слід вважати вибір альтернативи a_3 , яка має максимальний ступінь недопустимості.

Розв'яжемо задачу з використанням нечіткого логічного виводу.

Засновуючись на приведених вхідних даних про критерії та альтернативи, експертом сформовані правила:

- d_1 : «Якщо c_1 = ВИСОКА і c_2 = ДОБРА і c_3 = ПРИЙНЯТНІ, то Y = ЗАДОВІЛЬНИЙ»;
- d_2 : «Якщо c_1 = ДУЖЕ ВИСОКА і c_2 = ДОБРА і c_3 = ПРИЙНЯТНІ і c_4 = НИЗЬКИЙ і c_5 = ДУЖЕ НИЗЬКИЙ, то Y = БЕЗДОГАННИЙ»;
- d_3 : «Якщо c_1 = НИЗЬКА і c_2 = ПОГАНА і c_3 = ВИСОКИЙ, то Y = НЕЗАДОВІЛЬНИЙ».

Змінна Y задана на множині

$$J = \{0; 0,1; 0,2; \dots; 1\} \quad (1.11)$$

Значення змінної Y задані за допомогою наступних функцій приналежності:

S = ЗАДОВІЛЬНИЙ, $\mu_S(x) = x, x \in J$;

P = БЕЗДОГАННИЙ, $\mu_P(x) = \begin{cases} 1, & x = 1 \\ 0, & x < 1 \end{cases} x \in J$;

US = НЕЗАДОВІЛЬНИЙ, $\mu_{US}(x) = 1 - x, x \in J$.

У розглядуваній задачі оцінки інвестиційний проектів задані наступними нечіткими множинами:

- ВИСОКА (рентабельність), $A = \{0,5 | a_1; 0,1 | a_2; 1 | a_3\}$;

- ДУЖЕ ВИСОКА (рентабельність), $\mu_{A_1}(a) = \mu_A^2(a)$;
- НИЗЬКА (рентабельність), $\mu_{A_2}(a) = 1 - \mu_A(a)$;
- ДОБРА (оцінка ринків збуту), $B = \{0,7 \mid a_1; 0,5 \mid a_2; 0,2 \mid a_3\}$;
- ПОГАНА (оцінка ринків збуту), $\mu_{B_1}(a) = 1 - \mu_B(a)$;
- ПРИЙНЯТНІ (первинні засоби), $G = \{0,3 \mid a_1; 0,5 \mid a_2; 1 \mid a_3\}$;
- НИЗЬКИЙ (виробничий ризик), $D = \{0,5 \mid a_1; 0,3 \mid a_2; 0,9 \mid a_3\}$;
- НИЗЬКИЙ (інвестиційний ризик), $E = \{0,6 \mid a_1; 0,4 \mid a_2; 0,2 \mid a_3\}$;
- ДУЖЕ НИЗЬКИЙ (інвестиційний ризик), $\mu_{E_1}(a) = \mu_E^2(a)$;
- ВИСОКИЙ (інвестиційний ризик), $\mu_{E_2}(a) = 1 - \mu_E(a)$.

Додаткові градації лінгвістичних оцінок (зі словом ДУЖЕ) призначені для врахування найбільш важливих критеріїв. У даному випадку це рентабельність (с1) і інвестиційний ризик (с5).

З урахуванням введених позначень правила $d_1, \dots, d_3$ набувають вигляду:

- d_1 : «Якщо $(c = A \text{ і } B) \text{ і } G$, то $Y = S$ »;
- d_2 : «Якщо $(c = A_1 \text{ і } B) \text{ і } G \text{ і } D \text{ і } E_1$, то $Y = P$ »;
- d_3 : «Якщо $(c = A_2 \text{ і } B_2) \text{ і } E_2$, то $Y = US$ ».

Функції приналежності μ_{M_i} для лівих частин наведених правил мають вигляд:

$$\begin{aligned}
 d_1 : \mu_{M_1}(a) &= \min(\mu_A(a), \mu_B(a), \mu_G(a)); \\
 M_1 &= \{0,3 \mid a_1; 0,1 \mid a_2; 0,2 \mid a_3\}; \\
 d_2 : \mu_{M_2}(a) &= \min(\mu_A^2(a), \mu_B(a), \mu_G(a), \mu_D(a), \mu_E^2(a)); \\
 M_2 &= \{0,25 \mid a_1; 0,01 \mid a_2; 0,2 \mid a_3\}; \\
 d_3 : \mu_{M_3}(a) &= \min(1 - \mu_A(a), 1 - \mu_B(a), 1 - \mu_E(a)); \\
 M_3 &= \{0,3 \mid a_1; 0,5 \mid a_2; 0 \mid a_3\}.
 \end{aligned} \tag{1.12}$$

Правила набудуть наступного вигляду:

- d_1 : Якщо $c = M_1$, то $Y = S$;
- d_2 : Якщо $c = M_2$, то $Y = P$;
- d_3 : Якщо $c = M_3$, то $Y = US$;

Використовуючи для правил імплікацію Лукасевича, отримаємо нечіткі відношення $D_1, \dots, D_3$ на $U \times J$ і в результаті їх перетину функціональний розв'язок D :

$$D = \begin{array}{c|cccccccccccc} & 0 & 0,1 & 0,2 & 0,3 & 0,4 & 0,5 & 0,6 & 0,7 & 0,8 & 0,9 & 1 \\ \hline a_1 & 0,7 & 0,5 & 0,75 & 0,75 & 0,75 & 0,75 & 0,75 & 0,75 & 0,75 & 0,75 & 0,7 \\ a_2 & 0,9 & 0,99 & 0,99 & 0,99 & 0,99 & 0,99 & 0,99 & 0,8 & 0,7 & 0,6 & 0,5 \\ a_3 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 0,8 & 1 \end{array} \quad (1.13)$$

Для альтернатив обчислені наступні точкові оцінки:

$$\begin{aligned} F(a_1) &= 0,500; \\ F(a_2) &= 0,431; \\ F(a_3) &= 0,600 \end{aligned} \quad (1.14)$$

Максимальну оцінку дає третя альтернатива, отже, вона є найбільш бажаною.

Розв'яжемо задачу методом адитивної згортки.

Важливість критеріїв була задана нечіткими числами з функцією приналежності наступного виду:

- ВАЖЛИВИЙ (В) - $\mu_B = \{0,4;1 \mid 0,7;0 \mid 1\}$
- ДУЖЕ ВАЖЛИВИЙ (ДВ) - $\mu_{DB} = \{0 \mid 0,7;1 \mid 1\}$
- НЕ ДУЖЕ ВАЖЛИВИЙ (В) - $\mu_{HDB} = \{0 \mid 0,1;1 \mid 0,4;0 \mid 7\}$

Для оцінки альтернатив використовувались лінгвістичні значення:

- ДУЖЕ НИЗЬКИЙ (ДН) - $\mu_{DH} = \{1 \mid 0;0 \mid 0,3\}$
- НИЗЬКИЙ (Н) - $\mu_{DH} = \{0 \mid 0,1;1 \mid 0,3;0 \mid 0,5\}$
- СЕРЕДНІЙ (С) - $\mu_C = \{0 \mid 0,3;1 \mid 0,5;0 \mid 0,7\}$
- ВИСОКИЙ (В) - $\mu_B = \{0 \mid 0,5;1 \mid 0,75;0 \mid 1\}$
- ДУЖЕ ВИСОКИЙ (ДВ) - $\mu_{DB} = \{0 \mid 0,75;1 \mid 1\}$

Альтернативи отримали наступні оцінки за критеріями:

	a_1	a_2	a_3	B	
c_1	C	DH	DB	B	
c_2	B	C	H	DB	
c_3	H	C	DB	B	
c_4	C	H	DB	HDB	
c_5	B	C	DB	B	

(1.15)

Зважені оцінки альтернатив R_i мають наступні функції приналежності:

$$\begin{aligned}\mu_{R_1} &= \{0 \mid 0,68;1 \mid 1,96;0 \mid 3,69\}; \\ \mu_{R_2} &= \{0 \mid 0,37;1 \mid 1,17;0 \mid 2,75\}; \\ \mu_{R_3} &= \{0 \mid 0,94;1 \mid 2,31;0 \mid 3,5\}\end{aligned} \quad (1.16)$$

Оцінки бажаності альтернатив дорівнюють:

$$\begin{aligned}\mu(a_1) &= 0,90 \\ \mu(a_2) &= 0,62 \\ \mu(a_3) &= 1,0\end{aligned} \quad (1.17)$$

Кращою альтернативою є a_3 б гіршою - a_2 .

Розв'яжемо задачу методом аналізу ієрархій.

На заданому наборі критеріїв була побудована трирівнева ієрархія, на верхньому рівні якої зазначена ціль вибору. На другому рівні знаходяться узагальнені критерії: прибуток і ризик. На третьому рівні ієрархії розміщені зазначені критерії. Експертні судження і отримані пріоритети наведені в матрицях попарних порівнянь (1.18).

$$\begin{array}{c} \begin{array}{c|ccc|c} c_G & c_P & c_R & & w_G \\ \hline c_P & 1 & 1 & & 0,5 \\ c_R & 1 & 1 & & 0,5 \end{array} \quad \begin{array}{c|ccc|c} c_G & c_4 & c_5 & & w_R \\ \hline c_4 & 1 & 1/5 & & 0,167 \\ c_5 & 5 & 1 & & 0,833 \end{array} \\ \\ \begin{array}{c|ccc|c} c_P & c_1 & c_2 & c_3 & w_P \\ \hline c_1 & 1 & 4 & 4 & 0,667 \\ c_2 & 1/4 & 1 & 1 & 0,167 \\ c_3 & 1/4 & 1 & 1 & 0,167 \end{array} \quad \begin{array}{c|ccc|c} c_P & a_1 & a_2 & a_3 & w_1 \\ \hline a_1 & 1 & 3 & 1/7 & 0,149 \\ a_2 & 1/3 & 1 & 1/9 & 0,066 \\ a_3 & 7 & 9 & 1 & 0,785 \end{array} \\ \\ \begin{array}{c|ccc|c} c_2 & a_1 & a_2 & a_3 & w_2 \\ \hline a_1 & 1 & 3 & 6 & 0,667 \\ a_2 & 1/3 & 1 & 2 & 0,222 \\ a_3 & 1/6 & 1/2 & 1 & 0,111 \end{array} \quad \begin{array}{c|ccc|c} c_3 & a_1 & a_2 & a_3 & w_3 \\ \hline a_1 & 1 & 1/2 & 1/8 & 0,087 \\ a_2 & 2 & 1 & 1/5 & 0,162 \\ a_3 & 8 & 5 & 1 & 0,751 \end{array} \\ \\ \begin{array}{c|ccc|c} c_4 & a_1 & a_2 & a_3 & w_4 \\ \hline a_1 & 1 & 3 & 1/7 & 0,149 \\ a_2 & 1/3 & 1 & 1/9 & 0,066 \\ a_3 & 7 & 9 & 1 & 0,785 \end{array} \quad \begin{array}{c|ccc|c} c_5 & a_1 & a_2 & a_3 & w_5 \\ \hline a_1 & 1 & 2 & 7 & 0,615 \\ a_2 & 1/2 & 1 & 3 & 0,292 \\ a_3 & 1/7 & 1/3 & 1 & 0,093 \end{array} \end{array} \quad (1.19)$$

У результаті підрахунків отримані вектори пріоритетів альтернатив:

$$\begin{aligned}w_R &= \{0,538;0,254;0,208\} \\ w_P &= \{0,225;0,108;0,667\} \\ w_G &= \{0,381;0,181;0,438\}\end{aligned} \quad (1.19)$$

Альтернативою з найменшим ризиком є a_1 , а найбільший прибуток принесе a_3 . Та ж альтернатива має максимальний пріоритет відносно цілі вибору.

1.2.7 Аналіз результатів

На рисунку 1.3 наведені результати розв'язання задачі вибору раціонального інвестиційного проекту, отримані різними методами, після приведення до однотипного представлення, тобто сума усіх пріоритетів складає 1.

Незважаючи на те, що вхідна інформація в усіх розглянутих прикладах є послідовною і непротивічною, отримані результати помітно різняться. За думкою експертів, метод аналізу ієрархій дав результати, які гарно узгоджені з їх інтуїтивним представленням при раціональному підході до прийняття рішень [13].

Неспівпадання результатів, отриманих різними методами, пояснюється, з одного боку, різними способами представлення експертної інформації, а з іншої – різницею підходів до прийняття рішень. Так, в основу метода аналізу ієрархій та відношення уподобань закладений раціонально-зважений підхід, заснований на попарних порівняннях об'єктів і нормованих вагових коефіцієнтах.

Максимінна згортка і лінгвістична векторна оцінка є реалізаціями песимістичного підходу, який ігнорує добрі сторони альтернатив, коли кращою вважається альтернатива,

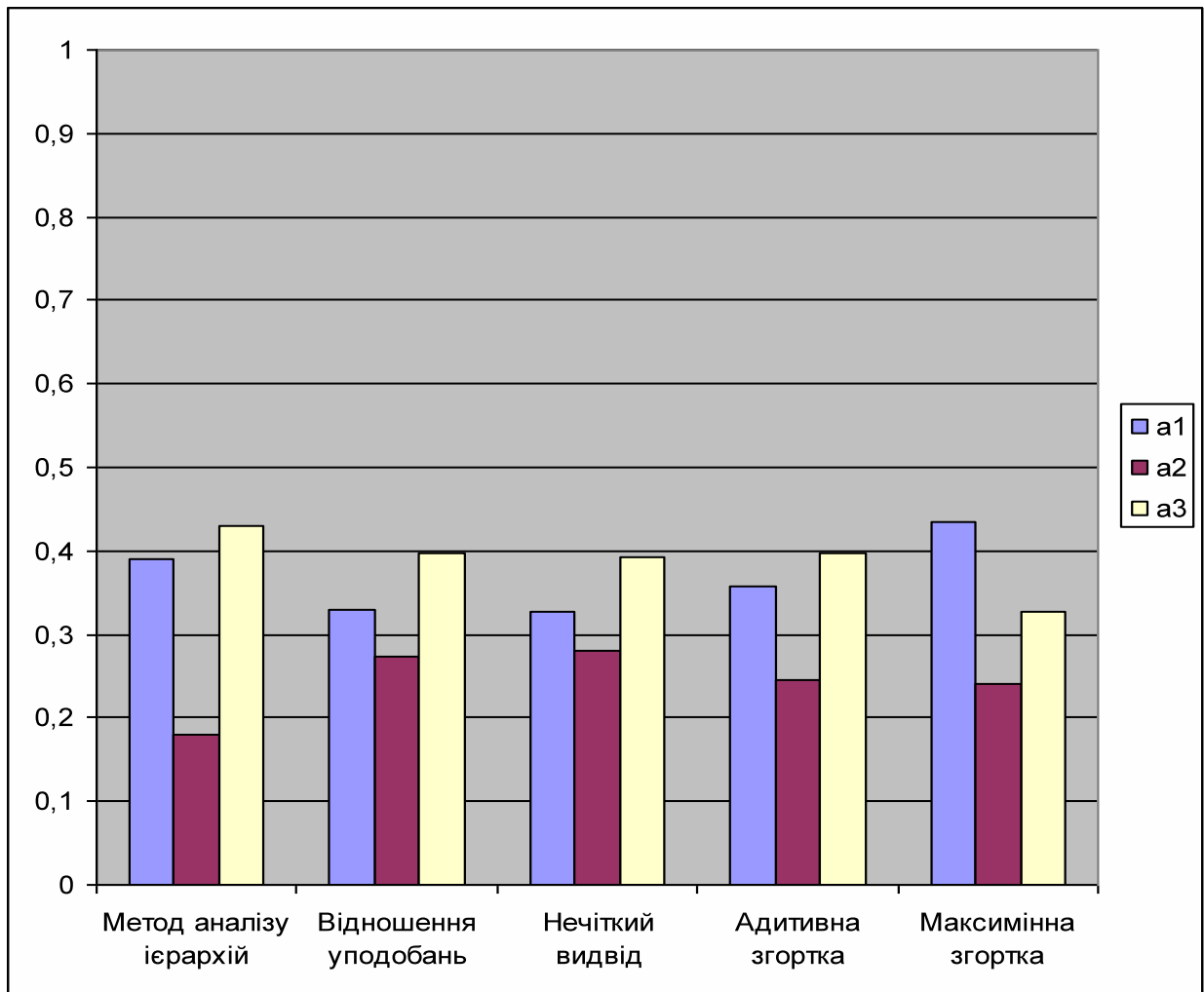


Рисунок 1.3 - Результати розв'язання задачі вибору раціонального інвестиційного проекту, отримані різними методами

яка має мінімальні недоліки за всіма критеріями. Адитивна згортка передбачає оптимістичний підхід, коли низькі оцінки по критеріях мають однаковий статус порівняно з високими. Нечіткий вивід на правилах реалізує евристичний підхід. Проаналізувавши наведені результати, можна зробити такі висновки:

- методи прийняття рішень на нечітких моделях дозволяють зручно і достатньо об'єктивно робити оцінку альтернатив за окремими критеріями. На відміну від інших методів, додавання нових альтернатив не змінює порядок раніше ранжованих наборів. При оцінці альтернатив за критеріями можлива як лінгвістична оцінка, так і оцінка на основі точкових оцінок з використанням функцій приналежності критеріїв.

- основною проблемою багатокритеріального вибору з застосуванням нечітких моделей є представлення інформації про взаємовідносини між критеріями та

способи обчислення інтегральних оцінок. Методи, які базуються на різних підходах, дають різні результати. Найбільш широкі можливості для представлення інформації дає евристичний підхід та метод аналізу ієрархій.

- більшість нечітких методів прийняття рішень показує слабку стійкість результатів відносно вхідних даних. Дослідження розглянутих методів показало, що найбільшою стійкістю характеризується метод, заснований на правилах.

Аналіз нечітких методів прийняття рішень дозволяє сформулювати вимоги до подальших розробок у цій області. Це розвиток теоретичних підходів до опису складних зв'язків між критеріями і більш широке застосування інтелектуальних методів на основі нечіткої логіки.

1.4 Постановка задачі

1.3.1 Формулювання цілей та задач проекту

В теперішній час в умовах швидкої зміни соціально-економічної, технологічної і політичної ситуацій виникли більш складні й більш відповідальні задачі прогнозування, в яких відсутня інформація про ймовірності, а прогнозування здійснюється на основі суб'єктивних оцінок особи, яка приймає рішення. Такі задачі відносяться до задач прийняття рішень в умовах невизначеності й виникають при стратегічному прогнозуванні, плануванні, розподілі ресурсів [6]. Коли економічні фактори зводяться до чисел у грошовому вимірі, кількість об'єктів, їх вага в тонах та час, необхідний для їх виробництва, розраховані й зроблені оцінки ймовірностей, часто виявляється, що ефективність процесу моделювання складних проблем досягла своєї межі. Можливості людини в цьому процесі сильно залежать від тих факторів, які ми можемо вимірювати. Існує ймовірність упущання деяких значних факторів при прийнятті полегшуючих допущень. Але саме вони можуть виявитися тими контролюючими факторами, з якими ми повинні мати справу і які повинні вміти вимірювати, щоб отримати реальні результати.

Необхідно по можливості відмовитися від прийняття полегшуючих допущень в моделях і приймати складні ситуації такими, якими вони є. Щоб бути

реалістичними, моделі повинні включати в себе й дозволяти вимірювати всі відчутні й невідчутні, кількісні й якісні фактори [7]. Для цього пропонується застосувати методи якісного моделювання.

Мета роботи – розробка програмного засобу підтримки прийняття рішень для багатокритеріальних задач з використанням методу аналізу ієрархій.

1.3.2 Вимоги до програмного забезпечення

Програмний продукт призначений для підтримки прийняття рішень у багатокритеріальних задачах з ієрархічними структурами цілей, які включають як осяжні, так і неосяжні фактори. Він повинен надавати змогу встановлювати попарні якісні характеристики для оцінюваних факторів та розраховувати міру пріоритетів альтернатив вибору.

Програма повинна мати такі характеристики:

- надавати можливість формування ієрархічної структури критеріїв необмеженої глибини і задавати потрібні альтернативи;
- шляхом опитування користувача давати йому змогу визначити якісні оцінки пріоритетів;
- розраховувати пріоритети альтернатив;
- повідомляти про узгодженість розрахунків та надавати можливість зміни суджень;
- повинна мати дружній інтерфейс для редагування та виводу зазначених даних;
- не повинна залежати від апаратної платформи.

Детальні вимоги до програмного продукту викладені в «Технічному завданні» (Додаток А).

Висновок до розділу 1

Необхідно по можливості відмовитися від прийняття полегшуючих допущень в моделях і приймати складні ситуації такими, якими вони є. Щоб бути реалістичними, моделі повинні включати в себе й дозволяти вимірювати всі відчутні й невідчутні, кількісні й якісні фактори. Для цього пропонується застосувати методи якісного моделювання.

2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Загальносистемні рішення

1.1.1 Розробка варіантів використання

Визначимо варіанти використання системи. Згідно з проведеними дослідженнями та постановкою мети розробки, програма повинна надавати користувачу можливість зберігати та редагувати ієрархічні системи задач та після опитування повідомляти рішення.

Виходячи з цього, були встановлені такі варіанти використання (рис. 2.1):

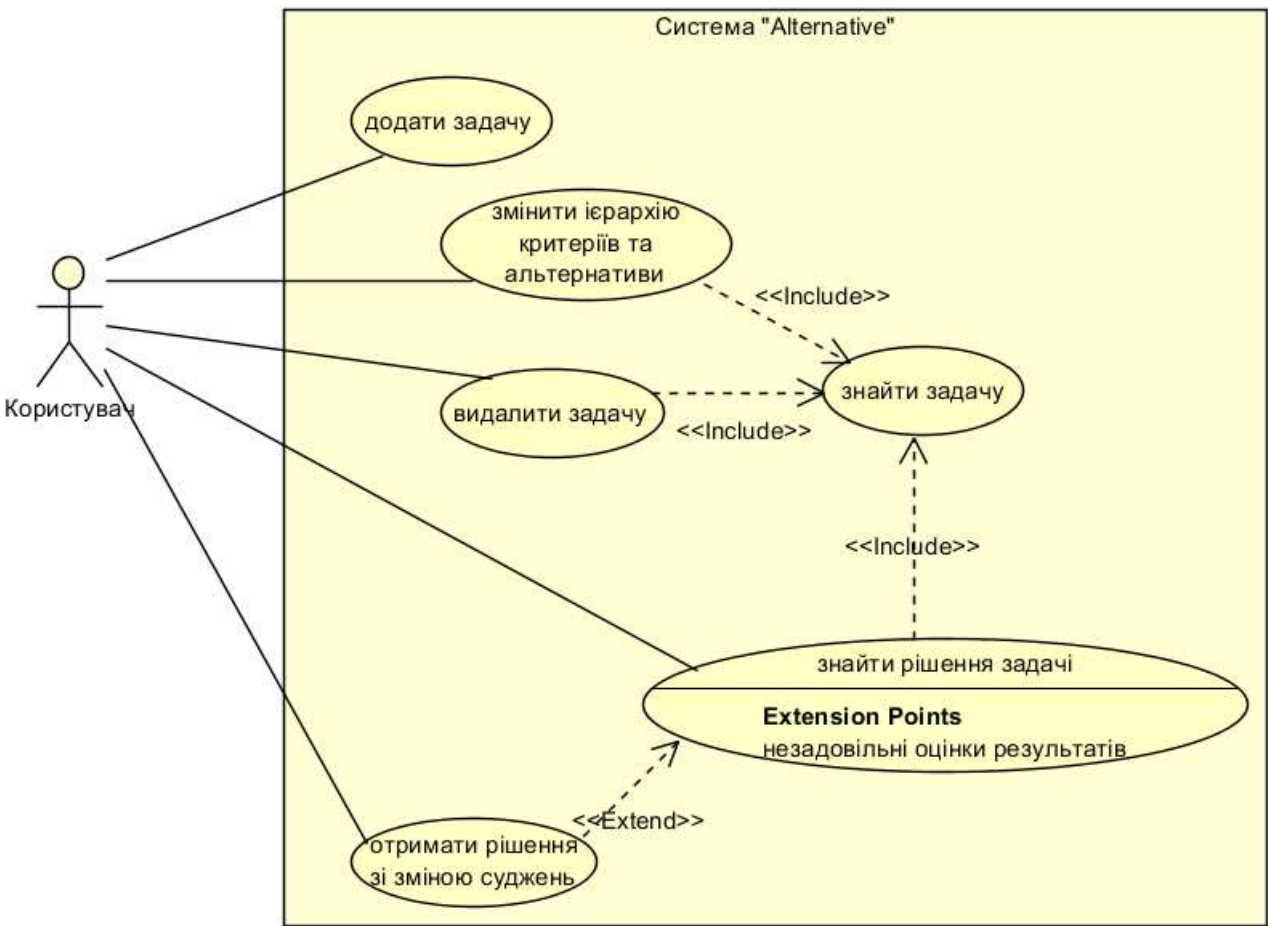


Рисунок 2.1 – Діаграма варіантів використання

Опис кожного з варіантів та їх альтернативних потоків поданий у таблиці 2.1.

Таблиця 2.1 – Опис варіантів використання

Варіант використання	Параметр	Опис
Додати задачу	Короткий опис	занесення в систему даних про нову задачу
	Основна дійова особа	користувач
	Передумови	система знаходиться в режимі очікування (не виконуються інші задачі)
	Дії, що запускають варіант використання	користувач обирає дію «додати задачу» на головній екранній формі програми
	Сценарій	1 Система відкриває редактор задачі. 2 Користувач за допомогою наданого інтерфейсу вводить цілі, ієрархії критеріїв та списку альтернатив.
	Умови завершення варіанту використання	Користувач вибрав дію «Зберегти»
	Результат	система отримала дані про нову задачу
	Альтернативні потоки	Відмінити Виправити дані

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
----------------------	----------	------

Знайти задачу	Короткий опис	знаходження серед існуючих задач, яка цікавить користувача
	Основна дійова особа	користувач
	Передумови	система знаходиться в режимі очікування (не виконуються інші задачі)
	Дії, що запускають варіант використання	користувач звертається до головної екранної форми програми
	Сценарій	1 Система переглядає сховище даних та видає назви тих задач, що в ньому зберігаються. 2 Користувач переглядає список назв, може переглянути збережені ієрархічні системи. 3 Користувач за допомогою екранного інтерфейсу робить вибір потрібної задачі.
	Умови завершення варіанту використання	Користувач зупинив свій вибір на одній із задач
	Результат	система отримала інформацію, яка саме з існуючих задач цікавить користувача
	Альтернативні потоки	Додати задачу

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
Змінити ієрархію критеріїв та альтернативи	Короткий опис	зміна даних існуючої задачі
	Основна дійова особа	користувач
	Передумови	система знаходиться в режимі очікування (не виконуються інші задачі)
	Дії, що запускають варіант використання	користувач обирає задачу та дію «Редагувати» за допомогою інтерфейсу
	Сценарій	1 Система шукає інформацію про обрану задачу у сховищі та відкриває редактор задачі із знайденими даними. 2 Користувач за допомогою наданого інтерфейсу вносить зміни до ієрархії критеріїв та списку альтернатив.
	Умови завершення варіанту використання	Користувач обрав дію «Зберегти» або «Відміна»
	Результат	система отримала змінену інформацію про задачу
	Альтернативні потоки	Відмінити Виправити дані

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
Видалити задачу	Короткий опис	видалення з даних існуючої задачі
	Основна дійова особа	користувач
	Передумови	система знаходиться в режимі очікування (не виконуються інші задачі)
	Дії, що запускають варіант використання	користувач обирає задачу та дію «Видалити» за допомогою інтерфейсу
	Сценарій	1 Система робить запит до користувача про підтвердження операції видалення. 2 Користувач підтверджує операцію 3 Система шукає інформацію про обрану задачу у сховищі та видаляє її. 4 Система поновлює інформацію про наявні задачі для користувача.
	Умови завершення варіанту використання	Якщо користувач підтвердив операцію, то дія безумовна. Якщо користувач не підтвердив операцію, то варіант використання завершується.
	Результат	система отримала змінену інформацію про задачу
	Альтернативні потоки	Відмінити

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант викорис тання	Параметр	Опис
Знайти рішення задачі	Короткий опис	попарне порівняння визначених системою відношень та отримання кінцевого результату пріоритетів та оцінок узгодженості суджень
	Основна дійова особа	користувач
	Передумови	система знаходиться в режимі очікування (не виконуються інші задачі)
	Дії, що запускають варіант використання	користувач обирає задачу та дію «Прийняти рішення» за допомогою інтерфейсу
	Сценарій	1 Система шукає інформацію про обрану задачу, формує набори пар критеріїв та альтернатив та відкриває редактор пріоритетів. 2 Користувач за допомогою інтерфейсу встановлює пріоритети, що відповідають його судженням, може повертатися до щойно встановлених та змінювати їх. 3 Користувач обирає дію «Проверсти розрахунки» за допомогою інтерфейсу. 4 Система обробляє введені дані та виводить детальний результат по всіх порівняннях та загальний результат в цілому. Також надається інформація про узгодженість суджень.
	Умови завершення варіанту використання	користувач обрав дію виходу з редактору пріоритетів
	Результат	користувач отримав інформацію по глобальних пріоритетах альтернатив, що його цікавлять, а також додаткову інформацію по кожному вузлу ієрархії.
	Альтернативні потоки	Відмінити

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
Отримати рішення зі зміною суджень	Короткий опис	зміна суджень попарних порівнянь визначених системою відношень та отримання нового кінцевого результату пріоритетів та нових оцінок узгодженості суджень
	Основна дійова особа	користувач
	Передумови	користувач щойно ввів пріоритети та розрахував результат, після чого виявилось, що оцінки результату незадовільні, користувач виявив бажання змінити судження
	Дії, що запускають варіант використання	користувач за допомогою інтерфейсу переходить від перегляду результатів до редагування пріоритетів
	Сценарій	1 Система відображає щойно вказані судження користувача та надає змогу змінювати будь-які з них в редакторі пріоритетів. 2 Користувач переглядає свої судження та встановлює нові. 3 Користувач обирає дію «Проверсти розрахунки» за допомогою інтерфейсу. 4 Система обробляє введені дані та виводить детальний результат по всіх порівняннях та загальний результат в цілому. Також надається інформація про узгодженість суджень.
	Умови завершення варіанту використання	користувач обрав дію виходу з редактору пріоритетів
	Результат	користувач отримав інформацію по глобальних пріоритетах альтернатив, що його цікавлять, а також додаткову інформацію по кожному вузлу ієрархії.
	Альтернативні потоки	Відмінити

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
Альтернативний потік «Відмінити»	Короткий опис	відміна внесених змін та зупинення поточної дії
	Основна дійова особа	користувач
	Передумови	розпочатий будь-який із вищевказаних варіантів використання
	Дії, що запускають варіант використання	користувач за допомогою інтерфейсу обирає дію «Відмінити» або закриває редактор
	Сценарій	Система не зберігає внесені зміни
	Умови завершення	Безумовно
	Результат	користувач переходить до попереднього інтерфейсу
	Післяумова	дані не змінились

Таблиця 2.1. Продовження – Опис варіантів використання

Варіант використання	Параметр	Опис
----------------------	----------	------

Альтернативний потік «Виправити дані»	Короткий опис	виправлення даних, що вказані при редагуванні, які не відповідають критеріям системи
	Основна дійова особа	користувач
	Передумови	користувач щойно ввів дані
	Дії, що запускають варіант використання	користувач за допомогою інтерфейсу обрав дію «Зберегти», система перевірила вхідні дані та виявила невідповідність критеріям
	Сценарій	1 Система за допомогою повідомлювача відображає для користувача, що саме є причиною неможливості збереження. Якщо була виявлена виключна ситуація, то детальна інформація заноситься до системного логу. 2 Користувач приймає до уваги, що саме є причиною неможливості збереження даних.
	Умови завершення варіанту використання	користувач припиняє дію повідомлювача
	Результат	Користувач отримав інформацію, що треба змінити для успішного збереження даних, і повернувся до попереднього стану редагування, в якому знаходився до обрання дії «Зберегти».
	Післяумова	дані не збережені

2.1.2 Інформаційна модель

Було встановлено, що для роботи розроблюваної системи потрібно зберігати інформацію про задачі. Задача посилається на 1 критерій (верхнього рівня, який є ціллю) та множину альтернатив. Кожний критерій у свою чергу містить

посилання на множину своїх критеріїв, для яких він є ціллю. Так забезпечується ієрархічна система даних системи.

Інформаційна модель даних за допомогою діаграми «сутність-зв'язок» в нотації Чена наведена на рисунку 2.2.

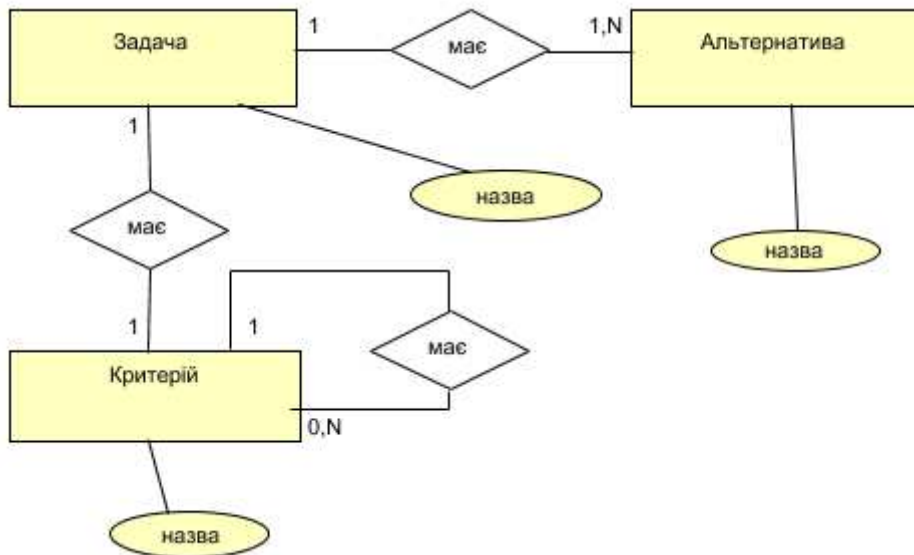


Рисунок 2.2 – Діаграма «сутність-зв'язок» інформаційної моделі даних

2.1.3 Проект інтерфейсу користувача

Для розробки проекту інтерфейсу з'ясуємо стани, в яких може перебувати розроблювана система під впливом дій користувача (табл. 2.2). Діаграму станів системи наведено на рисунку 2.3.

Таблиця 2.2 – Опис станів системи

Назва стану	Опис
Пошук збережених даних	Завантаження (оновлення) даних у списку доступних задач. В цей стан переходить система одразу після завантаження та після дій, які призвели до змін у даних (стани «Збереження даних» та «Видалення даних»). Після цього програма переходить в стан «Режим очікування вибору».

Режим очікування вибору	Система очікує на вибір задачі зі списку або на запит додавання задачі за допомогою інтерфейсу. В цей стан система переходить після завершення пошуку доступних задач після стану «Пошук збережених задач». З цього стану система може перейти в стани «Режим очікування», «Редагування ієрархії», «Завершення роботи програми».
Режим очікування	Система очікує на запити користувача за допомогою інтерфейсу. В цей стан вона переходить зі станів «Режим очікування вибору», «Відображення результату», «Редагування пріоритетів», «Очікування відповіді підтвердження». Наступними станами можуть бути «Редагування пріоритетів», «Очікування відповіді підтвердження», «Завантаження збережених даних задачі», «Редагування ієрархії», «Завершення роботи програми».
Редагування ієрархії	Редагування ієрархії критеріїв та списку альтернатив за допомогою візуальних компонентів. В цей стан переходить система після запиту на створення задачі зі станів «Режим очікування» та «Режим очікування вибору» та після стану «Завантаження збережених даних задачі». З нього вона може перейти в стани «Перевірка даних результатів» та «Режим очікування вибору».
Завантаження збережених даних задачі	Система шукає в сховищі збережені дані запитаної задачі, завантажує їх у пам'ять та передає до редактора. В цей стан система потрапляє зі стану «Режим очікування». З нього вона переходить до стану «Редагування ієрархії».

Таблиця 2.2. Продовження – Опис станів системи

Назва стану	Опис
Перевірка введених даних	Перевірка даних, введених користувачем у процесі редагування, на відповідність обмежуючим критеріям задачі. В цей стан потрапляє система зі стану «Редагування ієрархії». З нього вона може переходити до стану «Збереження даних» та повернутися до стану «Редагування ієрархії», якщо були знайдені невідповідності.

Збереження даних	Збереження даних, які були введені користувачем, до сховища даних. В цей стан переходить система зі стану «Перевірка введених даних» при позитивному результаті перевірки. З нього вона переходить до стану «Пошук збережених задач».
Очікування відповіді підтвердження	Очікування відповіді на запит до користувача про підтвердження бажання видалити задачу. До цього стану приходить система, зі стану «Режим очікування» після запиту на видалення. Якщо користувач підтверджує бажання, то система переходить до стану «Видалення даних», якщо ні, то до стану «Режим очікування».
Видалення даних	Видалення збережених даних для запитаної задачі. В цей стан приходить система зі стану «Очікування відповіді підтвердження» після позитивної відповіді. З нього вона переходить до стану «Пошук збережених даних».
Редагування пріоритетів	Редагування пріоритетів для складених пар порівнянь за допомогою інтерфейсу. До цього стану система приходить зі стану «Режим очікування» після запиту на пошук рішення. З нього вона може перейти до стану «Відображення результату» та повернутися до стану «Режим очікування» у разі відміни.
Відображення результату	Відображення результату розрахунків. В цей стан переходить система після запиту на здійснення розрахунків зі стану «Редагування пріоритетів». З нього вона може перейти до стану «Режим очікування» після виходу з редактору або повернутися до стану «Редагування пріоритетів».

Таблиця 2.2. Продовження – Опис станів системи

Назва стану	Опис
Завершення роботи програми	Кінцевий стан системи, в який вона переходить після запиту на вихід зі станів «Режим очікування» та «Режим очікування вибору».

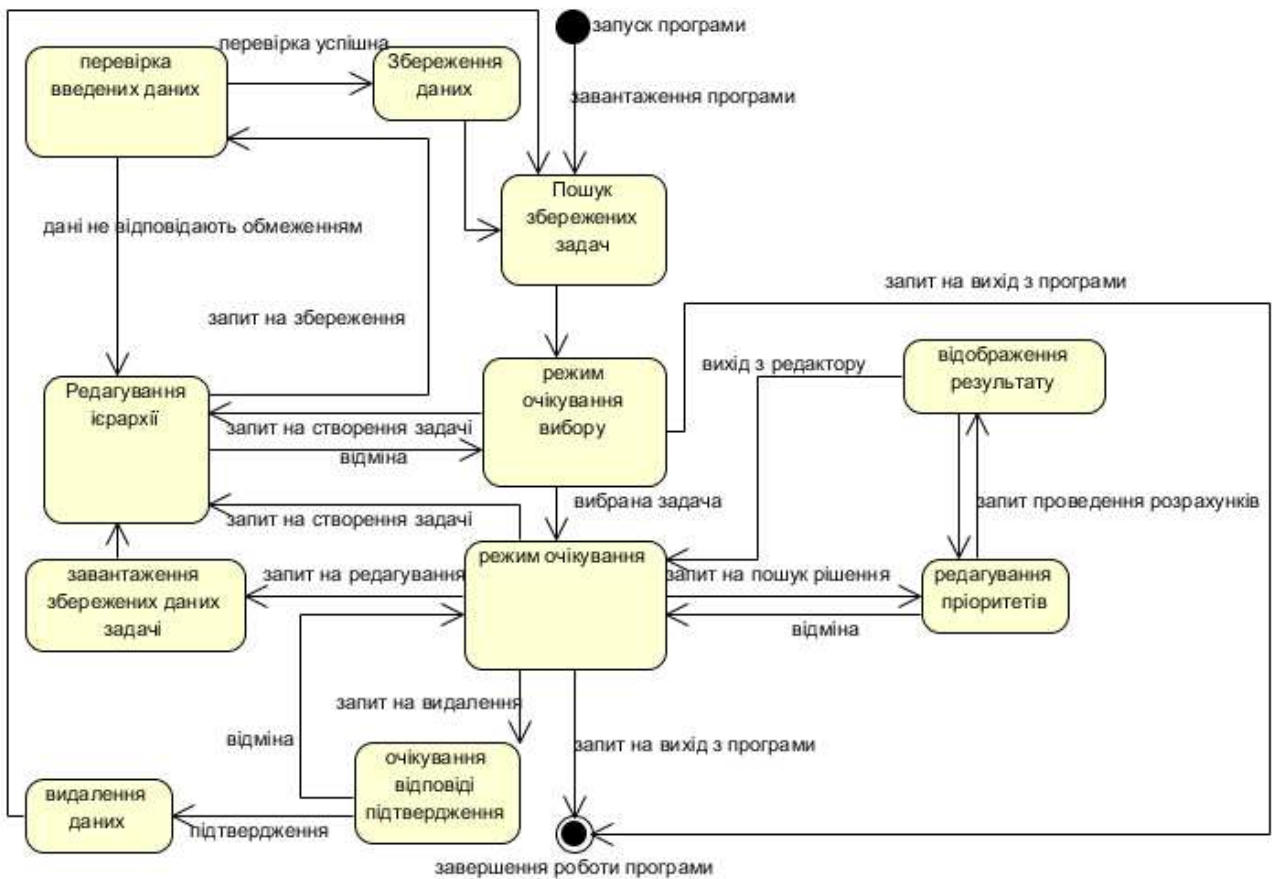


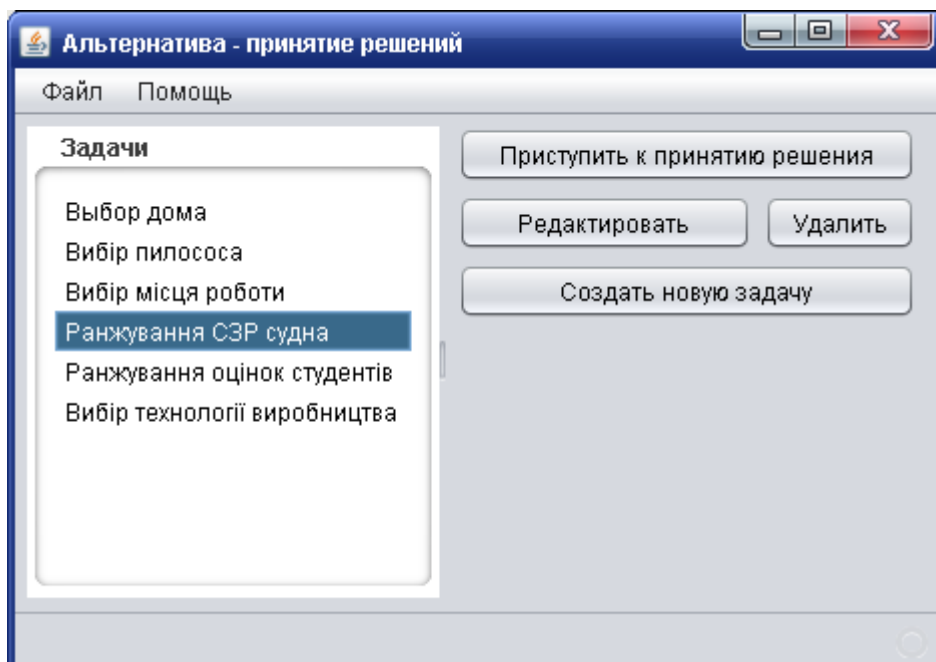
Рисунок 2.3 – Діаграма станів розроблюваної системи

Встановимо необхідні компоненти інтерфейсу користувача. Згідно визначених варіантів використання системи, інформаційної моделі та станів, в яких може система перебувати, в ньому повинні бути такі окремі головні екранні форми:

- форма для відображення списку задач та вибору з нього (рис. 2.4), на якій повинні бути присутні елементи для можливих запитів на створення нової задачі.

Так як 2 стани системи «Режим очікування» та «Режим очікування вибору» мають схожі переходи, то їх реалізацію можна поєднати в одній екранній формі, забезпечуючи за допомогою програмних засобів неможливість заборонених переходів та відповідно реагуючи на дії користувача, коли ці переходи стають можливими. Тому на цю форму можуть бути добавлені елементи для здійснення запитів на редагування, видалення та пошук рішення, доступ до яких буде можливим після вибору задачі;

- форма редагування ієрархії (рис. 2.5), на якій повинні бути присутні компоненти для зручного візуального редагування ієрархії цілей та критеріїв та списку альтернатив, а також компоненти для можливості здійснення запиту на збереження та відміну змін;
- форма редагування пріоритетів (рис. 2.6), яка повинна містити компоненти для розміщення великої кількості наборів попарних порівнянь та відображення результатів. Вона також повинна мати компоненти для зручного візуального редагування якісних оцінок пріоритетів з можливістю швидко повернутися до щойно встановлених значень та елементи для здійснення запитів на розрахунок та вихід з редактора.



Рисунк 2.4 - Форма для відображення списку задач та вибору з нього

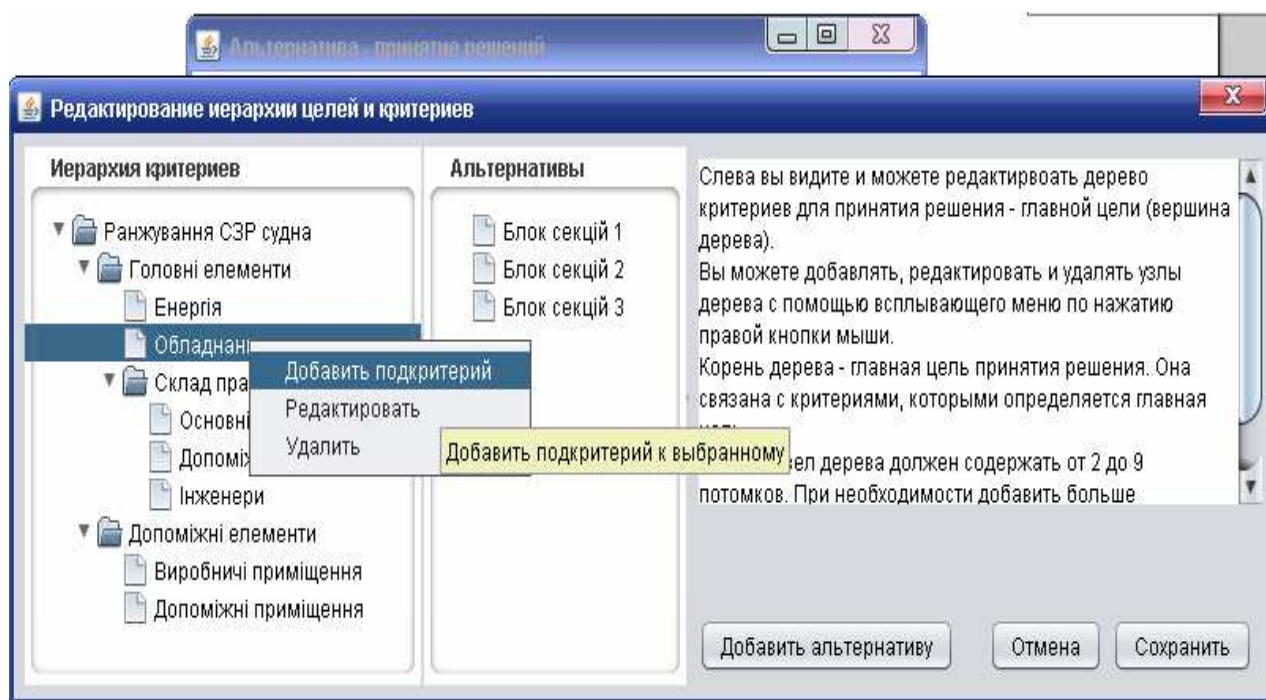


Рисунок 2.5 - Форма редагування ієрархії

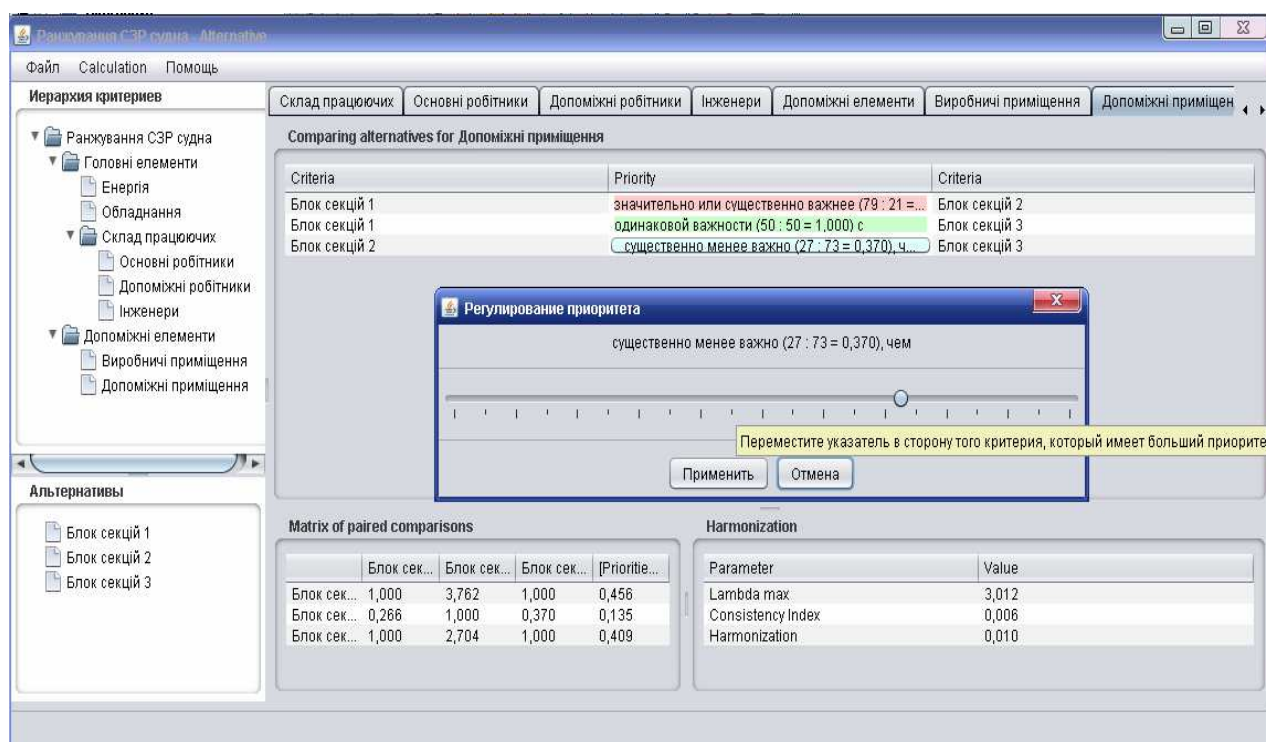


Рисунок 2.6 - Форма редагування пріоритетів

2.2 Рішення з інформаційного забезпечення

2.2.1 Статична модель даних

Модель даних, що зберігаються у файлі структури задачі, є досить простою та не вимагає додаткових пояснень (рис. 2.2).

На відміну від цього, необхідно подати діаграму класів, яка зазначена в Додатку Б. Наведемо специфікацію класів.

Таблиця 2.3 – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
	Problem	Клас, який представляє собою вирішувану задачу
aim	Aim	ціль задачі, яка є вершиною ієрархії
alternatives	Vector<Alternative>	список альтернатив
getAlternatives()	Enumeration <Alternative>	повертає набір альтернатив
listAlternatives()	Vector<Alternative>	повертає упоряднений список альтернатив, порядок в якому гарантовано не змінюється
setAim(Aim aim)	void	встановлення цілі для проблеми (може використовуватися виключно бібліотекою JAXB для конвертації даних в XML)
beforeUnmarshal(Unmarshaller u, Object parent)	void	callback-метод, який викликається перед конвертацією даних із XML
setAlternatives(Collection <Alternative> alternatives	void	метод, декларація якого потрібна для коректного розпізнавання бібліотекою JAXB, але не викликається

Таблиця 2.3. Продовження – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
<code>add(Alternative a)</code>	<code>void</code>	додавання альтернативи до існуючого списку
	<code>Aim</code>	Клас для представлення цілі і критеріїв
<code>canBeDeleted()</code>	<code>boolean</code>	чи може бути видалений
<code>setTitle(String title)</code>	<code>boolean</code>	встановлення назви
<code>getTitle()</code>	<code>String</code>	повертає назву
<code>getCriterias()</code>	<code>Vector<Aim></code>	повертає послідовний список підцілей (критеріїв)
<code>setCriterias(Vector<Aim> criterias)</code>	<code>void</code>	встановлення критеріїв (задекларовано для коректного розпізнавання бібліотекою JAXB)
<code>afterUnmarshal(Unmarshaller u, Object parent)</code>	<code>void</code>	callback-метод, який викликається після конвертації даних об'єкта з XML
<code>toStringWithAncestors ()</code>	<code>String</code>	повертає наглядне представлення критерія з його предками
	<code>Alternative</code>	Клас, який представляє собою альтернативу
<code>parent</code>	<code>MutableTreeNode</code>	посилання на віртуального предка (реалізовано виключно в цілях можливості зручного редагування, яке можливе з використанням компоненту JTree)

title	String	назва альтернативи
-------	--------	--------------------

Таблиця 2.3. Продовження – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
setTitle(String title)	void	встановлення назви
getTitle()	String	отримання назви
canBeDeleted()	boolean	чи може альтернатива бути видалена
toString()	String	повертає строкове відображення
getChildAt(int childIndex)	TreeNode	метод задекларований для можливості використання в якості компоненту JTree
getChildCount()	int	метод задекларований для можливості використання в якості компоненту JTree
setParent(MutableTreeNode newParent)	void	встановлення віртуального вищого вузла (реалізовано виключно для відображення)
getParent()	TreeNode	повертає посилання на віртуального предка (реалізовано виключно для відображення)
getIndex(TreeNode node)	int	задекларовано для відображення, завжди повертає -1
getAllowsChildren()	boolean	задекларовано для відображення, завжди повертає false
isLeaf()	boolean	задекларовано для відображення, завжди повертає true

<code>children()</code>	Enumeration	задекларовано для відображення, список завжди пустий
-------------------------	-------------	---

Таблиця 2.3. Продовження – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
<code>insert(MutableTreeNode child, int index)</code>	<code>void</code>	задекларовано для відображення
<code>remove(MutableTreeNode node)</code>	<code>void</code>	задекларовано для відображення
<code>removeFromParent()</code>	<code>void</code>	убирає посилання на віртуального предка (реалізовано для відображення)
<code>setUserObject(Object object)</code>	<code>void</code>	задекларовано для відображення
	<code>AimTreeModel</code>	Модель для передачі даних ієрархії компоненту JTree, який її відображає
<code>isCorrect()</code>	<code>boolean</code>	чи є коректними відредаговані дані
<code>getCorrectMessages()</code>	<code>List<String></code>	перевірка введених даних та повернення можливих помилок
<code>isCorrect(boolean answerOnly)</code>	<code>boolean</code>	чи є модель коректною
<code>checkAim(Aim aim, boolean answerOnly)</code>	<code>boolean</code>	перевірка на коректність вузла
<code>cleanMessages()</code>	<code>void</code>	очищення списку помилок
	<code>Priority</code>	клас для представлення пріоритету
<code>resourceBundle</code>	<code>ResourceBundle</code>	ресурс, з якого беруться строки для відображення якісних повідомлень

statusMessages	Vector<String>	строки для відображення якісних повідомлень
setValue(double value)	void	встановлення значення пріоритету
getValue()	double	повертає значення пріоритету

Таблиця 2.3. Продовження – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
getLeft()	double	повертає значення для лівої частини шкали
getRight()	double	повертає значення для правої частини шкали
getMessage()	String	повертає строкову для якісної характеристики
toString()	String	повертає строкове відображення пріоритету
equals (Priority other)	boolean	порівняння з іншим пріоритетом
Priority getCopy()	Priority	повертає копію пріоритету
	ComparisonResult	клас для представлення результату одного попарного порівняння
criteria	Vector<? extends Criteria>	список порівнюваних критеріїв або альтернатив
comparisons	Matrix	матриця попарних порівнянь
priorityVector	Vector<Double>	вектор пріоритетів
lambdaMax	double	$\lambda_{\max}$
consistencyIndex	double	індекс узгодженості (ІУ)
harmonization	double	відношення узгодженості (ВУ)

<code>getComparisons()</code>	<code>Matrix</code>	повернути матрицю попарних порівнянь
<code>getPriorityVector</code>	<code>Vector<Double></code>	повернути вектор пріоритетів
<code>getLambdaMax()</code>	<code>double</code>	повернути $\lambda_{\max}$
<code>getConsistencyIndex()</code>	<code>double</code>	повернути індекс узгодженості (ІУ)

Таблиця 2.3. Продовження – Специфікація класів

Атрибут/Метод	Тип параметру / значення, що повертає метод	Пояснення
<code>getHarmonization()</code>	<code>double</code>	повернути відношення узгодженості (ВУ)
	<code>GlobalResult</code>	клас для представлення загального результату
<code>leafPriorities</code>	<code>Vector<Double></code>	список пріоритетів найнижчих в ієрархії критеріїв
<code>finalPriorities</code>	<code>Matrix</code>	остаточні пріоритети альтернатив по кожному найнижчому в ієрархії критерію
<code>globalPriorities</code>	<code>Vector<Double></code>	глобальні пріоритети, які є шуканим результатом
<code>leafs</code>	<code>Vector<Aim></code>	список найнижчих в ієрархії критеріїв
<code>alternatives</code>	<code>Vector<Alternative></code>	список альтернатив
<code>getLeafPriorities()</code>	<code>Vector<Double></code>	повертає список пріоритетів
<code>getFinalPriorities()</code>	<code>Matrix</code>	повертає список остаточних пріоритетів альтернатив
<code>getGlobalPriorities()</code>	<code>Vector<Double></code>	повертає глобальні пріоритети
<code>getLeafs()</code>	<code>Vector<Aim></code>	повертає список найнижчих в ієрархії критеріїв

getAlternatives()	Vector<Alternative>	повертає список альтернатив
-------------------	---------------------	-----------------------------

2.2.2 Динамічна модель системи

Після встановлення структури системи і способів її взаємодії з зовнішнім середовищем встановимо, як саме поводить себе система, шляхом зображення взаємодії об'єктів всередині неї.

Динамічна модель системи за допомогою діаграми взаємодії наведена у Додатку В.

2.3 Рішення з програмного забезпечення

2.3.1 Вибір мови програмування

Мовою програмування була обрана Java. Однією з головних переваг цієї мови є те, що програми, написані на ній, можуть бути запущені на будь-якій платформі, на якій встановлена віртуальна Java-машина. У зв'язку з поширеним в останній час контролем за наявністю тільки ліцензійного програмного забезпечення на підприємствах і встановлення інших систем, окрім Windows, це дуже актуально. Також це об'єктно-орієнтована мова, на якій розроблювати програми, які не містять помилок, значно легше, ніж на деяких інших.

Крім того, для Java існують зручні середовища розробки, які сприяють кращій розробці, такі, як NetBeans та Idea.

Для зберігання інформації про структура вирішуваних користувачем задач обрано файли, в яких інформація буде представлена у форматі XML. Це було зумовлено невеликим обсягом інформації, яка не потребує використання бази даних, та бажанням не створювати додаткові обмеження для середовища розміщення системи, яка в такому разі може використовуватися будь-яким звичайним користувачем. Файли зберігаються у місці, визначеному системою для зберігання даних, належачих кожній програмі, специфічних для кожного

користувача (наприклад, у Windows XP це %USERDIR%\Application Data\Ann Popova\Alternative\).

XML-документ представляє собою звичайний текстовий файл, в якому за допомогою спеціальних маркерів задаються елементи даних, послідовність та вкладеність яких задає структуру документу і його зміст. Основною перевагою XML є те, що при відносно простому способі створення й обробки він дозволяє створювати структуровану інформацію, зрозумілу для програмних засобів.

Для конвертації даних в XML було використано бібліотеку JAXB.

2.3.2 Компоненти системи

Для того, щоб представити архітектурний вид розроблюваної системи для кращої формалізації маршрутів та прийняття і здійснення рішень щодо розподілу задач у ній, а також для отримання зв'язків з логічними компонентами програмного забезпечення, визначимо компоненти системи та зв'язки між ними. Компоненти представляють собою автономні інкапсульовані підрозділи в рамках системи чи підсистеми, яка надає деякий набір інтерфейсів. Зв'язки між компонентами слугують для представлення факту наявності залежності, коли зміна одного елементу моделі здійснює вплив чи призводить до зміни іншого елементу моделі. Діаграма основних компонентів системи наведена у Додатку Г.

2.3.3 Розміщення

Розроблювана система має розміщуватися локально на комп'ютері користувача і не використовує периферійних пристроїв та ресурсів, тому її розміщення досить просте і не потребує додаткових діаграм.

Система може бути запущена на будь-якій платформі, на якій встановлена віртуальна Java-машина.

2.3.4 Реалізація та тестування

Весь програмний код є досить об'ємним (біля 2000 строк), тому в Додатку Д наводяться основні класи.

Для контролю помилок найбільш потенційно важливі з цього боку компоненти було протестовано. Для цього використана модульна технологія тестування, або юніт-тестування, яке для Java надається бібліотекою JUnit.

Вона дозволяє перевірити на коректність окремі модулі програми. Пишуться тести для кожної нетривіального метода або функції, що дозволяє швидко в автоматичному режимі перевірити, чи не призвела чергова зміна коду до регресії, тобто до виникнення помилок у вже написаних і відтестованих місцях, а також забезпечує зручне виявлення та ліквідацію таких помилок.

Така технологія дозволяє проводити тестування як за стратегією «чорного ящика», так і «білого ящика». Також тестування за допомогою JUnit дозволяє використовувати метрики покриття коду (рис. 2.7, рис. 2.8).

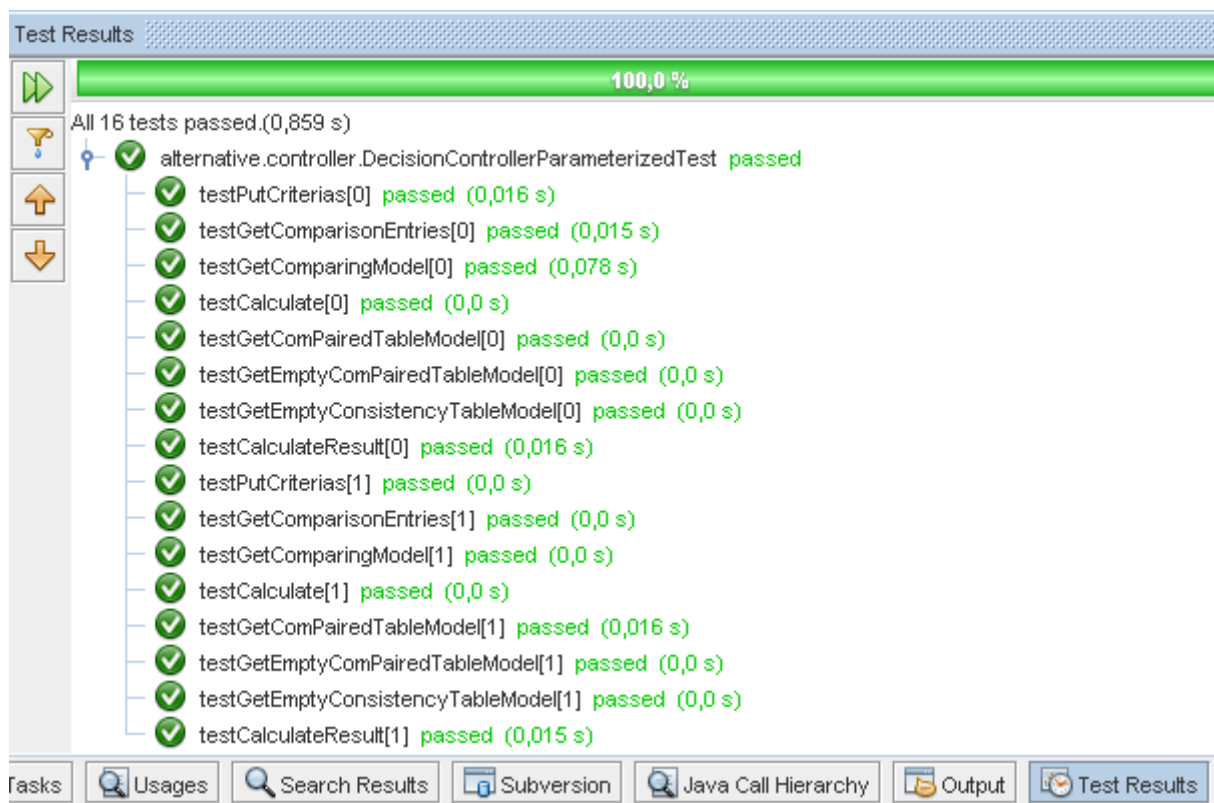


Рисунок. 2.7 – Результати виконання тестів JUnit

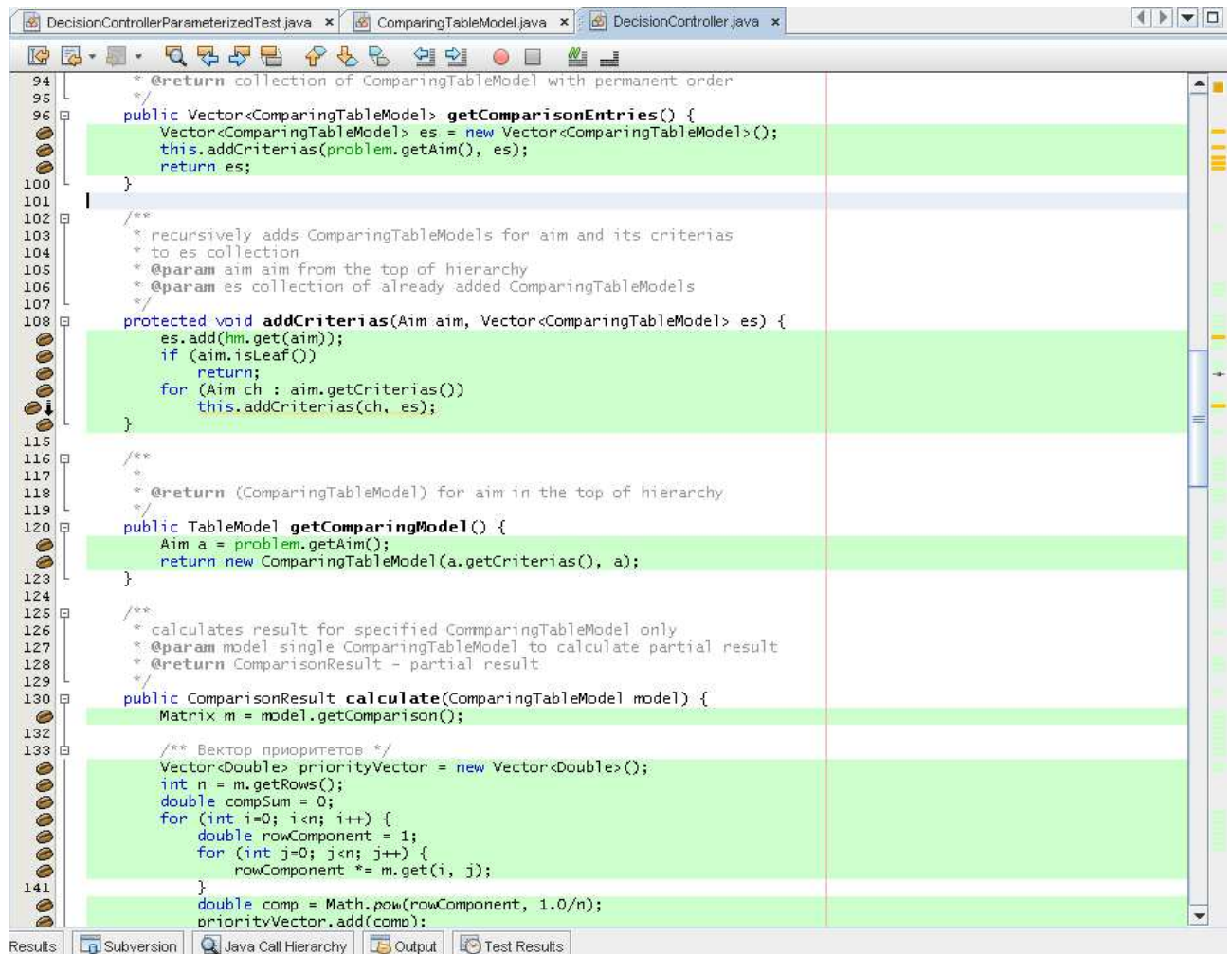


Рисунок 2.8 – Візуальне зображення покритого коду в результаті тестування

Текст юніт-тестів для особливо важливої частини програми наданий у додатку Е.

Висновок до розділу 2

Розроблювана система має розміщуватися локально на комп'ютері користувача і не використовує периферійних пристроїв та ресурсів, тому її розміщення досить просте і не потребує додаткових діаграм.

Система може бути запущена на будь-якій платформі, на якій встановлена віртуальна Java-машина.

3 РЕЗУЛЬТАТИ РОБОТИ

У даній роботі були проведені дослідження методів підтримки прийняття рішення у багатокритеріальних задачах з ієрархічними структурами цілей, які включають як осяжні, так і неосяжні фактори та розроблен проект інформаційної системи, що являє собою надавати змогу встановлювати попарні якісні характеристики для оцінюваних факторів та розраховувати міру пріоритетів альтернатив вибору.. Для цього були використані засоби автоматизованої розробки програмного забезпечення.

В процесі виконання роботи були пройдені наступні етапи:

- дослідження предметної області;
- формування вимог до програмного продукту;
- вибір методу розв'язання головної задачі;
- розробка математичної моделі системи;
- розробка технічного завдання на створення системи;
- розробка проектних рішень;
- рішення з інформаційного забезпечення;
- рішення з програмного забезпечення;
- розробка опису програми;
- розробка опису використання;
- розробка інструкції користувача;

Програмний продукт був розроблена на основі платформи Java SE. Він відповідає вимогам, зазначеним у Технічному завданні, що приводиться у додатку А.

У системі були повністю реалізовані наступні функції:

- Для цього необхідні такі функціональні можливості:
- надавати можливість формування ієрархічної структури критеріїв необмеженої глибини і задавати потрібні альтернативи. Критерії представляються у вигляді дерева, альтернативи – у вигляді списку. Це потребує створення форми редагування з двома відділами: візуального

редагування дерева, представленого вбудованим компонентом, та візуального редагування списку, який теж має бути представлений вбудованим компонентом;

- за допомогою вказаного редактора користувач може додавати задачі, редагувати їх та видаляти;
- надавати можливість перегляду існуючих задач, виводячи усі збережені у вигляді списку, та вибору з цього списку потрібної;
- шляхом опитування користувача після вибору задачі давати йому змогу визначити якісні оцінки пріоритетів.
- за створеною ієрархією згідно використовуваному методу Сааті визначати, через які саме порівняння потрібно провести користувача у опитуванні, та у зручній компактній формі виводити сформовані комплекти пар;
- введення пріоритету повинно відбуватися у візуальному режимі з відображенням і динамічною зміною якісної характеристики;
- надавати можливість змінити попередні судження, як під час введення пріоритетів, так і після отримання результату;
- розраховувати пріоритети альтернатив за методом аналізу ієрархій;
- повідомляти про узгодженість розрахунків та надавати можливість зміни суджень.

Вимоги до документації встановлюють необхідність таких документів:

- тексти програмних модулів;
- тексти JUnit-тестів;
- опис програмного засобу;
- інструкція користувача;
- методика випробувань.

У ході виконання роботи відповідно з вимогами технічного завдання була розроблена програмна документація до системи

Висновок до розділу 3

Програмний продукт був розроблена на основі платформи Java SE. Він відповідає вимогам, зазначеним у Технічному завданні, що приводиться у додатку А.

4 РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ ПРОГРАМНОГО ПРОДУКТУ

4.1 Загальна характеристика

У теперішній час в умовах швидкої зміни соціально-економічної, технологічної і політичної ситуацій виникли більш складні й більш відповідальні задачі прогнозування, в яких відсутня інформація про ймовірності, а прогнозування здійснюється на основі суб'єктивних оцінок особи, яка приймає рішення. Такі задачі відносяться до задач прийняття рішень в умовах невизначеності й виникають при стратегічному прогнозуванні, плануванні, розподілі ресурсів [6]. З розвитком інформаційних технологій і такі складні задачі можуть бути розв'язані за допомогою програм підтримки прийняття рішень, однією з яких є розроблюваний програмний засіб «Alternative», який дозволяє після встановлення цілі рішення, дерева критеріїв та набору альтернатив та проходження якісного опитування за попарними порівняннями визначити кінцеві пріоритети альтернатив.

Виходячи з цього, необхідно зробити розрахунок вартості створення даного програмного засобу, а також зробити оцінку його економічної ефективності.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування ціни програми.

Створення автоматизованої системи вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування системи. Економія від функціонування автоматизованої системи визначається з урахуванням витрат на її експлуатацію. Відношення цієї економії до витрат на створення автоматизованої системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за оптовими цінами, що діють на момент розрахунку, тарифами і ставками заробітної плати.

4.2. Розрахунок вартості створення програмного засобу.

Витрати на розробку системи складаються з витрат на:

- заробітну платню розробника;

- амортизацію ЕОМ, на якій виконується розробка;
- експлуатацію цієї ЕОМ;
- засоби розробки;
- матеріали і комплектуючі.

Виходячи з того, що основна заробітна платня розробників програмного забезпечення складає 2500 грн./міс., вартість сучасної ПЕОМ складає 6000 грн. і вартість кіловат-години електроенергії складає 0,2436 грн., розрахуємо вартість розробки системи.

Розрахунок заробітної плати:

$$З_{zn} = ЗП_{роз} \cdot T_{роз} \quad (4.1)$$

де $ЗП_{роз}$ – зарплатня розробника, за місяць; $T_{роз}$ – тривалість розробки (дослідження, створення, налагодження і впровадження).

Для розробки даної системи необхідно 4 місяці (за експертною оцінкою часу на розробку аналогічних систем). Кількість розробників – 1 людина.

З цього випливає, що загальна сума витрат на заробітну платню складе:

$$З_{zn} = ЗП_{роз} \cdot T_{роз} = 2500 \cdot 4 = 10000 \text{ (грн.)}$$

Витрати на амортизацію ЕОМ, на якій виконується розробка, розраховується за формулою:

$$З_{амор} = Ц_{варт} \cdot A \cdot T_{роз} \quad (4.2)$$

де $Ц_{варт} = 6000$ грн. – балансова вартість ЕОМ; $A = 60\%$ - амортизація за рік; термін служби ЕОМ – 5 років; $T_{роз} = 0,33$ року – час, необхідний для розробки системи.

$$З_{амор} = 6000 \cdot 0,6 \cdot 0,33 = 1188 \text{ (грн.)}$$

Витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії і розраховуються по формулі:

$$З_{Амор} = P_{ЕОМ} \cdot T_{розр} \cdot N_{р\delta} \cdot N_{рч} \cdot E_{ел} \quad (4.3)$$

де $P_{EOM} = 0,4$ квт./год. – потужність ЕОМ; $T_{розр} = 4$ місяці – тривалість розробки;
 $N_{рд} = 22$ дні – число робочих днів у місяці; $N_{рч} = 8$ годин – число годин у
робочому дні; $E_{ел} = 0,2436$ грн. – вартість 1 кВт/год електроенергії.

$$З_{Амор} = 0,4 \cdot 4 \cdot 22 \cdot 8 \cdot 0,2436 = 68,59 \text{ (грн.)}$$

Витрати на матеріали і комплектуючі вироби враховують витрати на папір для друкувальних пристроїв, на картридж і тонер для принтера, а також на непередбачені витрати. Розрахунок приведений у таблиці 1 (ціни взяті відповідно до прайс-листів та договірних цін на даний вид матеріалів).

Таблиця 4.1 – Розрахунок витрат на матеріали і комплектуючі вироби

№ з/п	Найменування виробів	Вартість, грн.
1	Папір для принтера	40,00
2	Картридж та тонер для принтера	250,00
3	Література (доступ до Internet)	100,00
4	Непередбачені витрати	60,00
Разом		450,00

Загальний кошторис витрат на створення системи з урахуванням вищевказаного представлений у таблиці 2.

Таблиця 4.2 – Загальний кошторис витрат на створення системи

№ з/п	Найменування витрат	Вартість, грн.
1	Заробітна платня основна	10 000,00
2	Заробітна плата додаткова (20% від п.1)	2 000,00
3	Відрахування на соціальне страхування (38% від пп. 1 та 2)	4560
4	Витрати на амортизацію ЕОМ	1 188,00
5	Витрати на експлуатацію ЕОМ	68,59
6	Витрати на матеріали і комплектуючі вироби	450,00
7	Адміністративні витрати (50% від основної заробітної платні)	5 000,00
Разом на створення системи		23 266,59

До витрат на впровадження системи можна віднести витрати на придбання технічного забезпечення, вартість програмного забезпечення, вартість навчання кадрів, витрати на монтаж и настроювання мережі.

Оскільки параметри технічних засобів, які вже є, відповідають вимогам, то їх вартість при розрахунку витрат враховувати не будемо.

Виходячи з вимог до програмного забезпечення, у розробці програми буде використане програмне забезпечення NetBeans 6.8, яке розповсюджується безкоштовно, та JDK 1.0.6u17, який теж є безкоштовним. Вимогам відповідає також безкоштовна операційна система Linux Ubuntu 9.10.

Згідно з досвідом створення аналогічних програм, приймаємо, що вартість підготовки кадрів дорівнює 300,00 грн. Витрати на супроводження програмного забезпечення дорівнюватимуть 1% від вартості програми и становлять 232,67 грн.

Інвестиційні витрати на впровадження системи з урахуванням вартості навчання кадрів і витрати на супровід представлені в таблиці 3.3.

Таблиця 4.3 – Інвестиційні витрати на впровадження системи

№	Найменування інвестицій	Вартість, грн.
1	Вартість устаткування	0
2	Вартість програмного забезпечення	0
3	Вартість підготовки кадрів	300,00
5	Супроводження системи	232,67
Разом на впровадження системи		532,67

Разом загальна сума витрат на створення і впровадження системи складає 23 799,26 грн.

3 Розрахунок економічної ефективності розробки і впровадження програмного продукту

Основним показником економічної ефективності функціонування програмного забезпечення є підвищення ефективності керування інформацією у вигляді

зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Економічна ефективність впровадження програмного засобу «Alterantive» очікується за рахунок вивільнення робочого часу працівників та підвищення продуктивності праці. Крім того, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації роботи та своєчасне отримання необхідної інформації про пріоритетність варіантів рішень тощо.

Використання даної програми дозволяє вивільнити 0,5 робочого часу. Так як з системою працює один робітник, то система умовно вивільнить 0,5 робітника.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 0,5 робітника.

Зарплата 0,5 робітника у рік складає $\Delta C = 3000 \cdot 12 \cdot 0,5 = 18000$ (грн.).

Річний економічний ефект розраховується за формулою 3.4.

$$E_{\text{рік}} = \Delta C - C_{\text{супр}} - E_n \cdot k, \quad (4.4)$$

де $C_{\text{супр}}$ - вартість супроводження системи; E_n - нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь – для обчислювальної техніки приймається = 0,5; k - додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

$$E_{\text{рік}} = 18000 - 232,67 - 0,5 \cdot 23799,26 = 5867,7$$

Строк окупності системи розраховується за формулою 3.5.

$$I = \frac{k}{\Delta C - C_{\text{супр}}} \quad (4.5)$$

$$I = \frac{23799,26}{18000 - 232,67} = 1,34 \quad (\text{року}).$$

Тобто, система окупиться через 1,34 року, що приблизно дорівнює 16 місяцям.

5 ОХОРОНА ПРАЦІ

5.1 Актуальність проведення аналізу

Основні терміни і поняття у сфері охорони праці викладені в ДСТУ 2293-93 «Охорона праці. Терміни та визначення». Ним визначено, що охорона праці — це система законодавчих актів, соціально-економічних, організаційних, технічних і лікувально-профілактичних заходів і засобів, що гарантують безпеку, збереження здоров'я і працездатності людини в процесі праці.

Охорона праці щільно пов'язана з економічною сферою. Створення безпечних і здорових умов праці сприяє підвищенню її продуктивності. Підвищення продуктивності відбувається за рахунок регулярного виходу на роботу, зниження стомлюваності працюючих протягом робочого часу, його раціонального використання. Збереження трудових ресурсів відбувається за рахунок покращення стану здоров'я і підвищення середньої тривалості життя в результаті покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Знижується напруження в колективі, покращуються взаємовідносини між співробітниками. Підвищується професійний рівень за рахунок зростання кваліфікації та майстерності.

Цілком безпечних і нешкідливих виробництв не існує. Реальні виробничі умови характеризуються наявністю деяких небезпечних і шкідливих виробничих факторів.

Головною метою охорони праці є створення на кожному робочому місці безпечних умов праці, безпечної експлуатації обладнання, зменшення або повна нейтралізація дії шкідливих і небезпечних виробничих факторів на організм людини і, як наслідок, зниження рівня виробничого травматизму та профільних захворювань.

Завдання системи охорони праці — звести до мінімуму ймовірність ураження чи захворювання працюючого з одночасним забезпеченням якомога більш комфортних умов при максимальній продуктивності праці, а також формування у працівників знань і навичок, які сприяють гарантуванню безпеки праці, охороні

власного життя і здоров'я, життя і здоров'я підлеглих і людей, що перебувають поруч. До завдання охорони праці відносять також:

- гуманізацію праці, під якою розуміють профілактику перевтоми, професійних захворювань, запобігання виробничому травматизму, підвищення змістовності праці, створювання умов для всебічного розвитку особистості;
- знаходження оптимальних співвідношень між різними факторами виробничого середовища;
- впровадження норм гранично допустимих рівнів виробничих факторів, визначення ступеня шкідливості і небезпеки праці;
- розробка та планування заходів щодо поліпшення умов праці;
- забезпечення безпеки виконання робіт працівниками;
- впровадження технічних засобів і заходів щодо боротьби з травматизмом і профзахворюваннями;
- розробка методів оцінки соціальної та економічної ефективності заходів з удосконалення умов і охорони праці.

Охорона праці включає декілька розділів: організаційно-правові питання, техніка безпеки, виробнича санітарія і пожежна профілактика.

Законодавчі і нормативно-правові акти з питань охорони праці визначають відповідальність працівників у сфері охорони праці, систему контролю на підприємстві, порядок розробки інструкцій з охорони праці т.і. Нормативні акти про охорону праці можуть бути міжгалузеві і галузеві. Їм надано чинність правових норм, обов'язкових для виконання. Вони розробляються, затверджуються або узгоджуються уповноваженим на це урядовим органом державного управління.

Наука про охорону праці тісно пов'язана з іншими науками. Вона широко використовує найновіші досягнення науки і техніки, базується на теоретичних розробках з фізики, хімії, математики, електроніки, медицини, економіки тощо. Важливе місце в розробці питань охорони праці займають такі наукові дисципліни, як ергономіка, інженерна психологія і фізіологія праці, технічна естетика. Взаємодія людини і техніки в системі виробництва (система «людина

— машина — виробниче середовище») має розглядатися під час проектування і створення безпечних умов праці, вирішення завдань оптимізації. Це і є предметом ергономіки. В період широкого застосування нової техніки в усіх галузях народного господарства проблема оптимізації взаємовідносин людини з машиною і виробничим середовищем стала однією з головних.

5.2 Аналіз шкідливих та небезпечних факторів

Чинники зовнішнього середовища в першу чергу впливають на функціонування органів дихання, зору, слуху, системи кровообігу людини. До чинників зовнішнього середовища відносять метеорологічні умови виробничих приміщень, склад повітряного середовища, освітленість, робочого місця, шум і вібрація, різні випромінювання.

5.2.1 Метеорологічні умови

До них відносяться температура, відносна вологість, швидкість руху і тиск повітря. Ці чинники визначають мікроклімат виробничих приміщень. Результати досліджень свідчать про те, що у виробничих умовах усі метеорологічні фактори впливають на людину одночасно.

Розглядаючи механізми впливу метеорологічних факторів виробничого середовища (температури, вологи, швидкості руху повітря, чинності променевої енергії нагрітих деталей і агрегатів) на людину, маємо на увазі, що людський організм прагне підтримати відносну динамічну сталість своїх функцій за різноманітних метеорологічних умов. Ця сталість забезпечує насамперед один з найважливіших фізіологічних механізмів — механізм терморегуляції. Вона спостерігається при певному співвідношенні теплоутворення (хімічної терморегуляції) і тепловіддачі (фізичної терморегуляції).

Відомо, що надлишкова вологість повітря негативно впливає на механізм терморегуляції організму. Особливо шкідливою є вологість повітря, яка перевищує 70—75 % за температури 30 °C і більше. За даними М. Є. Маршакова і В. Г. Давидова (1985), верхньою межею термальної рівноваги людини, що

перебуває у стані спокою, є температура повітря 30—31 °С за відносної вологості 85 % або 40 °С градусів за відносної вологості 30 %. Ці межі змінюються при виконанні фізичної роботи.

Фізична робота в умовах підвищеної температури призводить до прискорення серцебиття, зниження артеріального тиску. За низької температури може статися переохолодження організму, що спричинить простудне захворювання.

Згідно з результатами досліджень людина є працездатною і нормально себе почуває, якщо у службових приміщеннях температура повітря взимку підтримується від: +19 ОС літом - до: +24 ОС; відносна вологість: $50 \pm 10\%$, рухливість повітря: 0,15...0,3 м/с.

5.2.2 Захист від шуму і вібрації

Виробничі процеси супроводжуються шумом і вібраціями, джерелом виникнення яких є машини з неврівноваженими масами, що обертаються, технологічні установки і апарати, в яких переміщення рідин і газів відбувається з великими швидкостями і супроводжується пульсацією.

Вібрація робочого устаткування не повинна співпадати по частоті з частотою власних коливань тіла людини і його внутрішніх органів (4.400 Гц).

Тривала дія шуму знижує гостроту слуху, приводить до зміни кров'яного тиску, послаблює увагу, знижує гостроту зору, викликає зміни в рухових центрах. Особливо несприятлива дія шуму на серцево-судинну і нервову системи. Шум інтенсивністю більше 130 дБ викликає біль у вухах, а при 140 дБ настають необоротні порушення органу слуху.

5.2.3 Освітлення виробничого приміщення

Освітлення виробничих приміщень – один з основних факторів зовнішнього виробничого середовища, що впливають на людину в процесі праці [7, с. 3]. Безпечною та високопродуктивною праця може бути лише за умови гарного освітлення робочих місць.

Недостатнє та нерівномірне освітлення робочих місць призводить до перенапруги зору, перевтоми організму, до виникнення надмірної нервової дратівливості, ослаблення уваги. Доведено, що в половини працюючих у таких умовах спостерігаються дефекти зору – короткозорість, зниження гостроти зору та інше, а в 38% чоловіків виникає частковий дальтонізм.

Для освітлення приміщень університету використовується як природні, так і штучні джерела освітлення. Природне освітлення, через вікна, характерно для світлого часу доби, штучне освітлення – здійснюється лампами розжарення.

Розрізняють загальне, комбіноване і місцеве штучне освітлення. При загальному освітленні мінімальна освітленість 75 Лк при використанні ламп розжарювання, і 150 Лк - при використанні газорозрядних ламп.

Комбіноване освітлення складається із загального освітлення робочого приміщення і місцевого освітлення робочих поверхонь, приладів і т.д. Мінімальна освітленість їх повинна складати не менш 300 Лк.

Для місцевого освітлення використовуються світильники з відбивачами. Застосовувати тільки місцеве освітлення заборонено.

Аварійне освітлення для короткочасних виробничих робіт повинне включатися при відключенні основного освітлення, працювати від автономного джерела і складати не менше 10% від норми освітлення цих місць.

Освітлення приміщення повинне відповідати характеру зорової роботи. Необхідно забезпечити рівномірний розподіл яскравості на робочій поверхні, а також навколишнього простору.

5.2.4 Протипожежний захист

Для гасіння пожеж в початковій стадії застосовують пінні хімічні вогнегасники. Промисловість випускає вогнегасник ВХП-10 (вогнегасник хімічний, пінний, модель 10). Забороняється застосовувати цей тип вогнегасників для гасіння електроустановок, що знаходяться під напругою, оскільки піна, що утворюється, електропровідна. Час роботи вогнегасника 60...65 с, тому при його застосуванні необхідно приводити його в дію безпосередньо у вогнища пожежі.

Вуглекислотні вогнегасники застосовують для гасіння пожеж: у електроустановках, що знаходяться під напругою; вуглекислота не викликає псування матеріалів. Промисловість випускає вогнегасники ВВ-2, ВВ-5, ВВ-8 (вуглекислотні вогнегасники місткістю 2, 5, 8 л).

Водопостачання адміністративної будівлі повинно бути спроектовано так, щоб воно окрім задоволення господарських та питних потреб повністю забезпечувало подачу води у будь-який час доби в кількості, необхідному для пожежегасіння усередині будівлі.

Одним з найбільш надійних і доконаних видів водопостачання є водопровід, який розраховується на безперебійне отримання необхідної кількості води як для виробничо-господарських, так і для протипожежних потреб.

5.3 Розрахунок системи штучного освітлення приміщення відділу стратегічного планування приймальної компанії

Система освітлення повинна забезпечити:

- достатню освітленість і рівномірний розподіл світлового потоку на робочій поверхні;
- сталість величини освітлення;
- виключення глибоких тіней;
- відсутність різкої різниці в яскравості робочих поверхонь і навколишнього фону (стін, стелі);
- відсутність сліпучої дії джерела світла великої яскравості.

У даному підрозділі буде приведений розрахунок освітлення приміщення відділу стратегічного планування приймальної компанії НУК. Дане приміщення характеризується наступними розмірами:

- довжина приміщення $A = 6$ м;
- ширина приміщення $B = 4$ м;
- висота приміщення $H = 2,7$ м.

Висота робочої поверхні над підлогою у приміщенні відділу стратегічного планування приймальної компанії $h_p = 1$ м.

Освітлення приміщення здійснюється з використанням стандартного світильника з лампою розжарювання типу УПМ-15.

Також у якості початкових даних виступає рівень мінімальної освітленості $E_{\min}$, що визначається за таблицею норм освітленості робочих поверхонь у виробничих приміщеннях [7, с. 9]. Для нашого випадку при роботі за комп'ютером $E_{\min} = 100$ лк.

Для розрахунків необхідно встановити приблизні значення коефіцієнтів відбиття ρ стін, стелі та підлоги за відповідною таблицею. Приміщенні, що розглядається, має побілені стіни при незавішених вікнах та побілену стелю, тому їхні коефіцієнти відбиття $\rho_{\text{стін}} = 50\%$, $\rho_{\text{стелі}} = 50\%$, робочі столи та підлога темні, їх коефіцієнт відбиття $\rho_{\text{підлоги}} = 10\%$.

Параметри, що описують розміщення світильника в приміщенні приведені на рис. 6.1. На вказаному рисунку присутні наступні параметри:

- висота приміщення H , м;
- відстань від стелі до робочої поверхні H_0 , м;
- висота підвішування світильника над підлогою $H_{\text{п}}$, м;
- висота підвішування світильника над освітлюваною поверхнею h , м;
- відстань від стелі до світильника h_c , м;
- висота робочої поверхні над підлогою h_p , м.

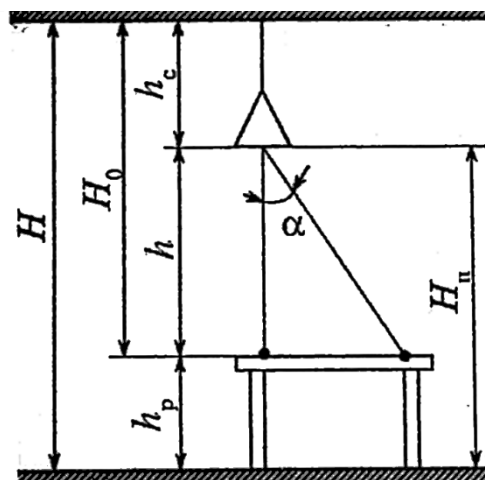


Рисунок 5.1 – Опис розміщення світильника в приміщенні

Відстань від стелі до робочої поверхні розраховується за формулою:

$$H_0 = H - h_p \quad (5.1)$$

Для нашого приміщення:

$$H_0 = 2,7 - 1 = 1,7 \text{ м}$$

Відстань від стелі до світильника розраховується за формулою:

$$h_c = 0,2 * H_0 \quad (5.2)$$

Для нашого приміщення:

$$h_c = 0,2 * 1,7 = 0,34 \text{ м}$$

Висота підвісу світильника над поверхнею, що освітлюється, розраховується за формулою:

$$h = H_0 - h_c \quad (5.3)$$

Для нашого приміщення:

$$h = 1,7 - 0,34 = 1,36 \text{ м}$$

Висота підвісу світильника над підлогою розраховується за формулою:

$$H_{\Pi} = h + h_p \quad (5.4)$$

Для нашого приміщення:

$$H_{\Pi} = 1,36 + 1 = 2,36 \text{ м}$$

Однією з характеристик є відстань між центрами світильників L , м.

Відношення L/h повинно бути в межах 1,4–1,8 при паралельному розміщенні світильників, або 1,8–2,5 при шаховому розташуванні. Для досягнення рівномірності освітлення приймаємо відношення $L/h = 1,5$. Тоді:

$$L = 1,5 * h = 1,5 * 1,36 = 2,04 \text{ м.}$$

Необхідна кількість світильників розраховується за формулою:

$$N = S/L^2 = (A*B)/L^2 \quad (5.5)$$

Для нашого приміщення:

$$N = (6*4)/2,04^2 = 5,77 = 6 \text{ світильників}$$

Розмістимо світильники у 2 ряди по 3 світильники.

Індекс приміщення розраховується за формулою:

$$i = (A*B)/(h*(A+B)) \quad (5.6)$$

У нашому випадку:

$$i = (6*4)/(1,36*(6+4)) = 1,76$$

Використовуючи i , $\rho_{\text{стін}}$, $\rho_{\text{стелі}}$, $\rho_{\text{підлоги}}$ знаходимо коефіцієнт η використання світлового потоку за відповідною таблицею [7, с. 43] для світильника типу УПМ-15. Для наших даних він дорівнює $\eta = 0,53$.

Світловий потік однієї лампи розраховується за формулою:

$$\Phi_p = (E_{\min} * S * K_3 * z) / (N * \eta), \quad (5.7)$$

де $S = A * B$;

K_3 – коефіцієнт запаса, дорівнює 1,3 з відповідної таблиці [7, с. 14];

z – тип ламп, для УПМ-15 $z = 1,5$.

Для нашого приміщення світловий потік однієї лампи:

$$\Phi_p = (100 * 6 * 4 * 1,3 * 1,5) / (6 * 0,53) = 1471,7 \text{ лм}$$

За знайденим світловим потоком з таблиці "Параметри ламп розжарювання загального призначення з розрахунковими напругами 130 і 220 В" [7, с. 24] обираємо лампу БК215-225,100 потужністю $P_{\text{л}} = 100$ Вт, що має світловий потік 1450 лм, найбільш близький до розрахункового.

Таким чином фактична освітленість $E_{\text{ф}}$ буде дорівнювати:

$$E_{\text{ф}} = E_{\min} * (\Phi_{\text{л}} / \Phi_p) \quad (5.8)$$

Для приміщення відділу стратегічного планування:

$$E_{\text{ф}} = 100 * (1450 / 1471,7) = 98,53 \text{ Лк}$$

Загальна потужність ламп розраховується за формулою:

$$P_{\text{заг}} = P_{\text{л}} * N \quad (5.9)$$

Для приміщення відділу стратегічного планування:

$$P_{\text{заг}} = 100 * 6 = 600 \text{ Вт}$$

5.4 Розроблення заходів щодо зменшення впливу шкідливих та небезпечних факторів

Зниження негативного впливу мікроклімату у приміщенні відділу стратегічного планування можна досягти за рахунок вжиття таких заходів:

- захисту працівників різними видами екранів;
- раціональної теплової ізоляції устаткування;

- раціонального розміщення устаткування;
- ефективного планування і конструкторського рішення виробничих приміщень;
- раціональної вентиляції та опалювання;
- раціоналізації режимів праці й відпочинку, перерви.

Для зменшення вібрацій і шуму від комп'ютерів передбачають використовувати рівне покриття підлоги та якісно виконані меблі, що не матимуть нестійкостей, які можуть спричинити додаткові вібрації. Приміщення відділу стратегічного планування має бути достатньо великим, щоб уникнути накладання шуму від роботи декількох комп'ютерів.

Освітлення приміщення відділу стратегічного планування необхідно виконати згідно з результатами розрахунків, що приведені у підрозділі 6.3, його необхідно оснастити шістьма лампами розжарювання типу БК215-225,100 потужністю 100 Вт, які слід розмістити у два ряди по три світильника.

Стосовно протипожежного захисту рекомендується оснастити приміщення відділу стратегічного планування одним вуглекислотним вогнегасником місткістю 5 літрів, ВВ-5, вуглекислота не викликає псування матеріалів. Вогнегасник типу ВХП-10 не застосовують для гасіння електроустановок, що знаходяться під напругою (до яких відносяться і комп'ютерна техніка), оскільки піна, що утворюється, електропровідна.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Актуальність охорони навколишнього середовища

Будь яка країна, що стала на шлях науково-технічного прогресу та широкомасштабного використання його результатів, вже не може і не повинна ігнорувати такі об'єктивні чинники, як вичерпаність багатьох природних ресурсів, насамперед невідтворювальних і невідновлюваних, вразливість навколишнього середовища, екологічну стійкість та екологічну місткість довкілля, межі його екологічної міцності і опірності щодо негативних і шкідливих антропогенних впливів тощо. Всі ці чинники необхідно всебічно враховувати в господарській діяльності і при визначенні темпів та масштабів соціально-економічного розвитку на майбутнє. Цей розвиток має бути врівноваженим і адекватним екологічній ситуації, узгоджуватися з природничими законами. А це можливо тільки за умови, що виробниче господарська діяльність суспільства ґрунтуватиметься на концепції сталого екологічного розвитку [8, с. 16].

Охорона навколишнього природного середовища, раціональне використання природних ресурсів, забезпечення екологічної безпеки життєдіяльності людини - невід'ємна умова сталого економічного та соціального розвитку України.

З цією метою Україна здійснює на своїй території екологічну політику, спрямовану на збереження безпечного для існування живої і неживої природи навколишнього середовища, захисту життя і здоров'я населення від негативного впливу, зумовленого забрудненням навколишнього природного середовища, досягнення гармонійної взаємодії суспільства і природи, охорону, раціональне використання і відтворення природних ресурсів.

За законом України про охорону навколишнього природного середовища [9, с. 3] завданням законодавства про охорону навколишнього природного середовища є регулювання відносин у галузі охорони, використання і

відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

6.2 Забруднення навколишнього середовища

Територія університету повинна постійно утримуватися у чистоті і порядку. Все сміття необхідно систематично вивозити.

Джерелами забруднення навколишнього середовища НУК є:

- дощові води;
- побутове та будівельне сміття;
- автотранспорт;
- котельня.

З огляду на те, що на території НУК присутні недобудовані корпуси і насипні будівельні матеріали (пісок, щебінь та ін.), дощові води мають негативний вплив на навколишнє середовище, розмиваючи незахищені будівельні ділянки та забруднюючи ґрунти. Але порівняно з виробництвами ці забруднення є не дуже значними.

НУК як і будь-яке інше підприємство або організація є джерелом побутового та будівельного сміття. У складі забруднюючих речовин цієї категорії переважають побутове сміття, харчові відходи, будівельне сміття, відходи опалювальних систем, предмети домашнього ужитку, що прийшли в непридатність, і т.п. У масштабах університету, порівняно з житловими будинками, наявні незначні кількості такого сміття. Воно утилізується централізовано, як і сміття суміжних житлових будинків.

Найактивніше забруднюється атмосферне повітря, через яке небезпечні речовини поширюються на інші компоненти, підвищуючи тим самим уже існуючий в них рівень забруднення.

Дуже гостро стоїть проблема забруднення атмосферного повітря автотранспортом, на який припадає понад 1/3 загального обсягу країни викидів в атмосферу. Вже тепер у багатьох великих містах викиди автотранспорту перевищують 70 % у загальному обсягу викидів. При роботі двигунів внутрішнього згорання інтенсивно виділяються оксиди азоту, свинець, вуглеводні, оксид вуглецю, сажа і інші речовини, що осідають на поверхню землі або поглинаються рослинами. У останньому випадку ці речовини також потрапляють в ґрунт і залучаються до круговороту, пов'язаного з харчовими ланцюгами. В результаті неповного згорання в двигуні частина вуглеводнів перетворюється в сажу, яка містить смолисті речовини, в тому числі канцерогенний бенз(а)пірен. Надзвичайно шкідливими у вихлопах є сполуки неорганічного свинцю. Так, в 1 л бензину міститься 1 г тетраетилсвинцю, який після згорання викидається в атмосферу.

Поряд з НУК знаходиться університетська автостоянка, яка межує з автозаправкою. Але вона не дуже значна за розмірами, бо у такому місті як Миколаїв не дуже багато людей даної категорії можуть дозволити собі користування особистим автотранспортом. Але наявність автозаправки безпосередньо біля університету порушує законодавчі норми.

Одними із головних джерел хімічного забруднення повітря є котельні установки. Вони виділяють дим, що утворюється в процесі спалювання твердого чи рідкого палива. Ступінь забруднення атмосфери продуктами згорання палива залежить від виду та якості палива, а також від технічних характеристик опалювального устаткування. Основними викидами при цьому є продукти повного (окиси сірки та азоту) та неповного згорання (головним чином, окис вуглецю, сажа, вуглеводні). Спостерігаються значні викиди оксид азоту, які утворюються в основному з азоту повітря при високих температурах горіння. Теплові електростанції, які спалюють нафту, майже не викидають золи, зате викидають в 3 рази більше сірчаного ангідриду.

Численні види палива поділяються на тверді - вугілля різних марок і дрова; рідкі - мазути, різні нафти і похідні від них; газоподібні - природний, коксовий,

кам'яновугільний газ, а також газ, які виділяються з нафтоносних пластів. Найчистішим із всіх видів палива вважається природний газ.

При повному згорянні палива в атмосферу надходить двооксид вуглецю (CO_2) і пари води, які самі по собі не є шкідливими домішками, а також невеликі кількості оксидів азоту і сірки. При неповному згорянні палива утворюються зола і шкідливі газ.

Основні шкідливі газ, які утворюються при неповному згорянні палива такі:

а) сполуки вуглецю:

- вуглекислий газ CO_2 - нешкідливий при малих концентраціях;
- оксид вуглецю (чадний газ) CO - дуже токсичний, але швидко дифундує в атмосферу;
- неспалені вуглеводні або окисзовані речовини (альдегіди та кислоти);

б) сполуки сірки: сірчистий ангідрид SO_2 , який може переходити в сірчаний ангідрид SO_3 і при наявності води або водяної пари утворювати сірчану кислоту H_2SO_4 ;

в) сполуки азоту: NO , NO_2 , сприятливі умови для їх утворення виникають при високих температурах;

г) сажі - неспалені частки вугілля або інших видів палива, вони мають розміри близько 1 мкм і дуже легкі, але можуть об'єднуватися між собою, створюючи крупні конгломерати;

д) дими - аеродисперсні системи, до яких входять найдрібніші тверді частинки, при спіканні створюються більш великі конгломерації і газ;

е) пил складається з часток вугілля, кількість яких іноді досягає 50 % складу, а також часток золи і породи.

6.3 Розрахунок викидів забруднюючих речовин в атмосферу від котлоагрегату котельні

При роботі котлоагрегату кількість викидів твердих частинок леткої золи (т/рік, г/с), що викидається в атмосферу з димовими газами котлоагрегату в одиницю часу при спалюванні мазута, розраховується по формулі:

$$M_{\text{п. неорг.}} = B \cdot AP \cdot f \cdot (1 - h_3), \quad (6.1)$$

де B – витрата палива за певний проміжок часу (т/рік, г/с);

AP – зольність палива у робочому стані, %;

f – коефіцієнт, який залежить від виду палива і типу топки;

h_3 – частка твердих частинок, які затримуються золоуловлювачем.

За формулою 6.1 розрахуємо приблизну кількість викидів твердих частинок за рік та за секунду:

$$M_{\text{сажа}} = 800 \cdot 0,1 \cdot 0,01 \cdot (1 - 0) = 0,800 \text{ т/рік};$$

$$M_{\text{сажа}} = 84,722 \cdot 0,1 \cdot 0,01 \cdot (1 - 0) = 0,085 \text{ г/с}.$$

Кількість викидів оксидів сірки (в перерахунку на SO_2), що викидаються в атмосферу з димовими газами котлоагрегату в одиницю часу (т/рік, г/с), розраховується за формулою:

$$M_{\text{SO}_2} = 0,02 \cdot B \cdot S_p \cdot (1 + \eta^{\text{I}}_{\text{SO}_2}) \cdot (1 - \eta^{\text{II}}_{\text{SO}_2}), \quad (6.2)$$

де B – витрата палива за певний проміжок часу (т/рік, г/с);

S_p – вміст сірки в паливі на робочу масу, %;

$\eta^{\text{I}}_{\text{SO}_2}$ - частка оксидів сірки, що зв'язуються леткою золою;

$\eta^{\text{II}}_{\text{SO}_2}$ - частка оксидів сірки, що не зв'язуються леткою золою.

З метою покращення санітарно-гігієнічних умов в робочій зоні і зниження викидів забруднюючих речовин в атмосферу рекомендовано впровадження в університеті наступних заходів:

- а) переведення роботи котла на території головного корпусу НУК з рідкого палива на природний газ;
- б) вжити заходів щодо перенесення автозаправки за територію головного корпусу;
- в) у найближчий час припинити будівництво та прибрати залишкові будматеріали якщо будівництво не планують завершити або перенести з відкритого повітря до критих споруд сипкі будівельні матеріали;
- г) як екологічну очистку території НУК запровадити додаткове озеленення території університету

ВИСНОВКИ

У даній роботі були досліджені методи підтримки прийняття рішень на основі нечіткої математики та розроблений програмний засіб підтримки прийняття рішень з їх використанням. У реалізації був використаний метод аналізу ієрархій Т.Сааті. Програмний засіб дозволяє встановлювати пріоритети альтернатив, виходячи з даних про ієрархію критеріїв та надані користувачем попарні порівняння.

Для цього у роботі було виконано:

- аналіз та порівняння методи підтримки прийняття рішень на основі нечіткої математики, зроблені висновки й обраний найбільш доцільний для реалізації метод;

- б) виконана постановка задачі на розробку програмного засобу реалізації;

- в) розроблено проект програмного продукту, який складається з:

- 1) ескізного проекту (проектування інтерфейсів програми, розробка специфікацій);

- 2) технічного проекту (розробка алгоритмів та логічних зв'язків);

- 3) робочого проекту (розробка фізичного представлення даних, вибір програмних засобів, кодування, тестування, випробування та розробка документації);

- г) розрахована економічну ефективність розробки і впровадження програмного продукту;

- е) проаналізовані програмні засоби підтримки прийняття рішень при ліквідації надзвичайних ситуацій;

- є) зроблений опис охорони праці і охорону навколишнього середовища при роботі з ПК у відділі програмного забезпечення;

- г) підсумовані загальні результати роботи.

Програма має перспективи для використання і застосовується.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ясінецька І.А. Державний земельний кадастр як інформаційна база/ Економіка і суспільство/2018. – 680 с.
2. Тимошевська Т.І. Застосування гіс-технологій під час виконання кадастрових робіт у населених пунктах/ Вчені записки ТНУ імені В.І. Вернадського/2021. – 283 с.
3. Попов А.С. Кадастрові та реєстраційні системи країн світу/ Монографія/2018. – 216 с.
4. Enemark S. Cadastral Systems – the Nordic Approach / S. Enemark / The Danish Journal for Mapping and Land Use. – Landinspektoren, 2019. – No. 5. – P. 464-471.
5. Enemark S. Land Administration Infrastructures for Sustainable Development / S. Enemark / The Seventh United Nations Regional Cartographic Conference for the Americas "On Spatial Data and Development: Building a Sustainable Infrastructure", 22-26 January, 2021, New York [Електронний ресурс]. – Режим доступу: http://www.fig.net/council/enemark_papers/unrcca_enemark_1.pdf
6. Sevatdal H. Land Administration in Norway / H. Sevatdal [Електронний ресурс]. – Режим доступу: http://www.myland.org.ua/userfiles/file/SEVATDAL_ML.pdf
7. Rossinelli S. The Swedish Cadastral System: Essay Report / S. Rossinelli [Електронний ресурс]. – Режим доступу: <http://www.geo21.ch/ethz/20018/essays/E1-Schweden.pdf>
8. Кошкарев А.В. Инфраструктура пространственных данных Финляндии / А.В. Кошкарев // Пространственные данные. – 2018. – №1. [Электронный ресурс]. – Режим доступу: <http://www.gisa.ru/44536.html>
9. Vitikainen A. Land Consolidation and the Guidelines on Land Administration / A. Vitikainen [Електронний ресурс] – Режим доступу:

- http://portal.survey.ntua.gr/main/labs/photo/research/wg_33/wpla/papers/TPS6.4.Arvo%20Vitikainen.doc
10. Евдокимов П. В. С. Практическое руководство / Наука і Техніка /2022 – 416 с.
 11. Черін, А.Г. Стандартизація геоінформаційних сервісів [Текст] // Вісн. геодез. та картогр. – 2019. – № 4. – С. 34-39.
 12. Lithuania: Country Report 2010 [Електронний ресурс] – Режим доступу: <http://www.cadastraltemplate.org/countryreport/Lithuania-27Sep2021.pdf>
 13. Остапцов С.А. Документообіг по охороні праці на підприємстві – загальні запитання / С.А. Остапцов, В.В. Новиков, С.А. Солод // Чрезвычайні ситуації: промислова та екологічна безпека. – 2018. – № 1. – С. 52-56. – ISSN 2305-6843
 14. Марк Прайс. C# 9 и .NET 5 Разработка и оптимизация / / C# 9 and .NET 5 – Modern Cross-Platform Development 5th Edition / 2022 – 832 с.
 15. Лященко А.А./ Сервіс-орієнтована архітектура кадастрових геоінформаційних систем та кадастрових геопорталів / 2018 – 256 с.
 16. Про затвердження Вимог до структури, змісту та формату оформлення результатів робіт із землеустрою в електронному вигляді (обмінного файлу) // Наказ Держкомзему від 02.11.2009 р. № 573, зареєстрований в Мінюсті України 15.02.2010 № 157/17452. URL: <https://zakon.rada.gov.ua/laws/show/z0157-10#Text>
 17. Лященко А.А., Ціпенко О.В. Наскрізні геоінформаційні технології грошової оцінки земель населених пунктів. Інженерна геодезія. Київ: КНУБА, 2019. Вип. 42. – С. 155–160.
 18. Закону України «Про державний земельний кадастр» [Електронний ресурс]. – Режим доступу: <https://xn--80aagahqwyibe8an.com/zakon-ukrajiny/stattya-printsipi-derjavnogo-zemelnogo-116636.html>
 19. Карпінський, Ю.О. Технологічні моделі розподілених кадастрових ГІС в Intranet/Internet-мережах [Текст] / Ю.О. Карпінський, А.А. Лященко //

- Містобудування та територіальне планування. – Вип.17. – К.: КНУБА, 2020. – С. 106
- 20.Постановою Кабінету Міністрів України. Порядок інформаційної взаємодії між кадастрами та інформаційними системами [Електронний ресурс]. – Режим доступу: www.tax.gov.ua/data/normativ/000/000/63039/ZATVERDZHENO.doc
 21. Бардус І.О., Лазарєв М.І., Ніценко А.О.. «Бази даних у схемах (на основі фундаменталізованого підходу)». - 2017
 - 22.Лященко, А. А. ... // Містобудування та територіальне планування : науково-технічний збірник / МОН України, КНУБА. – Київ : КНУБА, 2024 – С. 122
 - 23.Остапцов С.А. Документообіг по охороні праці на підприємстві – загальні запитання / С.А. Остапцов, В.В. Новиков, С.А. Солод // Чрезвычайні ситуації: промислова та екологічна безпека. – 2018. – № 1. – С. 52-56. – ISSN 2305-6843
 - 24.Петрова, Корощенко, Айзман. Охорона праці а підприємстві та навчальному процесі Петрова, Корощенко, Айзман. – 2017. – 336 с.
 - 25.Основи екології та охорони навколишнього середовища:навчально-методичний посібник / Е.Т. Тимонова, И. А. Тимонов. –2011. – 228 с.
 - 26.Pandas Development Team. pandas: powerful Python data analysis toolkit [Електронний ресурс]. Режим доступу: <https://pandas.pydata.org>.
 - 27.Scikit-learn Developers. Scikit-learn: Machine Learning in Python [Електронний ресурс]. Режим доступу: <https://scikit-learn.org/stable/> (дата звернення: 15.11.2025).
 - 28.TensorFlow Development Team. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Електронний ресурс]. Режим доступу: <https://www.tensorflow.org/>
 - 29.OMG. UML Specification — Unified Modeling Language [Електронний ресурс]. 2025. Режим доступу: <https://www.omg.org/spec/UML/>.

- 30.Bootstrap Team. Bootstrap Documentation 5.3 [Электронный ресурс]. 2025. Режим доступа: <https://getbootstrap.com/docs/5.3/>.
- 31.Flask Development Team. Flask Documentation (3.0.x) [Электронный ресурс]. 2025. Режим доступа: <https://flask.palletsprojects.com/en/3.0.x/>.
- 32.SQLite Consortium. SQLite Documentation [Электронный ресурс]. 2025. Режим доступа: <https://www.sqlite.org/docs.html>

ДОДАТОК ТЕХНІЧНЕ ЗАВДАННЯ

Програмний продукт призначений для підтримки прийняття рішень у багатокритеріальних задачах з ієрархічними структурами цілей, які включають як осяжні, так і неосяжні фактори. Він повинен надавати змогу встановлювати попарні якісні характеристики для оцінюваних факторів та розраховувати міру пріоритетів альтернатив вибору.

Програма повинна бути написана на об'єктно-орієнтованій мові програмування, бути мультиплатформовою та надавати дружній інтерфейс користувачу.

Для цього необхідні такі функціональні можливості:

- надавати можливість формування ієрархічної структури критеріїв необмеженої глибини і задавати потрібні альтернативи. Критерії представляються у вигляді дерева, альтернативи – у вигляді списку. Це потребує створення форми редагування з двома відділами: візуального редагування дерева, представленого вбудованим компонентом, та візуального редагування списку, який теж має бути представлений вбудованим компонентом;
- за допомогою вказаного редактора користувач може додавати задачі, редагувати їх та видаляти;
- надавати можливість перегляду існуючих задач, виводячи усі збережені у вигляді списку, та вибору з цього списку потрібної;
- шляхом опитування користувача після вибору задачі давати йому змогу визначити якісні оцінки пріоритетів.
- за створеною ієрархією згідно використовуваному методу Сааті визначати, через які саме порівняння потрібно провести користувача у опитуванні, та у зручній компактній формі виводити сформовані комплекти пар;

- уведення пріоритету повинно відбуватися у візуальному режимі з відображенням і динамічною зміною якісної характеристики;
- надавати можливість змінити попередні судження, як під час введення пріоритетів, так і після отримання результату;
- розраховувати пріоритети альтернатив за методом аналізу ієрархій;
- повідомляти про узгодженість розрахунків та надавати можливість зміни суджень;

Крім того, необхідно розробити структуру даних для зберігання інформації про склад вирішуваних задач, які повинні зберігатися на комп'ютері користувача у файлах. Ця інформація повинна відображати ієрархічну структуру, задану користувачем.

Вимоги до надійності передбачають зберігання даних про задачі кожного користувача на комп'ютері окремо, так що кожний з них має доступ тільки до своїх задач, але при бажанні може скопіювати файл з інформацією про задачу для іншого. Також програмний засіб повинен перевіряти дані, що зберігаються, на коректність та цілісність, і в разі її порушення виводити відповідні повідомлення.

Вимоги до складу та характеристик технічних засобів передбачають використання ПК будь-якої операційної системи зі встановленою JRE 1.6 або вище, яка має не менш 512МБ оперативної пам'яті.

Вимоги до інформаційної та програмної сумісності впливають із наведених потреб встановлюють мовою написання програмного засобу Java та збереження файлів у форматі XML.

Вимоги до документації встановлюють необхідність таких документів:

- тексти програмних модулів;
- тексти JUnit-тестів;
- опис програмного засобу;
- інструкція користувача;
- методика випробувань.

ДОДАТОК ПРОГРАМА Й МЕТОДИКА ВИПРОБУВАНЬ

1 Загальний опис

Об'єкт випробувань: розроблений програмний засіб «Alternative».

Ціль випробувань: перевірити відповідність розробленого засобу вимогам технічного завдання та його коректну роботу без збоїв.

Так як узгодженість з використанням методом і правильність розрахунків були перевірені за допомогою тестування з JUnit, то при випробуваннях варто перевірити інші параметри роботи.

Вимоги до програми:

- відповідність усім вимогам, вказаним у технічному завданні;
- коректна робота при будь-якій послідовності дій користувача;
- неможливість створення користувачем ситуації зберігання некоректних даних;
- у разі введення некоректних даних або виникнення таких у файлі програма повідомлює про це і продовжує свою роботу без збоїв;

Вимоги до програмної документації:

- доступна для розуміння;
- документація цілком відображає сутність програми;
- після прочитання документації у цілком непосвідченого у контекст розробки користувача не залишається питань щодо використання програми.

2 Засоби та порядок випробувань

Для випробувань потрібно мати декілька записів користувачів для однієї операційної системи.

Процедури перевірки включають такі:

- створити задачу, зберегти її, після чого перевірити відповідність збережених даних введенням;

- увійти в операційну систему під кожним користувачем та створити декілька задач, після чого перевірити, щоб користувачі не мали доступу до «чужих» задач;
- передати копіюванням задачу від одного користувача іншому і перевірити коректне її відображення у іншого;
- перевірити можливість збереження задачі тільки з кількістю критеріїв на одному рівні та альтернатив від 2 до 9;
- перевірити неможливість введення «пустих» критеріїв та альтернатив;
- змінити без допомоги програми збережений файл та перевірити роботу програми без збоїв.

3 Методи випробувань

Автоматичні тести з наборами тестів були створені при тестуванні, тому методи даних випробувань включають візуальну перевірку у відповідності з діями, що вказані у порядку випробувань.

Після перевірки правильності роботи програми цими методами доцільним є випробування програми саме з точки зору застосування при прийнятті рішень. З цією метою програму було надано у загальноосвітню школу №61 та у дослідницько-проектний центр кораблебудування КП «ДПЦК». Їх висновки та результати роботи програми наводяться у додатках.

ДОДАТОК ТЕКСТ ПРОГРАМИ РОЗРОБЛЕНИХ МОДУЛІВ

```

class TaskController < ApplicationController
  before_filter :access?, :only => [:new, :create, :delete, :edit, :update]
  layout "task", :except => [:rss_feed, :ban]
  def rss_feed
  end
  def ban
  end
  def show
    @task = Task.find(params[:id])
    @comments = Comment.find :all, :conditions => ["task_id = ?", @task.id]
  end
  def list
    @tasks = Task.find(:all)
  end

  def new
  end

  def create
    @task = Task.new(@params[:task])
    if @task.save
      redirect_to :action => "list", :controller => "task"
    else
      render :action => "new", :controller => "task"
    end
  end

  def delete
    @task = Task.find(params[:id])
    Comment.find(:all).each do |comment|
      if comment.task_id == @task.id
        comment.destroy
      end
    end
    @task.destroy
    redirect_to :action => "list", :controller => "task"
  end

  def edit
    @task = Task.find(@params[:id])

```

```

end

def update
  @task = Task.find(@params[:id])
  if @task.update_attributes(params[:task])
    redirect_to :action => "list", :controller => "task"
  else
    render :action => "edit", :id => @task.id, :controller => "task"
  end
end

end

class CommentController < ApplicationController

  def create
    @comment = Comment.new(params[:comment])
    @article = Article.find(params[:id])

    @time = Time.now
    @comment.time = @time
    @comment.article_id = @article.id
    if @comment.save
      @comments = Comment.find :all, :conditions => ["article_id = ?",
@article.id]
      flash[:notice] = "Comment was successfully added."
      respond_to do |format|
        format.html { redirect_to :action => "show", :controller => "article",
:id => @article.id }
        format.js
      end
    else
      render :action => "show", :controller => "article", :id => @article.id
    end
  end

  def delete
    @comment = Comment.find(params[:id])
    @article = Article.find(@comment.article_id)
    @comment.destroy
    redirect_to :action => "show", :controller => "article", :id => @article.id
  end

end

end

```

```

class AdminController < ApplicationController

  before_filter :access?, :except => [:login]

  def list
    @admins = Admin.find(:all)
  end

  def new
  end

  def create
    @admin = Admin.new(params[:admin])
    if @admin.save
      redirect_to :action => "list"
    else
      render :action => "new"
    end
  end

  def edit
    @admin = Admin.find(params[:id])
  end

  def update
    @admin = Admin.find(params[:id])
    if @admin.update_attributes(params[:admin])
      redirect_to :action => "list"
    else
      render :action => "edit", :id => session[:admin].id
    end
  end

  def delete
    @admin = Admin.find(params[:id])
    @admin.destroy
    if @admin == session[:admin]
      redirect_to :action => "logout"
    else
      redirect_to :action => "list"
    end
  end
end

```

```

def login
  Admin.find(:all).each do |admin|
    if admin.name.to_s == params[:admin][:name].to_s
      @admin = admin
    end
  end

  if ((!@admin.nil?) && (Admin.encrypted_password(params[:admin][:password],
@admin.salt) == @admin.hashed_password))
    session[:admin] = @admin
  else
    flash[:notice] = "Incorrect login/password"
  end

  redirect_to :action => "list", :controller => "article"
end

def logout
  session[:admin] = nil
  reset_session
  redirect_to :action => "list", :controller => "article"
end

end

class StatisticController < ApplicationController
  def index
    # reset_session
    # reset_session_data_test
    if session[:mode]== nil
      session[:mode] = -1
    end

    @mode= -1
    p"reset", params[:reset]
    p"mode", session[:mode]

    if params[:reset]== "Reset Page"
      print_session
      clear_session
      print_session
    end
  end
end

```

```

@categories = Category.find :all
if session[:mode].to_i == 0
  @categories = Category.find :all
else
  @ddrs = Resource.find :all, :conditions => "model like 'DD%'", :group =>
"model", :order => "model"
end

# initialization session,
# except session[:sql_parameter] session[:parameter_ids] if they exist
initializ_session

end

def hostupdate_remote

#   if not session[:host_ids].empty? && session[:host_ids] != nil
#     session[:host_ids].each do |hst|
#       if not session[:host_ids_temp].include?(hst.to_s)
#         session[:host_ids_temp] << hst
#       end
#     end
#   end

  session[:sql_host] = ""
  session[:host_ids] = Array.new
  if not session.nil?
    if not session[:category_ids].index(params[:id].to_i)
      session[:category_ids].push(params[:id].to_i)
    else
      @del_hosts = Resource.find :all, :conditions => ["category_id IN (?)",
params[:id].to_i], :order => "category_id, hostname"
      @del_hosts.each do |del_h|
        if session[:host_ids_temp].include?(del_h.hostname.to_s)
          ses-si-
on[:host_ids_temp].delete_at(session[:host_ids_temp].index(del_h.hostname.to_s).
to_i)
        end
      end
      ses-si-
on[:category_ids].delete_at(session[:category_ids].index(params[:id].to_i).to_i)
    end
  end
end

```

```

    @hosts = Resource.find :all, :conditions => ["category_id IN (?)", session[:category_ids].compact], :order => "category_id, hostname"

    render :partial => "hostlist"
  end
end

def generate_script
  print_session

  mode = session[:mode]#params[:mode]
  @sql_script = ""

  if params.has_key?("host_id")
    if not session[:host_ids_temp].index(Resource.find(params[:host_id]).hostname)
      session[:host_ids].push(Resource.find(params[:host_id]).hostname)
      session[:host_ids_temp].push(Resource.find(params[:host_id]).hostname)
    else
      session[:host_ids].delete_at(session[:host_ids].index(Resource.find(params[:host_id]).hostname).to_i)
      session[:host_ids_temp].delete_at(session[:host_ids_temp].index(Resource.find(params[:host_id]).hostname).to_i)
    end

    @l_hosts = ""
    session[:host_ids_temp].each do |host|
      @l_hosts += "'" + host + "', "
    end

    @l_hosts = @l_hosts.slice(0..@l_hosts.length - 3)
    session[:sql_host] = @l_hosts
  end

  if params.has_key?("ddr_id")
    if not session[:ddr_ids].index(Resource.find(params[:ddr_id]).model)
      session[:ddr_ids].push(Resource.find(params[:ddr_id]).model)
    else

```

```

        ses-si-
on[:ddr_ids].delete_at(session[:ddr_ids].index(Resource.find(params[:ddr_id]).mo
del).to_i)
    end
    @l_ddrs = ""
    session[:ddr_ids].each do |ddr|
        @l_ddrs += "'" + ddr + "', "
    end

    @l_ddrs = @l_ddrs.slice(0..@l_ddrs.length - 3)
    session[:sql_ddr] = @l_ddrs
end

if params.has_key?("parameter_id")
    if not session[:parameter_ids].index(params[:parameter_id])
        session[:parameter_ids].push(params[:parameter_id])
    else
        ses-si-
on[:parameter_ids].delete_at(session[:parameter_ids].index(params[:parameter_id]
).to_i)
    end

    @l_parameters = ""
    session[:parameter_ids].each do |parameter|
        @l_parameters += (session[:mode] == '0' ? "MAX(" : "AVG(") + parameter +
") AS " + parameter + ", "
    end

    @l_parameters = @l_parameters.slice(0..@l_parameters.length - 3)
    session[:sql_parameter] = @l_parameters

end

if session[:mode].to_i == 0
    if not session[:sql_parameter].nil? and not session[:sql_host].nil?
        if not session[:sql_parameter].empty? and not session[:sql_host].empty?
            @sql_script = "SELECT " + session[:sql_parameter] + " FROM
asup_summaries WHERE hostname IN (" + session[:sql_host] + ")"
        end
    end
end

if session[:mode].to_i == 1
    if not session[:sql_parameter].nil? and not session[:sql_ddr].nil?

```

```

        if not session[:sql_parameter].empty? and not session[:sql_ddr].empty?
            @sql_script = "SELECT " + session[:sql_parameter] + " FROM
asup_summaries WHERE model IN (" + session[:sql_ddr] + ")"
        end
    end
end

print_session
    render :partial => "sql_script"

end

def change_mode
#   reset_session_data_test

    session[:mode] = params[:mode_id]
#   @mode = 0
#   @mode = 1 if params[:mode_id] == "1"
    if not session[:parameter_ids].empty?
        @l_parameters = ""
        session[:parameter_ids].each do |parameter|
            @l_parameters += (session[:mode] == '0' ? "MAX(" : "AVG(") + parameter +
") AS " + parameter + ", "
        end

        @l_parameters = @l_parameters.slice(0..@l_parameters.length - 3)
        session[:sql_parameter] = @l_parameters
    end

    if session[:mode].to_i == 0
        @categories = Category.find :all
        p "sdaaaaaaa", @categories
        render :partial => "category_form"
    end
    if session[:mode].to_i == 1
        @ddrs = Resource.find :all, :conditions => "model like 'DD%'", :group =>
"model", :order => "model"
        render :partial => "ddr_form"
    end
end
end
end
end

```