

Міністерство освіти і науки України
Львівський національний університет природокористування
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

«Розробка інформаційної системи підтримки рішень із планування
парку комбайнів агрофірми»

Виконав: здобувач групи Іт-61
спеціальності 126
«Інформаційні системи та технології»

Шоп'як Б.І.

Керівник:

Запорожцев С.Ю.

Рецензент:

Семерак В.М.

ЛЬВІВ-2024

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

Другий (магістерський) рівень вищої освіти
Спеціальність 126 – „Інформаційні системи та технології”

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 202_ р.

ЗАВДАННЯ

на кваліфікаційну роботу здобувачу

Шоп’як Богдан Ігорович

1. Тема роботи: «Розробка інформаційної системи підтримки рішень із планування парку комбайнів агрофірми»

Керівник роботи Запорожцев Сергій Юрійович, к.т.н., доцент.

Затверджені наказом по університету від _____ 202_ року № _____

2. Строк подання здобувачем роботи 2.01.2024 р.

3. Вихідні дані до роботи: Загальні відомості про інформаційні системи підтримки прийняття рішень, варіанти підходів та інструментарію до вирішення задачі

4. Зміст розрахунково-пояснювальної записки:

Вступ

1. Аналіз стану питання та постановка завдання. Цифровізація сільськогосподарського виробництва. Системи підтримки прийняття рішень. Процес прийняття рішень, складові та завдання системи. Постановка задачі на розробку.

2. Обґрунтування, вибір та реалізація інструментарію вирішення задачі. Вибір методу прийняття рішень. Обґрунтування критеріїв оцінки альтернатив. Вибір мови програмування, середовища розробки та СУБД

3. Результати вирішення задачі. Структура інформаційної системи підприємства. Розробка структурної ієрархічної моделі вибору комбайнів. Програмна реалізація моделі. Підготовка та формування матриць порівнянь груп та критеріїв. Матриці порівнянь альтернатив по окремим критеріям. Розрахунки по ієрархічним рівням. Організація обміну з базою даних. Елементи графічного інтерфейсу користувача.

4. Охорона праці та безпека у надзвичайних ситуаціях

5. Визначення ефективності інформаційної системи.

Висновки та пропозиції.

Список використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): Мета роботи та аналіз предметної області. Інструментарій вирішення задачі. Структурна схема ІС. Структурна ієрархічна модель. Парні порівняння та обчислення вагових коефіцієнтів на різних рівнях ієрархії. Узагальнення та вибір альтернативи з найбільшою вагою. Демонстрація роботи програми моделювання. Обмін з БД та елементи графічного інтерфейсу. Висновки.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Запорожцев С.Ю., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання __ _____ 202_ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	1.09 - 21.09.23	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	22.09 - 15.10.23	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	16.10 - 5.11.23	
4.	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	6.11 - 27.11.23	
5.	<i>Написання розділу «Визначення ефективності...»</i>	28.11 - 11.12.23	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	12.12 - 23.12.23	
7.	<i>Завершення роботи в цілому</i>	24.12.23 - 2.01.24	

Студент _____ Шоп'як Б.І.
(підпис)

Керівник роботи _____ Запорожцев С.Ю.
(підпис)

РЕФЕРАТ

УДК 004.9 : 631.354.2

Розробка інформаційної системи підтримки рішень із планування парку комбайнів агрофірми

Шоп'як Б.І. Кафедра ІТ – Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 81 с. текст. част., 44 рис., 2 табл., 10 арк. ілюстраційного матеріалу, 30 джерел.

Об'єкт дослідження – автоматизація процесу прийняття рішень по вибору комбайнів підприємства агропромислового комплексу.

Мета роботи – розробка інформаційної системи підтримки рішень із планування парку комбайнів агрофірми.

Проведений аналіз предметної області та класифікована інформаційна система, розглянуті підходи до побудови таких систем. Описані методи вирішення задачі та обрано метод аналізу ієрархій. На прикладах розповсюджених комбайнів обґрунтовані критерії оцінки альтернатив. Зроблений вибір мови програмування, середовища розробки та СУБД. Розроблена структурна схема інформаційної системи. Створена структурна ієрархічна модель вибору комбайнів. Проведена реалізація моделі обраним інструментарієм та перевірена її робота на навчальних даних. Описана організація обміну даними між програмною частиною та БД. Запропоновані окремі елементи графічного інтерфейсу користувача. Розглянуті питання охорони праці та визначення економічної ефективності отриманих рішень.

Ключові слова: система підтримки прийняття рішень, метод аналізу ієрархій, вибір комбайнів з урахування планів агрофірми, структурна ієрархічна модель.

ABSTARCT

UDC 004.9 : 631.354.2

Development of an information system to support decisions on planning the fleet of combines of an agricultural company

Shopyak B.I. Department of IT – Dublyany, Lviv NEU, 2024.

Qualification work: 81 p. text, 44 pict., 2 tabl., 10 p. illustrative material, 30 sources.

The object of the research is the automation of the decision-making process on the choice of combines of the agro-industrial complex.

The purpose of the work is to develop an information system to support decisions on the planning of the combine harvester fleet of an agricultural company.

An analysis of the subject area and a classified information system have been carried out, approaches to the construction of such systems have been considered. The methods of solving the problem are described and the method of hierarchy analysis is chosen. On the examples of common combines, the criteria for evaluating alternatives are substantiated. The choice of programming language, development environment and DBMS has been made. A structural diagram of the information system has been developed. A structural hierarchical model for the selection of combines has been created. The implementation of the model by the selected tools was carried out and its work was checked on training data. The organization of data exchange between the software part and the database is described. Individual elements of the graphical user interface are proposed. The issues of labor protection and determination of the economic efficiency of the obtained solutions are considered.

Keywords: decision support system, hierarchy analysis method, selection of combines taking into account the plans of the agricultural company, structural hierarchical model.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- АПК – агропромисловий комплекс;
- ІТ – інформаційні технології;
- ІС – інформаційні системи;
- СППР – системи підтримки прийняття рішень;
- ЗПР – завдання прийняття рішення;
- CODM – Consensus Oriented Decision Making, прийняття рішень на основі консенсусу;
- MAI – метод аналізу ієрархій;
- ОПР – особа, яка приймає рішення;
- СУБД – система управління базами даних;
- ООП – об’єктно-орієнтоване програмування;
- GUI – Graphic User Interface, графічний інтерфейс користувача.

ЗМІСТ

ВСТУП.....	8
1. АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ	9
1.1. Цифровізація сільськогосподарського виробництва	9
1.2. Системи підтримки прийняття рішень.....	14
1.3. Процес прийняття рішень, складові та завдання системи	22
Постановка задачі дослідження.....	25
2. ОБҐРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ	
ВИРІШЕННЯ ЗАДАЧІ	26
2.1. Вибір методу прийняття рішень	26
2.2. Обґрунтування критеріїв оцінки альтернатив	35
2.3. Вибір мови програмування, середовища розробки та СУБД	42
Висновки до розділу	48
3. РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ	49
3.1. Структура інформаційної системи підприємства	49
3.2. Розробка структурної ієрархічної моделі вибору комбайнів	51
3.3. Програмна реалізація моделі	52
3.3.1. Підготовка та формування матриць порівнянь груп та критеріїв	52
3.3.2. Матриці порівнянь альтернатив по окремим критеріям	55
3.3.3. Розрахунки по ієрархічним рівням.....	62
3.4. Організація обміну з базою даних.....	65
3.5. Елементи графічного інтерфейсу користувача.....	67
Висновки до розділу	71
4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	72
5. ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	75
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80

ВСТУП

Управління агропромисловим комплексом (АПК) в сучасних умовах націлене на регулювання агропромислового виробництва та ринків сільгосппродукції, прогнозування його динаміки. Значна роль в управлінських процесах АПК відведена організації повноцінної системи обліку, контролю та аналізу, що підтримується за допомогою інформаційно-аналітичних систем.

Інтенсифікація сільськогосподарського виробництва великою мірою залежить від розвиненості ринку інформаційних ресурсів, які постачають виробників, виробничників, фахівців, керівні органи АПК якісною достовірною інформацією, що сприяє прийняттю раціональних управлінських рішень. У той самий час стримуючим чинником запровадження систем підтримки прийняття рішень з інструментами інтелектуального аналізу є трудомісткість процесу підготовки та уточнення вихідної інформації, яка потрібна на забезпечення функціональності даних систем.

Необхідний розвиток як повноцінної системи обробки даних, що фіксує всі зміни в регіональному АПК або на рівні окремої агрофірми, так і систем, які допомагають керівництву приймати обґрунтовані рішення. Такі системи носять назву систем підтримки прийняття рішень. Поступове впровадження таких систем дозволить уникнути організаційних конфліктів, пов'язаних із відривом реальних практичних завдань сільського господарства від моделей, які пропонуються сучасним програмним інструментарієм інтелектуального аналізу даних.

Таким чином, тема даної дипломної роботи є актуальною та зумовлена факторами, які безпосередньо впливають на ефективність сучасного агропромислового комплексу та рішень, які приймаються.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Цифровізація сільськогосподарського виробництва

Інформатизація та комп'ютеризація засновані на цифровому поданні інформації, яке в масштабах економічного та соціального життя як окремої країни, так і всього світу призводить до підвищення ефективності економіки та покращення якості роботи на підприємстві. В даний час діяльність усіх галузей економіки вже неможливо уявити без інформаційно-комунікаційних технологій (цифрових сервісів, продуктів з використанням big data тощо). Необхідною умовою цифровізації галузей економіки є досягнення високого рівня інформатизації та автоматизації. Перехід до цифрової економіки знаходить прояв у автоматизації бізнес-процесів, запровадження комп'ютерних технологій у виробничу діяльність сільськогосподарських підприємств, організацій сфери послуг, державних органів, фінансових установ та інших. Цифрові рішення дедалі активніше проникають у всі сегменти сільського господарства. [1-5]

Цифровізація сільського господарства – необхідна умова підвищення його конкурентоспроможності. Трансформація сфери АПК передбачає цифровізацію всіх напрямів сільськогосподарського виробництва: рослинництва, тваринництва, рибальства, птахівництва, селекції та генетики, тепличної галузі та інші.

Цифровізація означає розвиток передових бізнес-напрямків, Інтернет-торгівлі професійним комп'ютерним обладнанням та нових робочих спеціальностей, послуг та функцій щодо навчання використання інтелектуальних систем у тваринництві.

Існує багато факторів, які впливають на розвиток та використання цифрових технологій в АПК. Зазначимо основні фактори, що впливають на впровадження цифровізації в агропромисловому комплексі:

- торгова глобалізація;

- зміна клімату;
- зростання урбанізації;
- зростання чисельності населення;
- зміна споживчих переваг;
- розвиток технологій у сільському господарстві.

Цифрові технології сприяють підвищенню продуктивності праці, поліпшенню якості продукції і на який вплив вони на зниження невикористаних витрат. Розглянемо поточні умови застосування цифрових інновацій в агропромисловому комплексі країни та проблеми, з якими стикаються аграрії під час переходу до нових технологій.

У світі для зменшення ризику виникнення глобальних проблем у біологічній та продовольчій галузях людству необхідний удосконалений тип сільського господарства. Безвідходне виробництво та принцип сталого розвитку мають стати головною ланкою у модернізації агропромислового комплексу. Саме перехід до нової економічної моделі та «розумного» сільського господарства зараз є одним із обговорюваних питань, що стоїть перед світовою спільнотою.

Впровадження цифрових технологій у сільськогосподарське виробництво є одним із найважливіших елементів стратегічного розвитку цієї сфери. Біо- та нанотехнології, використання генної інженерії, можливість адаптації виробленої сільськогосподарської продукції до потреб покупців є важливими факторами підвищення конкурентоспроможності галузі. Однак без ефективного використання цифрових інноваційних технологій неможливо у короткий термін перетворити сільськогосподарське виробництво на високотехнологічну галузь.

Етапи цифровізації агропромислового комплексу можна відобразити наступним чином (табл. 1.1).

Сучасна цифрова економіка – це високотехнологічне виробництво з використанням цифрових технологій, тобто економіка, заснована на нових методах обробки, передачі, зберігання, генерування даних, а також цифрових комп'ютерних технологіях. Реалізація засад цифрової економіки дозволяє

суб'єктам господарювання перейти на новий сучасний формат економічних відносин.

Таблиця 1.1 – Етапи цифровізації агропромислового комплексу

Період	Назва	Характеристика
1970-1980	Автоматизація	Створення автоматизованих систем управління та контролю технологічних процесів, що відрізнялися малою потужністю та великою габаритністю.
1980-1990	Електронізація	Впровадження у сільськогосподарське виробництво комп'ютерів та електронних датчиків.
1990-початок 2000х	Інформатизація	Застосування більш сучасних комп'ютерів та використання інформаційного потенціалу Інтернету. Початок використання облікових програм.
Сучасний етап	Цифровізація	Використання в управлінській та виробничій діяльності комп'ютерів, електронних датчиків, смартфонів, за допомогою яких можна оптимізувати облік, контролювати виробничі процеси та автоматизувати поточні технологічні завдання.

Завдяки інноваційним технологіям в аграрному секторі змінюється співвідношення фінансових, матеріальних та трудових ресурсів, що використовуються, структура сільськогосподарського виробництва.

Використання сучасних ІТ-технологій дозволяє покращити умови ведення підприємницької діяльності, підвищити конкурентоспроможність галузі. Для цього необхідне вирішення завдань повної інформатизації галузі АПК та, як наслідок, підвищення якості управління агропромисловим комплексом країни.

В умовах інформатизації розвиток сільського господарства набуває іншого характеру. Аграрне виробництво стає високотехнологічною галуззю економіки, в якій ключове значення надається не кількості ресурсів, а їхньому раціональному використанню та ефективності. Сучасне сільське господарство базується на детальному вимірі всіх процесів, а високі технологи стають стандартом завдяки повсюдному використанню аналітичних та інформаційних систем. Сьогодні сфера сільськогосподарського виробництва може стати

середовищем для впровадження та розвитку нових технологій. Натомість цифровізація сама по собі не є метою. Завдання полягає у вирішенні проблем АПК, підвищенні ефективності сільгоспвиробництва за рахунок цифровізації для створення конкурентної продукції.

На реалізації процесу цифровізації в агробізнесі та державному управлінні агропромисловим комплексом дається взнаки його специфічний характер, який визначає як затребуваність, так і ефективність застосування тут цифрових технологій.

Особливо важливим для розвитку та функціонування сільського господарства є володіння достовірною та різнобічною інформацією, що своєчасно надходить до різних суб'єктів господарювання. Інформатизація у галузі у разі виступає як інструмент управління ризиками, спрощуючи відносини між виробниками, роздрібними продавцями, переробниками продукції і покупцями. Для експортоорієнтованого агропромислового комплексу нашої країни це має важливе значення. При цьому, як показує світовий досвід, державне інформаційне сприяння суб'єктам господарювання відіграє першорядне значення у стимулюванні проникнення та закріплення їх на зовнішніх ринках.

Однак у процесі впровадження нових технологій сільське господарство стикається з низкою проблем:

- не поінформованість споживачів про можливості цифрових технологій;
- мінімальні можливості сільгоспорганізацій для інвестування таких технологій, консалтинг та обробку даних;
- відсутність спеціалізованих державних проєктів, спрямованих на підтримку суб'єктів сільського господарства у галузі придбання інноваційної продукції та послуг;
- підготовка фахівців без тісної прив'язки до виробничих і технологічних потреб економіки, що змінюються;
- розрізненість використовуваних інформаційних систем та ін.

Агропромисловий комплекс має суттєві відмінності з інших сфер економіки, що визначають особливості його інформатизації. Це різноманітність сфер діяльності суб'єктів та споживачів інформації, територіальна віддаленість споживачів інформації від її джерел. Іншими словами, тут має місце різнотипна інформація, яка за своїм змістом та призначенням очікується та використовується різними групами споживачів: наукова, виробнича, комерційна, нормативно-регулююча та інші. Дані види інформації створюються в різних умовах, рухаються до користувачів різними каналами і відповідно вимагають неоднакових підходів до їх обробки та передачі споживачам.

Загалом, можна відзначити, що у сільськогосподарському виробництві є чотири ключові галузі, в яких ефективність від впровадження цифрових технологій проявляється найбільш яскраво:

- точне землеробство (управління продуктивністю посівів з урахуванням стану та зміни факторів довкілля);
- розробка та генна модифікація насінневого фонду (використання сучасних засобів та способів вимірювання, відображення та передачі інформації з метою виявлення та доступу до геному рослини);
- моніторинг якості сільськогосподарської продукції (використання датчиків та аналітики для попередження та запобігання хвороб та псування продукції харчового походження);
- управління виробничими ланцюжками та логістикою.

Елементи цифрової економіки лише починають проникати у сільське господарство. Для цього нині є всі передумови. Для комп'ютеризованих машин потрібно багато даних.

Автоматизацію отримання даних забезпечують сучасні системи: дрони, датчики, які встановлені на техніці, автоматичні лабораторії, погодні станції, супутники. Тому вручну вводиться невелика частина інформації, а автоматизація дозволяє отримувати дані болем швидко та точно. Це дозволяє безпомилково у найкоротші терміни провести аналіз ситуації для оперативного управління сільськогосподарським виробництвом. Отримана інформація

дозволяє оптимізувати сільськогосподарські процеси, прогнозувати врожайність, налаштовувати параметри технологій з урахуванням особливостей стану рослин кожної конкретної ділянки поля.

Ще більшого ефекту можна досягти, якщо застосовувати цифрову економіку у завданнях стратегічного планування агропромислового виробництва. Це оптимізація сівозміни, транспортно-елеваторного блоку, ремонту техніки, закупівель та розміщення запчастин.

Використання сучасних ІТ-технологій та створення сприятливих умов для розвитку цифрової економіки дозволять покращити та створити серйозні конкурентні переваги, як у сільському господарстві, так і в країні в цілому.

Важливість впровадження технологій цифровізації у сфері АПК полягає в тому, що вона дозволить знизити ризики, адаптуватися до зміни клімату, підвищити врожайність сільськогосподарських культур, забезпечити зниження витрат на виробництво продукції, підвищення її якості та конкурентоспроможності.

Для досягнення зростання прибутковості агробізнесу дуже важливо максимально використовувати інноваційні технології. Саме тому з метою скорочення витрат, збільшення продуктивності та оптимізації робочого процесу одним із ключових факторів розвитку сільського господарства є впровадження цифровізації.

1.2. Системи підтримки прийняття рішень

Системи підтримки прийняття рішень (СППР) - це клас інформаційних систем, які забезпечують керівників різних рівнів знаннями та інформацією, що дозволяють їм приймати більш обґрунтовані і правильні управлінські рішення в різних сферах діяльності. [6-9]

СППР призначені в першу чергу для підтримки середнього та вищого керівництва та планування та підвищення ймовірності прийняття

обґрунтованих рішень, навіть в умовах невизначеності та середовища, що швидко змінюється.

В даний час СППР розглядається не тільки як інструмент прийняття рішень, але і, в першу чергу, як засіб підвищення ефективності управління бізнес-процесами в цілому.

У цьому контексті СППР можна розглядати як інтерактивну програмну систему, розроблену для того, щоб допомогти особам, які приймають рішення, збирати корисну інформацію з комбінації необроблених даних, документів, особистих знань і бізнес-моделей для виявлення проблем і прийняття рішень.

Розрізняють такі класи СППР:

- пасивні – підтримують процес прийняття рішень, але самі не формулюють і не пропонують ці рішення, а лише надають інформацію та звіти: особа, яка приймає рішення, приймає рішення сама;

- активні – не тільки надають інформацію, а й можуть запропонувати рішення та альтернативи;

- комбіновані: забезпечують ітеративний процес взаємодії між особою, яка приймає рішення, та СППР для досягнення консолідованого рішення: особа, яка приймає рішення (або її консультант) може змінювати, доповнювати або допрацьовувати пропозиції СППР та надсилати їх назад до системи для розгляду. Цей процес триває до тих пір, поки не буде отримано оптимальне рішення.

За способом функціонування СППР розрізняють:

- комунікативні – забезпечують спільну діяльність кількох фахівців, що працюють над одним завданням;

- орієнтовані на дані – робота таких систем заснована на обробці даних компанії (СУБД, сховищ даних і вітрин даних), особливо часових рядів;

- документно-орієнтовані системи в основному працюють зі структурованими та неструктурованими даними, що містяться в документах різних типів і форматів.

- орієнтовані на знання – у процесі синтезу рішень використовують знання, які накопичують досвід вирішення аналогічних завдань, представлених у вигляді правил, залежностей і шаблонів;

- модельно-орієнтованість — передбачає розробку різних типів моделей бізнес-процесів (фінансових, статистичних, аналітичних, імітаційних тощо) та їх використання для генерації рішень.

Конструктивно СППР складається з трьох основних елементів:

- база даних (або знання);
- концептуальна модель предметної області;
- інтерфейс користувача.

Процес створення DSS-рішення можна розбити на чотири етапи:

- інтелектуальний – визначення умов, в яких потрібне рішення;
- проектний – розробка та аналіз рішення, а також можливих альтернатив;
- селективний — вибір рішення з безлічі альтернатив;
- реалізаційний — адаптація параметрів обраного рішення до конкретних умов.

Вхідні дані СППР – це атрибути, фактори та атрибути, які описують стан бізнес-об'єктів та бізнес-процесів, що вимагають прийняття рішень, а також знань співробітників компанії та профільних експертів.

Виходи СППР - це результати аналізу даних, на основі яких формуються рішення, а також самі рішення.

СППР, що реалізують когнітивні функції і засновані на технологіях штучного інтелекту і використанні інтелектуальних агентів, називаються інтелектуальними СППР.

Концепція СППР виникла в основному в результаті теоретичних досліджень проблем прийняття організаційних рішень, проведених в Технологічному інституті Карнегі в кінці 1950-х - початку 1960-х рр. У середині 1970-х років СППР стала окремим напрямком досліджень.

СППР орієнтована не тільки на менеджерів і менеджерів середньої ланки, а й на зміни, гнучкість і швидку реакцію. Акцент робиться на моделях,

припущеннях і відображенні графіки. В основі лежать професійні методи аналізу та проектування. Ці системи є свого роду ітеративними, а не жорсткими і ніколи не завершеними. Цього вимагає суть неструктурованих задач, які є оригінальними і незвичайними, для яких не існує алгоритмів, які потрібно вирішити, і кожна має свою відповідь.

З цієї причини DSS призначена для підтримки напівструктурованого та неструктурованого аналізу додатків, щоб допомогти розробити, оцінити альтернативи та контролювати їх впровадження. Найбільш поширеним видом СППР є генератор фінансових звітів.

Основою успішного функціонування виробничого середовища є прийняття рішень, адекватних умовам, в яких функціонують об'єкти. Системи підтримки прийняття рішень, що поєднують в собі потужні методи математичного моделювання, наук про управління та інформатики, є інструментом, покликаним допомогти менеджерам в їх діяльності в все більш складному динамічному світі.

Перевагою комп'ютера є його величезна швидкість і пам'ять, що робить його необхідним практично у всіх сферах людської діяльності.

У прийнятті рішень найважливішими напрямками, в яких комп'ютер стає найближчим помічником людини, є:

- швидкий доступ до інформації, накопиченої в комп'ютері особи, яка приймає рішення, або в комп'ютерній мережі;
- оптимізація або інтерактивне моделювання на основі математичних або евристичних моделей
- знаходження в базах даних раніше прийнятих рішень в ситуаціях, аналогічних досліджуванім, для використання особами, які приймають рішення у відповідний момент;
- використання знань кращих фахівців своєї справи, що входять до баз знань експертних систем;
- представлення результатів у найбільш прийнятній формі для осіб, які приймають рішення.

Але традиційне використання комп'ютерів не є найефективнішим. Крім інформації з бази даних, крім деяких економічних або технологічних розрахунків, керівник у своїй діяльності стикається з великою кількістю завдань з управління системою, які не вирішуються в рамках традиційних інформаційних технологій.

СППР - це системи обробки інформації з метою інтерактивної підтримки діяльності керівника в процесі прийняття рішень.

Можна виділити два основних напрямки такої підтримки:

- полегшення взаємодії між даними, процедурами аналізу та обробки даних і моделями прийняття рішень, з одного боку, та особами, які приймають рішення, як користувачами цих систем, з іншого;

- надавати допоміжну інформацію, особливо для неструктурованих або напівструктурованих завдань, для яких важко заздалегідь визначити дані та процедури для відповідних рішень.

Іншими словами, СППР - це комп'ютеризовані помічники, які допомагають керівнику в перетворенні інформації в дії, ефективні для керованої системи. Ці системи повинні володіти якостями, які роблять їх не тільки корисними, але і незамінними для осіб, які приймають рішення. Як і будь-яка інформаційна система, вони повинні відповідати конкретним інформаційним потребам процесу прийняття рішень. Крім того, і це здається головним, СППР повинен підлаштовуватися під свій стиль роботи, відображати його стиль мислення, допомагати всім (в ідеалі) або більшості важливих аспектів діяльності особи, яка приймає рішення. СППР повинна вміти адаптуватися до мінливих обчислювальних моделей, спілкуватися з користувачем на предметно-специфічній мові (в ідеалі природній) і представляти результати таким чином, щоб сприяти більш глибокому розумінню результатів.

При цьому, звичайно, роль СППР полягає не в тому, щоб замінити керівника, а в тому, щоб підвищити його ефективність. Метою СППР є не автоматизація процесу прийняття рішень, а реалізація кооперації, взаємодії системи та людини в процесі прийняття рішень. СППР повинен підтримувати

інтуїцію, вміти розпізнавати неясності і неповноту інформації, володіти засобами для їх подолання. Вони повинні бути доброзичливими до осіб, які приймають рішення, допомагати їм концептуально визначати завдання, пропонуючи знайомі уявлення про результати.

Кожен керівник має свої знання, талант, досвід і стиль роботи. Одна з цілей СППР - допомогти людині поліпшити ці якості. Крім загальновідомих вимог, що пред'являються до інформаційних систем (потужна СУБД, що забезпечує ефективний доступ до даних, їх цілісність і захист; передові аналітичні та обчислювальні процедури, що забезпечують обробку і аналіз даних; транспортабельність, надійність, гнучкість, можливість включення нових технологічних процедур) СППР повинна володіти специфічними особливостями:

- здатність розробляти рішення в особливих ситуаціях, які є несподіваними для осіб, які приймають рішення;

- здатність моделей, що використовуються в системах, адаптуватися до конкретної, конкретної реальності в результаті діалогу з користувачем;

- можливість інтерактивної системи генерації моделей.

У зв'язку з тим, що особа, яка приймає рішення, не завжди має чітко визначену мету в кожній ситуації, рішення є процесом дослідження, а СППР - засобом більш глибокого пізнання системи і вдосконалення стилю роботи менеджера. Як правило, СППР мають модульну структуру, що дозволяє включати нові процедури і модернізувати вже включені в систему відповідно до нових вимог.

Прийняття рішень передбачає послідовне виконання наступних етапів: осмислення проблеми, діагностика, концептуальне або математичне моделювання, розробка альтернатив і вибір тих, які максимально відповідають поставленим цілям, а також контроль за виконанням прийнятого рішення.

СППР покликані допомагати особам, які приймають рішення, на кожному з цих етапів, і, отже, прогрес у розвитку та розширенні сфери їх застосування

залежить як від концепції їх побудови, так і від досконалості відображення кожної з функцій, які вони підтримують.

Прогрес останніх років знайшов своє відображення в інтеграції систем, заснованих на знаннях, в СППР, яка надає поради і пояснення запропонованого рішення.

Еволюція СППР також характеризується рівнем допомоги, що надається особам, які приймають рішення, від пасивної підтримки до посиленої, активної підтримки. Пасивна підтримка забезпечує зручний інструмент, не претендуючи на зміну існуючого способу дій осіб, які приймають рішення. Якість цих СППР залежить від зручності і доступності програмного продукту, а точніше, від його інтерфейсу. По суті, це інтерактивні інформаційні системи, які надають керівнику тільки ті послуги, які він вимагає, і тільки у відповідь на його вимогу. Пасивний підхід включає в себе традиційні СППР, які відповідають на питання «а що, якщо?». (Що буде, якщо..?). Особа, яка приймає рішення, вибирає альтернативи і оцінює їх, вмюючи аналізувати прості альтернативи, узагальнюючи, підвищує ефективність процесу прийняття рішень.

Наразі назріли передумови для переходу до посиленої підтримки прийняття рішень з використанням нових, нетрадиційних напрямків, аналітичних методів і, зокрема, багатокритеріального аналізу. Такий підхід більш широко використовує нормативний аспект отримання ефективного рішення, ніж звичайні СППР. При цьому існують процедури аналізу та пояснення отриманого рішення та оцінки як вигод, так і можливих збитків.

Таким чином, особа, яка приймає рішення, може оцінити варіант, запропонований СППР, і прийняти рішення з більш широким поглядом як на саме рішення, так і на його наслідки, завдяки консультаціям, наданим системою.

Як правило, СППР використовує інформацію з баз даних і знань та/або надану особами, які приймають рішення. Відомо, що менеджери також використовують інформацію з текстових документів, звітів, спеціальних оглядів, статей тощо.

В даний час існує три класи СППР в залежності від складності розв'язуваних завдань і областей застосування.

Перший клас СППР, що володіє найбільшою функціональністю, призначений для використання в державних органах високого рівня (наприклад, міністерствах) і органах управління великих компаній при плануванні великих, інтегрованих цільових програм з метою обґрунтування рішень щодо включення в програму різних політичних, соціальних або економічних заходів і розподілу ресурсів між ними, виходячи з оцінки їх впливу на досягнення основної мети програми. СППР цього класу являють собою системи колективного користування, бази знань яких формуються багатьма фахівцями - фахівцями в різних областях знань.

СППР другого класу - це системи індивідуального використання, бази знань яких формуються самим користувачем. Вони призначені для використання державними службовцями середньої ланки, а також керівниками малих і середніх фірм для вирішення завдань оперативного управління.

СППР класу 3 – це системи для індивідуального використання, які адаптуються до досвіду користувача. Вони призначені для вирішення часто зустрічаються прикладних завдань системного аналізу і управління (наприклад, вибір суб'єкта кредитування, підбір виконавця робіт, призначення на посаду і т.д.). Такі системи забезпечують вирішення поточної проблеми на основі інформації про результати практичного використання рішень тієї ж проблеми, прийнятої в минулому. Конкурентоспроможне виробництво має ґрунтуватися на новітніх досягненнях і, в зв'язку з цим, досить легко переорієнтуватися на більш досконалі технології. Тому керівникам будь-якого рангу повинна бути надана необхідна допомога в розробці і обґрунтуванні рішень, адекватних мінливим умовам, в яких функціонує керована система, і впливам навколишнього середовища.

1.3. Процес прийняття рішень, складові та завдання системи

Загальна схема процесу прийняття рішень включає в себе наступні основні етапи. [10-15]

Стадія 1. Попередній аналіз проблеми. На цьому етапі визначаються:

- основні цілі;
- рівні розгляду, елементи і структура системи (процесу), види зв'язків;
- підсистеми, основні використовувані ними ресурси та критерії якості функціонування підсистеми;
- основні протиріччя, вузькі місця та обмеження.

Стадія 2. Постановка задачі. Формулювання конкретного завдання прийняття рішення включає:

- формулювання завдання;
- визначення типу завдання
- визначення безлічі альтернатив і основних критеріїв вибору кращої з них;
- вибір методу розв'язання задачі розробки.

Стадія 3. Отримання вихідних даних. На цьому етапі встановлюються способи вимірювання альтернатив. Це або збір кількісних (статистичних) даних, або методи математичного або імітаційного моделювання, або методи експертної оцінки. В останньому випадку необхідно вирішити завдання формування групи експертів, проведення експертних опитувань, попереднього аналізу експертних оцінок.

Стадія 4. Розв'язання задачі із залученням математичних методів та комп'ютерних технологій, експертів та особи, яка приймає рішення. На цьому етапі вихідна інформація математично обробляється, уточнюється і при необхідності модифікується. Обробка інформації може бути досить трудомісткою, і може знадобитися зробити кілька ітерацій, і ви можете застосувати різні методи для вирішення проблеми. Тому саме на даному етапі виникає потреба в комп'ютерній підтримці процесу прийняття рішень, яка здійснюється за допомогою автоматизованих систем прийняття рішень.

Стадія 5. Аналіз та інтерпретація результатів. Отримані результати можуть бути незадовільними і вимагати внесення змін до формулювання ЗПР. У цьому випадку вам потрібно буде повернутися до Етапу 2 або Етапу 1 і пройти весь шлях заново. Рішення ЗПР може зайняти досить тривалий проміжок часу, протягом якого середовище завдання може змінюватися і вимагати коригувань у формулюванні задачі, а також у вихідних даних (наприклад, можуть з'являтися нові альтернативи, що вимагають введення нових критеріїв). Завдання прийняття рішень можна розділити на статичні і динамічні. До перших відносяться завдання, які не вимагають повторних рішень через невеликі проміжки часу. До динамічних проблем можна віднести ті, які виникають досить часто. Отже, ітеративний характер процесу прийняття рішень можна вважати закономірним, що підтверджує необхідність створення і використання ефективних систем комп'ютерної підтримки. Однотактні ЛВП є скоріше винятком, ніж правилом.

Система підтримки прийняття рішень вимагає трьох основних компонентів: модулів управління, управління даними для збору даних і ручної обробки, а також управління діалогами для полегшення доступу користувачів до СППР. Користувач взаємодіє з СППР через інтерфейс користувача, вибираючи приватну модель і набір даних для використання, а потім СППР представляє результати користувачеві через той самий інтерфейс користувача. Моделі управління та управління даними функціонують в основному незалежно, починаючи від відносно простої загальної моделі в електронній таблиці і закінчуючи складною, складною моделлю планування, заснованою на математичному програмуванні.

Використовуючи електронну таблицю, таку як Lotus 1-2-3 або Microsoft Excel, ви створюєте моделі для прогнозування різних елементів вашої організації або фінансового стану. В якості даних використовується попередня фінансова звітність організації. Початкова модель включає в себе різні пропозиції щодо майбутніх тенденцій в категоріях витрат і доходів. Після ознайомлення з результатами базової моделі менеджер проводить серію

досліджень «що-якщо», змінюючи одне або кілька припущень для визначення їх впливу на початковий стан. Наприклад, менеджер може дослідити вплив на прибутковість, якщо продажі нового продукту зросли на 10% щорічно. Або менеджер може дослідити вплив більшого, ніж очікувалося, зростання цін на сировину, наприклад, 7% замість 4% на рік. Цей тип генератора фінансових звітів є простим, але потужним DSS для керівництва прийняттям рішень, включаючи фінансові рішення.

Генератор систем підтримки прийняття рішень - це система, яка надає набір можливостей для швидкого і простого створення конкретних СППР. Генератор СППР - це програмний пакет, призначений для вирішення слабо структурованих або неформальних завдань лише частково за допомогою комп'ютера.

Системи підтримки прийняття рішень допомагають знайти відповіді не тільки на пряме запитання «а що, якщо?», а й на подібні. Розглянемо задачі загальної системи підтримки прийняття рішень.

Аналіз кейсів - це оцінка вихідних значень для заданого набору вихідних змінних.

Параметричний (case analysis) аналіз - це оцінка поведінки вихідних значень при зміні значень вихідних змінних.

Аналіз чутливості - це вивчення поведінки результуючих змінних як функції зміни значення однієї або декількох вхідних змінних.

Аналіз можливостей – знаходження значень вхідної змінної, які забезпечують бажаний кінцевий результат (також відомий як «пошук об'єктивних рішень», «аналіз значень цілей», «управління цілями»).

Імпакт-аналіз - це ідентифікація всіх вхідних змінних, які впливають на її значення для обраної результуючої змінної, і оцінка величини зміни результуючої змінної при заданій зміні вхідної змінної, скажімо, на 1%.

Аналіз даних – безпосередньо введення ранніх відомих даних в модель і маніпулювання ними під час прогнозування.

Порівняння та агрегація – порівняння результатів двох або більше прогнозів, зроблених з різними вхідними припущеннями, або порівняння прогнозованих результатів з фактичними результатами, або поєднання отриманих результатів з різними прогнозами або для різних моделей.

Послідовності команд - це можливість використовувати, зберігати для подальшого використання ряд команд і повідомлень, які регулярно виконуються.

Аналіз ризику - це оцінка виконання вихідних змінних при випадкових змінах вхідних змінних.

Оптимізація: знаходження значень керованих вхідних змінних, які забезпечують найкраще значення для однієї або кількох результуючих змінних.

Постановка задачі дослідження

Підсумовуючи описане вище, можна класифікувати інформаційну систему, яка буде розроблятися в дипломній роботі, як систему підтримки прийняття рішень для формування парку комбайнів з урахуванням планів агрофірми. При цьому дані будуть зберігатись в базі даних, а щоб мати до неї доступ, треба буде використовувати клієнт-серверні технології.

Крім того, треба обрати метод, яким буде вирішуватись задача, а також відповідний інструментарій.

На етапі реалізації треба розробити модель вибору комбайнів та її програмну реалізацію, а також описати взаємодію з базою даних і запропонувати елементи графічного інтерфейсу користувача.

РОЗДІЛ 2

ОБҐРУНТУВАННЯ, ВИБІР ТА РЕАЛІЗАЦІЯ ІНСТРУМЕНТАРІЮ ВИРІШЕННЯ ЗАДАЧІ

2.1. Вибір методу прийняття рішень

Найважливішою функцією управління в кожній організації є прийняття рішень, які впливають на всіх, від низу до верху. [16-20] Керівники, спеціалісти, експерти, консультанти, аналітики та системні інженери беруть участь у прийнятті рішень, які впливають на ефективність роботи.

Індивідуальні рішення приймаються однією людиною. Здебільшого це повсякденні рішення. З іншого боку, групові рішення - це рішення, що приймаються групою індивідів, зібраних з цією метою.

Групові рішення, в порівнянні з індивідуальними, мають далекосяжні наслідки і впливають на ряд людей і відділів. Вони вимагають серйозного обговорення, обережності та дебатів. Нижче наведені переваги і недоліки групового прийняття управлінських рішень.

Переваги групових рішень:

- група має більше інформації, ніж одна особа; учасники, залучені з різних галузей, можуть надати більше інформації та знань про проблему;

- група може генерувати більшу кількість альтернатив; це може призвести до ширшого досвіду, більшого розмаїття думок і ретельнішого вивчення фактів, ніж окрема людина;

- участь у колективних рішеннях підвищує визнання та прихильність з боку людей, які тепер вважають це рішення своїм і зацікавлені в його успіху;

- учасники краще розуміють рішення, адже брали участь у його виникненні та розвитку, а отже, докладатимуть зусиль для безперешкодної реалізації рішення;

- взаємодія між людьми з різними поглядами веде до більшої креативності.

Недоліки групових рішень:

- в групах втрачається багато часу для формування управлінського рішення;

- групи створюють тиск, щоб дійти згоди; колективне мислення змушує своїх членів йти на компроміси, де хороші пропозиції втрачають свій блиск, тому що їх доводиться підкоряти спільним інтересам і судженням;

- присутність деяких впливових членів групи може лякати і заважати іншим членам вільно генерувати ідеї; домінування є контрпродуктивним і перешкоджає прийняттю кращих управлінських рішень у групах;

- зібрати потрібних членів команди буває дорого;

- колектив складається з розділених людей, тому легко переломити ситуацію і уникнути відповідальності.

Тобто, згідно з цим описом рішення по формуванню парку комбайнів в агрофірмі може бути як колегіальним (груповим), так і одноосібним (індивідуальним). Це в основному залежить від масштабу компанії та прийнятій структурі управління.

Крім того, рішення можуть бути запрограмованими і незапрограмованими.

Запрограмоване рішення є рутинним і повторюваним. Умови встановлюються заздалегідь для швидкого вирішення питань. Наприклад, у лікарні налаштована процедура прийому нових пацієнтів, і це допомагає кожному співробітнику швидко та легко вирішувати проблеми, навіть коли до лікарні одночасно приходять багато пацієнтів. Запрограмовані рішення не залишають місця для переосмислення. Їх потрібно певним чином дотримуватися. Такі рішення, як правило, приймаються персоналом нижчої ланки відповідно до встановлених правил і процедур.

Незапрограмовані рішення мають справу з унікальними або незвичайними проблемами. Ті, які виникають несподівано, і немає встановленого порядку або формули боротьби з ними. Рішення про розпуск команди, як реструктурувати організацію для підвищення ефективності та де розмістити новий склад компанії – приклади незапрограмованих рішень. Спільною рисою цих рішень є те, що вони завжди нові і не повторюються, і немає готових дій, до яких можна

вдаватися. Оскільки незапрограмовані рішення часто мають серйозні, довгострокові наслідки для організації, вони приймаються тільки персоналом вищого рівня.

Характеристики запрограмованих і незапрограмованих управлінських рішень викладені нижче.

Запрограмовані рішення:

- пов'язані з рутинними проблемами;
- рішення визначаються відповідно до якоїсь звички, правила або процедури;
- відносно прості і мають незначний вплив;
- інформація, що стосується цього типу проблем, є легкодоступною і може бути оброблена заздалегідь визначеним способом;
- рішення займають дуже мало часу і сил, тому що керуються заздалегідь встановленими правилами і процедурами;
- приймаються керівниками нижчої ланки.

Незапрограмовані рішення:

- пов'язані з унікальними та виникаючими проблемами, де рішення є неструктурованими, неповторюваними та погано визначеними;
- не існує заздалегідь встановлених методик або процедур, на які можна покластися; кожна ситуація індивідуальна і вимагає творчих рішень;
- такі рішення є відносно складними і мають довгостроковий вплив;
- потрібно багато часу, розсудливості і роздумів, щоб прийняти правильне рішення;
- в компетенції вищого керівництва.

Таким чином, можна сказати, що задача формування парку комбайнів відноситься до незапрограмованих рішень.

Розглянемо також особливості стратегічних, адміністративних та рутинних рішень.

Прийняття стратегічних управлінських рішень є обов'язком вищого керівництва. Це ключові, необхідні та найважливіші рішення, які впливають

практично на всю компанію. Вони вимагають значного розподілу ресурсів і орієнтовані на майбутнє з довгостроковими наслідками.

Адміністративні рішення стосуються операційних питань, а саме того, як забезпечити безперебійну реалізацію різних аспектів стратегічних рішень на різних рівнях організації. В основному їх приймають менеджери середньої ланки.

З іншого боку, рутинні рішення повторюються. Вони не вимагають довгих роздумів і зазвичай стосуються короткострокових зобов'язань. Вони, як правило, надають лише незначний вплив на роботу організації. Як правило, менеджери нижчої ланки контролюють такі механічні або операційні рішення.

З цього погляду - рішення про формування парку комбайнів однозначно відносяться до стратегічних рішень компанії.

Розглянемо основні стилі прийняття рішень. Зазвичай відокремлюють чотири таких стилі.

Авторитарний стиль - це коли рішення приймаються без участі інших. Найшвидший спосіб прийняття управлінських рішень, тому що вас не затримують інші люди, які висловлюють свою думку або обговорюють інші рішення. Такі рішення часто є найбільш ризикованими, оскільки альтернативи часто не розглядаються. Більшість надзвичайних ситуацій виправдовують авторитарний підхід, але багато управлінських рішень вимагають залучення додаткових людей.

Консультативний стиль - це коли лідер пропонує іншим людям вислухати їхню думку, але в кінцевому підсумку рішення приймає він один. Цей варіант займає більше часу, ніж авторитарний, оскільки розглядаються інші думки та можуть бути запропоновані альтернативи, що робить його менш ризикованим. Це найбільш пасивний спосіб залучення інших, який може бути використаний для того, щоб люди відчували, що вони брали участь у розробці рішення (навіть якщо вони в кінцевому підсумку не мають права голосу при остаточному висновку).

Стиль голосування - це коли варіанти обговорюються в групі, а потім призначається голосування, при якому вибирається найбільш сприятливий для більшості людей варіант. Цей метод ще називають демократичним, тому що думка кожної людини включається в остаточний висновок. Кожен, хто бере участь у голосуванні, однаково бере на себе відповідальність за прийняття управлінського рішення, що ще більше знижує ризик провалу. Голосування є відмінним варіантом дій, які повинні бути підтримані і виконані групою, саме тому цей метод найчастіше використовується в раді директорів або вищому керівництві.

Стиль або метод консенсусу полягає в тому, що група обговорює варіанти та рекомендації, поки всі не погодяться щодо однакового курсу дій. Це найскладніший і трудомісткий метод, тому що він вимагає згоди різних людей з різною мотивацією на один варіант. Цей метод слід використовувати помірно, виключно виходячи з часу, витраченого на досягнення згоди в групі. Також важливо, щоб меседж про те, що рішення прийнято і узгоджено, був очевидним для причетних.

Звісно, те, який метод буде прийнятним в компанії, залежить в основному від структури та прийнятих протоколів взаємодії. Але навіть в консультативному методі важливо, щоб експерти-консультанти змогли донести своє бачення до керівництва.

Також розглянемо методи ефективного групового прийняття рішень.

Модель прийняття рішень Хой-Тартера. У цій моделі основна мета полягає в тому, щоб визначити, хто саме повинен бути включений в процес прийняття управлінських рішень. Різні рішення вимагають різних внесків від різних людей, тому визначення того, хто повинен бути включений в процес прийняття рішень, а кого не слід враховувати, є ключовим кроком, який не можна ігнорувати. Включення неправильних людей або відсутність потрібних людей є помилкою, яка може мати серйозні наслідки. Ця модель вимагає створення матриці, яка буде заповнена оцінками досвіду і того, чи буде людина особисто

зацікавлена в рішенні. Може знадобитися деякий час, щоб зрозуміти, як саме працює ця модель, але це варте зусиль через її ефективність.

Метод множинного голосування. Якщо час від часу використовується голосування для прийняття важливих управлінських рішень, можна застосувати популярний метод прийняття рішень кількома голосами. За допомогою цього методу можна вибрати зі списку найпопулярніші варіанти, щоб отримати уявлення про консенсус групи. Багаторазове голосування не завжди є правильним висновком, але воно може бути ідеальним за певних обставин.

Спочатку складається список ідей, які будуть проголосовані. Також перед тим, як взяти участь у голосуванні, потрібно визначитися, скільки саме голосів може віддати кожен учасник. Зазвичай кожній людині дозволяється проголосувати приблизно за 1/3 ідей зі списку. Таким чином, якщо буде список з 15 пунктів, то кожній людині буде дозволено віддати по п'ять голосів. Після того, як всі голоси віддані і зібрані, залишається підрахувати результати і визначити ідеї-переможці. Потім ви можете звузити список ідей-претендентів і провести друге голосування, додатково сконцентрувавши список лише на кількох найсильніших варіантах. Множинне голосування є ідеальним способом оцінити думку великої групи, коли є кілька ідей.

Модифікований метод Борда є розширенням багаторазового голосування. Це працює аналогічно, тільки голоси зважуються для потенційного досягнення більш значущого результату. Замість того, щоб голосувати лише за один пункт або за кілька, ви повинні попросити кожного виборця присвоїти бали ідеям, які він обрав. Для вибору найкращого управлінського рішення бали для голосування просто підраховуються.

Множинне голосування та його розширення відносяться до методів прийняття командних рішень. Розглянемо ще деякі приклади таких методів.

Модель CODM Хартнетта. Аббревіатура CODM розшифровується як прийняття рішень на основі консенсусу (Consensus Oriented Decision Making). Ідея полягає в тому, щоб привести групу до консенсусу. Модель CODM вимагає виконання наступних семи кроків:

- виявлення проблеми;
- відкрита дискусія;
- виявлення основних проблем;
- розробка пропозицій;
- вибір напрямку;
- розробка кращого рішення;
- висновок.

Звичайно, може знадобитися трохи змінити модель, щоб відповідати потребам вашої організації.

Крім того, існують ще методи експертних оцінок, з яких розглянемо метод Дельфі та метод аналізу ієрархій.

Метод Дельфі. Основна ідея методу Дельфі полягає в тому, щоб надати кожному учаснику платформу, де вони можуть ділитися своїми чесними думками, не боячись наслідків. Для досягнення цієї мети метод Дельфі заснований на системі анонімності.

Протягом усього процесу кожен учасник може анонімно висловлювати свої думки та ідеї, щоб група відчувала себе комфортно, висловлюючи свої чесні погляди на тему. Навіть після того, як рішення буде прийнято і процес буде завершено, структура анонімності буде збережена.

Використання методу Дельфі для збору ідей з метою вирішення проблеми має ряд переваг. Можна легко уникнути двох поширених проблем при груповому прийнятті рішень – апеляції до влади та групового мислення. Ніхто не знатиме, звідки беруться ідеї, тому не буде спокуси апелювати до авторитету, який за ними стоїть. Крім того, оскільки на цьому етапі процесу немає особистих зустрічей, групове мислення також не буде проблемою.

Метод аналізу ієрархій (МАІ) - це певна сукупність прийомів і методів прийняття управлінських рішень в рамках системного підходу, обраного по відношенню до складних проблемних ситуацій. [21-23]

Успіх тієї чи іншої економічної діяльності багато в чому визначається рішеннями, прийнятими економічними агентами (як правило, власниками і

керівниками фірм). Процес прийняття управлінських рішень може бути реалізований в результаті застосування спеціальних методів. Одним з таких методів є метод аналізу ієрархії.

Даний метод є математичним інструментом, в результаті практичного використання якого особа, яка приймає рішення (ОПР), в інтерактивному режимі знаходить варіант (альтернативу) рішення, який найбільше відповідає розумінню цієї людиною суті розв'язуваної задачі та її вимог до її розв'язання.

Розробником методу аналізу ієрархій прийнято вважати американського математика Томаса Л. Сааті (1926 – 2017). Сьогодні цей метод та його розширення є невід'ємною частиною практичної управлінської діяльності в багатьох сферах.

МАІ ґрунтується як на математичному, так і на психологічному аспектах. Вони враховуються спеціальними програмними продуктами, призначеними для використання на комп'ютерах.

Сфера застосування МАІ широка і різноманітна. Вона сприяє вирішенню проблем та прийняттю рішень у таких сферах, як міжнародне співробітництво, державне управління, бізнес, промисловість, освіта, охорона здоров'я тощо.

У порівнянні з іншими методами прийняття управлінських рішень, перевагою методу аналізу ієрархій є:

- по-перше, в реалізації чіткої і раціональної структуризації складної проблеми у вигляді ієрархії;
- по-друге, кількісно оцінити альтернативні рішення;
- по-третє, є можливість порівняти оцінені варіанти.

МАІ передбачає, що аналіз проблеми повинен починатися з побудови ієрархічної структури. Обов'язковими елементами цієї структури є мета, критерії, альтернативи та інші фактори, що впливають на остаточний вибір. Побудована структура є своєрідним відображенням того, як особа, яка приймає рішення, розуміє проблему.

Ієрархія - це сукупність елементів, кожен з яких присвячений окремому аспекту розв'язуваної задачі. При цьому МАІ дає можливість враховувати

об'єктивні дані і суб'єктивні експертні оцінки, вимірювані кількісні параметри і якісні характеристики, матеріальні і нематеріальні фактори. Це дозволяє зробити висновок про те, що вирішення проблемної ситуації в даному випадку аналогічно процедурам і методам аргументації, що застосовуються на інтуїтивному рівні.

На вершині ієрархії знаходиться основна мета, елементи нижнього рівня представлені альтернативними варіантами досягнення цієї мети. На проміжному рівні ієрархії знаходяться елементи, які відповідають факторам або критеріям, що пов'язують первинну мету з альтернативами.

Факторами, що впливають на остаточну форму вибудовуваної ієрархії, є як об'єктивна сторона аналізованої проблеми, так і знання, судження, думки, бажання, системи цінностей учасників процесу аналізу проблеми і вироблення рішення. У більшості випадків ієрархічна структура виглядає як перевернуте дерево, де кожен елемент, крім самого верхнього, залежить від одного або декількох елементів, розташованих над ним.

Після того, як ієрархічна структура створена, особа, яка приймає рішення, повинна вирішити, які елементи є більш бажаними та важливими, а які – менш важливими. Іншими словами, розставити пріоритети, що допоможе провести процедуру попарних порівнянь.

Пріоритетами є числа (від нуля до одиниці), безрозмірні величини, що виражають відносні ваги елементів у кожній групі. За визначенням, сума пріоритетів членів, що належать до одного і того ж вищого члена ієрархії, дорівнює одиниці. Чим вище значення пріоритету, тим більш значущим є відповідний елемент.

Відмінною особливістю методу аналізу ієрархій є можливість (за рахунок безрозмірних пріоритетів) обґрунтовано порівнювати різні фактори.

Використання МАІ завершується виконанням так званої «лінійної згортки» (тобто синтезу, поєднанню) пріоритетів по ієрархії. Слідом за цим розраховуються пріоритети альтернативних рішень щодо основної мети особи,

яка приймає рішення. Найкраще альтернативне рішення – це те, яке в кінцевому підсумку отримує найвищий пріоритет.

Таким чином, використання МАІ дозволяє суб'єктам економічних відносин детально вивчити проблемну ситуацію, що виникла, і, беручи до уваги як об'єктивні кількісні дані, так і суб'єктивні оцінки експертів і вимоги особи, яка приймає рішення, сформулювати остаточний варіант управлінського рішення, спрямованого на приведення поточної ситуації до потрібного вигляду.

2.2. Обґрунтування критеріїв оцінки альтернатив

Для оцінки альтернатив треба визначитись з критеріями такої оцінки. Для цього необхідно провести огляд або аналіз можливих альтернатив та виявити спільні риси/ознаки, які можуть виступити в якості критеріїв порівняння. Крім того, доцільно розглянути можливості угруповання окремих критеріїв по їх відношенню до деяких властивостей альтернатив - наприклад по технічним якостям, економічним показникам тощо. Також треба враховувати доволі поширену послугу - можливість здачі комбайна в оренду дрібним господарствам, яким не вигідно купляти дорогу техніку або брати її в лізинг.

Тому розглянемо декілька альтернатив серед найбільш популярних комбайнів, які пропонуються на вітчизняному ринку. Для цього проаналізуємо джерела в Інтернеті [24-26] та з'ясуємо, що пропонують зараз великі дилерські мережі та роздрібні продавці як на первинному, так і на вторинному ринках.

По-перше, визначимо, яка сільськогосподарська техніка належить до класу комбайнів.

Зернозбиральний комбайн - це одна з основних сільськогосподарських машин, яка використовується для збирання зернових і бобових культур. У завдання цієї техніки входить скошування, обмолот, очищення і укладання зерна в спеціальний бункер. Комбайн також збирає соломку для того, щоб її прибрати. Крім того, ця сільськогосподарська машина використовується для збирання кукурудзи, різних трав та олійних культур.

При виборі комбайна в першу чергу слід відштовхуватися від його продуктивності, на яку впливає тип молотильного механізму. Потужність комбайна залежить від його розмірів, а також від конструктивних особливостей робочих механізмів. Також на потужність цієї машини впливає швидкість молотильного механізму і зроблені настройки. При цьому продуктивність комбайна також залежить від виду культури, яку потрібно зібрати, рельєфу місцевості і так далі.

Найчастіше можна зустріти молотильні машини класичного типу, але набирають популярність і ротаційні агрегати. Зернозбиральні комбайни можна розділити на такі типи:

- роторні;
- барабанні (клавійного типу);
- гібридні.

Ця класифікація заснована на типі молотильно-сепаруючого пристрою. Розглянемо докладніше ці види обладнання.

В процесі скошування барабанний комбайн зрізає колоски за допомогою жатки, потім вони поміщаються в спеціальний молотильний барабан. Тут завдяки швидкому обертанню зерно відокремлюється від соломи, яка надходить до розсіювальних ключів. Тут з качанів видаляються залишки зерна. Далі зерно направляється в бункер комбайна, а солома після подрібнення з нього видаляється. Ця солома перегниває і стає добривом на полі.

Роторний комбайн не має барабана, але є спеціальний фільтр, за допомогою якого зтягується солома. При цьому процеси обмолоту і сепарації здійснюються одночасно. Зерно відокремлюють від качанів протиранням. Особливістю цієї системи є її висока продуктивність, особливо поділ, здійснюваний робочими елементами машини. Ще однією перевагою є висока якість обмолоту. Але з недоліків варто виділити велику витрату палива (на 20-30% більше, ніж у інших моделей).

Гібридні комбайни використовуються в сільському господарстві для збирання зернових і бобових культур. Як зрозуміло з назви, механізм цього агрегату комбінований, тобто включає в себе і барабан, і ротор.

Але на цьому класифікація комбайнів не закінчується. Також вони поділяються на види в залежності від того, яку культуру вони призначені для збирання. Існують не тільки зернозбиральні комбайни, а й ті, які використовуються для збирання капусти, томатів і буряка.

Звісно, комбайни для збирання овочів не підходять для збирання зернових. Ми будемо розглядати зернозбиральні комбайни, як найбільш поширені, та розглянемо їх популярні моделі.

Зернозбиральний комбайн John Deere S690 є одним із найпопулярніших комбайнів в Україні та світі, оскільки призначений для збирання важких та густих культур, таких як ріпак, кукурудза, соняшник з високою врожайністю.

Технічні дані:

- потужність двигуна - 460 кВт;
- кількість циліндрів - 6;
- робочий об'єм двигуна - 13,5 л;
- паливний бак вміщує - 1155 л;
- зерновий бункер - 14,1 куб.м;
- швидкість завантаження зерна - 135 л/с;
- жатка - 10,7 м;

У S690 передні колеса були замінені гусеницями. Для подачі зерна використовуються три потоки. Конструкція ротора сприяє економії палива. Пропускна здатність підвищується завдяки гострому передньому конусу. Завдяки більшому простору навколо валу обмолот відбувається м'якше, покращуючи якість соломи та зберігаючи зерно навіть на чутливих культурах.

Зернозбиральний комбайн John Deere S690 оснащений:

- сепараційний барабан;
- транспортний барабан для соломорізок з 1060 об/хв;
- застосовує технологію очищення Dyna-Flow II;

- монітор контролю живлення Vision Trak і монітор відеоспостереження Info Trak;

- кондиціонер.

Для використання на схилах передбачена система регулювання екрану. Використовує електричний механізм обертання для видалення врожаю та регулювання дефлекторів розкидача соломи. Частота обертання котушки може автоматично регулюватися за допомогою центральної панелі керування, автоматичного жатки HeaderTrak і гідравлічного розвантажувального шнека 6,90 м з обертовим рухом 120 к/с. Можливість спостерігати за випускною трубою та розподільником соломи через камери спостереження.

Окрім роботи на високоврожайних олійних та зернобобових культурах, комбайн також використовується для скошування, сортування та обмолоту зернових у полі. S690 підходить для районів, у яких температура та вологість коливаються. Застосування зернозбиральних комбайнів вигідно при роботі в несприятливих умовах, таких як забруднене повітря, недостатнє освітлення. Країна виробник – США.

Найпопулярніша німецька марка зернозбиральних комбайнів - це Claas Lexion. З моделлю 8900 користувачі можуть вибрати з різних комбайнів і використовувати їх для максимального підвищення продуктивності, відрізняючи зерно від соломи.

Технічні характеристики:

- тип двигуна - D4276;
- номінальна потужність - 581 кВт;
- кількість циліндрів - 6;
- ротор – 445 мм;
- ширина ріжучої площадки – 13,8м;
- зерновий бункер – 15 000 л.

Комбайн оснащений:

- система динамічної потужності, завдяки чому двигун може краще справлятися з поточним навантаженням;

- кондиціонер;
- холодильник;
- автоматика для управління та контролю робочих процесів;
- потужне освітлення для роботи вночі;
- окремий лоток для збору каміння.

Claas Lexion 8900 може обробляти велику площу лише за один день, залежно від типу культури, якій віддає перевагу виробник. Завдяки своїй здатності працювати в суворих умовах, він широко сумісний з великими та середніми фермами. Країна виробник – Німеччина.

Зернозбиральний комбайн New Holland CR1090 є чудовим варіантом для високопродуктивних ферм. Здатний обробляти великі обсяги зерна навіть у складних умовах роботи. Додаткова конструкція призначена для захисту комбайна, і її можна використовувати для збирання на нерівних поверхнях. Гарантує бездоганну висоту зрізу та 100% накопичення зерна. Рівень продуктивності 90 т/год.

Технічні характеристики:

- двигун об'ємом 15,9 л;
- потужність 480 кВт;
- зерновий бункер – 14500 л.
- ємність паливного баку – 1300 л;
- система обмолоту – подвійний ротор.

До складу обладнання входять:

- системи кондиціонування та вентиляції;
- автоматичний контроль температури;
- можливість управління системою автоматичного керування Smart Steer;
- добре спроектована та комфортна кабіна з гарним оглядом;
- соломорізка з лотком для збору каміння;
- система захисту від перекидання;
- розкидач соломи;
- система очищення зерна.

Країна виробник – Італія.

CASE IH AXIAL FLOW 8250, один із найкращих зернозбиральних комбайнів на ринку, дозволяє якнайшвидше заповнювати причепа зерном. Це рішення вирішує проблеми, пов'язані з високоврожайними культурами, забезпечує швидку адаптацію до змін клімату і гарантує високу надійність та ефективність роботи.

Технічні характеристики:

- двигун FPT Cursor 13/Tier II;
- об'єм - 12,9 л;
- максимальна потужність - 420 кВт;
- ємність паливного баку – 1125 л;
- тип паливної системи – Common Rail високого тиску;
- об'єм зернового бункера – 14,4 куб.м;
- довжина вивантажувального шнека – 6,7 м.

До складу оснащення моделі входять:

- піддон для збору каміння;
- кондиціонер;
- розкидач соломи;
- автоматизована технологія збирання врожаю AFS .

Завдяки командам AFS-урожай водієві не потрібно пристосовуватися до мінливих умов під час збирання зернових. Ця технологія дозволяє контролювати всі ключові елементи вашого пристрою та оптимізувати такі налаштування:

- швидкість ротора та вентилятора;
- кут розкриття очисної решітки;
- регулювання положення направляючої корпусу ротора.

Для інтенсивних сільськогосподарських структур CASE IH AXIAL FLOW 8250 є найкращим вибором. Він підходить для ситуацій, коли збирання врожаю є складним. Система допомагає операторам-початківцям керувати

обладнанням, приймаючи самостійні рішення, які призводять до підвищення ефективності роботи. Країна виробник – США.

Підсумовуючи цей невеличкий огляд, можна побачити, що критерії оцінки альтернатив можна розділити на 4 групи за сутністю. Це такі групи критеріїв:

- технічні;
- економічні;
- експлуатаційні;
- ергономічні.

Розглянемо ці групи детально. Насамперед, при прийнятті зваженого рішення слід враховувати наступні технічні характеристики зернозбирального комбайна:

- потужність;
- вантажопідйомність;
- продуктивність;
- обсяг зернового бункеру;
- робоча ширина ріжучої частини.

Зернозбиральний комбайн є дуже складною і дорогою технікою, тому при виборі комбайна необхідно враховувати і економічні показники:

- вартість комбайна;
- витрати на технічне обслуговування;
- витрата пального;
- умови поставки.

Група експлуатаційних параметрів описує реальну можливість ефективного використання комбайна:

- надійність;
- сервісне обслуговування;
- довговічність;
- універсальність застосування.

Вибір наступних ергономічних параметрів має велике значення, оскільки вони впливають на ефективність роботи водія:

- комфортність керування;
- безпека оператора;
- оглядовість з кабіни;
- наявність клімат-контролю;
- рівень шуму в кабіні.

Слід зазначити, що даний перелік не є виключним та остаточним. При потребі користувачі можуть додавати або виключати необхідні критерії, формуючи модель під свої вимоги.

2.3. Вибір мови програмування, середовища розробки та СУБД

Кожен розробник використовує мову програмування для відображення, створення та використання додатків на різних платформах. Деякі мови більше підходять для конкретної мети, ніж інші, з точки зору того, які інструменти та функції вони пропонують користувачам. [27-28]

Мова програмування відноситься до різних інструкцій, команд і синтаксису, необхідних для розробки програмного забезпечення.

Мови, які використовуються розробниками для написання коду, відомі як мови високого рівня. Потім код розробника компілюється в низькорівневу мову, яку можна легко ідентифікувати та обробити апаратним забезпеченням. Високорівневі мови програмування, як правило, простіші для читання та аналізу, що дозволяє розробникам створювати вихідний код, використовуючи часто використовувані символи та слова природним способом. Більшість розробників можуть зрозуміти багато популярних мов програмування високого рівня через їх схожість один з одним.

До мов низького рівня належать машинні мови та мови асемблера. Мова асемблера має список базових інструкцій і, як правило, складніша для читання, ніж будь-яка мова високого рівня. Машинний код має набір двійкових кодів, які процесор може легко розшифрувати. Цей тип коду не вважається придатним для читання людиною.

Різноманіття мов програмування з часом ускладнює правильний вибір. Ідеальний вибір буде залежати від вимог проекту і потреб бізнесу.

Загальні характеристики мов програмування наведені в табл.2.1. Розглянемо мови програмування трохи докладніше.

JavaScript — це мова програмування, яка дозволяє розробникам додавати складні функції на веб-сайти. Це одна з трьох стандартних технологій для веб-розробки, поряд з HTML і CSS.

Мова на стороні клієнта. Для виконання JavaScript-коду викликається центральний процесор, а не WEB-сервер. Це призводить до зниження навантаження на сервер і економії пропускнуої здатності. Легкість в освоєнні. JavaScript відносно простий у вивченні, оскільки його синтаксис дуже схожий на англійську мову. JavaScript — це мова програмування, яка дозволяє розробникам додавати складні функції на веб-сайти.

Таблиця 2.1 - Загальні характеристики мов програмування

Мова	Резюме
Javascript	Інтерпретована структурована мова програмування, скриптова високого рівня зі слабкою динамічною типізацією та поліпарадигмальна.
SQL	Структурована мова запитів, яка дозволяє отримувати доступ до баз даних і керувати ними.
Python	Мова програмування, що дозволяє швидше розробляти та інтегрувати системи більш продуктивно.
Java	Об'єктно-орієнтована та кросплатформна мова програмування.
C#	Розроблена компанією MS, ця мова програмування вважається універсальною.
GO	Статично типізована, скомпільована мова програмування з відкритим кодом.

Мова на стороні клієнта. Для виконання JavaScript-коду викликається центральний процесор, а не WEB-сервер. Це призводить до зниження навантаження на сервер і економії пропускнуої здатності.

Легкість в освоєнні. JavaScript відносно простий у вивченні, оскільки його синтаксис дуже схожий на англійську мову.

Швидка функціональність JavaScript для кінцевих користувачів сприяє швидкому виконанню коду на стороні клієнта, забезпечуючи миттєве виконання завдань і відображення результатів.

Розширений функціонал WEB-сторінок. JavaScript вважається багатофункціональною та простою у використанні мовою для надання додаткової функціональності веб-сайтам.

JavaScript не залежить від платформи і дозволяє розробникам писати один скрипт і використовувати його в різних місцях, коли виникає необхідність.

Підтримується динамічна типізація, що полегшує визначення типів змінних на основі збереженого значення.

JavaScript підтримує дві ключові функції ООП, успадкування та інкапсуляцію, у вигляді шаблонів створення об'єктів та шаблонів повторного використання коду відповідно.

Функціональність. JavaScript забезпечує функціональне використання для розробників, оскільки об'єкти генеруються з функцій-конструкторів, і кожна функція представляє унікальний тип об'єкта.

Наступна мова - SQL або Structured Query Language. Це мова програмування, яка використовується для встановлення з'єднань з базою даних. Вважається стандартною мовою для систем управління реляційними базами даних (СУБД).

Гнучка обробка запитів. SQL дозволяє користувачам швидко витягувати великі обсяги даних. Стандартні операції, такі як введення, маніпуляції та видалення, зазвичай займають менше часу.

Навички програмування не потрібні. Написання великих обсягів коду, як правило, не потрібне для вилучення даних. SQL дозволяє використовувати

ключові слова і має прості правила синтаксису. Зазвичай вважається, що це зручна мова програмування.

Портативна мова. SQL використовується в програмах на всіх пристроях, незалежно від платформи, яку вони використовують.

Оптимальна продуктивність. SQL забезпечує високий рівень продуктивності програмування, коли користувачам доводиться справлятися з великими робочими навантаженнями та інтенсивним використанням.

Висока доступність. SQL забезпечує сумісність з провідними постачальниками баз даних. Ці СУБД пропонують підтримку SQL завдяки тому, що мова є відмінним варіантом для розробки розширень додатків.

SQL забезпечує значну безпеку для програмістів, оскільки вони можуть легко встановлювати дозволи для таблиць, подань і процесів обробки даних.

Перейдемо до третьої мови - Python. Це високорівнева об'єктно-орієнтована та інтерпретована мова з динамічною семантикою. Це чудова мова для швидкої розробки додатків, яка пропонує такі функції, як динамічна прив'язка, динамічна типізація та високорівневі структури даних.

Python пропонує простий у вивченні синтаксис, який сприяє легкості розуміння та зменшує витрати на обслуговування програми. Ця мова пропонує підтримку різних пакетів і модулів, які роблять можливим повторне використання коду.

Код на Python легко читається, а його синтаксис дуже схожий на англійську мову. Як наслідок, користувачам легко розуміти код Python.

Краща продуктивність. Python є продуктивною мовою програмування завдяки своїй простій природі. Розробники можуть зосередитися на своїх основних обов'язках, використовуючи цю мову. Python дозволяє програмістам звільнитися від звичної складності, пов'язаної з вивченням і освоєнням мови.

Інтерпретована мова. Python виконує пряме, порядкове виконання коду, оскільки це інтерпретована мова. Якщо під час виконання коду виникає помилка, виконання коду зупиняється і генерується звіт про помилку. В результаті мова полегшує налагодження коду.

Відкритий код. Доступ до Python можна отримати через офіційний сайт, і будь-хто бажаючий може завантажити його абсолютно безкоштовно.

Python надає всі можливості об'єктно-орієнтованого програмування для розробників, а також підтримку концепцій ООП, таких як об'єкти, класи та інкапсуляція.

Далі розглянемо мову Java - об'єктно-орієнтовану мову, розроблену таким чином, щоб мати мінімальну кількість залежностей. Вона підтримує WORA або write once run anywhere (пиши один раз, запускай будь-де). Це означає, що розробники можуть запускати скомпільований код Java на всіх платформах без необхідності будь-якої перекомпіляції. Ця мова має синтаксис, подібний до C та C++, але також має багато низькорівневих функцій.

Java відома своєю простотою та зручністю у використанні. Вона також проста в написанні і в налагодженні в порівнянні з іншими мовами.

Java дозволяє користувачам створювати стандартні програми та повторно використовувати написаний код, що полегшує розробку.

Код, написаний на мові програмування Java, може легко запускатися на будь-якому комп'ютері без необхідності використання будь-якого додаткового програмного забезпечення. Єдиною вимогою для запуску коду на Java є наявність JVM (віртуальна машина Java).

Java надає можливості розподілених обчислень, що дозволяє кільком комп'ютерам працювати разом в одній мережі.

Java вважається простою в освоєнні мовою програмування. Знання основ ООП ще більше спрощує роботу з мовою.

Java — це безпечна мова програмування з функціями, які дозволяють програмістам створювати надійні та безпечні програми. Вона підтримує методи автентифікації на основі шифрування з відкритим ключем.

Архітектурно-незалежна. Компілятор створює незалежний від архітектури формат об'єктного файлу, гарантуючи, що він правильно виконується на різних процесорах, що працюють під керуванням середовища виконання Java.

Наступна мова - C#. Це мова програмування загального призначення з підтримкою багатьох функцій, таких як сильна типізація, статична типізація, клас, декларативність, універсальність, імперативність, функціональність і компонентний підхід до програмування.

C# за своєю суттю є об'єктно-орієнтованою мовою програмування. Як наслідок, це дозволяє програмістам створювати програми, які складаються з модулів, які прості в обслуговуванні та мають код, який можна використовувати повторно.

Кросплатформна розробка. C# добре підходить для розробки програм на багатьох різних платформах.

Код на мові C# можна написати в будь-якому редакторі коду.

C# має вбудовану просту у використанні систему для автоматичного збору та видалення сміття, коли це необхідно. Мова також досить ефективна для контролю системи та підтримки її чистоти.

Це мова з відкритим кодом, яку можна використовувати на платформах Mac, Windows і Linux. C# є універсальним і може використовуватися для створення додатків для Android та iOS, додатків магазину Windows, бекенд-фреймворків, бібліотек тощо.

Остання мова, яку ми розглянемо - Go. Це статично скомпільована мова з синтаксисом, подібним до C, але з додатковими функціями, такими як збір сміття, безпека пам'яті, структурна типізація та паралелізм.

Go — це мінімалістична мова програмування, яка має дуже коротку специфікацію. Вивчати, читати та писати програми з його допомогою досить легко для більшості програмістів.

Прозорість коду. Go дотримується єдиного стандартного формату коду та розроблений для забезпечення найвищого рівня прозорості коду для користувачів. Все це також забезпечує краще розуміння коду.

Go створює двійкові файли для додатків за допомогою вбудованих залежностей і усуває необхідність для користувачів виконувати інсталяцію під час виконання.

Творці мови Go зробили його дуже легким для розуміння програмістами, включивши в нього певні об'єктно-орієнтовані принципи. Він має обмежені функції, але простий у використанні та дуже простий у вивченні.

Go пропонує надійну стандартну бібліотеку, що надається в пакетах. У ньому є всі інструменти та компоненти, необхідні розробникам.

Таким чином, можна зробити висновок, що для розробки додатка краще за все підходить мова Python. В якості мови зв'язку з базою даних обрано стандартну мову SQL.

Для Python існує багато середовищ розробки, які мають свої особливості. Автором обрано популярне середовище Google Colaboratory [29], більш відома як "Colab", є безкоштовною платформою для ноутбуків Jupyter. На додаток до середовища виконання ноутбука, Python і R Colab дозволяють ділитися безкоштовним доступом до обмеженої кількості графічних процесорів і TPU.

Висновки до розділу

В результаті виконання даного розділу вирішені такі питання:

- розглянуті методи вирішення задачі та обрано метод аналізу ієрархій;
- на прикладах розповсюджених комбайнів обґрунтовані критерії оцінки альтернатив;
- зроблено вибір мови програмування (Python), середовища розробки (Google Colab) та СУБД (MySQL) [30].

РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ

3.1. Структура інформаційної системи підприємства

Якщо розглянути типову структуру інформаційної системи підприємства (рис.3.1), то можна побачити її основні компоненти.

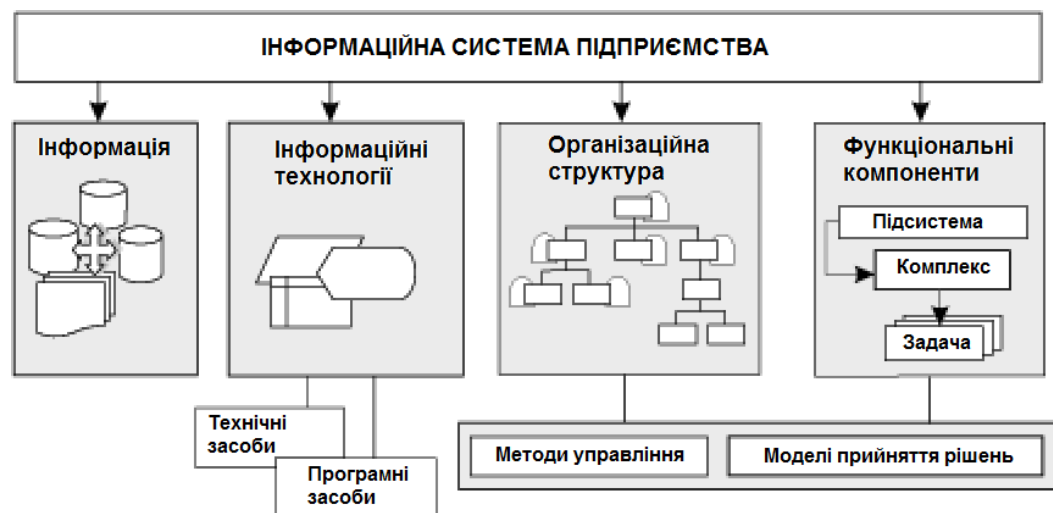


Рисунок 3.1 - Типова структура ІС підприємства

Це, в першу чергу, безпосередньо інформація або дані, які накопичуються, обробляються, зберігаються, передаються та виводяться в інформаційній системі в документах різного вигляду – чисельному, текстовому, графічному, тощо. Інформацію зазвичай розглядають на різних рівнях (наприклад, синтаксис, семантика, прагматика). Синтаксис пов'язаний з формою та структурою інформаційних повідомлень. Семантика визначає інформаційні смисли. Прагматика – це цінність інформації, зокрема, для системи управління.

Інформаційні технології визначають методи та засоби обробки інформації. Це стосується як первинної обробки (наприклад, в процесі збору та реєстрації), так і вторинної (переробка, зберігання та видача). Для виконання операцій у технічному процесі обробки інформації використовується поєднання технічних і програмних засобів.

Організаційна одиниця управління складається з управлінського персоналу (користувачів), який структурно розділений і використовує засоби ІС для виконання функцій управління в інформаційних технологіях. Інформаційні технології в першу чергу вибираються в рамках організаційної структури управління. Реалізація адміністративних функцій для організаційної одиниці передбачає встановлення завдань, визначення конфігурації вхідної та вихідної інформації, проектування інформаційних технологій та розробку інтерфейсів користувача. Обов'язком управлінського персоналу є введення первинних даних в ІС, оцінка та вибір альтернативних варіантів управління та здійснення управлінського контролю над цілями управління.

Функціональна структура ІС складається з набору функціональних підсистем, наборів завдань і процедур обробки інформації, які реалізують функціональні можливості системи управління. У системах управління великими підприємствами виділяють самостійні підсистеми (контури) функціонального та організаційного рівнів управління (рис.3.2).



Рисунок 3.2 - Функціональні компоненти ІС

Система підтримки прийняття рішень (рис. 3.3) є базовою складовою загальної ІС підприємства і, в основному, має три підсистеми, а саме: систему

управління даними, базу моделей рішень (в тому числі програмні реалізації) та користувальницький інтерфейс.

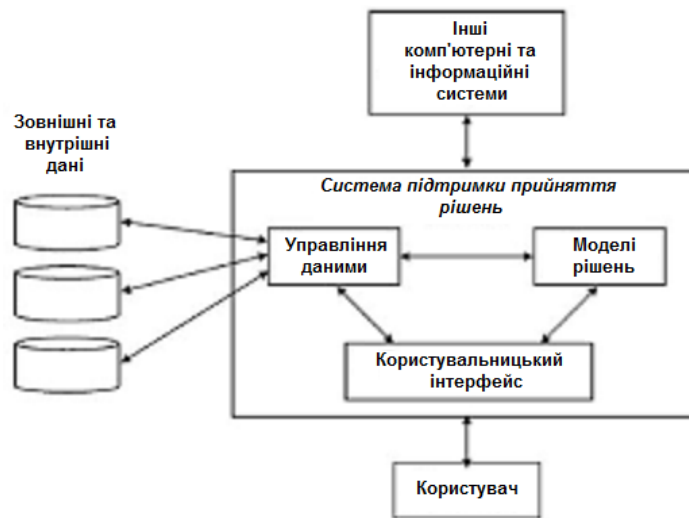


Рисунок 3.3 – Спрощена структура системи підтримки прийняття рішень

Розробимо одну з можливих моделей в цій підсистемі – модель вибору комбайнів методом аналізу ієрархій.

3.2. Розробка структурної ієрархічної моделі вибору комбайнів

Для створення моделі методом МАІ треба обрати альтернативи, з яких потрібно обрати найкраще рішення. В даному випадку такими альтернативами обираємо комбайни, які були розглянуті в розділі 2.2.

В якості альтернатив були обрані наступні моделі зернозбиральних комбайнів: John Deere S690; Claas Lexion 8900; New Holland CR1090; Case IH Axial Flow 8250.

Таким чином, верхнім рівнем ієрархії буде задача вибору комбайнів. На другому рівні будуть чотири групи критеріїв. Третій рівень - окремі критерії у відповідних групах (кількість та опис критеріїв розглянуті в розділі 2.2). Четвертий рівень - чотири альтернативи, які наведені вище.

Структурна ієрархічна модель вибору зернозбиральних комбайнів наведена на рис. 3.4.

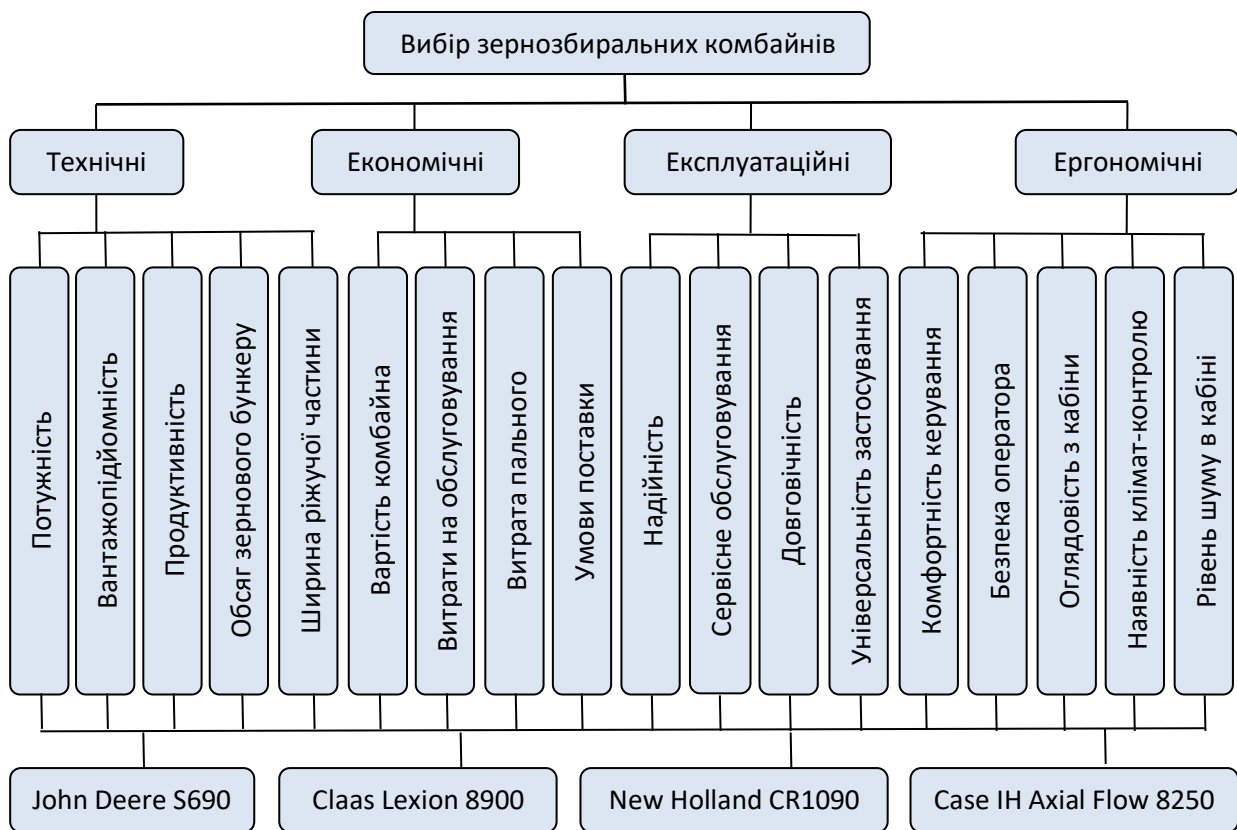


Рисунок 3.4 – Структурна ієрархічна модель вибору комбайнів

Реалізуємо дану модель тими інструментами, які ми обрали та описали в розділі 2.3.

3.3. Програмна реалізація моделі

3.3.1. Підготовка та формування матриць порівнянь груп та критеріїв

Першим етапом реалізації моделі є імпорт необхідних бібліотек для роботи з даними (рис. 3.5).

```

%matplotlib inline
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [16, 10]
import seaborn as sns
import random
  
```

Рисунок 3.5 - Імпорт основних бібліотек

Далі напишемо декілька підпрограм-функцій для обчислення геометричного середнього (рис.3.6, хоча можна і скористатись функцією `geometric_mean` модуля `statistics`), знаходження вагових коефіцієнтів матриць парних порівнянь (рис.3.7) та обчислення індексу узгодженості (рис.3.8).

```
def geom_mean(arr):
    return arr.prod() ** (1/arr.shape[0])
```

Рисунок 3.6 – Функція геометричного середнього

```
def koef_w(d_f, name_koef):
    d_f[name_koef] = 0
    lst = []
    for i in range(d_f.shape[0]):
        lst.append(geom_mean(d_f.iloc[i, :-1]))
    d_f[name_koef] = lst
    sm = d_f[name_koef].sum()
    d_f[name_koef] /= sm
```

Рисунок 3.7 – Функція визначення вагових коефіцієнтів в матрицях парних порівнянь

```
def index_c(d_f):
    n = d_f.shape[0]
    ind_list = []
    for i in range(n):
        ind_list.append(d_f.iloc[:, i].sum() * d_f.iloc[i, -1])
    return (sum(ind_list)-n)/(n-1)
```

Рисунок 3.8 – Функція обчислення індексу узгодженості

Після введення даних для матриць парних порівнянь можна обчислювати вагові коефіцієнти та індекси узгодженості, які мають бути менше, ніж 0,1. Матриця порівнянь груп критеріїв з обрахуванням вагових коефіцієнтів та індексу узгодженості наведена на рис.3.9. Аналогічні дії проводимо над матрицями окремих критеріїв, зокрема: технічних (рис. 3.10), економічних (рис. 3.11), експлуатаційних (рис. 3.12) та ергономічних (рис. 3.13).

```

koef_w(df_gr, 'W_gr')
df_gr

```

	Технічні	Економічні	Експлуатаційні	Ергономічні	W_gr
Технічні	1.000000	3.0	3.000000	3.0	0.474717
Економічні	0.333333	1.0	1.000000	5.0	0.236622
Експлуатаційні	0.333333	1.0	1.000000	3.0	0.208254
Ергономічні	0.333333	0.2	0.333333	1.0	0.080406

```

[ ] ind_gr = index_c(df_gr)
ind_gr

```

0.08514553818927606

Рисунок 3.9 – Матриця порівнянь груп критеріїв та індекс узгодженості

```

koef_w(df_tech, 'W_tech')
df_tech

```

	Потужність	Вантажопідйомність	Продуктивність	Обсяг бункеру	Робоча ширина	W_tech
Потужність	1.0	0.333333	1.000000	1.0	1.0	0.153057
Вантажопідйомність	3.0	1.000000	1.000000	1.0	3.0	0.295888
Продуктивність	1.0	1.000000	1.000000	1.0	3.0	0.237521
Обсяг бункеру	1.0	1.000000	1.000000	1.0	1.0	0.190668
Робоча ширина	1.0	0.333333	0.333333	1.0	1.0	0.122865

```

[ ] ind_tech = index_c(df_tech)
ind_tech

```

0.06117802256709037

Рисунок 3.10 – Порівняння технічних критеріїв та індекс узгодженості

```

koef_w(df_econ, 'W_econ')
df_econ

```

	Вартість	Витрати на ТО	Витрата пального	Умови поставки	W_econ
Вартість	1.0	0.333333	0.333333	1.0	0.131655
Витрати на ТО	3.0	1.000000	1.000000	3.0	0.394966
Витрата пального	3.0	1.000000	1.000000	1.0	0.300110
Умови поставки	1.0	0.333333	1.000000	1.0	0.173268

```

[ ] ind_econ = index_c(df_econ)
ind_econ

```

0.04882122225027589

Рисунок 3.11 – Порівняння економічних критеріїв та індекс узгодженості

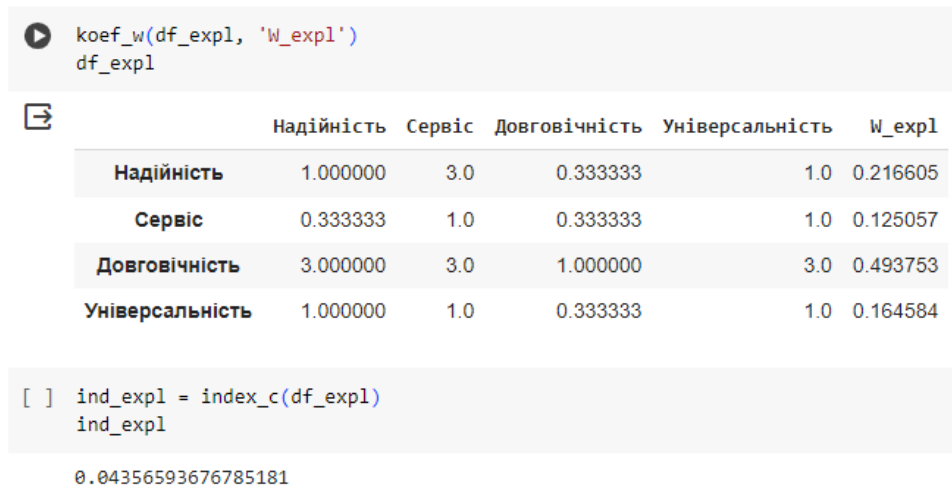


Рисунок 3.12 – Порівняння експлуатаційних критеріїв та індекс узгодженості

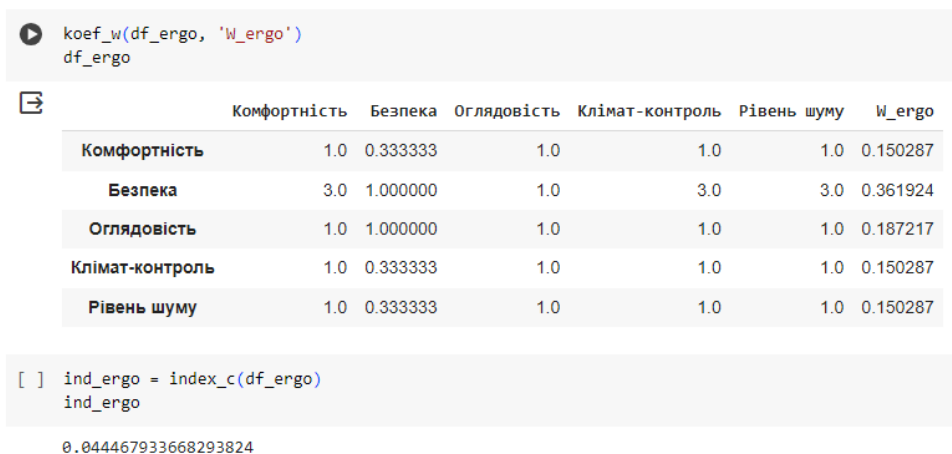


Рисунок 3.13 – Порівняння ергономічних критеріїв та індекс узгодженості

Як можна побачити – всі індекси узгодженості мають значення, менші ніж 0,1. Це означає, що матриці парних порівнянь узгоджені, і можна проводити подальші розрахунки.

3.3.2. Матриці порівнянь альтернатив по окремим критеріям

На цьому етапі формуємо матриці парних порівнянь для альтернатив по кожному критерію. Кожного разу розраховуємо вагові коефіцієнти та індекси узгодженості. Так, по технічній групі відповідні розрахунки наведені по наступним критеріям: потужність (рис.3.14), вантажопідйомність (рис. 3.15),

продуктивність (рис. 3.16), обсяг бункеру (рис.3.17), ширина робочої частини (рис.3.18).

```

koef_w(df_tech_pow, 'W_tech_pow')
df_tech_pow

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_tech_pow
JD S690	1.0	0.333333	1.000000	1.0	0.173268
CL 8900	3.0	1.000000	1.000000	3.0	0.394966
NH CR1090	1.0	1.000000	1.000000	3.0	0.300110
CIAF 8250	1.0	0.333333	0.333333	1.0	0.131655

```

[ ] ind_tech_pow = index_c(df_tech_pow)
ind_tech_pow

0.048821222250276186

```

Рисунок 3.14 – Порівняння альтернатив та узгодженість по потужності

```

koef_w(df_tech_cargo, 'W_tech_cargo')
df_tech_cargo

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_tech_cargo
JD S690	1.000000	3.0	1.000000	0.333333	0.20096
CL 8900	0.333333	1.0	0.333333	0.200000	0.07759
NH CR1090	1.000000	3.0	1.000000	0.333333	0.20096
CIAF 8250	3.000000	5.0	3.000000	1.000000	0.52049

```

[ ] ind_tech_cargo = index_c(df_tech_cargo)
ind_tech_cargo

0.015411592977224942

```

Рисунок 3.15 – Порівняння альтернатив та узгодженість по вантажопідйомності

```

koef_w(df_tech_prod, 'W_tech_prod')
df_tech_prod

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_tech_prod
JD S690	1.000000	3.0	0.333333	0.333333	0.150402
CL 8900	0.333333	1.0	0.333333	0.200000	0.076424
NH CR1090	3.000000	3.0	1.000000	0.333333	0.260504
CIAF 8250	3.000000	5.0	3.000000	1.000000	0.512669

```

[ ] ind_tech_prod = index_c(df_tech_prod)
ind_tech_prod

0.0642363476235334

```

Рисунок 3.16 – Порівняння альтернатив та узгодженість по продуктивності


```

koef_w(df_tech_bunk, 'W_tech_bunk')
df_tech_bunk

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_tech_bunk
JD S690	1.0	0.333333	1.0	2.0	0.187633
CL 8900	3.0	1.000000	3.0	5.0	0.537817
NH CR1090	1.0	0.333333	1.0	1.0	0.157780
CIAF 8250	0.5	0.200000	1.0	1.0	0.116770

```

[ ] ind_tech_bunk = index_c(df_tech_bunk)
ind_tech_bunk

0.011171899219334591

```

Рисунок 3.17 – Порівняння альтернатив та узгодженість по обсягу бункеру

```

koef_w(df_tech_width, 'W_tech_width')
df_tech_width

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_tech_width
JD S690	1.0	0.333333	2.0	1.0	0.185541
CL 8900	3.0	1.000000	5.0	3.0	0.531821
NH CR1090	0.5	0.200000	1.0	0.5	0.097097
CIAF 8250	1.0	0.333333	2.0	1.0	0.185541

```

[ ] ind_tech_width = index_c(df_tech_width)
ind_tech_width

0.001550599570493046

```

Рисунок 3.18 – Порівняння альтернатив та узгодженість по ширині робочої частини

По економічній групі аналогічні розрахунки проведені для таких критеріїв: вартість (рис. 3.19), витрати на ТО (рис.3.20), витрата пального (рис. 3.21), умови поставки (рис. 3.22).

```

koef_w(df_econ_price, 'W_econ_price')
df_econ_price

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_econ_price
JD S690	1.0	0.333333	2.0	1.0	0.195416
CL 8900	3.0	1.000000	3.0	3.0	0.492973
NH CR1090	0.5	0.333333	1.0	0.5	0.116195
CIAF 8250	1.0	0.333333	2.0	1.0	0.195416

```

[ ] ind_econ_price = index_c(df_econ_price)
ind_econ_price

0.021693484640344114

```

Рисунок 3.19 – Порівняння альтернатив та узгодженість по вартості

```

▶ koef_w(df_econ_cost, 'W_econ_cost')
df_econ_cost

JD S690  CL 8900  NH CR1090  CIAF 8250  W_econ_cost
JD S690  1.000000  2.0  3.0  1.000000  0.351187
CL 8900  0.500000  1.0  2.0  0.500000  0.188687
NH CR1090 0.333333  0.5  1.0  0.333333  0.108939
CIAF 8250 1.000000  2.0  3.0  1.000000  0.351187

[ ] ind_econ_cost = index_c(df_econ_cost)
ind_econ_cost

0.002762929496850456

```

Рисунок 3.20 – Порівняння альтернатив та узгодженість по витратам на ТО

```

▶ koef_w(df_econ_fuel, 'W_econ_fuel')
df_econ_fuel

JD S690  CL 8900  NH CR1090  CIAF 8250  W_econ_fuel
JD S690  1.0  0.333333  0.333333  0.333333  0.099105
CL 8900  3.0  1.000000  1.000000  1.000000  0.297315
NH CR1090 3.0  1.000000  1.000000  2.000000  0.353569
CIAF 8250 3.0  1.000000  0.500000  1.000000  0.250011

[ ] ind_econ_fuel = index_c(df_econ_fuel)
ind_econ_fuel

0.022419908242303777

```

Рисунок 3.21 – Порівняння альтернатив та узгодженість по витратам пального

```

▶ koef_w(df_econ_deliv, 'W_econ_deliv')
df_econ_deliv

JD S690  CL 8900  NH CR1090  CIAF 8250  W_econ_deliv
JD S690  1.000000  1.000000  3.0  3.0  0.373594
CL 8900  1.000000  1.000000  3.0  3.0  0.373594
NH CR1090 0.333333  0.333333  1.0  2.0  0.148094
CIAF 8250 0.333333  0.333333  0.5  1.0  0.104718

[ ] ind_econ_deliv = index_c(df_econ_deliv)
ind_econ_deliv

0.015222177289760866

```

Рисунок 3.22 – Порівняння альтернатив та узгодженість по умовам поставки

Аналогічні розрахунки проводимо в групі експлуатаційних критеріїв: надійність (рис.3.23), сервіс (рис.3.24), довговічність (рис.3.25), універсальність (рис.3.26).

```

koef_w(df_expl_rel, 'W_expl_rel')
df_expl_rel

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_expl_rel
JD S690	1.0	1.0	0.333333	0.200000	0.106181
CL 8900	1.0	1.0	0.333333	0.333333	0.120645
NH CR1090	3.0	3.0	1.000000	1.000000	0.361935
CIAF 8250	5.0	3.0	1.000000	1.000000	0.411238

```

[ ] ind_expl_rel = index_c(df_expl_rel)
ind_expl_rel

0.011312704372404466

```

Рисунок 3.23 – Порівняння альтернатив та узгодженість по надійності

```

koef_w(df_expl_serv, 'W_expl_serv')
df_expl_serv

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_expl_serv
JD S690	1.0	0.333333	1.0	1.0	0.186421
CL 8900	3.0	1.000000	1.0	1.0	0.322891
NH CR1090	1.0	1.000000	1.0	1.0	0.245344
CIAF 8250	1.0	1.000000	1.0	1.0	0.245344

```

[ ] ind_expl_serv = index_c(df_expl_serv)
ind_expl_serv

0.052527226516777624

```

Рисунок 3.24 – Порівняння альтернатив та узгодженість по сервісу

```

koef_w(df_expl_dur, 'W_expl_dur')
df_expl_dur

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_expl_dur
JD S690	1.0	0.333333	1.0	1.0	0.178712
CL 8900	3.0	1.000000	1.0	3.0	0.407376
NH CR1090	1.0	1.000000	1.0	1.0	0.235199
CIAF 8250	1.0	0.333333	1.0	1.0	0.178712

```

[ ] ind_expl_dur = index_c(df_expl_dur)
ind_expl_dur

0.05722713723512977

```

Рисунок 3.25 – Порівняння альтернатив та узгодженість по довговічності

```

koef_w(df_expl_univ, 'W_expl_univ')
df_expl_univ

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_expl_univ
JD S690	1.0	0.333333	1	1.0	0.178712
CL 8900	3.0	1.000000	3	1.0	0.407376
NH CR1090	1.0	0.333333	1	1.0	0.178712
CIAF 8250	1.0	1.000000	1	1.0	0.235199

```

[ ] ind_expl_univ = index_c(df_expl_univ)
ind_expl_univ

0.05722713723512918

```

Рисунок 3.26 – Порівняння альтернатив та узгодженість по універсальності

Залишається ергономічна група: комфортність (рис.3.27), безпека (рис.3.28), оглядовість (рис.3.29), клімат-контроль (рис.3.30), рівень шуму (рис.3.31).

```

koef_w(df_ergo_comf, 'W_ergo_comf')
df_ergo_comf

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_ergo_comf
JD S690	1.000000	3.0	1.000000	0.333333	0.20096
CL 8900	0.333333	1.0	0.333333	0.200000	0.07759
NH CR1090	1.000000	3.0	1.000000	0.333333	0.20096
CIAF 8250	3.000000	5.0	3.000000	1.000000	0.52049

```

[ ] ind_ergo_comf = index_c(df_ergo_comf)
ind_ergo_comf

0.015411592977224942

```

Рисунок 3.27 – Порівняння альтернатив та узгодженість по комфортності

```

koef_w(df_ergo_safe, 'W_ergo_safe')
df_ergo_safe

```

	JD S690	CL 8900	NH CR1090	CIAF 8250	W_ergo_safe
JD S690	1.000000	3.0	0.333333	0.333333	0.150402
CL 8900	0.333333	1.0	0.333333	0.200000	0.076424
NH CR1090	3.000000	3.0	1.000000	0.333333	0.260504
CIAF 8250	3.000000	5.0	3.000000	1.000000	0.512669

```

[ ] ind_ergo_safe = index_c(df_ergo_safe)
ind_ergo_safe

0.0642363476235334

```

Рисунок 3.28 – Порівняння альтернатив та узгодженість по безпеці

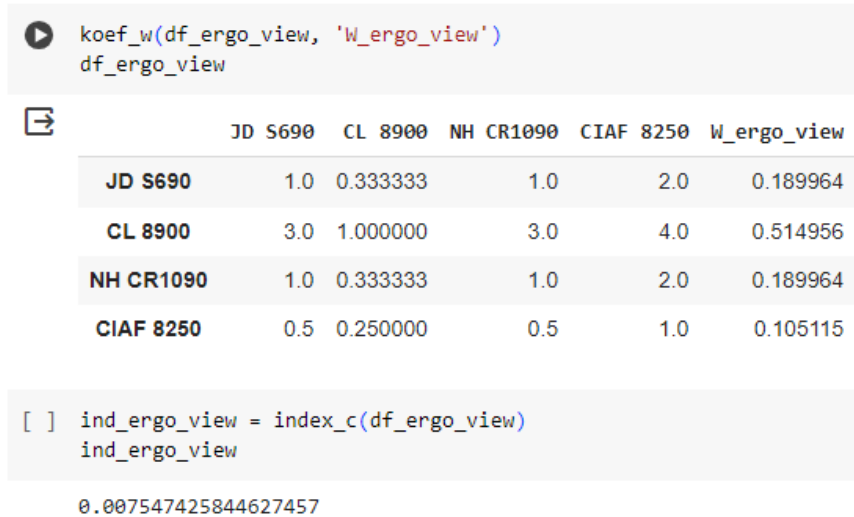


Рисунок 3.29 – Порівняння альтернатив та узгодженість по оглядовості

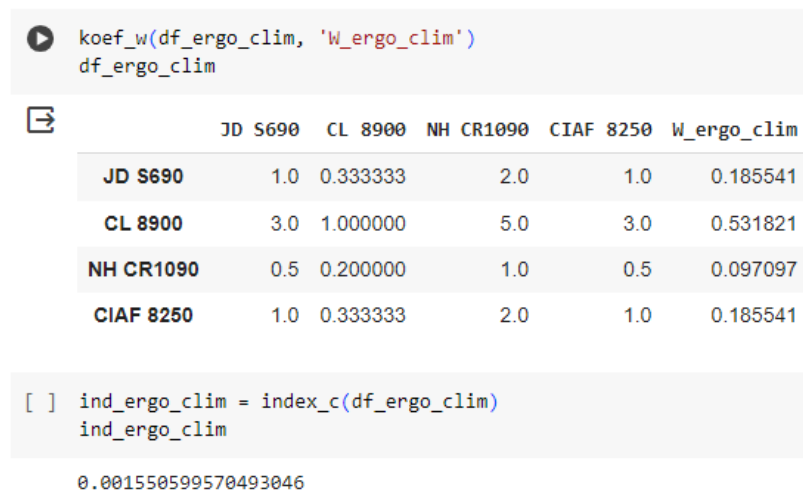


Рисунок 3.30 – Порівняння альтернатив та узгодженість по клімат-контролю



Рисунок 3.31 – Порівняння альтернатив та узгодженість по рівню шуму

3.3.3. Розрахунки по ієрархічним рівням

Черговий крок полягає в проведенні розрахунків коефіцієнтів для альтернатив по груповим та узагальненому критеріям. Ці розрахунки представлені на наступних рисунках: по технічним критеріям (рис. 3.32), по економічним (рис.3.33), по експлуатаційним (рис.3.34), по ергономічним (рис.3.35).

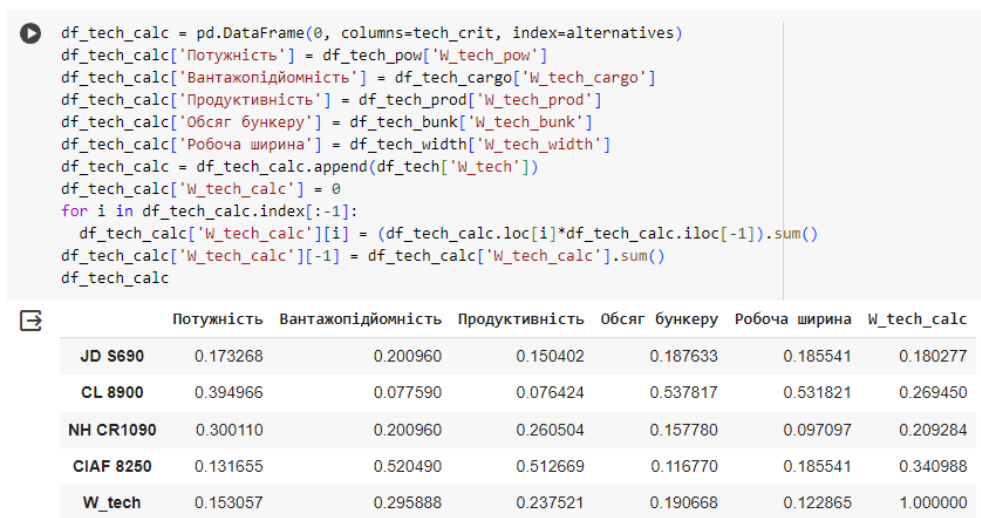


Рисунок 3.32 – Коефіцієнти для альтернатив по технічним критеріям

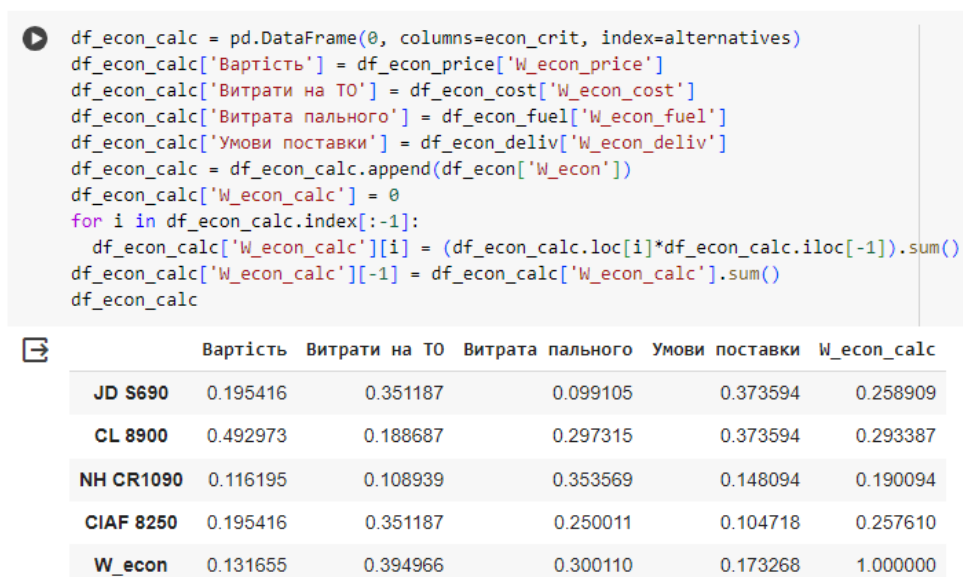


Рисунок 3.33 – Коефіцієнти для альтернатив по економічним критеріям

```

df_expl_calc = pd.DataFrame(0, columns=expl_crit, index=alternatives)
df_expl_calc['Надійність'] = df_expl_rel['W_expl_rel']
df_expl_calc['Сервіс'] = df_expl_serv['W_expl_serv']
df_expl_calc['Довговічність'] = df_expl_dur['W_expl_dur']
df_expl_calc['Універсальність'] = df_expl_univ['W_expl_univ']
df_expl_calc = df_expl_calc.append(df_expl['W_expl'])
df_expl_calc['W_expl_calc'] = 0
for i in df_expl_calc.index[:-1]:
    df_expl_calc['W_expl_calc'][i] = (df_expl_calc.loc[i]*df_expl_calc.iloc[-1]).sum()
df_expl_calc['W_expl_calc'][-1] = df_expl_calc['W_expl_calc'].sum()
df_expl_calc

```

	Надійність	Сервіс	Довговічність	Універсальність	W_expl_calc
JD S690	0.106181	0.186421	0.178712	0.178712	0.163966
CL 8900	0.120645	0.322891	0.407376	0.407376	0.334703
NH CR1090	0.361935	0.245344	0.235199	0.178712	0.254623
CIAF 8250	0.411238	0.245344	0.178712	0.235199	0.246708
W_expl	0.216605	0.125057	0.493753	0.164584	1.000000

Рисунок 3.34 – Розрахунок коефіцієнтів для альтернатив по експлуатаційним критеріям

```

df_ergo_calc = pd.DataFrame(0, columns=ergo_crit, index=alternatives)
df_ergo_calc['Комфортність'] = df_ergo_comf['W_ergo_comf']
df_ergo_calc['Безпека'] = df_ergo_safe['W_ergo_safe']
df_ergo_calc['Оглядовість'] = df_ergo_view['W_ergo_view']
df_ergo_calc['Клімат-контроль'] = df_ergo_clim['W_ergo_clim']
df_ergo_calc['Рівень шуму'] = df_ergo_noise['W_ergo_noise']
df_ergo_calc = df_ergo_calc.append(df_ergo['W_ergo'])
df_ergo_calc['W_ergo_calc'] = 0
for i in df_ergo_calc.index[:-1]:
    df_ergo_calc['W_ergo_calc'][i] = (df_ergo_calc.loc[i]*df_ergo_calc.iloc[-1]).sum()
df_ergo_calc['W_ergo_calc'][-1] = df_ergo_calc['W_ergo_calc'].sum()
df_ergo_calc

```

	Комфортність	Безпека	Оглядовість	Клімат-контроль	Рівень шуму	W_ergo_calc
JD S690	0.200960	0.150402	0.189964	0.185541	0.195416	0.177453
CL 8900	0.077590	0.076424	0.514956	0.531821	0.492973	0.289742
NH CR1090	0.200960	0.260504	0.189964	0.097097	0.116195	0.192104
CIAF 8250	0.520490	0.512669	0.105115	0.185541	0.195416	0.340702
W_ergo	0.150287	0.361924	0.187217	0.150287	0.150287	1.000000

Рисунок 3.35 – Розрахунок коефіцієнтів для альтернатив по ергономічним критеріям

І, нарешті, останній крок полягає в розрахунку узагальненого критерію та вибору альтернативи з максимальним значенням узагальненого критерію (рис.3.36).

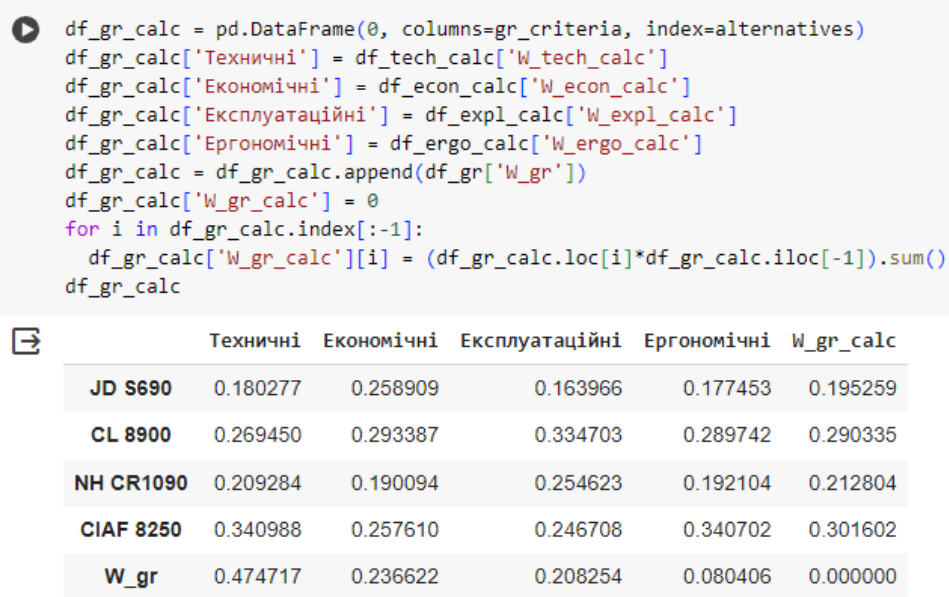


Рисунок 3.36 – Розрахунок коефіцієнтів для альтернатив по узагальненому критерію

Після розрахунку вибір найкращої альтернативи не є проблематичним (рис. 3.37).

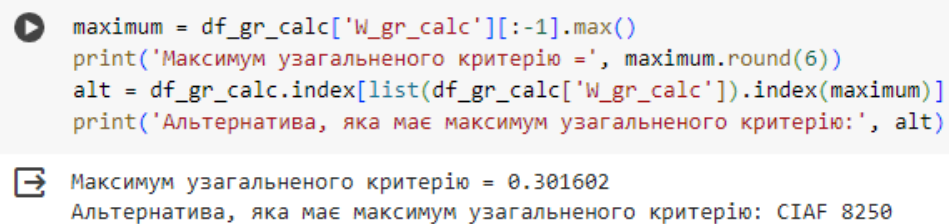


Рисунок 3.37 - Вибір альтернативи з максимальним значенням узагальненого критерію

Ми бачимо, що в нашому випадку найкращою альтернативою є зернозбиральний комбайн Case IH Axial Flow 8250. Аналізуючи матрицю результатів на рис.3.36, можна дійти висновку, що цей комбайн програє «конкурентам» по економічним та експлуатаційним критеріям, але значно випереджає їх по технічним і ергономічним показникам.

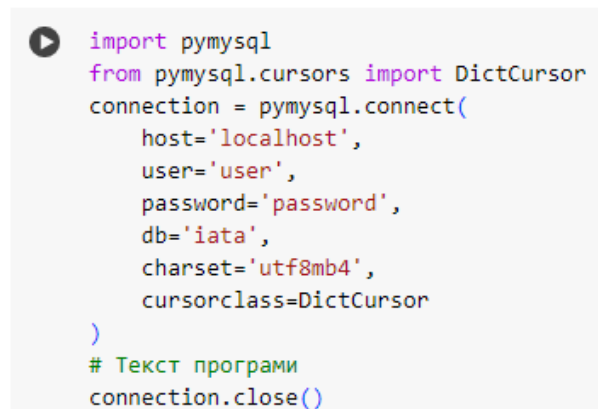
3.4. Організація обміну з базою даних

Для організації обміну програми на Python з будь-якою базою даних потрібні відповідні бібліотеки. В нашому випадку – це пакет PyMySQL. Це реалізація mysql-клієнта на чистому Python, тому ніякі додаткові C-бібліотеки ставити не доведеться.

Для встановлення бібліотеки виконуємо стандартну команду:

```
pip install PyMySQL
```

Наступний крок – імпорт необхідних пакетів і підпрограм, після чого можна вже організувати з'єднання з базою (цей об'єкт носить назву коннектор). Приклад цього показаний на рис.3.38.



```
import pymysql
from pymysql.cursors import DictCursor
connection = pymysql.connect(
    host='localhost',
    user='user',
    password='password',
    db='iata',
    charset='utf8mb4',
    cursorclass=DictCursor
)
# Текст програми
connection.close()
```

Рисунок 3.38 – Приклад організації коннектора зв'язку з БД

Можна звернути увагу, що був імпортований клас DictCursor і переданий до функції connect. Це потрібно для того, щоб отримати результат у вигляді словника, де ключами будуть назви колонок. Якщо програмісту зручно працювати з tuple, то можна нічого не передавати в cursorclass.

Рекомендується завжди явно закривати відкрите з'єднання з базою MySQL шляхом виклику метода close у об'єкта connection. Частіше за все розробники

прибігають до конструкції try/finally, але можна використовувати і closing з contextlib. Це можна побачити на рис. 3.39.

```

▶ from contextlib import closing
import pymysql
from pymysql.cursors import DictCursor
with closing(pymysql.connect(...)) as connection:
    # Текст програми контекстного блоку

```

Рисунок 3.39 – Використання контекстного блоку closing

В цьому випадку при виході з контекстного блоку автоматично буде здійснено виклик методу close. Щоб розпочати взаємодію з MySQL, необхідно створити курсор.

Приклад того, як це зробити – показано на рис. 3.40.

```

▶ with closing(pymysql.connect(...)) as connection:
    with connection.cursor() as cursor:
        query = """
        SELECT
            criteria_code
        FROM
            criterias
        ORDER BY
            criteria_code DESC
        LIMIT 5
        """
        cursor.execute(query)
        for row in cursor:
            print(row)
{'criteria_code': 'ZZW'}
{'criteria_code': 'ZZU'}
{'criteria_code': 'ZZO'}
{'criteria_code': 'ZZG'}
{'criteria_code': 'ZYR'}
# Пишемо в базу

```

Рисунок 3.40 – Створення курсору для взаємодії з базою даних

Наведемо приклад, як виглядає INSERT в найпростішу таблицю з однією колонкою. Опис схеми таблиці показаний на рис.3.41. Код вставки та читання результату операції наведений на рис.3.42.

```

▶ CREATE TABLE IF NOT EXISTS criterias (
  criteria VARCHAR(50)
) ENGINE=INNODB;

```

Рисунок 3.41 – Опис схеми простої таблиці з однією колонкою

```

▶ with closing(pymysql.connect(...)) as conn:
  with conn.cursor() as cursor:
    criterias = [
      'Технічні',
      'Економічні',
      'Експлуатаційні',
      'Ергономічні'
    ]
    query = 'INSERT INTO criterias (criteria) VALUES (%s)'
    cursor.executemany(query, criterias)
    # необхідно, тому що за-замовченням commit діє тільки після виходу
    # з контекстного менеджера, інакше ми б не побачили критеріїв
    conn.commit()
  with conn.cursor() as cursor:
    query = 'SELECT criteria FROM criterias'
    cursor.execute(query)
    for row in cursor:
      print(row['criteria'])

```

Рисунок 3.42 – Код вставки та читання результату операції

Таким чином, для операцій в базі даних використовують, як правило, стандартні SQL-запити, які можна створювати в Python у вигляді текстових строчок. Відповідно, створення, редагування та інші операції над SQL-запитами в Python – це фактично робота з текстовими даними і відправка їх через конектор до СУБД.

3.5. Елементи графічного інтерфейсу користувача

Для взаємодії з програмною частиною загальноприйнятно розробляти графічні інтерфейси користувача (graphic user interface, GUI).

На цьому шляху основною перепоною є велика кількість різноманітних платформ, систем та пристроїв. Розробка своєї версії для кожного з них потребує великої кількості ресурсів та часу. Однак є і добрі новини. Використовуючи сучасні інтернет-браузери, можна не перейматись проблемами сумісності з пристроями та операційними системами. Уніфікація

підходів в створенні гіпертекстових сторінок з таблицями стилів та додатковими скриптами дозволяє розробникам зробити такі інтерфейси швидко і з мінімальними ресурсами. Додатки, якими можна користуватись через веб-браузер, називаються веб-додатками. Вони схожі на звичайні веб-сайти, але є і суттєві відмінності.

Веб-сайт є статичним ресурсом, яким можна переглядати інформацію. Він зазвичай складається з кількох веб-сторінок, пов'язаних між собою гіперпосиланнями. Веб-сайти використовуються для надання інформації, реклами товарів або послуг, демонстрації фотографій та відео та багато іншого. Веб-сайт може бути створений для персонального використання, індивідуального підприємництва або бізнесу.

З іншого боку, веб-додаток є динамічним ресурсом, за допомогою якого користувач може взаємодіяти з різними функціями та виконувати різні дії. Веб-додаток пропонує складніші можливості, такі як реєстрація користувачів, авторизація, обробка даних, завантаження файлів та інші. Веб-додатки широко використовуються в електронній комерції, соціальних мережах, системах управління контентом та інших областях.

Графічний інтерфейс в веб-додатках в основному використовує віконний вигляд, що дозволяє розробникам задіяти подієво-орієнтовані бібліотеки, в яких та чи інша частина програмного коду виконується після настання тих чи інших подій.

В даній роботі не ставилась задача розробки повноцінного GUI, однак треба показати можливості по його створенню. В даному випадку мова йде про окремі елементи, які можуть бути використані розробниками при створенні повноцінного веб-додатку.

Для створення GUI зазвичай використовують графічні віконні Python-бібліотеки, такі, наприклад, як tk-inter або інші. Завдяки цим бібліотекам розробнику достатньо брати готові шаблони або елементи-віджети (об'єкти, породжені вбудованими класами), задавати їм необхідні параметри та зв'язки, а потім розміщувати на екрані відповідно до дизайну.

Основна графічна одиниця діалогового GUI – форма.

Форми в програмах на основі інтерфейсу користувача пропонують покращену взаємодію з користувачем завдяки їхньому інтерфейсу, спрощуючи взаємодію агента та підтримуючи контекст під час обробки зв'язаних записів.

Поширеною практикою є використання редактора форм для редагування та додавання або зміни елементів у формі. Редактор форм також надає зручний інтерфейс для редагування форм у різних програмах.

Дані вводяться у форми, аналізуються, а потім виводяться. Джерелом даних форми може бути запис таблиці або запит.

Форма надає такі можливості:

- введення та відображення інформації бази даних;
- редагування даних;
- друк;
- генерація повідомлень тощо.

Приклад форми для створення моделі наведено на рис.3.43.

Рисунок 3.43 – Форма «Створення моделі»

За допомогою форм можна також генерувати різноманітні запити до баз даних та отримувати з неї звіти. Завдяки формам на генерацію запитів користувач може вести діалог з програмою на зрозумілій мові, навіть не

здогадуючись про те, що програма по отриманій інформації буде формувати, можливо, дуже складний текстовий SQL-запит відповідно правилам СУБД.

Запити є основними інструментами для отримання інформації з однієї або кількох таблиць БД. За допомогою запитів можна вносити зміни в саму базу даних. Запити служать джерелами даних для форм, звітів і сторінок доступу до даних. Результатом запиту зазвичай є нова таблиця-звіт, яку можна переглянути, проаналізувати, а потім зберегти чи скасувати збереження.

Використання запитів дозволяє вирішувати численні завдання без знання програмування. Наприклад, щоб відобразити дані в зведеній формі, можна виконувати обчислення полів бази даних, групувати записи та використовувати різні статистичні функції для пошуку зведених значень для полів.

Запити на вибір, запити перехресних посилань і запити на дії є трьома типами питань, які зазвичай може генерувати СУБД.

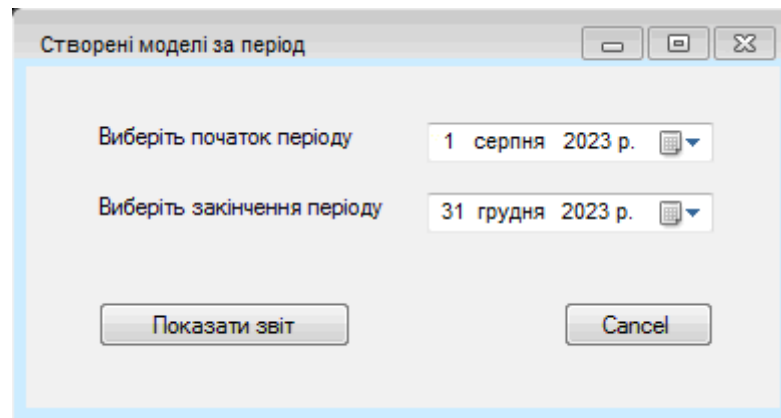
Запити на вибір є найпоширенішим типом запиту. Вони забезпечують такі функції: вибирає записи, які відповідають критеріям відбору; вставляє в потрібному порядку поля з однієї чи кількох таблиць у отриману таблицю; виконує обчислення полів бази даних та різні статистичні розрахунки для груп наборів даних.

Перехресний запит — це особливий вид підсумкового запиту. Результати розрахунку для значень полів відображаються у вигляді перехресної таблиці, в якій значення обраних стовпців створюють заголовки рядків та стовпців зі значень певного поля, а кінцеві значення обчислюються на їх перетинах.

Запит на дію — це запит, який вносить зміни та оновлює інформацію в самій базі даних.

Кінцевим користувачам зручно робити запити через форми. Користувач вводить лише необхідні параметри, а після натискання кнопки, наприклад, «Показати звіт» програма замінює параметри підготовленим шаблоном запиту та відправляє його в базу даних.

Приклад простої форми для генерації запиту на отримання звіту наведений на рис.3.44.



Створені моделі за період

Виберіть початок періоду 1 серпня 2023 р.

Виберіть закінчення періоду 31 грудня 2023 р.

Показати звіт Cancel

Рисунок 3.44 – Форма для запити «Створені моделі за період»

Висновки до розділу

В результаті виконання даного розділу були досягнуті такі результати:

- розроблена структурна схема інформаційної системи;
- створена структурна ієрархічна модель вибору комбайнів;
- проведена реалізація моделі обраним інструментарієм та перевірена її робота на навчальних даних;
- описана організація обміну даними між програмною частиною та БД;
- запропоновані окремі елементи графічного інтерфейсу користувача.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Робота з інформаційною системою передбачає наявність обчислювальних засобів (комп'ютерів) та автоматично означає, що на робочому місці можуть бути присутніми шкідливі і небезпечні фактори, які характерні для електроустановок.

У процесі роботи з електрообладнанням працівники можуть піддаватися впливу наступних шкідливих і (або) небезпечних виробничих факторів:

- підвищений рівень шуму на робочому місці;
- підвищений рівень вібрації на робочому місці;
- висока або низька температура повітря в робочій зоні, а також поверхонь обладнання та матеріалів;
- підвищена або знижена вологість повітря;
- підвищена запиленість повітря в робочій зоні;
- недостатня освітленість робочої зони;
- підвищена напруга в електричній мережі, яка може бути замкнута через тіло людини.

Залежно від умов праці, в яких використовується електрообладнання, працівники можуть піддаватися впливу і інших шкідливих і (або) небезпечних виробничих факторів.

З урахуванням шкідливих і (або) небезпечних виробничих факторів, що впливають на них, працівники забезпечуються засобами індивідуального захисту відповідно до законодавства.

Працівник має право відмовитися від виконання дорученої роботи з електроустановками у разі безпосередньої небезпеки для життя і здоров'я себе та оточуючих до усунення цієї небезпеки, а також у разі незабезпечення його засобами індивідуального захисту, що безпосередньо забезпечують безпеку праці.

Не дозволяється з'являтися на роботі працівників у стані алкогольного, наркотичного або токсичного сп'яніння, а також розпивати алкогольні напої, вживати наркотичні засоби, психотропні речовини, їх аналоги, токсичні речовини в робочий час або за місцем роботи.

Перед початком роботи з електрообладнанням оператор повинен переконаватися шляхом зовнішнього огляду, що:

- корпус обладнання не має тріщин та інших пошкоджень;
- кабель (шнур), його захисна трубка і вилка справні.

При виконанні робіт працівник повинен користуватися електрообладнанням, клас якого відповідає категорії приміщення і умовам використання в роботі, із застосуванням електрозахисних засобів в окремих випадках, в залежності від місця роботи.

Електричні кабелі повинні бути захищені від випадкових механічних пошкоджень і контакту з гарячими, вологими і маслянистими поверхнями. Не допускається розтягування, скручування і згинання кабелю, навантаження на нього, а також допускати його перетину з предметами, які можуть пошкодити ізоляцію.

При виконанні робіт із застосуванням електрообладнання працівник зобов'язаний:

- застосовувати безпечні методи і прийоми роботи, дотримуватись вимог охорони праці;
- підтримувати чистоту на робочому місці, не захаращувати проходи;
- оберігати ізоляцію кабелю (шнура) електрообладнання та проводів від обриву, механічних та інших пошкоджень;
- обережно поводитись з електрообладнанням, не піддавати його ударам, перевантаженням, бруду, вологи, нафтопродуктам, розчинникам тощо.

Оператору забороняється експлуатувати електрообладнання, якщо під час роботи виникають такі несправності:

- пошкодження штепсельного з'єднання, кабелю (шнура) або його захисної оболонки;

- нечітка робота вимикача;
- дим або запах, характерний для палаючої ізоляції;
- підвищений шум і вібрація.

Оператор повинен вимкнути електрообладнання вимикачем у разі раптової зупинки (наприклад, через втрату напруги в мережі тощо).

Після закінчення роботи з електрообладнання працівник зобов'язаний:

- відключити електрообладнання, місцеве освітлення та вентиляцію від електроживлення;
- привести в порядок робоче місце;
- інформувати безпосереднього керівника або іншу уповноважену посадову особу роботодавця про всі проблеми, що виникли під час роботи, та вжиті заходи щодо їх усунення.

При відчутті дії електричного струму при контакті з електрообладнанням необхідно негайно вимкнути його і повідомити про це свого безпосереднього керівника або іншу уповноважену посадову особу роботодавця, відповідального за утримання електрообладнання в справному стані.

При виникненні аварійної ситуації працівник повинен:

- негайно вимкнути джерело, яке спричинило надзвичайну ситуацію.
- припинити всі роботи, не пов'язані з ліквідацією аварії;
- вживати заходів щодо запобігання розвитку надзвичайної ситуації та впливу травмуючих факторів на інших осіб;
- забезпечувати виведення людей з небезпечної зони, якщо є небезпека для їх здоров'я та життя;
- повідомити про інцидент безпосереднього керівника або іншу уповноважену посадову особу.

У разі настання нещасного випадку на виробництві працівник зобов'язаний повідомити про нещасний випадок безпосереднього керівника або іншу уповноважену посадову особу роботодавця, сприяти у вжитті заходів щодо надання необхідної допомоги потерпілим і доставити їх в організацію охорони здоров'я.

РОЗДІЛ 5

ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Визначення ефективності інформаційних систем можливо декількома способами. Економічний ефект від впровадження засобів автоматизації може бути тільки непрямим, так як засоби автоматизації не є прямим джерелом доходу, а є або допоміжним засобом організації прибутку, або допомагають мінімізувати витрати.

Існує два базових способи оцінки економічного ефекту від використання програми: простий і складний (більш трудомісткий метод, але більш точний). Простий метод - це невелике спрощення складного методу. Наприклад, якщо матеріальні витрати не змінюються після реалізації програми, їх можна виключити з розрахунку, тим самим спростивши його. Як правило, повна оцінка за складним алгоритмом проводиться кваліфікованими фахівцями за результатами обстеження бізнес-процесів компанії. Але якщо потрібно швидко і приблизно оцінити ефективність впровадження інструменту автоматизації, то можна підставити в представлені формули кошторис витрат. Звичайно, якщо використовувати кошториси витрат, а не їх фактичні значення, економічний ефект не буде розрахований точно, але тим не менше дозволить оцінити рентабельність і необхідність автоматизації.

Основний економічний ефект від впровадження засобів автоматизації полягає в поліпшенні економіко-економічних показників діяльності підприємства, перш за все за рахунок підвищення ефективності управління і зниження трудовитрат на реалізацію процесу управління, тобто зниження витрат на управління.

Зниження трудовитрат на підприємстві можливо за рахунок автоматизації роботи з документами, зниження витрат на пошук інформації, тощо. Критерієм ефективності створення та впровадження нових засобів автоматизації є очікуваний економічний ефект. Він визначається за формулою:

$$E = E_p - E_n * K_p,$$

де:

- E_p - річна економія;
- E_n - нормативний коефіцієнт ($E_n = 0.15$);
- K_p - капітальні витрати на розробку і реалізацію, включаючи первісну вартість програми.

Річна економія E_p складається з економії експлуатаційних витрат і економії за рахунок підвищення продуктивності користувачів. Таким чином, отримуємо:

$$E_p = (P_1 - P_2) + \Delta P_p,$$

де:

- P_1 і P_2 - відповідно, операційні витрати до і після реалізації розроблюваної програми;
- ΔP_p - економія від підвищення продуктивності додаткових користувачів.

Якщо оцінювати економічний ефект з урахуванням всіх деталей, то капітальні витрати на проектування і реалізацію розраховуються з урахуванням тривалості робіт на даному етапі.

Щоб розрахувати витрати на етапі проектування, необхідно визначити тривалість кожної роботи, починаючи з підготовки технічного завдання і закінчуючи оформленням документів.

Тривалість робіт визначають або за нормативами (використовуються спеціальні таблиці), або розраховують на підставі експертних оцінок за формулою:

$$T_0 = (3 * T_{\min} + 2 * T_{\max}) / 5$$

де:

- T_0 - очікувана тривалість роботи;
- T_{\min} и T_{\max} – відповідно, найкоротший і найтриваліший термін роботи на думку експерта.

Якщо проектування і впровадження засобу автоматизації повністю здійснюється сторонньою організацією, то може бути використана спрощена

схема розрахунку, тобто в якості капітальних витрат на проектування і впровадження можуть бути прийняті суми, сплачені сторонній організації, включаючи первісну вартість засобу автоматизації.

Як приклад розглянемо невелику агрофірму, яка впроваджує систему підтримки прийняття рішень. Припустимо, вартість системи становить 100000 грн. Вартість послуг сторонньої організації з її реалізації становить 50000 грн.

В результаті капітальні витрати на реалізацію складуть:

$$K = 100000 + 50000 = 150000 \text{ грн.}$$

Розрахуємо витрати на утримання штату, виходячи з умови, що заробітна плата працівника становить 25000 грн.

$$Z = 1 * 25000 * (1 + 34\% / 100) = 33500 \text{ грн.}$$

У нашому прикладі для простоти розглянемо накладні та інші витрати до і після реалізації програми як незмінні. Таким чином, річна економія дорівнюватиме економії, пов'язаній зі збільшенням продуктивності користувача.

Економія, пов'язана з підвищенням продуктивності користувачів:

$$P = 33500 * 9 = 301500 \text{ грн.}$$

В результаті отримуємо наступну очікувану економічну ефективність:

$$E = 301500 - 150000 = 151500 \text{ грн.}$$

Навіть при приблизному розрахунку економічна ефективність від впровадження програмного забезпечення виявилася значною за рахунок підвищення продуктивності працівника.

За результатами розрахунку економічної ефективності проектування і впровадження інструменту автоматизації можна відразу сказати, що це вигідно. Хоча вигода непряма, зазвичай вона помітна в середньостроковій і довгостроковій перспективі. Впровадження інструментів автоматизації може призвести до коригувань самого бізнес-процесу, оскільки завдання виконуються швидше. У робочий час співробітники можуть обробляти великі обсяги інформації, які можуть бути використані або для зниження витрат на персонал, або для швидкого розвитку бізнесу, при цьому кількість співробітників, зайнятих в обробці інформації, залишається незмінною.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання кваліфікаційної роботи була розроблена інформаційна система підтримки прийняття рішень із планування парку комбайнів з урахуванням планів агрофірми. Виконані такі задачі:

1. Проведений аналіз предметної області та класифікована інформаційна система, що розробляється, як система підтримки прийняття рішень. Розглянуті підходи до побудови таких систем, визначені задачі роботи.

2. Розглянуті методи вирішення задачі та обрано метод аналізу ієрархій. На прикладах розповсюджених комбайнів обґрунтовані критерії оцінки альтернатив. Зроблений вибір мови програмування (Python), середовища розробки (Google Colab) та СУБД (MySQL).

3. Розроблена структурна схема інформаційної системи. Створена структурна ієрархічна модель вибору комбайнів. Проведена реалізація моделі обраним інструментарієм та перевірена її робота на навчальних даних. Описана організація обміну даними між програмною частиною та БД. Запропоновані окремі елементи графічного інтерфейсу користувача.

4. Розглянуті питання охорони праці та визначення економічної ефективності отриманих рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цифровізація в агросфері. URL: https://ndipzir.org.ua/wp-content/uploads/2020/12/Tezy_18_09_2020_20.pdf
2. Вплив цифровізації сільськогосподарських підприємств на розвиток сільських територій. URL: <https://echas.vnu.edu.ua/index.php/echas/article/download/682/557/>
3. Питання цифровізації сільського господарства в Україні. URL: <http://www.tsatu.edu.ua/tstt/wp-content/uploads/sites/6/manita-2020.pdf>
4. Цифровізація: можливості для фермерських господарств. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2022/12/112.pdf>
5. Вплив цифрових технологій на аграрне виробництво. URL: http://www.econ.vernadskyjournals.in.ua/journals/2019/30_69_6/30_69_6_2/8.pdf
6. Система підтримки рішень. URL: https://uk.wikipedia.org/wiki/Система_підтримки_рішень
7. Системи і методи підтримки прийняття рішень. URL: https://ela.kpi.ua/bitstream/123456789/48418/1/Systemy_i_metody_pidtrymky_pryiniattia_rishen.pdf
8. Системи підтримки прийняття рішень. URL: <http://kist.ntu.edu.ua/textPhD/sppr.pdf>
9. Конспект лекцій з навчальної дисципліни "Системи підтримки прийняття рішень". URL: <https://financial.lnu.edu.ua/wp-content/uploads/2015/10/konspekt-lektsiy-SPPR.pdf>
10. Етапи прийняття рішень. URL: <https://buklib.net/books/23875/>
11. Прийняття управлінських рішень: основи теорії та процес. URL: <https://osvita.ua/vnz/reports/management/15386/>
12. Сучасні аспекти прийняття рішень з управління бізнес-процесами підприємства. URL: <http://www.economy.nauka.com.ua/?op=1&z=4816>
13. Процес прийняття рішень. URL: <https://library.if.ua/book/3/411.html>
14. Ухвалення рішень. URL: https://uk.wikipedia.org/wiki/Ухвалення_рішень
15. Теорія рішень. URL: https://uk.wikipedia.org/wiki/Теорія_рішень

16. Моделі та методи прийняття рішень. URL: <http://csc.univ.kiev.ua/en/library/books/voloshyn-20.pdf>
17. Моделі й методи прийняття рішень. URL: <https://ir.nmu.org.ua/handle/123456789/153995>
18. Класифікація методів прийняття управлінських рішень. URL: <http://n-visnik.oneu.edu.ua/collections/2019/269/pdf/161-177.pdf>
19. Прийняття рішень: теорія та практика. URL: https://ns2000.com.ua/wp-content/uploads/2019/07/Pryuniattia_rishen-.pdf
20. Методи прийняття управлінських рішень. URL: <https://dspace.nuft.edu.ua/server/api/core/bitstreams/bcefa2f5-7bda-4a3c-93d5-1bcb81557a01/content>
21. Метод аналізу ієрархій. URL: https://uk.wikipedia.org/wiki/Метод_аналізу_ієрархій
22. Метод аналізу ієрархій. URL: <https://dss.tg.ck.ua/ahp-help>
23. Застосування методу аналізу ієрархій при виборі проекту в поліграфії. URL: <http://pvs.uad.lviv.ua/static/media/1-75/7.pdf>
24. Зернозбиральний комбайн. URL: https://uk.wikipedia.org/wiki/Зернозбиральний_комбайн
25. Комбайн Зернозбиральний. URL: <https://agro.ria.com/tag-kombajn/subtag-zernozbiralne/>
26. Зернові комбайни в Україні. URL: <https://prom.ua/ua/Zernovye-kombajny.html>
27. Вибір правильної мови програмування: тенденції й тренди. URL: <https://proit.org.ua/vibir-pravilnoyi-movi-proghramuvannia-tiendientsiyi-i-triendi/>
28. Який напрям обрати: рейтинг мов програмування. URL: <https://java.lviv.ua/yakuj-napryam-obraty-rejtyng-mov-programuvannya>
29. Вітаємо в Colaboratory. URL: <https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk>
30. База даних MySQL. URL: <https://promoter.net.ua/articles/baza-danix-mysql.html>