

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  
“\_\_\_” \_\_\_\_\_ 2026 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“Програмне забезпечення системи проектування ЦОД з**  
**використанням технологій кастомізації”**

Виконав здобувач вищої освіти  
IV курсу, групи КІ-22-2  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Рубан А.С.  
«\_\_\_» \_\_\_\_\_ 2026 р.

Керівник проекту  
кандидат фізико-математичних наук, доцент  
\_\_\_\_\_ Петренюк В.І.  
«\_\_\_» \_\_\_\_\_ 2026 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Освітній ступінь бакалавр  
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  
« 27 » лютого 2026 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

*Рубану Артему Сергійовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації*

2. Керівник роботи *Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент*  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 136-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи проектування ЦОД з використанням технологій кастомізації*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

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

*Структурна схема системи* *1 аркуш*

*Функціональна схема системи* *1 аркуш*

*Діаграма процесів* *1 аркуш*

*Блок-схема алгоритму роботи додатку* *2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Примітка |
|-------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|----------|
| 1.    | Аналіз існуючих систем                                                                      | 10.03.2026 р.                                                                                         |          |
| 2.    | Постановка задачі, оформлення ТЗ                                                            | 15.03.2026 р.                                                                                         |          |
| 3.    | Розробка моделі компонента                                                                  | 20.03.2026 р.                                                                                         |          |
| 4.    | Розробка структур даних                                                                     | 25.03.2026 р.                                                                                         |          |
| 5.    | Розробка алгоритмів зв'язку та відображення                                                 | 30.03.2026 р.                                                                                         |          |
| 6.    | Програмування алгоритмів                                                                    | 10.04.2026 р.                                                                                         |          |
| 7.    | Оформлення ПЗ                                                                               | 17.04.2026 р.                                                                                         |          |
| 8.    | Попередній захист роботи                                                                    | 23.05.2026 р.                                                                                         |          |
|       |                                                                                             |                                                                                                       |          |
|       |                                                                                             |                                                                                                       |          |
|       |                                                                                             |                                                                                                       |          |
|       |                                                                                             |                                                                                                       |          |

Дата видачі завдання  
« 27 » лютого 2026 р.

Підпис керівника

\_\_\_\_\_

Петренюк В.І.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 27 » лютого 2026 р.

Підпис здобувача

\_\_\_\_\_

Рубан А.С.  
(прізвище та ініціали)

## **АНОТАЦІЯ**

**Рубан А.С. Програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проектування ЦОД з використанням технологій кастомізації.

Метою розробки є програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації.

Результат роботи – програмна реалізація системи проектування ЦОД з використанням технологій кастомізації.

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, проектування ЦОД

## ABSTRACT

**Ruban A.S. Software for data center design system using customization technologies. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.**

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for a data center design system using customization technologies.

The purpose of the development is software for a data center design system using customization technologies.

The result of the work is a software implementation of a data center design system using customization technologies.

In the process of working on the software model, an analysis of existing hardware and software tools 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 OS Windows 10/11.

The program was developed in the Python environment.

**Keywords:** computer engineering, data center design

## ЗМІСТ

|                                                                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....                                                                                                               | 2  |
| ВСТУП.....                                                                                                                                                                  | 3  |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....                                                                                                                                 | 4  |
| 1.1 Призначення системи.....                                                                                                                                                | 4  |
| 1.2 Область застосування.....                                                                                                                                               | 5  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....                                                                                                                                | 7  |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... | 7  |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....                                                                                            | 24 |
| 2.3 Розгорнута постановка завдання .....                                                                                                                                    | 30 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....                                                                                                                               | 31 |
| 3.1 Опис функціонування системи .....                                                                                                                                       | 31 |
| 3.2 Розробка структурної схеми.....                                                                                                                                         | 35 |
| 3.3 Розробка функціональної схеми .....                                                                                                                                     | 41 |
| 3.4 Розробка діаграми процесів.....                                                                                                                                         | 50 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....                                                      | 52 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....                                                                                                       | 52 |
| 4.2 Захист розробленого програмного забезпечення.....                                                                                                                       | 63 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....                                                                                                                      | 68 |
| 6 ОСНОВНІ ВИСНОВКИ.....                                                                                                                                                     | 73 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                                                                            | 75 |

|                 |                |                 |              |             |                                                                                                |                     |              |                |
|-----------------|----------------|-----------------|--------------|-------------|------------------------------------------------------------------------------------------------|---------------------|--------------|----------------|
|                 |                |                 |              |             | <b>ВКРБ-123.26.0055.00.00.ПЗ</b>                                                               |                     |              |                |
| <b>Вим.</b>     | <b>Арк.</b>    | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> | <i>Програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації</i> | <b>Літ.</b>         | <b>Аркуш</b> | <b>Аркушів</b> |
| <i>Розроб.</i>  | Рубан А.С.     |                 |              |             |                                                                                                | <b>Б</b>            | 1            | 82             |
| <i>Перев.</i>   | Петренко В.І.  |                 |              |             |                                                                                                |                     |              |                |
| <i>Н.контр.</i> | Коваленко А.С. |                 |              |             |                                                                                                | <b>ЦНТУ КІ-22-2</b> |              |                |
| <i>Затв.</i>    | Смірнов О.А.   |                 |              |             |                                                                                                |                     |              |                |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ**

|     |   |                                       |
|-----|---|---------------------------------------|
| ІТ  | — | інформаційні технології               |
| ПЗ  | — | програмне забезпечення                |
| ЦОД | — | центр обробки даних                   |
| QoS | — | механізми контролю й керування якістю |

К6ПЗ\_2026

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 2    |

## ВСТУП

**Актуальність теми.** Світ думає, що розуміє центри обробки даних, поки штучний інтелект не змінив усе. До 2026 року будівництво центрів обробки даних вже не буде лише питанням масштабу чи часу безвідмовної роботи. Йдеться про щільність, ефективність, інтелект та швидкість. Зростання штучного інтелекту, передового охолодження, систем живлення наступного покоління та вимог сталого розвитку змушує галузь переосмислювати себе з нуля.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації.

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

- Огляд існуючих систем проектування ЦОД з використанням технологій кастомізації.
- Дослідження системи проектування ЦОД з використанням технологій кастомізації.
- Програмна реалізація системи проектування ЦОД з використанням технологій кастомізації.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі проектування ЦОД з використанням технологій кастомізації.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 3    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Ємність, надійність та час безвідмовної роботи завжди будуть ключовими характеристиками рішень для центрів обробки даних. Однак у 2026 році для сучасних локальних та регіональних центрів обробки даних важливі будуть ще кілька факторів.

### Джерела енергії

Згідно зі звітом IEEE «Зростання центрів обробки даних та готовність до мережі» (травень 2025 р.), масштаб сучасних центрів обробки даних тепер ставить їх в ту ж категорію, що й великі виробничі потужності, з точки зору їхнього впливу на електромережу. Об'єкти, які раніше споживали десятки мегават, тепер планують використовувати сотні, що виявляє обмеження в мережі передачі та створює нові виклики для інженерів. Така ситуація дає конкурентну перевагу центрам обробки даних, які мають доступ до надійного та масштабованого електропостачання, а також тим, хто співпрацює з великими регіональними постачальниками енергії з самого етапу планування їхньої інфраструктури.

### Штучний інтелект у процесах проектування та експлуатації

Штучний інтелект вже не лише є навантаженням для центрів обробки даних, але й стає ключовим інструментом для їх проектування та управління. Навіть на етапі проектування автоматизовані інструменти та внутрішні системи можуть пришвидшити планування та оптимізацію. В операційному середовищі машинне навчання вже використовується для динамічного управління потоком повітря, охолодженням, насосами та розподілом електроенергії. Наступним логічним кроком є створення конвергентних систем та «живих» цифрових двійників, які моделюють та оптимізують роботу об'єктів у режимі реального часу.

Це також змінює профіль спеціалістів у центрах обробки даних. Навички роботи з моделями штучного інтелекту, аналізу даних та управління інтегрованими цифровими платформами, особливо в середовищах з рідинним охолодженням та високоінтенсивними робочими навантаженнями, стають дедалі важливішими.

Це дозволяє створювати інструменти для:

- планування потужностей;
- профілактичне обслуговування;
- вимірювання ефективності;
- підтвердження сталості та відповідності нормативним вимогам.

Центри обробки даних, які можуть збирати, аналізувати та використовувати такі дані, матимуть значну перевагу в середовищі зі все більш суворими вимогами до моніторингу та підзвітності.

## 1.2 Область застосування

### Сталий розвиток як провідний пріоритет

Практики екологічної стійкості також ставатимуть дедалі важливішими майже для будь-якої організації, а центр обробки даних може бути невід'ємним партнером у політиці ESG бізнесу завдяки:

- вибір матеріалів з низьким вуглецевим слідом;
- зменшення «втіленого» вуглецю в конструкціях;
- управління водними ресурсами;
- повторне використання відпрацьованого тепла, як це реалізовано в центрі обробки даних AbsCloud.

Центри обробки даних, які можуть продемонструвати вимірювані результати – зниження PUE, повторне використання води, циркулярну економіку та відстежуваний вуглецевий слід – матимуть значно легший доступ до можливостей розвитку.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 5    |

У світовому масштабі стійкість центрів обробки даних тепер вимірюється не лише часом безвідмовної роботи, але й гнучкістю, або здатністю адаптуватися до змін у доступності мережі, нормативних актів та цін на енергоносії. Необхідно буде розробити системи, які не тільки стійкі до перебоїв, але й можуть взаємодіяти з енергомережею в режимі реального часу, динамічно реагувати на попит і підтримувати стабільну напругу.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ\_2026

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 6    |

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У 2026 році сектор центрів обробки даних стоїть на вирішальному роздоріжжі, зумовленому швидким розвитком штучного інтелекту, зростаючими енергетичними обмеженнями та зростанням суспільних очікувань щодо сталого розвитку.

Ці сили фундаментально змінюють пріоритети галузі. Нижче ми розглянемо десять основних тенденцій, що формують центри обробки даних.

#### Штучний інтелект та споживання енергії

Штучний інтелект зараз є основною рушійною силою як технологічної, так і економічної трансформації. Центри обробки даних є основою, що підтримує ШІ, забезпечуючи необхідну інфраструктуру для навчання, розміщення та запуску як генеративних, так і базових моделей ШІ.

Міжнародне енергетичне агентство (МЕА) прогнозує, що до 2026 року споживання енергії в центрах обробки даних може досягти 1050 ТВт·год, значною мірою через зростання потреб у робочих навантаженнях штучного інтелекту та використання графічних процесорів, які є значно енергоємнішими, ніж традиційні центри обробки даних. Тому я очікую ще більшої терміновості та зосередження на проектуванні стійких, ефективних та відповідальних центрів обробки даних, які можуть підтримувати революцію штучного інтелекту, мінімізуючи при цьому свій вплив на навколишнє середовище.

#### Рідинне охолодження, занурення та теплова щільність

Оскільки графічні процесори та стійки високої щільності стають все більш поширеними, традиційні методи охолодження більше не можуть задовольнити

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 7    |

потреби сучасних центрів обробки даних. У відповідь галузь швидко впроваджує передові технології охолодження, такі як рідинне охолодження безпосередньо до кристала, іммерсійне охолодження та двофазні системи охолодження.

Ці підходи продемонстрували здатність знизити споживання енергії, пов'язане з охолодженням, на цілих 50-60 відсотків у деяких тестах. Я передбачаю збільшення темпів впровадження цих технологій, які мають вирішальне значення для підтримки як економічної, так і екологічної стійкості кампусів центрів обробки даних, орієнтованих на штучний інтелект.

**Гібридний енергетичний баланс та автономність**

Щоб задовольнити зростаючі потреби в енергії та одночасно зменшити викиди вуглецю, оператори центрів обробки даних звертаються до поєднання джерел енергії та стратегій. Це включає інтеграцію відновлюваних джерел енергії, таких як сонячна та вітрова енергія, рішення для зберігання енергії, такі як довговічні акумулятори, та мікромережі.

Ми можемо очікувати, що в кількох країнах розглядатимуться або будуть пілотуватися більш радикальні рішення, такі як малі модульні реактори (ММР), для забезпечення стабільного та низьковуглецевого енергопостачання, особливо в регіонах, де існуюча енергомережа менш надійна.

**Регулювання, відповідальне використання води та місцева прийнятність**

Екологічні міркування стають центральними у розвитку центрів обробки даних. Органи влади в регіонах, які стикаються з нестачею води, тепер вимагають використання сухих або гібридних методів охолодження, а також використання «сірої» або переробленої води. Екологічні норми все більше зосереджуються на низьковуглецевих матеріалах, сертифікації BREEAM, комплексних оцінках впливу та ретельних методах встановлення на місці.

Залучення місцевої громади стало стратегічним фактором, оскільки проекти можуть зіткнутися із затримками або перешкодами без співпраці з громадою. Я прогножую, що ми побачимо більший акцент на місцевих програмах

інформаційно-просвітницької роботи та наймі фахівців зі зв'язків з громадськістю з акцентом на залучення місцевих органів влади.

### **Цифрові двійники, автоматизація та операційна оптимізація**

Впровадження цифрових двійників – віртуальних моделей, що використовуються для моделювання, моніторингу та оптимізації операцій у режимі реального часу – набуває все більшої зрілості в галузі. Ці інструменти дозволяють проводити прогнозне обслуговування, інтелектуальний розподіл навантаження, адаптивні стратегії охолодження та швидко реагувати на стан мережі.

Отже, ми можемо очікувати значного поштовху до автоматизації центрів обробки даних, включаючи управління тепловими потоками та робочими навантаженнями на основі штучного інтелекту, оскільки стає дедалі очевиднішим, що це ключовий важіль підвищення ефективності операцій.

### **Мегакампуси гігаватного масштабу**

Європа спостерігає появу мегакампусних центрів обробки даних, проекти яких перевищують сотні мегават, а іноді наближаються до одного гігавата або навіть перевищують його. Ці великомасштабні кампуси потребують величезних електричних ресурсів та надійної інфраструктури, включаючи високопродуктивне підключення, резервні джерела живлення та високоефективні системи охолодження. Вони представляють новий промисловий рубіж для цього сектору, і ми побачимо появу перших відчутних кампусів GW у Європі протягом наступних 12 місяців.

### **Циркулярність та ширші екологічні показники**

Хоча ефективність використання енергії (PuE) залишається фундаментальним показником, тепер вона доповнена додатковими показниками, такими як ефективність використання вуглецю (CuE), ефективність використання води (WuE) та комплексна оцінка життєвого циклу (LCA).

Ці орієнтири стають дедалі важливішими для інвесторів, регуляторів та клієнтів. Протягом наступного року ми побачимо, як промисловість почне

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 9    |

інтегрувати більше рішень для повторного використання матеріалів, модульного будівництва та рекуперації тепла, таких як забезпечення теплом міських або сільськогосподарських проектів, на найперших стадіях проектування.

### **Диверсифікація краю, географії та місць розташування**

Периферійні обчислення, що характеризуються розгортанням мікроцентрів обробки даних, швидко розвиваються, особливо в галузях, де низька затримка є критично важливою, таких як телемедицина, промислова автоматизація та автономні транспортні засоби. Ми також побачимо, як все більше великих операторів диверсифікують свою географічну присутність, щоб зменшити перевантаження енергомережі та використовувати місцеві відновлювані джерела енергії.

### **Дефіцит талантів та підвищення кваліфікації**

У цьому секторі спостерігається нестача спеціалізованих фахівців, зокрема інженерів у галузі механіки, енергетики, будівництва, охолодження та інфраструктури, пов'язаної зі штучним інтелектом. Подолання цієї прогалини вимагає потужних програм технічної підготовки, партнерства з університетами та зусиль щодо перепідготовки працівників із суміжних галузей. Очікується, що компанії, які успішно залучають, утримують та розвивають ці таланти, отримають значну конкурентну перевагу.

### **Масштабні інвестиції та соціальні проблеми**

Значні інвестиції продовжуватимуть надходити в інфраструктуру, пов'язану зі штучним інтелектом, включаючи придбання, консорціуми та державно-приватні партнерства, оскільки сектор переживає сплеск масштабних транзакцій.

Однак, спостерігається дедалі більша увага до вуглецевого сліду центрів обробки даних, прозорості щодо їхнього впливу та залучення місцевого бізнесу. Як наслідок, центрам обробки даних потрібно буде як ніколи раніше демонструвати, що вони є життєво важливим промисловим сектором, який відіграє ключову стратегічну роль у забезпеченні цифрового суверенітету.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 10   |

Керування сучасним центром обробки даних – нелегке завдання. Зі стрімким розвитком хмарних сервісів, робочих навантажень на основі штучного інтелекту та периферійних обчислень, центри обробки даних у 2026 році стали складнішими, ніж будь-коли. Програмне забезпечення для управління інфраструктурою центрів обробки даних (DCIM) стало важливим для контролю всіх рухомих частин – від серверів і стійок до живлення та охолодження. Ці інструменти надають операторам центрів обробки даних та менеджерам BIM «єдине джерело достовірної інформації» про активи та середовище об'єкта, усуваючи розрив між IT та управлінням об'єктами. Фактично, провідні організації інтегрують DCIM з іншими системами (від CMDB до платформ BIM та CAD), щоб створити єдине «скляне» уявлення про свою інфраструктуру, зменшуючи ручну роботу та покращуючи процес прийняття рішень у всіх командах.

Якщо ви менеджер BIM, архітектор або інженер, який займається проектуванням та експлуатацією центрів обробки даних, розуміння найкращих варіантів програмного забезпечення DCIM є надзвичайно важливим. Правильне рішення DCIM допомагає забезпечити відповідність ваших моделей проектування операційним реаліям, таким як потужність, вимоги до охолодження та розташування обладнання. Нижче ми оцінюємо та розглядаємо деякі з найкращих програм DCIM на 2026 рік, виділяючи те, що виділяє кожне з них у світі центрів обробки даних. Ці платформи обрані за їхні функції, можливості інтеграції та досвід у сприянні ефективній роботі центрів обробки даних.

### **Sunbird DCIM – зручне керування, орієнтоване на інтеграцію**

Sunbird DCIM широко визнаний як провідне рішення для DCIM, відоме своєю простотою використання та потужними можливостями моніторингу. Sunbird пропонує єдину централізовану панель інструментів для відстеження всього, від інвентаризації серверів до датчиків навколишнього середовища. Команди центрів обробки даних люблять моніторинг Sunbird у режимі реального часу критичних показників, таких як температура, вологість, навантаження на живлення та мережеве підключення. Програмне забезпечення надає чіткі,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 11   |

налаштовувані панелі інструментів та звіти, які допомагають швидко виявити потенційні проблеми. Sunbird також чудово справляється з управлінням активами – ви можете вести детальний перелік стійок, пристроїв, блоків живлення тощо, включаючи їх розташування та підключення.

Однією з найбільших переваг Sunbird є інтеграція. Він постачається з відкритим API та попередньо вбудованими інтеграціями, тому може взаємодіяти з іншими вашими системами (такими як система продажу квитків, CMDB або системи управління будівлями). Це означає, що Sunbird може бути частиною ширшої екосистеми для «єдиного огляду» операцій центру обробки даних. Незалежно від того, розгорнуто його як хмарний сервіс чи локально, Sunbird DCIM допомагає операторам центрів обробки даних оптимізувати потужності (простір, живлення, охолодження) та автоматизувати рутинні завдання. Акцент на зручності використання (інтуїтивно зрозумілий веб-інтерфейс і навіть підтримка мобільних пристроїв) та сильна підтримка, що не залежить від постачальників, роблять його найкращим вибором для команд, які прагнуть оптимізувати управління центром обробки даних та покращити видимість у режимі реального часу.

### **Nlyte – Надійний DCIM з інструментами проектування та планування**

Nlyte вже давно є провідним гравцем у сфері DCIM, відомим своїм комплексним набором функцій та надійністю. Багато великих підприємств покладаються на Nlyte для централізації відстеження активів своїх центрів обробки даних, планування потужностей та автоматизації робочих процесів. Платформа Nlyte пропонує моніторинг показників живлення та навколишнього середовища в режимі реального часу, а також детальне управління активами для серверів, мережевого обладнання, сховищ та обладнання об'єктів. Менеджери центрів обробки даних цінують багаті можливості візуалізації Nlyte – він включає інтерфейс у стилі CAD, який дозволяє вам відображати фізичне розташування вашого центру обробки даних. Це графічне представлення допомагає планувати розташування стійок та бачити, де кожен пристрій розташований у приміщеннях

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 12   |

та стійках, що неймовірно корисно для планування потужностей та управління змінами.

Однією з видатних особливостей Nlyte є його Smart Blueprints (розумні креслення). Цей інструмент дозволяє вам збирати та повторно використовувати шаблони дизайну центрів обробки даних і найкращі практики. Наприклад, ви можете спроектувати оптимальну конфігурацію стелажів або рядів і зберегти її як креслення для застосування на кількох об'єктах або проектах, забезпечуючи узгодженість і заощаджуючи час на нових проектах. Nlyte також має високі можливості налаштування та інтеграції – він пропонує відкриту архітектуру та API, що дозволяє підключати його до зовнішніх систем (таких як CMDB, інструменти моніторингу або бази даних BIM). Як усталене рішення DCIM, яке тепер підтримується великою інфраструктурною компанією, Nlyte має репутацію масштабованості та продуктивності. Якщо вам потрібна перевірена платформа DCIM, яка не тільки контролює та керує, але й допомагає в реплікації дизайну та майбутньому плануванні, Nlyte є головним претендентом.

### **Schneider Electric EcoStruxure IT – DCIM поєднується з корпоративною аналітикою**

EcoStruxure IT від Schneider Electric – це корпоративне рішення DCIM від одного з найбільших гравців у сфері живлення та охолодження для центрів обробки даних. EcoStruxure IT від Schneider (часто званий EcoStruxure IT Expert) надає централізовану платформу для моніторингу та управління всім середовищем вашого центру обробки даних – від розподілу живлення будівлі та систем ДБЖ до серверних стійок та датчиків навколишнього середовища. Ключовою перевагою EcoStruxure IT є його нейтральний підхід до постачальників. Незважаючи на те, що Schneider відома своїм апаратним забезпеченням, цей DCIM підтримує обладнання від різних постачальників, що дозволяє йому без проблем інтегруватися в різноманітні середовища центрів обробки даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 13   |

EcoStruxure IT пропонує надійний моніторинг та оповіщення в режимі реального часу. Він відстежує споживання енергії, температуру, вологість, продуктивність охолодження та стан мережі, надсилаючи сповіщення, коли будь-який показник виходить за межі визначених порогових значень, щоб ви могли вирішити проблеми, перш ніж вони спричинять простої. Що відрізняє EcoStruxure, так це його зосередженість на аналітиці та віддаленому управлінні. Він інтегрується з хмарним сховищем даних Schneider (EcoStruxure Data Lake), щоб надавати розширену аналітику та звіти про тенденції. Це допомагає операторам виявляти довгострокові тенденції потужності, прогнозувати, коли вам знадобиться більше енергії або охолодження, і навіть порівнювати продуктивність на різних об'єктах. Платформа підтримує віддалений доступ через безпечні хмарні шлюзи, що чудово підходить для розподілених центрів обробки даних або постачальників послуг колокейшн – ви можете контролювати кілька об'єктів з однієї консолі або мобільного додатка. Для організацій, які надають пріоритет оптимізації енергоспоживання, прогнозного обслуговування та інтеграції на рівні підприємства, EcoStruxure IT є провідним вибором DCIM у 2026 році.

### **Vertiv Trellis – оптимізація інфраструктури, орієнтована на енергоспоживання**

Vertiv Trellis – це платформа DCIM, розроблена Vertiv (раніше Emerson Network Power), компанією з глибоким корінням у системах живлення та охолодження для центрів обробки даних. Trellis спирається на досвід Vertiv, забезпечуючи комплексний моніторинг та контроль, особливо досягаючи успіхів у сфері управління живленням та екологічного контролю. Завдяки Trellis команди центрів обробки даних отримують у режимі реального часу інформацію про ключові показники, такі як споживання енергії на кожному рівні (від живлення об'єкта до окремих стійкових PDU), температура та вологість на поверсі, і навіть продуктивність повітряного потоку та охолодження. Панелі приладів та карти системи допомагають швидко виявляти гарячі точки або дисбаланс

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 14   |

навантаження, що дозволяє вносити проактивні корективи. Однією з сильних сторін Vertiv Trellis є детальне керування живленням. Він не тільки може контролювати споживання енергії, але й інтегрується з інтелектуальними стійковими блоками розподілу живлення (включаючи пристрої Vertiv Geist), що дозволяє дистанційно керувати розетками. Це означає, що оператори можуть виконувати такі дії, як дистанційне ввімкнення та вимкнення певного сервера для його перезавантаження або планування циклів увімкнення/вимкнення живлення для певного обладнання для економії енергії в періоди низького використання. Trellis може видавати сповіщення про будь-які порушення порогових значень (наприклад, стійка споживає занадто багато енергії або підвищується температура блоку CRAC), а також генерує детальні звіти про тенденції живлення та охолодження з плином часу. Ці аналітичні дані допомагають у плануванні потужностей та оптимізації енергоспоживання, що є надзвичайно важливим, оскільки багато центрів обробки даних у 2026 році стикаються з тиском на енергоефективність. Якщо пріоритетом управління вашим центром обробки даних є чітке управління системами живлення, охолодження та об'єктів разом з IT-обладнанням, Vertiv Trellis DCIM пропонує інтегрований спосіб зробити це з надійністю, яку ви очікуєте від спеціалізованого постачальника інфраструктури.

#### **Device42 – Комплексне управління активами з візуалізацією DCIM**

Device42 пропонує дещо інший погляд на DCIM, виходячи зі світу управління IT-активами (ITAM) та виявлення інфраструктури. Це потужна платформа, яка усуває розрив між IT-операціями та управлінням об'єктами. Device42 автоматично виявляє апаратне та програмне забезпечення у вашому середовищі – від серверів (фізичних та віртуальних) та мережевих комутаторів до програм та баз даних – створюючи живий перелік активів. Для використання в центрі обробки даних це означає, що ви завжди матимете актуальний запис про те, яке обладнання знаходиться в кожній стійці, що до чого підключено та як все це налаштовано.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.26.0055.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 15   |

Окрім цієї інвентаризації, Device42 надає багато функцій DCIM. Він створює візуальні карти вашого центру обробки даних : ви можете бачити схеми поверхів, висотні показники стійок і навіть схеми топології мережі, які генеруються на основі виявлених даних. Використовуючи зручний інтерфейс (з редагуванням перетягуванням), ви можете планувати зміни в схемах стійок або мережевих з'єднаннях і негайно розуміти їхній вплив. Device42 також включає моніторинг даних про навколишнє середовище та енергетичних навантажень у режимі реального часу, а також сповіщення про умови, які потребують уваги – тому це не просто статичний інструмент документування. Планування потужностей стає простішим, коли ви можете візуалізувати та шукати свої активи за такими атрибутами, як доступний простір у стійці або запас потужності.

Ще однією сферою, де Device42 сяє, є інтеграція та автоматизація. Він має розширені API та може надсилати або витягувати дані з інших систем, таких як CMDB, системи продажу квитків та інструменти моніторингу. Він навіть відстежує такі речі, як ліцензії на програмне забезпечення та IP-адреси, розмиваючи межу між DCIM та повним управлінням IT-інфраструктурою. Для організацій, які хочуть мати єдине джерело достовірної інформації як для фізичного центру обробки даних, так і для IT-систем, що працюють у ньому, Device42 є переконливим вибором. Він ефективно об'єднує управління активами, управління мережею та DCIM в одну платформу, що може значно спростити операції для IT-команд та команд, що тісно співпрацюють.

### **FNT Command – цифровий двійник та DCIM, керований робочим процесом**

FNT Command – це корпоративне рішення DCIM, відоме тим, що створює багатогранний «цифровий двійник» центру обробки даних. FNT, що походить з Німеччини, використовується телекомунікаційними компаніями та операторами великих центрів обробки даних по всьому світу для контролю над своєю інфраструктурою. За допомогою FNT Command ви моделюєте весь свій центр обробки даних, включаючи детальне розташування стійок, пристроїв, кабелів,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.26.0055.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 16   |

розподілу живлення та навіть площі підлоги, в централізованій базі даних. Ця цифрова репліка відображає фізичну конфігурацію, забезпечуючи потужну візуалізацію та планування. Наприклад, ви можете клацнути на стійці на плані поверху та детально переглянути, які сервери там розміщені, їхні мережеві канали зв'язку, споживання енергії та навіть побачити запас потужності для додавання додаткового обладнання.

Програмне забезпечення підтримує моніторинг показників навколишнього середовища та енергоспоживання в режимі реального часу, подібно до інших провідних інструментів DCIM, надсилаючи сповіщення про будь-які аномалії. Але FNT Command йде далі завдяки вбудованим робочим процесам та управлінню змінами. Воно дозволяє планувати переміщення/додавання/зміни та проводити аналіз впливу перед їх виконанням, щоб побачити, чи може переміщення сервера або додавання нового блейд-шасі, наприклад, перевантажити ланцюг живлення або порушити поріг охолодження. Оскільки FNT робить акцент на процесах, багато команд центрів обробки даних використовують його для впровадження стандартизованих процедур для робочих нарядів, забезпечуючи документування та відстеження змін у DCIM.

Інтеграція – ще один напрямок для FNT. Вона надає інтерфейси відкритого стандарту (наприклад, REST API) для підключення до інших систем управління, уникаючи прив'язки до постачальника. Це означає, що ви можете інтегрувати FNT зі своєю системою продажу квитків, системою управління мережею або навіть системою управління будівлею для досягнення єдиного уявлення. Загалом, FNT Command ідеально підходить для тих, хто хоче цілісний, процесно-орієнтований DCIM, де цифровий двійник об'єкта знаходиться в центрі планування та експлуатації. Вона особливо потужна в середовищах, де планування потужностей та мінімізація ризиків під час змін є головними пріоритетами.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 17   |

## CommScope iTRACS – 3D-візуалізація та уніфіковане керування

iTRACS (від CommScope) – це платформа DCIM, яка здобула свою репутацію завдяки насиченій 3D-візуалізації та єдиному підходу до управління інфраструктурою центру обробки даних. iTRACS забезпечує інтерактивне 3D-вигляд вашого центру обробки даних, ефективно створюючи цифрового двійника, подібного до підходу FNT, але з акцентом на графічне представлення. В iTRACS ви можете віртуально «прогулятися» своїм залом обробки даних, збільшити зображення стійок та візуалізувати, як обладнання розташоване та підключене. Це більше, ніж просто приємний вигляд – це допомагає операторам центрів обробки даних та менеджерам об'єктів візуально виявляти проблеми, такі як виявлення переповнених стійок, спостереження за блокуванням повітряного потоку або трасування підключення від сервера до комутатора.

Функціонально iTRACS охоплює всі основні бази DCIM: управління активами, моніторинг навколишнього середовища та живлення, управління потужностями та планування змін. Він об'єднує ці функції в одній платформі, а це означає, що вам не потрібні окремі модулі, скажімо, для відстеження живлення та активів – все інтегровано. Однією з переваг такого уніфікованого підходу є те, що коли ви розглядаєте зміни (наприклад, переміщення сервера), iTRACS може автоматично виявляти наслідки для різних доменів (чи досягнуть мережеві з'єднання? Чи достатньо охолодження в цій стійці? тощо). iTRACS відомий своєю нейтральністю до постачальників та здатністю обробляти гетерогенні середовища, поширені у великих центрах обробки даних.

CommScope, будучи провідним гравцем у сфері кабельної та мережевої інфраструктури, також гарантує, що iTRACS відмінно справляється з картуванням мережевих з'єднань і навіть інтегрується з інтелектуальними кабельними системами (imVision від CommScope часто поєднується з ним для відстеження портів у режимі реального часу). Для BIM-менеджера або інженера детальний візуальний контекст, який надає iTRACS, може бути надзвичайно цінним – це як мати постійно оновлювану 3D-модель об'єкта, яка відображає

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 18   |

поточний стан інфраструктури. У 2026 році, коли віддалена співпраця все ще важлива, наявність такого цифрового двійника, доступного для всіх зацікавлених сторін (ІТ, об'єктів, проектних груп), може спростити сеанси вирішення проблем і планування. Якщо вам потрібне візуальне управління та єдина платформа, iTRACS – це рішення для управління цифровими комунікаціями, яке варто розглянути.

### **Siemens Datacenter Clarity LC – моніторинг та моделювання високої чіткості**

Datacenter Clarity LC від Siemens – це пакет керування цифровими інтегрованими системами (DCIM), який переносить досвід Siemens в галузі автоматизації будівель у сферу центрів обробки даних. Ключовою особливістю Clarity LC є його середовище 3D-візуалізації високої чіткості. Подібно до деяких інших рішень, воно створює детальну модель фізичної та ІТ-інфраструктури центру обробки даних, але Siemens робить особливий акцент на інтеграції з системами будівель. Ця платформа може бути підключена до сенсорних мереж та систем управління будівлями (BMS), враховуючи досвід Siemens, щоб забезпечити справді цілісне уявлення про живлення, охолодження, безпеку та ІТ-обладнання об'єкта разом.

Clarity LC постійно контролює критичні параметри – споживання енергії, продуктивність охолоджувального блоку, температуру, рівень вологості, схеми потоку повітря – і представляє їх як у вигляді панелі інструментів, так і візуально на 3D-моделі. Наприклад, ви можете побачити теплове накладання на 3D-макеті, яке показує коливання температури в приміщенні в режимі реального часу. Оператори можуть встановлювати порогові значення та отримувати миттєві сповіщення, якщо, скажімо, блок CRAC виходить з ладу або температура стійки виходить за межі безпечного діапазону. Одним із найпотужніших аспектів Siemens Clarity є його можливості моделювання та прогнозування. Ви можете віртуально розмістити нове обладнання в моделі та змодельовати його вплив на навантаження живлення та розподіл охолодження, перш ніж фізично його

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.26.0055.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 19   |

встановлювати. Цей аналіз «що, якщо» допомагає командам уникнути неприємних сюрпризів, спочатку перевіряючи зміни відповідно до обмежень потужності та моделей потоку повітря.

З точки зору управління активами, Clarity LC веде детальний перелік обладнання з метаданими, такими як гарантійний та сервісний періоди, що допомагає в управлінні життєвим циклом. Він також підтримує керування кількома об'єктами, тому великі підприємства з кількома центрами обробки даних можуть керувати ними під одним дахом. З огляду на зосередженість Siemens на ефективності, Clarity LC, як правило, приваблює тих, хто хоче максимізувати продуктивність та енергоефективність своїх об'єктів. Це комплексне управління активами, особливо підходить для критично важливих об'єктів, де інтеграція із засобами контролю об'єктів та розширене планування мають першочергове значення.

#### **openDCIM – DCIM з відкритим вихідним кодом для гнучкості та економії коштів**

Для організацій, які віддають перевагу рішенням з відкритим кодом або мають бюджетні обмеження, openDCIM є популярним вибором. openDCIM – це безкоштовне програмне забезпечення DCIM, розроблене спільнотою, яке надає багато важливих функцій, необхідних для управління центром обробки даних, хоча й без деяких переваг комерційних продуктів. Однак не варто його недооцінювати – openDCIM охоплює основні потреби, такі як відстеження активів, керування живленням та мережевими підключеннями, а також моніторинг потужності. Користувачі можуть документувати всі свої стійки та пристрої в openDCIM, відстежувати, які активи знаходяться в яких стійках, та відображати підключення живлення (наприклад, як сервер підключається до PDU та вихідного джерела живлення), а також мережеві з'єднання між пристроями.

Інтерфейс openDCIM містить корисні візуальні засоби, такі як інтерактивні карти кімнат (ви можете завантажити зображення плану поверху та визначити на ньому зони/шафи) та базовий графічний вигляд стійки. Він також

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 20   |

надає накладання, які дозволяють візуалізувати показники потужності, такі як навантаження на живлення, простір та температура за зонами, що допомагає виявляти гарячі точки або недостатньо використаний простір. Хоча openDCIM може не мати вишуканої 3D-графіки, він певною мірою підтримує інтеграцію та автоматизацію. Він має RESTful API (у розробці) для отримання/передавання даних і навіть такі функції, як моделювання впливу для резервування живлення (наприклад, ви можете імітувати збій живлення та побачити, які пристрої будуть постраждали). Останні оновлення демонструють постійні вдосконалення, такі як покращена звітність, журнали аудиту та навіть можливість експорту даних для аналізу CFD (обчислювальна гідродинаміка).

Однією з найбільших переваг openDCIM, звичайно ж, є вартість (його можна використовувати безкоштовно за ліцензією GPL) та гнучкість налаштування, якщо у вас є власні розробники. Багато невеликих центрів обробки даних, лабораторій або навіть великих підприємств з кваліфікованими ІТ-командами використовують openDCIM, щоб уникнути прив'язки до постачальника. Спільнота навколо нього поширює плагіни та поради щодо розширення його функціональності. Якщо у вас є технічна підкованість для його підтримки, openDCIM може стати міцною основою для DCIM, яку ви можете адаптувати до своїх потреб. Він втілює «менталітет відкритого коду» спільної розробки, що може бути привабливим для команд, які хочуть мати повний контроль над своїм набором інструментів DCIM.

### **За межами традиційного DCIM: інтеграція дизайну та штучного інтелекту з ArchiLabs**

Хоча програмне забезпечення DCIM чудово підходить для операційного управління інфраструктурою центрів обробки даних, команди, що дивляться в майбутнє, у 2026 році також шукають способи пов'язати ці операційні дані з процесом проектування та планування. Саме тут на допомогу приходять такі платформи, як ArchiLabs. ArchiLabs – це операційна система на базі штучного інтелекту для проектування центрів обробки даних, яка об'єднує весь ваш

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 21   |

технологічний стек – від електронних таблиць та баз даних DCIM до інструментів CAD/BIM (таких як Revit та інші), аналітичного програмного забезпечення та навіть спеціальних власних додатків – в єдине, завжди синхронізоване джерело достовірної інформації. По суті, ArchiLabs діє як місток між інформацією про центр обробки даних у реальному часі, яку надає DCIM, та проектними моделями та документами, з якими працюють менеджери та інженери BIM.

Уявіть, що ви можете автоматично вносити найновішу інформацію про активи вашого DCIM (стійки, моделі обладнання, з'єднання портів тощо) до вашої BIM-моделі, гарантуючи, що плани проектування завжди відображають те, що насправді є на підлозі. ArchiLabs робить це можливим завдяки двонаправленій інтеграції (наприклад, така функція, як DCIM  $\rightleftharpoons$  Revit Sync, синхронізує висоту шаф у Revit з даними DCIM у режимі реального часу про U-позиції, силові блоки та зміни обладнання). Завдяки зв'язку моделей проектування та даних DCIM, архітектори та інженери можуть виявляти конфлікти на ранній стадії – наприклад, якщо під час операцій встановлюється нове обладнання, BIM-модель може оновлюватися, щоб відобразити його, зберігаючи єдине джерело достовірної інформації в усіх дисциплінах.

Що справді відрізняє ArchiLabs, так це використання штучного інтелекту для автоматизації повторюваних завдань планування в робочому процесі проектування центру обробки даних. За допомогою спеціалізованих агентів штучного інтелекту ArchiLabs ви можете автоматизувати такі завдання, як планування розташування стійок і рядів, проектування кабельних траєкторій та розміщення обладнання. Наприклад, ArchiLabs може зчитувати простий експорт запланованого обладнання в Excel або DCIM, а потім автоматично генерувати розташування стійок у вашій CAD-моделі, дотримуючись заданих вами правил проектування (подумайте про правила утримання гарячих/холодних проходів, вимоги до зазорів або обмеження щільності потужності на ряд). Він навіть може ефективно прокладати кабельні траєкторії, уникаючи конфліктів і дотримуючись стандартів, заощаджуючи дизайнерам незліченну кількість годин ручного

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 22   |

креслення. Ці «агенти» штучного інтелекту, по суті, є другими пілотами, яких ви навчаєте керувати робочими процесами в будь-якому програмному забезпеченні у вашому стеку. Потрібно отримати дані про навантаження на охолодження з інструменту аналізу та оновити електронну таблицю планування потужності? Або експортувати однолінійну електричну схему з вашої BIM-моделі в програму аналізу, таку як ETAP? ArchiLabs може автоматично оркеструвати ці багатоетапні процеси. Користувачі навчили агентів ArchiLabs зчитувати та записувати дані безпосередньо на різні платформи САПР (через API або за допомогою відкритих форматів, таких як IFC), взаємодіяти із зовнішніми базами даних та API, і навіть надсилати оновлення до інших корпоративних систем щоразу, коли відбуваються зміни в проекті.

Важливо, що ArchiLabs – це комплексна платформа, а не просто плагін для одного програмного забезпечення. Це не просто «автоматизація для Revit» – вона охоплює всю екосистему інструментів, які використовують команди центрів обробки даних. Підключаючись до вашого програмного забезпечення DCIM, ArchiLabs гарантує, що фаза планування (моделі BIM, схеми поверхів, плани потужностей) та фаза експлуатації (дані активів та датчиків у реальному часі) завжди синхронізовані. Це зменшує кількість помилок, спричинених застарілою інформацією, та пришвидшує виконання проекту, автоматизуючи виснажливі завдання координації. Для менеджерів BIM це означає більше часу, зосередженого на високоцінних інженерних рішеннях, замість ручного передавання даних або повторюваного креслення. Для операторів центрів обробки даних це означає, що цифрові моделі та документація постійно оновлюються та є точними, що безцінно для експлуатації, технічного обслуговування та майбутніх оновлень. Коротше кажучи, такі платформи, як ArchiLabs, представляють собою наступну еволюцію управління центром обробки даних, де штучний інтелект та інтеграція об'єднують проектування, будівництво та експлуатацію в один інтелектуальний робочий процес.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 23   |

## Висновок

Оскільки центри обробки даних продовжують масштабуватися та розвиватися у 2026 році, інструменти, які ми використовуємо для їх проектування та управління, повинні йти в ногу з часом. Найкращі програмні рішення DCIM, перелічені вище, пропонують надійні способи моніторингу інфраструктури, оптимізації потужності та запобігання простоям завдяки аналітиці на основі даних. Кожне з них має свою нішу – будь то зручні панелі інструментів, розширене керування живленням чи захопиви 3D-цифрові двійники – тому вибір правильного DCIM залежить від пріоритетів вашої організації. Однак одна спільна тенденція очевидна: інтеграція та єдине джерело достовірної інформації є найважливішими. BIM-менеджери, архітектори та інженери об'єктів все частіше співпрацюють, і поєднання DCIM з інструментами проектування (та використання платформ штучного інтелекту, таких як ArchiLabs) може значно посилити цю співпрацю. Об'єднуючи операційні дані в режимі реального часу з інтелектуальною автоматизацією в процесі проектування, команди можуть усунути ізолюваність та працювати ефективніше, ніж будь-коли. Результатом є центр обробки даних, який планується розумніше, працює плавніше та готовий відповідати зростаючим вимогам майбутнього. Незалежно від того, чи модернізуєте ви існуючий об'єкт, чи будujete нові центри обробки даних, інвестування в правильне програмне забезпечення DCIM – та подальший погляд на інтеграцію на основі штучного інтелекту – буде ключем до успіху в найближчі роки.

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

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 24   |

програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

### **Швидкість виконання коду Python**

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

### **Використання Python**

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 25   |

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.

### **Версії Python**

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з'являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп'ютері. Одна з основних функцій процесора – це обробка даних згідно комп'ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп'ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп'ютеру. За допомогою машинної мови

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 26   |

створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

### **Отримати**

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

### **Декодувати**

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

### **Виконати**

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 27   |

тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

### **Завантаження Python**

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

### **Середовище програмування для Python**

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 28   |

роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.

- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.

- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.

- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.

- Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

- Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

- PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

- Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

- Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи проектування ЦОД з використанням технологій кастомізації.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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;

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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 30   |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Хоча публічна хмара все ще надзвичайно важлива для бізнесу, вона явно не замінює центри обробки даних. З огляду на це, ось огляд 10 головних тенденцій центрів обробки даних на 2026 рік. Це 10 ключових прогнозів, які повинен знати кожен ІТ-лідер.

#### Швидке зростання гібридних та мультихмарних архітектур

Підприємства все частіше впроваджують гібридні моделі, які поєднують гнучкість публічної хмари з продуктивністю та економічною ефективністю приватних або локальних середовищ. Саме тому зрілі організації зараз вибірково переносять робочі навантаження, а не повністю відмовляються від хмари.

Ця тенденція важлива, оскільки гібридні архітектури забезпечують свободу розміщення робочих навантажень на основі витрат, продуктивності та потреб у дотриманні вимог. Лідери отримують більш детальний контроль над економікою інфраструктури та можуть уникнути прив'язки до певних послуг та змінних тарифів, що є поширеними у великомасштабних хмарних розгортаннях.

#### Збільшення хмарної репатріації для робочих навантажень з інтенсивним використанням даних

Організації, що використовують робочі навантаження з аналітики, машинного навчання, медичної візуалізації або фінансової звітності, часто стикаються з високими витратами на вихідні дані та проблемами з продуктивністю в середовищах публічної хмари. Ці робочі навантаження переміщують великі обсяги даних між службами, створюючи непередбачувані витрати.

Ця тенденція важлива, оскільки обсяги даних зростають швидше, ніж розвиваються моделі ціноутворення на хмарні послуги. Керівники повинні

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 31   |

розробляти інфраструктурні стратегії, які відповідають як короткостроковим потребам у продуктивності, так і довгостроковій бюджетній стабільності. Репатріація пропонує передбачувану продуктивність пропускну здатності та усуває плату за гігабайт, що може покращити загальну вартість володіння (ТСО) для великих наборів даних.

### **Зростаючий попит на колокейшн як альтернативу повноцінному локальному розміщенню**

Багато підприємств хочуть контролювати локальні середовища без тягаря управління фізичними об'єктами. Колокейшн пропонує виділене живлення, охолодження, підключення та фізичну безпеку, водночас дозволяючи компаніям розгортати власне обладнання. Це відображається в тому факті, що стратегії репатріації все більше покладаються на колокейшн як цільове середовище.

Ця тенденція важлива, оскільки вона дозволяє організаціям зменшити операційні витрати, зберігаючи при цьому архітектурну суверенність. Колокейшн також підтримує гібридні стратегії, надаючи високошвидкісні з'єднання з низькою затримкою для постачальників публічних хмарних послуг, що зменшує складність робочих процесів у кількох середовищах.

### **Більша увага до передбачуваної продуктивності критично важливих систем**

Мінливість продуктивності залишається поширеною проблемою в розгортанні публічних хмар через ефекти шумних сусідів та спільні шари віртуалізації. Непередбачувана затримка стала поширеною причиною, чому організації повертають такі програми, як бази даних, кластери аналітики та платформи ERP, до виділеної інфраструктури.

Ця тенденція важлива, оскільки продуктивність безпосередньо впливає на дохід, задоволеність клієнтів та безперервність роботи. Передбачувана продуктивність забезпечує послідовне дотримання угоди про рівень обслуговування (SLA), скорочує цикли усунення несправностей та підтримує програми зі суворими вимогами до часу реагування, такі як механізми

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 32   |

фінансових транзакцій або логістичні системи.

### **Розширені вимоги щодо дотримання вимог та прозорості аудиту**

Регульовані галузі, такі як охорона здоров'я, фінанси та уряд, стикаються з важким тягарем дотримання вимог у хмарі через модель спільної відповідальності. Як наслідок, хмарні розгортання часто потребують кількох додаткових інструментів для реєстрації, аудиту та шифрування, щоб відповідати таким стандартам, як HIPAA, PCI DSS та GDPR.

Ця тенденція важлива, оскільки дотримання вимог стає все суворішим, а не менш суворим. Керівникам потрібна інфраструктура, яка забезпечує повну видимість контролю безпеки та спрощує підготовку до аудиту. Репатріація та гібридні моделі дозволяють організаціям централізувати засоби контролю, зменшити фрагментацію інструментів та ефективніше демонструвати дотримання вимог.

### **Зростаюча важливість прямої, високоякісної технічної підтримки**

Постачальники публічних хмарних послуг часто покладаються на багаторівневі моделі підтримки, які надають пріоритет великим клієнтам. Обмежений та/або непрямий доступ до інженерів може затримати вирішення інцидентів і, отже, призвести до перебоїв, що тривають довше, ніж необхідно.

Ця тенденція важлива, оскільки ризик простою зростає зі складністю системи. Підприємствам потрібна оперативна підтримка критично важливих робочих навантажень, особливо тих, що пов'язані з клієнтськими послугами або регульованими операціями. Постачальники колокації та приватної хмари, що пропонують практичну підтримку, можуть скоротити середній час вирішення проблем та мінімізувати перебої в обслуговуванні.

### **Оптимізація інфраструктури у відповідь на інфляцію вартості хмарних технологій**

Тенденції ціноутворення на хмарні послуги демонструють зростання витрат на зберігання даних, передачу даних, мережеві послуги та передові обчислювальні ресурси. Багато організацій відзначають, що щорічне зростання

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 33   |

витрат на хмарні послуги випереджає зростання доходів. Тому непередбачуваність витрат стала основним фактором репатріації.

Ця тенденція має значення, оскільки інфляція витрат знижує рентабельність інвестицій у впровадження хмарних технологій. Керівники повинні оцінити, чи виправдовують робочі навантаження преміальне ціноутворення на хмарні технології, чи може виділена інфраструктура стабілізувати витрати. Стратегічне перенесення ресурсів та гібридні підходи покращують точність бюджетування та захищають рентабельність.

### **Ширше впровадження архітектур відкритих стандартів для уникнення прив'язки до певного рівня**

Багато проектів репатріації виявляють залежність від власних хмарних сервісів, таких як безсерверні функції, керовані бази даних та платформи штучного інтелекту. Після вбудовування ці залежності ускладнюють міграцію та обмежують стратегічну гнучкість. Організації можуть витрачати місяці на усунення прив'язки до постачальника під час проектів репатріації.

Ця тенденція важлива, оскільки архітектурна незалежність стає важливою для довгострокової стійкості. Впровадження відкритих стандартів, таких як контейнери, Kubernetes та бази даних з відкритим кодом, забезпечує мобільність робочого навантаження та зменшує залежність від одного постачальника.

### **Розширення вимог до периферійних та близьких до периферійних обчислень**

Додатки, що інтенсивно працюють з даними та чутливі до затримки, все частіше потребують обробки ближче до місць генерації даних. Регіони публічних хмар не завжди можуть відповідати цим вимогам щодо близькості, що робить периферійні середовища більш придатними.

Ця тенденція важлива, оскільки обробка з низькою затримкою підтримує системи управління виробництвом у режимі реального часу, візуалізацію в охороні здоров'я, автономні системи та аналітику роздрібної торгівлі. Репатріація в центри колокації, розташовані поблизу населених пунктів, дозволяє компаніям

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 34   |

працювати ближче до клієнтів та пристроїв, зберігаючи при цьому централізоване управління.

### **Збільшення інвестицій в інструменти автоматизації приватних хмар**

Організації, які впроваджують репатріацію або гібридну інфраструктуру, хочуть хмарної автоматизації, зберігаючи при цьому контроль над обладнанням. Інструменти для інфраструктури як коду, автоматизованого забезпечення та самостійного управління ресурсами тепер виходять за межі публічних хмарних середовищ.

Ця тенденція важлива, оскільки автоматизація приватної хмари зменшує операційне навантаження, покращує узгодженість розгортання та забезпечує швидше масштабування. Керівники можуть модернізувати локальні середовища без шкоди для ефективності чи гнучкості.

### **3.2 Розробка структурної схеми**

Нижче наведено 10 головних тенденцій, які формують планування, проектування та будівництво центрів обробки даних наступного покоління по всьому світу:

- Будівництво центру обробки даних, оптимізованого для штучного інтелекту.
- Будівництво центру обробки даних з рідинним охолодженням.
- Будівництво модульного центру обробки даних.
- Будівництво центрів обробки даних з обмеженим енергоспоживанням.
- Будівництво центрів обробки даних з використанням відновлюваних джерел енергії та інтегрованих систем BESS.
- Модернізація та модернізація будівництва центрів обробки даних.
- Будівництво центру обробки даних з використанням цифрового двійника та штучного інтелекту.
- Будівництво периферійних та розподілених центрів обробки даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 35   |

- Будівництво сталого та низьковуглецевого центру обробки даних.
- Будівництво швидкозростаючого регіонального центру обробки даних.

### **Об'єкти, готові до використання штучного інтелекту, стають стандартом**

Ера центрів обробки даних загального призначення відходить у минуле. У 2026 році більшість нових центрів обробки даних будуть спроектовані з урахуванням робочих навантажень штучного інтелекту, а не традиційних обчислень. Це означає вищу щільність стійок, компонування, орієнтовані на графічні процесори, перероблені системи живлення та теплові архітектури, здатні витримувати тривалі багатомегаватні робочі навантаження. Штучний інтелект більше не є варіантом використання; він є рушійною силою архітектури. До 2026 року більшість нових центрів обробки даних будуть розроблені з урахуванням робочих навантажень штучного інтелекту, а не загальних обчислень, а проекти гіпермасштабування призведуть до вищої щільності стійок, передових рішень для охолодження та енергоємних кластерів графічних процесорів.

### **Рідке охолодження стає мейнстрімом**

Оскільки стійки потужністю перевищують 50–100 кВт, а кластери графічних процесорів нагріваються, рідинне охолодження в центрах обробки даних стає новою нормою. Пряме охолодження на кристалі, теплообмінники на задніх дверцятах та занурювальні баки більше не є «експериментальними»; вони стали стандартними специфікаціями. Справжній зсув полягає не лише в ефективності, а й у необхідності. Повітря не може охолоджувати штучний інтелект у великих масштабах, але рідина може.

Стійки зі штучним інтелектом високої щільності спонукають все більше операторів до прямого охолодження на кристалі та іммерсійного охолодження, що робить рідинне охолодження стандартним елементом дизайну, а не преміальним оновленням, причому впровадження швидко прискорюється разом зі зростанням робочих навантажень штучного інтелекту та високопродуктивних обчислень.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.26.0055.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 36   |

## Будівництво модульних центрів обробки даних бере гору

Швидкість тепер є конкурентною перевагою. Модульне будівництво центрів обробки даних використовує заводські модулі, попередньо виготовлені блоки живлення та контейнерні білі простори для пришвидшення розгортання, дозволяючи операторам вводити нові потужності в експлуатацію за місяці, а не за роки.

У 2026 році кожен великий гіперскейлер впроваджує модульність для зниження ризиків, пришвидшення термінів та поступового масштабування потужностей, що відображає прогнозоване сильне зростання ринку модульних центрів обробки даних у найближчі роки.

## Влада стає найбільшим вузьким місцем

Нова земельна війна ведеться не за землю, а за енергію. У багатьох регіонах потужність мережі визначає, де можна будувати центри обробки даних, а де ні. 2026 рік змушує забудовників переглянути традиційний вибір місць та впроваджувати локальну генерацію, мікромережі, газові турбіни та великомасштабні акумуляторні системи лише для забезпечення надійної доступності електроенергії. Ринкові дослідження та галузеві офіційні документи висвітлюють обмеження в потужності та багаторічні черги як основний обмежувач нових поставок.

## Відновлювані джерела енергії на місці та BESS переносять мережеві робочі навантаження поза мережею

Сталий розвиток – це вже не відповідність вимогам, це потужність. Нові об'єкти все частіше поєднують сонячну, вітрову енергію та системи акумуляторного накопичення енергії (BESS) з основним джерелом живлення. Це допомагає зменшити залежність від напружених мереж і підтримує ESG-орієнтовані конструкції.

Центр обробки даних 2026 року буде частково електростанцією, частково обчислювальним центром. Галузеві дослідження та приклади використання операторами вказують на зростання впровадження BESS для вирішення проблем

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 37   |

затримок у мережі та забезпечення стійкості.

**Модернізація випереджає будівництво з нуля**

Замість того, щоб чекати три роки на новий об'єкт, оператори вважають, що модернізація старих центрів обробки даних є швидшою та економічно вигіднішою. У 2026 році відбудеться масштабна заміна залів з повітряним охолодженням на рідинні, стійки з вищою щільністю, нові системи ДБЖ та покращена електрична/теплова інфраструктура. Модернізація стає стратегією, а не компромісом. Кілька галузевих експертів зараз стверджують, що модернізація забезпечує швидше нарощування потужності та кращі результати сталого розвитку, ніж багато нових проєктів.

**Будівництво на основі штучного інтелекту та цифрові двійники беруть на себе виконання**

Проектування центру обробки даних у 2026 році означає використання цифрових двійників, імітаційних моделей та планування будівництва на основі штучного інтелекту. Це зменшує конфлікти, скорочує затримки, спрощує оцінку витрат та значно покращує готовність. Будівництво центрів обробки даних стає більш передбачуваним, автоматизованим та інтелектуальним, як і самі об'єкти. Академічні та галузеві звіти демонструють, що цифрові двійники зменшують втрати та час введення в експлуатацію.

**Центри обробки даних на периферії стають глобальними**

Раніше периферійні технології були нішевими. У 2026 році вони всюди. Інтернет речей, автономні системи, розумні міста, промислова автоматизація та штучний інтелект створюють попит на тисячі менших, розподілених об'єктів, готових до периферійних технологій. Більшість з них є модульними, збірними та мають надзвичайно низьку затримку. Великі центри обробки даних не зникають; вони стають частиною ширшої, більш розподіленої екосистеми. Прогнози ринку периферійних технологій показують швидке зростання до 2030 року, зумовлене застосунками з низькою затримкою.

## Сталий розвиток впливає на кожен долар капітальних витрат

Тепер оператори проектують з урахуванням вуглецевих бюджетів, а не лише фінансових. 2026 рік надає пріоритет низьковуглецевим матеріалам, ефективним електричним та охолоджувальним системам, інтеграції відновлюваних джерел енергії, методам циркулярного будівництва та кращим цілям PUE (коефіцієнта енергоефективності). Нормативи посилюються в усьому світі, а сталий розвиток став невід'ємною частиною будівництва центрів обробки даних, від проектування до введення в експлуатацію. Мандати щодо сталого розвитку та тиск інвесторів все частіше згадуються як рушійні сили проектування у звітах та заявах операторів.

Такі регіони, як Індія, Південно-Східна Азія, Африка та Латинська Америка, стають основними зонами зростання центрів обробки даних завдяки цифровій трансформації та місцевому попиту на хмарні технології.

Легкий аналіз ринку: Азіатсько-Тихоокеанський регіон демонструє одну з найвищих кривих зростання в будівництві центрів обробки даних у світі; місцеві закони про суверенітет даних прискорюють будівництво в регіонах; модульне розгортання допомагає обійти вузькі місця в інфраструктурі. Ринки, що розвиваються, не наздоганяють; вони випереджають сучасні, модульні, готові до штучного інтелекту конструкції. Дані регіонального ринку та галузеві огляди показують великі будівельні портфелі в Азіатсько-Тихоокеанському регіоні та зростання інвестицій у регіони, що розвиваються.

Центри обробки даних більше не є «допоміжним активом». Вони є критично важливою національною інфраструктурою, що формує енергетичну політику, міське планування, цифрові екосистеми та економічний розвиток.

План дій для будівництва центрів обробки даних на 2026 рік чіткий: будувати на основі штучного інтелекту, охолоджувати за допомогою рідин, розгортати модульні системи, використовувати гібридну та відновлювану енергію, масштабувати за допомогою периферії та проектувати для довгострокової стійкості. Організації, які приймуть цей перехід, побудують

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 39   |

найшвидшу, найчистішу та найперспективнішу цифрову інфраструктуру десятиліття.

## **SPB/TRILL**

Всередині дата-центру реалізується протокол SPB/TRILL.

Протокол Spanning Tree Protocol (STP), стандартизований у стандарті IEEE 802.1D, служив основним механізмом запобігання петель у мостових мережах Ethernet протягом понад трьох десятиліть. Незважаючи на широке поширення, STP демонструє значні обмеження щодо часу конвергенції, використання пропускної здатності та масштабованості, які дедалі більше несумісні з вимогами сучасних корпоративних мереж та мереж центрів обробки даних. У даній роботі був проведений аналіз п'яти основних протоколів другого рівня в лінійці STP: STP (IEEE 802.1D), Rapid Spanning Tree Protocol (RSTP, IEEE 802.1w), Multiple Spanning Tree Protocol (MSTP, IEEE 802.1s), Shortest Path Bridging (SPB, IEEE 802.1aq) та Transparent Interconnection of Lots of Links (TRILL, RFC 6325). Аналіз проводиться за чотирма основними вимірами: час конвергенції, масштабованість, експлуатаційна складність та міркування безпеки. Спираючись виключно на рецензовану літературу, стандарти IETF/IEEE та опубліковані операційні тести, робота завершується структурованою оцінкою актуальності цих протоколів у середовищах програмно-визначених мереж (SDN) та сучасних архітектурах Ethernet-матриць, пропонуючи концептуальну основу для прийняття рішень щодо вибору протоколу.

Розглянемо конкретні реалізації SPB/TRILL.

12900-е збирають за допомогою технології IRF у стек, що встановлюється в ядро мережі, на них же можна запустити й L3. Як комутатори доступу використовуються пристрої серії 5900, причому в цьому випадку LAGG не знадобиться, тому що за множинні шляхи L2 відповідає TRILL, хоча LAGG дозволяє скоротити кількість шляхів і весь домен буде сходитися швидше за рахунок спрощення загальної топології мережі.

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРБ-123.26.0055.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>40</b> |

З більших комутаторів TRILL підтримують пристрою серій 10500, 11900 і 12900; з комутаторів поменше, рівня ToS або ядра серверної, – серії 7900 і 5900. SPB підтримується на всіх перерахованих лінійках, крім 10500.

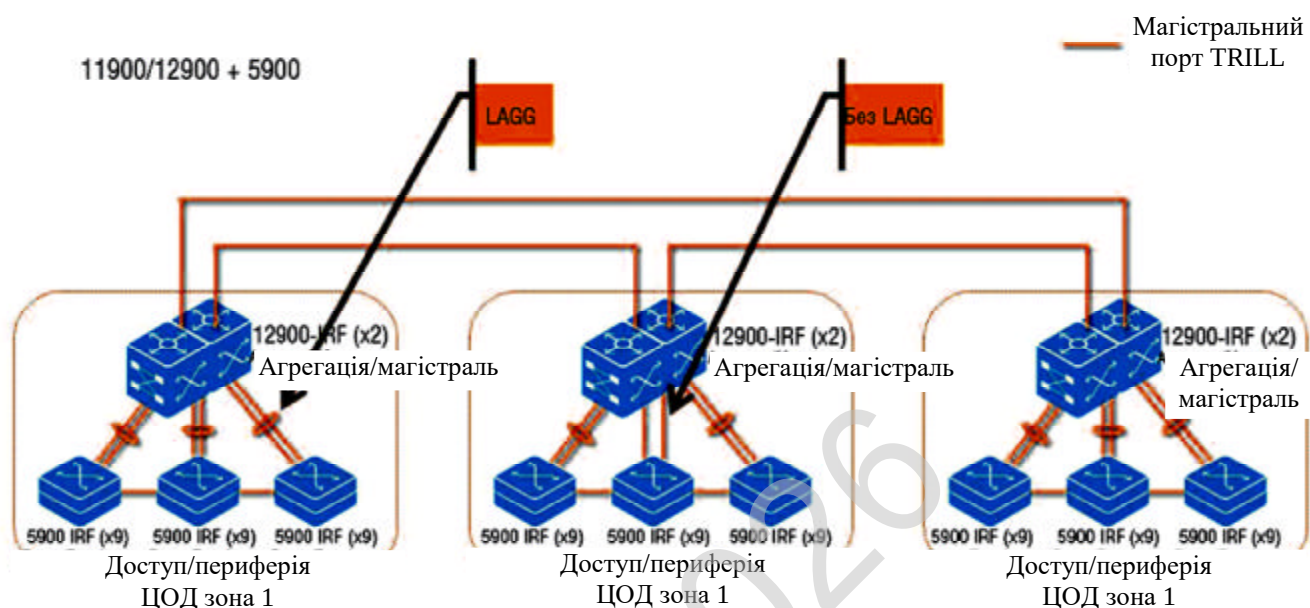


Рисунок 3.1 – Структурна схема системи

Одне із ключових питань, що виникає при проектуванні TRILL/SPB, складається в організації розвантаження L3 Offload – точки, у якій трафік L2 перетворюється в L3. Можливі три варіанти реалізації L3 Offload:

- організація зовнішньої петлі на пристрої;
- розвантаження усередині комутатора на мікросхемі, наприклад на 11900;
- використання Multitenant Device Context (MDC).

### 3.3 Розробка функціональної схеми

Перейдемо до розгляду функціональної схеми. Сформулюємо й вирішимо завдання аналізу функціональної структури системи проектування ЦОД з використанням технологій кастомізації. Для цього розроблена концепція

проведення аналізу, вирішені завдання формального опису структури й обчислення її характеристик: навантаження на канали зв'язку, вузли й структуроутворююче устаткування системи проектування ЦОД з використанням технологій кастомізації.

На рисунку 3.2 представлена функціональна схема системи, які відповідає запропонованому підходу. Основною метою аналізу структури є визначення параметрів потоків даних, що проходять по каналах зв'язку системи проектування ЦОД з використанням технологій кастомізації й вступні на вузли системи проектування ЦОД з використанням технологій кастомізації. Однак, завдання структури системи проектування ЦОД з використанням технологій кастомізації тільки як сукупності вузлів і зв'язків між ними, не дозволяє досліджувати потоки даних, оскільки потоки даних формуються розв'язуваними на системи проектування ЦОД з використанням технологій кастомізації завданнями, точніше додатками, які запускаються на вузлах системи проектування ЦОД з використанням технологій кастомізації й обмінюються між собою даними. Отже, для аналізу системи проектування ЦОД з використанням технологій кастомізації необхідно відомості про функціональну структуру доповнити відомостями про додатки і їхнє розміщення в системи проектування ЦОД з використанням технологій кастомізації.

Результатами аналізу повинні стати математичні залежності, що дозволяють обчислювати чисельні значення характеристик системи проектування ЦОД з використанням технологій кастомізації з урахуванням особливостей конкретної структури системи проектування ЦОД з використанням технологій кастомізації. Запропоновано концептуальний підхід до аналізу функціональної структури системи проектування ЦОД з використанням технологій кастомізації, заснований на дослідженні взаємодії додатків (завдань) як незалежних джерел і приймачів даних. У цьому випадку спочатку визначаються параметри потоків даних між додатками при виконанні всього комплексу завдань (будується інформаційна модель), а потім, залежно від розміщення додатків по вузлах системи проектування ЦОД з використанням технологій кастомізації й використовуюваного устаткування, визначаються параметри потоків даних між

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 42   |

вузлами системи проектування ЦОД з використанням технологій кастомізації (будується технічна модель). При цьому не тільки повністю враховуються всі взаємодії між додатками, але й з'являється можливість проведення аналізу складних ієрархічних мережних структур, шляхом декомпозиції на підмережі системи проектування ЦОД з використанням технологій кастомізації, що часто застосовується в технологіях VLAN і VPN. Даний підхід розвивається й застосовується в даній роботі. Відзначимо, що отримані результати аналізу потоків даних, можна надалі використовувати для проведення більше глибоких досліджень із застосуванням відомих методів і моделей, наприклад, СеМО.

Для аналізу функціональної структури й розрахунку характеристик системи проектування ЦОД з використанням технологій кастомізації визначені правила її формального опису, що однозначно задають властивості додатків і структуру системи проектування ЦОД з використанням технологій кастомізації, що дозволяють будувати моделі для проведення розрахунків. Відповідно до концептуального підходу до проведення аналізу функціональної структури системи проектування ЦОД з використанням технологій кастомізації виділені дві складові її структури інформаційна й технічна:

– Інформаційна структура визначає інформаційні потоки між інформаційними вузлами, на яких установлене програмне забезпечення й представляє сукупність вузлів і інформаційних ресурсів, розміщених на цих вузлах. Маючи у своєму розпорядженні дані про інформаційну структуру системи проектування ЦОД з використанням технологій кастомізації можна приймати рішення про організацію каналів зв'язку між вузлами системи проектування ЦОД з використанням технологій кастомізації, визначати необхідні параметри телекомунікаційної системи, формувати структуру системи проектування ЦОД з використанням технологій кастомізації.

– Технічна структура – сукупність структуроутворюючого устаткування, технічних вузлів системи проектування ЦОД з використанням технологій кастомізації й каналів зв'язку. Технічному вузлу можуть відповідати декілька інформаційних.

Таким чином, для повноцінного аналізу функціональної структури

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 43   |

реальної системи проектування ЦОД з використанням технологій кастомізації необхідно провести аналіз складових її інформаційної й технічної структур і зв'язати результати аналізу.

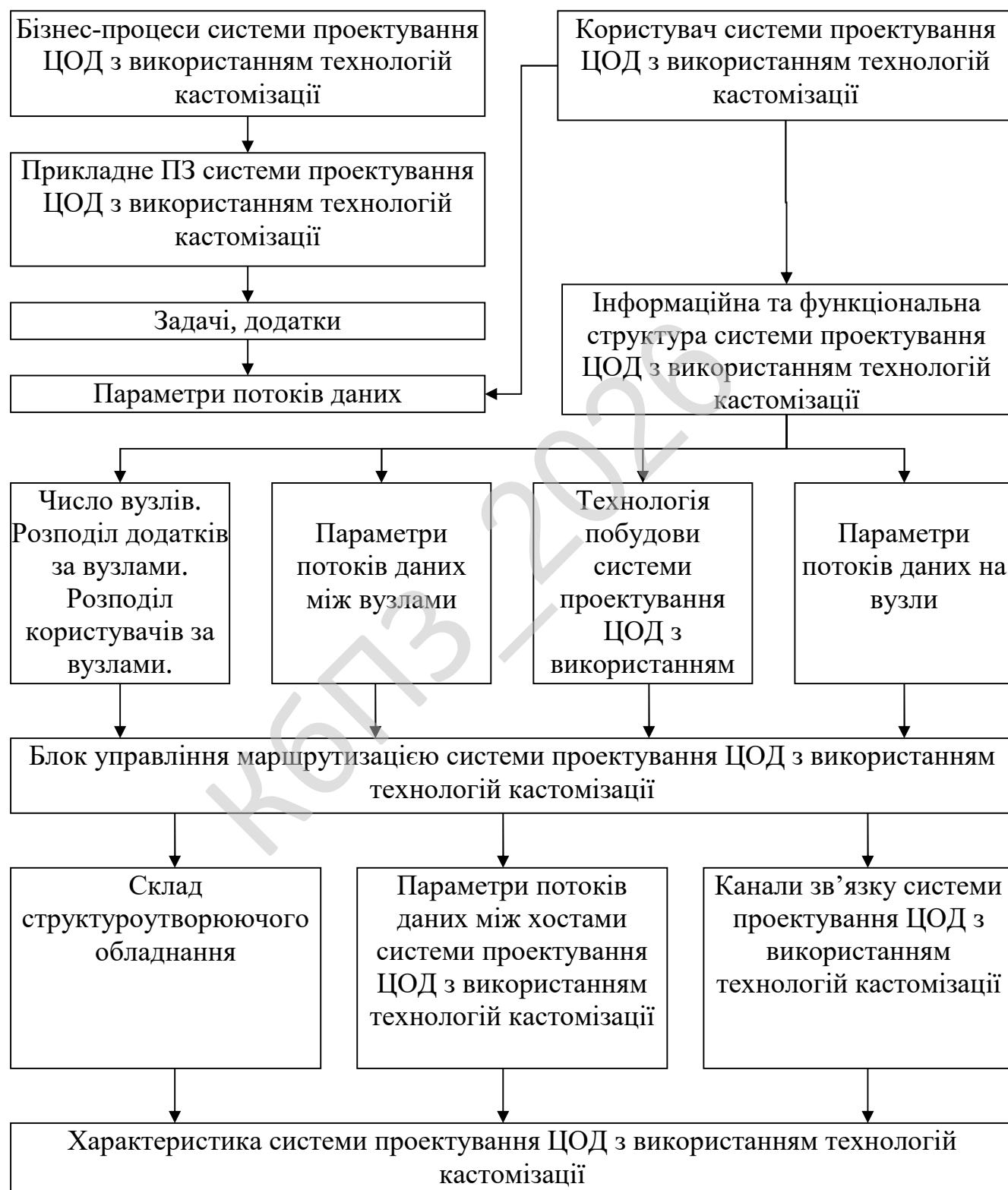


Рисунок 3.2 – Функціональна схема системи

Зв'язування результатів аналізу інформаційної й технічної структур має на увазі відображення характеристик інформаційної структури в характеристики технічної структури й визначення параметрів технічної структури на основі параметрів і характеристик інформаційної структури.

Інформаційна структура системи проектування ЦОД з використанням технологій кастомізації задається набором параметрів:

$$SI = \{N, M, D, L, R, S_k (k=1, 2, \dots, L), A_{km} (k=1, 2, \dots, L; m=1, 2, \dots, D), G, H, S\},$$

де:

- $N$  – число працюючих користувачів;
- $M$  – число задіяних вузлів;
- $D$  – число використовуваних додатків;
- $L$  – число розв'язуваних завдань;
- $R$  – число баз даних;
- $S_k = \{p_k, d_k, u_k, W_k\}$ , ( $k=1, 2, \dots, L$ ) – набір даних, що описують

розв'язувані завдання, де:

- $p_k = (p_{k1}, p_{k2}, \dots, p_{kD})$ , – вектор-рядок, що визначає завданням  $k$  додатка, що запускаються;

- $d_k = (d_{k1}, d_{k2}, \dots, d_{kR})$  – вектор-рядок, що визначає використовувані завданням  $k$  бази даних (сховища даних);

- $u_k = (u_{k1}, u_{k2}, \dots, u_{kN})$ , – вектор-рядок, що визначає завдання  $k$  користувачів, що запускають  $W_k = \|w_{kij}\|$  ( $i=1, 2, \dots, D$ ;  $j=1, 2, \dots, D$ ) – матриця, що встановлює послідовність запуску додатків завданням  $k$ ;

- $A_{km} = \{v_{km}, b_{km}\}$ , ( $k=1, 2, \dots, L$ ;  $m=1, 2, \dots, D$ ) – набір даних, що описують додатки, використовувані завданнями, де:

- $v_{km} = (v_{km1}, v_{km2}, \dots, v_{kmR})$ , – вектор-рядок, що задає обсяги даних, якими обмінюється додаток  $m$  з базами даних, при рішенні завдання  $k$ ;

- $\mathbf{b}_{km} = (b_{km1}, b_{km2}, \dots, b_{kmD})$  – вектор-рядок, що задає обсяги даних, якими обмінюється додаток  $m$  з іншими додатками, при рішенні завдання  $k$ ;
- $\mathbf{G}$  матриця розміщення додатків по вузлах;
- $\mathbf{H}$  – матриця підключення користувачів до вузлів;
- $\mathbf{S}$  – матриця розміщення баз даних по вузлах.

Набір однозначно визначає інформаційну структуру системи проектування ЦОД з використанням технологій кастомізації.

Для заданого набору  $\mathbf{SI}$  отримані результати, що дозволяють визначити параметри потоків даних між вузлами системи проектування ЦОД з використанням технологій кастомізації.

При цьому використовувалася матриця інтенсивностей потоків запитів користувачів на запуск завдань  $\Lambda = \|\lambda_{ij}\|$ ,  $(i=1,2,\dots,N; j=1,2,\dots,L)$ .

Показано, що матриця інтенсивностей потоків даних між вузлами системи проектування ЦОД з використанням технологій кастомізації при рішенні завдання  $k$  обчислюється за формулою:  $\Lambda_k = \lambda_k \mathbf{Z}_k$ ,  $(k=1,2,\dots,L)$ . , де  $\mathbf{Z}_k$  – матриця обсягів даних, переданих між вузлами, при рішенні завдання. Обчислені також матриця інтенсивностей потоку запитів на запуск додатка номер  $j$ , на вузлі номер  $i$  –  $\mathbf{B}^*$  і матриця сумарної інтенсивності потоку запитів до бази даних номер  $j$ , на вузлі номер  $i$  –  $\Phi$ .

Таким чином, визначена множина параметрів потоків даних інформаційної структури системи проектування ЦОД з використанням технологій кастомізації:  $\mathbf{PSI}(\mathbf{SI}) = \{\Lambda, \mathbf{Z}_k, \Lambda_k (k=1,2,\dots,L), \mathbf{A}, \mathbf{B}^*, \Phi\}$ .

Розроблено моделі для аналізу ієрархічної трьохрівневої інформаційної структури системи проектування ЦОД з використанням технологій кастомізації, що дозволяють визначити завантаження каналів зв'язку й мережного устаткування в системі проектування ЦОД з використанням технологій кастомізації, що складається із сукупності підмереж і корпоративних серверів.

Отримано наступні результати:

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 46   |

– матриці інтенсивностей потоків даних між групами й усередині груп кожного рівня  $A_1(C_1) = C_1 A(C_1)^T$ ,  $A_2(C_2) = C_2 [A_1(C_1) - dg(A_1(C_1))](C_2)^T$ ;

– матриці інтенсивностей потоків даних утворених кожним завданням  $A_{1k}(C_1)$  і  $A_{2k}(C_1)$ , ( $A = \sum_{k=1}^L A_k$ ).

Досліджено властивості матриць, що відбивають взаємозв'язки потоків даних окремих завдань, сформульовані у вигляді доведених тверджень.

Як міра ефективності ієрархічної структури запропоновані коефіцієнти поглинання інтенсивностей потоків даних на кожному рівні. Коефіцієнт поглинання на кожному рівні відбиває частку сумарної інтенсивності потоків, що локалізується усередині рівня.

Результати аналізу інформаційної структури дозволяють визначати параметри потоків даних між логічними об'єднаннями вузлів системи проектування ЦОД з використанням технологій кастомізації. Інформаційна структура реалізується конкретними технічними засобами й втілюється у вигляді технічної структури. При цьому можливо невідповідність інформаційної й технічної структур, пов'язане з технічними характеристиками й можливостями апаратури. Наприклад, на одному технічному вузлі можуть бути встановлені кілька інформаційних вузлів. У зв'язку із цим розроблені методи й засоби аналізу технічної структури системи проектування ЦОД з використанням технологій кастомізації, створюваної на базі інформаційної структури.

Для аналізу роботи реальної системи проектування ЦОД з використанням технологій кастомізації розроблений метод формального опису її технічної структури. Структура реальної системи проектування ЦОД з використанням технологій кастомізації формується із застосуванням структуроутворюючого устаткування (комутатори, маршрутизатори), до якого підключаються вузли системи проектування ЦОД з використанням технологій кастомізації, при цьому створюється багаторівнева (часто трьохрівнева) мережа. Такий підхід застосовується, як правило, при створенні системи проектування ЦОД з

використанням технологій кастомізації на базі технології VLAN. При цьому групи першого рівня інформаційної структури становлять віртуальні локальні системи проектування ЦОД з використанням технологій кастомізації. Другий і третій рівні інформаційної структури призначені для з'єднання цих мереж між собою.

Число комутаторів, використовуваних для з'єднання вузлів технічної структури системи проектування ЦОД з використанням технологій кастомізації при створенні груп першого рівня (комутатори першого рівня), визначається особливостями реальної системи проектування ЦОД з використанням технологій кастомізації, технічними можливостями комутаторів. Позначимо це число  $K_1^*$ , ( $K_1^* \geq K_1 \geq 1$ ). Число комутаторів, для з'єднання комутаторів першого рівня й створення груп другого рівня (комутатори другого рівня) –  $K_2^*$ , ( $K_2^* \geq K_2 \geq 1$ ). Число комутаторів третього рівня для з'єднання комутаторів другого рівня –  $K_3^* \geq 0$ .

Технічна структура системи проектування ЦОД з використанням технологій кастомізації задається множиною **ST**, елементами якого є:

–  $Y_1^* = \|y_{1ij}^*\|$ , ( $i = 1, 2, \dots, M$ ;  $j = 1, 2, \dots, K_1^*$ ) матриця з'єднань технічних вузлів системи проектування ЦОД з використанням технологій кастомізації (робітники станції, сервери) з комутаторами першого рівня;

–  $Y_2^* = \|y_{2ij}^*\|$ , ( $i = 1, 2, \dots, K_1^*$ ;  $j = 1, 2, \dots, K_2^*$ ) матриця з'єднань комутаторів першого рівня з комутаторами другого рівня;

–  $Y_3^* = \|y_{3ij}^*\|$ , ( $i = 1, 2, \dots, K_2^*$ ;  $j = 1, 2, \dots, K_3^*$ ) матриця з'єднань комутаторів третього рівня з комутаторами другого рівня;

–  $X_1^* = \|x_{1ij}^*\|$ , ( $i, j = 1, 2, \dots, K_1^*$ ) матриця з'єднань комутаторів першого рівня;

–  $X_2^* = \|x_{2ij}^*\|$ , ( $i, j = 1, 2, \dots, K_2^*$ ) матриця з'єднань комутаторів другого

рівня;

–  $\mathbf{X}_3^* = \|x_{3ij}^*\|$ ,  $(i, j = 1, 2, \dots, K_3^*)$  матриця з'єднань комутаторів третього

рівня.

Для кожного рівня задані матриці, що задають пропускні здатності каналів зв'язку, використовуваних при побудові мереж:

$$\mathbf{C}_1^*(\mathbf{Y}_1^*), \mathbf{C}_2^*(\mathbf{Y}_2^*), \mathbf{C}_3^*(\mathbf{Y}_3^*), \mathbf{C}_1^*(\mathbf{X}_1^*), \mathbf{C}_2^*(\mathbf{X}_2^*), \mathbf{C}_3^*(\mathbf{X}_3^*).$$

Для заданої множини  $\mathbf{ST}$  отримані наступні результати:

– матриця інтенсивностей інформаційних потоків, між комутаторами першого рівня й усередині груп технічних вузлів, підключених до комутаторів першого рівня:  $\mathbf{A}_1^*(\mathbf{Y}_1^*) = (\mathbf{Y}_1^*)^T \mathbf{A} \mathbf{Y}_1^*$ ;

– матриця сумарних інтенсивностей інформаційних потоків для комутаторів другого рівня:  $\mathbf{A}_2^*(\mathbf{Y}_2^*) = (\mathbf{Y}_2^*)^T [\mathbf{A}_1^*(\mathbf{Y}_1^*) - dg(\mathbf{A}_1^*(\mathbf{Y}_1^*))] \mathbf{Y}_2^*$ ;

– матриця інтенсивностей інформаційних потоків, між і усередині комутаторів третього рівня:  $\mathbf{A}_3^*(\mathbf{Y}_3^*) = (\mathbf{Y}_3^*)^T [\mathbf{A}_2^*(\mathbf{Y}_2^*) - dg(\mathbf{A}_2^*(\mathbf{Y}_2^*))] \mathbf{Y}_3^*$ ;

– вектори інтенсивностей інформаційних потоків, що надходять на комутатори всіх рівнів:  $\lambda_1^* = (\lambda_{11}^*, \lambda_{12}^*, \dots, \lambda_{1K_1^*}^*)$ ,  $\lambda_2^* = (\lambda_{21}^*, \lambda_{22}^*, \dots, \lambda_{2K_2^*}^*)$ ,  $\lambda_3^* = (\lambda_{31}^*, \lambda_{32}^*, \dots, \lambda_{3K_3^*}^*)$ ;

– вектор сумарних інтенсивностей потоків даних, переданих по каналах зв'язку  $\gamma_1^* = (\gamma_{11}^*, \gamma_{12}^*, \dots, \gamma_{1M}^*)$ ,  $(i = 1, 2, \dots, M)$ .

Отримані також формули для обчислення параметрів потоків даних кожного завдання.

Оскільки в корпоративних мережах великої розмірності використовується маршрутизація, то для обчислення параметрів потоків необхідно застосовувати відповідні алгоритми маршрутизації на графах. Для цього побудована матриця графа зв'язків між комутаторами  $\Omega$ . Показано, що матриця  $\Omega$  може бути представлена у вигляді:

$$\Omega = \begin{pmatrix} \mathbf{X}_1^* & \mathbf{Y}_2^* & \mathbf{0} \\ (\mathbf{Y}_2^*)^T & \mathbf{X}_2^* & \mathbf{Y}_3^* \\ \mathbf{0} & (\mathbf{Y}_3^*)^T & \mathbf{X}_3^* \end{pmatrix}.$$

Таким чином, параметри потоків даних у системи проектування ЦОД з використанням технологій кастомізації для заданої технічної структури визначаються множиною:

$$\mathbf{PST}(\mathbf{ST}) = \{\mathbf{A}_1^*(\mathbf{Y}_1^*), \mathbf{A}_2^*(\mathbf{Y}_2^*), \mathbf{A}_3^*(\mathbf{Y}_3^*), \lambda_1^*, \lambda_2^*, \lambda_3^*, \gamma_1^*, \Omega\}.$$

Множина  $\mathbf{PST}(\mathbf{ST})$  визначає параметри, як сумарних, так і часток потоків даних для окремих завдань (додатків), переданих по каналах зв'язку.

Якість рішення завдань залежить від того, яка частина пропускної здатності каналу (смуга) виділена для кожного завдання, що звичайно досягається шляхом застосування режиму гарантованої якості обслуговування (QoS). Тому при рішенні завдань розрахунку потоків із застосуванням систем з гарантованою якістю обслуговування проводиться розширення множини  $\mathbf{PST}(\mathbf{ST})$  шляхом додавання множини коефіцієнтів поділу каналів –  $\mathbf{PKST}$ . Це дозволяє визначити залежність характеристик системи проектування ЦОД з використанням технологій кастомізації від смуги пропускання в каналі (комутаторі), що відводиться для завдання.

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

### 3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврської роботи, наведена на рисунку 3.3.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 50   |

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.



Рисунок 3.3 – Діаграма взаємодії процесів

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

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.

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

## 4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

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

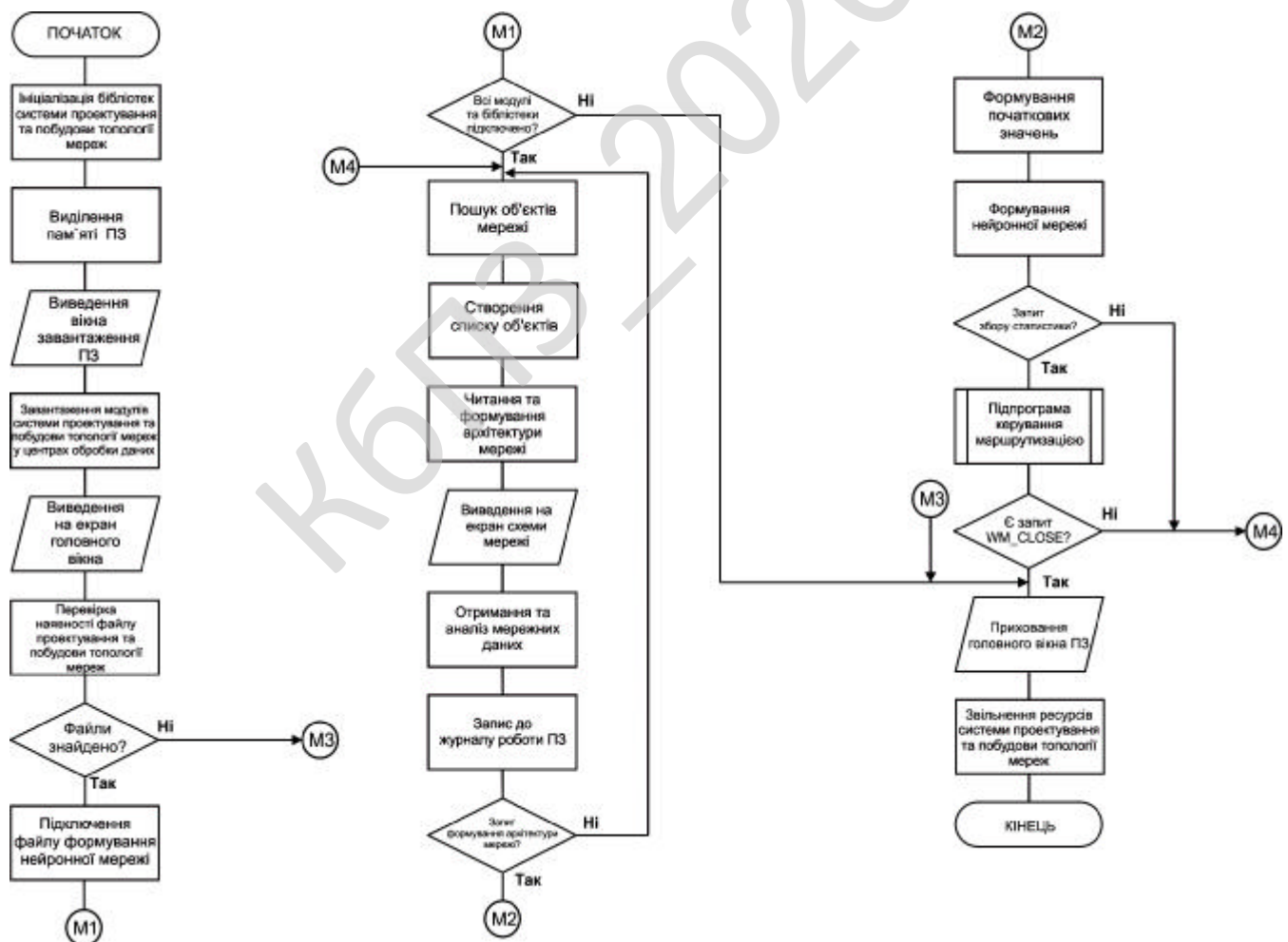


Рисунок 4.1 – Блок-схема основної програми

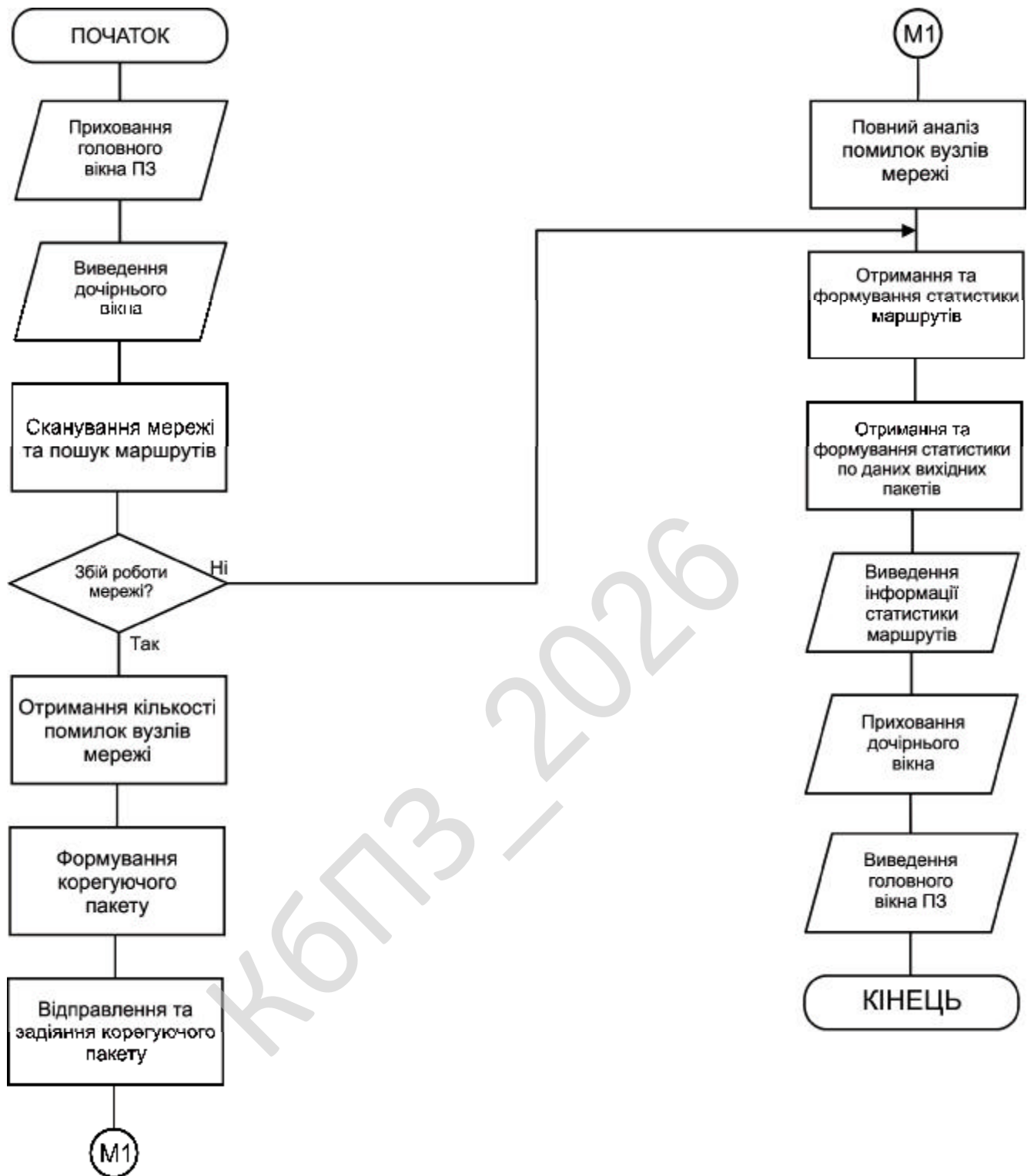


Рисунок 4.2 – Блок-схема роботи підпрограми

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 – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної 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 програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи проектування ЦОД з використанням технологій кастомізації.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може

дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Також при розробці бакалаврської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 55   |

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилалися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 57   |

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. \*), "одиниця або більше" (1 .. \*). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. \*, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовани ромбиком.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 58   |

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

### Опис системи

Програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації призначається для створення індивідуальної конфігурації центру обробки даних відповідно до вимог замовника, площі приміщення, кількості серверних шаф, енергоспоживання, резервування живлення, охолодження та мережевої інфраструктури.

Система працює як прикладне програмне забезпечення на Python. Користувач вводить параметри майбутнього ЦОД, після чого програма виконує розрахунок базових ресурсів, формує технічну конфігурацію, перевіряє обмеження та створює короткий звіт. Кастомізація реалізується через вибір рівня

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 59   |

надійності, типу шаф, кількості серверів, схеми живлення, охолодження та мережевого підключення.

### Архітектура системи

Система має модульну архітектуру. Основу становить модуль введення даних, модуль розрахунків, модуль перевірки конфігурації, модуль формування рекомендацій та модуль звітності. Дані зберігаються у внутрішніх структурах Python, а результати виводяться у вигляді текстового звіту.

### Основні модулі

1. Модуль введення параметрів приймає дані про площу, кількість шаф, серверів та рівень резервування.
2. Модуль розрахунку навантаження визначає сумарну потужність обладнання.
3. Модуль охолодження оцінює необхідну холодопродуктивність.
4. Модуль мережі формує базову кількість комутаторів і портів.
5. Модуль перевірки аналізує відповідність конфігурації мінімальним вимогам та модуль звіту формує підсумковий опис спроектованого ЦОД.

### Основні функції системи у вихідному коді

Система дозволяє створювати кастомізований проект ЦОД, оцінювати енергоспоживання, визначати потребу в охолодженні, розраховувати кількість мережевих портів, перевіряти резервування та формувати рекомендації щодо покращення конфігурації.

Розглянемо частину розробленого вихідного коду.

```
from dataclasses import dataclass
from typing import List
```

```
@dataclass
class DataCenterConfig:
    area_m2: float
    racks_count: int
    servers_per_rack: int
    server_power_w: int
    redundancy_level: str
    network_ports_per_server: int
```

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 60   |

```

@dataclass
class DataCenterReport:
    total_servers: int
    total_power_kw: float
    cooling_required_kw: float
    network_ports: int
    switches_required: int
    recommendations: List[str]

#Опис функціоналу
#Клас виконує базове проектування ЦОД за параметрами користувача

class DataCenterDesigner:
    def __init__(self, config: DataCenterConfig):
        self.config = config

    def calculate_total_servers(self) -> int:
        return self.config.racks_count * self.config.servers_per_rack

    def calculate_power(self) -> float:
        total_servers = self.calculate_total_servers()
        base_power_w = total_servers * self.config.server_power_w
        if self.config.redundancy_level == "N+1":
            base_power_w *= 1.15
        elif self.config.redundancy_level == "2N":
            base_power_w *= 2
        return round(base_power_w / 1000, 2)

    def calculate_cooling(self) -> float:
        power_kw = self.calculate_power()
        cooling_reserve = 1.25
        return round(power_kw * cooling_reserve, 2)

    def calculate_network_ports(self) -> int:
        total_servers = self.calculate_total_servers()
        return total_servers * self.config.network_ports_per_server

    def calculate_switches(self) -> int:
        ports = self.calculate_network_ports()
        ports_per_switch = 48
        return (ports + ports_per_switch - 1) // ports_per_switch

    def validate_configuration(self) -> List[str]:
        recommendations = []

        if self.config.area_m2 < self.config.racks_count * 2:
            recommendations.append("Площа приміщення є недостатньою для зручного розміщення шаф")

        if self.config.server_power_w > 800:
            recommendations.append("Потрібно передбачити посилене охолодження для високопродуктивних серверів")

```

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 61   |

```

        if self.config.redundancy_level not in ["N", "N+1", "2N"]:
            recommendations.append("Потрібно обрати коректний рівень
резервування")

        if self.config.network_ports_per_server < 2:
            recommendations.append("Доцільно використовувати не менше двох
мережевих портів на сервер")

        if not recommendations:
            recommendations.append("Конфігурація відповідає базовим вимогам
проектування ЦОД")

        return recommendations

def generate_report(self) -> DataCenterReport:
    return DataCenterReport(
        total_servers=self.calculate_total_servers(),
        total_power_kw=self.calculate_power(),
        cooling_required_kw=self.calculate_cooling(),
        network_ports=self.calculate_network_ports(),
        switches_required=self.calculate_switches(),
        recommendations=self.validate_configuration()
    )

#Опис функціоналу
#Функція формує текстовий звіт для пояснювальної записки
def print_report(report: DataCenterReport) -> None:
    print("Звіт проектування ЦОД")
    print(f"Кількість серверів: {report.total_servers}")
    print(f"Сумарне енергоспоживання кВт: {report.total_power_kw}")
    print(f"Необхідна потужність охолодження кВт:
{report.cooling_required_kw}")
    print(f"Кількість мережевих портів: {report.network_ports}")
    print(f"Необхідна кількість комутаторів: {report.switches_required}")
    print("Рекомендації")
    for item in report.recommendations:
        print(f"* {item}")

#Опис функціоналу
#Приклад запуску системи з тестовими параметрами
def main() -> None:
    config = DataCenterConfig(
        area_m2=80,
        racks_count=12,
        servers_per_rack=20,
        server_power_w=450,
        redundancy_level="N+1",
        network_ports_per_server=2

```

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 62   |

```

)
designer = DataCenterDesigner(config)
report = designer.generate_report()
print_report(report)
if __name__ == "__main__":
    main()

```

## 4.2 Захист розробленого програмного забезпечення

Дані у програмному забезпеченні я захищаю за допомогою NTRU. NTRUEncrypt (абревіатура Nth-degree TRUncated polynomial ring або Number Theorists aRe Us) – це криптографічна система з відкритим ключем, що раніше називалася NTRU.

Криптосистема NTRUEncrypt, заснована на ґратчастій криптосистемі, створена як альтернатива RSA і криптосистемам на еліптичних кривих (ECC). Стійкість алгоритму забезпечується труднощами пошуку найкоротшого вектора ґрати, що більше стійка до атак, здійснюваним на квантових комп'ютерах. На відміну від своїх конкурентів RSA, ECC, Elgamal, алгоритм використовує операції над кільцем:

$$\mathbb{Z}[X]/(X^N - 1),$$

усічених багаточленів ступеня, що не перевершує  $N - 1$ :

$$\mathbf{a}(X) = \mathbf{a} = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}.$$

Такий багаточлен можна також представити вектором:

$$\vec{a}(X) = \vec{a} = \sum_{i=0}^{N-1} a_i X^i = [a_0, a_1, a_2, \dots, a_{N-2}, a_{N-1}].$$

Як і будь-який молодий алгоритм, NTRUEncrypt погано вивчений, хоча й був офіційно затверджений для використання в сфері фінансів комітетом Accredited Standards Committee X9.[1]

Існує реалізації NTRUEncrypt з відкритим вихідним кодом.[2]

NTRUEncrypt, що споконвічно називався NTRU, був винайдений в 1996 році й представлений на конференціях CRYPTO, Конференція RSA, Eurocrypt.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 63   |

Причиною, що послужила початком розробки алгоритму в 1994 році, стала стаття [3], у якій говорилося про легкість злому існуючих алгоритмів на квантових комп'ютерах, які, як показало час, не за горами[4]. У цьому ж році, математики Jeffrey Hoffstein, Jill Pipher і Joseph H. Silverman, що розробили систему разом із засновником компанії NTRU Cryptosystems, Inc. (пізніше перейменованої в SecurityInnovation), Даніелем Ліманом (Daniel Lieman) запатентували свій винахід.[5]

### Кільця усічених багаточленів

NTRU оперує над багаточленами ступеня не переважаючої  $N - 1$ :

$$a = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1},$$

де коефіцієнти  $a_0, \dots, a_{N-1}$  – цілі числа. Щодо операцій додавання й множення за модулем багаточлена  $X^N - 1$ . Такі багаточлени утворюють кільце  $R$ , називане кільцем усічених багаточленів, що ізоморфно кільцю відносин:

$$\mathbb{Z}[X]/(X^N - 1).$$

NTRU використовує кільце усічених багаточленів  $R$  разом з діленням за модулем на взаємно прості числа  $p$  і  $q$  для зменшення коефіцієнтів багаточленів.

У роботі алгоритму також використовуються зворотні багаточлени в кільці усічених багаточленів. Слід зазначити, що не всякий багаточлен має зворотний, але якщо зворотний поліном існує, то його легко обчислити.[6][7]

### Генерація відкритого ключа

Для передачі повідомлення від Аліси до Боба необхідні відкритий і закритий ключі. Відкритий знають як Боб, так і Аліса, закритий ключ знає тільки Боб, що він використовує для генерації відкритого ключа. Для цього Боб вибирає два «маленьких» поліноми  $f$  і  $g$  з  $R$ . «Малість» поліномів мається на увазі в тому розумінні, що він маленький щодо довільного полінома за модулем  $q$ : у довільному поліномі коефіцієнти повинні бути приблизно рівномірно розподілені за модулем  $q$ , а в малому поліномі вони багато менше  $q$ . Малість поліномів визначається за допомогою чисел  $df$  і  $dg$ :

Поліном  $f$  має  $df$  коефіцієнтів рівних «1» і  $df-1$  коефіцієнтів рівних «-1», а інші – «0». У цьому випадку говорять, що:

$$\mathbf{f} \in \mathcal{L}_f$$

Поліном  $g$  має  $dg$  коефіцієнтів рівних «1» і стільки ж рівних «-1», інші – «0». У цьому випадку говорять, що:

$$\mathbf{g} \in \mathcal{L}_g$$

Причина, по якій поліноми вибираються саме таким чином, полягає в тому, що  $f$ , можливо, буде мати зворотний, а  $g$  – однозначно немає ( $g(1) = 0$ , а нульовий елемент не має зворотного).

Боб повинен зберігати ці поліноми в секреті. Далі Боб обчислює зворотні поліноми  $\mathbf{f}_p$  й  $\mathbf{f}_q$ , тобто такі, що:

$$\mathbf{f} \cdot \mathbf{f}_p \equiv 1 \pmod{p} \quad \text{й} \quad \mathbf{f} \cdot \mathbf{f}_q \equiv 1 \pmod{q}$$

Якщо  $f$  не має зворотного полінома, то Боб вибирає інший поліном  $f$ .

Секретний ключ – це пари  $(\mathbf{f}, \mathbf{f}_p)$ , а відкритий ключ  $h$  обчислюється за формулою:

$$\mathbf{h} = (p\mathbf{f}_q \cdot \mathbf{g}) \pmod{q}.$$

Приклад:

Для приклада візьмемо  $df=4$ , а  $dg=3$ . Тоді як поліноми можна вибрати

$$\mathbf{f} = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$

$$\mathbf{g} = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}$$

Далі для полінома  $f$  шукаються зворотні поліноми за модулем  $p=3$  й  $q=32$ :

$$\mathbf{f}_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9$$

$$\mathbf{f}_q = 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7 + 20X^8 + 18X^9 + 30X^{10}$$

Заключним етапом є обчислення самого відкритого ключа  $h$ :

$$\mathbf{h} = (p\mathbf{f}_q \cdot \mathbf{g}) \pmod{32} = 8 + 25X + 22X^2 + 20X^3 + 12X^4 + 24X^5 + 15X^6 + 19X^7 + 12X^8 + 19X^9 + 16X^{10}.$$

### Шифрування

Тепер, коли в Аліси є відкритий ключ, вона може відправити зашифроване повідомлення Бобові. Для цього повідомлення представити у вигляді полінома  $m$  з коефіцієнтами за модулем  $p$ , обраними з діапазону  $(-p/2, p/2]$ . Тобто

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 65   |

$m$  є «малим» поліномом за модулем  $q$ . Далі Алісі необхідно вибрати інший «малий» поліном  $g$ , що називається «сліпучої», обумовлений за допомогою числа  $dr$ :

Поліном  $g$  має  $dr$  коефіцієнтів рівних «1» і стільки ж рівних «-1», інші – «0». У цьому випадку говорять, що:

$$g \in \mathcal{L}_r.$$

Використовуючи ці поліноми, зашифроване повідомлення виходить за формулою:

$$e = (r \cdot h + m) \bmod q.$$

При цьому кожній, хто знає (або може обчислити) осліплюючий поліном  $g$ , зможе прочитати повідомлення  $m$ .

Приклад:

Припустимо, що Аліса хоче послати повідомлення, представлене у вигляді полінома:

$$m = -1 + X^3 - X^4 - X^8 + X^9 + X^{10},$$

і вибрала «осліплюючий» поліном, для якого  $dr=3$ :

$$r = -1 + X^2 + X^3 + X^4 - X^5 - X^7.$$

Тоді шифротекст  $e$ , готовий для передачі Бобові буде:

$$e = (r \cdot h + m) \bmod 32 = 14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 + 7X^7 + 25X^8 + 6X^9 + 19X^{10}$$

### Розшифрування

Тепер, одержавши зашифроване повідомлення  $e$ , Боб може його розшифрувати, використовуючи свій секретний ключ. Спочатку він одержує новий проміжний поліном:

$$a = (f \cdot e) \bmod q.$$

Якщо розписати шифротекст, то одержимо ланцюжок:

$$a = (f \cdot e) \bmod q = (f \cdot (r \cdot h + m)) \bmod q = (f \cdot (r \cdot pf_q \cdot g + m)) \bmod q$$

і остаточно:

$$a = (pr \cdot g + f \cdot m) \bmod q.$$

Після того, як Боб обчислив поліном  $a$  за модулем  $q$ , він повинен вибрати його коефіцієнти з діапазону  $(-q/2, q/2]$  і далі обчислити поліном  $b$ , одержуваний з полінома  $a$  приведенням за модулем  $p$ :

$$\mathbf{b} = \mathbf{a} \bmod p = (\mathbf{f} \cdot \mathbf{m}) \bmod p,$$

так як:

$$(\mathbf{p} \mathbf{r} \cdot \mathbf{g}) \bmod p = 0.$$

Тепер, використовуючи другу половину секретного ключа й отриманий поліном  $b$ , Боб може розшифрувати повідомлення:

$$\mathbf{c} = (\mathbf{f}_p \cdot \mathbf{b}) \bmod p.$$

Неважко бачити, що:

$$\mathbf{c} \equiv \mathbf{f}_p \cdot \mathbf{f} \cdot \mathbf{m} \equiv \mathbf{m} \pmod{p}.$$

У такий спосіб отриманий поліном  $c$  дійсно є вихідним повідомленням  $m$ .

Приклад:

Боб одержав від Аліси шифроване повідомлення  $e$ :

$$\mathbf{e} = 14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 + 7X^7 + 25X^8 + 6X^9 + 19X^{10}$$

Використовуючи секретний ключ  $f$  Боб одержує поліном  $a$ :

$\mathbf{a} = \mathbf{f} \cdot \mathbf{e} \pmod{32} = 3 - 7X - 10X^2 - 11X^3 + 10X^4 + 7X^5 + 6X^6 + 7X^7 + 5X^8 - 3X^9 - 7X^{10} \pmod{32}$ ,  
з коефіцієнтами, що належать проміжку  $(-q/2, q/2]$ . Далі перетворить поліном  $a$  у поліном  $b$ , зменшуючи коефіцієнти за модулем  $p$ .

$$\mathbf{b} = \mathbf{a} \pmod{3} = -X - X^2 + X^3 + X^4 + X^5 + X^7 - X^8 - X^{10} \pmod{3}$$

Заключний крок – перемножування полінома  $b$  із другою половиною закритого ключа  $\mathbf{f}_p$ :

$$\mathbf{c} = \mathbf{f}_p \cdot \mathbf{b} = \mathbf{f}_p \cdot \mathbf{f} \cdot \mathbf{m} \pmod{3} = \mathbf{m} \pmod{3}$$

$$\mathbf{c} = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}$$

Який є вихідним повідомленням, що передавала Аліса.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 67   |

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання бакалаврської роботи. Розроблене програмне забезпечення системи проектування ЦОД з використанням технологій кастомізації складається з наступних функціональних блоків:

- Навігаційне меню: Формат; Налаштування; Довідка.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші манипулятора миші.

- Функціональних кнопок ПЗ: Пошук об'єктів мережі; Статистика; База даних статичної інформації мережі; Перегляд сформованої мережі; Налаштування фільтру; Вікно прогнозування завантаженості мережі; Параметри потоків даних; Побудова карти мережі.

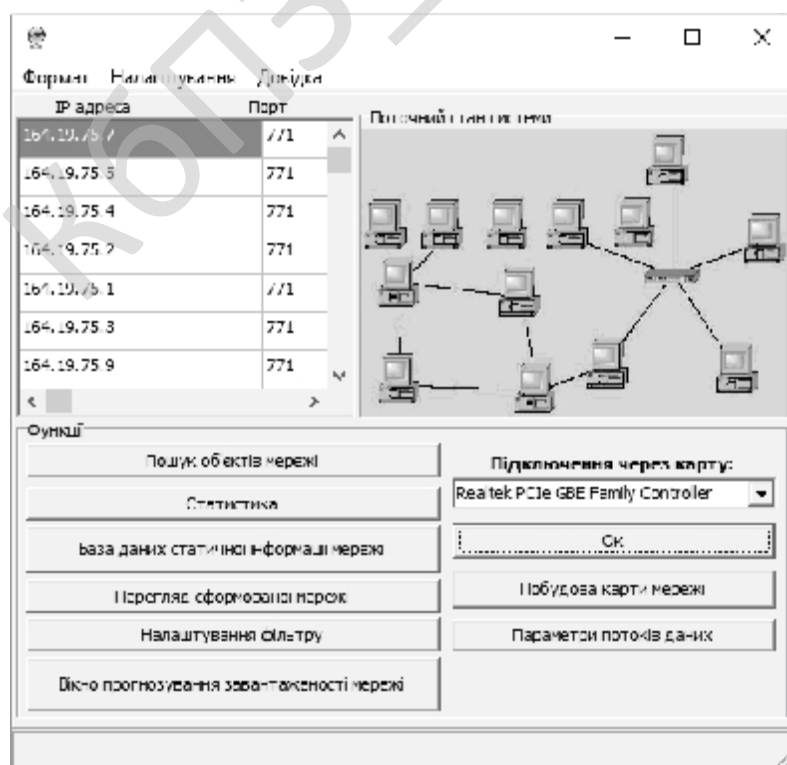


Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

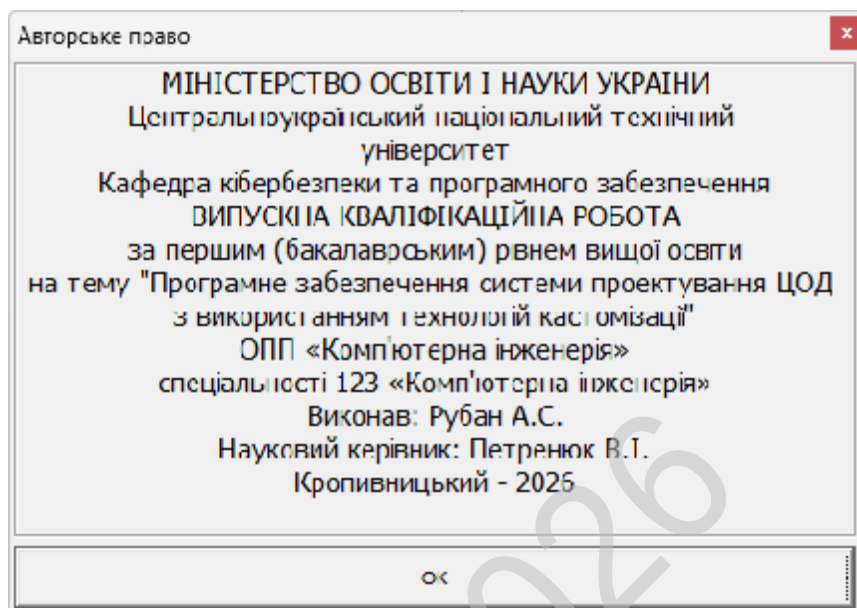


Рисунок 5.2 – Вікно розробника ПЗ

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 білої скриньки та чорної скриньки.

Тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Проводилось тестування 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.
- 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», чію 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;
- 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 – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи проектування ЦОД з використанням технологій кастомізації.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем проектування ЦОД з використанням технологій кастомізації.

- Досліджена система проектування ЦОД з використанням технологій кастомізації.

- На основі отриманих результатів досліджень створена програмна реалізація системи проектування ЦОД з використанням технологій кастомізації.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання проектування ЦОД з використанням технологій кастомізації.

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.

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 призначені для системи проектування ЦОД з використанням технологій кастомізації. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 73   |

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 NTRU.

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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 74   |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп’ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка»* (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
5. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
6. 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.
7. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.
8. 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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 75   |

9. 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.

10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

11. 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.

12. 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.

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. 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.

15. 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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 76   |

16. 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.

17. 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.

18. 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.

19. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 78   |

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

37. 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.

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

39. 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.

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

41. 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.

42. 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.

43. 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).

44. 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

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.26.0055.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 80   |

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

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

47. 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.

48. 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.

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

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". - Випуск 5 (142). - Х.: ХУПС - 2016. - С. 148-152.

51. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

52. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових

праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. - 2016. - С. 121-127.

53. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. - Випуск 2 (46) - Х.: ХУПС - 2016. - С. 146-149.

54. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

55. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

56. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

57. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

58. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

59. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник. – Кіровоград: КНТУ 2012. – 454 с