

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи керування**  
**інфраструктурою ЦОД”**

КБПЗ - 2024

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Щельник О.В.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Дреєв О.М.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 123 "Комп'ютерна інженерія"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Щельнику Олександр Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи керування інфраструктурою ЦОД

2. Керівник роботи Дреєв Олександр Миколайович, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи керування інфраструктурою ЦОД

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування ІТ-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Щельник О.В. Дослідження та програмна реалізація системи керування інфраструктурою ЦОД. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування інфраструктурою ЦОД.

Метою розробки є дослідження та програмна реалізація системи керування інфраструктурою ЦОД.

Об'єктом дослідження є процес керування інфраструктурою ЦОД.

Предметом дослідження є методи керування інфраструктурою ЦОД.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи керування інфраструктурою ЦОД.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

**Ключові слова:** комп'ютерна інженерія, керування інфраструктурою ЦОД

## ABSTRACT

**Shchelnik O.V. Research and software implementation of the data center infrastructure management system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the data center infrastructure management system.

The purpose of the development is the research and software implementation of the data center infrastructure management system.

The object of research is the process of data center infrastructure management.

The subject of the research is methods of managing data center infrastructure.

Research methods are based on methods of cloud technologies, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the data center infrastructure management system.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

**Keywords:** computer engineering, data center infrastructure management

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання .....	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	22
3.1 Опис функціонування системи .....	22
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми .....	37
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	58
6 НАУКОВА НОВИЗНА .....	64

						ВКРМ-123.24.0049.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата		Літ.	Аркуш	Аркушів
Розроб.	Щельник О.В.				Дослідження та програмна реалізація системи керування інфраструктурою ЦОД	М	1	92
Перев.	Дресв О.М.					ЦНТУ КІ-23М		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	65
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	65
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	66
7.3	Вибір методу оцінки вартості ПЗ .....	68
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	69
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	69
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	71
7.7	Визначення ключових факторів успіху конкретного проєкту.....	73
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	74
8.1	Вступ.....	74
8.2	Аналіз умов праці .....	75
8.3	Техніка безпеки та протипожежна профілактика .....	77
8.4	Розрахункова частина .....	80
8.5	Висновки до розділу.....	83
9	ОСНОВНІ ВИСНОВКИ.....	84
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	86

КБПЗ-2024

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>



## ВСТУП

**Актуальність теми.** Системи керування інфраструктурою ЦОД дозволяють власникам ЦОД одержати повне подання про те, що відбувається на об'єкті.

Таке різноманітне господарство, яким є сучасний обчислювальний центр, або, у сучасній термінології, центр обробки даних, вимагає однакового керування. Однак численні підсистеми, з яких складається його інфраструктура, традиційно контролювалися за допомогою окремих спеціалізованих систем керування. Системи керування інфраструктурою ЦОД покликані якщо не замінити, то об'єднати їх всі, щоб власник ЦОД міг мати повне подання про те, що відбувається на об'єкті.

В останні роки ринок систем керування інфраструктурою ЦОД (Data Center Infrastructure Management, DCIM) бурхливо розвивається, а найближчі перспективи виглядають ще більш райдужними. По оцінках Navigant Research, в 2023 році його обсяг склав 663 млн доларів (при цьому враховувалися продажі програмного забезпечення й послуг). Відповідно до недавно опублікованого дослідження MarketsandMarkets, за період з 2023 по 2029 рік ринок DCIM виросте з 307 млн до 3,14 млрд доларів. Таким чином, його середньорічний ріст складе 47,33%. На думку аналітиків цього агентства, основним драйвером росту є потреба в забезпеченні високої доступності й стійкого розвитку центрів обробки даних.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи керування інфраструктурою ЦОД.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем керування інфраструктурою ЦОД.
- Дослідження системи керування інфраструктурою ЦОД.

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Програмна реалізація системи керування інфраструктурою ЦОД.

*Об'єктом дослідження* є процес керування інфраструктурою ЦОД.

*Предметом дослідження* є методи керування інфраструктурою ЦОД.

*Методи дослідження* базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод керування інфраструктурою ЦОД.

– Розроблено вітчизняний продукт керування інфраструктурою ЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування інфраструктурою ЦОД.

**Достовірність наукових результатів** підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічній конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування інфраструктурою ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Разом з тим системи DCIM поки не займають у ЦОД того місця, якого вони, безсумнівно, заслуговують. Навіть до 2027 року, відзначає Gartner, вони будуть застосовуватися лише в 60% великих ЦОД у США (площею понад 300 м<sup>2</sup>). На тобто безліч причин, але, мабуть, головна з них – відсутність єдиноначальності в багатьох ЦОД і взаємодії між різними підрозділами, через що найчастіше кожна з підсистем працює нормально, а в цілому функціонування ЦОД далеко від оптимального.

Що ж дає використання DCIM? Дані, надавані DCIM, дозволяють оптимізувати живлення, охолодження й фізичний простір у машинному залі, що, у свою чергу, дає можливість відстрочити капітальні вкладення в облаштуваність нових площ. Наявний у багатьох систем інструментарій дозволяє моделювати сценарії «що, якщо», тобто зрозуміти, як різні дії, наприклад розміщення додаткових стійок або встаткування, вплинуть на ситуацію в ЦОД. Відповідно до розрахунків Gartner, за рахунок однієї тільки економії енергії впровадження DCIM окупається за три роки.

Разом з тим жодна із представлених на ринку систем поки не має ту функціональність, який повинна володіти DCIM, та й саме питання про її обов'язкові компоненти залишається відкритим. До того ж повномасштабна реалізація цієї системи потрібно далеко не завжди. Просуваються під парасолькою DCIM продукти ведуть свій родовід від самих різних рішень. Відповідно, вони можуть бути сильні в одних областях (облік і інвентаризація, моніторинг енергопостачання, контроль СКС) і слабкіше в інші. Все це, поряд з відсутністю безпосереднього бенефіціара, утрудняє впровадження DCIM і ускладнює вибір.

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 1.2 Область застосування

Щоб домогтися значної економії енергії, зовсім не обов'язково впроваджувати повномасштабну дорогу систему DCIM, для цієї мети підійде й система контролю мікроклімату ЦОД, наприклад Synapsense. Як свідчить досвід реалізації проектів цієї американської компанії, впроваджене рішення може окупитися за дуже короткий час, іноді буквально за місяць.

Synapsense призначена для постійного контролю кліматичних параметрів ЦОД у режимі реального часу. Її основу становлять бездротові датчики температури, вологості й тиску, які підключаються до шлюзів за технологією 802.15. Хоча система працює в тому же частотному діапазоні 2,4 ГГц, що й традиційний Wi-Fi, завдяки частотно-часовому поділу каналів вона не буде створювати перешкод для розгорнутої в ЦОД мережі WLAN.

У невеликих ЦОД (від 100 до 1000 м<sup>2</sup>) рекомендується використовувати два шлюзи для резервування. Оскільки мережа має ніздрювату топологію, у випадку виходу з ладу обох шлюзів датчики будуть по ланцюжку передавати інформацію один іншому, і в остаточному підсумку вона потрапить у програмне забезпечення, що є ядром даної системи. Якщо датчик «бачить», що канал зайнятий Wi-Fi, він регулярно посилає запити, очікуючи вільного вікна, і у випадку його появи передає інформацію на базовий шлюз. При відсутності такої можливості дані переміщуються на сусідній датчик, що акумулює їх і передасть у той момент, коли радіоканал звільниться. Ця технологія робить систему відказостійкою і перешкодозахищеною. Для кожної стійки рекомендується придбати комплект із семи датчиків, які встановлюються на фронтальну й тильну частини стійки, угорі, у середині й унизу, а опорний датчик температури розташовують під фальшполом. На кондиціонери датчики температури ставляться на вході й виході повітряного потоку. Кожне із цих пристроїв вимірює вологість. Систему можна розширити за рахунок датчиків контролю електроживлення (датчики Холу). На підставі отриманих даних Synapsense

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

розраховує PUE і безліч інших показників, які допоможуть оптимізувати електроспоживання ЦОД, а також виявляти ще що тільки зароджуються проблеми – наприклад, коли при перемиканні кондиціонерів на резервні потужності термодинамічні процеси переходять із квазістаціонарних у перехідні. Температуру й вологість, та й інші параметри, виміряти неважко. Основу Synapsense становить програмне забезпечення, що цю інформацію обробляє, відображаючи її в реальному часі на картах температури, тиску й вологості. Відомості, що зберігаються в базі даних, у будь-який момент можна відтворити для відновлення подій, які відбувалися й місяць, і рік назад. Так, при виникненні якихось проблем, можна подивитися, звідки почався перегрів, – якщо «винуватий» кондиціонер, на плані відразу будуть видні червоні області й перепади тиску в цих місцях. У порівнянні з аналогами, що використовуються в обчислювальних центрах, система моніторингу Synapsense легко встановлюється у вже існуючому або знову споруджуваному ЦОД, оскільки її бездротові базові станції не вимагають зовнішніх ліній зв'язку або зовнішнього живлення (вони здатні працювати від батарейок до шести років при частоті опитування раз у п'ять мінут). Гнучкість налаштування інтерфейсу керуючого ПЗ дозволить надати кожному користувачеві тільки ті дані, які відповідають його рівню доступу. Візуалізація даних, що збираються системою Synapsense, дозволяє оперативно відслідковувати зміни температури, вологості або тиску на побудовані для декількох рівнів картах з колірною індикацією, а сигнал про перевищення встановлених порогів значення миттєво надійде на пульт чергової зміни.

Як затверджується, система може працювати не тільки із програмними, але й з апаратними протоколами (SNMP, Modbus, BUCnet). Це дозволяє розширити її практично до повнофункціональної системи DCIM.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування інфраструктурою ЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Огляд автоматизованої системи диспетчеризації й керування (АСДК) у застосуванні до сучасних центрів обробки даних: архітектура рішення, можливості, переваги й особливості експлуатації.

Сучасний мир усе більше залежить від інформаційних систем. Не секрет, що для успіху в бізнесі необхідні високоефективні ІТ-рішення, які, з одного боку, повною мірою задовольняли б потреби бізнесу, а з іншого боку – не ставали б для компаній тяжким вантажем у вигляді витрат, що збільшуються, на ІТ і їхню підтримку. Сучасні центри обробки даних (ЦОД) – це економічно виправдані рішення, що консолідують ІТ-ресурси організації й здатні значно скоротити загальні витрати на ІТ за рахунок впровадження централізованої моделі обчислень. Однак постійне ускладнення ІТ-інфраструктури, збільшення енергоспоживання й тепловиділення в ЦОД накладають на роботу обслуговуючих інженерних підсистем ряд додаткових вимог: дуже висока надійність, керованість, безпека, адаптивність до змін бізнесу.

Надійності подібних систем і попередженню майбутніх проблем сьогодні приділяється дуже велика увага. Цілодобовий моніторинг, комплексний аналіз параметрів устаткування, попередження відмов і мінімальний час реакції – це найважливіші вимоги до диспетчерських служб, що контролюють інженерні підсистеми ЦОД, а робота персоналу в подібних службах стає усе більше відповідальною. Варто відзначити, що для повсякденного контролю інженерних підсистем потрібні фахівці в різних областях, таких, як електрика, вентиляція й кондиціонування, обслуговування різного спеціального устаткування.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Автоматизована система диспетчеризації й керування (АСДК) являє собою цілісну платформу для керування всіма інженерними підсистемами й створюється як багаторівнева автоматична система, що забезпечує контроль стану й керування технологічним устаткуванням ЦОД з виводом даних на екрани автоматизованих робочих місць операторів. АСДК веде безперервний моніторинг інженерних систем з реєстрацією основних параметрів і забезпечує контроль і керування інженерним комплексом з єдиного диспетчерського центра.

Організація диспетчерського центра на основі рішення АСДК дозволяє впровадити нові стандарти якості в керування експлуатаційним устаткуванням, що забезпечує, підвищити експлуатаційну готовність ЦОД, знизити поточні витрати на керування інженерними системами, забезпечити документування й протоколювання збоїв, створити базу для оперативного усунення аварійних ситуацій.

### **Архітектура рішення**

Сучасна АСДК має трирівневу архітектуру. Нижній рівень утворюють периферійні пристрої й інженерне встаткування, що формують первинні дані. Другий рівень – контролери, що приймають і обробляють інформацію, і мережа передачі даних. Верхній рівень – це ПЗ, що надає засобу візуалізації, архівації, публікації даних, що надходять. На робочі місця диспетчерів (АРМ) надходить структурована консолідована інформація в потрібному форматі. Аналітичний модуль постійно відслідковує робочі параметри систем на предмет відхилення від норми й здатний автоматично запускати процедури відповідно до закладених інструкцій, наприклад, подати сигнал тривоги або запустити аварійний дизель-генератор. Важливе завдання аналітичного модуля – завчасні попередження про прийдешні відмови.

Зібрані дані можна:

- передати операторам і представити їх у виді, що читається легко;
- зберегти в базі даних;
- проаналізувати й представити у вигляді статистичних звітів;

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– використовувати як керуючий сигнал при реакції на певні події для запуску систем в автоматичному режимі.

До складу рішення може входити система відеоспостереження, одночасно із сигналом тривоги вивідну картинку з аварійною підсистемою на монітор оператора. Як правило, у системі передбачений Web-інтерфейс, крім того, її можна інтегрувати із системами моніторингу IT-інфраструктури ЦОД.

При використанні в ЦОД комплексних систем керування, наприклад IBM Tivoli або HP OpenView, адміністратори одержують контроль над інформаційними бізнес-сервісами й пов'язаними з ними програмними й апаратними ресурсами ЦОД. АСДК може бути інтегрована з подібними рішеннями, і тоді інженерні підсистеми будуть мати безпосередній зв'язок із системами більше високого рівня, що підвищить експлуатаційну готовність ЦОД.

#### **Реєстрація й обробка подій**

Інженерні системи ЦОД складаються з безлічі взаємопов'язаного встаткування, тому при настанні якої-небудь тривожної події буває важко визначити, де конкретно виникла проблема. Для приклада візьмемо проблему в контурі живлення, між розподільним щитом і активним мережним устаткуванням (рисунок 2.1). Система локалізує проблему, визначає рівень можливих наслідків і відображає інформацію про конкретну систему у вікні тривоги. Екранна форма зі схемою системи показує відносини між взаємозалежним устаткуванням і можливими наслідками неполадок в окремих компонентах.

АСДК централізовано фіксує подія в базі даних і сповіщає диспетчера про виникнення проблеми й необхідності її дозволу. Далі система визначає рівень серйозності події й привласнює події певний пріоритет. Пріоритет необхідний, щоб підвищити ефективність реакції персоналу на подію. Наприклад, якщо сигналізація, що спрацювала, говорить про необхідність заміни фільтра системи кондиціонування повітря, оператор повинен розуміти, у які строки й з яким пріоритетом дозволити сформовану ситуацію.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

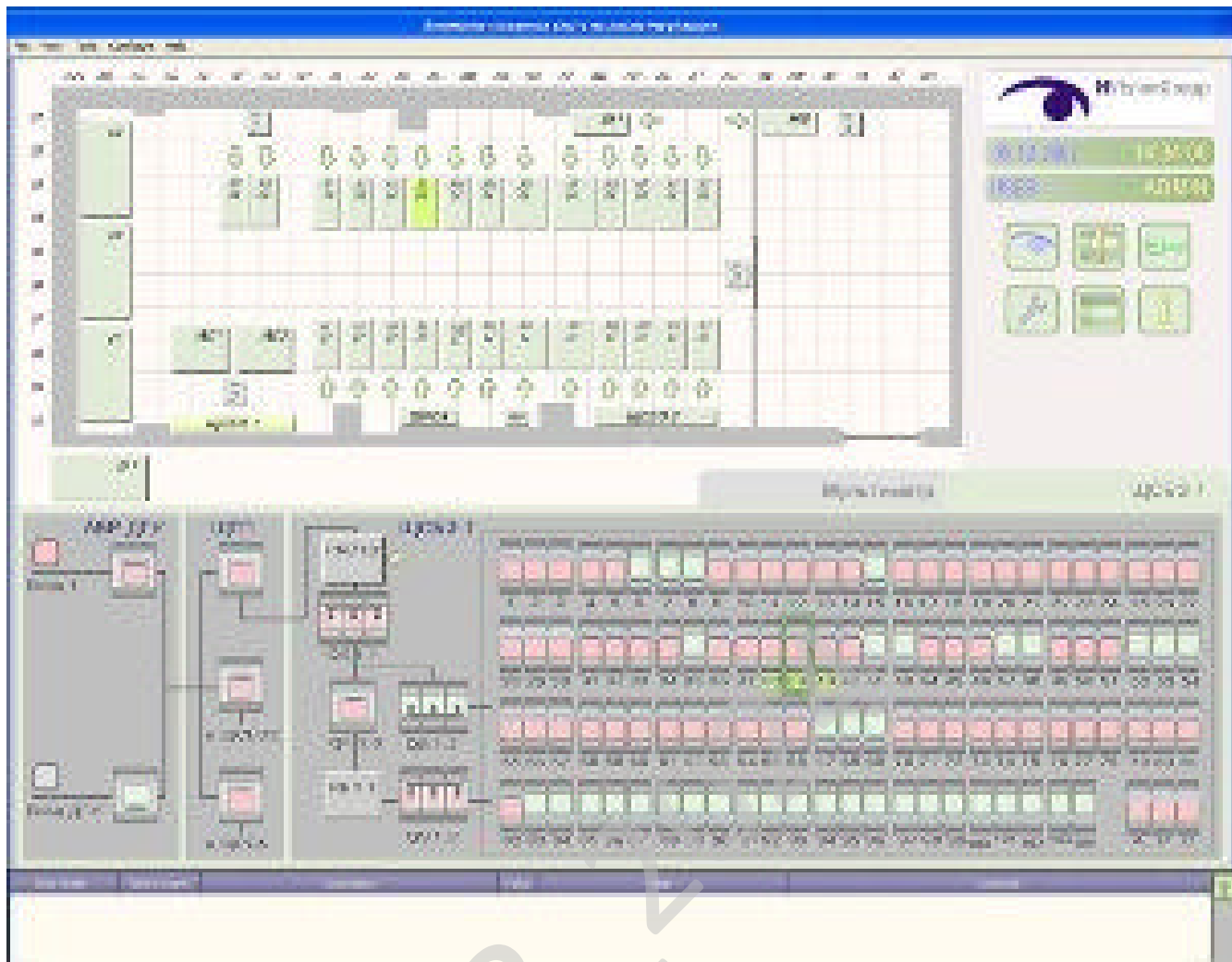


Рисунок 2.1 – Щит безперебійного живлення. Зв'язок устаткування й автоматичних вимикачів

Система виводить повідомлення про вихід параметрів, що відслідковуються, за встановлені раніше межі, а також повідомлення про критичний час наробітку експлуатованого інженерного встаткування. Наприклад, це можуть бути дані про стан акумуляторних батарей, температури й вологості в стійках. Інформація представляється в доступному для адміністраторів і диспетчерів і виді, що читається легко.

Одна з найважливіших функцій АСДК – своєчасне оповіщення про виниклі ситуації всіх відповідальних осіб, що обслуговують підсистеми ЦОД. Система має функції оперативного оповіщення диспетчерів, адміністраторів і

керівних осіб об'єкта по електронній пошті або за допомогою повідомлень SMS, а також інтегрується з іншими доступними способами сигналізації відповідно до встановленого регламенту.

### **Експлуатаційна готовність і безпека**

Алгоритми й регламенти відповідних дій на подію, що відбулася, програмуються в АСДК, і від правильності настроювання подібних регламентів прямо залежить експлуатаційна готовність. Варто визначити й конкретних осіб, що виконують та або інша дія (керування встаткуванням, підтвердження тривожного повідомлення й т.д.). Для розмежування відповідальності за обслуговування різних систем АСДК має можливість управляти повноваженнями диспетчерів. Автоматизована система надає функції розмежування доступу різних груп диспетчерів із прив'язкою до певних завдань або контрольованих систем. У протилежному випадку, якщо тривожні сигнали й повідомлення доставляються абстрактному «диспетчерові» без прив'язки до конкретної людини, складно визначити відповідального за реакцію на ту або іншу позаштатну ситуацію.

Нижче ми коротко охарактеризуємо основні контрольовані підсистеми й параметри моніторингу АСДК.

### **Моніторинг і фіксація критичних змін параметрів навколишнього середовища ЦОД**

Відмова встаткування може бути наслідком не тільки занадто високої температури, але й швидкої її зміни. Система відслідковує температуру й вологість на рівні стійок з устаткуванням і сповіщає диспетчера про те, що зафіксовані потенційно небезпечні значення температури й вологості. Хронологічні дані й параметри навколишнього середовища можуть виводитися у вигляді графіків, що читаються легко (рисунок 2.2).

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

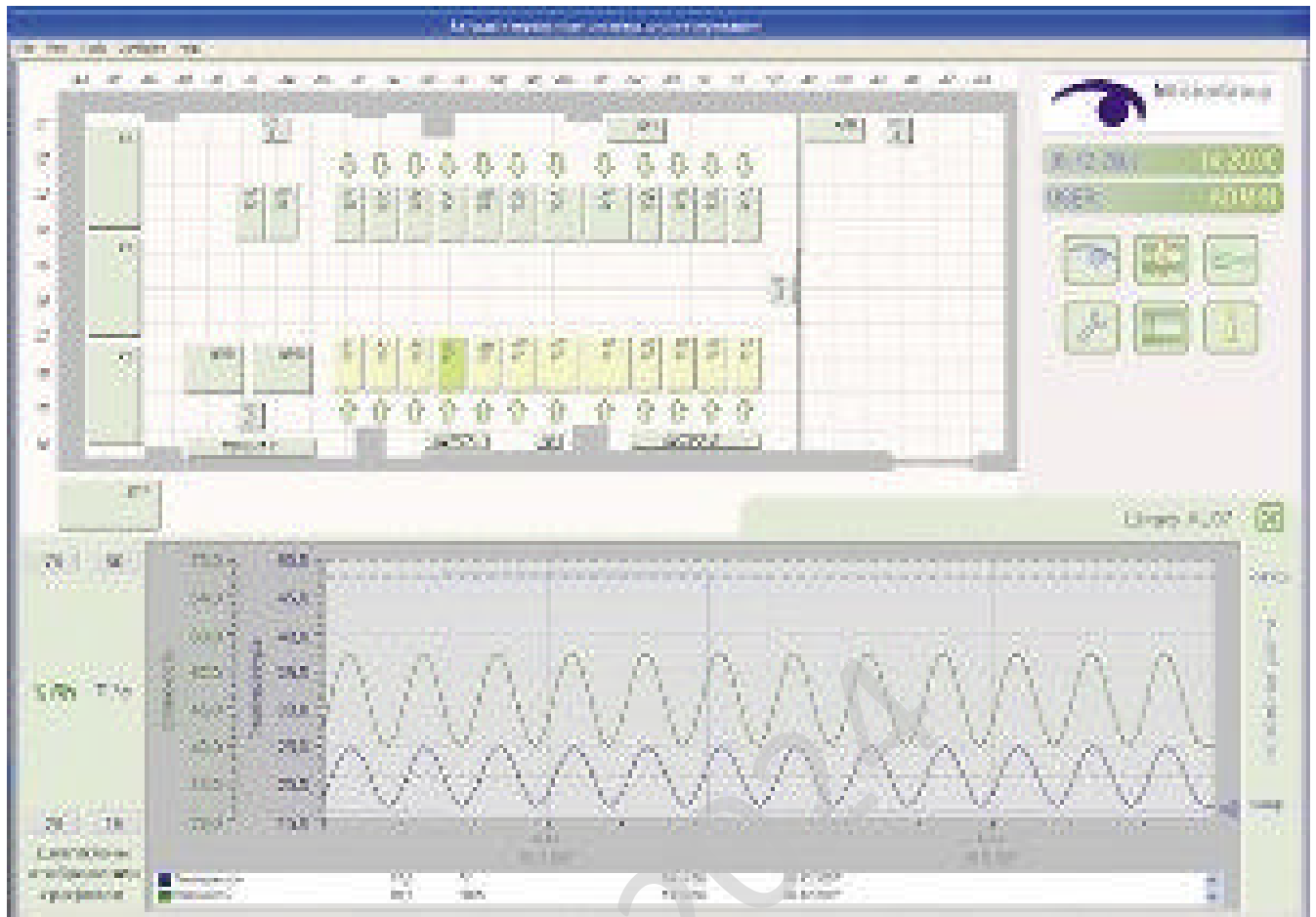


Рисунок 2.2 – Графік зміни температури й вологості в часі для обраної шафи

### Моніторинг і фіксація змін у споживанні електроживлення активним устаткуванням

У міру появи в ЦОД нового обладнання потреби в електроживленні й охолодженні можуть перевершити наявні ресурси, результатом чого стануть перебої в роботі. Зокрема, інженерні системи ЦОД вимагають додаткової уваги в міру старіння батарей ІБЖ. Рівень старіння батарей залежить від інтенсивності їхнього використання й температури. АСДК відслідковує споживання струму для кожної галузі ланцюга або стійки й сповіщає відповідальних осіб про ситуації, що загрожують виникненням перевантаження. Вона також інформує їх про всіх ІБЖ, у яких час автономної роботи виявляється менше мінімуму або в яких перевищується граничне значення навантаження.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

### **Відстеження електроживлення встаткування**

Несправність устаткування або ліній подачі електроживлення, а також некоректні дії обслуговуючого персоналу можуть привести до знеструмлення встаткування. АСДК оперативно сповіщає диспетчера про наявність або відсутність живлячої напруги на споживачах.

### **Відстеження якісних і кількісних характеристик електроживлення**

Неякісне електроживлення приводить до виходу з ладу або передчасному спрацюванню устаткування. Зміна навантаження на систему електроживлення (включення/вимикання кліматичного встаткування, додавання встаткування ЦОД і т.д.) може викликати ситуацію, коли система безперебійного електроживлення не в змозі забезпечити резервування. АСДК надає обслуговуючому персоналу централізовану інформацію про якість електроживлення й розподіл навантаження по ЦОД у режимі реального часу, а також зберігає цю інформацію в базі даних для подальшого з'ясування причин відмови встаткування.

### **Визначення надійності електроживлення**

Оперативне відстеження стану встаткування, що забезпечує гарантоване й безперебійне електроживлення (ІБЖ, ДГУ), неможливо без централізованого збору й відображення інформації із цих пристроїв. АСДК надає диспетчерові централізовану інформацію про стан устаткування, що забезпечує.

### **Забезпечення температурного режиму роботи встаткування**

Кліматичний режим ЦОД може порушуватися через неправильні режими роботи кліматичного встаткування. За нерівномірного розподілу встаткування в ЦОД іноді виникають зони локального перегріву, що може зажадати змін у режимах роботи кліматичного встаткування. Обслуговуючий персонал не завжди зауважує тимчасовий вихід температури або вологості за межі норми, що приведе до проблем при визначенні причин збоїв у роботі активного встаткування. Крім того, кліматичний режим ЦОД може порушуватися через неправильні режими роботи або аварій на кліматичному встаткуванні. АСДК відслідковує температуру й вологість у телекомунікаційних стійках (рисунок 2.3) і сповіщає

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

диспетчера про те, що вони досягли потенційно небезпечних значень, а також зберігає цю інформацію в БД і видає її в зручному для наступного аналізу виді. Система надає диспетчерові інтерфейс для зміни режимів роботи кліматичного встаткування й оперативно сповіщає відповідальних про збої в його роботі (рисунок 2.4).



Рисунок 2.3 – Контроль температурно-вологісного режиму в телекомунікаційних шафах

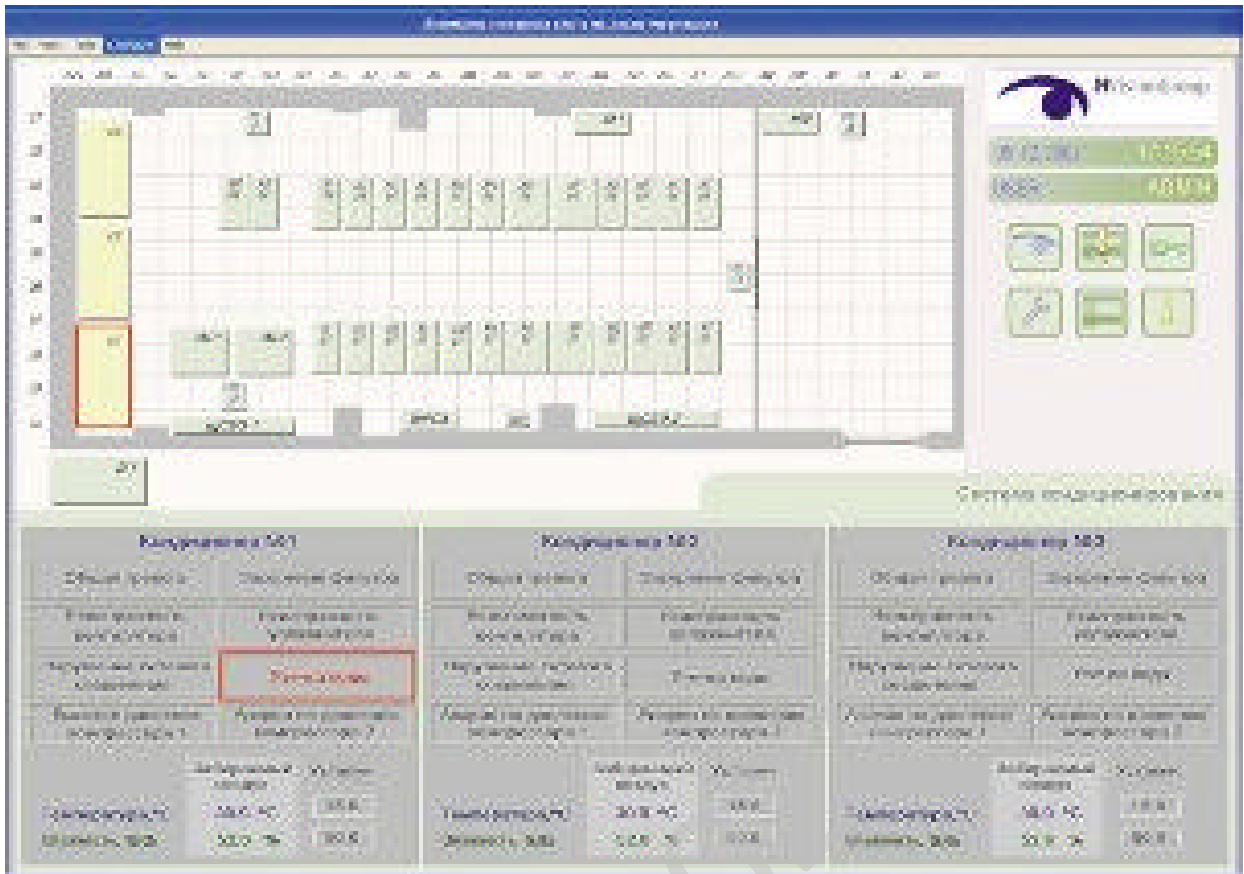


Рисунок 2.4 – Система кондиціонування: сигналізація про витік рідини

На АСДК також покладені функції мінімізації наслідків пожежі в ЦОД. При виникненні пожежі несвоєчасне оповіщення персоналу, а також робота кондиціонерів і непогодженість роботи інших підсистем у ЦОД може ускладнити роботу системи пожежогасіння й знизити її ефективність. АСДК сповіщає диспетчера про спрацьовування пожежної сигналізації й станції пожежогасіння, а також має можливість автоматично відключити кондиціонери й вентиляцію. Після спрацьовування системи пожежогасіння необхідно визначати якість повітря в приміщеннях і виводити цю інформацію на АРМ диспетчера.

Визначення й відстеження показників готовності ЦОД – складне й неоднозначне завдання. АСДК виступає тут як засіб інтеграції всіх інженерних і технологічних підсистем ЦОД у цілісну й керовану систему. Аналітична частина АСДК надає інструментарій для визначення причин простоїв і планування рівня надмірності інженерних систем.

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – мова програмування, що активно розвивається, нові версії (з додаванням/зміною мовних властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхня роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18



Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування інфраструктурою ЦОД.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ\_2024

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Усього лише п'ять років тому ринок систем керування інфраструктурою дата-центра (DCIM) тріпотів від захвату: першим адептам була надана можливість скористатися функціями моделювання й перегляду у високому дозволі поетажних планів і стійок, розташованих усередині дата центра.

На першому етапі розвитку ринку DCIM це було воістину вражаючим досягненням: створити точну модель дата-центра за допомогою одного лише зображення, яким можна маніпулювати на екрані монітора й виводити в різних формах залежно від цілей документування. Більш ніж десяток розроблювачів перших DCIM видавали на-гора одну демOVERсію за іншою, при цьому головним критерієм якості в рекламних проспектах вказувався ступінь реалістичності зображення стійок.

Кінцеві користувачі оцінили результати візуалізації, надані ранніми версіями DCIM, але проте, вони чекали більшого: нова технологія повинна була допомагати в рішенні комерційних завдань. У деяких випадках можливість легко моделювати й документувати свою ІТ-структуру була сама по собі достатньою, і рішення купувався винятково як засіб візуалізації, але ринок чекав, коли ж виявить себе комерційна цінність цієї концепції.

У другій, сучасній фазі розвитку технології DCIM завдання з'єднання DCIM-моделі із зовнішнім миром у реальному часі стала ключовий. Основна комерційна цінність будь-якого рішення DCIM полягає в тому, наскільки добре воно інтегрується із загальною системою керування дата-центром (яка включає керування будинком). Мова йде не тільки про джерела живлення й датчиках температури, але й про повний процес керування інформаційними сервісами й системами (ITSM). Чим більше інформації, тим якісніше прийняті на її основі

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

рішення. У світі, орієнтованому на створення рентабельних Каталогів Сервісів, можливість такого підключення стає єдиним способом розрахувати реальні витрати, пов'язані з наданням тих або інших послуг.

Націлена на зниження експлуатаційних витрат, система DCIM сьогодні – чи не краще рішення по оптимізації дата-центра і його поточних операцій. DCIM використовується для забезпечення й реалізації найкращих методів, а також стандартизації роботи підприємства й всіх його дата-центрів. DCIM створює найбільш експлуатаційно-придатні структури, де всі використовувані процеси створення й підтримки інформаційних систем можуть бути оптимізовані, захищені й відтворені незалежно від того, хто їх створив і де вони перебувають на будь-якому етапі розвитку підприємства.

Сьогодні можна спостерігати деякі ключові тенденції в комплексі DCIM. Тут я приведу десять із них.

#### **Два аспекти керування**

На сьогоднішній момент сформувалися дві основні групи пропозицій у сегменті ринку DCIM:

- керування інформаційними ресурсами в стійках протягом тривалих тимчасових проміжків;
- керування механічними, електричними елементами й водопровідною арматурами, або керування енергопостачанням і охолодженням об'єкта.

Більшість розроблювачів починають визнавати наявність обох груп і необхідність розробляти такі рішення, які стерли б границю між ними. Сьогодні кожний вендор, працюючи над рішенням, фокусується, як правило, на одному аспекті, у більшості випадків лише несміливо переборюючи границю між ними, щоб в обмеженому обсязі задовольнити потреби іншої сторони. Чи вдасться звести ці аспекти воєдино? Час покаже. Надія є, але поки створити реальне рішення для такої єдиної платформи DCIM не вдається жодному із гравців.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## **DCIM як інструмент стратегії**

У міру того як відношення до технології мінялося (від опції до предмета першої необхідності), більшість кінцевих користувачів зрозуміли, що DCIM повинна стати не просто ще одним тактичним інструментом, але й частиною стратегії керування.

Дата-центри протягом останнього років росли, поступово розширюючи свого функціонала з появою кожного нового інструмента. Сьогоднішній тренд – мислити стратегічно й знаходити більше продуктивні рішення для всіх зацікавлених осіб. Найбільш успішні проекти показали, що від впровадження DCIM починають вигравати широкі шари користувачів багатьох галузей, внаслідок чого відношення до DCIM міняється від «непогано б мати» до «необхідно мати». Деякі з нині діючих впроваджених DCIM-моделей використовуються сотнями клієнтів зі сфери інформаційних технологій, фінансів, інфраструктурних проектів, навіть управлінського сектора вищої ланки. DCIM проявляє себе щонайкраще, якщо вона розглядається як стратегічний інструмент, і її ефективність прямо може бути пов'язана з витратами на будівництво й зміст дати-центра.

### **Статус Системи Зберігання Даних**

У міру посилення взаємного проникнення цифрових технологій у дата-центрі споживачі мають потребу у всі меншій кількості інструментів. Стало тенденцією шукати більше зроблені рішення, у яких використовується принцип обміну даними. Інструменти, що вимагають дублювання даних, поступово відмирають. Інтеграція на рівні конвергентної інфраструктури (CI) має критичне значення. У найбільш успішних DCIM-інсталяціях інтеграція репозиторія DCIM з іншими системами настільки висока, що DCIM сама по собі стала розглядатися як «система записів», або «база даних» активів дати-центра.

Фактично DCIM стає самою точною моделлю дати-центра, у багатьох випадках навіть більше точної, ніж бухгалтерські реєстри.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## **Зміни в дата-центрі через реінжиніринг процесів**

Дата-центри завжди були площадками, на яких відбуваються зміни: там увесь час щось міняється. Бездоганно-прозорий дизайн дати-центра швидко перетворюється в хаотичне накопичення відновлень, які погано задокументовані й часто зовсім неефективні. Сьогоднішній тренд – перспективне планування всіх змін, які відбудуться в даті-центрі.

По ідеї, ефективність дата-центра не буде падати, якщо використовується стругаюча система керування змінами. Якщо зміни параметрів виробляються за допомогою системи керування змінами (яка є частиною DCIM), ця модель завжди буде залишатися точним відбиттям стану самого дата-центра й дозволить підтримувати його в оптимізованому стані. Таким чином, сьогодні тенденція полягає в тому, щоб шукати рішення комерційних питань шляхом аналізу DCIM-моделі, а не в процесі блукання між рядами стійок дата-центра в надії одержати відповіді лише на підставі спостереження за тим, що відбувається в серверному залі.

### **Ріст популярності SaaS**

Хоча впровадження DCIM спочатку розглядалося як рішення локальне, багато розроблювачів уже тоді почали пропонувати програмне забезпечення як сервіс (SaaS). Можливості DCIM як сервісу стають ідентичними локальному рішенню окремого розроблювача.

Причина цього тренда, тобто використання SaaS-версії DCIM, криється в тому, в тому, що пропозиція у вигляді сервісу SaaS можна віднести в статтю операційних витрат, а не капітальних витрат. Хоча процес прийняття ринком цих сервісів може бути однаковий, проте існує думка, що пропозиції SaaS більше відкриті й гнучкі. Важливо помітити, що в той час як DCIM традиційно відносять до сегмента сервісів, підхід до реалізації DCIM безпосередньо на площадці, або у вигляді SaaS-варіанта повинен бути однаковим, якщо ви хочете успішно впровадити цю систему.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

## **DCIM – більше не «річ у собі»**

У той час як раніше функціональність досягалася за рахунок окремо взятих можливостей, нині особливо важливої є інтеграція в системи керування IT-інфраструктурою (ITSM), включаючи керування змінами й керування конфігураційними базами даних (CMDBs). Сьогоднішня тенденція – побудувати більше високий рівень інтеграції на етапі вибору DCIM-провайдеру. Інтеграція в системи ITSM необхідна для того, щоб зробити DCIM стратегічною частиною інфраструктури керування дата-центром. Взагалі, підхід до впровадження DCIM без обліку подальшої інтеграції залишився далеко в минулому.

Перспективи використання DCIM у співвідношенні з іншими ключовими інструментами керування тепер повністю прогнозуються з першого дня розробки проекту.

### **Тісна інтеграція із засобами віртуалізації**

На сьогоднішній день уже більше половини серверів у світі віртуалізовані. Таким чином, має сенс тісно інтегрувати найбільш успішні DCIM-рішення з топовими галузевими віртуальними платформами.

Тенденція сьогодні – шукати рішення, що дозволяють одержати в реальному часі подання про статус віртуального сервера на кожному з фізичних пристроїв. DCIM повинна забезпечувати зображення контрольованого вузла і його операційних систем у режимі реального часу. Зараз прагнуть використовувати зв'язану картину віртуалізації для планування навантаження від операційної системи до сервера, і, в остаточному підсумку, до необхідних фізичних ресурсів (приміщення, енергопостачання, охолодження).

### **Ріст використання убудованих датчиків**

Завдання контролювати параметри середовища в дата-центрі існували завжди, але рішення були громіздкими й дорогими. Сьогодні виробники IT-Устаткування шукають рішення, що дозволяють вбудовувати апаратні датчики в кожний пристрій промислового класу, що випускається ними. Ці датчики можуть надавати звіт про енергоспоживання, про температуру усередині різних зон дата-

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

центра й про фізичну безпеку. Тенденція полягає в тому, щоб найбільше ефективно використовувати убудовані датчики й комбінувати їх з додатковими (звичайно бездротовими) для тих пристроїв, які не перебувають на площадці основного встаткування.

### **Точна оцінка активів конвергентної інфраструктури**

Серед корпоративних дата-центрів простежується тенденція до забезпечення процесів перевірки й балансування з метою увесь час зберігати впевненість в тому, що DCIM-модель точно відбиває ситуацію в серверному залі. Такий подвійний контроль приймає різні форми, від перевірки вручну з використанням електронних таблиць, до використання бездротового зчитувача штрихкоду й/або локальних технологій, заснованих на автоматизованому використанні радіочастотних міток (RFID). Можлива навіть комбінація інформації, отриманої візуальним способом, з даними від системи виявлення або від коннекторів конфігураційної бази CMDB.

У кожному разі, є тенденція до підвищення точності відображення моделей дата-центрів будь-якими можливими засобами, через ITSM-інтеграцію, інтеграцію із системою віртуалізації, а також за допомогою технологій, наявних на самій площадці.

### **Загалькорпоративне впровадження DCIM**

Рішення DCIM спочатку пропонувалися як засіб візуалізації десятка стійок. З ростом комерційної ефективності зі стратегічної точки зору масштаб впровадження DCIM значно розширився, аж до того, що деякі постачальники DCIM-рішень тепер обслуговують десятки тисяч стійок у десятках дата-центрів.

Не варто забувати, що в рамках DCIM-пропозиції всі об'єкти, включаючи ті що розташовано усередині іншого будинку, модульні, а також комерційні ЦОДи із сервісами колокейшен, розглядаються як площадки дати-центра. Іноді можна зустріти десятки площадок, які потрібно сукупно представити в єдиному DCIM-рішенні. Сьогоднішні рішення прагнуть охопити все компоненти, у будь-якому масштабі, незалежно від їхнього фізичного розташування.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

У цей час ми спостерігаємо розвиток технології DCIM, що відбуває в основному завдяки зрушенню від ідеї дата-центра як тактичного набору функцій до дата-центру – добре осмисленому стратегічному інструменту. Постачальники DCIM розуміють, що їм потрібно фокусуватися на високоефективній послугі в декількох підсистемах, а не заспокоюватися на досягненнях у сфері підвищення деталізації картинки. Кінцеві користувачі тепер повинні формулювати свої індивідуальні вимоги до DCIM набагато більш докладно, ніж коли-або в минулому. Фактично, від IT-організацій тепер вимагають чітко роз'яснити состав портфеля сервісів, виділяючи кожний сервіс в окремий рядок каталогу. Система DCIM стає життєво важливим компонентом у створенні таких елементів каталогів сервісів, оскільки її завдання зрештою зводиться до керування витратами.

Отже, як же виглядає наступна фаза розвитку DCIM? Очевидно, це буде контроль і оркестровка. Це зрівноважить моделювання й зв'язність, удосконалені в перших двох фазах, і привнесе новий рівень автоматизації з використанням електронних систем керування, що замінюють участь людини при прийнятті рішень.

У більшості випадків пристрої автоматизації стають засобом виконання складних наборів правил, заснованих на більших обсягах метричної інформації. Незважаючи на те, що наступна фаза буде визрівати набагато довше внаслідок різномірності середовища провайдерів послуг дата-центрів, саме ця фаза DCIM дозволить знизити рівень витрат до бажаного.

### 3.2 Розробка структурної схеми

Як ми вже відзначали, представлені на ринку рішення DCIM мають самих різних корінь. У числі перших на перспективі цього ринку звернули увагу постачальники структурованих кабельних систем, оскільки в більшості з них уже були інтелектуальні напівавтоматизовані системи для керування кабельними

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

з'єднаннями. При величезній кількості фізичних з'єднань у ЦОД наявність актуального кабельного журналу є серйозною підмогою, якщо необхідно, наприклад, визначити, які перекомутації потрібні для підключення нових серверів. Традиційний підхід, коли системний адміністратор тримає в пам'яті всі зроблені комутації або в найкращому разі веде кабельний журнал в електронному виді, у сучасної віртуалізованому середовищу чреватий численними помилками. Уникнути їх допомагають зручні інструменти, які надаються системами керування кабельною інфраструктурою.

Спеціально розроблена програма, що враховує необхідну послідовність операцій на кожному етапі, допоможе значно спростити весь цей процес, у тому числі й при будівництві нового ЦОД. Компанія TE Connectivity пропонує відразу дві системи моніторингу кабельних з'єднань – AMPTRAC і Quareo, у яких використовуються різні принципи ідентифікації з'єднань. Якщо в AMPTRAC для детектування з'єднань застосовується традиційна технологія дев'ятого провідника, то в Quareo – технологія Connection Point Identification. В останньому випадку у вилку кожного коннектора вбудовується мікрочип, а в гнізді комутаційної панелі розташовуються додаткові ламелі, які при підключенні цієї вилки зчитують інформацію й реєструють її в ПЗ.

На чипи заводським способом заноситься наступна інформація про комутаційний шнур: категорія, довжина, колір, полярність і т.д. Аналогічні відомості втримуються й у комутаційній панелі. У системі не передбачається використання центральних сканерів – кожна панель і кожний крос мають свій IP-адресу, так що до них можна звернутися з Web-Браузера, навіть не використовуючи спеціальне ПЗ. І в качані, і в гнізді комутаційної панелі є лічильник підключень, що по здійсненні заданої кількості підключень подасть сигнал про необхідність проведення регламентних робіт.

Даний підхід дозволив мінімізувати тимчасові й трудові витрати. Опіраючись на схожі принципи, у даній роботі розробимо власне ПЗ Infrastructure Configuration Manager (ICM), що для всіх елементів бази даних (не

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

тільки пасивного й активного встаткування, але також кімнат, кабінетів, поверхів, будинків, лотків і т.д.) може генерувати QR-код.

Код може бути роздрукований і наклеєний на встаткування. Потім QR-коди зчитуються убудованою камерою мобільного пристрою за допомогою додатка ICM Mobile і розпізнаються, а інформація із сервера ICM по даному об'єкті або встаткуванню, отримана по каналах Wi-Fi, виводиться на екран. Таким чином, всі операції по адмініструванню можна відслідковувати, зміни в конфігурації з'єднань – документувати, а потрібна інформація, наприклад про поточний статус будь-якого кінцевого пристрою, оперативно виводиться на екран пристрою.

ICM може застосовуватися і як окреме програмне рішення, і разом із системами моніторингу кабельних з'єднань, які компанія TE Connectivity пропонує (у цьому випадку підвищується ступінь автоматизації операцій комутації й гарантується актуальність інформації в базі даних). За допомогою API воно може бути інтегроване у вищестоящі системи керування мережею (NMS) і інфраструктурою ЦОД (DCIM).

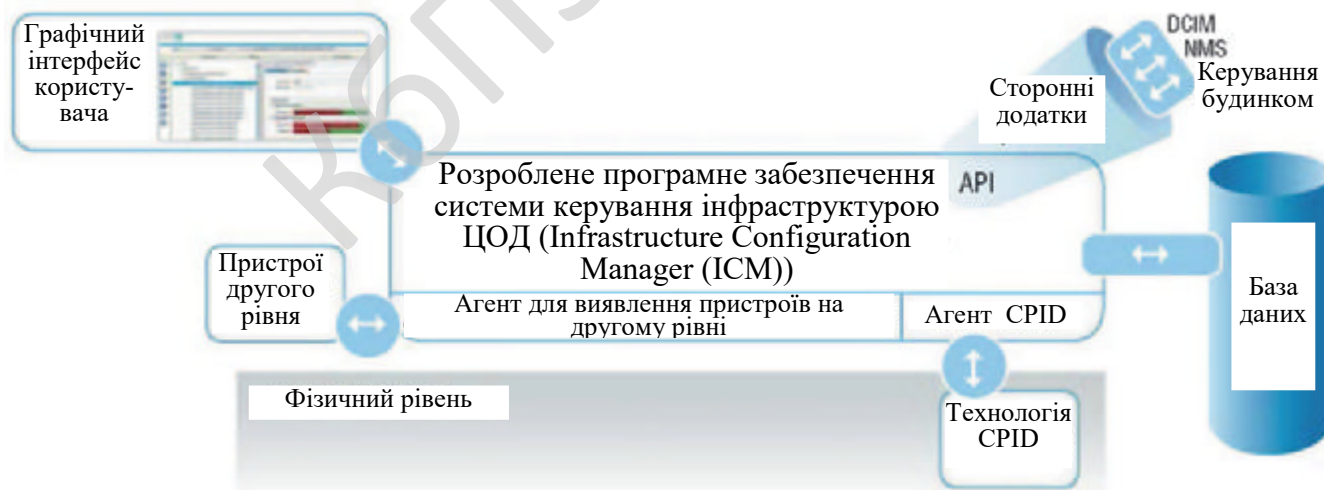


Рисунок 3.1 – Структурна схема системи

Ідентифікація точок з'єднання (Connection Point Identification – CPID) якраз забезпечується за рахунок використання мікрочіпів. Дані мікрочіпи містять інформацію про фізичні характеристики шнура, включаючи порядковий номер з'єднувача, тип, клас, довжину тощо, що відноситься до підключених комутаційних шнурів. При підключенні ця інформація зчитується і реєструється в базі даних ICM за допомогою вбудованого в панель сканера.

Ізраїльська компанія Ri Technologies першою запропонувала систему інтерактивного керування СКС ще в 2002 році. З тих пір вона постійно неї вдосконалювала, і в результаті з'явилася принципово нова система PV+. Розроблювачам удалося перебороти традиційний недолік CIU для СКС, через якого вони виявлялися надмірно дорогими в реалізації: уперше подібна система дозволяє обійтися без кросс-коннекта.

Через необхідність застосування кросс-коннекта доводилося піклуватися про виділення для нього місця в шафі й придбанні додаткових розподільних панелей, що, у свою чергу, вело до додаткових прямих і непрямих витрат. Ri розробила спеціальну вставку з мікрочіпом і діодом для установки в порти активного встаткування. За допомогою вставки ідентифікуються порти активного встаткування й, якщо потрібне виконання операцій підключення/кроссування, здійснюється їхнє індивідуальне підсвічування. Інтерконнект можна використовувати як для оптики, так і міді.

На основі рішення програмної оболонки PV+ був розроблений продукт CenterMind, що згодом компанія має намір розвинути в повноцінну систему DCIM. Поки ж, крім адміністрування кабельної підсистеми, CenterMind дозволяє контролювати ще дві апаратні підсистеми: P+ для моніторингу живлення за допомогою інтелектуальних панелей живлення (Power Distribution Unit, PDU) і G+ для контролю навколишнього середовища – температури, вологості, протікання і т.д.

Крім контролю пасивних з'єднань, CenterMind «уміє» автоматично виявляти встаткування, у тому числі детектувати підключення

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

авторизованих/неавторизованих пристроїв. Наприклад, можна задати, що до конкретної розетки дозволено підключати тільки пристрою з певними адресами й іменами. (По суті, у цій частині вона дублює можливості системи керування активним мережним устаткуванням.) Модуль автоматичного виявлення Automated Discovery дозволяє скласти точну візуальну карту центра обробки даних з усіма стійками, шафами, комутаторами, серверами й іншими пристроями. На карті відображається інформація про фізичне розташування пристроїв, з'єднаннях між ними, про IP- і MAC-адреси, тип і ім'я хоста.

Інструментарій для планування ємності, виділення ресурсів і керування життєвим циклом активів допомагає оптимальним образом розмістити нове встановлюване обладнання з обліком таких змінних, як місце, живлення, охолодження, доступні порти й т.п. Наприклад, при розміщенні сервера досить вибрати відповідну модель із великого списку підтримуваного встаткування (якщо потрібної моделі в списку ні, те її можна туди внести самостійно). Система сама врахує вагу пристрою, габарити, кількість необхідних інформаційних розеток, рівень енергоспоживання й запропонує оптимальне місце для його розміщення.

Через відсутність деяких компонентів CenterMind поки не можна назвати повноцінної DCIM – зокрема, вона не дозволяє контролювати систему кондиціонування. Однак за допомогою наявного комплекту SDK її можна об'єднати з іншими системами рівня систем керування будинком (Building Management System, BMS).

### **Повний облік і контроль**

Спроби впровадження платформи DCIM часом закінчуються невдачею через неможливість виявляти й урахувувати всі активи, конфігурації й дані про статус серверів, систем зберігання, мережного встаткування, інженерних і інших допоміжних систем, особливо у випадку успадкованих архітектур. Процес збору даних вручну надмірно витратний, забирає багато часу й чреватий помилкам. Однак без реальної картини поточного стану ЦОД неможливо побудувати

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

повноцінну модель центра обробки даних, а виходить, рішення по оптимізації процесів експлуатації й модернізації ЦОД можуть виявитися необґрунтованими й помилковими.

Різні підсистеми ЦОД, наприклад обчислювальна й телекомунікаційна, тісно взаємозалежні, і багато замовників хотіли б документувати їх в одному середовищі. Необхідні засоби їм надає програмний пакет FNT Command німецької компанії FNT Software. Компанія споконвічно створювала семирівневу (збіг із числом рівнів OSI випадково) модель даних, що охоплює й IT-інфраструктуру (IT-модули), і телекому-складові, і інженерні системи. При документуванні ЦОД можна вносити всі необхідні дані й атрибути самого будинку й інженерної інфраструктури, ураховувати наявні фізично активи (де коштують стійки й перебуває те або інше «важке» устаткування), указувати логічні з'єднання поверх існуючих фізичних і т.п. Якщо в обчислювальному середовищі або середовищі передачі даних є віртуалізовані системи, їхньої характеристики теж можна документувати із прив'язкою до фізичних активів, і т.д. – аж до рівня додатків і бізнес-сервісів.

Якщо інформація вже представлена в електронному виді, її можна імпортувати в систему. При наявності можливості підключатися до встаткування по SNMP або інших протоколах система автоматично збирає необхідну інформацію (можлива інтеграція з наявними засобами моніторингу). Із системою поставляється бібліотека встаткування на 60 тис. одиниць – телекомунікаційні пристрої, сервери, кабелі й багато чого іншого. При відсутності встаткування в базі даних воно буде додано в рамках технічної підтримки. Мобільний клієнт дозволяє сканувати штрих-коди й працювати з RFID-мітками.

Система містить потужний модуль планування. При підготовці змін завжди можна одержати інформацію про поточний статус системи, подивитися на план поверху або стійки й сформувати вбрання на роботу, що буде деталізований і з потрібної гранулярністю виданий виконавцям. Дуже часто ця функціональність використовується як база для розрахунків за роботи, зроблені

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

зовнішніми організаціями, або контролю їхньої якості й строків виконання. Крім того, система дозволяє спрогнозувати, до чого можуть привести заплановані зміни.

Найчастіше впровадженню подібних систем перешкоджає їхня дорожнеча. FNT Software пропонує привабливу модель ліцензування: незалежно від масштабу ЦОД ліцензування здійснюється по кількості користувачів, що мають право на перегляд і внесення змін (останні дорожче), – ні стійки, ні встаткування не враховуються. Систему FNT Command доцільно впроваджувати при наявності 100 і більше стійок.

У кожному разі порахувати ROI від впровадження такої системи непросто. На підприємстві Volkswagen Group, де рішення FNT використовується вже 10 років, IT-відділ обґрунтував його необхідність у такий спосіб. Керівництво вимагало забезпечити зниження операційних витрат на 5% у рік, а цього можна було домогтися тільки шляхом автоматизації типових дій співробітників, тому що звільняти їх або змушувати працювати позаурочно законодавство не дозволяє. Оскільки не можна поліпшити операційну діяльність, не задокументував споконвічний стан всієї IT і телекомунікаційної інфраструктури (із чимсь треба порівнювати), вивід про впровадження FNT напрашувався сам собою. У результаті поставлена перед IT-Фахівцями бізнес-завдання успішно виконується.

### **Обґрунтування інвестицій в DCIM**

Дотепер багато компаній не до кінця розуміють, які вигоди вони одержать від впровадження DCIM. Відповідно до опитування, проведеному серед 825 керівників американських компаній, відповідальних за прийняття рішень у сфері IT, 90% з них хотіли б мати можливість явного розрахунку й аналізу повернення вкладень для обґрунтування своїх інвестицій. Щоб допомогти їм підготувати таке формальне обґрунтування, Emerson Network Power (при участі Forrester Research) розробила калькулятор ROI.

Методологія, що використовує Forrester Research, називається Total Economy Impact (TEI). Вона дозволяє поряд з витратами й вигодами враховувати

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

стратегічні переваги й потенційні ризики. При її застосуванні передбачається більше широкий погляд на те, як впровадження рішення вплине на бізнес: що зміниться для користувача, наскільки підвищиться його ефективність, чи зручно йому буде вирішувати варті перед ним завдання (зокрема, чи скоротиться час установки сервера), чи прискориться вивід продуктів на ринок і т.д. Специфіка моделі складається в обліку потенційних ризиків і неминучої невизначеності: у рамках проведеного дослідження визначається, яка ймовірність того, що розрахована віддача виявиться недосягнутою, очікувані переваги не будуть отримані або для реалізації проекту знадобиться більше засобів, чим передбачено в затвердженому бюджеті. Таким чином, аналіз ризиків завершує аналіз ТЕІ.

Як здійснюється аналіз? У першу чергу проводиться всебічна оцінка (due diligence), тобто разом із замовником фахівці Emerson виявляють проблемні місця, потреби ЦОД і т.д. Після цього опитуються різні групи співробітників, на роботу яких може вплинути впровадження нового рішення. Велика кількість людей передають у ЦОД інформацію й витягають неї. Це можуть бути фахівці з експлуатації, що займаються організацією ЦОД, корпоративні співробітники, для яких ЦОД надає критичні бізнес-сервіси, менеджери 3-рівня, що намагаються витягти статистику про тенденції й т.д.

У кожній групі виникають свої питання щодо функціонування ЦОД: в одних стратегічні, в інших операційні. Залежно від виконуваної ролі їм потрібно зрозуміти, яку ємність буде потрібно додати в майбутньому, наскільки ефективно працює ЦОД, потрібно чи вносити зміни, як треба реагувати на ті або інші події й т.д. Інтерв'ю з ними проводяться з погляду бізнес-вигід. На підставі зібраних і узагальнених даних складається фінансова модель. При цьому як база для порівняння використовується еталонний приклад (benchmark), що був складений за результатами проекту для великого банку з декількома ЦОД.

Ми робимо аналіз ТЕІ, тому що ринок DCIM вимагає такого обґрунтування. Ми знаємо, як важко менеджерам ЦОД показати цінність DCIM для бізнесу. DCIM-калькулятор Emerson допоможе їм представити можливі вигоди. Оскільки ми збираємо дані, які надають нам замовники, довіра до

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

отриманих результатів дуже високо. Ми не оперуємо віртуальними цифрами, щоб зрозуміти, які будуть переваги.

### **DCIM на практиці**

У першу чергу переваги від впровадження DCIM здатні одержати великі центри обробки даних із сотнями стійок. Однак, як показує приклад Microsoft Technology Center (MTC), де використовується DCIM-система Schneider DC Expert, ця система може виявитися корисна й у великій серверній.

У центрі представлене різне партнерське встаткування, що часто обновляється. При цьому необхідно забезпечувати рівномірне завантаження стійок (дорога електроенергія повинна витратитися ефективно), а також документувати підключення пристроїв, оскільки схеми комутації постійно міняються. За інфраструктуру відповідає всього одна людина, тому система моніторингу інфраструктури автоматизована й ретельно продумана.

Наявність системи моніторингу дозволяє швидко ідентифікувати події й оперативно вживати заходів по усуненню несправності, які виконуються по більшій частині в автоматичному режимі (за винятком серйозних аварій, таких як протикання води, про що оперативно повідомляються інженерні служби будинку). Наприклад, вихід з ладу циркуляційного насоса може бути компенсований включенням резервного, а поломка чиллера або недостатній рівень холоду – включенням ще одного чиллера.

Для оперативної оцінки стану інфраструктури використовуються панелі моніторингу: відповідна графіка виводиться на монітори, установлені при вході в ЦОД, і замовники можуть бачити реальні дані. Наприклад, на підставі інформації про збільшення вологості можна зробити вивід про занадто активну роботу уволожувача й необхідності коректування його налаштувань. На Рисунку 6, наприклад, видно, що система повідомляє про відмову однієї з батарей.

Для моніторингу додатків використовуються дві системи – StructureWare Data Center Expert і System Center Operations Manager, які інтегруються за допомогою Management Pack. Міжрядні кондиціонери InRow і ІБЖ Symmetra становлять основу ЦОД, тому використання Expert для моніторингу інженерної

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

інфраструктури представляється логічним. Expert виводить звіти на панелі моніторингу й направляє повідомлення. При відсутності реакції на них (наприклад, температура продовжує рости) сервери починають автоматично відключатися, починаючи з менш важливих, – аж до припинення роботи всього ЦОД щоб уникнути більше важких наслідків. За прийняття таких мір відповідає System Center Operations Manager, він же здійснює моніторинг додатків.

Надалі планується підвищити ефективність використання електроенергії за рахунок перерозподілу віртуальних машин між шафами шляхом інтеграції StructureWare Data Center Expert з Microsoft System Center Virtual Machine Manager.

В останні кілька років системи DCIM удосконалювалися швидкими темпами, на ринку представлені рішення різного масштабу й призначення. Вони покликані надати адміністраторам повну інформацію про функціонування центра обробки даних і полегшити планування його подальшого розвитку. DCIM спрощує керування ЦОД і допомагає оптимізувати його інфраструктуру з метою підвищення ефективності використання енергії, устаткування й доступних площ. Однак на цьому шляху необхідно усунути ще цілий ряд перешкод.

### 3.3 Розробка функціональної схеми

Функціональна схема системи зображена на рисунку 3.2.

З рисунку видно, що розроблена система складається з наступних блоків:

- Моніторинг трафіку системи керування інфраструктурою ЦОД.
- Робота з файлами системи керування інфраструктурою ЦОД.
- Монітор з'єднань системи керування інфраструктурою ЦОД.
- Статистика подій системи керування інфраструктурою ЦОД.
- Робота з ресурсами мережі системи керування інфраструктурою ЦОД.
- Робота з сесіями системи керування інфраструктурою ЦОД.
- Функції для роботи з мережею системи керування інфраструктурою ЦОД.

ЦОД.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

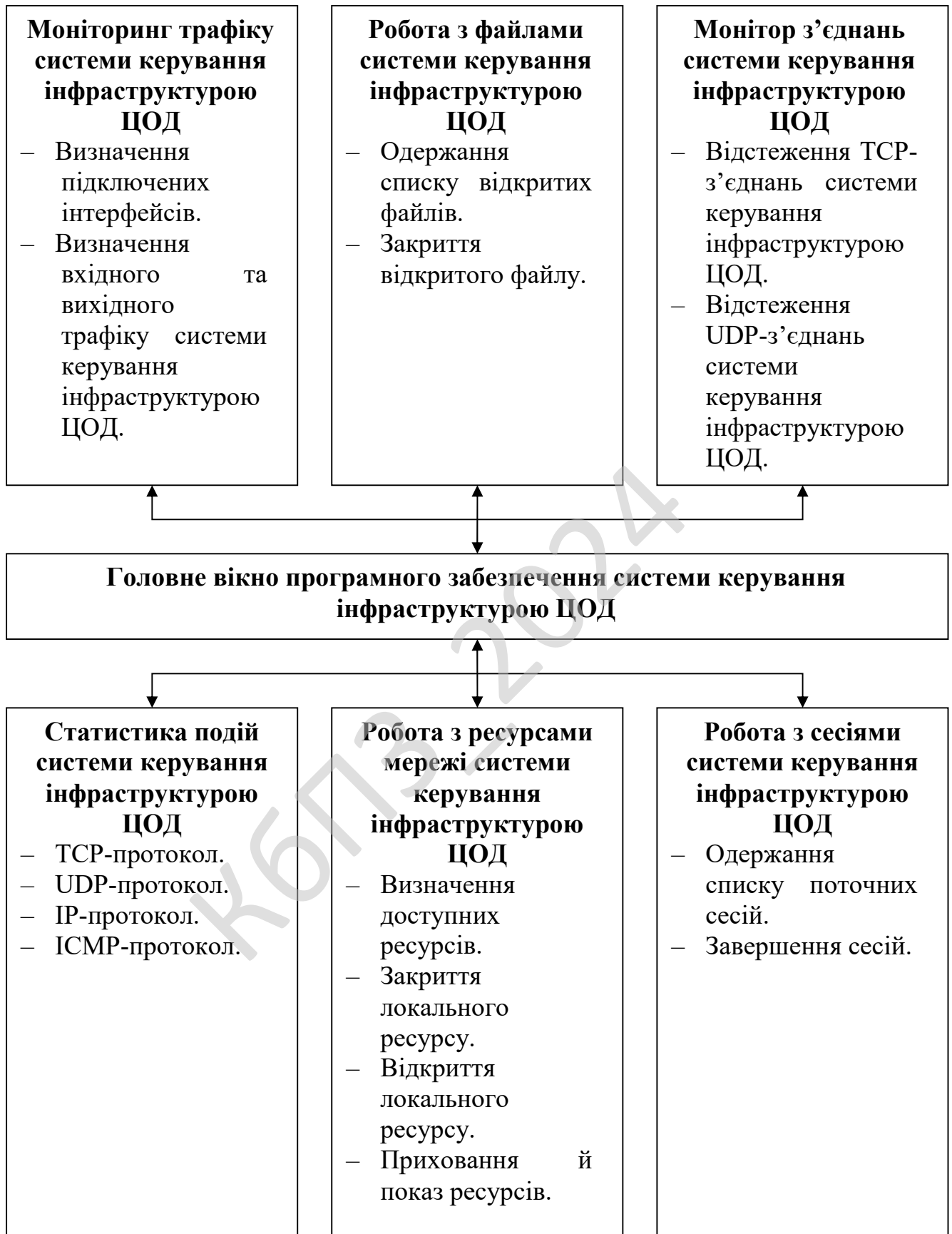


Рисунок 3.2 – Функціональна схема системи



– ICMP-протокол.

Статистика ведеться по цілому ряду параметрів. Наприклад для TSP-протоколу фіксується: Тип алгоритму повторної передачі, мінімальний тайм-аут, максимальний тайм-аут, максимальна кількість помилок з'єднання, активні з'єднання, пасивні з'єднання, невдалі спроби відкриття, скидання встановлених з'єднань системи керування інфраструктурою ЦОД, отримані сегменти, надіслані сегменти, повторно передані сегменти, помилки тощо.

Робота з ресурсами мережі системи керування інфраструктурою ЦОД включає в себе:

- Визначення доступних ресурсів.
- Закриття локального ресурсу.
- Відкриття локального ресурсу.
- Приховання й показ ресурсів.

Система відображає наявні у мережі системи керування інфраструктурою ЦОД ресурси у вигляді дерева. Пошук ресурсів можна здійснювати по заданим умовам: локальні чи глобальні ресурси; всі ресурси, тільки файли, чи тільки принтери, тощо.

Можна додавати до загальних ресурсів мережі системи керування інфраструктурою ЦОД свої власні, а також закривати їх потім.

Робота з сесіями системи керування інфраструктурою ЦОД включає в себе:

- Одержання списку поточних сесій.
- Завершення сесій.

Програма дозволяє переглянути список відкритих сесій, що включає в себе: назву сесії, користувача, що її розпочав, номер сесії, час роботи та час очікування.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>



Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ\_2024

					VKPM-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

## 4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю керування інфраструктурою ЦОД.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

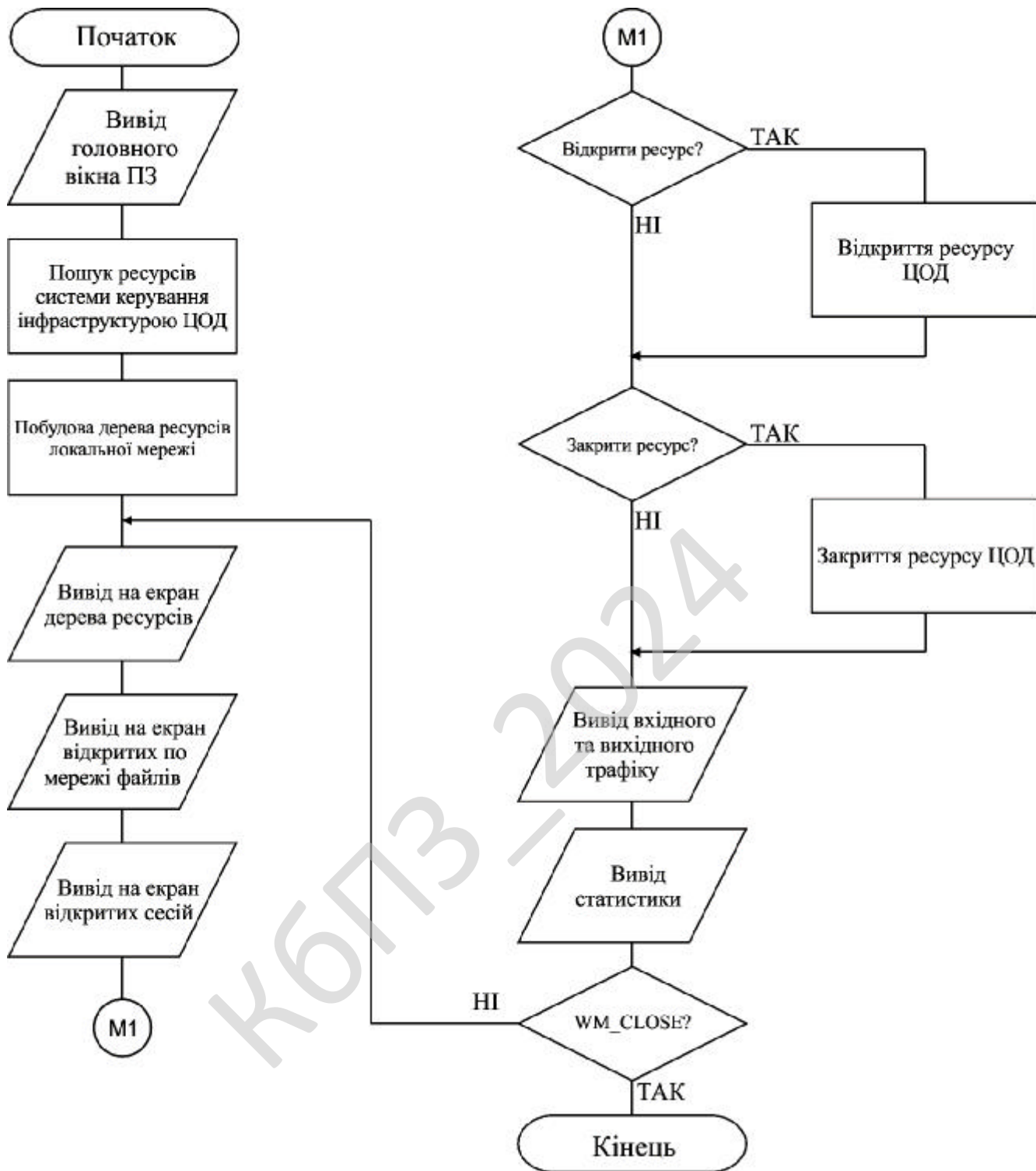


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

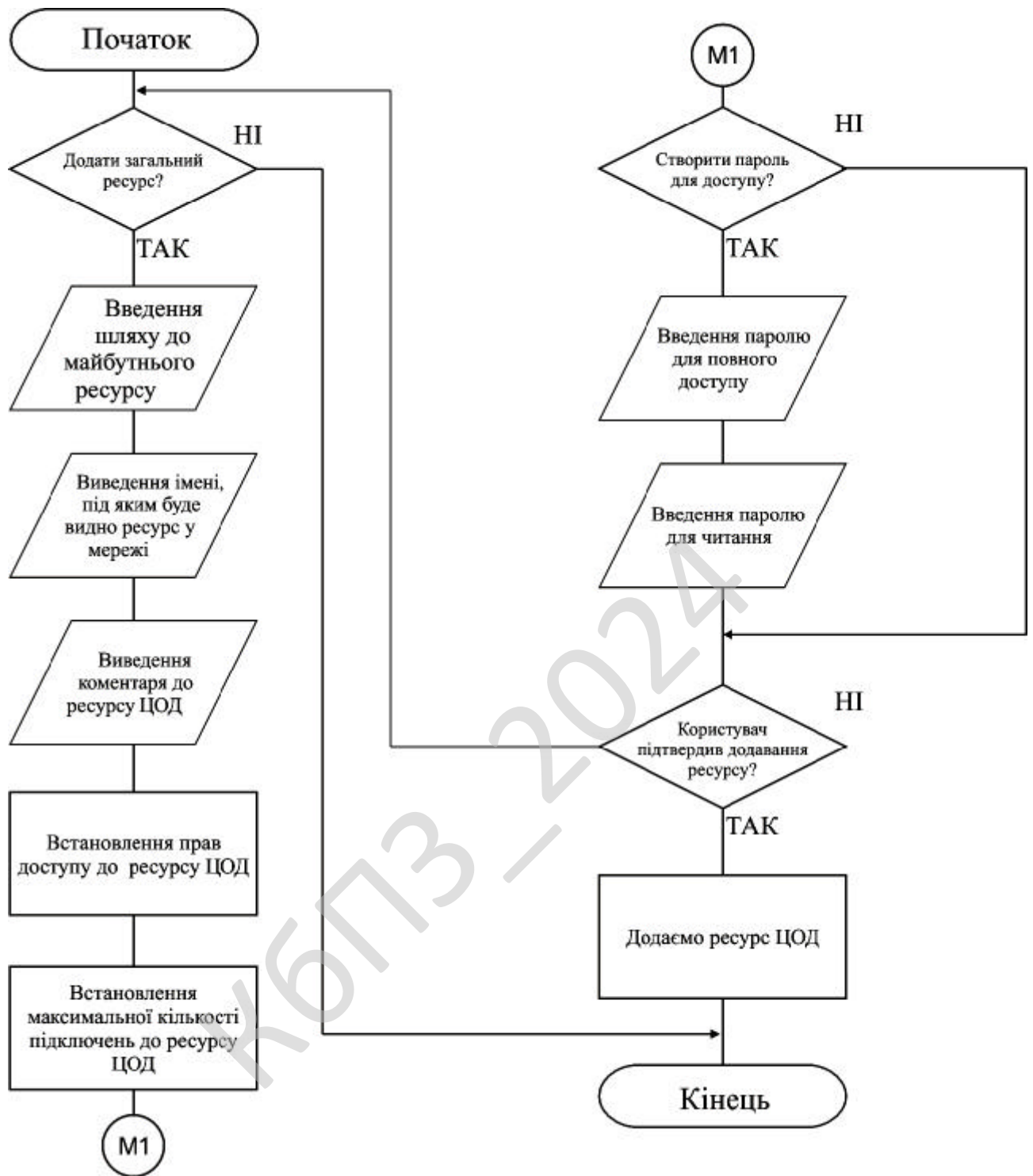


Рисунок 4.2 – Блок-схема роботи підпрограми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого

програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення.

UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

**Архітектура клієнт-сервер** є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою

відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47





розмітки XML (Extensible Markup Language).

**Управління вимогами** це процес запису, аналізу, трасування, пріоритетизації і узгодження вимог та контролю змін і доведення до їх зацікавлених сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покривають також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

### **Завдання управління вимогами**

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзанню меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, екпортуючи дані вимог.

### **Аналіз здійсненості**

На стадії аналізу здійсненості визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використанням інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?».

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягара користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив, що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроєктувати автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

Розробка та тестування. На стадії розробки і тестування, основна

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення прототипу і ітераційне тестування. Для програмного додатка користувацький інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

Була використана водоспадна (каскадна) модель життєвого циклу ПЗ (waterfall model) – послідовний метод розробки програмного забезпечення, названий так через діаграму схожу на водоспад.

Ця модель розробки запозичена з системної інженерії у виробництві та будівництві – областях, в яких зміни на пізніх етапах дуже дорогі, або неможливі. Наприклад, для створення складних інженерних конструкцій (споруд, літаків, мостів і т.п.). Зміни в проекті фундаменту будинку після того, як покладений дах коштують дуже дорого, тому перфекціонізм на початкових етапах проектування просто необхідний. Інженери, які починали займатись розробкою програмного забезпечення перейшовши з інших галузей, просто адаптували звичну модель, тому що на ранніх етапах розвитку комп'ютерної техніки не було методологій створених саме для програмування. Проте, схожі методології застосовуються для програмного забезпечення й далі, у випадках коли вимоги фіксовані, і вимагається висока якість та надійність, наприклад в системах для військових чи медичних потреб.

Перший формальний опис водоспадної моделі, після якої вона стала популярною був здійснений В. В. Ройсом у 1970. Попри те, що стаття містить переважно критику методу, на неї часто посилаються.

Переваги методу:

– Ніяких переробок.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

- Гарна специфікація перетікає в гарну документацію.
- Зрозуміла модель.
- Розробники можуть мати низьку кваліфікацію.

Недоліки:

- Необхідний перфекціонізм на кожному етапі.
- Важко вносити зміни (якщо взагалі можливо).
- Надлишкове проектування.
- Поділ розробників на "perfect" та "code monkeys".

Модифікації. Через те що цей метод погано підходить для розробки саме ПЗ, частіше використовують його модифікації.

Найвідоміша модифікація – Sashimi. Названа так через японську страву сашімі (суші нарізане і сервіроване так, що складені рядочком шматочки накладаються один на одного). В моделі розробки Сашімі фази життєвого циклу йдуть одна за одною, але при цьому перекриваються одна з одною в часі.

#### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ДСТУ 28147:2009, що є класичним алгоритмом симетричного шифрування на основі мережі Фейстеля (рисунок 4.3). Даний алгоритм шифрує інформацію блоками по 64 біта (такі алгоритми називаються "блоковими"). Зміст мережі Фейстеля полягає в тому, що блок шифруємої інформації розбивається на два або більше субблоків, частина яких обробляється за певним законом, після чого результат цієї обробки накладається (операцією побітового додавання за модулем 2) на необроблені субблоки. Потім субблоки міняються місцями, після чого обробляються знову й т.д. певне для кожного алгоритму число раз – раундів.

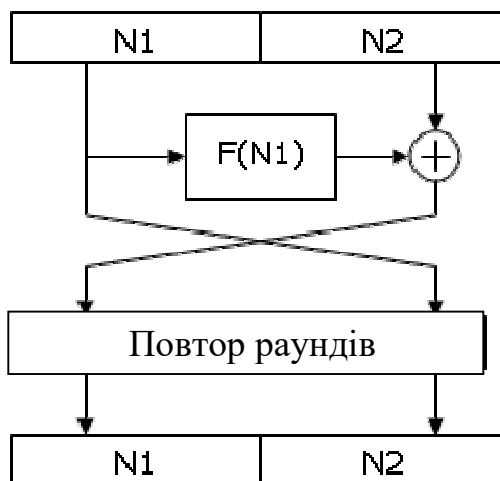


Рисунок 4.3 – Мережа Фейстеля

Основна відмінність алгоритмів симетричного шифрування друг від друга складається саме в різних функціях обробки субблоків. Дана функція часто називається "основним криптографічним перетворенням", оскільки саме вона несе основне навантаження при шифруванні інформації. Основне перетворення алгоритму ДСТУ 28147:2009 є досить простим, що забезпечує високу швидкість алгоритму; у ньому виконуються наступні операції (рисунок 4.4).



Рисунок 4.4 – Основне перетворення алгоритму ДСТ 28147:2009

1. Додавання субблоку з певним фрагментом ключа шифрування за модулем  $2^{32}$ .  $K_x$  – це 32-бітна частина ("підключ") 256-бітного ключа шифрування, якому можна представити як конкатенацію 8 підключів:

$$K = K_0K_1K_2K_3K_4K_5K_6 K_7.$$

Залежно від номера раунду й режиму роботи алгоритму (про їх – нижче), для даної операції вибирається один з підключів.

2. Таблична заміна. Для її виконання субблок розбивається на 8 4-бітних фрагментів, кожний з яких прогоняється через свою таблицю заміни. Таблиця заміни містить у певній послідовності значення від 0 до 15 (тобто всі варіанти значень 4-бітні фрагменти даних); на вхід таблиці подається блок даних, числове подання якого визначає номер вихідного значення. Наприклад, подається значення 5 на вхід наступної таблиці: "13 0 11 74 91 10 143 5 122 15 8 6". У результаті на виході виходить значення 9 (оскільки 0 замінюється на 13, 1 – на 0, 2 – на 11 і т.д.).

3. Побітове циклічне зрушення даних усередині субблока на 11 біт уліво.

КБПЗ\_2024

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи керування інфраструктурою ЦОД яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Вікно налаштувань.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

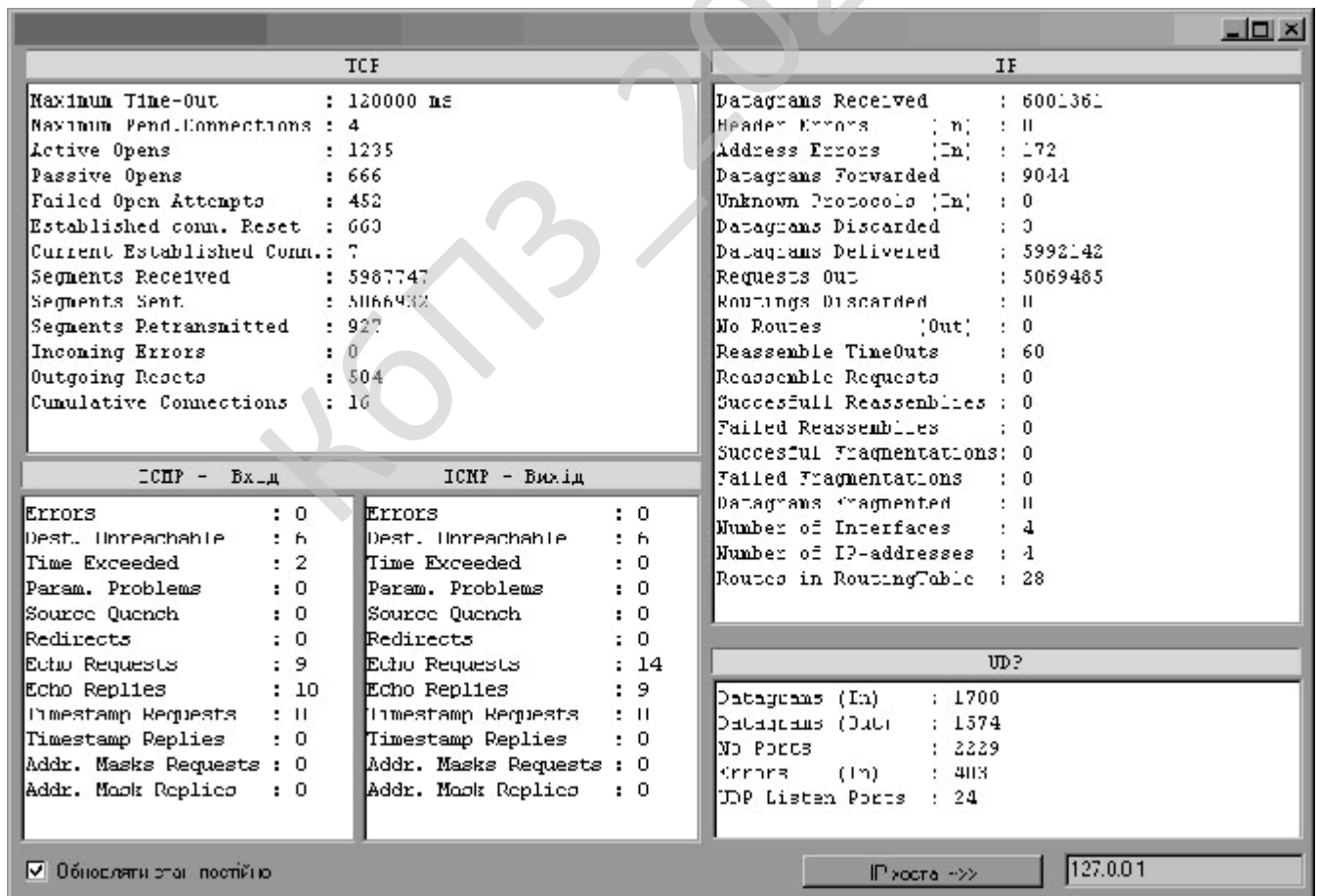


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

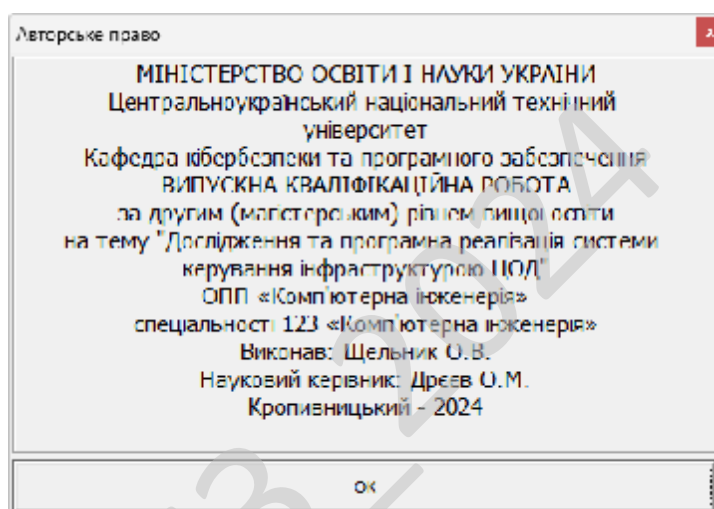


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частиною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом чорної скриньки

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>61</b>

- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєструватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування інфраструктурою ЦОД.

*Метою розробки є дослідження та програмна реалізація системи керування інфраструктурою ЦОД.*

*Об'єктом дослідження є процес керування інфраструктурою ЦОД.*

*Предметом дослідження є методи керування інфраструктурою ЦОД.*

*Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод керування інфраструктурою ЦОД.
- Розроблено вітчизняний продукт керування інфраструктурою ЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи керування інфраструктурою ЦОД можуть бути цікаві наступним категоріям користувачів і організацій (рисунок 7.1).

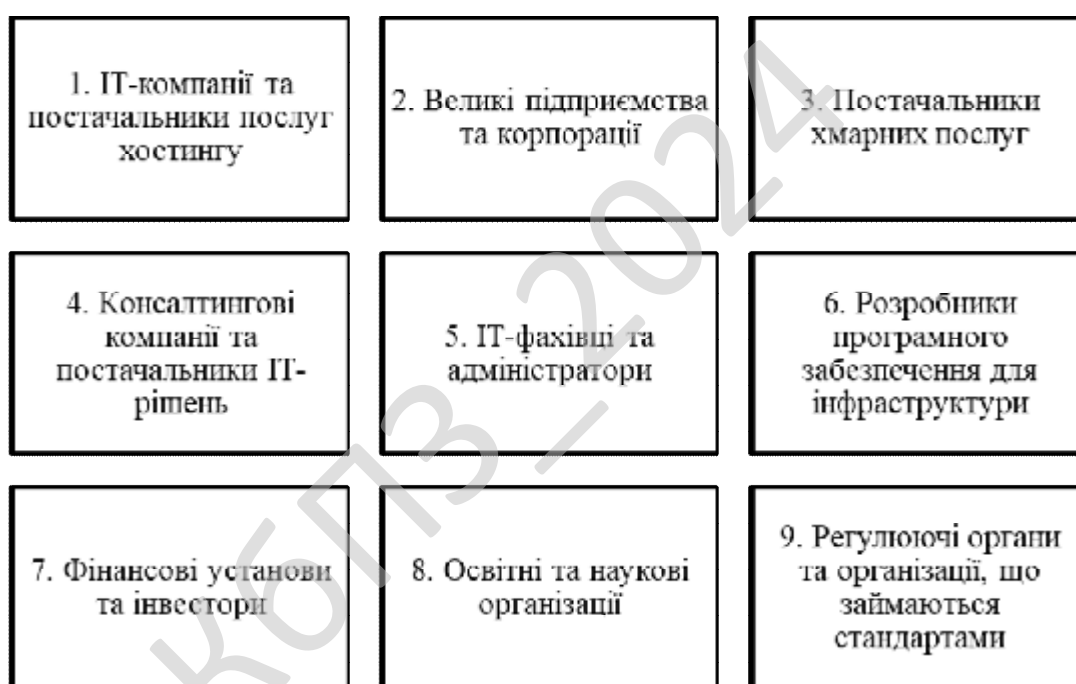


Рисунок 7.1 – Цільова аудиторія

Таким чином, результати дослідження і програмної реалізації системи керування інфраструктурою ЦОД можуть мати широкий спектр застосувань у різних сферах, зокрема для підвищення ефективності, зниження витрат і підвищення безпеки ІТ-інфраструктури.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості проекту програмної реалізації системи керування інфраструктурою ЦОД шляхом застосування методів експертних оцінок передбачає використання структурованого підходу для аналізу його переваг та недоліків.

Експертна оцінка передбачає визначення набору критеріїв, які важливі для оцінки привабливості системи керування інфраструктурою ЦОД. Це можуть бути такі критерії окреслені на рисунку 7.2.

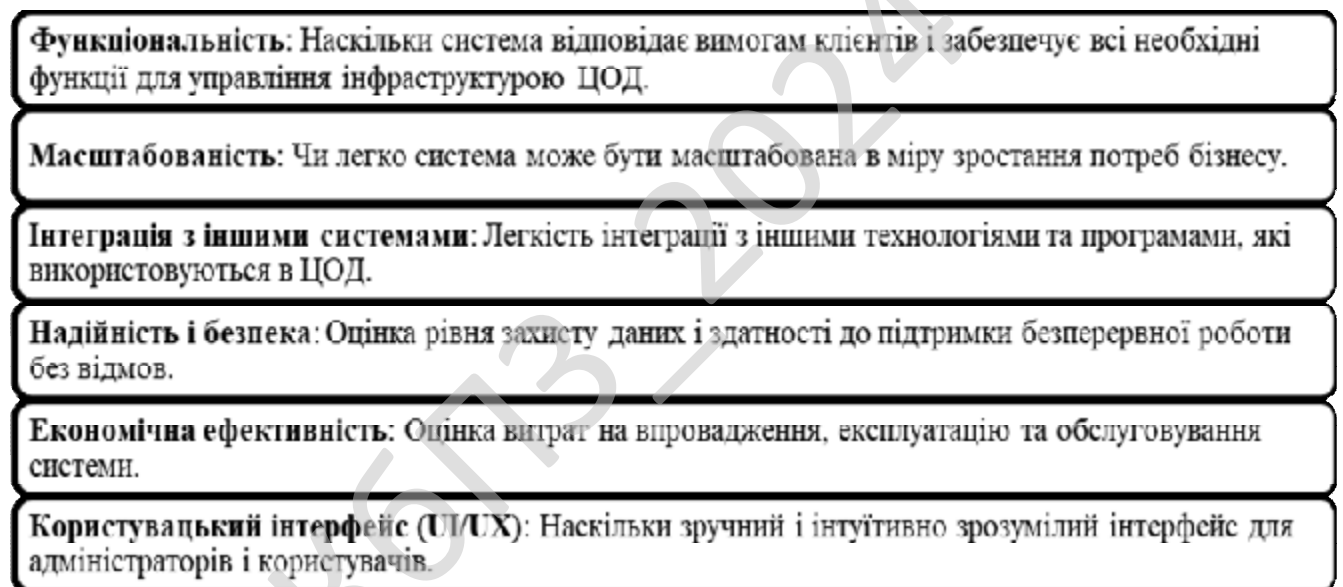


Рисунок 7.2 – Критерії експертної оцінки

Для оцінки варто залучити групу експертів з різних областей: спеціалісти з ІТ та інфраструктури (системні адміністратори, інженери), експерти з бізнес-аналітики та економіки, представники з технічної підтримки і користувачів, представники з безпеки інформації.

Експертна оцінка може бути проведена за допомогою таких методів:

– **Метод Дельфі** – експерти незалежно оцінюють кожен критерій за шкалою (наприклад, від 1 до 5). Потім проводиться колективний аналіз результатів, і оцінки коригуються на основі отриманих зворотних зв'язків до досягнення консенсусу.

– **Метод парних порівнянь** – кожен критерій оцінюється через порівняння з іншими, визначаючи, який із них є більш важливим, і з яким рівнем ваги.

Експерти оцінюють кожен критерій по 5-бальній шкалі, де 1 – дуже погано, 5 – відмінно. Отримані оцінки групуємо в таблицю 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Оцінка	Оцінка	Оцінка	Оцінка	Оцінка
	1	2	3	4	5
Функціональність	4	5	4	4	5
Масштабованість	5	4	4	5	4
Інтеграція з іншими системами	3	5	4	3	5
Надійність і безпека	5	4	4	5	5
Економічна ефективність	4	5	4	4	5
UI/UX	4	3	5	4	45.

Для кожного критерію визначаються вагові коефіцієнти, які показують важливість кожного критерію для загальної оцінки: функціональність – 0.25, масштабованість – 0.20, інтеграція з іншими системами – 0.15, надійність і безпека – 0.20, економічна ефективність – 0.10, ui/ux – 0.10. Коефіцієнти визначено шляхом проведення додаткового голосування серед експертів або за допомогою методу парних порівнянь.

Після того, як кожен критерій отримав оцінку, загальна оцінка проекту визначається за формулою:

$$O = \sum (O_i \times W_i) \quad (7.1)$$

де:

$O$  – загальна оцінка проекту,

$O_i$  – оцінка для кожного критерію,

$W_i$  – ваговий коефіцієнт для кожного критерію.

Після підрахунків оцінка по кожному критерію виглядає наступним чином: функціональність: 4.5, масштабованість: 4.5, інтеграція з іншими системами: 4, надійність і безпека: 4.75, економічна ефективність: 4.5, ціна: 4.0

Загальна оцінка буде:

$$O = (4.5 \times 0.25) + (4.5 \times 0.20) + (4.0 \times 0.15) + (4.75 \times 0.20) + (4.5 \times 0.10) + (4.0 \times 0.10) = 4.475$$

Оцінка 4.475 може свідчити про високу привабливість проекту. Якщо загальна оцінка є більшою за визначений поріг (наприклад, 4.0), це означає, що проект має значний потенціал для успішного впровадження та розвитку.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи керування інфраструктурою Центрів Обробки Даних (ЦОД) оптимально використовувати метод аналогії.

Метод аналогій (порівняльний метод) – це метод, який базується на порівнянні вартості розробки системи з вартістю подібних проектів, що були реалізовані в минулому. Порівнюється не тільки загальна вартість, але й специфікації системи, складність та масштаби реалізації.

Ключовою перевагою методу є проста та швидка оцінка, якщо є доступ до інформації про аналогічні проекти. Може використовуватися на ранніх етапах проекту.

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Варто враховувати недоліки – висока залежність від наявності подібних проєктів і їхніх характеристик, може бути не точним, якщо відмінності між проєктами значні.

#### **7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості**

Економічна ефективність від впровадження системи керування інфраструктурою ЦОД може бути оцінена через ряд фінансових і нефінансових показників (рисунок 7.3).

Впровадження системи керування інфраструктурою ЦОД дає значну економічну вигоду завдяки зниженню витрат на обслуговування, енергоспоживання, ризиків та підвищенню продуктивності. Загальна економічна ефективність оцінюється в \$650,000 на рік, що підтверджує вигідність такого проєкту для організації.

#### **7.5 Пропозиція алгоритму просування проєкту розробки ПЗ**

Алгоритм просування проєкту програмної реалізації системи керування інфраструктурою ЦОД може бути розділений на кілька етапів, які сприятимуть успішному запуску, поширенню та моніторингу ефективності (рисунок 7.4).

Алгоритм просування проєкту програмної реалізації системи керування інфраструктурою ЦОД включає кроки від глибокого аналізу цільової аудиторії до моніторингу результатів і коригування стратегії. Залучення правильних партнерів, впровадження активних маркетингових заходів і надання якісної підтримки користувачам дозволяє значно підвищити шанси на успішне просування та реалізацію проєкту.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69







## 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху для проєкту програмної реалізації системи керування інфраструктурою ЦОД включають кілька основних аспектів (рисунок 7.6), які впливають на ефективність, стабільність і конкурентоспроможність продукту.



Рисунок 7.6 – Ключові фактори успіху проєкту

Ключові фактори успіху проєкту програмної реалізації системи керування інфраструктурою ЦОД включають технічну надійність, безпеку, зручність використання, оптимізацію витрат, гнучкість у налаштуваннях, ефективну підтримку та супровід, а також здатність адаптуватися до змін та вимог ринку. Важливо забезпечити безперебійну роботу системи і надавати високий рівень сервісу для користувачів.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Стрімкий розвиток галузі інформаційних технологій (ІТ), який відбувається не тільки у сфері створення програмних продуктів, а й в управлінні виробництвом, банківській системі, бізнесі, системі освіти, на транспорті, у сфері обслуговування призвів до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння.

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливості. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, поперек тощо.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

## 8.2 Аналіз умов праці

Приміщення розташовано на третьому поверсі п'ятиповерхового будинку. У приміщенні розташовано 3 робочих місць з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м<sup>2</sup>, об'єм не менше 20 м<sup>3</sup>. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м<sup>2</sup>. Необхідна площа на 3 робочих місця із установленими ПК складає 18 м<sup>2</sup>, що не перевищує фактичну. Обсяг приміщення, що припадає на одного працюючого, складає 31,5 м<sup>3</sup>, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м<sup>3</sup> [2].

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань. Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається бібліограф при роботі з ПК, приведені в таблиці 8.1.

Таблиця 8.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

По категорії вибухо- і пожежонебезпеки, дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії пожежонебезпеки і

поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [13] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

Зв ступенем небезпеки поразки людей електричним струмом відділ, класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціонування в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціонування» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіНЗ.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин» [2].

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м<sup>3</sup>.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

### **8.3 Техніка безпеки та протипожежна профілактика**

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [13] будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізолюваний кабель. Висота проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1 м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемми заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 В, а одна, забезпечує подачу напруги в 36 В. Про це є відповідні написи на кожному розподільному щиті.

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

#### **8.4 Розрахункова частина**

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм., (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною  $L=1,6$  м., та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>80</b>



Еквівалентний діаметр вертикального електрода (кутка) [12]:

$$D_B = 0,95 \cdot K = 0,95 \cdot 63 = 59,85 \text{ мм} = 0,0598 \text{ м.}$$

де  $K = 63 \text{ мм}$ . – розмір металевого кутка (заданий).

Відношення  $A/L = 3/1,6 = 1,87$ .

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [12]:

$$R_O = 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_B) + (1/2) \cdot \lg((4T+L)/(4T-L))] = \\ = 0,366 \cdot (54,5/1,6) \cdot [(\lg(2 \cdot 1,6/0,0598) + (1/2) \lg((4 \cdot 1,4 + 1,6)/(4 \cdot 1,4 - 1,6)))] = 23 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів  $K_{св} = 0,62$  при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [12].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при  $R_{ЗН} = 4 \text{ Ом}$ :

$$N = R_O / (K_{св} R_{ЗН}) = 23 / (0,62 \cdot 4) = 9,3 \approx 9 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{П} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 9 = 28,35 \approx 28 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту  $K_{П}$  [12]:

$$R_{П} = 0,366 \cdot (\rho \cdot K_{П} / L_{П}) \lg(2 \cdot L_{П}^2 / (B \cdot t)) = \\ = 0,366 \cdot (40 \cdot 5 / 28) \cdot \lg((2 \cdot 28^2) / (0,06 \cdot 0,6)) = 12,2 \text{ Ом.}$$

де  $K_{П} = 5$  – табличне значення кліматичного коефіцієнта питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [12]:

$B = 60 \text{ мм} = 0,06 \text{ м}$  – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [12]:

$$R = (R_O \cdot R_{П}) / (R_O \cdot \eta_{П} + N \cdot R_{П} \cdot K_{св}) = \\ = (23 \cdot 12,2) / (23 \cdot 0,6 + 9 \cdot 12,2 \cdot 0,62) = 3,48 \text{ Ом.}$$

де  $\eta_{П} = 0,6$  – табличне значення коефіцієнта екранування з'єднуючої полоси [12].

Умова  $R \leq R_{ЗН}$  виконується ( $3,48 \leq 4$ ).

Оскільки при 9 вертикальних електродах  $R$  суттєво менше  $R_{3H}$ , зменшимо кількість вертикальних електродів  $N$  до 8 і виконаємо перерахунок. У результаті остаточно отримали:  $R = 3,91$  Ом. при кількості вертикальних електродів  $N=8$ .

### 8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>83</b>

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування інфраструктурою ЦОД.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів керування інфраструктурою ЦОД.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем керування інфраструктурою ЦОД.
- Досліджена система керування інфраструктурою ЦОД.
- На основі отриманих результатів досліджень створена програмна реалізація системи керування інфраструктурою ЦОД.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування інфраструктурою ЦОД.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 28147:2009.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>85</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Щельник О.В. Дослідження та програмна реалізація системи керування інфраструктурою ЦОД // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
4. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
5. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
6. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
7. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
8. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
9. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
10. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
11. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

					<b>БКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

12. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
13. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
14. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
15. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
16. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.
17. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
18. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

19. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

20. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

21. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

22. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

23. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

24. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

25. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

26. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

27. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

28. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

29. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

30. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

31. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

32. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

33. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

34. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

35. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

36. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

37. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

38. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

39. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного

					ВКРМ-123.24.0049.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

40. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

41. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

42. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

43. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

44. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

45. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

46. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

					<b>ВКРМ-123.24.0049.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

47. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

48. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

50. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

51. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

52. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-123.24.0049.00.00.ТЗ</b>		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Щельник О.В.				Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.			М			
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи керування інфраструктурою ЦОД.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи керування інфраструктурою ЦОД.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0049.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи керування інфраструктурою ЦОД;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0049.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0049.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті техніка безпеки та протипожежна профілактика.

					ВКРМ-123.24.0049.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

– Наукова новизна	– 1 аркуш.
– Структурна схема системи	– 1 аркуш.
– Функціональна схема системи	– 1 аркуш.
– Діаграма процесів	– 1 аркуш.
– Блок-схема алгоритму роботи програми	– 2 аркуша.
– Показники економічної ефективності	– 1 аркуш.
– Пояснювальна записка	– 92 аркуші.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2024 р.

					ВКРМ-123.24.0049.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Дреєв О.М.

*Дослідження та програмна реалізація  
системи керування інфраструктурою ЦОД*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 17

Літера: РП

Кропивницький – 2024 року

## Основна програма

```
import psutil
import os
import time
import logging
from datetime import datetime

# Налаштування журналювання
logging.basicConfig(filename='dc_management.log', level=logging.INFO)

class DCManagement:

    def __init__(self):
        self.servers = []
        self.networks = []
        self.storage_systems = []

    # Додавання сервера до списку серверів ЦОД
    def add_server(self, server):
        self.servers.append(server)

    # Додавання мережевого пристрою
    def add_network_device(self, network_device):
        self.networks.append(network_device)

    # Додавання системи зберігання
    def add_storage_system(self, storage):
        self.storage_systems.append(storage)

    # Моніторинг ресурсів сервера
    def monitor_resources(self):
        for server in self.servers:
            cpu_usage = server.get_cpu_usage()
            memory_usage = server.get_memory_usage()
            disk_usage = server.get_disk_usage()
            logging.info(f'Server: {server.name} | CPU: {cpu_usage}% | Memory: {memory_usage}% | Disk: {disk_usage}%')

    # Управління резервним копіюванням
    def backup_management(self):
        for storage in self.storage_systems:
            storage.perform_backup()

    # Управління енергоспоживанням
    def energy_management(self):
        for server in self.servers:
            server.adjust_energy_usage()

    # Захист від кіберзагроз
    def cybersecurity_management(self):
        for network_device in self.networks:
            network_device.scan_for_threats()

    # Управління змінами інфраструктури
    def change_management(self):
        for server in self.servers:
            server.apply_pending_updates()

class Server:

    def __init__(self, name, ip_address):
        self.name = name
        self.ip_address = ip_address

    # Отримання використання ЦП
    def get_cpu_usage(self):
        return psutil.cpu_percent(interval=1)
```

```

# Отримання використання пам'яті
def get_memory_usage(self):
    return psutil.virtual_memory().percent

# Отримання використання диску
def get_disk_usage(self):
    return psutil.disk_usage('/').percent

# Налаштування енергоспоживання
def adjust_energy_usage(self):
    if self.get_cpu_usage() < 10:
        logging.info(f'{self.name}: Switching to low power mode.')
    else:
        logging.info(f'{self.name}: Maintaining normal power mode.')

# Встановлення оновлень
def apply_pending_updates(self):
    logging.info(f'{self.name}: Checking for updates.')
    # Сюди можна додати запит до системи оновлень для встановлення оновлень
    logging.info(f'{self.name}: Applying updates.')

class NetworkDevice:

    def __init__(self, name, ip_address):
        self.name = name
        self.ip_address = ip_address

    # Сканування на наявність загроз
    def scan_for_threats(self):
        logging.info(f'{self.name}: Performing security scan.')
        # Додати логіку для сканування мережевих загроз
        logging.info(f'{self.name}: Security scan complete. No threats
detected.')

class StorageSystem:

    def __init__(self, name, capacity_gb):
        self.name = name
        self.capacity_gb = capacity_gb
        self.used_space_gb = 0

    # Виконання резервного копіювання
    def perform_backup(self):
        logging.info(f'{self.name}: Starting backup.')
        time.sleep(2) # Імітація резервного копіювання
        logging.info(f'{self.name}: Backup complete.')

# Головна функція для запуску програми
if __name__ == "__main__":

    # Ініціалізація інфраструктури
    dc_management = DCManagement()

    # Створення серверів
    server1 = Server(name="Server1", ip_address="192.168.1.1")
    server2 = Server(name="Server2", ip_address="192.168.1.2")

    # Додавання серверів до системи управління
    dc_management.add_server(server1)
    dc_management.add_server(server2)

    # Створення мережевих пристроїв
    router = NetworkDevice(name="Router", ip_address="192.168.0.1")
    switch = NetworkDevice(name="Switch", ip_address="192.168.0.2")

    # Додавання мережевих пристроїв
    dc_management.add_network_device(router)
    dc_management.add_network_device(switch)

```

```
# Створення системи зберігання
storage1 = StorageSystem(name="Storage1", capacity_gb=1000)

# Додавання системи зберігання до ЦОД
dc_management.add_storage_system(storage1)

# Періодичне виконання операцій
while True:

    # Запис початкового часу операції
    start_time = datetime.now()

    # Виконання моніторингу ресурсів
    dc_management.monitor_resources()

    # Виконання резервного копіювання
    dc_management.backup_management()

    # Управління енергоспоживанням
    dc_management.energy_management()

    # Управління кібербезпекою
    dc_management.cybersecurity_management()

    # Управління змінами інфраструктури
    dc_management.change_management()

    # Запис завершального часу операції
    end_time = datetime.now()

    # Логування часу виконання
    logging.info(f'Cycle complete. Duration: {end_time - start_time}')

    # Очікування 10 секунд перед наступним циклом
    time.sleep(10)
```

```
import logging

class DatabaseManager:
    def __init__(self, db_type, db_name, user, password):
        self.db_type = db_type
        self.db_name = db_name
        self.user = user
        self.password = password
        self.connection = None

    # Підключення до бази даних
    def connect(self):
        logging.info(f'Connecting to {self.db_type} database: {self.db_name}')
        # Тут додається код для підключення до бази даних
        # Наприклад, psycopg2.connect() для PostgreSQL

    # Виконання запитів
    def execute_query(self, query):
        logging.info(f'Executing query: {query}')
        # Тут код для виконання запиту

    # Закриття підключення до бази даних
    def close_connection(self):
        logging.info('Closing database connection')
        # Тут код для закриття підключення
```

КБПЗ\_2024

## Файл auto\_scaling.py

```
import logging

class AutoScaler:
    def __init__(self, cpu_threshold, memory_threshold):
        self.cpu_threshold = cpu_threshold
        self.memory_threshold = memory_threshold

    # Перевірка метрик для масштабування
    def check_metrics(self, cpu_usage, memory_usage):
        if cpu_usage > self.cpu_threshold or memory_usage >
self.memory_threshold:
            self.scale_up()
        else:
            self.scale_down()

    # Масштабування вгору
    def scale_up(self):
        logging.info('Scaling up resources')
        # Тут додається код для збільшення ресурсів

    # Масштабування вниз
    def scale_down(self):
        logging.info('Scaling down resources')
        # Тут додається код для зменшення ресурсів
```

КБПЗ\_2024

**Файл network\_traffic\_monitoring.py**

```
import logging

class NetworkTrafficMonitor:
    def __init__(self):
        self.traffic_data = []

    # Моніторинг вхідного трафіку
    def monitor_incoming_traffic(self, data):
        logging.info(f'Incoming traffic: {data} MB')
        self.traffic_data.append(data)

    # Моніторинг вихідного трафіку
    def monitor_outgoing_traffic(self, data):
        logging.info(f'Outgoing traffic: {data} MB')
        self.traffic_data.append(data)

    # Аналіз трафіку
    def analyze_traffic(self):
        logging.info('Analyzing network traffic')
        # Тут код для аналізу зібраного трафіку
```

**Файл user\_management.py**

```
import logging

class UserManager:
    def __init__(self):
        self.users = {}

    # Додавання нового користувача
    def add_user(self, username, role):
        self.users[username] = role
        logging.info(f'User {username} added with role {role}')

    # Видалення користувача
    def remove_user(self, username):
        if username in self.users:
            del self.users[username]
            logging.info(f'User {username} removed')

    # Призначення ролі користувачеві
    def assign_role(self, username, role):
        if username in self.users:
            self.users[username] = role
            logging.info(f'Role {role} assigned to user {username}')
```

**Файл access\_control\_integration.py**

```
import logging

class AccessControlSystem:
    def __init__(self):
        self.access_logs = []

    # Перевірка доступу користувача
    def check_access(self, user_id):
        logging.info(f'Checking access for user {user_id}')
        # Код для перевірки прав доступу

    # Логування доступу
    def log_access(self, user_id, access_granted):
        self.access_logs.append((user_id, access_granted))
        logging.info(f'Access log: User {user_id}, Access Granted:
{access_granted}')
```

**Файл environment\_monitoring.py**

```
import logging

class EnvironmentMonitor:
    def __init__(self, location):
        self.location = location

    # Перевірка температури
    def check_temperature(self):
        logging.info(f'Checking temperature in {self.location}')
        # Код для зчитування температури

    # Перевірка вологості
    def check_humidity(self):
        logging.info(f'Checking humidity in {self.location}')
        # Код для зчитування рівня вологості
```

**Файл notifications.py**

```
import logging

class NotificationSystem:
    def __init__(self):
        self.notifications = []

    # Надсилання сповіщення
    def send_notification(self, message):
        self.notifications.append(message)
        logging.info(f'Notification sent: {message}')

    # Отримання критичних подій
    def get_critical_events(self):
        logging.info('Fetching critical events')
        # Код для отримання критичних подій
```

**Файл analytics.py**

```

import logging

class Analytics:
    def __init__(self):
        self.data_points = []

    # Додавання точки даних
    def add_data_point(self, metric, value):
        self.data_points.append((metric, value))
        logging.info(f'Data point added: {metric} - {value}')

    # Аналіз історичних даних
    def analyze_data(self):
        logging.info('Analyzing historical data')
        # Код для аналітики

```

**Файл virtualization\_integration.py**

```

import logging

class VirtualizationIntegration:
    def __init__(self):
        self.virtual_machines = []

    # Додавання віртуальної машини
    def add_virtual_machine(self, vm_name, resources):
        self.virtual_machines.append((vm_name, resources))
        logging.info(f'Virtual machine {vm_name} added with resources {resources}')

    # Видалення віртуальної машини
    def remove_virtual_machine(self, vm_name):
        self.virtual_machines = [vm for vm in self.virtual_machines if vm[0] != vm_name]
        logging.info(f'Virtual machine {vm_name} removed')

```

**Файл load\_balancing.py**

```

import logging

class LoadBalancer:
    def __init__(self):
        self.server_loads = {}

    # Додавання серверного навантаження
    def add_server(self, server_name):
        self.server_loads[server_name] = 0
        logging.info(f'Server {server_name} added to load balancer')

    # Оновлення навантаження сервера
    def update_load(self, server_name, load):
        if server_name in self.server_loads:
            self.server_loads[server_name] = load
            logging.info(f'Load updated for {server_name}: {load}')

    # Розподілення навантаження
    def balance_load(self):
        logging.info('Balancing load across servers')
        # Код для балансування навантаження між серверами

```

**Файл extended\_logging.py**

```
import logging
from datetime import datetime

class ExtendedLogger:
    def __init__(self, log_file):
        self.log_file = log_file
        logging.basicConfig(filename=self.log_file, level=logging.DEBUG)

    # Логування критичних подій
    def log_critical_event(self, event):
        logging.critical(f'{datetime.now()} - CRITICAL: {event}')

    # Логування помилок
    def log_error(self, error):
        logging.error(f'{datetime.now()} - ERROR: {error}')

    # Логування попереджень
    def log_warning(self, warning):
        logging.warning(f'{datetime.now()} - WARNING: {warning}')

    # Логування інформаційних подій
    def log_info(self, info):
        logging.info(f'{datetime.now()} - INFO: {info}')
```

**Файл os\_update\_management.py**

```
import logging

class OSUpdateManager:
    def __init__(self, os_type):
        self.os_type = os_type

    # Перевірка наявності оновлень
    def check_for_updates(self):
        logging.info(f'Checking for updates on {self.os_type}')
        # Код для перевірки наявності оновлень для конкретної ОС

    # Завантаження та встановлення оновлень
    def install_updates(self):
        logging.info(f'Installing updates for {self.os_type}')
        # Код для завантаження та встановлення оновлень
```

**Файл performance\_testing.py**

```
import logging

class PerformanceTester:
    def __init__(self, component_name):
        self.component_name = component_name

    # Виконання тесту продуктивності
    def run_performance_test(self):
        logging.info(f'Running performance test on {self.component_name}')
        # Код для проведення тесту

    # Збереження результатів тесту
    def save_test_results(self, results):
        logging.info(f'Saving performance test results for
{self.component_name}')
        # Код для збереження результатів
```

**Файл cloud\_integration.py**

```
import logging

class CloudIntegration:
    def __init__(self, provider_name):
        self.provider_name = provider_name

    # Підключення до хмарного провайдера
    def connect_to_provider(self):
        logging.info(f'Connecting to cloud provider: {self.provider_name}')
        # Код для підключення до хмари

    # Синхронізація даних із хмарою
    def sync_data(self):
        logging.info(f'Syncing data with cloud provider: {self.provider_name}')
        # Код для синхронізації даних
```

**Файл log\_storage\_management.py**

```
import logging

class LogStorageManager:
    def __init__(self, storage_limit):
        self.storage_limit = storage_limit

    # Очищення старих журналів
    def clear_old_logs(self):
        logging.info('Clearing old logs to free up storage')
        # Код для очищення старих журналів

    # Збереження журналів у довготривалі сховище
    def archive_logs(self):
        logging.info('Archiving logs for long-term storage')
        # Код для архівування журналів
```

**Файл disaster\_recovery.py**

```
import logging

class DisasterRecovery:
    def __init__(self, recovery_plan):
        self.recovery_plan = recovery_plan

    # Запуск плану відновлення
    def initiate_recovery(self):
        logging.info(f'Initiating disaster recovery plan: {self.recovery_plan}')
        # Код для запуску плану відновлення

    # Перевірка готовності до відновлення
    def check_recovery_status(self):
        logging.info('Checking disaster recovery readiness')
        # Код для перевірки готовності до аварійного відновлення
```

**Файл cooling\_optimization.py**

```
import logging

class CoolingOptimizer:
    def __init__(self, target_temperature):
        self.target_temperature = target_temperature

    # Налаштування системи охолодження
    def adjust_cooling(self, current_temperature):
        if current_temperature > self.target_temperature:
            logging.info('Increasing cooling power')
            # Код для збільшення охолодження
        else:
            logging.info('Maintaining current cooling level')
            # Код для підтримки поточного рівня охолодження
```

**Файл physical\_security.py**

```
import logging

class PhysicalSecurityManager:
    def __init__(self):
        self.security_logs = []

    # Відстеження доступу до приміщення
    def track_access(self, personnel_id):
        logging.info(f'Tracking access for personnel ID: {personnel_id}')
        # Код для реєстрації доступу

    # Оновлення рівня безпеки
    def update_security_level(self, level):
        logging.info(f'Security level updated to {level}')
        # Код для зміни рівня безпеки
```

**Файл vulnerability\_scanning.py**

```
import logging

class VulnerabilityScanner:
    def __init__(self):
        self.vulnerabilities = []

    # Сканування на наявність вразливостей
    def scan_for_vulnerabilities(self):
        logging.info('Scanning for vulnerabilities')
        # Код для виконання сканування

    # Збереження результатів сканування
    def save_scan_results(self, results):
        logging.info('Saving vulnerability scan results')
        # Код для збереження результатів
```

## Файл sla\_reporting.py

```
import logging

class SLAReportManager:
    def __init__(self):
        self.reports = []

    # Генерація звіту SLA
    def generate_sla_report(self, data):
        logging.info('Generating SLA report')
        self.reports.append(data)

    # Відправка звіту
    def send_sla_report(self, recipient):
        logging.info(f'Sending SLA report to {recipient}')
        # Код для відправки звіту
```

КБПЗ\_2024

## Файл multi\_zone\_support.py

```
import logging

class MultiZoneSupport:
    def __init__(self, zones):
        self.zones = zones

    # Додавання нової зони
    def add_zone(self, zone_name):
        self.zones.append(zone_name)
        logging.info(f'Zone {zone_name} added')

    # Моніторинг зон
    def monitor_zones(self):
        logging.info('Monitoring multiple zones')
        # Код для моніторингу зон
```

КБПЗ\_2024

```
import logging

class MaintenanceManager:
    def __init__(self):
        self.maintenance_schedule = []

    # Додавання обслуговування до розкладу
    def schedule_maintenance(self, date, equipment):
        self.maintenance_schedule.append((date, equipment))
        logging.info(f'Maintenance scheduled for {equipment} on {date}')

    # Виконання планового обслуговування
    def perform_maintenance(self, equipment):
        logging.info(f'Performing maintenance on {equipment}')
        # Код для виконання обслуговування
```

КБПЗ\_2024

**Файл resource\_forecasting.py**

```
import logging

class ResourceForecaster:
    def __init__(self):
        self.forecast_data = []

    # Додавання даних для прогнозування
    def add_data(self, metric, value):
        self.forecast_data.append((metric, value))
        logging.info(f'Forecast data added: {metric} - {value}')

    # Прогнозування ресурсів
    def forecast_resources(self):
        logging.info('Forecasting future resource needs')
        # Код для прогнозування
```

**Файл failure\_monitoring.py**

```
import logging

class FailureMonitor:
    def __init__(self):
        self.failures = []

    # Додавання нової відмови
    def record_failure(self, component, description):
        self.failures.append((component, description))
        logging.info(f'Failure recorded: {component} - {description}')

    # Моніторинг відмов компонентів
    def monitor_failures(self):
        logging.info('Monitoring for component failures')
        # Код для моніторингу
```