Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

ПРОГРАМНА СИСТЕМА ДОСЛІДЖЕННЯ СЛАБОСТРУКТУРОВАНИХ ЗАДАЧ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ

Текстова частина до дипломної роботи
за спеціальністю „Інженерія програмного забезпечення” 121

Керівник дипломної роботи
канд. техн. н., доц. Франчук О.В.
(прізвище та ініціали)

_____
(підпис)
“09” червня 2021 р.

Виконав студент 2 курсу
магістерської програми
Тригуб Р.О.
(прізвище та ініціали)

“09” червня 2021 р.

Київ-2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри мережних технологій,
д. ф.-м.н., пр _____Г. І. Малашонок

«____»_____2020р

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на дипломну роботу
студенту ***Тригубу Роману Олександровичу*** факультету ***інформатики*** 2 курсу магістерської програми «Інженерія програмного забезпечення»

ТЕМА: «Програмна система дослідження слабоструктурованих задач багато-критеріальної оптимізації»

Зміст текстової частини до дипломної роботи:

Індивідуальне завдання
Календарний план
Реферат
Вступ
1 Метод аналізу ієрархій
2 Алгоритм побудови рекомендацій альтернативі, яка програла
3 Опис програмної системи
Висновки
Список використаних джерел

Дата видачі «___» _____ 2020 р. Керівник_____
(підпис)

Завдання отримав _____
(підпис)

Тема роботи:

«Програмна система дослідження слабоструктурованих задач багатокритеріальної оптимізації»

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проєкту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	20.10.2020	
2.	Огляд літератури за темою роботи. Метод аналізу ієрархій (MAI)	15.11.2020	
3.	Розробка інтерфейсної частини майбутньої програмної системи	1.12.2020	
4.	Реалізація авторського варіанту MAI Т.Сааті	30.12.2020	
5.	Тестування першого варіанту програмної системи	20.01.2021	
6.	Розробка алгоритму побудови рекомендацій альтернативі, яка програла	10.02.2021	
7.	Реалізація розробленого у п.5 алгоритму у програмній системі	1.03.2021	
8.	Виконання робіт по відлагодженню та тестуванню програмної системи	1.04.2021	
9.	Написання теоретичної частини дипломної роботи	25.04.2021	
10.	Аналіз отриманих результатів з науковим керівником	14.05.2021	
11.	Попередній захист курсової роботи	14.05.2021	
12.	Корегування роботи за результатами попереднього захисту	28.05.2021	
13.	Підготовка до публічного захисту. Розробка презентації	29.05.2021	
14.	Написання плану виступу та доповіді	2.06.2021	
15.	Остаточне оформлення дипломної роботи, програмного коду та слайдів презентації	09.06.2021	
16.	Захист дипломної роботи	16.06.2021	

Студент _____

Керівник _____

“_____” _____

РЕФЕРАТ

Обсяг роботи 58 сторінок, 17 ілюстрацій, 4 таблиці, 10 джерел посилань, 3 додатки.

АЛЬТЕРНАТИВИ, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, КРИТЕРІЇ, МЕТОД АНАЛІЗУ ІЄРАРХІЙ, РОЗРОБКА РЕКОМЕНДАЦІЙ, СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, СЛАБОСТРУКТУРОВАНІ ЗАДАЧІ.

Серед багатокритеріальних задач прийняття рішень, що особливо часто виникають на практиці, актуальними залишаються задачі вибору альтернатив. Математично такі задачі описуються набором альтернатив, для кожної з яких задаються значення певних показників (критеріїв). Розв'язком такої задачі вважається альтернатива, яка має найкращі (за сукупністю) значення критеріїв, які в загальному випадку відрізняються різною вагомістю.

Існуючі на сьогодні програмні продукти розв'язання наведеного класу задач обмежуються лише знаходженням найкращої альтернативи, тоді як запропонована програмна система крім вирішення цієї нетривіальної задачі ще дозволяє розробити для будь-якої з «програвших» альтернатив рекомендації (так би мовити «настанови до дій»), дотримання яких дозволить їй стати найкращою. Алгоритм, який генерує ці інструкції є основним науковим результатом роботи, а його реалізація у вигляді програмної системи – основним практичним результатом.

Зауважимо, що для обраної користувачем альтернативи програмна система згенерує в певному сенсі «інтелектуальний» список рекомендацій, виконання якого дозволить даній альтернативі перемогти. Під «інтелектуальністю» розуміється формування таких інструкцій для даної альтернативи, які б, з одного боку, потребували якомога менше зусиль (змін) альтернативи в порівнянні з її попереднім станом, та, з іншого, цих зусиль (змін) вистачало б для того, щоб дана альтернатива стала найкращою.

Розроблена програмна система має універсальний характер, може застосовуватися в різноманітних сферах людської діяльності для розв'язання складних задач багатокритеріальної оптимізації.

ЗМІСТ

ВСТУП

Серед багатокритеріальних задач прийняття рішень [1-5], що особливо часто виникають на практиці, актуальними залишаються задачі вибору альтернатив [6-8]. Математично такі задачі описуються набором альтернатив, для кожної з яких задаються значення певних показників (критеріїв). Розв'язком такої задачі вважається альтернатива, яка має найкращі (за сукупністю) значення критеріїв, які в загальному випадку відрізняються різною вагомістю.

Як правило, людина завжди намагається здійснити найкращий вибір. Але можливості людини щодо глибокого аналізу інформації не є безмежними. Це особливо відчувається сьогодні, в епоху інформаційного суспільства, коли, з одного боку, з кожним днем у нас все більше знань про навколишній світ, а з іншого, ми не встигаємо цю інформацію переосмислити, бо часто вимушені приймати рішення за умов часових обмежень. Який же вихід? На нашу думку в таких ситуаціях людині могли б допомогти «розумні» комп'ютерні програми із зручним інтерфейсом, які б відігравали роль своєрідних помічників. Очевидно, що з часом саме такі системи становитимуться все більш затребуваними, що свідчить про ***актуальність*** та перспективність даних досліджень.

Об'єкт дослідження – процеси прийняття оптимальних рішень для слабоструктурованих (слабо формалізованих) задач багатокритеріальної оптимізації.

Предмет дослідження – математичні моделі та методи розв'язання слабоструктурованих задач багатокритеріальної оптимізації.

Мета науково-дослідницької роботи – створити програмну систему підтримки процесу знаходження оптимального вибору в задачах багатокритеріальної оптимізації для особи чи колегіального органу, які приймають відповідальні рішення.

Завдання дослідження:

1. реалізувати у системі процес розв'язання задачі знаходження найкращої альтернативи за сукупністю критеріїв;
2. розробити і реалізувати у системі алгоритм, який для будь-якої «програвшої» альтернативи побудує рекомендації, дотримання яких гарантує їй перемогу.

Для досягнення мети в роботі застосовані наступні ***методи досліджень***:

1. Метод аналізу ієрархій – для формалізації слабоструктурованої задачі багатокритеріальної оптимізації та знаходження оптимального рішення;
2. методи математичного програмування – для побудови рекомендацій «програвшим» альтернативам, дотримання яких гарантує їм перемогу;
3. обчислювальний експеримент – для відлагодження системи і розробки дружнього інтерфейсу, орієнтованого на непрофесійних користувачів комп'ютерів.

Огляд систем підтримки прийняття рішень (в тому числі і для слабоструктурованих задач) можна знайти у роботах [9,10]. Існуючі на сьогодні програмні засоби розв'язання даного класу задач обмежуються лише знаходженням найкращої альтернативи, тоді як запропонована система, крім вирішення цієї задачі ще дозволяє розробити для будь-якої з «програвших» альтернатив інструкції («настанови до дій»), дотримання яких дозволить їй стати найкращою. Це і є основним результатом роботи.

Функціонально система складається з двох основних частин: перша методом аналізу ієрархій (MAI) знаходить найкращу альтернативу, друга за розробленим алгоритмом згенерує для будь-якої іншої альтернативи «настанови до дій».

Робота має нові як теоретичні, так і практичні результати.

Основним науковим результатом роботи є побудова алгоритму для розв'язання таких задач багатокритеріальної оптимізації, в яких альтернативи можуть змінювати (покращувати) свій стан. Вперше розроблено алгоритм, який дозволяє отримати для однієї з «програвших» альтернатив рекомендації, дотримання яких гарантує альтернативі перемогу.

Практичне значення роботи полягає у побудові завершеного програмного продукту, який може бути використаний особами, які приймають відповідальні рішення (в різних сферах людської діяльності).

Програмна система розроблена в інтегрованому середовищі Delphi 7 у відповідності з концепцією GUI (англ. Graphical user interface), тобто система характеризується дружнім інтерфейсом, орієнтованим на непрофесійних користувачів комп'ютера.

Апробація роботи та публікації з теми роботи.

Матеріалами роботи доповідались на наступних конференціях:

1. Міжнародна конференція, присвячена 90-річчю від дня народження академіка В.М.Глушкова «Сучасна інформатика: проблеми, досягнення та перспективи розвитку» (Тези доповідей). України, м.Київ, Інститут кібернетики ім. В.М.Глушкова НАН України, 12-13 вересня 2013 р.
2. VI Міжнародна конференція імені академіка І.І.Ляшка. Обчислювальна та прикладна математика. Матеріали конференції 5-6 вересня 2013 р., м.Київ. – Researching semistructured problems of multicriteria optimization.
3. 25th European Union Contest for Young Scientists EUCYS 2013. Prague. Researching semistructured problems of multicriteria optimization using the software system.
4. Міжнародна наукова конференція "Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних технологій" (присвячена пам'яті акад.І.І.Ляшка): 2-4 березня 2018 р., Рівне: 2018. – С.90-92.

За результатами роботи здійснено 5 публікацій:

1. Наукові записки НаУКМА. Том 151. Комп'ютерні науки. Researching semistructured problems of multicriteria optimization using the software system. Київ, 2013. – с.79-88.
2. Конкурс науково-технічних проектів учнів Intel-Техно Україна 2013. Тези робіт. Частина 3. Категорія "Комп'ютерні науки"/ Укладачі Дмитренко Д.А., Куссуль О.М. – К.: НТУУ «КПІ», 2013. – с.33-35.
3. 25th European Union Contest for Young Scientists EUCYS 2013. Prague. Researching semistructured problems of multicriteria optimization using the software system. p.76.
4. VI Міжнародна конференція імені академіка І.І.Ляшка. Обчислювальна та прикладна математика. Матеріали конференції 5-6 вересня 2013 р., м.Київ. – Researching semistructured problems of multicriteria optimization. – pp.44-45.
5. Міжнародна конференція, присвячена 90-річчю від дня народження академіка В.М.Глушкова «Сучасна інформатика: проблеми, досягнення та перспективи розвитку» (Тези доповідей). України, м.Київ, Інститут кібернетики ім. В.М.Глушкова НАН України, 12-13 вересня 2013 р. – с.308

.

РОЗДІЛ 1
МЕТОД АНАЛІЗУ ІЄРАРХІЙ

Метод аналізу ієрархій (MAI) був розроблений американським математиком Т. Сааті в 1980 р. під назвою «analytic hierarchy process» [8]. На сьогодні MAI є одним з найбільш відомих методів розв'язання слабоструктурованих багатокритеріальних задач прийняття рішень. Основними його етапами є:

1. структурування задачі вибору найкращої альтернативи у вигляді ієрархії. В мінімальному вигляді така ієрархія має складатися з трьох рівнів: мети задачі (найвищий рівень), через критерії, які враховуються при вирішенні задачі (середній рівень) до альтернатив, з яких знаходиться найкраща (найнижчий рівень). Цей етап MAI має назву «принцип ідентичності та декомпозиції».
2. Попарні порівняння між собою елементів одного рівня ієрархії з точки зору їх впливу на елемент ієрархії, розташований рівнем вище. Назва цього етапу – «принцип дискримінації та порівняльних суджень».
3. Отримання локальних пріоритетів елементів одного рівня ієрархії; вони характеризують відносний вплив кожного з елементів даного рівня ієрархії на елемент, розташований рівнем вище.
4. Отримання глобальних пріоритетів для всіх альтернатив; алгоритм обчислення враховує розраховані раніше локальні пріоритети. Фактично, це і є завершальним етапом розв'язання задачі, тобто альтернатива з найбільшим глобальним пріоритетом є найкращою. Етапи 3 та 4 мають об'єднану назву «принцип синтезу».

Такий підхід до розв'язку проблеми вибору виходить із природної здатності людини думати логічно і творчо, визначати події і встановлювати стосунки між ними. Відзначимо, що людині властиві дві характерні ознаки аналітичного мислення: перша – уміння спостерігати і аналізувати спостереження; друга –

здатність встановлювати стосунки між спостереженнями, оцінюючи рівень взаємозв'язків між відносинами, а потім синтезувати ці відносини в загальне сприйняття спостережуваного. Фактично МАІ відтворює цей процес, складовими якого є принцип ідентичності і декомпозиції, принцип дискримінації та порівняльних суджень і принцип синтезу. Розглянемо суть кожного з наведених етапів більш детальніше.

1.1. Принцип ідентичності і декомпозиції

Принцип ідентичності і декомпозиції передбачає структурування задачі у вигляді ієрархії або мережі, що є першим етапом застосування МАІ.

Існує кілька видів ієрархій. Найпростіші – домінантні ієрархії, які схожі на обернене дерево. Ієрархія вважається повною, якщо кожний елемент даного рівня є критерієм для всіх елементів нижче розташованого рівня, інакше ієрархія є неповною.

Процес структурування задачі учасниками може потребувати проведення додаткового аналізу, щоб мати впевненість, що критерії і альтернативи охоплюють всі наявні вподобання учасників обговорення і побудована ієрархія адекватно їх відображає. Не обов'язково, щоб усі учасники в процесі планування прийшли до абсолютної згоди по всім елементам ієрархії, бо в подальшому учасники процесу висловлюють своє бачення «значущості» (або вагомості) елементу ієрархії при проведенні попарних порівнянь. І якщо хтось з учасників обговорення вважає, що даний елемент не є суттєвим, він саме так його і оцінить. В цьому полягає певна «демократичність» МАІ. Проте обов’язковим є згода учасників процесу прийняття рішення по найвищому рівню ієрархії – меті (вершині) задачі, оскільки це визначить характер їх подальших суджень. Визначення мети може потребувати тривалих попередніх міркувань і переговорів.

1.2. Принцип дискримінації і порівняльних суджень

Після побудови ієрархії постає питання: як встановити пріоритети критеріїв і оцінити всі альтернативи за цими критеріями, обравши найкращу з них?

1.2.1. Попарні порівняння

Елементи ієрархії одного рівня порівнюються попарно відносно їхнього впливу на елемент ієрархії, розташований рівнем вище. Порівнюючи елементи один з одним, отримаємо квадратну матрицю вигляду:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \text{ де } a_{ij} = \frac{1}{a_{ji}} \ ,$$

тобто для матриці виконується властивість зворотної симетричності (індекси i та j позначають рядок та стовпчик відповідно.

Нехай $A_1, A_2, \dots, A_n$ – множина n елементів і $w_1, w_2, \dots, w_n$ – їх вагомості; у табл. 1.1 наведені їх попарні порівняння.

Таблиця 1.1

Попарні порівняння вагомостей елементів між собою

	$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\cdots$	$\boldsymbol{A_n}$
$\boldsymbol{A_1}$	1	w_1/w_2	$\cdots$	w_1/w_n
$\boldsymbol{A_2}$	w_2/w_1	1	$\cdots$	w_2/w_n
$\cdots$	$\cdots$	$\cdots$	1	$\cdots$
$\boldsymbol{A_n}$	w_n/w_1	w_n/w_2	$\cdots$	1

На основній діагоналі таких матриць завжди будуть розміщені 1, бо в цих клітинках порівнюється вагомість одного і того самого елементу. У випадку, коли вагомості $w_1, w_2, \dots, w_n$ невідомі, попарні порівняння елементів проводяться на основі суб'єктивних суджень, які чисельно оцінюються за рекомендованою шкалою (опис шкали наведений далі в роботі).

Схожі матриці повинні бути побудовані і на всіх інших рівнях ієрархії (тобто мають бути проведені попарні порівняння елементів певного рівня

відносно їх впливу на елемент, розташований рівнем вище). Наприклад, коли ієрархія складається з трьох рівнів (рівень мети, рівень критеріїв та рівень альтернатив), n кількість критеріїв, m – кількість альтернатив, то ми матимемо $(n+1)$ квадратну матрицю попарних порівнянь: 1 матрицю розмірності $n \times n$, в якій попарно порівнюються вагомості критеріїв з точки зору їх впливу на досягнення мети задачі, а також ще n матриць розмірності $m \times m$, в кожній з яких попарно порівнюються між собою m альтернатив по відношенню до кожного з n критеріїв.

1.2.2. Рекомендована шкала відносної вагомості

Щоб мати можливість чисельно виразити суб'єктивні попарні порівняння, необхідна певна шкала, в межах якої ці порівняння будуть здійснюватися. У табл. 1.2 наведена шкала, яку використовує MAI.

Таблиця 1.2

Рекомендована шкала порівняння

Показник відносної вагомості	**Визначення**
1	Рівна вагомість
3	Помірна перевага
5	Суттєва перевага
7	Сильна перевага
9	Абсолютна перевага
2, 4, 6, 8	Проміжні значення
Зворотні величини наведених чисел	Елемент поступається аналогічним чином

Доведено, що ця шкала є коректною і водночас достатньою для розв'язання багатьох практичних задач багатокритеріальної оптимізації.

При заповненні матриць попарних порівнянь за наведеною шкалою треба дотримуватись певних правил, а саме:

1. для кожної клітинки матриці порівнювати вагомість лівого елементу з вагомістю елементу, розташованому зверху: якщо перша вагомість більша за другу, ставимо ціле число зі шкали, інакше – зворотне;

2. в діагональні клітинки матриць заносяться одиниці;
3. в симетричні клітинки матриць заносяться зворотні величини; таким чином, в загальному випадку для заповнення матриці розмірності $n \times n$ достатньо здійснити $n(n-1)/2$ порівнянь.
4. при проведенні попарних порівнянь альтернатив між собою стосовно критерію ставлять питання яка з альтернатив більш бажана, а при попарних порівняннях критеріїв між собою стосовно мети задачі, – який з критеріїв більш вагомий.

1.3. Синтез пріоритетів

1.3.1. Синтез: локальні пріоритети

З групи матриць попарних порівнянь елементів одного рівня ієрархії ми формуємо набір локальних пріоритетів, які виражають відносний вплив цих елементів на елемент ієрархії, розташований рівнем вище.

На практиці для наближеного обчислення локальних пріоритетів досить часто використовують геометричне середнє, коли треба перемножити елементи кожного рядка матриці і взяти від добутку корінь n-ї степені (де n – кількість елементів рядка). Отриманий таким чином стовпчик чисел треба нормалізувати, розділивши кожне число на суму всіх чисел. Наприклад, для наведеної вище табл. 1.1 компоненти вектору локальних пріоритетів L_i можна отримати наступним чином:

$$L_i = \frac{\sqrt[n]{\frac{w_i}{w_1} \cdot \frac{w_i}{w_2} \cdot \ldots \cdot \frac{w_i}{w_n}}}{\sum_{j=1}^{n} \sqrt[n]{\frac{w_j}{w_1} \cdot \frac{w_j}{w_2} \cdot \ldots \cdot \frac{w_j}{w_n}}}, \quad i = \overline{1,n}.$$

Як правило, такі обчислення починають з другого рівня ієрархії (рівень критеріїв), поступово продовжують для всіх наступних рівнів і завершують побудовою локальних пріоритетів для найнижчого рівня (рівень альтернатив).

1.3.2. Узгодженість локальних пріоритетів

Дуже важливим параметром для кожної з матриць задачі є величина узгодженості (ВУ), яка надає інформацію про степінь порушення як порядкової (транзитивної), так і чисельної (кардинальної, $a_{ij} \cdot a_{jk} = a_{ik}$) узгодженості. В загальному випадку ми маємо не узгоджені матриці. Тому нам бажано мати критерій оцінки степені узгодженості матриць при розв'язанні практичної задачі. Саме таку роль і відіграє параметр ВУ.

Ідеальної узгодженості досить важко досягти на практиці. Але цього і не потрібно. Достатньо контролювати знаходження параметру ВУ в певних межах, і коли значення ВУ виходить за них, то особі, яка приймає рішення (ОПР) потрібно здійснити корекцію даних даної матриці.

Наведемо алгоритм наближеного обчислення величини узгодженості ВУ:

1. порахувати суму кожного стовпчика матриці;
2. суму першого стовпчика $\sum_{j=1}^{n} a_{1j}$ помножити на першу компоненту вектору пріоритетів L_1, суму другого стовпчика $\sum_{j=1}^{n} a_{2j}$ – на другу компоненту вектору пріоритетів L_2 і т.д.;
3. порахувати суму отриманих чисел (позначимо її як λ_{max}), тобто

$$\lambda_{max} = \sum_{i=1}^{n} \left(L_i \cdot \sum_{j=1}^{n} a_{ij} \right).$$

4. порахувати індекс узгодженості ІУ=$(\lambda_{max} - n)/(n-1)$, де n – порядок матриці; (для обернено симетричної матриці $\lambda_{max} \geq n$);
5. обчислити параметр ВУ=ІУ/ВВУ, де ВВУ – величина випадкової узгодженості, яка вийшла б при випадковому виборі порівняльних суджень зі шкали $\left\{\frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \dots, \frac{1}{2}, 1,2, \dots, 9\right\}$ за умови оберненої симетричності матриці. Значення ВВУ для матриць різного порядку становлять:

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

Розрахований параметр ВУ не повинен перевищувати $10 \div 20\%$. Для поліпшення узгодженості можна рекомендувати пошук додаткової інформації і коригування даних, введених під час проведення попарних порівнянь елементів певного рівня ієрархії.

1.3.3. Синтез: глобальні пріоритети

На заключному етапі МАІ здійснюється синтез (лінійна згортка) локальних пріоритетів на ієрархії, в результаті обчислюються пріоритети альтернатив відносно мети задачі (глобальні пріоритети). Найкращою вважається альтернатива, яка має максимальне значення глобального пріоритету. Наведемо алгоритм побудови глобальних пріоритетів:

1. пріоритети синтезуються, починаючи з другого і завершуючи найнижчим рівнем ієрархії;
2. локальні пріоритети перемножуються на пріоритет відповідного критерію на вище розташованому рівні і підсумовуються по кожному елементу відповідно до критеріїв, на які впливає цей елемент (на другому рівні ієрархії кожний елемент множиться на одиницю, тобто на вагу мети найвищого рівня ієрархії).

Це дає складовий, або глобальний, пріоритет того елемента, який потім використовується для оцінки вагомості локальних пріоритетів елементів, які порівнюються по відношенню до нього як до критерію і розташованих рівнем нижче. Процедура триває до найнижчого рівня. Наприклад, для ієрархії, яка складається з трьох рівнів (мета, критерії, альтернативи) глобальні пріоритети обчислюються за формулою:

$$G_k = \sum_{i=1}^{n} (L_i \cdot L_{ki}), \;\; k = \overline{1,m} \;,$$

де G_k – глобальний пріоритет k-ї альтернативи; L_i – локальний пріоритет i-го критерію відносно мети задачі, L_{ki} – локальний пріоритет k-ї альтернативи відносно i-го критерію. Розраховані значення глобальних пріоритетів і будуть шуканими розв'язками задачі, а саме, – найкращою буде та альтернатива, яка має максимальне значення G_k.

РОЗДІЛ 2
АЛГОРИТМ ПОБУДОВИ РЕКОМЕНДАЦІЙ АЛЬТЕРНАТИВІ, ЯКА ПРОГРАЛА

Для задач багатокритеріальної оптимізації в яких альтернативи можуть змінювати свій стан програмна система розробляє рекомендації, як бажаній альтернативі стати найкращою. Нехай n – кількість критеріїв, m – кількість альтернатив. $A = \{A_1, A_2, \dots, A_m\}$ – множина альтернатив, A_{best} – найкраща на даний час альтернатива, $A^* \in A \setminus A_{best}$ – альтернатива, для якої розробляються рекомендації.

Пошук рекомендацій для альтернативи A^* розпочинається знаходженням певної середньої альтернативи (по кожному з критеріїв вона може бути інша), відносно якої аналізуються зміни стану альтернативи A^*. Вибір такої середньої альтернативи здійснюється на основі локальних пріоритетів на рівні критеріїв (тобто в межах одного критерію буде обрана альтернатива, яка не є найкращою, не є найгіршою, а її локальний пріоритет по цьому критерію знаходиться приблизно посередині). Такий підхід дозволяє провести повноцінний аналіз можливих змін стану альтернативи A^* на відміну від випадку, коли для порівняння обирається будь-яка інша альтернатива. Слід зауважити, що для порівняння не доцільно обирати альтернативу A_{best} , тому що це може призвести до невиправдано завищених рекомендацій для альтернативи A^*, тоді як порівняння з середньою альтернативою дозволить отримати такі рекомендації, які потребуватимуть від альтернативи A^* найменших змін. Для проведення повноцінного аналізу доцільно мати можливість робити відносно незначні зміни даного стану альтернативи A^*, які разом з тим могли б призвести до досягнення мети (щоб альтернатива A^* стала найкращою). Саме це і забезпечує середня альтернатива.

Альтернатива A^* стане найкращою, якщо її глобальний пріоритет стане максимальним. Для цього необхідно покращити значення локальних пріоритетів A^* по певним критеріям, а це вимагає побудови нових локальних пріоритетів для нового (зміненого) стану A^*.

Для розрахунку локальних пріоритетів для нових станів альтернативи A^* потрібно створити матриці попарних порівнянь усіх можливих покращень A^* відносно середньої альтернативи по всім критеріям. Розглянемо цей підхід на прикладі деякого критерію k. Нехай за критерієм k середньою альтернативою є C_k. Порівнюємо ці дві альтернативи A^* та C_k всіма можливими способами для кожного значення зі шкали $\{1/9, \dots, 9\}$ і заповнюємо матрицю попарних порівнянь нового стану альтернативи A^* (див. клітинки табл. 2.1 червоного кольору); при цьому значення попарних порівнянь A^* з іншими альтернативами оновлюються (див. клітинки табл. 2.1 зеленого кольору, в яких $w_{A^*_{new}}$ – нове значення вагомості альтернативи A^* після покращення).

Таблиця 2.1

Матриця попарних порівнянь вагомостей альтернатив після зміни стану альтернативи A^*

	$\mathbf{A_1}$	$\mathbf{A^*}$	$\mathbf{C_k}$	…	$\mathbf{A_m}$
$\boldsymbol{A_1}$	1	$w_1/w_{A^*_{new}}$		…	w_1/w_m
$\boldsymbol{A^*}$	$w_{A^*_{new}}/w_1$	1	$\{1/9, \dots, 9\}$	…	$w_{A^*_{new}}/w_m$
$\boldsymbol{C_k}$		$\{9, \dots, 1/9\}$	1		
…	…	…		1	…
$\boldsymbol{A_m}$	w_m/w_1	$w_m/w_{A^*_{new}}$		…	1

Для отриманої матриці попарних порівнянь розраховуємо різницю локальних пріоритетів нового стану альтернативи A^* та найкращої альтернативи A_{best}. Нехай $L_{i,k}(A)$ – новий локальний пріоритет за критерієм k, $i \in \{9, \dots, 1/9\}$ для альтернативи A; $L_k(A)$ – старий (початковий) локальний пріоритет за критерієм k; w_k – вагомість k-го критерію. Тоді внеском до глобального пріоритету за критерієм k при умові, що альтернатива A^* покращиться відносно середньої альтернативи C_k в межах шкали $i \in (1/9 \dots 9)$ буде величина:

$$\Delta_{i,k} = ((L_{i,k}(A^*) - L_{i,k}(A_{best})) - (L_k(A^*) - L_k(A_{best})) * \mathrm{w}_k$$

Розраховані значення $\Delta_{i,k}$ заносимо до табл. 2.2. Задача знаходження рекомендацій для альтернативи A^* зводиться до наступної задачі оптимізації:

$$h(\Delta) = \sum_{k=1}^{n} \Delta_{i_k,k} - (G(A_{best}) - G(A^*)) \rightarrow \min, \qquad (2.1)$$

де величини $h(\Delta) > 0, i_k \in \{1/9, \dots, 9\}$.

Таким чином необхідно знайти такі індекси i_k по кожному критерію ($k = \overline{1,n}$), які мінімізують $h(\Delta)$ – сумарний показник зусиль для альтернативи A^*. Проілюструємо сказане на табл. 2.2, яка показує степінь можливого наближення за глобальним пріоритетом до найкращої альтернативи A_{best}.

Таблиця 2.2

Напрям покращення вагомостей альтернативи A^* відносно середньої (за кожним з критеріїв) альтернативи

Шкала	**Критерій K_1**	**Критерій K_2**	**Критерій K_3**	**…**	**Критерій K_{n-1}**	**Критерій K_n**
9	$\Delta_{9,1}$	$\Delta_{9,2}$	$\Delta_{9,3}$		$\Delta_{9,n-1}$	$\Delta_{9,n}$
8		↑ 0		↑ 0		
…	↑ 0		↑ 0			↑ 0
…					↑ 0	
1/8						
1/9	$\Delta_{1/9,1}$	$\Delta_{1/9,2}$	$\Delta_{1/9,3}$		$\Delta_{1/9,n-1}$	$\Delta_{1/9,n}$

Тут 0 – це ті клітинки, які відповідають початковому (старому) стану альтернативи A^* відносно середньої альтернативи по K_i-му критерію; ці значення не дадуть ніякого покращення до нового глобального пріоритету і тому в них поставлений 0, тобто $\Delta_{k_{start},k} = 0$; (k_{start} – початкове значення попарного порівняння альтернативи A^* відносно середньої за критерієм k, тобто

$k_{start} \in \{1/9,1/8,\dots,8,9\}$). Побудова рекомендацій для кожного критерію (кожного стовпчику) здійснюється в напрямку стрілок.

Крім сумарного показника зусиль, який розраховується за формулою (2.1), можна обчислити кількість кроків, які треба зробити альтернативі A^* щоб стати найкращою (для кожної з рекомендацій). Тут під кроком розуміється перехід на 1 позицію в межах шкали $\{1/9,\dots,9\}$ (наприклад, одним кроком є зміна стану альтернативи з $1/8$ до $1/7$, або з 2 до 3). Позначимо через num функцію, яка множині елементів шкали MAI ставить у взаємно-однозначну відповідність множину натуральних чисел від 1 до 17, тобто $num: \{1/9,\dots,9\} \to \{1,\dots,17\}$. Тоді сумарна кількість кроків, потрібних для досягнення мети складатиме:

$$\sum_{k=1}^{n} (num(i_k) - num(k_{start})), \qquad h(\Delta) > 0 \tag{2.2}$$

$$i_k \in \{1/9,\dots,9\}.$$

Але більш інформативними та корисними будуть зважені зусилля за вагою критерію, які можна обчислити за наступною формулою:

$$\sum_{k=1}^{n} (w_k * (num(i_k) - num(k_{start}))), \qquad h(\Delta) > 0 \tag{2.3}$$

Ця формула враховує вагомість кожного з критеріїв для досягнення мети.

Наведемо покрокове виконання алгоритму в спрощеному вигляді:

0) k=1; $j_1 = j_{1(start)}$, $sum = 0$.

1) Якщо $k = 0$, тоді завершити роботу.

2) Якщо $j_k \geq 9$, перейти до п.3, інакше

$sum = sum - \Delta_{j_k,k}$, $j_k = j_k + 1$, $sum = sum + \Delta_{j_k,k}$,

якщо $sum - (G(A_{best}) - G(A^*)) > 0$ – зафіксувати j-ту рекомендацію, перейти до п.3,

інакше якщо $k < n$ тоді $k = k + 1, j_k = j_{k(start)}$, перейти до п.1.

3) $sum = sum - \Delta_{j_k,k}$, $k = k - 1$, перейти до п.1.

РОЗДІЛ 3
ОПИС ПРОГРАМНОЇ СИСТЕМИ НА ПРИКЛАДІ ЗАДАЧІ ВИБОРУ НАЙКРАЩОГО ФУТБОЛІСТА ЄВРОПИ

Постановка задачі. На звання найкращого футболіста Європи 2012 р. претендують декілька кандидатів. Потрібно, враховуючи певну сукупність критеріїв, здійснити оптимальний вибір.

Розв'язання задачі за допомогою програмної системи. Після запуску системи розв'язання нової задачі починається з команди «Створити новий проект…», яка знаходиться в пункті меню «Файл». Команда виведе на екран діалогове вікно (рис. 3.1), в якому треба ввести назву задачі, кількість критеріїв та альтернатив; у полі «Коментар» можна ввести більш розгорнутий опис задачі:

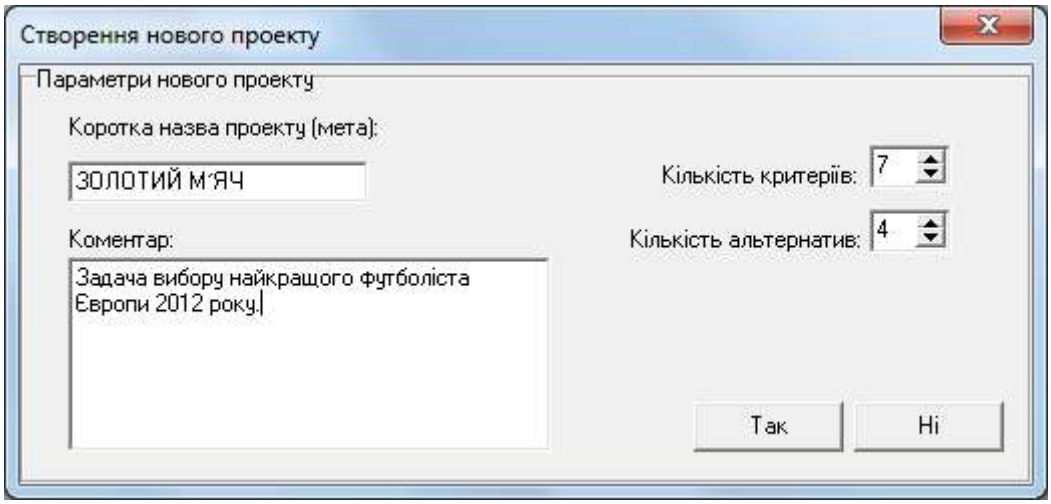


Рис. 3.1 Створення нового проекту

Припустимо, що потрібно зробити вибір з 4 наступних альтернатив:

{Роналду, Мессі, Ібрагімовіч, Інієста}, при цьому врахувати 7 критеріїв:

{Техніка, Атлетизм, Пас, Удар, Швидкість, Диспетчер, Витривалість}.

Після натискання кнопки «Так» з'явиться ієрархічне представлення задачі з відповідною кількістю критеріїв та альтернатив (рис. 3.2). В цьому вікні у користувача є можливість з контекстного меню елементу ієрархії додати або вилучити критерій або альтернативу, редагувати його назву, робити коментарі, завантажувати фотографії. У системі можна задавати до 9 критеріїв та до 9 альтернатив. Кількість рівнів ієрархії становить 3 (мета – найвищий рівень, крите-

рії – 2 рівень, альтернативи – найнижчий рівень). Для більшості практичних задач багатокритеріальної оптимізації ці обмеження не є суттєвими.

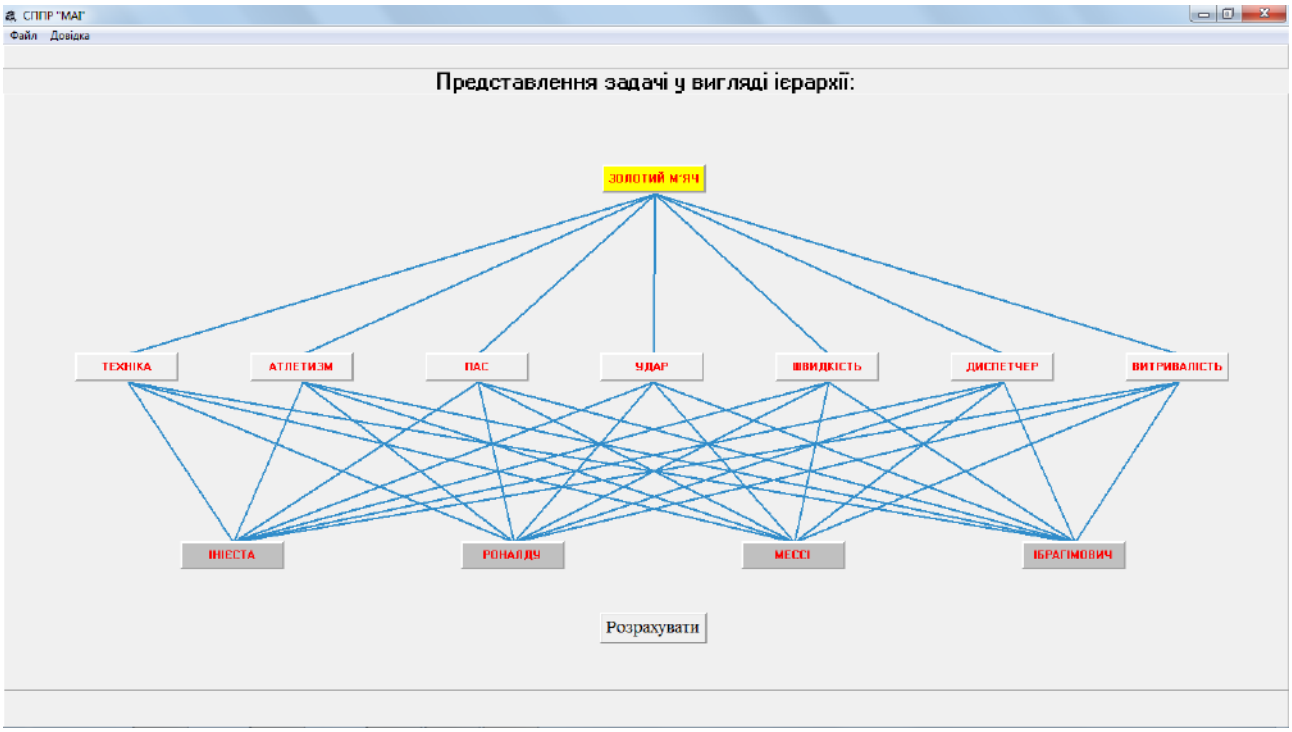


Рис. 3.2 Представлення задачі у вигляді ієрархії

На рис. 3.3 показано діалогове вікно введення даних для альтернатив.

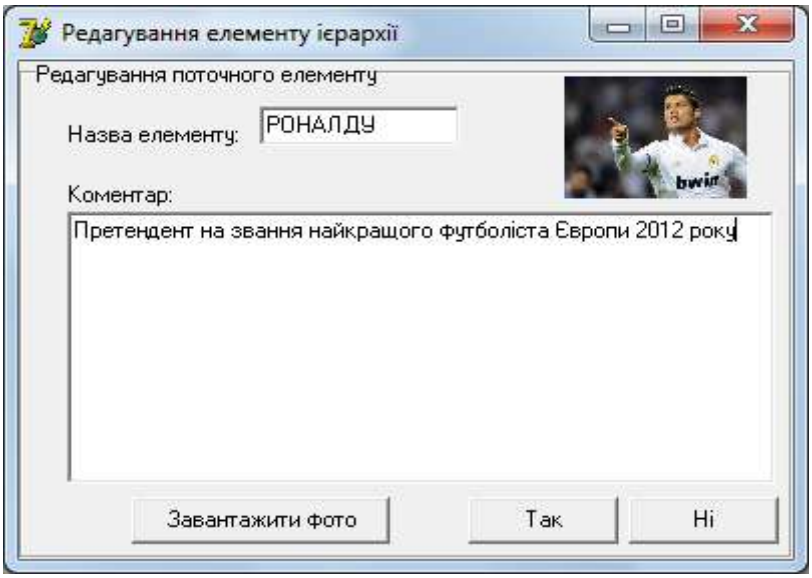


Рис. 3.3 Редагування елементу ієрархії

Клацання лівою кнопкою миші на найвищому елементі ієрархії виведе діалогове вікно для введення матриці попарних порівнянь критеріїв. Заповнюємо її за допомогою лінгвістичної шкали, розташованої в нижній частині вікна. Правий стовпчик матриці виводить розраховані локальні пріоритети, а в нижній

частині можна побачити величину узгодженості даної матриці. У випадку, коли вона перевищує 20% система повідомляє про це червоним кольором (рис. 3.4).

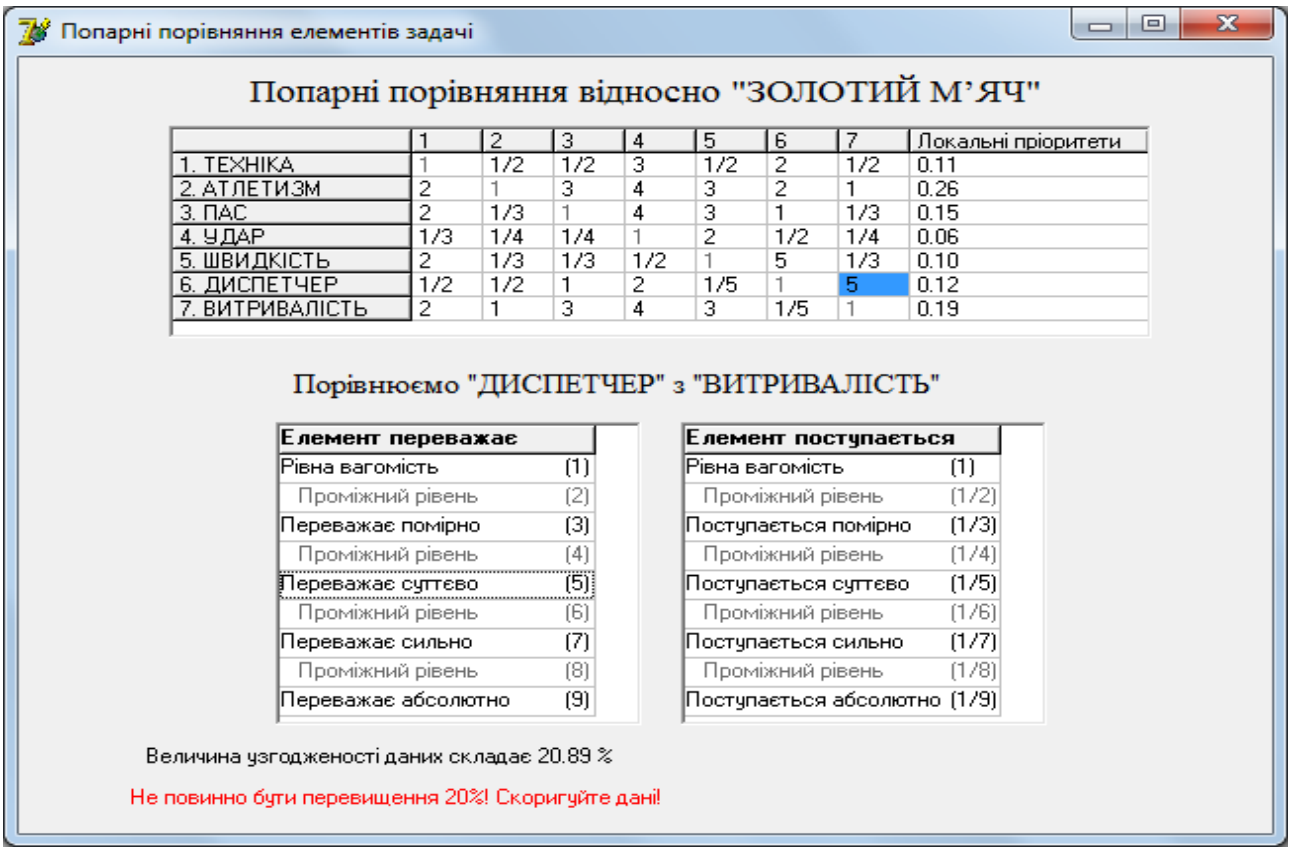


Рис. 3.4 Попарні порівняння критеріїв

Клацання лівою кнопкою на будь-якому критерію виведе на екран схоже вікно для введення попарних порівнянь альтернатив між собою (рис. 3.5).

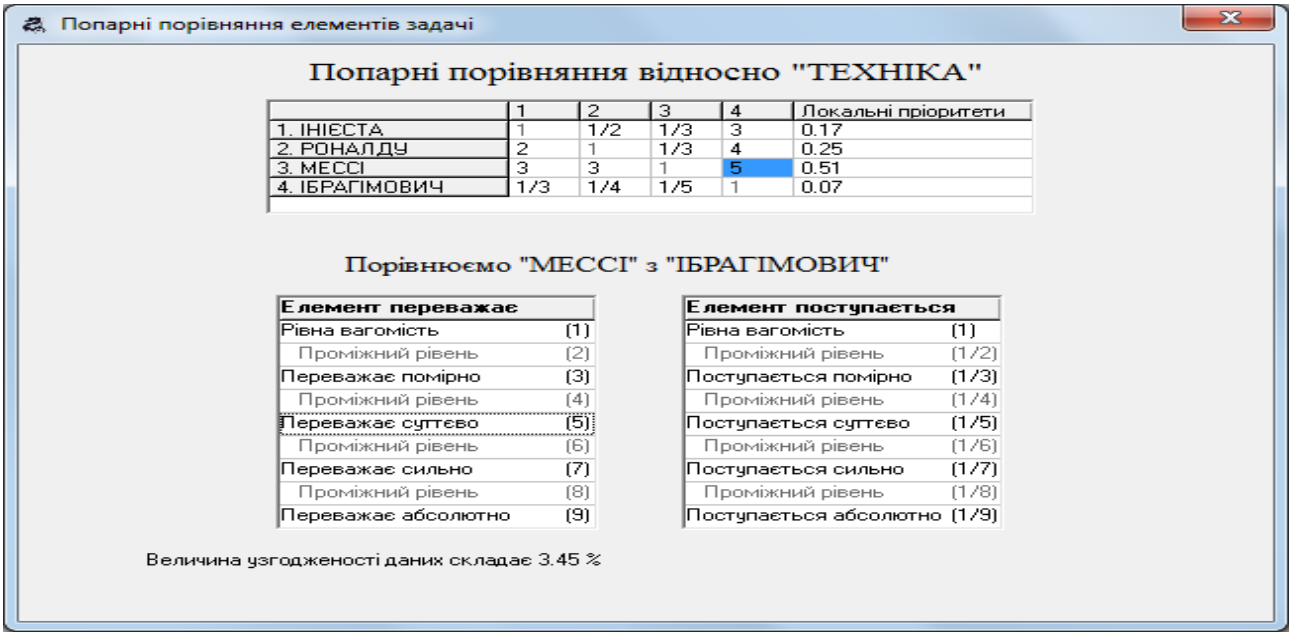


Рис. 3.5 Попарні порівняння альтернатив

Після введення даних користувач має натиснути кнопку «Розрахувати». На рис. 3.6 показаний результат.

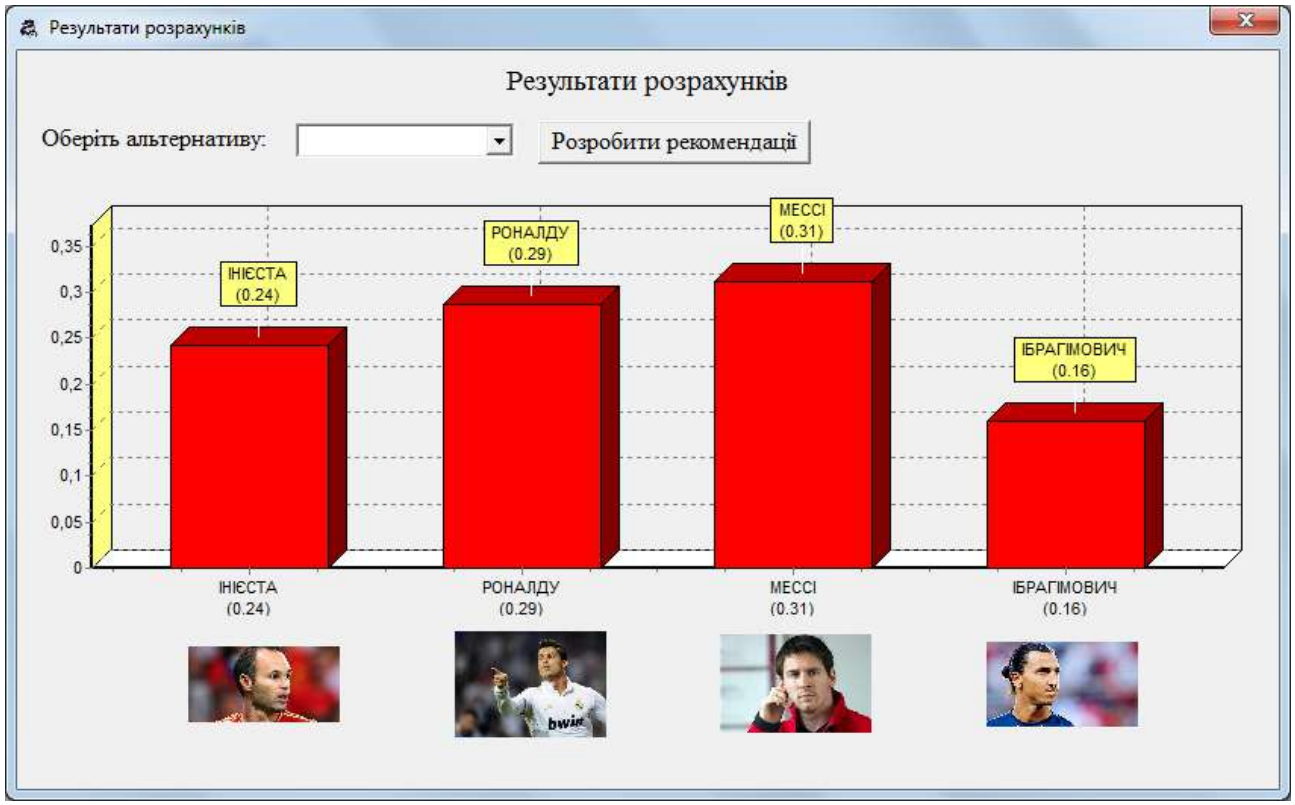


Рис. 3.6 Результати розрахунків

Переміг Мессі. Розробимо рекомендації для перемоги, наприклад, Інієсти. Оберемо цю альтернативу з випадаючого списку угорі вікна і натиснемо кнопку «Розробити рекомендації». На рис. 3.7 виведений результат.

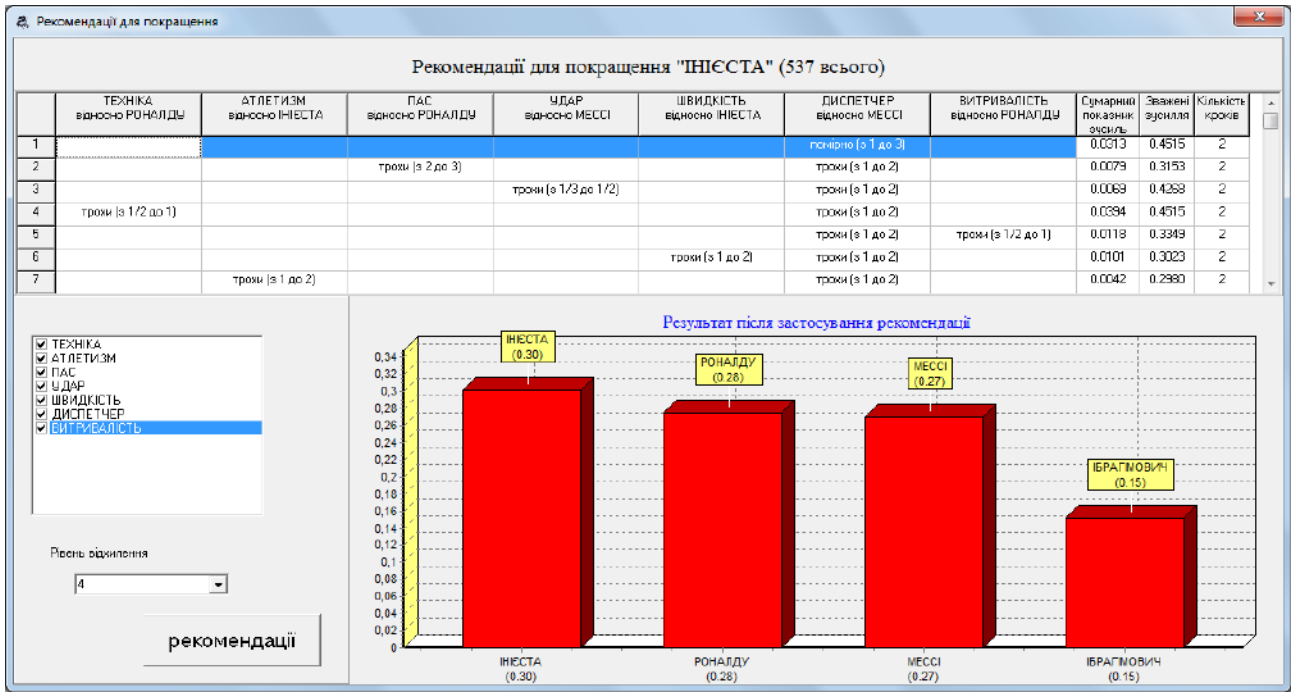


Рис. 3.7 Результати розрахунків для програвшої альтернативи після покращення

Результатом розрахунків є 537 рекомендацій (1 рядок – це 1 рекомендація) для альтернативи «Інієста», виведених у верхній частині вікна. Будь-яка з них гарантує альтернативі «Інієста» перемогу.

Стовпчиками кожної рекомендації є середні альтернативи, відносно яких треба покращувати свій стан. В дужках виводяться числа від і до, які показують який був і яким має стати стан альтернативи. Так, на наведеному рис. 3.7 виведена рекомендація №1, відповідно до якої альтернативі «Інієста» треба помірно покращити відносно альтернативи «Мессі» диспетчерські якості (з 1 до 3).

У наведеному розрахунку для альтернативи «Інієста» рекомендації побудовані на аналізі всіх 7 критеріїв. Але система дозволяє (дивись ліву нижню частину наведеного вікна) користувачу обирати не всі, а лише певні критерії для побудови рекомендацій. Доцільно обирати ті критерії, покращити стан за якими для даної альтернативи найпростіше. Додатково для кожного обраного критерію система надає можливість вказати рівень відхилення, на яку за даним критерієм альтернатива може покращити свій стан в порівнянні з попереднім.

Якщо рекомендацій дуже багато, система дозволяє їх впорядкувати за одним з трьох параметрів (див. останні 3 стовпчики на наведеному вище рис. 3.7), а саме: за сумарним показником зусиль, за зваженими зусиллями, або за кількістю кроків. Це відповідає формулам (2.1), (2.3) та (2.2) відповідно. Для впорядкування достатньо клацнути на бажаному стовпчику і тоді першими будуть виведені рекомендації з найменшими значеннями цього параметру.

Безумовно, остаточне рішення яких саме рекомендацій дотримуватися прийматиметься людиною. Але запропонована система надає в руки людині дуже ефективний інструмент не лише аналізу стану альтернативи на даний час, а і дозволяє оцінити перспективи її покращення в майбутньому, формуючи «настанови до дій». Дуже важливо, що ці рекомендації є максимально конкретними, принаймні настільки, наскільки це взагалі можливо при розв’язанні слабоструктурованих задач багатокритеріальної оптимізації.

ВИСНОВКИ

В результаті виконання науково-дослідницької роботи «Програмна система дослідження слабоструктурованих задач багатокритеріальної оптимізації» були опановані базові теоретичні та практичні засади теорії прийняття рішень, досліджені методи розв'язання задач багатокритеріальної оптимізації (задачі вибору або ранжування альтернатив), їх програмна реалізація.

В роботі розглянуто такий клас задач багатокритеріальної оптимізації, в яких альтернативи дозволяють змінювати (покращувати) свій стан. Існуючі програмні продукти, які вирішують такого класу задачі, обмежуються лише знаходженням та констатацією факту яка саме альтернатива за даних умов є найкращою, тоді як запропонована програмна система не лише розв'язує цю класичну задачу вибору, але і дозволяє для будь-якої з альтернатив, які програли, генерувати рекомендації, дотримання яких гарантувало б для даної альтернативи перемогу. Тобто у системі породжуються певні «настанови до дій», які вказують що саме і на яку величину треба змінити, щоб стати найкращою альтернативою. В цьому полягає ***наукова новизна*** роботи.

Побудована програмна система є ***основним практичним результатом*** досліджень. Підкреслимо, що для обраної альтернативи програмна система згенерує «інтелектуальний» список рекомендацій, виконання якого дозволить даній альтернативі перемогти. Під «інтелектуальністю» розуміється формування таких інструкцій для даної альтернативи, які б, з одного боку, потребували якомога менше зусиль (змін) альтернативи в порівнянні з її попереднім станом, та, з іншого, цих зусиль (змін) вистачало б для того, щоб дана альтернатива стала найкращою.

Основним теоретичним результатом досліджень є побудова відповідного алгоритму, за яким для одної з альтернатив, яка програла, формується «інтелектуальний» список рекомендацій, дотримання яких гарантує для даної альтернативи перемогу.

Робота має завершений вигляд, містить нові як теоретичні, так і практичні результати та носить очевидний прикладний характер.

Запропонована програмна система має універсальний характер, може застосовуватися в різних сферах людської діяльності для розв'язання складних задач багатокритеріальної оптимізації не лише для знаходження найкращої альтернативи, а і для отримання максимально конкретних рекомендацій для бажаної альтернативи, що саме і на скільки їй треба покращити, щоб перемогти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гафт М.Г. Принятие решений при многих критериях: брошюра / М.Г.Гафт – М.: Знание, 1979. – 64с. – ISBN 5-1346227-А : 0.00.
2. Гафт М.Г. О построении решающих правил в задачах принятия решений / М.Г.Гафт, В.В.Подиновский // Автоматика и телемеханика. – 1981. – №6. – С.128-138.
3. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных Странах: учебник [Текст] / О.И.Ларичев – М.: Логос, 2000. – 296с. – ISBN 5-88439-046-7.
4. Ногин В.Д. Принятие решений при многих критериях: Учебно-методическое пособие / В.Д.Ногин – СПб.: ЮТАС, 2007. – 104 с. – ISBN 978-5-91185-018-4.
5. Подиновский В.В. Многокритериальные задачи с упорядоченными по важности критериями // Автоматика и телемеханика. – 1976. – №11. – С.118-127.
6. Саати Т. Принятие решений: Метод анализа иерархий / Т.Саати – М.: Радио и связь, 1993. – 314 с. – ISBN 5-256-00443-3.
7. Саати Т. Л. Принятие решений при зависимостях и обратных связях: Аналитические сети / Т.Л.Саати – М.: Издательство ЛКИ, 2008. – 360 с. – ISBN 978-5-397-01622-3.
8. Saaty T.L. The analytic hierarchy process. – N.-Y.: McGraw Hill, 1980. – 288p. – ISBN 0-07-054371-2.
9. Ларичев О. И. Системы поддержки принятия решений. Современное состояние и перспективы их развития / О.И.Ларичев, А.В.Петровский // Итоги науки и техники. Сер. Техническая кибернетика. – Т.21. М.: ВИНИТИ, 1987. – С.131-164.
10. Сараев А. Д. Системный анализ и современные информационные технологии / А.Д.Сараев, О.А.Щербина // Труды Крымской Академии наук. – Симферополь: СОНАТ, 2006. – С. 47-59.

ДОДАТОК А
ЗАСТОСУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ДО ЗАДАЧІ ВИБОРУ НАЙКРАЩОГО АВТОМОБІЛЮ

Постановка задачі. З вказаної множини автомобілів обрати найкращий за сукупністю заданих критеріїв. Врахувати наступні 6 критеріїв: {«Вартість», «Надійність», «Дизайн», «Оснащення», «Комфорт», «Економічність»}. Дослідити наступні 4 альтернативи: {«Honda Civic», «Hyindai i30», «Peugeot 508», «VW Golf»}.

Розв'язання задачі за допомогою комп'ютерної системи. Спочатку вказуємо кількість критеріїв, альтернатив, назву та коментар задачі (рис.1).

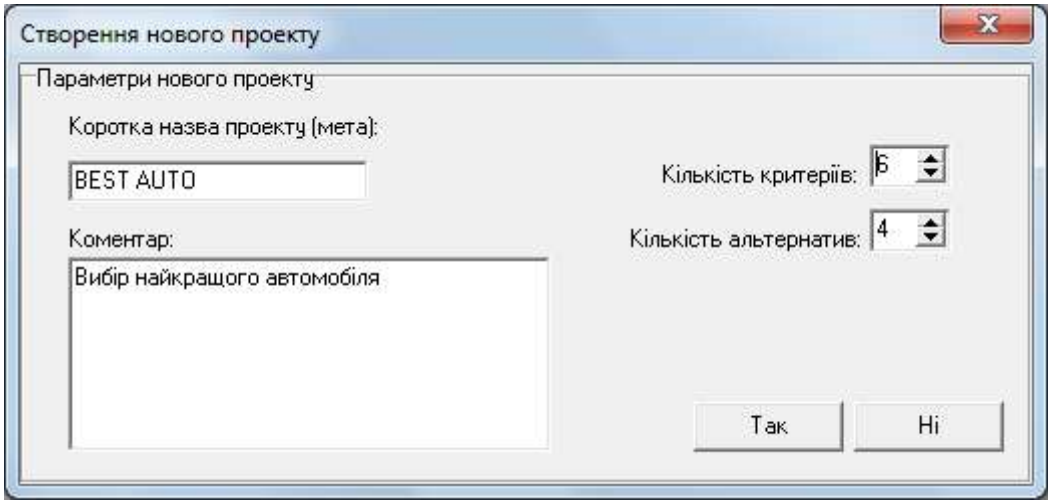


Рис.1 Створення нового проєкту

Далі будуємо ієрархію і вводимо назви та коментарі її елементів (рис.2).

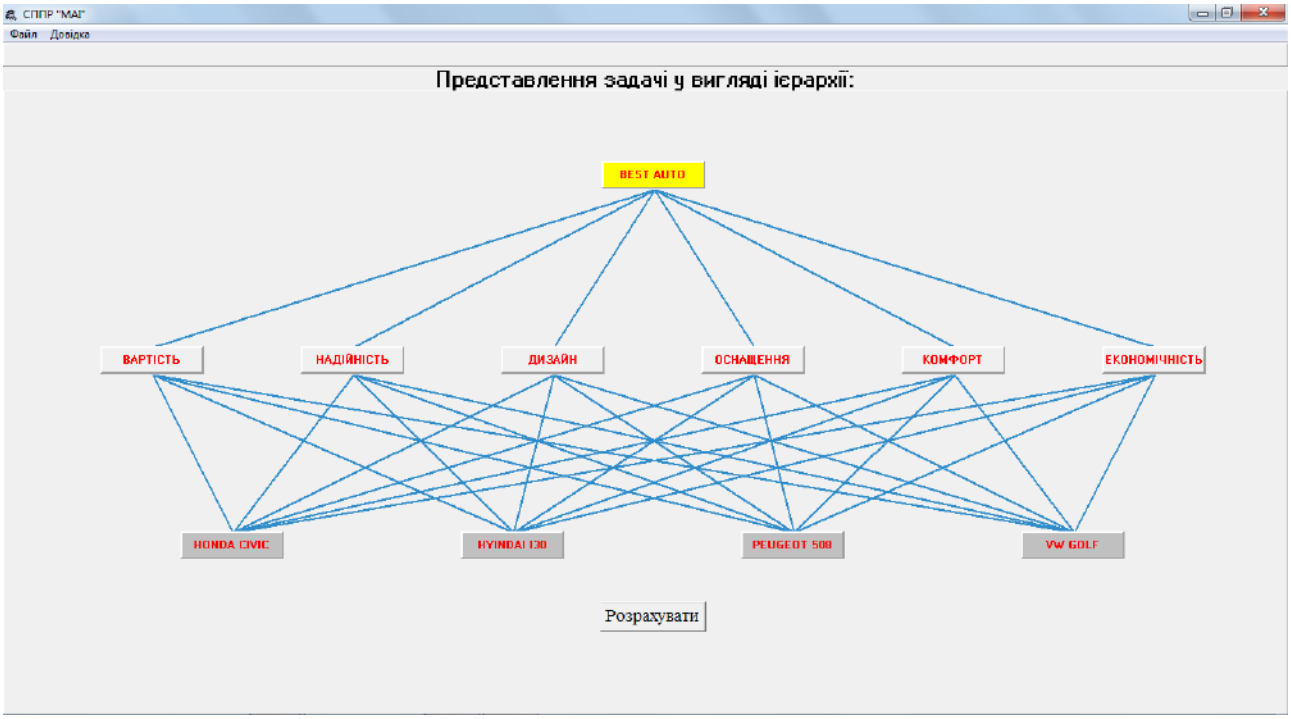
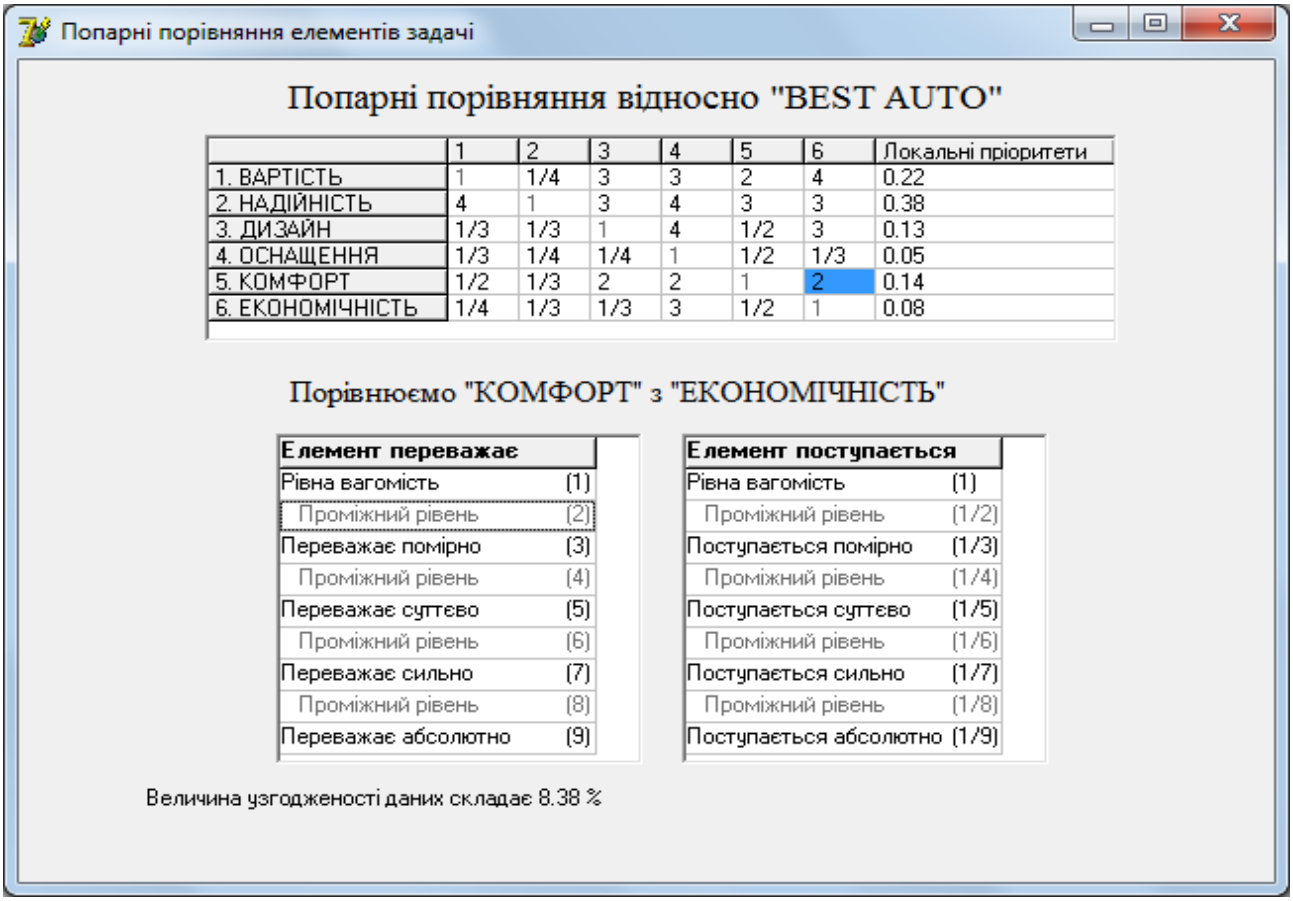


Рис.2 Представлення задачі у вигляді ієрархії

Проводимо попарні порівняння критеріїв відносно мети задачі (рис.3).

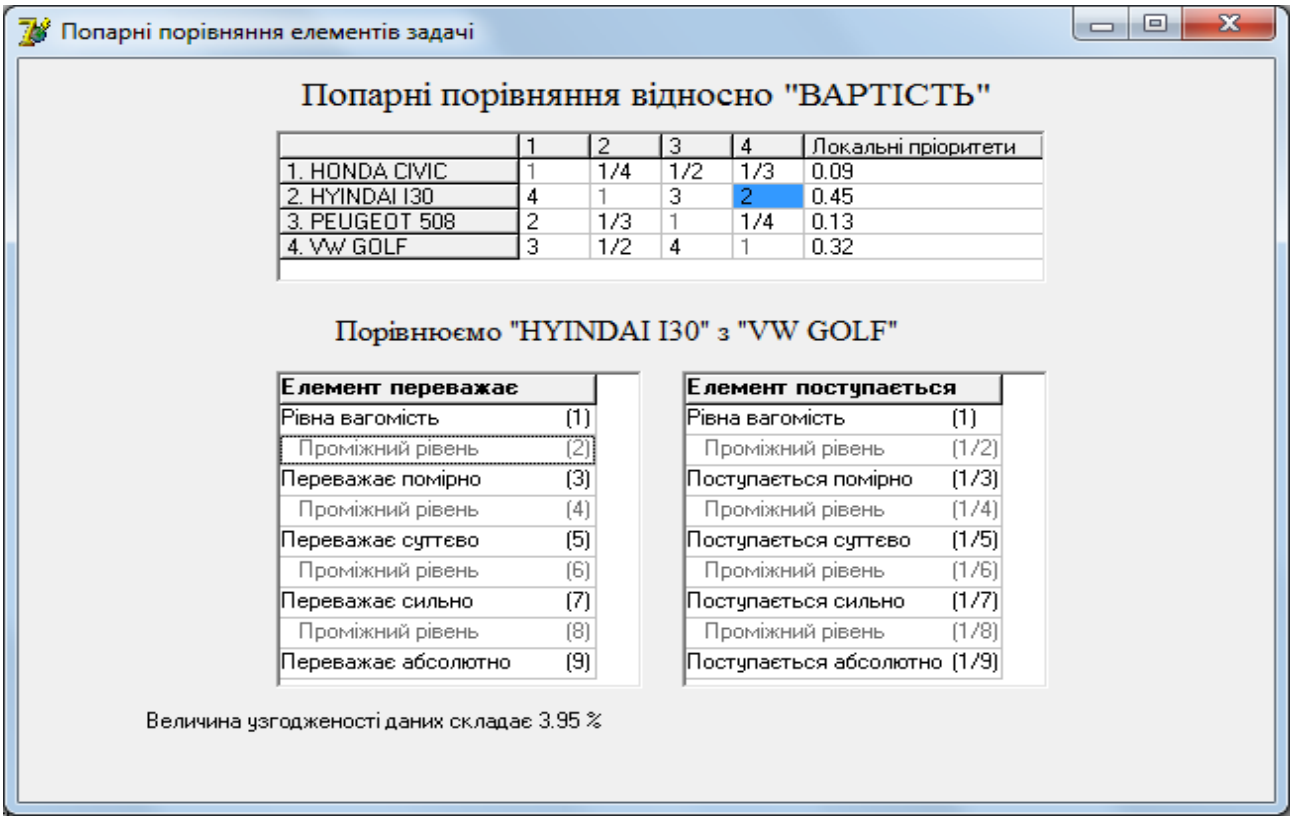


	1	2	3	4	5	6	Локальні пріоритети
1. ВАРТІСТЬ	1	1/4	3	3	2	4	0.22
2. НАДІЙНІСТЬ	4	1	3	4	3	3	0.38
3. ДИЗАЙН	1/3	1/3	1	4	1/2	3	0.13
4. ОСНАЩЕННЯ	1/3	1/4	1/4	1	1/2	1/3	0.05
5. КОМФОРТ	1/2	1/3	2	2	1	2	0.14
6. ЕКОНОМІЧНІСТЬ	1/4	1/3	1/3	3	1/2	1	0.08

Елемент переважає		Елемент поступається	
Рівна вагомість	(1)	Рівна вагомість	(1)
Проміжний рівень	(2)	Проміжний рівень	(1/2)
Переважає помірно	(3)	Поступається помірно	(1/3)
Проміжний рівень	(4)	Проміжний рівень	(1/4)
Переважає суттєво	(5)	Поступається суттєво	(1/5)
Проміжний рівень	(6)	Проміжний рівень	(1/6)
Переважає сильно	(7)	Поступається сильно	(1/7)
Проміжний рівень	(8)	Проміжний рівень	(1/8)
Переважає абсолютно	(9)	Поступається абсолютно	(1/9)

Рис.3 Попарні порівняння критеріїв

Порівнюємо між собою альтернативи відносно кожного з критеріїв (рис.4).



	1	2	3	4	Локальні пріоритети
1. HONDA CIVIC	1	1/4	1/2	1/3	0.09
2. HYINDAI I30	4	1	3	2	0.45
3. PEUGEOT 508	2	1/3	1	1/4	0.13
4. VW GOLF	3	1/2	4	1	0.32

Елемент переважає		Елемент поступається	
Рівна вагомість	(1)	Рівна вагомість	(1)
Проміжний рівень	(2)	Проміжний рівень	(1/2)
Переважає помірно	(3)	Поступається помірно	(1/3)
Проміжний рівень	(4)	Проміжний рівень	(1/4)
Переважає суттєво	(5)	Поступається суттєво	(1/5)
Проміжний рівень	(6)	Проміжний рівень	(1/6)
Переважає сильно	(7)	Поступається сильно	(1/7)
Проміжний рівень	(8)	Проміжний рівень	(1/8)
Переважає абсолютно	(9)	Поступається абсолютно	(1/9)

Рис.4 Попарні порівняння альтернатив

Проводимо розрахунок задачі, результат виведений на рис.5.

Рис.5 Результати розрахунків

Найкращим вибором за даних умов буде альтернатива «Peugeot 508». Розробимо рекомендації для перемоги альтернативи «Hyindai i30» (рис.6).

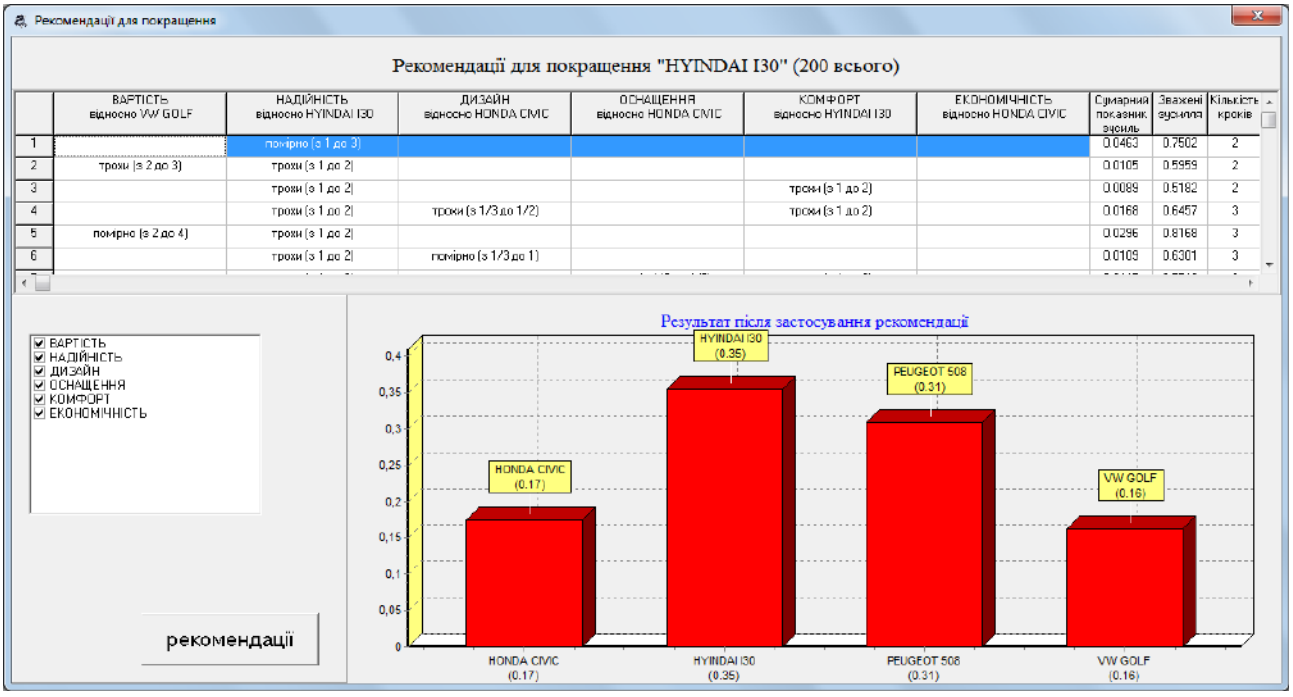


Рис.6 Результати розрахунків для програвшої альтернативи після покращення

На рисунку виведена рекомендація, коли зміна навіть одного критерію дає результат:достатньо помірно (з 1 до 3) покращити надійність «Hyindai i30».

ДОДАТОК Б
ЗАСТОСУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ДО ЗАДАЧІ ВИБОРУ БРОНЕТРАНСПОРТЕРУ

Постановка задачі. МОУ обирає машину (бронетранспортер) для виконання завдання: розвідка окупованої території глибиною 100 км на пересічній місцевості з водними перешкодами; термінова доставка з окупованої в безпечну зону поранених. Час виконання – 5 годин. З вказаної множини автомобілів обрати найкращий за сукупністю заданих критеріїв. Врахувати наступні 5 критеріїв: {«Екіпаж/десант», «Бронювання», «Озброєння», «Рухомість», «Прохідимість»}. Дослідити наступні 5 альтернатив: {«БТР-70», «БТР-80А», «БРДМ-2», «БТР-4 ПАРУС», «Дозор-Б»}.

Розв'язання задачі за допомогою комп'ютерної системи. Спочатку вказуємо кількість критеріїв, альтернатив, назву та коментар задачі (рис.1).

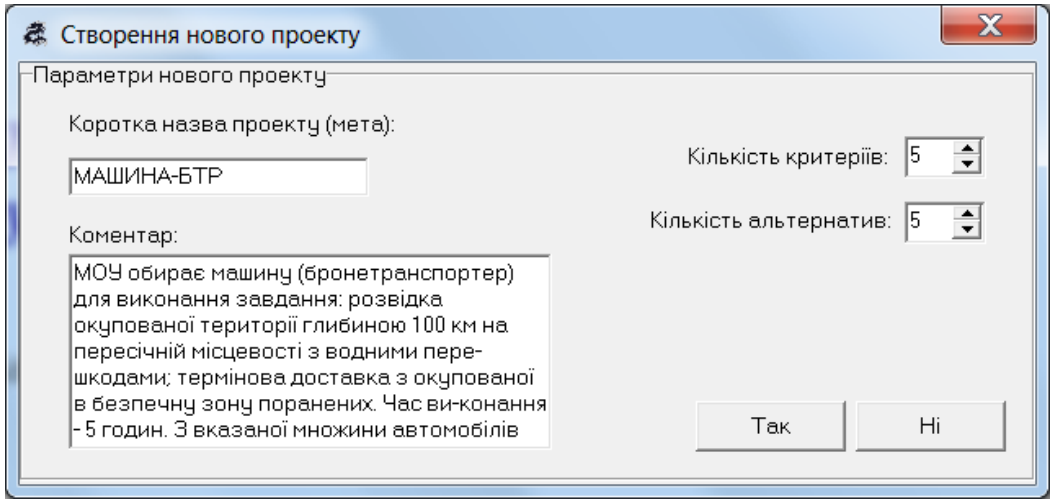


Рис.1 Створення нового проєкту

Далі будуємо ієрархію і вводимо назви та коментарі її елементів (рис.2).

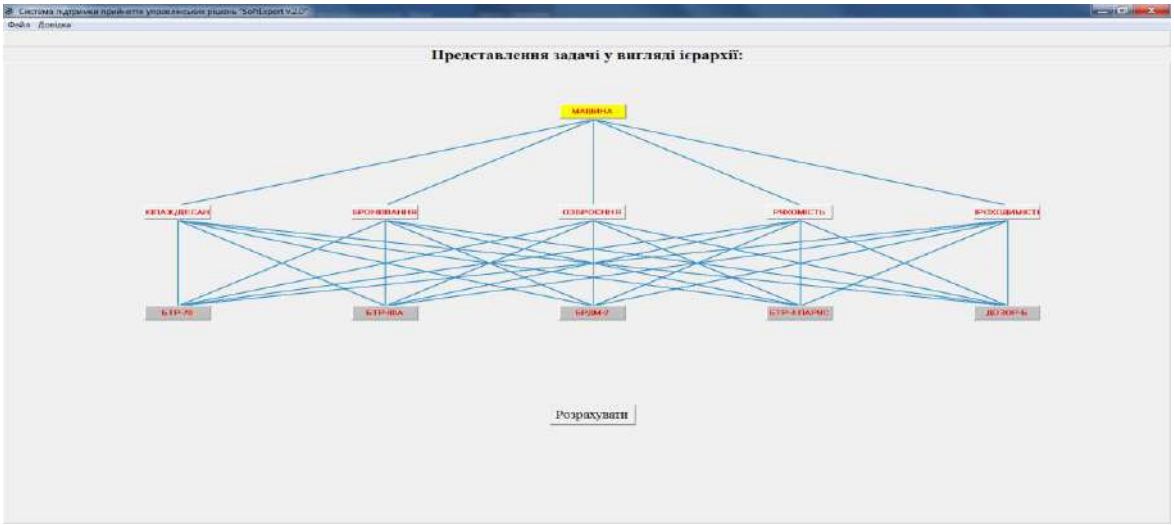


Рис.2 Представлення задачі у вигляді ієрархії

Після проведеггя попарних порівнянь 5 критеріїв між собою та 5 альтернатив (так само, між собою), натискаємо кнопку «Розрахувати». Результат розрахунків представлений на рис.3

Рис.3 Результати розрахунків

Найкращим вибором за даних умов буде альтернатива «БТР-4 ПАРУС». Розробимо рекомендації для перемоги альтернативи «БТР-80А» (рис.4).

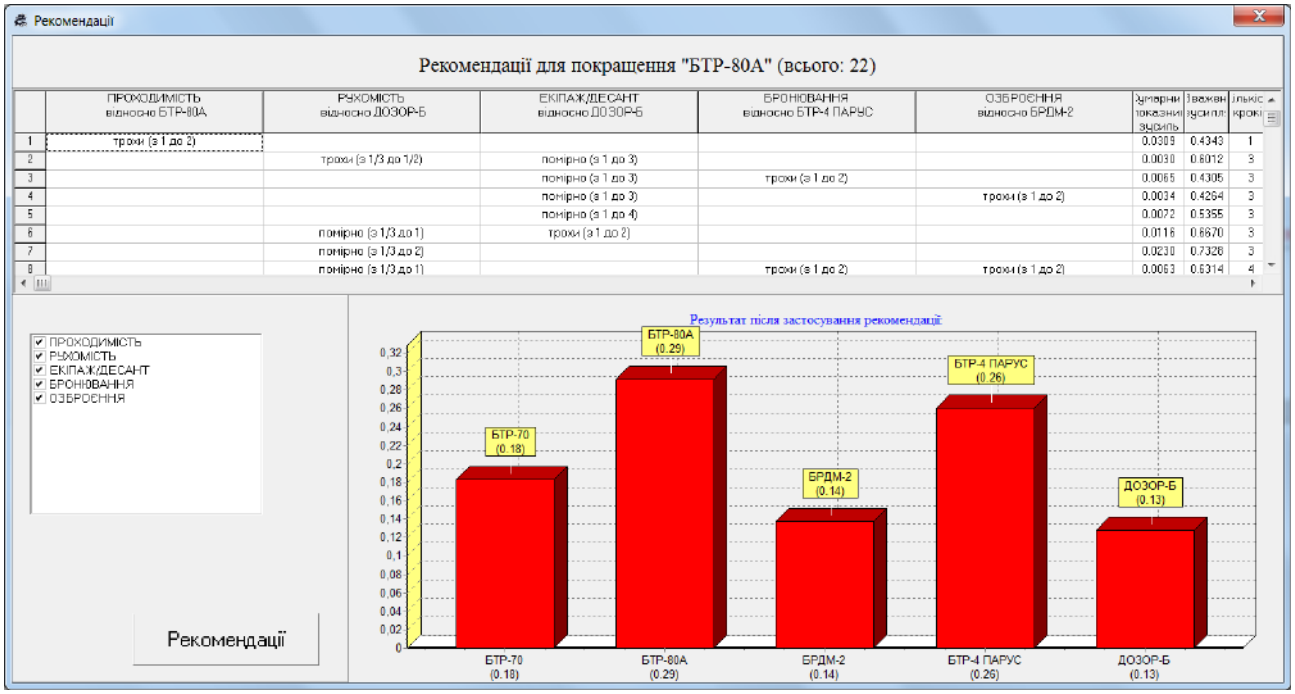


Рис.4 Результати розрахунків для програвшої альтернативи після покращення

На рис.4 виведена рекомендація, коли зміна навіть одного критерію «Проходимість» дає результат: достатньо трохи (з 1 до 2) покращити характеристики проходимості бронетранспортеру «БТР-80А».

ОСНОВНІ ФРАГМЕНТИ ПРОГРАМНОГО КОДУ

Unit1.~pas

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, ExtCtrls, Grids, Outline, DirOutln, StdCtrls, Spin,
  Menus, Unit2, Unit3, Unit4, jpeg, pngimage;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem; //пункт меню Файл
    N2: TMenuItem; //пункт меню Допомога
    N3: TMenuItem; //команда Файл=>Створити новий проект...
    N4: TMenuItem; //команда Файл=>Відкрити проект...
    N5: TMenuItem; //команда Файл=>Зберегти проект
    N6: TMenuItem; //команда Файл=>Зберегти проект як...
    N7: TMenuItem; //відокремлююча лінія -------
    N8: TMenuItem; //команда Вихід
    N9: TMenuItem; //відокремлююча лінія
    N11: TMenuItem; //команда Довідка=>Про програму
    N12: TMenuItem; //команда Файл=>Закрити проект
    Label1: TLabel; //для виведення "Представлення задачі у вигляді ієрархії"
    Panel1: TPanel; //панель угорі головної форми
    dtree: TButton; //невидима мітка, потрібна для обробки кліку правої кнопки
на формі
    PopupMenu1: TPopupMenu; //формує вигляд контексного меню
    N111: TMenuItem; //команда Додати контекстного меню
    N221: TMenuItem; //команда Видалити контекстного меню
    N331: TMenuItem; //команда Редагувати... контекстного меню
    dline: TButton; //кнопка для запуску процедури перемальовування ієрархії
(після зміни елементів)
    OpenDialog1: TOpenDialog; //діалогове вікно Відкриття файлу
    SaveDialog1: TSaveDialog; //діалогове вікно Збереження файлу
    Panel2: TPanel; //для виведення "шапки" з назвою задачі при запуску програми
    Label4: TLabel; //для виведення другого рядка "шапки" з назвою задачі
    Label6: TLabel; //для виведення четвертого рядка "шапки" з назвою задачі
    Label5: TLabel; //для виведення третього рядка "шапки" з назвою задачі
    Label3: TLabel; //для виведення першого рядка "шапки" з назвою задачі
    ptree: TPanel;  //на цій панелі будується ієрархія задачі
    Label2: TLabel;
    result: TButton; //кнопка Розрахувати (Calculate)
    Memo1: TMemo; //для збереження коментарів до задачі

    procedure FormCloseQuery(Sender: TObject; var CanClose:
Boolean);//Підтвердження
                                          //виходу з програми
    procedure N11Click(Sender: TObject);  //Виведення інформації "Про програму"
    procedure N3Click(Sender: TObject);   //Обробка команди "Створити новий
проект..."
                                          //головного меню
    procedure N8Click(Sender: TObject);   //Обробка команди "Вихід" головного
меню
    procedure FormActivate(Sender: TObject);//Вигляд форми при запуску програми
    procedure dtreeClick(Sender: TObject); //Будує на панелі ptree головної
форми ієрархію
    procedure N12Click(Sender: TObject);  //Обробка команди "Закрити проект"
                                          //головного меню
    procedure Label2MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);//Визначає об'єкт, на якому клацнув
користувач,
    //а також вигляд контекстного меню та позицію його появи на головній формі
Form1
    procedure N221Click(Sender: TObject); //Обробка команди "Видалити"
контекстного меню
    procedure dlineClick(Sender: TObject); //Перемальовування ієрархії задачі
    procedure N111Click(Sender: TObject); //Обробка команди "Додати"
```

```
контекстного меню
    procedure N331Click(Sender: TObject); //Обробка команди "Редагувати..."
контекстного меню
    procedure resultClick(Sender: TObject); //Обробка команди "Розрахувати"
    procedure N5Click(Sender: TObject); //Обробка команди "Зберегти проект"
    procedure N4Click(Sender: TObject); //Обробка команди "Відкрити проект..."
    procedure N6Click(Sender: TObject); //Обробка команди "Зберегти проект
як..."

  private
    { Private declarations }
  public
    { Public declarations }
  end;
  TSaveProjectRec = record //використовуємо запис для збереження інформації про
задачу
    NamePnl: string[15];      //Коротка назва задачі
    CommPnl: string[255];     //Коментар до задачі
    NameCrit, NameAlter: array[1..9] of string[15];   //назви критеріїв та
альтернатив
    n, m: Integer;    //Кількість критеріїв та альтернатив відповідно
    CommCrit, CommAlter: array[1..9] of string[255];//коментарі до критеріїв та
альтернатив
    pp: array[0..9,1..9,1..9] of string[3];  //масив збереження всіх попарних
порівнянь
    pn: array[0..9,0..9] of real; //локальні пріоритети
    img: array[1..9] of string[255]; //збереження зображень альтернатив
  end;


var  //Глобальні змінні:
  Form1: TForm1; //Головна форма
  SavePr: TSaveProjectRec; //запис для збереження інфомації про задачу
  ActiveEl, FileExt: string; //Робоча змінна про активний елемент ієрархії:
потрібна для
  //зберігання інформації про елемент задачі, над яким виконав певну дію
користувач
  a,b:array[1..9] of TPanel; //масиви панелей для збереження інформації про
                             //критерії та альтернативи
  full : array[0..9,-9..9] of real;
  rs:array[0..10] of real;
  ainf, binf: array[0..10] of string;
  cA: array[0..9] of integer; // CalcAlt
  bestA, chA:integer;
  pnl: TPanel;  //панель для збереження даних про найвищий елемент ієрархії
(мету)
  Image1: TImage; //робочий об'єкт, потрібен для коректного промальовування
ієрархії
  saveFlag, OpenFlag, chengeFlag, st: Boolean;

const  //Глобальні константи:
  title='Hierarchy of the problem:';
  crit='CRITERION ';
  alter='ALTERNATIVE ';
  vertical=200; //для встановлення вертикальної відстані між рівнями ієрархії

implementation

uses Unit5, Unit6;

 {$R *.dfm}

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
//підтвердження виходу з програми
var
  dlgSave: Integer;
begin
 if chengeFlag then
```

```
begin
 dlgSave:= MessageDlg('Save project?', mtWarning, mbYesNoCancel, 0);
 case dlgSave of
   mrYes:
     begin
       N5.Click;
       if not saveFlag then CanClose:= False else CanClose:= True;
     end;
   mrNo: CanClose:= True;
   mrCancel: CanClose:= False;
 end;
end else
 CanClose:= MessageDlg('Exit?',mtConfirmation,[mbYes,mbNo],0)=IDYES;
end;

procedure TForm1.N11Click(Sender: TObject);
//Виведення інформації "Про програму"
begin
 AboutBox.ShowModal;


end;

procedure TForm1.N3Click(Sender: TObject);
//Обробка команди "Створити новий проект..." головного меню
//Створення об'єкту Image1 на панелі ptree головної форми Form1
//Виклик форми3 ("Створення нового проекту")
var
  dlgSave: Integer;
begin
 if chengeFlag then
 begin
  dlgSave:= MessageDlg('Save project?', mtWarning, mbYesNoCancel, 0);
  case dlgSave of
    mrYes:
      begin
        N5.Click;
        if not saveFlag then Exit;
      end;
    mrNo: chengeFlag:= False;
    mrCancel: Exit;
  end;
 end;
 Form3.ShowModal;
 //panel2.Visible:=false; //приховуємо шапку з назвою задачі
end;

procedure TForm1.N8Click(Sender: TObject);
//Обробка команди "Вихід" головного меню
begin
 Form1.Close; //Закриття головної форми Form1
end;

procedure TForm1.FormActivate(Sender: TObject);
//Вигляд форми при запуску програми
begin
 if not st then
 begin
  OpenDialog1.InitialDir:= ExtractFilePath(Application.ExeName);
  ptree.Visible:=False;
  N12.Enabled:=False;
  N5.Enabled:=False;
  N6.Enabled:=False;
  st:= True;
  Form1.WindowState:=wsMaximized; //Форма розгорнута на весь екран
  panel2.Left:=(Form1.Width-panel2.Width) div 2; // місце виведення шапки задачі
по гориз
  panel2.Top:=(Form1.Height-panel2.Height) div 2 -30; // те ж але по вертикалі
```

```
 end;
end;

procedure TForm1.dtreeClick(Sender: TObject);
//Вперше малює ієрархію задачі на панелі ptree (через настання
//події клацання на допоміжній невидимій кнопці dtree головної форми Form1)
var
 i,j: integer;
 length_space_crit, length_space_alter:integer; //для обрахунку вільного місця
на формі
begin
 Label1.Caption:=title;//(title='Представлення задачі у вигляді ієрархії:')
 Label1.Visible:=true;//виводимо заголовок
 //Створюємо найвищий елемент ієрархії (мету) як панель;
 //задаємо її розміри, властивості та розташування
 pnl:=TPanel.Create(Panel1);//pnl - об'єкт найвищого рівня ієрархії (мета)
 pnl.Parent:=Form1.ptree; //панель pnl розміщується на панелі ptree
 pnl.Name:='pnl'; //задаємо ім'я панелі pnl
 pnl.OnMouseUP:= Label2.OnMouseUp;//Для обробки події натискання кнопок миші на
                                  //об'єкті pnl викликаємо процедуру
 //Задаємо розмір та бордюр елементу:
 pnl.Width:=110;
 pnl.Height:=30;
 pnl.BevelInner:=bvRaised;
 pnl.BevelOuter:=bvRaised;
 pnl.BorderStyle:=bsNone;
 pnl.Font.Style:=[fsBold];
 pnl.Font.Color:=clRed;
 pnl.Color:=clYellow;
 //Задаємо відцентроване розташування елементу найвищого рівня (мета):
 pnl.Top:=Form1.Label1.Height+vertical div 4;
 pnl.Left:=(Form1.Width - pnl.Width) div 2;
  result.Left:=(Form1.Width - result.Width) div 2;

 pnl.Caption:=SavePr.NamePnl;//виводимо на панелі коротку назву задачі
 //Розміщуємо відцентровані рядки критеріїв та альтернатив:
 length_space_crit:=(Form1.Width-SavePr.n*pnl.Width) div (SavePr.n+1);
 length_space_alter:=(Form1.Width-SavePr.m*pnl.Width) div (SavePr.m+1);
 For i:=1 to SavePr.n do
 begin
     a[i]:=TPanel.Create(Panel1);
     a[i].Name:='cr'+IntToStr(i);
     a[i].OnMouseUP:=Label2.OnMouseUp;//Для обробки події натискання кнопок миші
на
                                      //об'єкті-критерії a[i] викликаємо
процедуру
     a[i].Parent:=Form1.ptree;
     a[i].Width:=110;
     a[i].Height:=30;
     a[i].BevelInner:=bvRaised;
     a[i].BevelOuter:=bvRaised;
     a[i].BorderStyle:=bsNone;
     a[i].Font.Style:=[fsBold];
     a[i].Font.Color:=clRed;
     a[i].Top:=pnl.Top+vertical;
     a[i].Left:= i*length_space_crit+(i-1)*pnl.Width;
     if OpenFlag then a[i].Caption:=SavePr.NameCrit[i] else
     begin
      SavePr.NameCrit[i]:=crit+IntToStr(i);
      a[i].Caption:=SavePr.NameCrit[i];
     end;
 end;
 For i:=1 to SavePr.m do
 begin
     b[i]:=TPanel.Create(Panel1);
     b[i].Parent:=Form1.ptree;
     b[i].Name:='al'+IntToStr(i);
     b[i].OnMouseUP:=Label2.OnMouseUp;//Для обробки події натискання кнопок миші
```

```
на
                                        //об'єкті-альтернативі b[i] викликаємо
процедуру
     b[i].Width:=110;
     b[i].Height:=30;
     b[i].BevelInner:=bvRaised;
     b[i].BevelOuter:=bvRaised;
     b[i].BorderStyle:=bsNone;
     b[i].Font.Style:=[fsBold];
     b[i].Font.Color:=clRed;
     b[i].Color:=clSilver;
     b[i].Top:=pnl.Top+2*vertical;
     b[i].Left:= i*length_space_alter+(i-1)*pnl.Width;
     if OpenFlag then b[i].Caption:=SavePr.NameAlter[i] else
     begin
      SavePr.NameAlter[i]:=alter+IntToStr(i);
      b[i].Caption:=SavePr.NameAlter[i];
     end;
 end;
   //Будуємо лінії зв'язку між метою та критеріями:
   Image1.Canvas.MoveTo(pnl.Left+pnl.Width div 2, pnl.Top+pnl.Height);
   Image1.Canvas.Pen.Width:=2;  //встановлюємо товщину ліній
   Image1.Canvas.Pen.Color:=rgb(47,140,202); //встановлюємо колір ліній
   For i:=1 to SavePr.n do
   begin
      Image1.Canvas.LineTo(a[i].Left+a[i].Width div 2, a[i].Top);
      Image1.Canvas.MoveTo(pnl.Left+pnl.Width div 2, pnl.Top+pnl.Height);
   end;
   //Будуємо лінії зв'язку між критеріями та альтернативами:
   For i:=1 to SavePr.n do
   begin
      Image1.Canvas.MoveTo(a[i].Left+a[i].Width div 2, a[i].Top+pnl.Height);
      For j:=1 to SavePr.m do
      begin
         Image1.Canvas.LineTo(b[j].Left+b[j].Width div 2, b[j].Top);
         Image1.Canvas.MoveTo(a[i].Left+a[i].Width div 2, a[i].Top+pnl.Height);
      end;
   end;
 result.Visible:=True;
end;

procedure TForm1.N12Click(Sender: TObject);
//Обробка команди "Закрити проект" головного меню
var l, dlgSave, i, j, k: integer;
begin
 if chengeFlag then
 begin
  dlgSave:= MessageDlg('Save previous project?', mtWarning, mbYesNoCancel, 0);
  case dlgSave of
    mrYes:
      begin
        N5.Click;
        if not saveFlag then Exit;
      end;
    mrCancel: Exit;
  end;
 end;
 Image1.Free; //знищуємо об'єкти (лінії), намальовані на об'єкті Image1 панелі
ptree
 For l:=1 to SavePr.n do  a[l].destroy;//знищуємо панелі критеріїв
 For l:=1 to SavePr.m do  b[l].destroy;//знищуємо панелі альтернатив
 pnl.Free;
 ptree.Visible:=False;
 with SavePr do
 begin
 NamePnl:= '';
 CommPnl:= '';
 for i:=1 to 9 do
```

```
begin
 NameCrit[i]:= '';
 CommCrit[i]:= '';
 NameAlter[i]:= '';
 CommAlter[i]:= '';
 img[i-1]:= '';
end;
for i:=0 to 9 do
 for j:=0 to 9 do
  pn[i, j]:= 0;
for k:=0 to 9 do
 for i:=1 to 9 do
  for j:=1 to 9 do
   pp[k, i, j]:= '';
end;
for i:=0 to 9 do
begin
 cA[i]:= 0;
 for j:=-9 to 9 do
  full[i, j]:= 0;
end;
for i:=0 to 10 do
begin
 rs[i]:= 0;
 ainf[i]:= '';
 binf[i]:= '';
end;
bestA:= 0;
chA:= 0;
result.Visible:=False;
Label1.Visible:=false;//приховуємо заголовок "Представлення задачі у вигляді
ієрархії:"
N3.Enabled:=True;
N4.Enabled:=True;
N5.Enabled:=False;
N6.Enabled:=False;
N12.Enabled:=False;
chengeFlag:=False;
saveFlag:=False;
panel2.Visible:=true; //відновлюємо виведення шапки з назвою задачі
end;

procedure TForm1.Label2MouseUp(Sender: TObject; Button: TMouseButton;
          Shift: TShiftState; X, Y: Integer);
//Визначає активний об'єкт (той, на якому натиснув кнопку миші користувач),
//а також вигляд контекстного меню та позицію його появи на головній формі Form1
var
  foo: TPoint;
begin
 ActiveEl:=Tpanel(Sender).Name;//ім'я активного елементу
 if(ActiveEl='pnl') then  //задаємо вигляд контекстного меню для мети
 begin
    PopupMenu1.Items.Items[0].Visible:=false;
    PopupMenu1.Items.Items[1].Visible:=false;
 end else //задаємо вигляд контекстного меню для критеріїв та альтернатив
         //та враховуємо, що об'єктів не може бути більше ніж 9
 begin
   PopupMenu1.Items.Items[0].Visible:=true;
   PopupMenu1.Items.Items[1].Visible:=true;
  if(ActiveEl[1]='c') then
       if (SavePr.n=9) then PopupMenu1.Items.Items[0].Visible:=false
                       else if (SavePr.n=2) then
PopupMenu1.Items.Items[1].Visible:=false
                       else begin PopupMenu1.Items.Items[0].Visible:=true;
                            PopupMenu1.Items.Items[1].Visible:=true;
                            end;
  if(ActiveEl[1]='a') then
  begin
```

```
                                        Unit1.~pas
  if (SavePr.m=9) then PopupMenu1.Items.Items[0].Visible:=false
                       else if (SavePr.m=2) then
PopupMenu1.Items.Items[1].Visible:=false
                       else begin PopupMenu1.Items.Items[0].Visible:=true;
                            PopupMenu1.Items.Items[1].Visible:=true;
                       end;
  end;
 end;
 GetCursorPos(foo);//знаходимо точку foo, де клацнув мишкою користувач
 if Button = mbRight then  PopupMenu1.Popup(foo.X, foo.Y);//вказуємо, щоб
 //контекстне меню з'явилось в позиції курсора миші (в точці (foo.X,foo.Y))
 if Button = mbLeft then
  if not(ActiveEl[1]='a') then Form5.ShowModal;//починаємо попарні порівняння
end;

procedure TForm1.N221Click(Sender: TObject);
//Обробка команди "Видалити" контекстного меню
var i, l, h, jj, ii: Integer;
begin
 if pos('cr',ActiveEl)>0 then //здійснили клік на критерії
 begin
   i:=StrToInt(ActiveEl[3]); //знайшли, що буде видалятись саме і-й критерій
 //знищуємо і-й стовпчик та і-й рядок головної матриці (матриці попарних
порівнянь критеріїв):
   with SavePr do
    if i=9 then
     for ii:=1 to 9 do
     begin
      pp[0,i,ii]:='1';
      pp[0,ii,i]:='1';
     end
    else
    begin
     for jj:=i to 8 do
      for ii:=1 to 9 do
       pp[0,ii,jj]:=pp[0,ii,jj+1];

     for ii:=i to 8 do
      for jj:=1 to 9 do
       pp[0,ii,jj]:=pp[0,ii+1,jj];
     for ii:=1 to 9 do
     begin
      pp[0,n,ii]:='1';
      pp[0,ii,n]:='1';
     end;
    end;

 //враховуємо знищення одного об'єкта при подальшому відображенні:

   for l:=i to SavePr.n-1 do
   begin
      a[l].Caption:=a[l+1].Caption;
     SavePr.pp[l]:=SavePr.pp[l+1];
     a[l].name:= 'cr'+IntToStr(l);
     SavePr.NameCrit[l]:= SavePr.NameCrit[l+1];
     SavePr.CommCrit[l]:= SavePr.CommCrit[l+1];
   end;
   a[SavePr.n].destroy; //знищуємо 1 (останній) елемент масиву критеріїв a[i]
   SavePr.n:=SavePr.n-1; //зменшуємо кількість критеріїв на 1
  end // Завершили знищення одного критерію, передбачаємо те ж саме для
альтернатив:
 else
 begin
   i:=StrToInt(ActiveEl[3]); //знайшли, що буде видалятись саме і-та
альтернатива
  //знищуємо і-й стовпчик та і-й рядок кожної матриці критеріїв
  //(тому присутній зовнішній цикл по всім критеріям):
   for h:=1 to 9 do
                                        Страница 7
```

```
   begin
    with SavePr do
    if i=9 then
     for ii:=1 to 9 do
     begin
      pp[h,i,ii]:='1';
      pp[h,ii,i]:='1';
     end
    else
    begin
     for jj:=i to 8 do
      for ii:=1 to 9 do
       pp[h,ii,jj]:=pp[h,ii,jj+1];

     for ii:=i to 8 do
      for jj:=1 to 9 do
       pp[h,ii,jj]:=pp[h,ii+1,jj];


     for ii:=1 to 9 do
     begin
      pp[h,m,ii]:='1';
      pp[h,ii,m]:='1';
     end;
    end;
   end;
  //враховуємо знищення одного об'єкта при подальшому відображенні:
    for l:=i to SavePr.m-1 do
    begin
      b[l].Caption:=b[l+1].Caption;
      b[l].name:= 'al'+IntToStr(l);
      SavePr.NameAlter[l]:= SavePr.NameAlter[l+1];
      SavePr.CommAlter[l]:= SavePr.CommAlter[l+1];
      SavePr.img[l]:= SavePr.img[l+1];
    end;
      SavePr.img[SavePr.m]:= '';
      b[SavePr.m].destroy; //знищуємо 1 (останній) елемент масиву альтернатив
b[i]
      SavePr.m:=SavePr.m-1; //зменшуємо кількість альтернатив на 1
 end;  // Завершили знищення однієї альтернативи
 dline.Click;//Викликаємо процедуру перемалювання ієрархії після знищення
об'єкта
 chengeFlag:=True;
end;

procedure TForm1.dlineClick(Sender: TObject);
//Перемалювання ієрархії після знищення або додавання елементів
//(через настання події клацання на допоміжній невидимій кнопці dline
//панелі Panel1 головної форми Form1)
var
  i, j: Integer;
  length_space_crit, length_space_alter: Integer;
begin
         //Задаємо параметри та розміщуємо рядки критеріїв та альтернатив
 length_space_crit:=(Form1.Width-SavePr.n*pnl.Width) div (SavePr.n+1);
 length_space_alter:=(Form1.Width-SavePr.m*pnl.Width) div (SavePr.m+1);
 For i:=1 to SavePr.n do
 begin
     a[i].Width:=110;
     a[i].Height:=30;
     a[i].BevelInner:=bvRaised;
     a[i].BevelOuter:=bvRaised;
     a[i].BorderStyle:=bsNone;
     a[i].Top:=pnl.Top+vertical;
     a[i].Left:= i*length_space_crit+(i-1)*pnl.Width;
 end;
 For i:=1 to SavePr.m do
 begin
```

```
      b[i].Width:=110;
      b[i].Height:=30;
      b[i].BevelInner:=bvRaised;
      b[i].BevelOuter:=bvRaised;
      b[i].BorderStyle:=bsNone;
      b[i].Top:=pnl.Top+2*vertical;
      b[i].Left:= i*length_space_alter+(i-1)*pnl.Width;
  end;

  //задаємо колір ліній та розмір полотна
  Image1.Canvas.brush.Color:=clBtnFace;
  Image1.Canvas.Rectangle(-10,-10,screen.Width+10,screen.Height+10);

  //Будуємо нові лінії зв'язку між метою та критеріями
  Image1.Canvas.MoveTo(pnl.Left+pnl.Width div 2, pnl.Top+pnl.Height);
  For i:=1 to SavePr.n do
      begin
        Image1.Canvas.LineTo(a[i].Left+a[i].Width div 2, a[i].Top);
        Image1.Canvas.MoveTo(pnl.Left+pnl.Width div 2, pnl.Top+pnl.Height);
      end;
  //Будуємо нові лінії зв'язку між критеріями та альтернативами
  For i:=1 to SavePr.n do
      begin
        Image1.Canvas.MoveTo(a[i].Left+a[i].Width div 2, a[i].Top+pnl.Height);
        For j:=1 to SavePr.m do
        begin
          Image1.Canvas.LineTo(b[j].Left+b[j].Width div 2, b[j].Top);
          Image1.Canvas.MoveTo(a[i].Left+a[i].Width div 2, a[i].Top+pnl.Height);
        end;
      end;

end;

procedure TForm1.N111Click(Sender: TObject);
//Обробка команди "Додати" контекстного меню
var i, l, ii, jj, h: Integer;
begin
  if pos('cr',ActiveEl)>0 then
  begin
   with SavePr do
   begin
    n:=n+1;
    i:=StrToInt(ActiveEl[3]);
    a[n]:=TPanel.Create(Panel1);
    a[n].Name:='cr'+inttostr(n);
    a[n].OnMouseUP:=Label2.OnMouseUp; //Для обробки події натискання кнопок миші
  //на об'єкті-критерії a[n+1] викликаємо процедуру
    a[n].Parent:=Form1.ptree;
    a[n].Width:=110;
    a[n].Height:=30;
    a[n].BevelInner:=bvRaised;
    a[n].BevelOuter:=bvRaised;
    a[n].BorderStyle:=bsNone;
    a[n].Font.Style:=[fsBold];
    a[n].Font.Color:=clRed;
    a[n].Caption:=crit+IntToStr(n);
    for jj:=9 downto i+1 do
     for ii:=9 downto 1 do
      pp[0,ii,jj]:=pp[0,ii,jj-1];
    for ii:=9 downto i+1 do
     for jj:=9 downto 1 do
      pp[0,ii,jj]:=pp[0,ii-1,jj];
    for l:=n downto i+1 do
    begin
      a[l].Caption:=a[l-1].Caption;
      pp[l]:= pp[l-1];
      NameCrit[l]:= NameCrit[l-1];
      CommCrit[l]:= CommCrit[l-1];
```

```
      a[l].name:= 'cr'+inttostr(l);
     end;
     for ii:=1 to 9 do
     begin
       pp[0,i,ii]:='1';
       pp[0,ii,i]:='1';
     end;
     for jj:=1 to 9 do
      for ii:=1 to 9 do
       pp[l,ii,jj]:='1';
     a[i].Caption:= crit;
     NameCrit[i]:= crit;
     CommCrit[i]:= '';
    end;
   end // Критерії
   else
   begin
    with SavePr do
    begin
    i:=strtoint(ActiveEl[3]);
    for h:=1 to 9 do
    begin
     for jj:=9 downto i+1 do
      for ii:=9 downto 1 do
       pp[h,ii,jj]:=pp[h,ii,jj-1];
     for ii:=9 downto i+1 do
      for jj:=9 downto 1 do
       pp[h,ii,jj]:=pp[h,ii-1,jj];
     for ii:=1 to 9 do
     begin
      pp[h,i,ii]:='1';
      pp[h,ii,i]:='1';
     end;
    end;
    m:=m+1;
    b[m]:=TPanel.Create(Panel1);
    b[m].Name:='al'+inttostr(m);
    b[m].OnMouseUP:=Label2.OnMouseUp; //Для обробки події натискання кнопок миші
   //об'єкті-альтернативі b[m+1] викликаємо процедуру
    b[m].Parent:=Form1.ptree;
    b[m].Width:=110;
    b[m].Height:=30;
    b[m].BevelInner:=bvRaised;
    b[m].BevelOuter:=bvRaised;
    b[m].BorderStyle:=bsNone;
    b[m].Font.Style:=[fsBold];
    b[m].Font.Color:=clRed;
    b[m].Color:=clSilver;
    b[m].Caption:=alter+IntToStr(m);
    for l:=m downto i+1 do
    begin
     b[l].Caption:=b[l-1].Caption;
     b[l].name:= 'al'+inttostr(l);
     CommAlter[l]:= CommAlter[l-1];
     NameAlter[l]:= NameAlter[l-1];
     img[l]:= img[l-1];
    end;
    b[i].Caption:= alter;
    NameAlter[i]:= alter;
    CommAlter[i]:= '';
    img[i]:= '';
    end;
  end;  // Альтернативи
 dline.Click;//Виклик процедури перемалювання ієрархії
 chengeFlag:=True;
end;

procedure TForm1.N331Click(Sender: TObject);
```

```
//Обробка команди "Редагувати..." контекстного меню
begin
 Form4.ShowModal; //Виклик форми "Редагування елементу ієрархії" Form4
end;

procedure TForm1.resultClick(Sender: TObject);
//Обробка натискання на кнопку Розрахувати головної форми Form1
var
  i: Integer;
begin
  ActiveEl:='pnl';
  Form5.Show; Form5.Close;
  Form1.Show;
  for i:=1 to SavePr.n do
  begin
   ActiveEl:='cr'+inttostr(i);
   Form5.Show; Form5.Close;;
   Form1.Show; ;
  end;
  Form6.ShowModal; //Виводимо форму з результатами розрахунків
end;

procedure TForm1.N5Click(Sender: TObject);
//Обробка команди Файл->Зберегти проект
var
  FileSave: file of TSaveProjectRec;
begin
 if saveFlag then
 begin
     AssignFile(FileSave, FileExt);
     Rewrite(FileSave);
     Write(FileSave, SavePr);
     CloseFile(FileSave);
     chengeFlag:=False;
 end
 else N6.Click;//Виклик процедури обробки команди Файл->Зберегти проект як...
end;

procedure TForm1.N4Click(Sender: TObject);
//Обробка команди Файл->Відкрити проект...
var
  FileSave: file of TSaveProjectRec;
begin
  if OpenDialog1.Execute then
  begin
    OpenFlag:=True;
    if N12.Enabled then N12.Click;
    FileExt:=OpenDialog1.FileName;
    AssignFile(FileSave, FileExt);
    Reset(FileSave);
    read(FileSave, SavePr);
    CloseFile(FileSave);
    N3.Enabled:=False;
    N4.Enabled:=False;
    N12.Enabled:=True;
    N5.Enabled:=True;
    N6.Enabled:=True;
    ptree.Visible:=true;
    Image1:=TImage.Create(ptree);
    Image1.Parent:=ptree;
    Image1.Align:=alClient;
    Image1.Canvas.brush.Color:=clBtnFace;
    Image1.Canvas.Rectangle(-10,-10,screen.Width+10,screen.Height+10);
    dtree.Click;
    saveFlag:=True;
    chengeFlag:=False;
    OpenFlag:=False;
    panel2.Visible:=false; //приховуємо виведення шапки з назвою задачі
```

```
  end;
end;

procedure TForm1.N6Click(Sender: TObject);
//Обробка команди Файл->Зберегти проект як...
var
  FileSave: file of TSaveProjectRec;
begin
 if SaveDialog1.Execute then
  begin
     saveFlag:=True;
     FileExt:=SaveDialog1.FileName;
     AssignFile(FileSave, FileExt);
     Rewrite(FileSave);
     Write(FileSave, SavePr);
     CloseFile(FileSave);
     chengeFlag:=False;
  end;
end;

end.
```

Unit2.pas

```
unit Unit2; //Форма "Про програму"

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls;

type
  TAboutBox = class(TForm)
    Panel1: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    OKButton: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AboutBox: TAboutBox;

implementation

{$R *.dfm}

end.
```

```
unit Unit3; //форма "Створення нового проекту", в якій задаємо основні параметри
            //задачі, яка розв'язується

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, Spin;

type
  TForm3 = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Memo1: TMemo;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;

    procedure Button2Click(Sender: TObject);//Обробка натискання на кнопку "Ні"
форми Form3
    procedure Button1Click(Sender: TObject);//Обробка натискання на кнопку "Так"
форми Form3

  private
    { Private declarations }
  public
    { Public declarations }

  end;

var
  Form3: TForm3;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm3.Button2Click(Sender: TObject);
//Обробка натискання на кнопку "Ні" форми Form3
begin
 Form1.panel2.Visible:=true; //відновлюємо виведення шапки з назвою задачі
 Close;  //Закриваємо форму Form3
 end;

procedure TForm3.Button1Click(Sender: TObject);
var l, i, j: Integer;
//Обробка натискання на кнопку "Так" форми Form3
begin
 Form1.panel2.Visible:=false; //приховуємо шапку з назвою задачі
 if Form1.N12.Enabled then Form1.N12.Click;
 with Form1 do
 begin
  N3.Enabled:=False;
  N4.Enabled:=False;
  ptree.Visible:=True;   // Показуємо панель ptree
  N12.Enabled:=True;
  N5.Enabled:=True;
  N6.Enabled:=True;
  chengeFlag:=True;
  saveFlag:=False;
  OpenFlag:=False;
```

```
  Image1:=TImage.Create(ptree);
  Image1.Parent:=ptree;
  Image1.Align:=alClient;
  Image1.Canvas.brush.Color:=clBtnFace;
  Image1.Canvas.Rectangle(-10,-10,screen.Width+10,screen.Height+10);
 end;
 for l:=0 to 9 do
  for j:=1 to 9 do
   for i:=1 to 9 do
    SavePr.pp[l,i,j]:='1';   //масиву попарних порівнянь присвоюємо початкове
значення 1
 SavePr.n:=SpinEdit1.Value;  //Кількість критеріїв
 SavePr.m:=SpinEdit2.Value;  //Кількість альтернатив
 SavePr.NamePnl:=Edit1.Text;    //Назва (мета) задачі
 SavePr.CommPnl:=Memo1.Text;    //Опис задачі
 if length(SavePr.NamePnl)<1 then SavePr.NamePnl:='GOAL';
 Close;
 Form1.dtree.Click;  //Вперше малює ієрархію задачі на панелі ptree (через
настання
 //події клацання на допоміжній невидимій кнопці dtree головної форми Form1)
end;

end.
```

Unit4.~pas

```
unit Unit4;//форма для редагування мети, критеріїв або альтернатив

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtDlgs, ExtCtrls, jpeg;

type
  TForm4 = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    OpenDialog1: TOpenDialog;
    Image1: TImage;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button3Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;
  i: Integer;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm4.Button2Click(Sender: TObject);
//Обробка натискання кнопки "Ні" форми "Редагування елементу ієрархії" Form4
begin
 OpenDialog1.FileName:='';
 Close; //Закриття форми "Редагування елементу ієрархії" Form4
end;

procedure TForm4.Button1Click(Sender: TObject);
//Обробка натискання кнопки "Так" форми "Редагування елементу ієрархії" Form4
begin
 if(ActiveEl='pnl') then
 begin
  pnl.Caption:=Edit1.Text;
  SavePr.NamePnl:=Edit1.Text;
  SavePr.CommPnl:=Memo1.Text;
 end
 else
 begin
  if(ActiveEl[1]='c') then
  begin
   a[i].Caption:=Edit1.Text;
   SavePr.NameCrit[i]:=Edit1.Text;
   SavePr.CommCrit[i]:=Memo1.Text;
  end;
  if(ActiveEl[1]='a') then
  begin
   b[i].Caption:=Edit1.Text;
```

```
    SavePr.NameAlter[i]:=Edit1.Text;
    SavePr.CommAlter[i]:=Memo1.Text;
    SavePr.img[i]:=ExtractFileName(OpenDialog1.FileName);
    if DirectoryExists(ExtractFilePath(Application.ExeName)+'Images\') then
     CopyFile(PChar(OpenDialog1.FileName),
PChar(ExtractFilePath(Application.ExeName)+'Images\'+
ExtractFileName(SavePr.img[i])), True)
    else
    begin
     CreateDir(ExtractFilePath(Application.ExeName)+'Images\');
     CopyFile(PChar(OpenDialog1.FileName),
PChar(ExtractFilePath(Application.ExeName)+'Images\'+
ExtractFileName(SavePr.img[i])), True);
    end;
    OpenDialog1.FileName:='';
  end;
 end;
 chengeFlag:=True;
 Close; //Закриття форми "Редагування елементу ієрархії" Form4
end;

procedure TForm4.FormActivate(Sender: TObject);
//дії при активації форми4 "Редагування елементу ієрархії"
begin
 Image1.Visible:=False;//приховали зображення альтернативи
 Button3.Visible:=False;//приховали кнопку Завантажити фото
 if(ActiveEl='pnl') then
 begin
  Memo1.Text:=SavePr.CommPnl;
  Edit1.Text:=SavePr.NamePnl;
 end
 else
 begin
  i:= StrToInt(ActiveEl[3]);
  if(ActiveEl[1]='c') then
  begin
   Edit1.Text:=SavePr.NameCrit[i];
   Memo1.Text:=SavePr.CommCrit[i];
  end;
  if(ActiveEl[1]='a') then
  begin
   Edit1.Text:=SavePr.NameAlter[i];
   Memo1.Text:=SavePr.CommAlter[i];
   Button3.Visible:=True;//кнопка "Завантажити фото" стала видимою
   if not(Length(SavePr.img[i])=0) then
   begin
    SavePr.img[i]:= ExtractFilePath(Application.ExeName)+
   'Images\'+ ExtractFileName(SavePr.img[i]);
    image1.Picture.LoadFromFile(SavePr.img[i]);
    SavePr.img[i]:= ExtractFileName(SavePr.img[i]);
    image1.Visible:=True;//Фото альтернативи стало видимим
   end;
  end;
 end;
 Edit1.SetFocus; Edit1.SelectAll;
end;

procedure TForm4.Button3Click(Sender: TObject);
begin
 if OpenDialog1.Execute then
 begin
  image1.Picture.LoadFromFile(OpenDialog1.FileName);
  image1.Visible:=true;
 end;
end;

end.
```

Unit5.~pas

```
unit Unit5;
{1.створює таблицю для попарних порівнянь (StringGrid1)
 2.створює таблиці лінгвістичної шкали (StringGrid2 та StringGrid3)
 3.обчислює локальні пріоритети (масив pn)
 4.обчислює величину узгодженості (x)
 5.результати попарних порівнянь зберігає у 3-вимірному масиві pp
   та зберігає результати у 3-вимірному масиві pp}
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, StdCtrls,Math, ExtCtrls;

type
  TForm5 = class(TForm)
    ListBox1: TListBox;//містить 9 фраз від "Рівна вагомість" до "Переважає
абсолютно"
    ListBox2: TListBox;//містить 9 фраз від "Рівна вагомість" до "Поступається
абсолютно"
    Button1: TButton;
    Bevel1: TBevel;
    Panel1: TPanel;//для центрування заголовку елементів, які порівнюються
    Panel2: TPanel;//для центрування таблиці попарних порівнянь
    Panel3: TPanel;//для центрування обох частин лінгвістичної шкали
    StringGrid1: TStringGrid;//таблиця для введення попарних порівнянь елементів
    StringGrid2: TStringGrid;//таблиця для збереження 1-ої частини лінгвістичної
шкали
    StringGrid3: TStringGrid;//таблиця для збереження 2-ої частини лінгвістичної
шкали
    Label4: TLabel;//для виведення назви елемента вищого рівня, по відношенню до
якого

порівнюються елементи
    Label3: TLabel;//для виведення назв елементів, які порівнюються між собою
    Label2: TLabel;//для виведення повідомлення про перевищення величини
узгодженості 20%
    Label1: TLabel;//для виведення величини узгодженості
    procedure StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;
    Rect: TRect; State: TGridDrawState);//промальовує таблицю для попарних
порівнянь
    procedure StringGrid2DrawCell(Sender: TObject; ACol, ARow: Integer;
    Rect: TRect; State: TGridDrawState);//промальовує таблицю з 1-шою частиною
лінгв. шкали
    procedure StringGrid3DrawCell(Sender: TObject; ACol, ARow: Integer;
    Rect: TRect; State: TGridDrawState);//промальовує таблицю з 2-ою частиною
лінгв. шкали
    procedure StringGrid2SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
    procedure StringGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
    procedure StringGrid3SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
    procedure FormActivate(Sender: TObject);//реалізує дії після активації форми
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
end;

const //величини випадкової узгодженості для матриць порядку до 10
      //взято з довідника
  Vu : Array [1..10] Of real =(0, 0.01, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45,
1.49);

var
```

```
  Form5: TForm5;
  posx, posy: Integer;

implementation

uses Unit1;

{$R *.dfm}
Function RealToStr(r: Real; p: byte): String;
{Перетворює тип Real на String, p - кількість символів дробної частини}
begin
 Str(r:30:p,Result); //Перетворення real - string
 Result:=Trim(Result); //Відкидаємо пропуски ліворуч та праворуч
end;

procedure TForm5.StringGrid2DrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
//Створення першої частини лінгвістичної шкали коли елемент переважає інший
begin
 if (ACol=0) and (ARow=0) then //ліва верхня комірка лінгвістичної шкали
 with StringGrid2.Canvas do
  begin
    FillRect(Rect);
    Font.Color:= clBlack;
    Font.Style:= [fsBold];
    TextOut(Rect.Left, Rect.Top, 'Element exceeds');
  end;
 Font.Size:=3;
 if (ACol=0) and (ARow>=1) then //комірки першого стовпчика лінгвістичної шкали
 with StringGrid2.Canvas do
  begin
    FillRect(Rect);
    if (ARow mod 2) = 1 then Font.Color:= clBlack чорний колір для основних
рівнів
                        else Font.Color:= clGrayText;//блідий для проміжних
    TextOut(Rect.Left, Rect.Top, ListBox1.Items[ARow-1]);//виводимо фразу
    TextOut(Rect.Right-15, Rect.Top,'('+inttostr(ARow)+')');//виводимо у дужках
число
  end;
end;

procedure TForm5.FormActivate(Sender: TObject);
//реалізує дії при активації форми Form5
var l, i, j, ind: Integer;   flag: Boolean;
begin
 Label1.Visible:=false;//Приховуємо інформацію про величину узгодженості
 posx:=1; posy:=1; //номери комірки в таблиці по х та у відповідно
 StringGrid1.ColWidths[0]:=120;//ширина першого стовпчика таблиці
 if (ActiveEl[1]='p') then //значить клацнули на вершині ієрархії
 begin
  label4.Caption:='Pairwise comparisons relatively "'+pnl.Caption+'"';

  for i:=1 to SavePr.n do //виводимо в таблицю nxn початкові значення з масиву
pp
   for j:=1 to SavePr.n do
    StringGrid1.Cells[i,j]:= (SavePr.pp[0,i,j]);

  for j:=1 to SavePr.n do
   StringGrid1.Cells[j,i]:= inttostr(j);
  StringGrid1.RowCount:=SavePr.n+1;//Всього рядків в таблиці
  StringGrid1.ColCount:=SavePr.n+2;//Всього стовпчиків в таблиці

  for l:=1 to SavePr.n do
   StringGrid1.Cells[0,l]:=IntToStr(l)+'. '+a[l].Caption;//виводимо в перший
стовпчик назви

критеріїв
  for l:=1 to SavePr.n do
```

```
   StringGrid1.Cells[l,0]:=inttostr(l);//виводимо номери стовпчиків у першому
рядку таблиці
 end;
 if (ActiveEl[1]='c') then //значить клацнули на одному з критеріїв
 begin
  ind := strtoint(copy(ActiveEl,3,length(ActiveEl)));//визначаємо, на якому саме
критерії був

клік
  label4.Caption:='Pairwise comparisons relatively "'+a[ind].Caption+'"';

  for i:=1 to SavePr.m do //виводимо в таблицю mxm початкові значення з масиву
pp
   for j:=1 to SavePr.m do
    StringGrid1.Cells[i,j]:= (SavePr.pp[ind,i,j]);

  StringGrid1.RowCount:=SavePr.m+1;//Всього рядків в таблиці
  StringGrid1.ColCount:=SavePr.m+2;//Всього стовпчиків в таблиці

  for l:=1 to SavePr.m do
   StringGrid1.Cells[0,l]:=IntToStr(l)+'. '+b[l].Caption;//виводимо в 1-й
стовпчик назви

альтернатив
  for l:=1 to SavePr.m do
   StringGrid1.Cells[l,0]:=inttostr(l);//виводимо номери стовпчиків у першому
рядку таблиці
 end;
 StringGrid1.Cells[StringGrid1.ColCount-1,0]:='Local priorities';//назва
останнього стовпчика
 //Визначаємо ширину та висоту таблиці, яка виводиться:
 for l:=1 to  StringGrid1.ColCount-1 do
  StringGrid1.ColWidths[l]:=35;//ширина стовпчиків попарних порівнянь
 StringGrid1.ColWidths[StringGrid1.ColCount-1]:=120;//ширина стовпчика лок.
пріоритетів
 //ширина всієї таблиці попарних порівнянь:
 StringGrid1.Width:=(StringGrid1.DefaultColWidth+1)*(stringgrid1.ColCount-

2)+2*StringGrid1.ColWidths[0]+5;
 //висота всієї таблиці попарних порівнянь:
 StringGrid1.Height:=StringGrid1.DefaultRowHeight*(StringGrid1.RowCount-

1)+StringGrid1.RowHeights[0]+20;
 //висота форми:
 Form5.Height:=  534 - 30-
(9-StringGrid1.RowCount)*StringGrid1.DefaultRowHeight;
 //центруємо таблицю по горизонталі:
 StringGrid1.Left:=  (Form5.Width -  StringGrid1.Width) div 2;
 StringGrid1SelectCell(Sender,1,1,flag);
 Button1.Click;
end;

procedure TForm5.StringGrid2SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
  var flag: Boolean; s: string;
begin
 s:='1/'+inttostr(arow); if arow=1 then   s:=inttostr(arow);
 StringGrid1.Cells[posy,posx]:= s;
 StringGrid1.Cells[posx,posy]:= inttostr(arow);
 StringGrid1SelectCell(Sender,posX,posY,flag);
 Button1.Click;
end;

procedure TForm5.StringGrid1SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);//дії з виділеною коміркою таблиці
  var Rect: TGridRect;
  dat,s1,s2:string;
begin
```

```
 StringGrid1.Options:= StringGrid1.Options-[goEditing]; //забороняємо
редагування
                                                        //елементів в таблиці
 if ActiveEl='pnl' then
  if(ACol=SavePr.n+1) then Exit;
 if ActiveEl[1]='c' then
  if (ACol=SavePr.m+1) then Exit;
 if acol=arow then //для діагональних елементів забороняємо робити зміни
                  //за допомогою лінгвістичної шкали:
 begin
  StringGrid2.Enabled:=false;//перша частина лінгвістичної шкали недоступна
  StringGrid3.Enabled:=false;//друга частина лінгвістичної шкали недоступна
 end
 else //для недіагональних елементів дозволяємо робити зміни
                  //за допомогою лінгвістичної шкали:
 begin
  StringGrid2.Enabled:=true;//перша частина лінгвістичної шкали доступна
  StringGrid3.Enabled:=true;//друга частина лінгвістичної шкали доступна
 end;

 posy:=arow; posx:=acol; rect.Left:=0;  rect.Right:=0;
 dat:=StringGrid1.Cells[acol,arow];
 if pos ('/',dat)=0 then //Обробка цілих чисел
 begin
  rect.Top:= StrToInt(dat); rect.Bottom:= StrToInt(dat);
  StringGrid2.Selection:=rect; rect.Top:=-1; rect.Bottom:=-1;
  StringGrid3.Selection:=rect;
  StringGrid2.Focused;
 end else //Обробка обернених чисел
 begin
  delete( dat,1,2);
  rect.Top:= StrToInt(dat); rect.Bottom:= StrToInt(dat);
  StringGrid3.Selection:=rect;
  StringGrid3.Focused;
  rect.Top:=-1; rect.Bottom:=-1;
  StringGrid2.Selection:=rect;
 end;
 s1:=  StringGrid1.Cells[0, ARow] ;   delete(s1,1,3);
 s2:=  StringGrid1.Cells[0, ACol] ;   delete(s2,1,3);
 label3.Caption:=' Compare "'+   s1 + '" with "'+ s2+'"';
end;

procedure TForm5.StringGrid3DrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
//Створення 2-ї частини лінгвістичної шкали коли елемент поступається іншому
begin
 if (ACol=0) and (ARow=0) then //ліва верхня комірка лінгвістичної шкали
 with StringGrid3.Canvas do
  begin
    FillRect(Rect);
    Font.Color:= clBlack;
    Font.Style:= [fsBold];
    TextOut(Rect.Left, Rect.Top, 'Element yields');
  end;
 Font.Size:=3;
 if (ACol=0) and (ARow>=1) then //комірки 1-го стовпчика лінгвістичної шкали
 with StringGrid3.Canvas do
  begin
    FillRect(Rect);
    if (ARow mod 2) = 1 then Font.Color:= clBlack  //нормальний колір для
основних рівнів
                        else Font.Color:= clGrayText; //блідий колір для
проміжних рівнів
    TextOut(Rect.Left, Rect.Top, ListBox2.Items[ARow-1]);
    if (ACol=0) and (ARow=1) then TextOut(Rect.Right-25,
Rect.Top,'('+inttostr(ARow)+')') else
       TextOut(Rect.Right-25, Rect.Top,'(1/'+inttostr(ARow)+')');
```

```
  end;
end;

procedure TForm5.StringGrid3SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
    //Обробка натискання комірки з другої частини лінгвістичної шкали
    var flag: Boolean; s: string;
begin
 s:='1/'+inttostr(arow); if arow=1 then   s:=inttostr(arow);
 StringGrid1.Cells[posx,posy]:= s;//Ставимо обернене число
 StringGrid1.Cells[posy,posx]:= IntToStr(arow);//Ставимо ціле число в симетричну
комірку
 StringGrid1SelectCell(Sender,posX,posY,flag);//Виділяємо комірку
 Button1.Click;
end;

procedure TForm5.StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
  //Малюємо таблицю попарних порівнянь і записуємо дані в комірки таблиці
begin
 with StringGrid1.Canvas do
  begin
    FillRect(Rect);
    if (ARow <> Acol)  then Font.Color:= clBlack
                       else Font.Color:= clGrayText;//Блідий колір для діагоналі
    TextOut(Rect.Left+3, Rect.Top, StringGrid1.Cells[acol,arow]);
  end;
end;

procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);
var i,j,ind:integer;
//Зберігаємо введені в таблицю дані (в масиві pp) під час закриття форми
begin
 if (ActiveEl[1]='p') then //Якщо оброблялася матриця відносно мети
   for i:=1 to SavePr.n do
    for j:=1 to SavePr.n do
     SavePr.pp[0,i,j]:= (StringGrid1.Cells[i,j]);
 if (ActiveEl[1]='c') then //Якщо оброблялася матриця відносно одного з
критеріїв
  begin
   ind:= StrToInt(ActiveEl[3]);//ind - Який саме критерій (його номер)
   for i:=1 to SavePr.m do
    for j:=1 to SavePr.m do
     SavePr.pp[ind,i,j] :=StringGrid1.Cells[i,j];
  end;
  chengeFlag:=True;
end;

procedure TForm5.Button1Click(Sender: TObject);
var r, rs: array[1..10] of real; t:String;
    l, i, ai: integer;
    rt, sum ,x ,zm: real;
 {ai - визначає, з якою матрицею ми працюємо:
      0 - з головною, i-й - з матрицею i-го критерію
  масив r[i] - це корені певної степені з добутку елементів рядка),
  sum - це сума цих коренів (на неї треба поділити кожен корінь)
  масив rs[i] - обчислює суму елементів по кожному стовпчику матриці
  x - величина узгодженості}
begin
 if pos('pn',ActiveEl)>0 then  ai:=0 else ai:=StrToInt(ActiveEl[3]);
 sum:=0;
//Заповнюємо матрицю даними:
 for i:=1 to stringgrid1.ColCount-2 do
 begin
  r[i]:=1;
  for l:=1 to stringgrid1.RowCount-1 do
  begin
    t:=stringgrid1.cells[l,i];
```

```
    if pos('/',t)>0 then
    begin delete(t,1,2); rt:=1/StrToInt(t); end else rt:=StrToint(t);
    r[i]:=r[i]*rt;//це добуток всіх елементів рядка
  end;
 r[i]:=power(r[i],1/(stringgrid1.RowCount-1));//це вже корінь певної степені з
добутку
 sum:=sum+r[i];//це сума всіх коренів певної степені з добутку елементів рядка)
 end;
 for i:=1 to stringgrid1.ColCount-2 do
 begin
  rs[i]:=0;
  for l:=1 to stringgrid1.RowCount-1 do
  begin
   t:=stringgrid1.cells[i,l];
   if pos('/',t)>0 then
                     begin
                       delete(t,1,2);
                       rt:=1/StrToInt(t);
                     end
                   else rt:=StrToint(t);
   rs[i]:=rs[i]+rt;//це сума чисел по кожному стовпчику матриці
  end;
 end;
//Виводимо локальні пріоритети:
 zm:=1000;
 for i:=1 to stringgrid1.ColCount-2 do
 begin
  SavePr.pn[ai,i]:=r[i]/sum;//це локальні пріоритети; зберігаємо їх в масиві pn
  stringgrid1.Cells[stringgrid1.ColCount-1,i]:=RealToStr(r[i]/sum,2);
 if  abs(SavePr.pn[ai,i]-1/SavePr.m)<zm then
                                       begin
                                         CA[ai]:=i;
                                         zm:=abs(SavePr.pn[ai,i]-1/SavePr.m);
                                       end;
 end;
//Виводимо величину узгодженості:
 for i:=1 to stringgrid1.ColCount-2 do x:=x+rs[i]*r[i]/sum;
 x:=( (x- ( stringgrid1.ColCount-2)) /  ( stringgrid1.ColCount-2) )/
Vu[stringgrid1.ColCount-2];
 Label1.Visible:=true;
 label1.Caption:='Consistency of data is '+RealToStr(x*100,2)+ ' %';
 if x>0.2 then Label2.Visible:=true else Label2.Visible:=false;
end;

end.
```

Unit6.pas

```
unit Unit6;
{1. розраховує глобальні пріоритети для кожної альтернативи (масив rs)
 2. запам'ятовує номер альтернативи, яка виграла (у змінній bestA)
 3. малює гістограму
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls,unit1,unit5,Math, ExtCtrls, TeeProcs, TeEngine, Chart,
  Series;

type
  TForm6 = class(TForm)
    Memo1: TMemo;
    Panel1: TPanel;
    Timer1: TTimer;
    Panel2: TPanel;
    Label1: TLabel;
    Bevel1: TBevel;
    Panel3: TPanel;
    ComboBox1: TComboBox;
    Button1: TButton;
    Label2: TLabel;
    Chart1: TChart;
    Series1: TBarSeries;
    procedure FormActivate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;
     c: array[0..11] of TImage;
     chmax: real;
     ch: array[0..8] of integer;
implementation

uses Unit7;

{$R *.dfm}

Function RealToStr(r: Real; p: byte): String;
{Конвертирует Real в строку, p - символов в дробной части}
begin
 Str(r:30:p,Result); // конвертация real - string
 Result:=Trim(Result); // отбрасывание пробелов слева и справа
end;

procedure TForm6.FormActivate(Sender: TObject);
var l, i: integer;
    sum: real;
begin
 combobox1.Clear;
 for l:=1 to SavePr.m do combobox1.Items.Add(b[l].Caption);
 Chart1.Series[0].Clear;
 memo1.Clear;
 sum:=0;    bestA:=1;    chmax:=0;
 for i:=1 to SavePr.m do
 begin
  rs[i]:=0;
  for l:=1 to SavePr.n do rs[i]:=rs[i]+SavePr.pn[0,l]*SavePr.pn[l,i];
 end;
 for i:=1 to SavePr.m do sum:=sum+rs[i];
```

```
 for i:=1 to SavePr.m do
 begin
  rs[i]:=rs[i]/sum;//це глобальні пріоритети альтернатив
  if rs[bestA]/sum<rs[i]/sum then   bestA:=i;
  memo1.Lines.Add(b[i].caption+'  '+RealToStr(rs[i]/sum,2));

Chart1.Series[0].Add((rs[i]/sum),b[i].caption+chr(13)+'('+RealToStr(rs[i]/sum,2)
+')'{,rgb

(122,122,122)});
  if (rs[i]/sum)>chmax then chmax:=rs[i]/sum;
  for l:=0 to 8 do
    if not ((rs[l+1]/sum)=chmax) then ch[l]:=0
    else ch[l]:=1;
 end;
 Timer1.Enabled:=true;
end;

procedure TForm6.Timer1Timer(Sender: TObject);
var rs:array[0..10] of real;
    l, i: Integer;
    sum: real;
begin
 Timer1.Enabled:=false;
 chart1.MarginTop:=1;
 chart1.SeriesList.Series[0].CalcXPos(0);
 For i:=1 to 9 do
 if(c[i]=nil) then c[i]:=TImage.Create(Panel1);
 For i:=1 to SavePr.m do
 begin
  c[i].Parent:=Panel1;
  c[i].Name:='img'+IntToStr(i);
  c[i].Proportional:=true; ;
  c[i].Width:=100;
  c[i].Height:=70;
  c[i].Top:=10;
  if Length(SavePr.img[i])=0 then c[i].Visible:=False else c[i].Visible:=True;
  chart1.SeriesList.Series[0].CalcXPos(0);
  c[i].Left:=chart1.SeriesList.Series[0].CalcXPos(i-1);
  if (length(SavePr.img[i])>4) then
   c[i].Picture.LoadFromFile( ExtractFilePath(Application.ExeName)+ 'Images\'+
  ExtractFileName(SavePr.img[i]));
  c[i].Center:=true;
 end;
 For i:=SavePr.m+1 to 9 do   c[i].Visible:=false;
end;

procedure TForm6.Button1Click(Sender: TObject);
begin
 Form7.Show;//виклик форми7 після натискання на кнопку "Create recommendation"
end;

procedure TForm6.ComboBox1Change(Sender: TObject);
//робимо кнопку "Create recommendation" недоступною для альтернативи, яка
виграла
var
  l: integer;
begin
  if ch[ComboBox1.ItemIndex]=1 then button1.Enabled:= False
  else button1.Enabled:= True;
end;

end.
```