

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ **Ю. П. Кондратенко**
«___» _____ 2022 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СИСТЕМА ОЦІНЮВАННЯ ЗВО НА ОСНОВІ НЕЧІТКОГО
ГІБРИДНОГО МЕТОДУ ПРИЙНЯТТЯ РІШЕНЬ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401. 21810103

Виконав студент 4-го курсу, групи 401
_____ **Я.Ю. Белоусова**
«___» _____ 202_ р.

Керівник: д-р техн. наук, проф.
_____ **Ю.П. Кондратенко**
«___» _____ 202_ р.

Миколаїв – 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інформаційних технологій і програмних систем

Освітньо-кваліфікаційний рівень **бакалавр**

Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____Ю. П. Кондратенко
«__» _____ 2021 р.

ЗАВДАННЯ

на виконання дипломної роботи

Видано студенту групи 401 факультету комп'ютерних наук Белоусова Яна
Юріївна

1. Тема кваліфікаційної роботи «Система оцінювання ЗВО на основі
нечіткого гібридного методу прийняття рішень»

Керівник роботи Кондратенко Юрій Пантелійович, д-р техн. наук, професор.

Затв. наказом Ректора по ЧНУ ім. Петра Могили від «__» _____ 2021 р. №

2. Строк представлення кваліфікаційної роботи «__» _____ 2022 р.

3. Вхідні (початкові) дані до роботи: оцінки ЗВО від експертів

Очікуваний результат: розроблена система оцінювання ЗВО на основі нечіткого
гібридного методу прийняття рішень

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки)

- аналіз предметної сфери, постановка задачі;
- аналіз та вибір інструментальних засобів розробки системи;

- порівняльний аналіз результатів застосування обраного методу прийняття рішень для розв’язання поставленої задачі.

5. Перелік графічних матеріалів: презентація.

6. Завдання до спеціальної частини: «Охорона праці та безпека у надзвичайних ситуаціях» складається з одного розділу.

Необхідно проаналізувати стан охорони праці на робочому місці та у кімнаті під час виконання роботи.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Ст. викладач О.В. Макарова	

Керівник роботи __д-р техн. наук, проф. Кондратенко Ю. П._____
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання __Бєлоусова Я.Ю._____
(прізвище та ініціали)

(підпис)

Дата видачі завдання «_23_» _____ листопада ____ 2021 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Створення системи оцінювання ЗВО на основі нечіткого гібридного методу прийняття рішень.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	26.10.21	30.10.21	Виконано
2	Отримання завдання на виконання БКР	24.11.21	24.11.21	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	08.12.21	09.12.21	Виконано
4	Отримання завдання на переддипломну практику	20.05.22	21.05.22	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	23.05.22	04.06.22	Виконано
6	Створення застосунку	20.04.22	13.05.22	Виконано
7	Оформлення звіту з БКР	25.04.22	13.05.22	Виконано
8	Попередній захист БКР на засіданні комісії кафедри	30.05.22	31.05.22	Виконано
9	Доробка та остаточне оформлення БКР	02.06.22	20.06.22	Виконано
10	Подання БКР рецензенту	14.06.22	18.06.22	Виконано
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	20.06.22	22.06.22	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	27.06.22	29.06.22	Виконано

Розробив студент Бєлоусова Я.Ю. _____
(прізвище та ініціали) (підпис)

Керівник роботи д-р техн. наук, проф. Кондратенко Ю. П. _____
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

« ____ » _____ 202__ р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи
студента групи 401 ЧНУ ім. Петра Могили
Белоусова Яна Юріївна
«Система оцінювання ЗВО на основі нечіткого гібридного методу
прийняття рішень»

Мета: підвищення ефективності оцінювання ЗВО для допомоги вибору вступу абітурієнтам.

Об'єкт: процеси оцінювання ЗВО.

Предмет: методи прийняття рішень та технології рейтингування закладів вищої освіти.

Пояснювальна записка бакалаврської дипломної роботи складається з вступу, чотирьох розділів, висновків та додатків.

У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об'єкт дослідження.

У першому розділі описується існуючі системи оцінювання. Також, аналізується існуючі системи оцінювання.

У другому розділі роботи представлено різні методи прийняття рішень та їх аналіз між собою.

У третьому розділі описуються основні скрипти та програми, що допомогли у створенні системи оцінювання ЗВО на основі нечіткого гібридного методу прийняття рішень.

У четвертому розділі проводиться тестування розробленого програмного забезпечення, аналіз роботи та отриманих даних. Представлені скріни інтерфейсу застосунку та його результатів роботи.

У п'ятому розділі описується аналіз умов праці, безпеки праці, гігієни праці та виробничої гігієни. Також описані вимоги освітлення і мікроклімату.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота містить 72 стор. основної частини, 4 таблиці, 28 рисунки, 2 додатки, 25 джерел.

Ключові слова: системи оцінювання, методи прийняття рішень, ЗВО, нечіткий метод, TOPSIS, АНР.

ABSTRACT

of the Bachelor's Thesis

«Free evaluation system based on fuzzy hybrid decision-making method»

Student: Bielousova Yana

Objective: to increase the effectiveness of the evaluation of free economic zones to help applicants choose the entrance.

Object: free assessment processes.

Subject: decision-making methods and rating technologies of higher education institutions.

The explanatory note of the bachelor's thesis consists of an introduction, four chapters, conclusions and appendices.

The introduction determines the relevance of the topic to be taken as a goal and a brief overview of the task, the subject of research and the object of research.

The first section describes existing assessment systems. Also, existing evaluation systems are analyzed

The second section of the thesis presents various methods of decision making.

The third section describes the main scripts and programs that have helped to create a free evaluation system based on a fuzzy hybrid decision-making method.

The fourth section tests the developed software, analyzes the work and the obtained data. Staples of the application interface and its results are presented.

The fifth section describes the analysis of working conditions, occupational safety, occupational health and occupational hygiene. The requirements of fire safety and lighting and microclimate are also described. In conclusions, an analysis of work and results were carried out.

Qualification work contain 72p. The main part, 4 tables, 28 figures, 2 appendix, 25 sources.

Key words: evaluation systems, decision-making methods, ZVO, fuzzy method, TOPSIS, AHP.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ.....	7
1.1 Опис проблеми вибору ЗВО	7
1.2 Огляд та аналіз наявних аналогів та публікацій.....	10
1.3 Постановка задачі	21
Висновки до розділу 1	21
2 МЕТОДИ, АЛГОРИТМИ ОЦІНЮВАННЯ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ	22
2.1 Гібридні методи прийняття рішень	22
2.2 АНР і нечіткий АНР	23
2.3 TOPSIS і нечіткий TOPSIS	28
Висновки до розділу 2	31
3 АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ СИСТЕМИ	32
3.1 Аналіз варіантів інструментальних засобів для розробки системи	32
3.2 Платформа .NET та мова програмування C#	36
3.3 Використання запитів MySQL при реалізації додавання, видалення, пошуку та оновлення даних полів в таблицях	39
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ	41
4.1 Опис програмної реалізації.....	42
4.2 Огляд користувацького інтерфейсу	44
Висновки до розділу 4	55
5 ОХОРОНА ПРАЦІ	57
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТОК А.....	71
ДОДАТОК Б	74

ПЕРЕЛІК СКОРОЧЕНЬ

ЗВО – заклади вищої освіти

СКМ – Система Кепітал Менеджмент

БФРУ – Благодійний фонд «Розвиток України»

АНР – Процес аналітичної ієрархії

TOPSIS – Техніка визначення порядку переваги за схожістю з ідеальним рішенням.

Пояснювальна записка

до кваліфікаційної роботи

на тему:

«СИСТЕМА ОЦІНЮВАННЯ ЗВО НА ОСНОВІ НЕЧІТКОГО ГІБРИДНОГО МЕТОДУ ПРИЙНЯТТЯ РІШЕНЬ»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810103

Виконав студент 4-го курсу, групи 401

_____ **Я.Ю. Бєлоусова**

«___» _____ 202_ р.

Керівник: д-р техн. наук, проф.

_____ **Ю.П. Кондратенко**

«___» _____ 202_ р.

Миколаїв – 2022

ВСТУП

Актуальність теми. Вигадати собі професію в 16-17 років – це досить серйозний виклик, з яким не всі можуть впоратися. Більшість школярів 16-18 років у своєму віці не знаю, куди вони хочуть вступати та ким бути у майбутньому. Вони завжди вивчають відгуки знайомих, друзів, тих чи інших закладів, гадаючи, куди подавати документи.

Враховуючи, що навчання є полем для додаткового вдосконалення політичного, грошово-кредитного, соціального та послідовного рівня суспільства, характер інструктивного управління є вирішальним елементом для розвитку суспільної економіки в цілому. Оскільки інструктивні адміністрації, як правило, забезпечують вищі навчальні заклади, визначення посади є методом перевірки характеру їхніх управлінь.

Однак освіта потрібна. Це дозволяє людині перейти на інший, більш значний рівень вдосконалення, відкриває дивовижні двері, допомагає вийти на нові сфери діяльності. Якісне навчання – це гарантія освіченості людини. Проте у вищій освіті головне не диплом, а отримані знання.

ЗВО допоможе вам вибрати покликання, яке буде популярним і над яким неможливо домінувати поодиночі. Підготовка допоможе вам визначити свої межі. Вища освіта дозволяє не тільки отримати спеціальність або, як зараз всі воліють казати, скоринку. Поглиблена освіта дає можливість сформуванню справді необхідні здібності в повсякденному житті: розуміння дедлайнів та важливості їх проходження.

Наша країна — член Гаазької конвенції про спрощену процедуру легалізації документів. Україна, поряд із 43 різними державами, є учасником Лісабонської конвенції про визнання кваліфікацій щодо вищої освіти, яка поширюється практично на всі європейські країни. Україна посідає 10-12 місце у світі за рівнем охоплення населення вищою освітою.

Мета: підвищення ефективності оцінювання ЗВО для допомоги вибору

вступу абітурієнтам.

Предмет: методи прийняття рішень та технології рейтингування закладів вищої освіти.

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

- проаналізувати сучасний стан задачі оцінювання ЗВО та аналоги систем;
- виокремити основні проблеми та запропонувати їх рішення;
- здійснити збір інформації;
- розробити програмне забезпечення.

Практичне значення: модуль призначений для використання ЗВО та абітурієнтам. Розроблений модуль здійснює збір інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Опис проблеми вибору ЗВО

Вибір покликання для підлітка є клопітною справою. Значна частина з них, обираючи покликання, залежить не від своїх прагнень і нахилів, а від умов життя: оцінки людей, доходів, професійного розвитку, слави. Переслідуючи оцінки інших, індивіди нехтують головними складовими: своїми потягами, інтересами, вдачею. Недолік цих змінних часто змушує далеко від найкращих результатів. Потрапляючи прямо в робочий простір, молодий експерт розуміє, що вибрав якесь неприйнятне поле руху, якесь неприйнятне покликання. Розуміє, що це «не його».

Крім того, перед ним стоїть рішення або віддати життя нелюбимій причині, або повернутися додому. Очевидно, це, безумовно, не просте рішення, особливо для «попереднього дублера». Щоб свідомо вибрати правильний напрямок майбутнього руху, важливо серйозніше поглянути на питання професійного спрямування, дуже багато читати зі школи.

Рішення про покликання є, мабуть, найважчим рішенням, яке стоїть перед людиною, оскільки всі усвідомлюють, що людина проводить більшість свого часу на землі, працюючи. Нині людина не може існувати без праці, з точки зору матеріальної, але, крім того, з точки зору етики. Чим більше працює індивід, тим більше він відчуває свою цінність, відчуває себе частиною суспільства, частиною широкої публіки, в якій вона існує. Але це також людська потреба, на яку не можна просто заплющити очі чи нехтувати нею. Я міг би погодитися з тим, що людина буде працювати постійно, відчуваючи постійну потребу в роботі, і недарма люди кажуть: «Сталь гартується у вогні — а людина в праці».

Для певних людей праця є життєвим значенням. Робота освітлює життя людини, робить можливим постійне отримання нових корисних знань. Очевидно, що відпочинок також важливий для людини, однак найбільш значущим і повним є відпочинок, який настає після того, як людина справді

пригнічується. Людина швидко відчуває задоволення від того, що день не був безкінечним і що він придумав, як зробити щось величезне. Так само, неодмінна частина робочого дня – обмін інформацією, думками, досвідом, покращення власних якостей. Спрацьовує те, що станеться експерт і, якоюсь мірою, самосвідомість особистості.

Таким чином, ми можемо припустити, що покликання та праця людини значною мірою вписують його власні характеристики, а рішення про покликання слід розглядати виключно серйозно.

Сучасні молоді люди отримують мало даних про покликання в школі, але, як свідчать результати огляду, вони дуже зацікавлені в створенні членами сім'ї та товаришами. Навряд чи когось цікавить, яка їх ідентичність. Майже у всіх є явні залежності.

Навряд чи якась школа пропонує позакласні вправи, крім предметних олімпіад. Щоб перетворитися на особу з дитячого театру, ансамблю, спортивного сегмента, потрібно шукати гуртки та студії деінде, а для цього дуже потрібен розумний вибір малюка чи допомога опікунів. В огляді не зовсім 50% школярів кажуть, що ходять у гуртки, пов'язані з покликанням та уявою.

Більшість активно вибрали, де їм потрібно навчатися, і вони усвідомлюють, з яких предметів для цього потрібно планувати. Як би там не було, люди фактично не довели практику створення обкладинок, і мало хто з них знає, що таке звичайний день на майбутній роботі.

Критична міра сприймає, що нові думки та починання можуть відвести їх від старих. Це означає, що вони не знають про своє рішення і розуміють, що інтригуючи дані про інше покликання можуть змусити їх змінити свою точку зору.

Багато в школі активно сприйняли те, з чим їм потрібно працювати. Але що робити, якщо в нинішніх умовах отримати ту спеціальність, яку ви прагнете, важко? Комусь заважають недостатньо високі оцінки, хтось змушений працювати після 10 класу. Не кожна людина має цінний шанс оплатити свої

обстеження. Такі випускники намагаються потрапити до школи, де вони можуть приєднатися до роботи та занять.

Щоб ваше бажання здійснилося, вам дійсно хочеться працювати і просуватися настільки, наскільки можна було очікувати. Практика спрямування професії розширить ваші перспективи та змінить вашу точку зору на речі, які нещодавно здавалися очевидними. Хтось зрозуміє, що шлях до вищої освіти лежить не тільки через золоту прикрасу в 11-му класі. Інші відкинуть цю думку, оскільки знайти роботу після коледжу цілком можна.

Часто школярі думають про конфірмацію як про перепустку до дарів життя. Сьогодні це не так. Не всі випускники коледжу можуть самостійно отримати нову професію, а основи факультативного навчання, які суперечать нормі, часто допомагають підприємствам, допомагаючи швидко отримати нове напрямки роботи. Крім того, численні галузі експертів швидко розвиваються, і, можливо, покликання фантазії поки що не існує, але через 3-5 років воно з'явиться. Відповідно, важливо постійно вчитися, створювати та слідувати поточному плану. Немає миттєвих рецептів для вправних досягнень; це для кожного різне.

Вибираючи покликання, слід враховувати супутні змінні: інтереси (розум, досвідченість, інтерес до покликання, схильності), здібності (як розумові компоненти, необхідні для результату в певному виді руху), настроїв, характер, рівень підготовки. (шкільне виконання), стан благополуччя, свідомість всесвіту покликань, соціальний клімат, домашні обставини, повчальний ступінь опікунів та багато інших.

Варто обережно розглянути свої бажання і точно визначити межі. Справді, навіть найкращий кмітливий Конфуцій сказав: «Виберіть завдання, яке вам подобається, і вам не доведеться працювати один день у вашому житті». Однак не можна погодитися з незрівнянним філософом. Якщо людина любить свою роботу, а не нехтує нею, то за умови, що вона цінує кожен мить, її ефективність буде набагато вищою, і це тільки допоможе підприємству.

Очевидно, що асоціація може переконати своїх представників значними винагородами та різними винагородами, але найкращим натхненням є її власне бажання безкінечно функціонувати чудово. Тоді, в цей момент, характер роботи буде вищим, і представник буде рухатися до поставлених завдань все більш інноваційно, представляючи справді нове, щось своє, і, можливо, це принесе успіх організації.

Завдяки особам, які точно вибрали своє покликання, було зроблено велику кількість головних одкровень у багатьох сферах діяльності, а також були зроблені останні досягнення даних. Саме завдяки сталості, нелегкій праці та інтересу до свого покликання стало мислимо боротися з найвидатнішими та найризикованішими хворобами.

1.2 Огляд та аналіз наявних аналогів та публікацій

В Україні працюють сотні вищих навчальних закладів. Але далеко не в кожному з них можна здобути якісну освіту.

Випускники шкіл, які хочуть продовжити навчання, стають перед вибором ЗВО. Насамперед, необхідно визначитися зі спеціальністю. Допоможе в цьому перелік спеціальностей та напрямків, за якими готують спеціалістів у ЗВО України.

Другий крок має на увазі вибір ЗВО, адже одну і ту ж спеціальність можна отримати в декількох ЗВО навіть в одному місті. Отоді й приходять на допомогу рейтинги ЗВО України. Рейтинг - числовий чи порядковий показник, що відображає важливість чи значущість певного об'єкта чи явища. Через різні критерії рейтинги ЗВО можуть відрізнятися один від одного, творці рейтингів застосовують різні методики оцінювання.

В Україні існує кілька рейтингів ЗВО, які оцінюють різні організації. Найбільш поширені рейтинг ЗВО України "Топ-200", розроблений Організацією Об'єднаних Націй з питань освіти, науки та культури (ЮНЕСКО), рейтинг "Компас" та рейтинг журналу "Гроші".

Також дослідження рейтингів проводяться серед ЗВО якогось одного напрямку. Наприклад, рейтинг ЗВО, які навчають фахівців у ІТ-сфері.

Попри різну методику використовувану упорядниками щодо рейтингів, на перших позиціях зазвичай одні й самі ЗВО.

Питання оцінки рейтингу, а саме стратегії їх виконання, обговорюється як українськими, так і малознайомими дослідниками, серед яких, наприклад, Ван Дайк Н., Вітлінський В., Джоббінс Д. та інші [1].

Рейтинг ЗВО є ознакою прогресу освіти за правилами, які характеризують процедуру, що позиціонує університет як серйозний у порівнянні з іншими. Під конкурентоспроможністю ЗВО маємо на увазі зацікавленість як абітурієнтів, так і роботодавців, щоб відшукати представників серед випускників цієї вільної валютної зони.

Далі, щоб сформувати важливість рейтингових рамок серед випускників факультативних шкіл як основних споживачів, які згодом стануть абітурієнтами ЗВО, було проведено статистичне спостереження. Результат висвітлюємо за допомогою рис. 1.

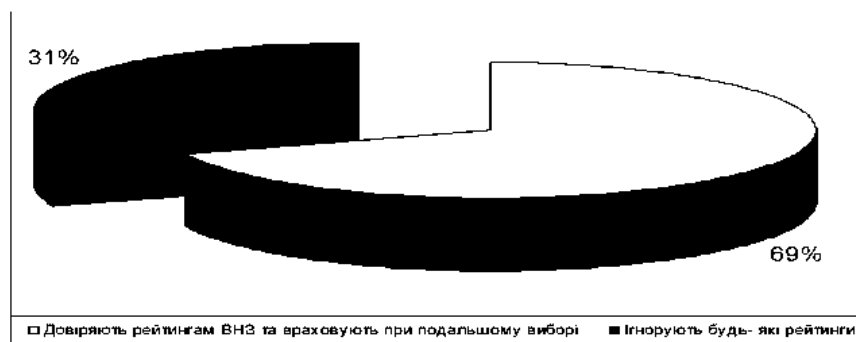


Рисунок 1.1 – Співвідношення довіри до рейтингів ЗВО серед випускників загальноосвітніх шкіл, %

Дослідження фактичних уявлень показує, що місця розташування якоїсь вільної фінансової зони є життєво важливими для абітурієнта і здебільшого структурують думку про престижність ЗВО, що є одним із центральних питань, у світлі того, що більшість випускників визнають, що престижне ЗВО забезпечить безкоштовне грошове навчання майбутнє, тобто критичні ситуації в

позиціонуванні, сприяють рекламі ЗВО. [2]

Серед українських рейтингів, найбільш популярними є:

- топ - 200 Україна;
- рейтинг «Середній бал ЗНО на контракт»;
- рейтинг ЗВО за показниками Scopus.

Методика проекту «Топ-200 Україна» залежить від використання інформації з прямих оцінок та висококваліфікованих оцінок і за час своєї присутності дещо адаптувалася до змін у системі передової освіти країни. Як би там не було, загальне розташування маркерів для різного роду закладів залишалося незмінною передумовою процедури, що дозволяло поглянути на наслідки їх вправ. Стратегія розроблена відповідно до Берлінських принципів рейтингу університетів.

На другій конференції Міжнародної експертної групи з рейтингу (IREG) у Берліні «Стратегія та стандарти якості рейтингів», координується Центром розвитку вищої освіти (CNE) (Німеччина), Інститутом політики вищої освіти (США), SEPES Члени ЮНЕСКО (Бухарест, Румунія) з 19 країн підтримували Берлінські принципи рейтингу вищих навчальних закладів. Центр міжнародних проектів «Євроосвіта» відзначив і поділяє ці стандарти.

Розміщення закладів та інструктивних проектів – це світова особливість. Вони задовольняють певні потреби: вирішення запитів споживачів щодо доступу до очевидно введених даних про ситуацію з організаціями вищої освіти; стимулюють суперництво між ними; гарантувати правомірність несення субсидування; допомагають організувати різного роду навчальні основи, навчальні плани та тренінги. Так само, якщо їх належним чином досягнути та розшифрувати, вони додають значення «цінності» передової освіти в їхніх країнах, доповнюючи багаторівневі вправи громадських та вільних ліцензованих асоціацій у сфері оцінювання та оцінювання природи освіти. Це причина, чому рейтинги розширеної освіти перетворилися на необхідну частину громадського контролю якості та циклів підтвердження, і чому дедалі зростаюча кількість

країн сприятиме рейтингу пізніше. Враховуючи цю закономірність, важливо, щоб рейтингові асоціації відповідали за характер своєї інформації, техніку та розповсюдження даних.

Наслідком цього прагнення стало вдосконалення структури для створення та поширення публічних, територіальних та всесвітніх рейтингів, що здебільшого спонукатиме до організації постійного вдосконалення та уточнення стратегій, які використовуються при веденні таких рейтингів. Враховуючи різноманітність використовуваних методів, прийняті стандарти позиціонування будуть цінними для роботи над позиціонуванням та дослідження характеру навчання.

Рейтинг українських ЗВО «Компас» був ініційований Компанією «Систем Кепітал Менеджмент» (СКМ) та Благодійним фондом Ріната Ахметова «Розвиток України» (БФРУ) у рамках програми «Сучасна освіта». Ідея проекту також знайшла підтримку з боку Світового банку (World Bank) та інших українських та міжнародних організацій.

Рейтинг ЗВО «Компас» став першим загальнонаціональним рейтингом, що відображає оцінку якості освіти випускниками та потенційними роботодавцями.

База даних Scopus постійно зберігає понад 20 000 логічних розподілів у спеціалізованих, клінічних та гуманітарних науках. Набір даних додатково включає розповсюдження логічних щоденників, процедур зустрічей та книг.

У таблиці позиціонування українські заклади позиціонуються за індексом Гірша - кількісним показником, що ґрунтується на кількості наукових публікацій та кількості їх цитувань.

В основу ранжування взято індекс Хірша (h-index). Він показує дані щодо співвідношення кількості опублікованих наукових праць та кількості цитування цих праць іншими вченими.

Проект «ТОП - 200 Україна» є найбільш популярним серед випускників. Діяльність ЗВО визначалася з допомогою загального індексу рейтингової оцінки — I_3 . [2] Цей індекс є інтегральним і визначається трьома комплексними

критеріями (індексами):

$$I_3 = I_{\text{нп}} + I_{\text{н}} + I_{\text{мв}},$$

де $I_{\text{нп}}$ — індекс якості науково-педагогічного потенціалу, значення якого змінюються в діапазоні [0—50%];

$I_{\text{н}}$ — індекс якості навчання, який змінюється в діапазоні [0—30%];

$I_{\text{мв}}$ — індекс міжнародного визнання, змінюється в діапазоні [0—20%].

Кожен з цих комплексних критеріїв визначається групою індикаторів (табл. 1). Вагові коефіцієнти визначалися групою фахівців високого рівня в галузі науки та освіти із застосуванням методу експертного оцінювання.

Таблиця 1.1 - Структура критеріїв рейтингової системи ТОП - 200 України

Назва критерію	Індикатори критерію
Якість науково педагогічного потенціалу	– кількість штатних представників, обраних академіками НАН України, – кількість штатних представників, обраних відповідними особами НАН України, – кількість викладачів серед працівників ЗВО – кількість академічних керівників у штаті ЗВО. держава – кількість спеціалістів науки у штаті ЗВО, – кількість здобувачів наук серед штатних представників ЗВО. – кількість штатних представників, нагороджених премією в галузі науки та інновацій або Державною премією. Т. Шевченка кількість дублерів, переможців та призерів всесвітніх олімпіад (суперників),
Якість навчання	– кількість дублерів, чемпіонів і призерів всесвітніх

Назва критерію	Індикатори критерію
	змагань (викликів), – кількість дублерів, чемпіонів та призерів кожної української олімпіади (суперників), – відношення кількості начальників до кількості самотніх чоловіків і підготовлених фахівців, розміру вільних валютних зон.
Міжнародне визнання	– кількість іноземних студентів, – зарахування інструктивної організації до Європейської асоціації університетів, – зарахування інструктивної організації до Великої хартії університетів, – вступ до Євразійської асоціації університетів, – зарахування інструктивної організації в організацію коледжів Чорноморського округу.

Порівнюючи діяльність різних ЗВО України, усі вказівники та показники, а також загальний запис оцінки рейтингу зведені до стандартизованої структури, щоб вони змінювалися в межах охоплення [0-100]

Як видно з рис. 2, керівниками рейтингу є національний університет імені Тараса Шевченка та НТУУ «Київський політехнічний інститут», чиї рейтингові рекорди помірно на схожому рівні, що може свідчити про їх протилежність один одному. Також, розбиваючи оцінки безкоштовних фінансових зон та 2018 року, бачимо, що місце НТУУ «Київський політехнічний інститут» дещо зросло до 84,79, хоча позиціонування зменшилося з першого місця у 2017 році (77,02) до другого місця у 2019 році (80 , 9), тоді як КНУ ім. Тарас Шевченко піднявся з другої ситуації у 2018 році до першої. Третє та четверте місця рейтингу залишаються незмінними протягом 2017-2018 років. У 2017 та 2018 роках

Національний медичний університет імені О.О. Богомольця, тоді як у 2019 році НТУ «Харківський політехнічний інститут» посів п'яте місце, а НМУ імені О.О. Богомольця відмовився від своїх позицій і перейшов на 6 позицію.

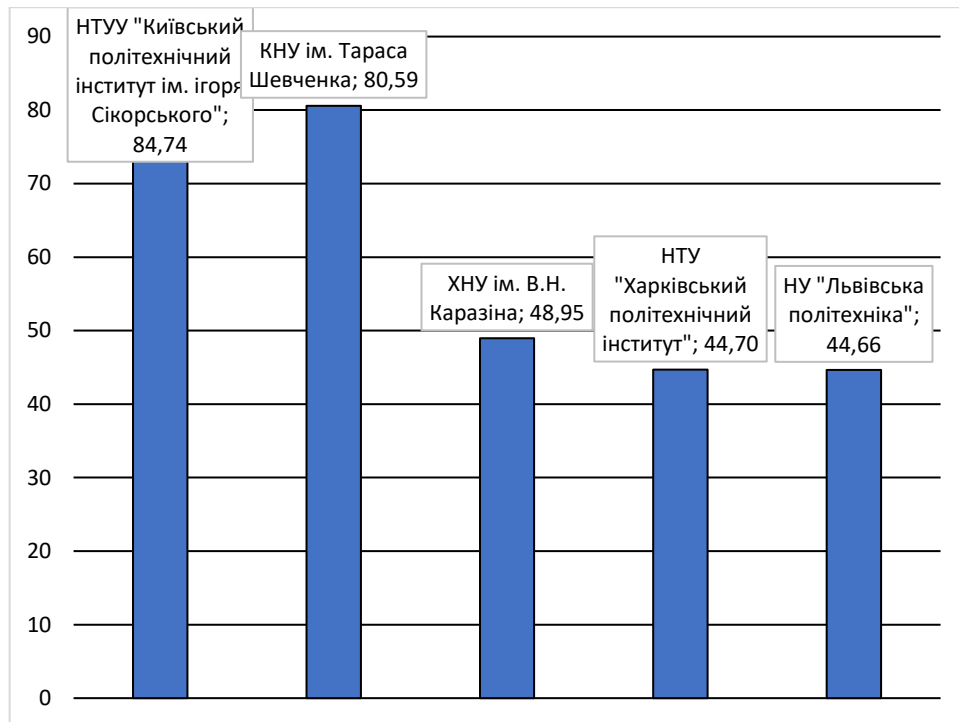


Рисунок 1.2 – Рейтинг ЗВО України в 2018 році «ТОП-200 Україна» відповідно до інтегрального показника діяльності

КНУ ім. Тараса Шевченка має найвищий серед українських ЗВО рейтинг цитування своїх наукових праць за системою Scopus, високий рівень затребуваності його випускників ринком праці, особливо в секторах державного управління, юриспруденції, економіки і політики.

КПІ ім. Ігоря Сікорського став головним навчально-науковим центром країни у сфері інженерії і високих технологій. Він продовжує розвивати українську «Кремнієву долину», яка стала відомою в Україні і світі як інноваційна екосистема «Sikorsky Challenge».

Високі показники своєї діяльності демонструє також ХНУ ім. В.Н. Каразіна. Він стабільно протягом останніх років обіймає третє місце в національному академічному рейтингу «Топ 200 Україна».

У 2018 році у першій десятці лідерів зберегли своє перебування Національний технічний університет "Харківський політехнічний інститут" (4 місце), Національний університет "Львівська політехніка" (5 місце), Національний університет біоресурсів і природокористування України (6 місце), Національний медичний університет імені О.О. Богомольця (7 місце), Національний технічний університет «Дніпровська політехніка» (8 місце), Львівський національний університет імені Івана Франка (10 місце).

Рейтинг України «Компас» ведеться з використанням методик індивідуальних і телефонних інтерв'ю, вивчення електронної пошти та онлайн-оглядів. Ця методика експертизи була обрана для охоплення як численних делегатів різних вузів, так і важкодоступних респондентів (агентів керівників, спеціалістів).

Консультації з підприємствами проводилися в Інтернеті та, до певної міри, електронною поштою. SurveyMonkey.com був використаний для веб-огляду.



Рисунок 1.3 – логотип сайту для онлайн опитування

Опитування роботодавців було спрямоване на оцінку їхньої задоволеності результатом діяльності українських вишів (наданою ними освітою) та визначення кращих навчальних закладів на думку роботодавців. На основі відповідей роботодавців обчислювався критерій рейтингу, який відображав думку роботодавців щодо відповідності знань та навичок випускників ЗВО потребам ринку праці в цілому, та окремо по кожному з обраних напрямків спеціалізації.

Опитування випускників проводилося методом особистого, телефонного та онлайн-опитування та, частково, методом опитування за допомогою електронної пошти. Така комбінація методів була обрана з метою охопити якомога більшу кількість випускників вузів, які брали участь у рейтингу, та

запропонувати респонденту найзручніший для нього варіант участі у дослідженні. Опитування випускників було спрямовано оцінку їх задоволеності своєю освітою та можливістю її використання у трудовій діяльності. А також з'ясовувалась їхня думка щодо кращих навчальних закладів. На основі відповідей випускників обчислювався критерій рейтинга, що відображає задоволеність випускників вузів своєю освітою та можливістю застосування його у трудовій діяльності.

Опитування експертів. У межах дослідження було проведено опитування двох груп експертів:

- спеціалісти агенцій з відбору персоналу (з усієї України);
- фахівці державної служби зайнятості населення (з усієї України).

На підставі відповідей експертів та їх подальшої статистичної обробки було вираховано вагові коефіцієнти для кожного з критеріїв рейтингу, які враховувалися при побудові підсумкового рейтингу ЗВО.

Опитування студентів. Оцінки студентів не були включені до загального рейтингу, оскільки в ході аналізу зібраних даних вони взагалі не пов'язані з оцінками роботодавців та експертів, мінімально корелюючи лише з оцінками випускників. Як видається, це обумовлено тим, що студенти здебільшого ще не діють повноцінно на ринку праці, а тому переважно оцінюють своє місце навчання під іншим кутом зору, ніж випускники та роботодавці. Тому було вирішено не враховувати їх оцінки як такі, що не відповідають цілям побудови рейтингової оцінки вузів.

За результатами 2012 року перелік лідерів не зазнав істотних змін порівняно з попереднім роком.

Перше місце у рейтингу стабільно утримує Національний технічний університет України «Київський політехнічний інститут» (90 балів).

На другому місці з незначним відставанням — Київський національний університет імені Тараса Шевченка (83 бали).

Третю сходинку рейтингу розділили Національний університет «Києво-

Могилянська академія» (43 бали) та Київський національний економічний університет імені Вадима Гетьмана (42 бали).

На четвертому місці – Національний університет «Львівська політехніка» (36 балів), а замикає п'ятірку найкращих Київський національний університет будівництва та архітектури (30 балів). [3]

Ще однією оцінкою робіт є рейтинг безкоштовних фінансових маркерів за вказівниками Scopus. Наукометричний етап Scopus – це етап, який розглядає еталонний рейтинг логічних розподілів ЗВО України.

Стратегія визначення рейтингового місця включає:

- кількість публікацій у Scopus;
- кількість посилань у Scopus.

Таким чином, рейтинг залежить від значення індексу Гірша (h – список) – ознаки впливу дослідника, у світлі кількості його статей та посилань на них. Як вказує остання інформація, поширена етапом Scopus, першопрохідцем серед українських вільних фінансових зон є Київський національний університет імені Тараса Шевченка, чий найстрашніший список – 72, що стосується 11 644 розповсюдження та 43 198 посилань (рис. 4).

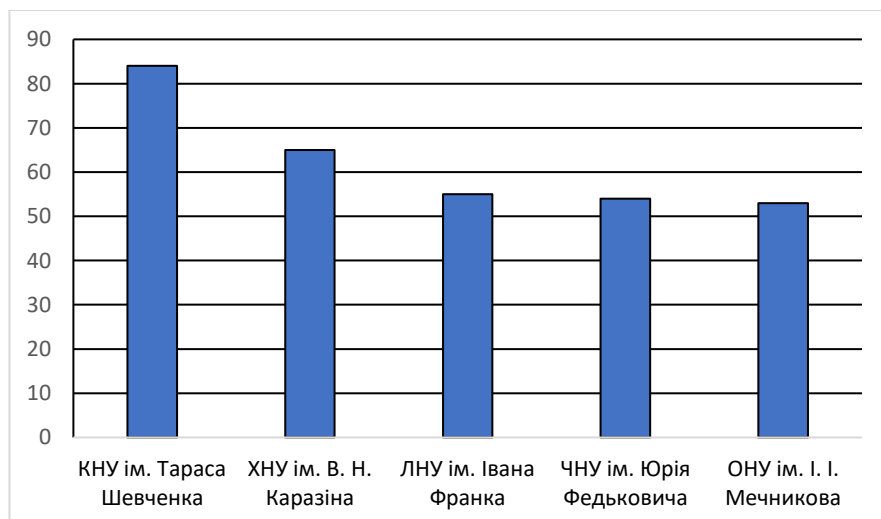


Рисунок 1.4 - Рейтинг наукометричної платформи Scopus у 2018

Станом на квітень 2018 року набір даних Scopus охоплює 162 фонди передової освіти України, що на 26 інструктивних організацій більше, ніж у

квітні 2017 року.

У рейтинговій таблиці заклади України ранжирувані за індексом Гірша - кількісним показником, що ґрунтується на кількості наукових публікацій та кількості їх цитувань.

За даними рейтингу, найвищий індекс Гірші серед університетів України мають Київський національний університет ім. Шевченка – 84 (79 у рейтингу 2017 року), Харківський національний університет ім. Каразіна – 65 (60) та Львівський національний університет ім. Франка – 55 (50), який посунув із третьої сходинки Чернівецький національний університет ім. Федьковича – 54 (51).

Розбираючи вищенаведені рейтингові стратегії, ми можемо дійти супутніх резолюцій: рейтинг «ТОП-200 Україна» досліджує вправи організацій передової освіти, використовуючи більш інкорпоровані списки. Водночас правила рейтингу «Компас» відображають оцінку вищої освіти випускниками та бізнесом на відповідність ступеню інструктивності адміністрацій потребам ринку праці, тобто останній варіант передбачає абстрактну оцінку незалежної більш правдивої інформації, яка запам'ятовується за файлами, як-от кількість вихователів, незнайомих дублерів тощо.

Дещо новим у всьому світі є багаторівневий рейтинг U - Multirank

U - Multirank – це затія ще одного багаторівневого позиціонування коледжів на планеті, яку започаткувала та підтримала Європейська комісія. Інженером процедури є відносини геологорозвідувальних асоціацій, які називаються організацією CHERPA (Консорціум з оцінки ефективності вищої освіти та досліджень).

U - Multirank оцінює коледжі по всій планеті, а не лише європейські; охоплює різні компоненти руху коледжу, включає як загальне розташування коледжів, так і позиціонування за напрямками підготовки; оцінює широкий спектр навчальних закладів та екзаменаційних закладів; вирішує питання різних партнерів.

Фундаментальні стратегічні стандарти цього рейтингу очікують, що його результати не будуть узагальнені в загальній оцінці, але дадуть комплексну оцінку за різними критеріями. Оскільки будь-яка оцінка допускає певну помилку, а контраст між коледжами може бути невеликим, щоб не перебільшувати відмінності між фондами, наслідки позиціонування не будуть наведені як викладка коледжів, очевидно характеризуваних місць, однак у групах із досить багато порівняльних вказівок. [4]

1.3 Постановка задачі

Тому згідно з актуальністю теми **об'єктом дослідження** процеси оцінювання ЗВО. Згідно з цим **предметом дослідження** є методи прийняття рішень та технології рейтингування закладів вищої освіти.

Мета: підвищення ефективності оцінювання ЗВО для допомоги вибору вступу абітурієнтам.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- проаналізувати сучасний стан задачі оцінювання ЗВО та аналоги систем;
- виокремити основні проблеми та запропонувати їх рішення;
- здійснити збір інформації;
- розробити програмне забезпечення.

Висновки до розділу 1

Тема дипломної роботи присвячена розробці системі оцінювання ЗВО. В першому розділі розглянута актуальність застосунку та її аналоги.

Метою є підвищення ефективності оцінювання ЗВО на основі нечіткого гібридного методу прийняття рішень.

2 МЕТОДИ, АЛГОРИТМИ ОЦІНЮВАННЯ ЗАКЛАДІВ ВИЩОЇ ОСВІТИ

2.1 Гібридні методи прийняття рішень

Обираючи методи для розробки системи потрібно зробити порівняльний аналіз відомих методів.

Методи прийняття багатокритеріальних рішень (MCDM) часто використовуються для вирішення реальних проблем із кількома суперечливими та несумірними критеріями. Проблеми MCDM, як правило, класифікуються як безперервні або дискретні, залежно від області альтернатив. Хван і Юн (1981) [5] класифікували методи MCDM на дві категорії: багатоцільове прийняття рішень (MODM) і MADM. MODM широко вивчається за допомогою методів математичного програмування з добре сформульованими теоретичними рамками. Методи MODM мають значення змінних рішень, які визначаються в безперервній або цілочисельній області з нескінченною або великою кількістю альтернативних варіантів, найкращий з яких повинен задовольняти обмеження і пріоритети особи, що приймає рішення. Методи MADM, з іншого боку, використовувалися для вирішення проблем з дискретними просторами рішень і заздалегідь визначеною або обмеженою кількістю альтернативних варіантів. Процес вирішення MADM вимагає порівняння між і внутріатрибутами і включає неявні або явні компроміси. [6]

Гібридні системи підтримки прийняття рішень є поєднанням текстових, гіпертекстових, орієнтованих на використання баз даних, табличних та орієнтованих на моделі структур. Переважна більшість фреймворків СППР, які зараз використовуються, є міжпородними фреймворками.

Для явних передумов та умов підтримується структура СППР, яка містить позитивні властивості різних систем і обмежує негативні частини комбінованих систем. Є два основні підходи до інтеграції СППР: гніздування та синергія. Стратегії гніздування включають обмін властивостями вкладної системи до

системи яка прийняла цю підсистему. Наприклад, Word може прийняти чи використовувати деякі властивості таблиці Excel яка вкладена в документ Word.

У синергетичному підході до інтеграції різних систем, немає вкладеності, немає домінуючої технології, немає вкладених систем і методів. Всі методи включені в основний інструмент, який дозволяє використовувати їх автономно один від одного, або об'єднати кілька технік у рамках однієї діяльності.

Нечіткі методи MADM були розроблені через недостатню точність в оцінці відносної ваги атрибутів і оцінок продуктивності альтернатив у реальних проблемах. Ця неточність може виникати з різних джерел, таких як: інформація, яка не піддається оцінці; неповна інформація; інформація, яку неможливо отримати; та/або часткове незнання (Chen, Hwang, & Hwang, 1992). [Класичні методи MADM не можуть ефективно впоратися з проблемами з неточною або нечіткою інформацією. Коли Беллман і Заде (1970), а кількома роками пізніше Циммерман (1985) ввели нечіткі набори в поле, вони розчистили шлях для нового сімейства методів для вирішення проблем, які були недоступні та нерозв'язні стандартними методами MCDM. .

У нечітких MCDM неточність і нечіткість, пов'язані з якісними даними, можна представити більш логічно за допомогою лінгвістичних змінних та функцій належності, що перекриваються. Крім того, дані, які вимірюються в різних одиницях, можна використовувати безпосередньо без стандартизації. Основною перевагою нечіткої логіки є те, що її можна використовувати як компенсуючу та не компенсуючу в одній моделі в різних контекстах, використовуючи умовиводи через судження, надані DM. Різниця між компенсаційною та некомпенсаційною оцінкою полягає в тому, що перша враховує компроміс між критеріями оцінки, а друга ігнорує ціннісні компроміси (Keeney, 1980).

2.2 АНР і нечіткий АНР

Аналітичний ієрархічний процес (АНР) — це добре відомий підхід MADM, запропонований Сааті (1977, 1980) [5] для спрощення складних і погано

структурованих проблем шляхом впорядкування атрибутів і альтернатив рішення в ієрархічній структурі за допомогою серія парних порівнянь. Дайер і Форман (1992) [6] описують переваги АНР в груповій обстановці наступним чином: обговорення зосереджено як на матеріальних, так і на нематеріальних, індивідуальних і спільних цінностях; дискусія може бути зосереджена на цілях, а не на альтернативах; обговорення можна структурувати так, що кожен атрибут можна розглядати по черзі; і обговорення триває доти, доки не буде розглянута вся відповідна інформація та не буде досягнуто консенсусного вибору альтернатив рішення. Saaty (2005) [7] стверджує, що DM, природно, вважає, що легше порівнювати дві речі, ніж порівнювати всі речі разом у списку. АНР також перевіряє узгодженість DM та дозволяє переглядати їхні відповіді. АНР застосовувався до багатьох різноманітних рішень через інтуїтивну природу процесу та його потужність у вирішенні складнощів у проблемах, пов'язаних із судженням. Незважаючи на його широке використання та популярність, звичайний метод АНР не здатний впоратися з невизначеністю та невизначеністю, пов'язаними з відображенням якісних переваг DM на точкові оцінки в матрицях парного порівняння. Проблема генерування вектора пріоритету з невизначеної матриці попарного порівняння є основою для концепції нечіткої АНР. Існує кілька процедур для генерування пріоритетних векторів у нечітких задачах АНР. Метод середнього геометричного, нечіткий логарифмічний метод найменших квадратів, синтетичний розширений аналіз, метод нечітких найменших квадратів, метод прямої фазифікації, програмування нечітких переваг і двоступеневе логарифмічне програмування є деякими з цих методів. Останні застосування нечіткого АНР, серед іншого, включають оцінку ефективності автобусних компаній; оцінка інформаційних технологій; рішення щодо розробки нових продуктів; оцінка управлінських талантів; оцінка критичних факторів успіху в електронній комерції; оцінка планів управління водними ресурсам; оцінка дослідницьких проектів і розробок; оцінка управління безпекою; оцінка критичних факторів успіху в обміні знаннями; оцінка систем

планування ресурсів підприємства; вибір людських ресурсів; оцінка операторів з кількома навичками; аналіз якості медичних послуг; вибір процесу виготовлення пластин та оцінка ризику.

Fuzzy АНР є розширенням АНР і є корисною технікою для вирішення складних проблем прийняття рішень [7]. Будь-яку складну проблему можна розкласти на різні ієрархічні рівні критеріїв. У кожній ієрархії проводиться серія парних порівнянь, щоб визначити важливість критеріїв. У АНР експерти використовують чіткі числа (наприклад, 1, 2, ... 9), щоб визначити важливість критеріїв, тоді як експерти з нечітких АНР використовують природні лінгвістичні терміни (наприклад, однаково важливі, слабо важливі), щоб висловити свої судження. Лінгвістичні терміни представляють відповідні нечіткі числа, визначені у нечітких функціях належності [9].

Чанг [10] запропонував підхід нечіткого АНР, заснований на методі аналізу протяжності, який широко використовується в задачах вибору. Цей метод використовує лінгвістичні змінні для вираження порівняльних суджень, які дають особи, які приймають рішення. Нехай $X = \{x_1, x_i, \dots, x_n\}$ представляють набір об'єктів, а $G = \{g_1, g_j, \dots, g_m\}$ поставлена ціль. У методі, запропонованому Чангом, береться кожен об'єкт x_i та виконується аналіз протяжності для кожної мети g_j . Таким чином, можна отримати m значень аналізу протяжності для кожного об'єкта з такими ознаками:

На другому кроці розробляється попарна матриця нечіткого порівняння. Судження попарного порівняння представлені нечіткими трикутними числами, позначеними $\tilde{a}_{ij} = (l_{ij}, m_{ij}, u_{ij})$. Як і в звичайній АНР, $n(n-1)/2$ суджень потрібно для кожної групи порівняння для рівня для побудови позитивної нечіткої матриці взаємного порівняння $\tilde{A} = \{\tilde{a}_{ij}\}$.

Враховуючи вищезгадану теорію, запропонована процедура нечіткого методу АНР визначається таким чином:

Крок 1. АНР використовує кілька невеликих підзадач, щоб представити складну проблему вирішення. Таким чином, перша дія полягає в розкладі

проблеми прийняття рішення на ієрархію з метою у верхній частині, критеріями та підкритеріями на рівнях і підрівнях та альтернативами рішень у нижній частині ієрархії.

Крок 2. Матриця порівняння передбачає порівняння в парах елементів побудованої ієрархії. Мета полягає в тому, щоб встановити їхні відносні пріоритети щодо кожного з елементів на наступному вищому рівні.

$$D = \begin{matrix} & \begin{matrix} C_1 & C_2 & C_3 & \cdots & C_n \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & & x_{2n} \\ x_{31} & x_{32} & x_{33} & & x_{3n} \\ \vdots & & & & \\ x_{n1} & x_{n2} & x_{n3} & & x_{nn} \end{bmatrix} \end{matrix} \quad (2.1)$$

Елементи $\{x_{ij}\}$ можна інтерпретувати як ступінь переваги і-го критерію над j-им критерієм. Виявляється, що визначення ваги критеріїв є більш надійним при використанні парних порівнянь, ніж при їх безпосередньому отриманні, тому що легше провести порівняння між двома атрибутами, ніж зробити загальне присвоєння ваги. Перед усіма обчисленнями вектора пріоритетів порівняльну матрицю необхідно нормувати в діапазон $[0,1]$ за рівнянням (2.2):

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}} \quad (2.2)$$

$$R = \begin{matrix} & \begin{matrix} C_1 & C_2 & C_3 & \cdots & C_n \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{matrix} & \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ r_{21} & r_{22} & r_{23} & & r_{2n} \\ r_{31} & r_{32} & r_{33} & & r_{3n} \\ \vdots & & & & \\ r_{n1} & r_{n2} & r_{n3} & & r_{nn} \end{bmatrix} \end{matrix} \quad (2.3)$$

Крок 3. АНР також обчислює індекс невідповідності (або коефіцієнт узгодженості), щоб відобразити узгодженість суджень особи, яка приймає рішення, на етапі оцінки. Індекс невідповідності як у матриці рішень, так і в матрицях попарного порівняння можна розрахувати за допомогою рівняння (2.4) [11]:

$$CI = \frac{\lambda_{\max} - N}{N - 1} \quad (2.4)$$

де $\lambda_{\max}$ – головне власне значення матриці суджень,

n – порядок матриці суджень.

Чим ближче індекс невідповідності до нуля, тим більше узгодженість. Узгодженість оцінок забезпечується, якщо для всіх критеріїв виконується рівність ($a_{ij}a_{jk} = a_{ik}$, i, j, k). Відповідний індекс має бути нижчим за 0,10, щоб визнати результати АНР узгодженими. Якщо це не так, особа, яка приймає рішення, повинна повернутися назад і переробити оцінки та порівняння [10].

Крок 4. На наступному кроці перетворіть дійсні елементи матриці R у нечіткі числа.

Крок 5. Перед тим, як провести всі обчислення вектора пріоритетів, порівняльну матрицю D необхідно нормувати.

Крок 6. Щоб знайти вагові коефіцієнти критеріїв, обчисліть середнє значення елементів кожного рядка з матриці, отриманої на кроці 4, за наступним визначенням.

Нехай $\tilde{a} = (a_1, a_2, a_3)$, $\tilde{b} = (b_1, b_2, b_3)$ — два трикутних нечітких числа, тоді визначено метод для обчислення поділеного між $\tilde{a}$ та $\tilde{b}$, як рівняння (2.4) – (2.7) [9]:

$$\frac{\tilde{a}_\alpha}{\tilde{b}_\alpha} = \left[\frac{(a_2 - a_1)\alpha + a_1}{-(b_3 - b_2)\alpha + b_3}, \frac{-(a_3 - a_2)\alpha + a_3}{(b_2 - b_1)\alpha + b_1} \right] \quad (2.5)$$

де $\alpha = 0$,

$$\frac{\tilde{a}_0}{\tilde{b}_0} = \left[\frac{a_1}{b_3}, \frac{a_3}{b_1} \right] \quad (2.6)$$

де $\alpha = 1$,

$$\begin{aligned}\frac{\tilde{a}_1}{\tilde{b}_1} &= \left[\frac{(a_2 - a_1) + a_1}{-(b_3 - b_2) + b_3}, \frac{-(a_3 - a_2) + a_3}{(b_2 - b_1) + b_1} \right] \\ \frac{\tilde{a}_1}{\tilde{b}_1} &= \left[\frac{a_2}{b_2}, \frac{a_2}{b_2} \right]\end{aligned}\quad (2.7)$$

Отже, наближене значення $\tilde{a}=\tilde{b}$ буде

$$\frac{\tilde{a}}{\tilde{b}} = \left[\frac{a_1}{b_3}, \frac{a_2}{b_2}, \frac{a_3}{b_1} \right] \quad (2.8)$$

2.3 TOPSIS і нечіткий TOPSIS

TOPSIS спочатку був запропонований Хваном і Юном (1981). [6] Згідно з цією технікою, найкращою альтернативою є та, яка найближча до ідеального рішення і найдалша від найнижчого (негативного ідеального) рішення. Ідеальним рішенням є рішення, яке максимізує критерії вигоди і мінімізує критерії витрат, тоді як рішення надір є рішенням, яке максимізує критерії витрат і мінімізує критерії вигоди. Іншими словами, ідеальне рішення складається з усіх найкращих значень, які можна досягти за критеріями, тоді як розв'язок найгіршого рівня складається з усіх найгірших значень, які можна досягти за критеріями. Було показано, що TOPSIS є одним із найкращих методів MADM у вирішенні проблеми зміни рангу, тобто зміни рейтингу альтернатив, коли вводиться неоптимальна альтернатива. Ця особливість узгодженості значною мірою цінується в практичних застосуваннях. Більше того, зміна рангу в TOPSIS нечутлива до кількості альтернатив і має найгірші результати лише у випадку дуже обмеженої кількості атрибутів. Відносно перевагою TOPSIS є його здатність швидко визначати найкращу альтернативу. Незважаючи на його популярність і простоту концепції, звичайний TOPSIS часто критикують через його нездатність впоратися з невизначеністю і неточністю, властивими реальним проблемам. У звичайній формулюванні TOPSIS судження DM представлені точними числовими значеннями. Однак часто в практичних випадках DM

можуть бути не в змозі призначити числові значення своїм судженням. TOPSIS широко застосовувався для вирішення різноманітних проблем із багатьма атрибутами. 482 М. Тавана та ін.

Останні застосування нечіткого TOPSIS включають: оцінку ризику моста; вибір консультантів з повного управління якістю; оцінка накопичення теплової енергії в системах концентрованої сонячної енергії; аварії з розливом нафти в морі; аналіз ділової конкуренції в авіаційній галузі; оцінка стійких транспортних систем; енергетичне планування; рішення щодо прийняття продукту; вибір менеджера; оцінка бізнес-аналітики для корпоративних систем; планування розташування банку; вибір бездротової мережі та планування розташування об'єктів.

Нечіткий метод TOPSIS був запропонований Ченом [17] для вирішення багатокритеріальних проблем прийняття рішень у невизначеності. Метод TOPSIS заснований на виборі найкращої альтернативи, яка має найкоротшу відстань від позитивно-ідеального рішення і найбільшу відстань від негативно-ідеального рішення. Нечіткий TOPSIS працює наступним чином. Матрицю нечітких рішень (X) можна нормалізувати за допомогою рівняння (2.9) для формування нормованої матриці рішень (R):

$$\tilde{R} = [\tilde{r}_{ij}]_{n \times m} \quad (2.9)$$

де i – альтернативи; $i = 1; 2; \dots, m$;

j – критерії; $j = 1; 2; \dots, n$;

r_{ij} - агрегований нечіткий рейтинг.

$$\tilde{r}_{ij} = \left(\frac{l_{ij}}{\sqrt{\sum_i u_{ij}^2}}, \frac{m_{ij}}{\sqrt{\sum_i u_{ij}^2}}, \frac{u_{ij}}{\sqrt{\sum_i u_{ij}^2}} \right) \quad (2.10)$$

де l, m, u - три точки трикутного нечіткого числа A .

Зважена нормована матриця рішень (V) обчислюється шляхом множення ваг критеріїв (W) на відповідні елементи нормалізованої матриці рішень (R).

$$\tilde{V} = [\tilde{v}_{ij}]_{n \times m} \quad (2.11)$$

де v_{ij} формулюється як:

$$\tilde{v}_{ij} = \tilde{r}_{ij} \times \tilde{w}_j \quad (2.12)$$

Позитивне ідеальне рішення (A^+) буде визначено шляхом вибору найбільшої нормованої та зваженої оцінки для кожного критерію (Рівняння (2.15)). Аналогічно негативне ідеальне рішення A визначається шляхом вибору найменш нормалізованої та зваженої оцінки кожного критерію [51] (рівняння (2.16)).

$$\tilde{A}^+ = \{\tilde{v}_1^+, \tilde{v}_2^+, \dots, \tilde{v}_n^+\} \quad (2.13)$$

$$\tilde{A}^- = \{\tilde{v}_1^-, \tilde{v}_2^-, \dots, \tilde{v}_n^-\} \quad (2.14)$$

Відстань кожної альтернативи до позитивного ідеального рішення (d^+) та негативного ідеального рішення (d^-) ($i=1, 2; \dots; m$) розраховується з використанням формули 2.15, вираженого у визначеннях TFN.

$$\begin{aligned} \tilde{A} \ominus \tilde{B} &= (l_1, m_1, u_1) \ominus (l_2, m_2, u_2) \\ &= (l_1 - u_2, m_1 - m_2, u_1 - l_2) \end{aligned} \quad (2.15)$$

$$d_i^+ = \sum_{j=1}^n d_v(\vartheta_{ij}, \vartheta_j^+) \quad (2.16)$$

$$d_i^- = \sum_{j=1}^n d_v(\vartheta_{ij}, \vartheta_j^-) \quad (2.17)$$

Використовуючи ці значення відстані, потім обчислюється індекс близькості (CI) для кожної альтернативи за допомогою рівняння (2.18) [34]:

$$(CI) = \frac{(d_i^-)}{(d_i^+) + (d_i^-)} \quad (2.18)$$

Індекс близькості може приймати значення від 0 до 1, а альтернатива з найвищим CI вибирається як найкраща [49].

Висновки до розділу 2

Перш за все, усі оціночні рамки мають на меті дозволити абітурієнту зупинитися на якомусь ЗВО, а насправді на той час влаштувати своєрідний конкурс для такого ж абітурієнта як життєво важливої фігури та гарантії будь-якого фундаменту передової освіти. Як правило, рейтингові системи, як державні, так і світові, дозволяють уявити огляд недоліків організації передової освіти на відміну від інших. Відповідно, дослідження стратегій позиціонування дозволяє закладам виділяти для роботи над влаштуванням інструктивних адміністрацій щодо їх якості. Таким чином, вважаю доцільним для нашої власної структури вжити супутніх заходів для оцінки ЗВО:

- індекс якості науково-педагогічного потенціалу;
- індекс міжнародного визнання;
- оцінка задоволеність випускників ЗВО отриманим ними освітою та можливістю застосування його в трудової діяльності.

3 АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ СИСТЕМИ

3.1 Аналіз варіантів інструментальних засобів для розробки системи

Python — це розшифрована, об'єктно-впорядкована мова програмування значного рівня з динамічною семантикою. Незаперечні інформаційні структури, притаманні рівню, поєднані з динамічним компонуванням і динамічним обмеженням, роблять його надзвичайно привабливим для швидкого вдосконалення додатків, а також щодо використання в якості мови попередньої аранжування або приєднання до інтерфейсу існуючих частин разом. Проста, проста для вивчення лінгвістична структура Python підкреслює змістовність і таким чином зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що дає змогу програмі виміряти якість та повторно використовувати код. Посередник Python і величезна стандартна бібліотека доступні у вихідному коді або об'єднані безкоштовно на кожному важливому етапі та можуть бути відкрито поширені [5].

Виходячи з цих тверджень можна зробити загальний висновок, що для розробки додатку для аналізу країн світу по рівню економічного розвитку є Python. Завдяки бібліотекам мови можна розробити гарно спроектований графічний інтерфейс, метод ієрархічної кластеризації, який допоможе у рішенні задачі кластеризації.

Так як основним об'єктом для застосування інформаційної системи для класифікації комерційних банків, які використовують локальні способи зберігання даних, а також з обраною мовою програмування Python, було вирішено обрати програму PyCharm.

PyCharm доступний у трьох виданнях: Professional, Community та Edu. Видання The Community та Edu - це проекти з відкритим кодом, і вони безкоштовні, але мають менше можливостей. PyCharm Edu пропонує курси та допомагає вивчати програмування за допомогою Python. Видання Professional є

комерційним та пропонує видатний набір інструментів та функцій [13].

Бібліотека PyQt5 - це повний набір прив'язок Python для Qt v5. Він реалізований як понад 35 модулів розширення і дозволяє використовувати Python як альтернативну мову розробки додатків для C++ на всіх підтримуваних платформах, включаючи iOS та Android.

Варто зазначити, що не менш важливим є вибір формату, у якому будуть зберігатися дані, адже програма повинна бути налагоджена таким чином, щоб її функціонал задовольняв потребам більшості користувачів.

Отже, так як для багатьох компаній зберігання даних у базі даних є звичайною практикою, однак так як для пошуку асоціативних правил виду «Якщо купив X, то купити і Y» необхідна лише вибірка даних – список куплених продуктів кожної транзакції. Саме тому при виборі типу формату даних, який є вхідним для програми, було обрано формат таблиці.

Excel - це електронна таблиця, розроблена та опублікована корпорацією Майкрософт. Він є частиною набору програм для підвищення продуктивності Microsoft Office.

На відміну від текстового процесора, такого як Microsoft Word, Excel упорядковує дані у стовпці та рядки. Рядки і стовпці перетинаються в просторі, який називається клітиною. Кожна комірка може містити окремі дані, такі як текст, числове значення або формула.

Excel є інструментом для організації та виконання розробок за даними. Він може аналізувати дані, розшифровувати статистику, створювати таблиці поворотів та представляти дані у вигляді діаграм або графіків. Допомогає працювати з великим та середнім об'ємом даних.

Для порівняння методів використовувалася саме ця мова програмування та додаток.

Трикутне нечітке число A — це нечіткое число з кусково-лінійною функцією належності оцінки продуктивності та ваги оцінюються за допомогою лінгвістичних термінів. Мовні ваги для представлення важливості критеріїв є

дуже низькими (VL), низькими (L), середніми (M), високими (H) і дуже високими (VH).

```
VERY_LOW(new double[] {0.0, 0.0, 0.25}),
LOW(new double[] {0.0, 0.25, 0.50}),
GOOD(new double[] {0.25, 0.50, 0.75}),
HIGH(new double[] {0.50, 0.75, 1.0}),
VERY_HIGH(new double[] {0.75, 1.0, 1.0});
```

Встановлення балів попарного порівняння та обчислення усіх відповідних чисел для АНР

```
void setPairwiseComparisonArray(double[] a) {
    int i = 0;
    for (int row = 0; row < nrAlternatives; row++) {
        for (int col = row + 1; col < nrAlternatives; col++) {
            //System.out.println(row + "/" + col + "=" + a[i]);
            getMtx().setEntry(row, col, a[i]);
            getMtx().setEntry(col, row, 1.0 / getMtx().getEntry(row, col));
            comparisonIndices[i][0] = row;
            comparisonIndices[i][1] = col;
            i++;
        }
    }
}
```

Повернення отриманих вагів для альтернатив

```
double[] getWeights() {
    return weights;
}
```

```

Output:
Calculating AHP Criteria weighting:
Importance of Bandwidth compared to Speed= 1.0
Importance of Bandwidth compared to Availability= 5.0
Importance of Bandwidth compared to Security= 7.0
Importance of Bandwidth compared to Price= 9.0
Importance of Speed compared to Availability= 5.0
Importance of Speed compared to Security= 6.0
Importance of Speed compared to Price= 8.0
Importance of Availability compared to Security= 3.0
Importance of Availability compared to Price= 3.0
Importance of Security compared to Price= 2.0
Consistency Index: 0.0300
Consistency Ratio: 2.4202%
Weights:
Bandwidth: 0.4073
Speed: 0.3886
Availability: 0.1084
Security: 0.0573
Price: 0.0385
*****
Calculating Fuzzy TOPSIS:
Public=[LOW, VERY_HIGH, VERY_HIGH, GOOD, VERY_HIGH]
Mobile=[VERY_HIGH, GOOD, HIGH, HIGH, VERY_LOW]
Edge=[VERY_HIGH, HIGH, HIGH, HIGH, LOW]
unweighted fuzzy values: {Public=[0.0, 0.25, 0.5, 0.75, 1.0, 1.0, 0.75, 1.0, 1.0, 0.25, 0.5, 0.75, 0.75, 1.0,
weighted fuzzy values: {Public=[0.0, 0.09714034130469436, 0.19428068260938872, 0.2914210239140831, 0.388561365
D+ for Public is: 1.9433
D+ for Mobile is: 3.6362
D+ for Edge is: 5.3132
D- for Public is: 0.9809
D- for Mobile is: 2.2196
D- for Edge is: 3.4756
closeness coefficient set is: {Edge=0.395455571823559, Mobile=0.37904836990577323, Public=0.3354325910373735}
*****
Final Ranking:
Edge: 0.3955
Mobile: 0.3790
Public: 0.3354

```

Рисунок 3.1 – Програмне порівняння методів

Результатом представлено ранжуванням за отриманими результатами.

Порівняння обох методів базувалося на наборі необхідних характеристик методів для адекватного вирішення проблеми вибору, як представлено в Розділі У більшості випадків Fuzzy TOPSIS працює краще, ніж Fuzzy AHP, за винятком випадків, коли критеріїв дуже мало. Крім того, збільшення кількості альтернатив накладає деякі обмеження на Fuzzy AHP. Що стосується Fuzzy TOPSIS, то це не є обмеженням для використання методу. У разі кількості критеріїв, внутрішні обмеження, накладені методом Fuzzy AHP, можна подолати шляхом розгортання критеріїв у структурі FuzzyANPhierarchy. Водночас, нечіткий TOPSIS не обмежує кількість критеріїв, він не дозволяє розгортати критерії на підкритерії, що можна розуміти як слабкість методу при застосуванні до проблеми вибору. Подальше дослідження може бути зосереджено на адаптації Fuzzy TOPSIS, щоб

вмістити критерії та підкритерії в матрицю рішень. Що стосується часової складності, то вона в цілому нижча для Fuzzy AHP, ніж для Fuzzy TOPSIS. Однак, якщо виконується тест на узгодженість матриці рішень Fuzzy AHP, який часто необхідний, то перевага методу Fuzzy AHP не настільки виражена. Fuzzy AHP вимагає більш складних обчислень, ніж Fuzzy TOPSIS. Обидва методи адекватно підтримують групове прийняття рішень. Варто зазначити, що середнє зважене може використовуватися для агрегування суджень замість звичайного середнього арифметичного. Роблячи це, можна надавати різне значення різним особам, які приймають рішення.

3.2 Платформа .NET та мова програмування C#

Одним з головних результатів під час практики є вибір засобів розробки з врахуванням вимог до програмного забезпечення.

Враховуючи цільову клієнтську частину на Unity, головним середовищем для розробки серверної та клієнтської частини було обрано мову програмування C#. Основою реалізації усіх проєктів за допомогою C#, а саме Windows Forms є робота з формами та їхніми елементами. До кожної форми в проєкті створюється свій клас, для подій можна прив'язати виконання функцій. Усередині .NET Framework, Windows Forms реалізується та компілюється в рамках однойменних рамках простору імен «System.Windows.Forms».

Мова програмування C# — це широко використовувана мова програмування, яка включає в себе складний дизайн, розширення лексики, основні, широко поширені предмети (на основі класів) і дисципліни програмування розділів. C# є одним із діалектів програмування, створених для загальномовної інфраструктури (CLI).

За планом, C# є мовою програмування, яка прямо відображає структуру фундаментальної мови (CLI). Більша частина його домашнього асортименту пов'язана з типами характеристик, які виконує CLI. Тим не менш, окремій мові не потрібно турбуватися про вік коду компілятора, тобто вона не визначає, що

компілятор C # повинен орієнтуватися на звичайний клімат під час виконання мови або створювати інший однозначний дизайн. Спекулятивно, компілятор C# може створювати машинний код, як звичайні компілятори C++ і Fortran [11].

C # підтримує запитуючий послідовний тип даних, bool. Голови, які сприймають умови, наприклад, коли і якщо, вимагають таких заяв, які відтворюють справжнє пояснення як послідовний погляд. Незважаючи на те, що C ++ має гарний вигляд, він має тенденцію справді відрізнятися від нескінченних чисел, а також пояснення, наприклад, якщо (a) по суті вимагає a to bool, що дозволяє a бути int або вказівником. C# блокує цей «математичний чи неправильний» підхід, оскільки обмеження програмістів використанням пояснень, які повертають просто bool, може запобігти певним видам помилок програмування, наприклад, якщо (a = b) (використання завдання = не відповідає ==) .C # підтримує суворо складені формулювання зрозумілих факторів за допомогою ключового слова var, а також підтверджені експонати з новим лозунгом [].

C # безпечніший, ніж C ++. Звичайні зміни для одного тесту – це ті, які розглядаються як захищені, як розширення чисел. Це використовується в загальному часі під час JIT і знову і знову під час виконання. Немає ніяких таємних змін між науковими характеристиками та цілими числами, а також між зареєстрованими людьми та цілими числами (за винятком суворого 0, який, очевидно, може бути витіснений будь-яким виділеним видом). Будь-яку контрольовану зміну слід чітко характеризувати як однозначну або неявну, замість конструкторів C++ і звичайно очевидних виконавців змін.

C # має експрес-довідку щодо коваріації та контррізноманітності в багатьох видах, включаючи C ++, який має певний ступінь допомоги проти різновидів просто через семантику повернутих типів віртуальних стратегій. C # не дозволяє глобальні фактори чи працює. Усі методи та особи повинні бути повідомлені всередині занять. Статичні особистості з відкритих класів можуть витіснити світові чинники та роботи [11].

Фактори поблизу не можуть приховати фактори, що містяться в блоці, на відміну від C і C ++.

C # підтримує ретельно впорядковані вказівники до межі за допомогою спеціального секретного ключа. Як і псевдо-C ++ фреймворк Qt, C # має семантику, включаючи екземпляри стилю участі дисперсії, хоча C # використовує делегатів.

На відміну від C ++, C # не підтримує численні застарілі варіанти, хоча клас може запускати безліч інтерфейсів. Це була організація головного спеціаліста з мови, щоб уникати проблем і подальшого розвитку потреб у конфігурації через CLI. Тобто дві процедури з самою назвою, які сприймають межі порівнянного типу в порівнянному питанні, C# дозволяє вам виконувати кожен методологію з огляду на точку підключення, яка називається цією стратегією, або, подібно до Java, дозволяє виконати один раз і мають один вимагати кожного виклику через інтерфейс будь-якого класу.

Microsoft працює над довідковим компілятором із відкритим вихідним кодом C# і відсіком для пристроїв із кодом "Roslyn". Компілятор, який повністю складається з контрольованого коду (C #), було відкрито, і утиліта відображається як API. Це дозволяє архітекторам виконувати рефакторинг і демонструвати апарати.

Інші компілятори C# (деякі включають мовний клімат за замовчуванням і бібліотеки класів .NET).

Мова програмування C# була обрана як мова програмування первинного ключа в цій пропозиції.

C # натякає на діалекти з C-подібною структурою речень, граматики яких досить близька до C ++ і Java. Придбання великої кількості своїх предків - C++, Java та інших. - C#, з огляду на їх прагматичне використання та забороняє деякі моделі, які визнали себе небезпечними для покращення програмування. Наприклад, C#, на відміну від C++, не підтримує множину.

3.3 Використання запитів MySQL при реалізації додавання, видалення, пошуку та оновлення даних полів в таблицях

MySQL - це реляційна система управління базами даних з відкритим вихідним кодом на основі мови SQL. Вона була розроблена і оптимізована для веб-додатків і може працювати на будь-якій платформі. Так як з розвитком Інтернеті з'явилися нові вимоги, веб-розробники вважають за краще використовувати для веб-додатків платформу MySQL. База даних MySQL призначена для обробки мільйонів запитів і тисяч транзакцій, тому її часто вибирають компанії електронної комерції, яким потрібно керувати великою кількістю грошових переказів. Гнучкість у міру необхідності - основна характеристика MySQL [14].

MySqlConnection має багато методів, з якими можна працювати та облегшити роботу з таблицями. Далі треба відкрити з'єднання з нашою базою даних за допомогою:

```
MySqlConnection connection = new
MySqlConnection("server=127.0.0.1;port=3306;username=root;database=kurs");
```

MySqlConnection – ініціалізує новий екземпляр класу MySqlConnection.

Обов'язковою умовою для роботи із таблицями є одне правило – треба закривати усе, коли закінчили. Для цього після обробки запиту викликається метод `connection.Close()`;

В основі MySQL можна виділити три найпоширеніші команди – `insert`, `delete`, `update`. Підключення вже існує, тому для подальшої роботи створюємо команду-запит в форматі MySQL:

```
MySQLCommand command = new MySQLCommand("SELECT * FROM client WHERE nameclient =
@nameclient OR lastname = @last", db.getConnection());

command.Parameters.Add("@nameclient", MySqlDbType.VarChar).Value =
nameclient.Text;
command.Parameters.Add("@last", MySqlDbType.VarChar).Value = lastname.Text;
```

Де `"@nameclient"` – заглужки - емуляція деякого елемента системи, із заданими властивостями [15]. Виконується запит SQL, результатом чого дані, введені у поля, відправляються до таблиці. Таким самим методом реалізовані інші

можливості програми:

- Пошук за ідентифікатором:

```
if (string.IsNullOrEmpty(find.Text))
{
    this.clientTableAdapter.Fill(this.appData.client);
    clientBindingSource.DataSource = this.appData.client;
}
else
{
    var query = from o in this.appData.client
                where o.nameclient.Contains(find.Text)
                select o;
    clientBindingSource.DataSource = query.ToList();
}
```

- Показати таблицю:

```
this.clientTableAdapter.Fill(this.appData.client);
clientBindingSource.DataSource = this.appData.client;
```

- Видалити:

```
if (e.KeyCode == Keys.Delete)
{
    if (MessageBox.Show("Ви точно бажаєте видалити цей запис?", "Message",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        clientBindingSource.RemoveCurrent();
}
```

- Оновити:

```
try
{
    clientBindingSource.EndEdit();
    clientTableAdapter.Update(this.appData.client);
    idclient.Enabled = false;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
```

Висновки до розділу 3

Отже, в третьому розділі описано методи, фреймворки та основні скрипти, що використовувались для розробки системи оцінювання. Описано універсальний фреймворк з відкритим вихідним кодом для C#, який було обрано для створення програмного забезпечення. Обґрунтовано вибір бази даних, середовищ та мов програмування для розробки. Описано роботу з MySQL – Hibernate, що дозволяє розробнику працювати з базою даних не на пряму, як це можна зробити за допомогою бібліотеки JDBC у вигляді класів.

Було реалізовано в програмному коді алгоритм вирішення задачі з використанням методу TOPSIS.

В додатку Б описана реалізація системи оцінювання за допомогою згаданого метода.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ

4.1 Опис програмної реалізації

Для кожного застосунку потребується графічне зображення. Тому що для початківців буде простіше використовувати програмні засоби. Вони дозволяють простим користувачам відкривати меню, переміщати файли, запускати програми або шукати в Інтернеті, не вказуючи комп'ютера функції для виконання через командний рядок.

Використовується найпопулярніша мова програмування – Python [17]. Ця програма не має переваг у сфері розробки інтерфейсів, але вона має ряд модулів для створення графічних інтерфейсів.

При розробці інформаційної системи використовувалась бібліотека – PyQt.

PyQt - це прив'язка Python міжплатформенного набору графічних інтерфейсів Qt, реалізована як модуль Python. PyQt - це безкоштовне програмне забезпечення, розроблене британською фірмою Riverbank Computing. Він доступний на умовах, подібних до версій Qt, старших за 4.5; це означає різноманітні ліцензії, включаючи загальну публічну ліцензію GNU (GPL) та комерційну ліцензію, але не GNU Lesser General Public License (LGPL). PyQt підтримує Microsoft Windows, а також різні версії UNIX, включаючи Linux і MacOS (або Дарвін) [18].

Для кожної з форм автоматично створюється клас, який є похідним від класу Form.

У середині .NET Framework, Windows Forms реалізується та компілюється в рамках однойменних рамок простору імен «System.Windows.Forms».

Клас Menu.cs інкапсулює внутрішні деталі завантаження, знаходження та відкриття інших класів та їх даних. Тому щоб реалізувати роботу функціонування інших класів треба створити екземпляр та відправляти йому відповідні накази (наприклад, “створити нову форму класу Rooms.cs та завантажити її базу даних”). Наприклад:

```
private void rooms_Click(object sender, EventArgs e)
{
    this.Visible = false;
    Rooms obj4 = new Rooms();
    obj4.ShowDialog();
}
```

Код Menu.cs (для завантаження головного меню) наведений в додатку Б.

Код та дані, які описують клас називають членами. Дані, які задані класом називають полями або змінні. Код – містить в собі функції, наприклад найпоширеніший представник – методи. Метод – це аналог підпрограми. Методи класу можуть містити код, який взаємодіє з полями, що визначає клас.

Щоб не створювати кожен раз один і той же метод у кожному класі було вирішено створити метод, який буде доступним для усіх класів. Створюємо загальний клас DB, котрий успадковує усі форми з бази даних та усі потрібні йому методи у класу Database (System.Data). Щоб робити запити до бази даних потрібен клас, який має методи для роботи із запитами, в цьому випадку використовується MySql.Data.MySqlClient.

```
using MySql.Data.MySqlClient;
public class DB
{
    MySqlConnection connection = new
    MySqlConnection("server=127.0.0.1;port=3306;username=root;database=kurs");

    public void openConnection()
    {
        if (connection.State == System.Data.ConnectionState.Closed)
            connection.Open();
    }
    public void closeConnection()
    {
        if (connection.State == System.Data.ConnectionState.Open)
            connection.Close();
    }

    public MySqlConnection getConnection()
    {
        return connection;
    }
}
```

Успадкування надає процес, у ході якого один об'єкт отримує властивості іншого об'єкту. У програмній реалізації додатку це дуже важливий процес, тому що він забезпечує та підтримує усі принципи ієрархічної класифікації. Завдяки наслідуванню більша частина методів піддається систематизації завдяки

класифікаціям.

Якщо не використовувати ієрархію наслідування, то для кожного об'єкта треба було би створювати та явно визначати його властивості. А якщо використовувати наслідувати, то достатньо буде лише визначити ті властивості, які роблять об'єкт особливим та відмінним від інших об'єктів у програмі.

Більшість сучасних об'єктно-орієнтованих мов програмування (C #, Java, Delphi і ін.) підтримують можливість одночасно спадкуватися від головного класу і реалізовувати методи декількох інтерфейсів одним і тим же класом. Цей механізм дозволяє багато в чому замінити множинне наслідування, а саме - методи інтерфейсів необхідно перевизначити явно, що виключає помилки при наслідуванні функціональності однакових методів різних головних класів.

Щоб під'єднати базу даних до проекту треба у властивостях проекту додати / створити цю саму базу яка буде мати назву «AppData.xsd».

Після цього ми під'єднаємо базу даних до нашого проекту та створюємо таблиці з полями, які вказані у таблицях 2.1, та показані на рис.3.1. До кожної таблиці за змістом були створені зовнішні ключі для взаємодії в майбутньому.

4.2 Огляд користувацького інтерфейсу

На старті відкривається форма Form1.cs, яка дає змогу зайти користувачеві в систему(рисунок 4.1).

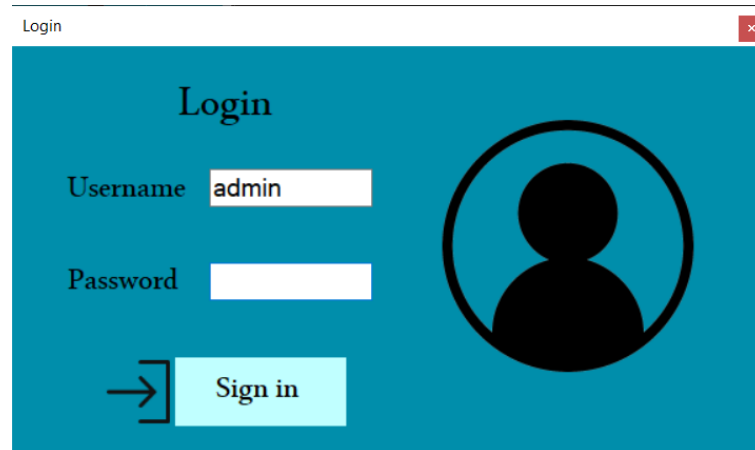


Рис. 4.1 – меню входу до системи

Для додатку був створений лише один аккаунт головного користувача, а саме для адміністратора, тому кнопка реєстрація не є обов’язковою, проте може бути додана у майбутньому. Логін користувача – admin, а пароль – “”.

Після того, як адміністратор введе потрібні дані – відкриється форма Menu.cs (див. рисунок 4.2). Форма Menu.cs складається із 4 Button. Всередині можна завантажити дані з файлу Excel або ввести вручну. Після цього дані можна рейтингувати. А також, якщо до цього процес оцінювання вже відбувався, то можна продивитись минулі результати. А також ще одного Button, «Вихід», який закриває програму.



Рисунок 4.2 – Зображення меню системи

Якщо користувач ввів неправильний логін або пароль – система оповістить про це та дає ще спробу(див. рисунок 4.3)

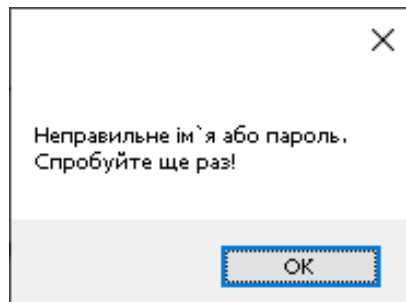


Рисунок 4.3 – помилка про некоректність даних

Якщо ж набір даних не буде обраний, і ви почнете взаємодіяти з кнопками, то побачите повідомлення про помилку

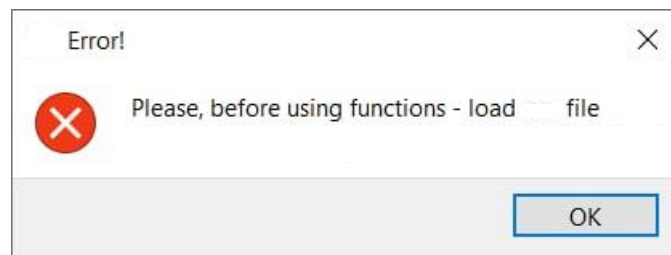


Рисунок 4.4 – Обробка помилки при спробі використати функції без завантаженого файлу

Або якщо користувач відкриє провідник та зачинить його не вибравши файл:

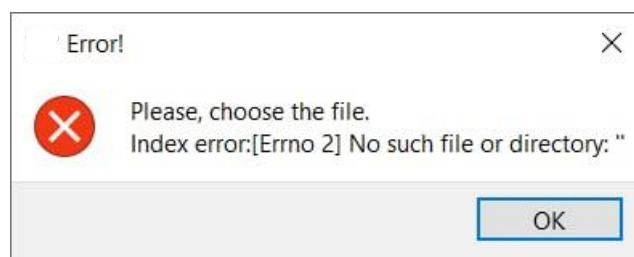


Рисунок 4.5 – Обробка помилки про не відкритий файл

Після обрання необхідного списку даних можемо обрати функцію рейтингу. Ми побачимо вікно з назвами закладів та відповідними значеннями оцінок, які були у файлі «University.xlsl». Дані для оцінювання були взяті з сайту для більш точної оцінки [12].

Кафедра інтелектуальних інформаційних систем
Система оцінювання ЗВО на основі нечіткого гібридного методу прийняття рішень

	Quality	International	Pleasure
ЧНУ ім. Петра Могили	12	8	10
КНУ ім. Тараса Шевченка	36	22	20
НУК ім. Макарова	10	4	7
ЛНУ ім. Івана Франка	13	14	14
ХНУ ім. В.Н. Каразіна	20	14	14
НТУ "Харківський політехнічний інститут"	18	14	11
НТУУ "КПІ ім. Ігоря Сікорського"	36	23	24



Рисунок 4.6 – Список закладів з відповідними оцінками

Далі потрібно натиснути кнопку «Calculate», щоб побачити результат рейтингування за методом TOPSIS. Після цього ми побачимо вікно, яке покаже нам кожен університет із відповідним значення індексу близькості за сортування від найбільшого до найменшого. Університет із значенням 1 буде виділений окремо від інших та пофарбований у зелений колір.

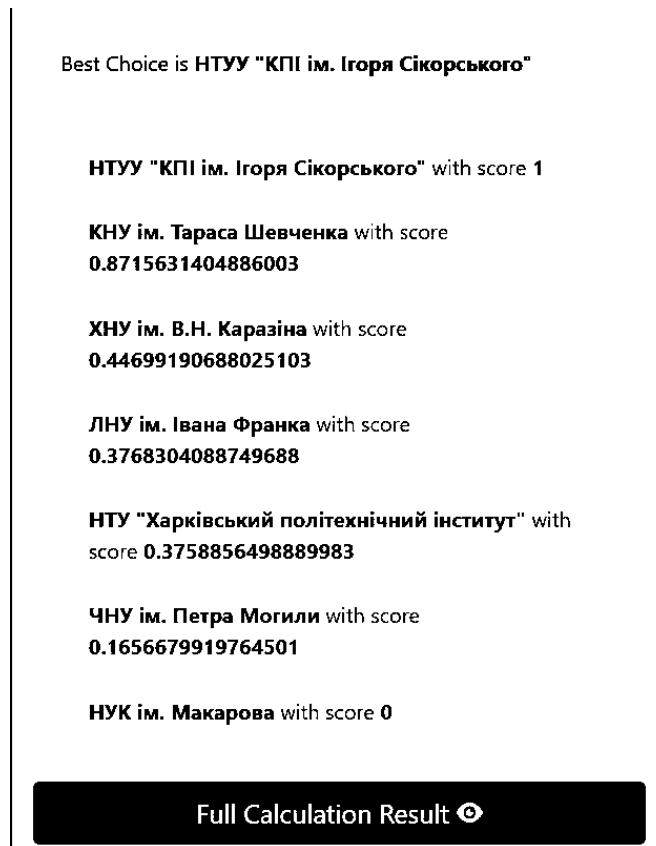


Рисунок 4.7 – Результат рейтингування закладів програмою

Натиснувши кнопку «Full Calculation Result» можна побачити повний розрахунок.

TOPSIS ANALYSIS



Normalized Decision Matrix is [[0.06550339347073006, 0.06504065040650406, 0.0823610351845668], [0.19651018041219018, 0.17886178861788618, 0.1647220703691336], [0.05458616122560839, 0.03252032520325203, 0.05765272462919677], [0.0709620095932909, 0.11382113821138212, 0.11530544925839353], [0.10917232245121677, 0.11382113821138212, 0.11530544925839353], [0.09825509020609509, 0.11382113821138212, 0.09059713870302347], [0.19651018041219018, 0.1869918699186992, 0.1976664844429603]]

Best Answer Vector is [0.19651018041219018, 0.1869918699186992, 0.1976664844429603]

Worst Answer Vector is [0.05458616122560839, 0.03252032520325203, 0.05765272462919677]

Choices Distance From Best Vector is [0.21290895893295964, 0.03393276647468609, 0.25220574594175266, 0.16703185112794847, 0.14058875320760475, 0.16270176252017124, 0]

Choices Distance From Worst Vector is [0.0422759757039362, 0.2302652729648743, 0, 0.10100409527694536, 0.11363673635889514, 0.09799046878525215, 0.25220574594175266]

Closeness Vector of Each Choices [0.1656679919764501, 0.8715631404886003, 0, 0.3768304088749688, 0.44699190688025103, 0.3758856498889983, 1]

Рисунок 4.8 – Повний результат розрахунку за методом

Якщо користувач не має створеного файлу, то він може записати дані вручну за допомогою кнопки «Додати вручну», натиснувши її відкривається форма Client.cs (див. рисунок 4.9).

Рисунок 4.9 – Форма додавання закладів вручну

Додавання нового закладу (див. рисунки 4.10-4.11):

№	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	НКИ	2	1	3	

Рисунок 4.10 – Додавання закладу

Search

Name
HKI

Quality
2

International
1

Pleasure of graduates
3

Rating Save
Export

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	HKI	2	1	3	

Рисунок 4.11 – Додавання даних

При додаванні закладу з однаковою назвою система сповістить, що даний заклад вже існує в базі.

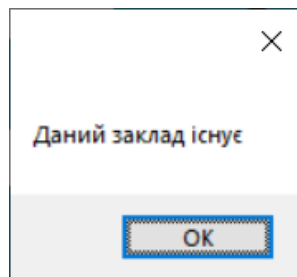


Рисунок 4.12 – Попередження про існування закладу

Також можна видалити заклад, якщо користувач передумав його оцінювати. Видалення закладу – виділяємо дані та натискаємо клавішу «Delete» (див. рисунки 4.13-4.14):

Search

Name
Павл Скоропадський

Quality
3

International
2

Pleasure of graduates
1

Rating Save
Export

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	НКИ	2	1	3	
3	КНУ ім. Тараса Шевченка	4	5	4	
4	Павл Скоропадський	3	2	1	

Рисунок 4.13 – Видалення виділеного запису

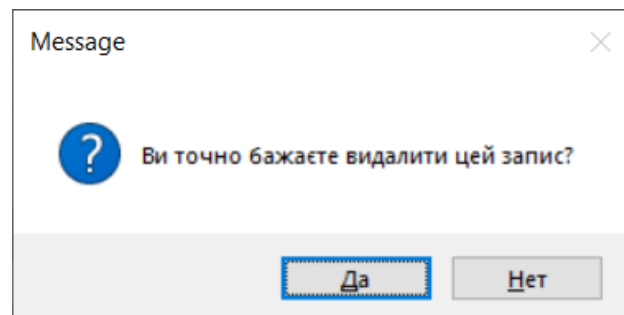


Рисунок 4.14 – Попередження про видалення

Пошук за назвою (див. рисунок 4.15).



Search

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	

Name:

Quality:

International:

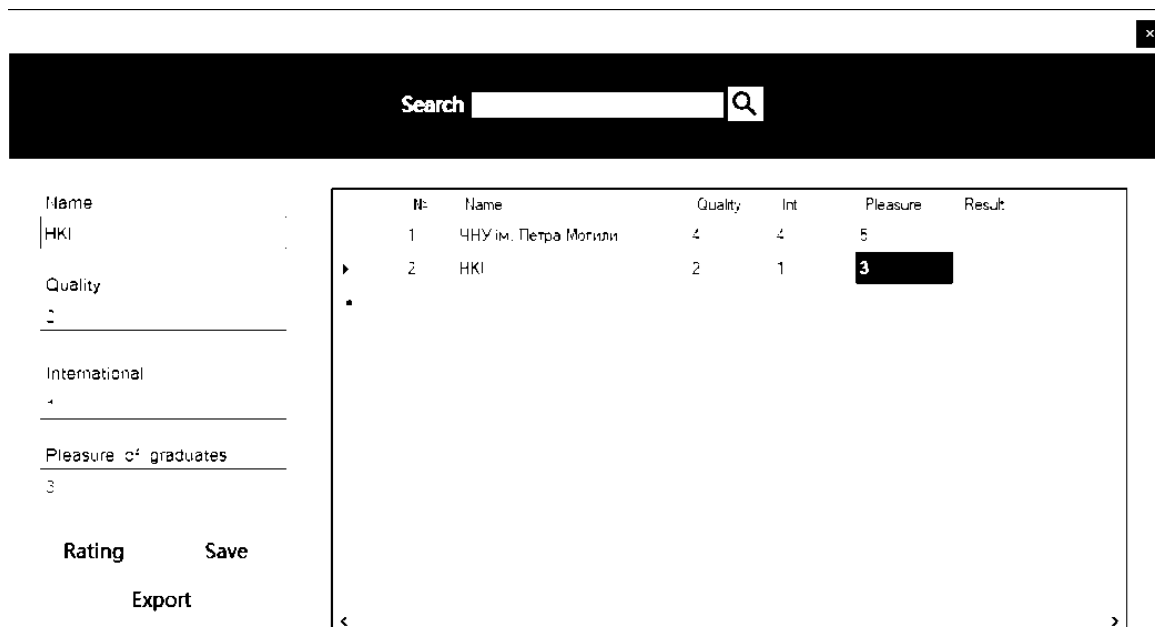
Pleasure of graduates:

Rating Save
Export

Рисунок 4.15 – Пошук закладу за назвою

Коли користувач введе назву в пошуку, у незаповнені поля виведеться уся інформація, яка записана.

Очищення полів (див. рисунки 4.16-4.17).



Search

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	НКИ	2	1	3	

Name:

Quality:

International:

Pleasure of graduates:

Rating Save
Export

Рисунок 4.16 – Форма до очищення

Search

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	НКІ	2	1	3	

Name

Quality

International

Pleasure of graduates

Rating Save Export

Рисунок 4.17 – Очищення полів

Введену інформацію можна редагувати. Редагувати можна як у таблиці, так й у відповідних полях та натискаючи кнопку «Save» (див. Рис. 4.18).

Search

#	Name	Quality	Int	Pleasure	Result
1	ЧНУ ім. Петра Могили	4	4	5	
2	НКІ	2	2	3	
3	Поплавський	1	2	2	

Name

Quality

International

Pleasure of graduates

Rating Save Export

Рисунок 4.18 – Редагування обраного запису

Також можна відсортувати за оцінкою, яка цікава користувачу (див. Рис. 4.19).

#	Name	Quality	Int	Pleasure	Result
3	Поплавський	1	2	2	
2	HKI	2	2	3	
1	ЧНУ ім. Петра Могили	4	4	5	

Рисунок 4.19 – Сортування за бажаною оцінкою

Після додавання необхідних для аналізу закладів користувачу потрібно натиснути кнопку «Rating» і після цього він перейде до форми рейтингування.

	Quality	International	Pleasure
ЧНУ ім. Петра Могили	4	4	5
HKI	2	1	3

Best Choice is **ЧНУ ім. Петра Могили**

ЧНУ ім. Петра Могили with score
0.09937518539672312

HKI with score **0**

Full Calculation Result

Рисунок 4.20-21 – Рейтингування введених даних

Якщо користувач натисне кнопку «минулі дані», то він попаде до розрахунку минулих закладів, що були додати або через імпорт файлу, або 2022 р.

введені вручну. В нашому випадку останніми були дані введені вручну.

TOPSIS ANALYSIS

×

```
Normalized Decision Matrix is [ [ 0.027747685928647813, 0.03931785497463923,
0.048683853153082164 ], [ 0.013873842964323906, 0.009829463743659807,
0.0292103118918493 ], [ 0.06936921482161953, 0.03931785497463923, 0.06815739441431502
], [ 0.09017997926810539, 0.1376124924112373, 0.13631478882863005 ], [
0.13873842964323907, 0.1376124924112373, 0.13631478882863005 ], [ 0.12486458667891516,
0.1376124924112373, 0.10710447693678077 ], [ 0.24972917335783032, 0.2260776661041756,
0.2336824951347944 ] ]
Best Answer Vector is [ 0.24972917335783032, 0.2260776661041756, 0.2336824951347944 ]
Worst Answer Vector is [ 0.013873842964323906, 0.009829463743659807, 0.0292103118918493 ]
Choices Distance From Best Vector is [ 0.3440632287087508, 0.37973661348271537,
0.3079082666393493, 0.20679096334831928, 0.17212060416511235, 0.19859316869911395, 0
]
Choices Distance From Worst Vector is [ 0.03796402962335483, 0, 0.0739335969095416,
0.18336438539959607, 0.20830515212508943, 0.18631947980236532, 0.37973661348271537 ]
Closeness Vector of Each Choices [ 0.09937518539672312, 0, 0.19362360172452692,
0.4699778844197527, 0.5475579628372668, 0.48405652692301504, 1 ]
```

Рисунок 4.22 – Повний аналіз минулих даних

Висновки до розділу 4

Отже, в четвертому розділі зображено створену систему оцінювання ЗВО на основі методу TOPSIS. Визначено та протестовано основні функції розробленого модулю: рейтингування ЗВО. Показано результати роботи програмного забезпечення.

Спеціальний розділ

ОХОРОНА ПРАЦІ

до кваліфікаційної роботи

на тему:

«СИСТЕМА ОЦІНЮВАННЯ ЗВО НА ОСНОВІ НЕЧІТКОГО ГІБРИДНОГО МЕТОДУ ПРИЙНЯТТЯ РІШЕНЬ»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.21810103

Виконав студент 4-го курсу, групи 401

_____ ***Я.Ю. Бєлоусова*** _____

(підпис, ініціали та прізвище)

«___» _____ 202_ р.

Консультант _____старший викладач_____

_____ ***О.В. Макарова*** _____

(підпис, ініціали та прізвище)

«___» _____ 202_ р.

5 ОХОРОНА ПРАЦІ

Неправильна організація робочого місця сприяє загальній та локальній напрузі м'язів спини, шиї, тулуба та розвитку остеохондрозу. Також неправильна посадка та освітлення робочого місця сприяє погіршенню зору та розвитку зорових хвороб. [19]

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Повинні бути впроваджені сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників.

Метою написання даного розділу дипломної роботи є створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях. Нижче наведено шляхи оцінки та створення комфортних умов праці на робочому місці:

- визначення оптимальних умов праці;
- аналіз робочого місця;
- розрахунок освітлення приміщення;
- оцінка шкідливих чинників та їх усунення;
- організація робочого місця;

5.1 Аналіз приміщення

Створення дипломної роботи проходило у приміщенні відповідної установи. Приміщення має сучасний ремонт. Стіни приміщення мають приємний жовтий колір. Розміри приміщення складають 12м². Біля робочого столу знаходиться велике вікно розмірами 300*400 см, яке має металеві жалюзі для запобігання потраплянню надлишкового сонячного світла.

Згідно з розміром площі для одного робочого місця оператора

2022 р. Белоусова Я.Ю. – БКР – 401.21810103

персонального комп'ютера має бути не менше 6м², а об'єм - не менше 20 м³. Отже, дане приміщення цілком відповідає зазначеним нормам. У приміщенні розташоване одне робоче місце обладнане ПК з рідкокристалічним дисплеєм і приєднане до локальної мережі. Все обладнання підключене до джерела живлення під напругою в 220 В.

Задля дотримання мікроклімату, у приміщенні встановлена система опалення, але відсутня система кондиціонування та вентиляції. Освітлюється приміщення LED-лампою, яка розташована на стелі, та світильником, який розташований безпосередньо над робочим місцем [19].

5.2 Напруженість праці користувача

Під час роботи програміста відбувається значна розумова та фізична напруга на людину. Задля запобігання негативного впливу на м'язи, у приміщенні встановлений куточок для фізичних тренувань, де працівник може розім'яти м'язи [3]. Також у приміщенні знаходиться ліжко на якому можна відпочити. У процесі роботи необхідно робити невеликі паузи для відпочинку та зняття фізичної та психологічної напруги. Нижче наведена таблиця, яка містить відомості про регламентовані перерви в залежності від тривалості робочого дня.

Таблиця 5.1 – Час регламентованих перерв під час роботи за ПЕОМ

Категорія роботи	Рівень навантаження за робочу зміну при видах роботи з ВДТ			Сумарний час регламентованих перерв, мін	
	Група А, кількість знаків	Група Б, кількість знаків	Група В, годин	При 8-годинній зміні	При 12-годинній зміні
I	до 20000	до 15000	до 2,0	30	70
II	до 40000	до 30000	до 4,0	50	90
III	до 60000	до 40000	до 6,0	70	120

5.3 Рівень освітлення

В приміщеннях, в яких експлуатуються персональні комп'ютери, повинно бути природне та штучне освітлення. Природне освітлення – освітлення приміщень денним світлом, що потрапляє через світлові прорізи, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%.

Використовуваний для даних цілей спеціалізований прилад зветься люксометром, але у наш час виміряти цей параметр вже можна навіть за допомогою програми на смартфоні. Освітленість може бути заміряна як для світла, що дається одним світильником, так і для приміщення у цілому.

Luxmeter

З допомогою цієї програми можна виміряти освітленість якогось певного місця (наприклад, кімнати або робочого столу) і порівняти результати з стандартами. Якщо світла мало — пора міняти лампочку на потужнішу, щоб не псувати постійно очі при роботі.



Рисунок 5.1 – Головне меню програми та приклад використання

У головному вікні — три пункти меню: «Стандарти освітленості», «Вимірювання» і «Історія». В першому розділі змиритися освітленість і порівнюється зі стандартними значеннями. Якщо все добре, і світла достатньо, в рядках стандартів з'являються відповідні позначки.

Вимірювання проводяться трьома методами: постійно, одноразово і по таймером. Перші два варіанти пояснювати не потрібно, а ось вимір по таймером означає всього лише затримку початку виміру в 10 секунд. Треба думати, щоб встигнути включити всі лампочки.

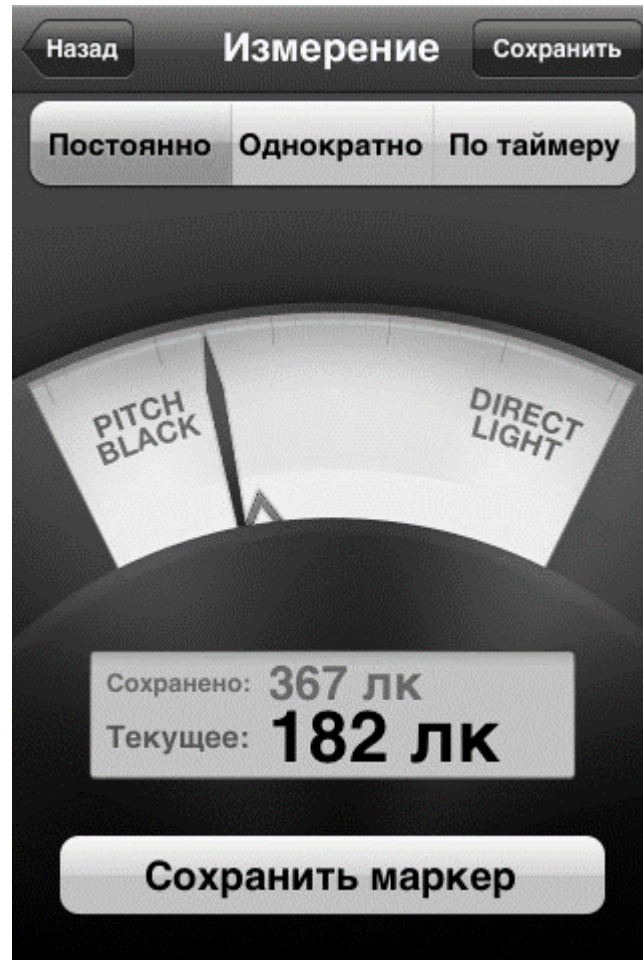


Рисунок 5.2 – Вимірювання

На екрані «Вимірювання» немає порівняння з стандартними значеннями, зате є зображення вимірювального приладу, того самого люксметра. Варіанти вимірювань такі ж, як і на попередньому екрані. Можна встановити на шкалою приладу маркер, що відзначає, наприклад, результат попереднього вимірювання (до заміни лампочки на нову) — зручна функція.

Штучне освітлення – застосовується при роботі в темний час доби і вдень, коли не вдається забезпечити нормовані значення коефіцієнта природного освітлення (похмура погода, короткий світловий день). Штучне освітлення в приміщеннях з робочими місцями має здійснюватися системою загального рівномірного освітлення [21]. Штучне освітлення підрозділяється на робоче, аварійне, евакуаційне, охоронне. Робоче освітлення, у свою чергу, може бути загальним або комбінованим. Загальне освітлення, при якому світильники розміщуються у верхній зоні приміщення рівномірно, або, як розташоване

2022 р.

устаткування.

При виконанні робіт категорії високої зорової точності величина коефіцієнта природного освітлення повинна бути не нижче 1,5%, а при зоровій роботі середньої точності – не нижче 1,0%. В якості джерел штучного освітлення звичайно використовуються люмінесцентні лампи типу ЛБ, або LED, які попарно об'єднуються в світильники, які повинні розташовуватися рівномірно над робочими поверхнями.

Для збереження зору працівників у хорошому стані необхідно робити невеликі перерви, щоб дати очам відпочити, а також дотримуватися певної відстані, на якому працівники повинні перебувати від комп'ютера. Також щоб не відбувалося напруження зору все поле зору повинне бути освітлено достатньо рівномірно – ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

В даному приміщенні використовується комбінація штучного та природного світла. При виборі місця для встановлення світильника можна знехтувати проблемою відблиска світла від монітору, оскільки монітор має матове антиблікове покриття. Розрахунок освітлення проводиться для кімнати площею 12 м² та об'ємом 36 м³. Розміри кімнати: довжина – 6 м, ширина – 2 м, висота – 3 м. Для розрахунку кількості світильників використаємо формулу світлового потоку (4.1).

$$F = E \cdot k \cdot S \cdot Z, \quad (5.1)$$

де F – розраховується світловий потік, Лм;

E – нормована мінімальна освітленість, $E = 400$ Лм;

k – коефіцієнт запасу, враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації $K = 1,5$;

S – площа освітлюваного приміщення $S = 12$ м²;

Z – відношення середньої освітленості до мінімальної $Z = 1,1$; η – коефіцієнт використання світового потоку від світильника.

Значення η визначимо по таблиці коефіцієнтів використання різних

світильників. Для цього обчислимо індекс приміщення за формулою 4.2.

$$I = Sh \cdot (A+B), \quad (5.2)$$

де S – площа приміщення, $S = 12 \text{ м}^2$

h – розрахункова висота підвісу, $h = 2.9 \text{ м}$;

A – ширина приміщення, $A = 2 \text{ м}$;

B – довжина приміщення, $B = 6 \text{ м}$.

Підставивши значення у формулу 4.2 отримаємо індекс для заданого приміщення.

$$I = 122,9 \cdot (2+6) = 0,517. \quad (5.3)$$

Знаючи індекс приміщення I , знаходимо $\eta = 0,57$.

Підставимо всі значення у формулу для визначення світлового потоку F .

$$F = 400 \cdot 1,5 \cdot 12 \cdot 1,10,57 = 13\,894. \quad (5.4)$$

У світильнику у робочому приміщенні використовується світлодіодна лампа на 60 Вт, світловий потік якої 2500 Лм.

Розрахуємо необхідну кількість ламп за формулою 4.5.

$$N = F \cdot F_{\text{л}}, \quad (5.5)$$

де N – обумовлене число ламп;

F – світловий потік, $F = 13\,894 \text{ Лм}$;

$F_{\text{л}}$ – світловий потік лампи, $F_{\text{л}} = 2500 \text{ Лм}$.

Виконавши розрахунки було встановлено, що для приміщення необхідно одиниць світильників, який містить 5-6 ламп.

5.4 Мікроклімат

Мікроклімат приміщення – це клімат приміщення, що складається з температури та вологості повітря. Робота над дипломним проектом відбувалася сидячи за робочим столом та не потребувала фізичних навантажень, тому категорія приміщення Іа. Комп'ютери випромінюють велику кількість тепла, що

сприяє підвищенню середньої температури приміщення та зниженню відносної вологості повітря [25].

Оптимальні значення для температури, відносної вологості та швидкості руху повітря у приміщенні, зазначенні у таблиці 5.2.

Таблиця 5.2 – Норми мікроклімату робочого приміщення

Період року	Категорія робіт	Температура С ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22-24	40-60	0,1
Тепла	легка-1 а	23-25	40-60	0,1

Для забезпечення оптимальної відносної вологості повітря у приміщенні був використаний ультразвуковий зволожувач повітря DELFA DH-410 G [23].

У приміщенні відсутня система кондиціонування, тому для провітрювання використовувався природний протяг з вулиці. Недоліком використання кондиціонування за допомогою відкритих вікон є підвищення рівню шуму у приміщенні та підвищення швидкості руху повітря.

5.5 Ергономіка робочого місця

Головними елементами робочого місця програміста є робочий стіл та крісло. Згідно з ДСТУ 7299:2013 [22] конструкція робочого місця і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. При побудові робочого місця слід враховувати положення людини відносно столу, висоту столу, висоту крісла, кут нахилу монітора та відстань від очей до монітора [22].

Основне положення при роботі за комп'ютером – це сидяче, тому необхідно обрати крісло, яке відповідає ортопедичним нормам, та викликає мінімальні дефекти хребта та м'язів спини та шиї. Також необхідно відрегулювати висоту крісла відносно столу так, щоб кут нахилу голови до

монітору був 30°, та відстань до очей не менше 30 см.

Тепер необхідно створити таблицю (Табл. 5.3) порівняння характеристик робочого місця до відповідних вимог.

Робочий стіл має кутову форму із розмірами 150*150 см, має достатньо місця для ніг та висоту 75 см. Поверхня сидіння м'яка, передній край закруглює, а кут нахилу спинки - регульований. Крісло, що використовується в якості робочого сидіння, є підйомно-поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке

і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м (регулюється по висоті +3 см. 2, по нахилу від -10° до +20° щодо вертикалі).

Таблиця 5.3 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше
Глибина простору для ніг, мм	700	не менше
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

5.6 Гігієна праці та виробнича санітарія

Гігієна праці та виробнича санітарія вивчає виробничі шкідливі чинники з метою розробки та реалізації заходів для забезпечення безпечних умов праці людини. Гігієна праці вивчає чинники та вплив навколишнього та виробничого середовищ на організм людини з метою розробки заходів задля уникнення шкідливого впливу на організм та здоров'я людини.

Одним із найважливіших чинників є рівень шуму у приміщенні. Шум на робочому місці може бути спричинений звуками з вулиці, шумом від роботи комп'ютера та іншої офісної техніки та шумом від спілкування людей, які можуть знаходитися у приміщенні [23]. Вимірявши рівень шуму, можна побачити, що рівень шуму у приміщенні відповідає нормі. А саме:

- мінімальний рівень шуму – 38 дБ;
- максимальний рівень шуму – 64 дБ;
- середній показник – 47 дБ.

Висновки до розділу 5

Вивчено основні вимоги до організації робочого місця при роботі над проектом. Розраховано оптимальну кількість джерел штучного світла для освітлення даного приміщення. У результаті визначено, що для освітлення приміщення необхідно встановити 5-6 світлодіодних ламп світимістю 2500 Лм. Запропоновані світлодіодні світильники мають строк служби 50 тисяч годин, що значно краще ніж у люмінесцентних ламп, де строк рівний 10–20 тисяч годин, і крім того залежить від кількості переключень.

Проведено аналіз робочого місця. А саме, аналіз мікроклімату, ергономічності, освітлення та рівню шуму. У результаті дослідження виявлено рівень шуму у приміщенні знаходиться у допустимих межах. Також виявлено, що відсутність системи кондиціонування негативно впливає на мікроклімат робочого приміщення.

ВИСНОВКИ

В результаті виконання бакалаврської кваліфікаційної роботи створена систему оцінювання ЗВО на основі нечіткого гібридного методу з виконанням усіх поставлених завдань. Виконано аналіз предметної області, досліджено усі специфікації до вимог розробки ПЗ. Виконано порівняння існуючих аналогів, досліджені переваги та недоліки кожного з них. Реалізовано усі функції, що вирішують поставлені проблеми для користувачів. Виконано порівняльний аналіз з іншим нечітким методом. Описано усі застосовані методи та засоби розробки. Отже, усі поставлені завдання виконані успішно з дотримання усіх поставлених вимог специфікації.

Використовуючи створену систему користувач значно зекономить свій час у вирішенні проблеми пошуку найкращого ЗВО для своїх уподобань.

Розроблений застосунок зручний та зрозумілий у використанні. Усі данні користувача захищені, а це дуже важливо в сучасному світі. Для розробки застосунка використано лише сучасні технології та обрана ефективна архітектура для досягнення мети.

Вибір ЗВО для абітурієнтів дуже непросто задача, адже кожен має свої мрії та захоплення. Створену систему можливо інтегрувати в будь-який пристрій. А завдячуючи цьому можливо надати результат для користувача будь-де та будь-коли. Це допоможе вирішити, який заклад краще обрати для абітурієнта.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рейтингові системи ранжування вищих навчальних закладів [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/bitstream/123456789/114/1/10.pdf>
2. Рейтинг «ТОП-200 Україна» 2018 года [Електронний ресурс]. – Режим доступу: <https://osvita.ua/vnz/rating/60985/>
3. Рейтинг «Компас» года [Електронний ресурс]. – Режим доступу: <https://www.yourcompass.org/>
4. Центр міжнародних проєктів НДІ прикладних інформаційних технологій Євроосвіта [Електронний ресурс]. – Режим доступу: <http://www.euroosvita.net/?category=19&id=239>
5. T. Yang, C.C. Hung, Multiple-attribute decision-making methods for plant layout design problem, Robotics and Computer-integrated manufacturing 23 (2007) 126–137.
6. L. De Boer, E. Labro, P. Morlacchi, A review of methods supporting supplier selection, Eur. J. Purch. Supply Manage. 7 (2001) 75–89.
7. T.L. Saaty, The Analytic Hierarchy Process, first ed., McGraw Hill, New York, 1980
8. Chou, Y.-C.; Sun, C.-C.; Yen, H.-Y. Evaluating the criteria for human resource for science and technology based on integrated fuzzy AHP and fuzzy DEMATEL approach. Appl. Soft Comput. 2012, 12, 64–71.
9. D.A.R. Carrera, V. Mayorga, Supply chain management: a modular fuzzy inference system approach in supplier selection for new product development, Int. J. Intell. Manuf. 19 (2008) 1–12.
10. H.L. Kwang, First Course on Fuzzy Theory and Applications, Springer-Verlag, Berlin-Heidelberg/New York, 2005, pp. 137–144.
11. Ендрю Троелсен — Мова програмування C# 5.0 та платформа .Net 4.5 / 2013, – 1312с.

12. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft
13. Гурь Є.Ю. Windows Forms Application Tutorial with Example [Електронний ресурс] – Режим доступу: <https://www.guru99.com/c-sharp-windows-forms-application.html>.
14. Діго С.М. Проектування і використання баз даних. - М .: Фінанси і статистика, 1995.
15. Джеймс Р.Г. SQL / Р.Г. Джеймс, Н.В. Пол – Львів, 2019. – 960 с.
16. Buckley, J.J. Fuzzy hierarchical analysis. Fuzzy Sets Syst. 1985, 17, 233–247. [CrossRef]
17. Boudewijn Rempt. GUI Programming with Python: QT Edition. — OpenDocs, 2002.
18. Mark Summerfield. Rapid GUI Programming with Python and Qt. — 1st. — Prentice Hall, 2008. — P. 648.
19. Організація робочого місця для виконання ручних робіт [Електронний ресурс]. — Режим доступу: <https://uahistory.co/pidruchniki/sydorenko-labor-training-for-girls-5-class-2013/11.php>
20. Вентиляції та кондиціювання повітря у робочій зоні виробничих приміщень [Електронний ресурс]. – Режим доступу: <https://dsp.gov.ua/ventyliatsii-ta-kondytsiiuvannia-povitria-u-robochii-zoni-vyrobnychkh-prymishchen/>
21. СНиП II-4-79. Природне і штучне освітлення [Електронний ресурс] – Режим доступу: https://dnaop.com/html/45036/doc-СНиП_II-4-79
22. ДСТУ 7299:2013 Дизайн і ергономіка. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки. Київ -1999 р.
23. Наказ Про затвердження Державних санітарних норм допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови
24. DELFA DH-410 G – Режим доступу: <https://bt.rozetka.com.ua/delfa-dh->

[410-g/p395969/characteristics/](https://www.sop.com.ua/article/82-osnovn-vimogi-ta-zahodi-z-normalzats-mkroklmatichnih-umov-na-robochih-mstsyah)

25. Основні вимоги та заходи з нормалізації мікрокліматичних умов на робочих місцях [Електронний ресурс]. – Режим доступу:

<https://www.sop.com.ua/article/82-osnovn-vimogi-ta-zahodi-z-normalzats-mkroklmatichnih-umov-na-robochih-mstsyah> - Загол. з екрану.

ДОДАТОК А

Порівняння нечітких методів АНР та TOPSIS

```
import org.apache.commons.math3.linear.Array2DRowRealMatrix;
import org.apache.commons.math3.linear.EigenDecomposition;
import org.apache.commons.math3.linear.RealVector;

public class AHP {

    // Singleton instance
    private static AHP instance;

    // Random Consistency Index
    private static double[] RI = {0.0, 0.0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45,
1.49};

    // The matrix
    private Array2DRowRealMatrix mtx;

    // Pairwise Comparison Array of criteria
    private double[] pairwiseComparisonArray;

    // Number of alternatives
    private int nrAlternatives;

    // The resulting weights/priorities
    private double[] weights;

    // Corresponds to the weights
    private String[] labels = null;

    private AHP(int nrAlternatives) {
        this.nrAlternatives = nrAlternatives;
        mtx = new Array2DRowRealMatrix(nrAlternatives, nrAlternatives);
        weights = new double[nrAlternatives];

    /**
     * Set the pairwise comparison scores and calculate all relevant numbers
     * @param a
     */
    void setPairwiseComparisonArray(double[] a) {
        int i = 0;
        for (int row = 0; row < nrAlternatives; row++) {
            for (int col = row + 1; col < nrAlternatives; col++) {
                //System.out.println(row + "/" + col + "=" + a[i]);
                getMtx().setEntry(row, col, a[i]);
                getMtx().setEntry(col, row, 1.0 / getMtx().getEntry(row, col));
                comparisonIndices[i][0] = row;
                comparisonIndices[i][1] = col;
                i++;
            }
        }
    }

    /**
     * Sets the EigenDecomposition
     */
    void setEvd(){
        evd = new EigenDecomposition(getMtx());
    }
}
```

```

        evIdx = 0;
        for (int k = 0; k < evd.getRealEigenvalues().length; k++) {
            //System.out.println(evd.getRealEigenvalues()[k]);
            evIdx = (evd.getRealEigenvalue(k) > evd.getRealEigenvalue(evIdx)) ? k :
evIdx;
        }
        //System.out.println("evIdx=" + evIdx);
        //System.out.println("EigenValue=" + evd.getRealEigenvalue(evIdx));

        double sum = 0.0;
        RealVector v = evd.getEigenvector(evIdx);
        for (double d : v.toArray()) {
            sum += d;
        }
        //System.out.println(sum);
        for (int k = 0; k < v.getDimension(); k++) {
            weights[k] = v.getEntry(k) / sum;
        }
    }

    /**
     *
     * @param arrayIdx
     * @return
     */
    int[] getIndicesForPairwiseComparison(int arrayIdx) {
        return comparisonIndices[arrayIdx];
    }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        for (int i=0; i<nrAlternatives; i++)
            sb.append(getMtx().getRowVector(i)).append("\n");
        return sb.toString();
    }

    /**
     * Created for the GDM to assign A_i to mtx
     * @return mtx
     */
    public Array2DRowRealMatrix getMtx() {
        return mtx;
    }

    /**
     * Created for setting Ac to mtx
     * @param mtx
     */
    public void setMtx(Array2DRowRealMatrix mtx) {
        this.mtx = mtx;
    }
}

class Topsis {

    private TreeMap<String, ArrayList<Double>> sitesMatrix;
    private TreeMap<String, ArrayList<Fuzzy>> availableSites;

    Topsis(){
        sitesMatrix = new TreeMap<>();
        availableSites = new TreeMap<>();
    }
}

```

```

void start(){

    Profiler p = new Profiler();
    availableSites = p.start();

    sitesMatrix = transformToFuzzyValues(availableSites);
    TreeMap<String, ArrayList<Double>> normalisedWeightedFuzzyMatrix =
calculateWeightedFuzzyMatrix(sitesMatrix);

    TreeMap<String, Double> idealDistances =
calculateDistance(normalisedWeightedFuzzyMatrix, true);
    TreeMap<String, Double> antiIdealDistances =
calculateDistance(normalisedWeightedFuzzyMatrix, false);

    TreeMap<Double, String> ccStar = calculateRelativeCloseness(idealDistances,
antiIdealDistances);

    System.out.println("*****");
    System.out.println("Final Ranking:");

    for (Map.Entry<Double, String> entry: ccStar.entrySet()){
        System.out.println(entry.getValue() + ": " +
Config.df.format(entry.getKey()));
    }
}

private TreeMap<String, ArrayList<Double>> transformToFuzzyValues(TreeMap<String,
ArrayList<Fuzzy>> availableSites) {

    for (Map.Entry<String, ArrayList<Fuzzy>> entry: availableSites.entrySet()) {

        System.out.println(entry);

        ArrayList<Double> fuzzyMatrix = new ArrayList<>();

        for (int k = 0; k < entry.getValue().size(); k++) {
            for (double fuzzyValue:entry.getValue().get(k).getValue()) {
                fuzzyMatrix.add(fuzzyValue);
            }
        }

        sitesMatrix.put(entry.getKey(), fuzzyMatrix);
    }
    System.out.println("unweighted fuzzy values: " + sitesMatrix);
    return sitesMatrix;
}

```

ДОДАТОК Б

Система оцінювання

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp2
{
    public partial class Menu : Form
    {
        public Menu()
        {
            InitializeComponent();
        }

        private void client_Click(object sender, EventArgs e)
        {
            this.Visible = false;
            Clients obj1 = new Clients();
            obj1.ShowDialog();
        }

        private void insert_Click(object sender, EventArgs e)
        {
            this.Visible = false;
            Insert obj2 = new Insert();
            obj2.ShowDialog();
        }

        private void past_Click(object sender, EventArgs e)
        {
            this.Visible = false;
            Settlement obj3 = new Settlement();
            obj3.ShowDialog();
        }

        private void import_Click(object sender, EventArgs e)
        {
            this.Visible = false;
            Import obj4 = new Import();
            obj4.ShowDialog();
        }

        private void exitmenu_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```

```

}
public partial class ManageClients : Form
{
    public ManageClients()
    {
        InitializeComponent();
    }

    public Boolean existsClient()
    {
        DB db = new DB();
        DataTable table = new DataTable();
        MySqlDataAdapter adapter = new MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("SELECT * FROM client WHERE
name = @name = @last", db.getConnection());

        command.Parameters.Add("@name", MySqlDbType.VarChar).Value = name.Text;

        adapter.SelectCommand = command;
        adapter.Fill(table);

        if (table.Rows.Count > 0)
        {
            MessageBox.Show("Заклад існує");
            return true;
        }
        else
        {
            return false;
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Menu obj1 = new Menu();
        obj1.Show();
        this.Close();
    }

    private void cleanclient_Click(object sender, EventArgs e)
    {
        name.Text = "";
        last.Text = "";
        ph.Text = "";
        cou.Text = "";
    }

    private void save_Click(object sender, EventArgs e)
    {
        try
        {
            existsClient();
        }
    }
}

```

```

        clientBindingSource.EndEdit();
        clientTableAdapter.Update(this.appData.client);

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void cancel_Click(object sender, EventArgs e)
{
    clientBindingSource.ResetBindings(false);
}

private void dataGridView1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Delete)
    {
        if (MessageBox.Show("Ви точно бажаєте видалити цей запис?",
        "Message", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            clientBindingSource.RemoveCurrent();
    }
}

private void button1_Click_1(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(find.Text))
    {
        this.clientTableAdapter.Fill(this.appData.client);
        clientBindingSource.DataSource = this.appData.client;
    }
    else
    {
        var query = from o in this.appData.client
                     where o.name.Contains(find.Text) == find.Text
                     select o;
        clientBindingSource.DataSource = query.ToList();
    }
}

}
public class Topsis {

    private List<Alternative> alternatives;

    //Attributes below are all populated from calculateOptimalSolution method
    private List<Criteria> criteria;
    private int numberOfAlternatives = 0;
    private int numberOfCriteria = 0;
    private double[][] scoreMatrix;
    private double[][] normalizedDecisionMatrix;
    private double[] idealBest;
    private double[] idealWorst;
    private double[] distancesFromIdealWorst;
    private double[] distancesFromIdealBest;

    public Topsis(List<Alternative> alternatives) {
        super();
        this.alternatives = alternatives;
    }
}

```

```

    }

    public Topsis() {
        super();
    }

    public void addAlternative(Alternative alternative) {
        if (alternatives == null) {
            alternatives = new ArrayList<Alternative>();
        }
        this.alternatives.add(alternative);
    }

    public List<Alternative> getAlternatives() {
        return alternatives;
    }

    public List<Alternative> calculateOptimalSolutionSortedList() throws
TopsisIncompleteAlternativeDataException {

        validateData();
        populateScoreMatrix();
        calculateNormalizedDecisionMatrix();
        findIdealBestAndWorst();
        calculateDistancesFromIdealBestAndWorst();
        calculatePerformanceScore();
        sortAlternativesByPerformanceScoreDesc();

        return alternatives; // Sorted result from the ideal solution to the
worse one.
    }

    public Alternative calculateOptimalSolution() throws
TopsisIncompleteAlternativeDataException {
        return this.calculateOptimalSolutionSortedList().get(0); // Top
solution after
        public void writeResultsIntoCSVFile(String path, String fileName ) throws
Exception {

            if (this.alternatives.isEmpty()) throw new Exception("Alternatives
should not be empty");

            final Writer writer = new BufferedWriter(new OutputStreamWriter(
                new FileOutputStream(path + fileName + ".csv"), UTF_8));

            StringBuilder header = new StringBuilder();
            header.append("Alternative name").append(";")
                .append("Criteria values").append("\n");

            writer.write(header.toString());

            for (Alternative alternative : this.alternatives) {
                writer.write(alternative.getName() + ";" +
alternative.getCriteriaValues().toString() + "\n");
            }

            writer.close();
        }
    }

```

```

private void calculatePerformanceScore() {

    for (int a = 0; a < numberOfAlternatives; a++) {
        double performanceScore = distancesFromIdealWorst[a] /
        (distancesFromIdealBest[a] + distancesFromIdealWorst[a]);

        alternatives.get(a).setCalculatedPerformanceScore(performanceScore);
    }

private void sortAlternativesByPerformanceScoreDesc() {

    Collections.sort(alternatives, new Comparator<Alternative>() {

        public int compare(Alternative a1, Alternative a2) {
            return Double.compare(a2.getCalculatedPerformanceScore(),
            a1.getCalculatedPerformanceScore());
        }

    });
}

@SuppressWarnings("unused")
private void printIdealBestAndWorst() {

    System.out.print("Ideal best is : [ ");
    for (double d : idealBest) {
        System.out.print(d + " ");
    }
    System.out.println("]");

    System.out.print("Ideal worst is : [ ");
    for (double d : idealWorst) {
        System.out.print(d + " ");
    }
    System.out.println("]");
}
}

```