

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

На правах рукопису

Мороз Владислав Віталійович

**Програмне забезпечення системи керування гібридною ІТ-
інфраструктурою**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітній ступінь: бакалавр

Науковий керівник:

Доренський Олександр Павлович

(підпис)

(дата)

кандидат технічних наук

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри

_____ О.А. Смірнов

(підпис)

ПШ

« _____ » 2021 р.

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
О.А.Смірнов
« 11 » січня 2021 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Морозу Владиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи керування гібридною IT-інфраструктурою

керівник роботи Доренський Олександр Павлович, канд. техн. наук

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

затверджені наказом вищого навчального закладу № 204-02 від 28.12.2020 року

2. Строк подання студентом роботи до захисту 22.05.2021 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: Метою розробки є програмне забезпечення системи керування гібридною IT-інфраструктурою

4. 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

5. 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

6. Дата видачі завдання « 11 » січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	14.05.2021 р.	

Студент _____

(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____

(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Мороз В.В. Програмне забезпечення системи керування гібридною ІТ-інфраструктурою. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи керування гібридною ІТ-інфраструктурою.

Метою розробки є програмне забезпечення системи керування гібридною ІТ-інфраструктурою.

Результат роботи – програмна реалізація системи керування гібридною ІТ-інфраструктурою.

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 архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерна інженерія, система керування, гібридна ІТ-інфраструктура

ABSTRACT

Moroz V.V. Hybrid IT infrastructure management system software. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification, software has been developed that is designed for a hybrid IT infrastructure management system.

The purpose of the development is the software of the hybrid IT infrastructure management system.

The result is a software implementation of a hybrid IT infrastructure management system.

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

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the Delphi 10.4 environment.

Keywords: computer engineering, control system, hybrid IT infrastructure

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	23
2.3 Розгорнута постановка завдання	29
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	31
3.1 Опис функціонування системи.....	31
3.2 Розробка структурної схеми	33
3.3 Розробка функціональної схеми.....	37
3.4 Розробка діаграми процесів.....	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	42
4.2 Захист розробленого програмного забезпечення	53
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	56
6 ОСНОВНІ ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60

КБР-123.21.0036.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Мороз В.В.			Програмне забезпечення системи керування гібридною ІТ-інфраструктурою	Лім.	Аркуш	Аркушів
Перев.		Доренський О.П.				Б	1	68
Н.контр.		Гермак В.С.			ЦНТУ КІ-18-3СК			
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ПЗ	–	Програмне забезпечення
ТСО	–	Загальна вартість володіння
ЦОД	–	Центр обробки даних
AIOps	–	Artificial Intelligence for IT Operations. Штучний інтелект для керування ІТ на базі багатошарової платформи
CI/CD	–	безперервне вдосконалювання, інтеграція й впровадження основних ІТ функцій
ІТ	–	Інформаційні технології
HCI	–	Гіперконвергентна інфраструктура
SDS	–	Програмно-обумовлені сховища даних
STIG	–	VxRail Security Technical Implementation Guide
VDI	–	Virtual Desktop Infrastructure, віртуалізація серверів, баз даних, бізнес-застосунків
WSSD	–	Windows Server Software Defined
UCS	–	Unified Computing System

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Цифрова трансформація поєднує в собі використання нових технологій і їх впровадження за короткий проміжок часу. Це вимагає концентрації зусиль на написання, доробку нових застосунків, скорочення часу впровадження нових релізів, взаємодія різних застосунків через API інтерфейси, використання машинних агентів і Інтернету речей. Нові підходи й технології, недоступні раніше, роблять традиційні стратегії й методи керування IT відстаючими й не придатними для швидкої трансформації.

Аналітики пророкують, що ми побачимо значні зміни в поточних методиках і практиках ефективного управління гібридними IT-інфраструктурами. І ключ до цих змін – нова платформа, яку Gartner називає AIOps. І яка повинна відповісти на виклики нових технологій.

AIOps розшифровується як Artificial Intelligence for IT Operations. По суті це штучний інтелект для керування IT на базі багат шарової платформи, який автоматизує обробку даних і ухвалення рішення за допомогою машинного навчання й аналітики великих даних, які приходять із різних елементів IT-інфраструктури в режимі реального часу.

AIOps складається із двох основних компонентів: «великі дані» і «машинне навчання». Таким чином, IT фахівці повинні відійти від логування й відстеження безлічі окремих подій (siloe IT), які активно використовуються зараз, а покластися на машинне навчання й аналіз даних, які приходять від систем моніторингу, журналів нарядів на роботи і т.д.

Бажаний результат – безперервний аналіз, який дасть відповіді й дозволить реалізувати безперервні поліпшення й виправлення в роботі IT-інфраструктури. Платформа AIOps зв'язує три дисципліни – керування послугами, керування продуктивністю й автоматизацію – для досягнення

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

бажаного результату й може розглядатися як безперервне вдосконалювання, інтеграція й впровадження (CI/CD) основних ІТ функцій.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи керування гібридною ІТ-інфраструктурою.

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

- Огляд існуючих систем керування гібридною ІТ-інфраструктурою.
- Дослідження системи керування гібридною ІТ-інфраструктурою.
- Програмна реалізація системи керування гібридною ІТ-інфраструктурою.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування гібридною ІТ-інфраструктурою.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування гібридною ІТ-інфраструктурою, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

AIOps це нова платформа, яка безперечно наступний еволюційний крок у керуванні ІТ на який впливають наступні потреби:

– Складність керування в ручному режимі сучасною інфраструктурою. Сучасне ІТ-середовище включає керовані й не керовані хмари, послуги третіх осіб, різні мобільні застосунки й технології і т.д. Це постійно мінливий організм і традиційні підходи до керування не можуть працювати в такому динамічному середовищі. Відстеження всіх подій і керування цим середовищем без залучення машинного навчання й великих даних неможливо.

– Обсяг даних необхідних ІТ-службі експоненційно росте. Моніторинг продуктивності ІТ-середовища приводить до значного збільшення кількості подій і оповіщень, що веде до неможливості створення звітів у ручному або напівавтоматичному режимі.

– Проблеми з ІТ середовищем повинні вирішуватися швидко. Компанії з використанням ІТ роблять гроші й ІТ це не принтери, картриджі й текстові редактори – це важлива складова бізнес середовища. Індустрія споживання змінила очікування користувачів. Усе потрібно тут і зараз, тому всі проблеми повинні вирішуватися, тому що прямо впливають на продуктивність компанії.

– Усе більше й більше ресурсів переноситься за межі будинку, ядра свого корпоративного середовища. Хмарні технології, оренда інфраструктури як сервіс, сервіси від третіх осіб давно розмило границі й вносить проблеми в контролі над розподіленою інфраструктурою, яка часто не твоя, а когось іншого й для вибудовування діалогу з ними потрібні дані.

– Розроблювачі застосунків і сервісів на коні й мають більше влади й впливу, але відповідальність за роботу однаково лежить на ІТ. Нові технології

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

DevOps і Agile змушують програмістів побрати більше контрольних функцій на прикладному рівні, але відповідальність за загальний стан екосистеми й взаємодія між застосунками, сервісами й інфраструктурою як і раніше залишається під ІТ. ІТ-служба бере на себе більшу відповідальність, тому що мережі стають усе більш складними.

1.2 Область застосування

У наші дні практично все може бути адаптоване відповідно до потреб, у тому числі ІТ-інфраструктура вашого підприємства.

От чому ви чи навряд почуєте фразу: «Ви можете розміщати свої робочі навантаження де завгодно, поки вони перебувають у хмарі».

Тому як правда в тому, що розміщення 100% робочих навантажень у хмарному середовищі має позитивний ефект тільки для деяких вибраних організацій. Звичайно, зворотна сторона також вірна. Варіант розміщення 100% робочих навантажень у локальному середовищі – також не самий оптимальний. Тому найбільш ефективна стратегія розподілу робочого навантаження перебуває десь посередині.

Гібридна інфраструктура дозволяє підприємствам легко адаптувати варіанти розміщення робочих навантажень на основі обсягів роботи й застосунків. Це означає, що деякі робочі навантаження можуть залишатися в локальному середовищі, а інші розміщатися в хмарі. Можливість індивідуального налаштування й швидкої адаптації має величезні переваги для бізнес-процесів і бізнесу в цілому. Наприклад, критично важливі робочі навантаження або ті навантаження, які пов'язані з конфіденційними даними, можуть залишатися в локальному середовищі, у той час як інші застосунки, такі як електронна пошта або застосунки по обміну файлами, можуть розміщатися в хмарі. Можливість регулювати розміщення робочих навантажень не тільки підвищує продуктивність, але також може бути більш раціональною й ефективною

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

економічно. Поки у вас є надійна стратегія розміщення робочих навантажень, ви можете пожинати плоди гібридної ІТ-інфраструктури.

Стратегія відіграє ключову роль

Стратегія має вирішальне значення, незалежно від того, чи відіграєте ви в «Монополію», готуетесь до великої гри або думаєте над тем, де розмістити свої робочі навантаження. Насправді, як відзначила компанія IDC у випущеному недавно технічному документі «Workload Placement Separates the Winners from the Losers in IT», правильна стратегія розміщення робочого навантаження може розділити переможців, що й програли. Гарна стратегія адаптується до конкретних потреб вашого підприємства, і те, що підійде однієї компанії, може не підійти вашої.

Навіть у рамках однієї галузі різні компанії мають свої унікальні потреби. Їх визначає безліч факторів: розмір компанії, її місцезнаходження, галузь, життєвий цикл робочого навантаження, пріоритети і т.д. Саме тому не існує універсальної структури, яка підходила б для всіх. Правильна відповідь, імовірно, містить у собі комбінація локальної й хмарної інфраструктури. Але як визначити їхню правильну комбінацію? Це питання на мільйон доларів. Усі не так просто, як видається.

Розробка ефективної стратегії розміщення робочих навантажень – одна з найскладніших завдань, що коштують сьогодні перед лідерами ІТ-індустрії. Необхідно зважити незліченну безліч варіантів і врахувати ризики. Однак час, витрачене на дослідження й розробку надійної стратегії, має вирішальне значення для успіху вашої компанії.

Мати вибір – це добре, чи не так? Більшість людей погодяться, що чим більше варіантів, тем краще. Але насправді занадто велика кількість варіантів може привести в замішання й викликати сумніву в зробленому виборі (згадаєте, як ви з тугою дивилися на чийсь їжу, шкодуючи про власний вибір).

Така невизначеність ще раз указує на важливість вибору чітко певної стратегії розміщення робочих навантажень. Чим більше пошуків і аналізу було

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

проведено, тим більше ви будете впевнені в обраному варіанті розміщення своїх робочих навантажень. Але часто компанії ухвалюють рішення без ретельного розгляду всіх наявних варіантів. Наприклад, переміщення 100% робочих навантажень у хмарне середовище може здатися привабливим, але багато компаній на власному досвіді довідаються, які неприємні наслідки це може мати. Рішення піти ва-банк може привести до збільшення витрат або виникненню погрози безпеки в майбутньому. Не варто поміщати все в хмарне середовище тільки тому, що це спрацювало для когось іншого, і сподіватися, що це спрацює й для вас. Для деяких робочих навантажень це можливо. Але можуть бути й інші, які краще залишити в локальному середовищі. І краще з'ясувати це раніше, чим потім відшкодувати витрати. Компанія IDC відзначає: «Самі просунуті лідери ІТ-індустрії розуміють, що переклад усієї ІТ-інфраструктури підприємства в публічну хмару позбавляє їхній можливості модифікувати свою стратегію й адаптуватися до прискорюваних темпів зміни робочих процесів».

У свіжому технічному документі «Перехід стратегії підприємства від хмарної технології Cloud First до технології Cloud If» компанія Moor Insights & Strategy затверджує, що «принцип хмарних технологій повинен ґрунтуватися на технології Cloud If, але не те хнології Cloud Always». Це пояснює, що публічна хмара є інструментом, а не стратегією. «Нехай ваші вимоги до робочих навантажень визначають використання хмарних технологій, але не дозволяйте хмарному божевіллю лагодити перешкоди вашої ІТ-діяльності», говориться в документі.

Публічна хмара не є еліксиром, який розв'яже всі існуючі проблеми в області інформаційних технологій, і не задовольнить усі потреби, які будуть виникати в майбутньому. Як і в будь -якому інформаційному просторі, йти ва-банк завжди ризиковано.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування гібридною ІТ-інфраструктурою, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- Містить засоби для реплікації даних, резервного копіювання й зберігання на рівні хмари.
- Надає сервіси керування даними, забезпечує відказостійкість і якість обслуговування корпоративного рівня.
- Єдина точка підтримки для програмного й апаратного забезпечення пристрою.

Керування здійснюється за допомогою сервера VMware vCenter Server, добре відомого користувачам. vCenter забезпечує ще більш тісну інтеграцію й широку функціональність керування життєвим циклом. Пристрої VxRail дають існуючим користувачам VMware можливість працювати у звичному середовищі, прискорюючи трансформацію IT в інфраструктурі VMware.

Пристрою VxRail поставляються з повним набором інтегрованих критично важливих сервісів керування даними, розроблених Dell EMC. Зокрема, без додаткової плати пропонуються засоби реплікації й резервного копіювання. Це єдине в галузі гіперконвергентний пристрій, у якому інтегровані VMware vSAN і гіпервізор vSphere на рівні ядра. Усе це забезпечує унікальні й неперевершені показники продуктивності й ефективності. Як повністю оптимізовані й підтримувані рішення на базі VMware, ця інфраструктура інтегрована з розробленою VMware платформою керування хмарою й розв'язками для середовищ кінцевих користувачів. Крім цього, VxRail служить платформою, на якій можна представити сучасні пропозиції для програмно-обумовлених ЦОД, у тому числі NSX, vRealize Automation і vRealize Operations.

Мінімальна початкова конфігурація з можливістю розширення

Пристрій Dell EMC VxRail дозволяє почати із системи малого розміру (усього із трьох вузлів) і поступово нарощувати її, легко масштабуючи ємність і продуктивність до 64 вузлів на кластер без переривання роботи. Масштабування по одному вузлу й розширення ємності зберігання забезпечують передбачуване розширення з витратами в міру необхідності й можливістю майбутнього

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

вертикального й горизонтального масштабування при росту вимог бізнесу й користувачів без попереднього планування.

Гнучкість і широкий вибір

Dell EMC VxRail – це гіперконвергентні пристрої із самими широкими можливостями конфігурації. Вони забезпечують приголомшливу гнучкість спеціалізованих платформ, призначених для роботи із самими ресурсномісткими застосунками й робочими навантаженнями.

Пристрою VxRail оновлені за допомогою технологій нового покоління, що гарантують готовність вашої інфраструктури до майбутніх завдань. Зокрема, в VxRail включені накопичувачі кеш-пам'яті NVMe, порти для мережних підключень 25 гігабіт Ethernet, графічні процесори NVIDIA P40 і Масштабовані процесори Intel® Xeon® з конфігураціями з більшим обсягом пам'яті. Крім того, пристрою VxRail підтримують нові сценарії автоматизованого посилення захисту й методологію VxRail Security Technical Implementation Guide (STIG).

У сімейство VxRail входять наступні вузли

– VxRail серії G – пристрої загального призначення для різних сценаріїв використання гіперконвергентних рішень. До 4 вузлів в одному шасі формфактора 2U. Доступні конфігурації All-Flash і гібридні конфігурації.

– VxRail серії E – пристрої, які можна розгорнути як у базовому ЦОД, так і на периферії. Їхня висока щільність, можливість створення груп накопичувачів і збалансована комбінація ресурсів у компактному формфакторі 1U дозволяють розгортати ці пристрої для широкого спектра сценаріїв використання. Модельний ряд містить у собі VxRail E560 (гібридний рішення) і VxRail E560F (рішення All-Flash). Ці моделі на базі стійково го сервера PowerEdge R640 оснащено одним або двома процесорами.

– VxRail серії V – пристрої, оптимізовані для VDI і роботи із графікою, з підтримкою трьох графічних прискорювачів для особливих сценаріїв використання (наприклад, 2D/3D-візуалізації вищого класу). Кожний пристрій має по одному вузлу в шасі формфактора 2U. Модельний ряд містить у собі

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

VxRail V570 (гібридне рішення) і VxRail V570F (рішення All-Flash). Ці моделі на базі стійкового сервера PowerEdge R740 оснащено двома процесорами.

– VxRail серії P – це пристрої, оптимізовані для ресурсномістких робочих навантажень, наприклад для баз даних. Кожний пристрій має по одному вузлу в шасі формфактора 2U. Модельний ряд містить у собі VxRail P570 (гібридний рішення) і VxRail P570F (рішення All-Flash). Ці моделі на базі стійкового сервера PowerEdge R740 оснащено одним або двома процесорами.

– VxRail серії S – це пристрої з високою щільністю зберігання. Вони призначені для ресурсномістких застосунків, наприклад віртуалізованих середовищ Microsoft Sharepoint або Exchange, великих даних і аналітики. Кожний пристрій має по одному вузлу в шасі формфактора 2U. Доступні тільки гібридні конфігурації. Серія містить у собі модель VxRail S570 (тільки гібридний рішення) на базі стійкового сервера PowerEdge R740xd з одним або двома процесорами.

– VxRail серії D – надійна платформа, яка забезпечує повну потужність системи VxRail для робочих навантажень на край, у складних умовах або обмежених просторах. Це означає, що у вас є всі можливості автоматизації й оркестровка із системним програмним забезпеченням VxRail HCI і єдиною точкою підтримки 24x7. Моделі D560 і D560F призначені для роботи в екстремальних умовах, таких як сильна жара й холод, удари, вібрація, пил, вологість і EMI. Із серією VxRail D як адміністратори, так і кінцеві користувачі одержують погоджене середовище від розгортання й керування, щоб користувачький досвід. Серія D пропонує ті ж переваги, що й інша частина портфеля VxRail – простота, маневреність, керування життєвим циклом – але в міцному форм-факторі малої глибини, який стійкий до температури, ударопрочний, що й легко транспортується.

– Конфігурації для будь-яких сценаріїв використання гіперконвергентної інфраструктури.

– Консолідація ресурсів обчислень, зберігання, віртуалізації й керування.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Можливість лінійного масштабування від 3 до 64 вузлів, підтримка від 40 до декількох тисяч VM.

Гіперконвергентна система HPE Simplivity 380

HPE Simplivity 380 – це стійковий модуль форм-фактора 2U з можливістю масштабування: розширення або скорочення конфігурації. Він інтегрує мережі зберігання інформації, СЗД і різні служби в систему, доступну для серверів HPE Proliant DL380 Gen9.

Цей рішення поєднує в собі інфраструктуру й службу обробки інформації в сфері віртуалізованих операцій із просунутою серверною платформою. Обслуговується процесором Intel®xeon®e5-2600 v4, підтримує 5-12 твердотільних накопичувачів загальним обсягом до 40 Тбайт, має два блоки живлення.

Можливості системи

Уніфіковане керування пристроєм дозволяє швидко розгортати платформу й оперативно реагувати на потреби бізнесу. Система стискає й оптимізує дані, використовує апаратну потокову дедуплікацію, що дозволяє зменшити число циклів вводу-виводу й обсяг мережного трафіку. Це дозволяє знизити ємність використовуваного простору.

Функція резервного копіювання, відказостійкість і можливість відновлення після збою надійно захищають дані. Централізоване налаштування віртуальних машин на основі політик забезпечує раціональну роботу платформи.

Набір устаткування для традиційної інфраструктури за вартістю перевищує витрати на такий рішення, управляти яким простіше:

- Гіперконвергентна система з розширеним функціоналом.
- 2 Intel Xeon E5-2600 v4.
- 5-12 твердотільних накопичувачів загальним обсягом 6-40 Тбайт.

Гіперконвергентна система Cisco Hyperflex HX240c M5

Системи Cisco Hyperflex™ з масштабованими процесорами Intel® Xeon® забезпечують гіперконвергенцію для зручної роботи з будь-якими застосунками,

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

- Прискорення роботи: підтримка мереж 40 Гбіт/с.
- Розширення можливостей захисту даних: інтеграція із провідними корпоративними системами резервного копіювання.
- Інтеграція із системами зберігання сторонніх виробників.
- 2RU.
- Адаптивна платформа, яка працює з будь-якими застосунками й у будь-якому місці завдяки простоті гіперконвергентної інфраструктури.

Сертифікована система Lenovo ThinkAgile серії VX

Перетворіть свою негнучку, дорогую й складну в обслуговуванні інфраструктуру в середовище, яке підтримує поступове масштабування, забезпечує високу продуктивність і надійність світового класу. Легко й швидко розгорніть гіперконвергентний рішення на основі VMware vSAN для систем Lenovo ThinkAgile серії VX. Завдяки системам серії VX ви зможете зосередитися на розвитку бізнесу, а не на створенні інфраструктури.

Сертифіковані вузли ThinkAgile серії VX поєднують у собі високу потужність і неперевершену надійність устаткування з багатим набором передвстановленого ПЗ. Вони допоможуть спростити IT-інфраструктуру й скоротити витрати. На всі сертифіковані вузли VX поширюються послуги й технічна підтримка Lenovo.

Компоненти й послуги, що входять до складу сертифікованих вузлів ThinkAgile серії VX і надавані разом з ними:

- Повністю перевірене й інтегроване встаткування й мікрокод Lenovo із сертифікацією VMware vSAN.
- Попередньо завантажене програмне забезпечення Esxi для швидкого розгортання й одержувана окремо ліцензія на програмне забезпечення VMware vSphere і vSAN.
- Централізована підтримка Lenovo ThinkAgile Advantage для швидкого вирішення проблем у цілодобовому режимі без вихідних.
- Додаткові послуги Lenovo по розгортанню.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Звичайно замовники самостійно встановлюють нову гіперконвергентну інфраструктуру. Їм доводиться проводити налаштування, тестування, налагодження, оптимізацію, перевірку й багато інші дії. Однак при використанні систем ThinkAgile серії VX усі ці дії виконуються ще на заводі, а виходить, ви зможете заощадити час і гроші в процесі установки. У комплект поставки систем серії VX входить усе необхідне для початку роботи, включаючи професійні послуги Lenovo по установці й налаштуванню.

Для ефективної роботи гіперконвергентного рішення потрібна надійна платформа. Якщо ви використовуєте сервери як для зберігання даних, так і для проведення обчислень, пов'язаних із критично важливими застосунками, забезпечення їх безперервної роботи знаходить ключове значення. Системи ThinkAgile серії VX містять у собі самі надійні у світі сервери Lenovo і віртуальну мережу VMware vSAN, щоб закласти основу для зростаючого бізнесу. Ці системи дозволяють створювати гнучкі конфігурації для будь-яких варіантів використання. Вони доступні у форматі All-Flash і в гібридних конфігураціях. Масштабування обчислювальних ресурсів і вбудованої пам'яті не представляє складності. Ви можете почати з малого – усього лише із трьох вузлів, а потім просто додавати додаткові вузли або додаткову вбудовану пам'ять у кожний вузол для збільшення продуктивності і ємності. Це можна робити без перерв у роботі.

Характеристики сертифікованих вузлів VX:

– Сертифікований вузол ThinkAgile VX 1U, створений на основі Thinksystem SR630, – це двопроцесорний стійковий сервер висотою 1U, що прекрасно підходить для малого бізнесу й великих підприємств, яким необхідно оптимальне за вартістю, продуктивний і універсальний рішення, що відповідає корпоративним стандартам надійності, керованості й безпеки.

– Сертифікований вузол 2U VX, заснований на Thinksystem SR650, – оптимальний двопроцесорний сервер висотою 2U, найпоширеніший у світі. Його використовують підприємства великого й середнього бізнесу, постачальники

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

важливим для бізнесу трафіком. Підключення зовнішніх мережних пристроїв реалізовано за допомогою шести Ethernet-портів пропускнуою здатністю до 100 Гбіт/с.

– Хмарна операційна система Lenovo Cloud Network Operating System (CNOS) для стійкового комутатора NE2572 Rackswitch тепер підтримує протоколи Radius-Accounting/Mscharp2 і інфраструктуру відкритих ключів (PKI), включаючи розширення для журналювання користувацьких транзакцій і застосування списків керування доступом (ACL) при передачі Ір-пакетів через VLAN інтерфейси третього рівня для інтеграції з Microsoft Azure Stack.

- Стійка – 25U – 42U.
- До 12 серверів SR650.
- 2 Intel Xeon Scalable.
- 40 ТБ фізичної ємності+рівень кешування.

Сертифікована система Lenovo ThinkAgile MX Certified Node SR650

Рішення Microsoft WSSD – це ретельно спроектовані й перевірені системи з технологією Storage Spaces Direct. Сертифіковані вузли ThinkAgile MX Certified Node являють собою новий етап розвитку цих рішень. З їхньою допомогою ви зможете розгортати рішення WSSD на зручних серверах, які відрізняються наступними особливостями:

- Зручність при виборі вузлів і оформленні замовлення.
- Примусове дотримання правил конфігурації (для забезпечення її правильності).
- Використання найбільш оптимального мікрокоду.
- Програма ThinkAgile Advantage (недоступна в окремих країнах).
- Додаткові послуги (розгортання, керування й т.п.).

Відповідь на запитання, пов'язані з розвитком сховища даних

Стрімке зростання потреб у зберіганні даних спричинив бурхливий розвиток таких технологій, як гіперконвергентна інфраструктура (HCI) і програмно-обумовлені сховища даних (SDS). Операційна система Windows

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

дозволяє створювати гнучкі рішення, що поєднують у собі більшу ємність і високу продуктивність для підтримки важких робочих навантажень.

– Ідеальною апаратною платформою для цього рішення служать серверні вузли SR650 висотою 2U. Ці вузли, оснащені двома потужними процесорами сімейства Intel® Xeon® Scalable, відрізняються найвищими показниками надійності, містять великий обсяг оперативної пам'яті й підсистему зберігання даних високої ємності. Усі ці особливості, укупі з більшою кількістю рознімачів вводу-виводу, роблять пропоновані вузли ідеальною основою рішення.

– Конфігурації серверних вузлів SR650, застосовувані в цьому рішення, відрізняються високою щільністю розміщення накопичувачів. Поряд з високопродуктивним ярусом кешу, що полягають із чотирьох накопичувачів SSD або NVMe і 10 звичайних жорстких дисків, вони містять два завантажувальні накопичувачі з підтримкою дзеркалювання. Ви можете почати із двох вузлів і потім наростити їхню кількість до 16.

– SR650 регулярно посідає перше місце в рейтингах по надійності встаткування і задоволеності замовників, складених за результатами незалежних опитувань. Продуктивність, доступність, ємність і якість виготовлення цього серверного вузла більш ніж достатні для підтримки хмарного середовища вашого центр^-центра-дати-центру.

– Кожний вузол обладнаний або двопортовим адаптером Mellanox Connectx-4 стандарту 10 Gbe або 25 Gbe, або двопортовим адаптером Cavium Fastlinq стандарту 10 Gbe або 25 Gbe для високошвидкісного обміну даними. Ці адаптери допоможуть розв'язати будь-яке завдання, пов'язану з побудовою віртуалізованої інфраструктури.

– Вони підтримують технології RDMA over Roce або RDMA over iwarp на апаратному рівні, механізми розвантаження протоколів Ethernet без контролю стану, оверлейні мережі й технологію Gruidirect для забезпечення максимальної продуктивності. Lenovo рекомендує використовувати з'єднання 25Gbe, особливо при роботі з накопичувачами великого обсягу. Стійковий комутатор Thinksystem

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

NE2572 Rackswitch, оптимізований для взаємодії із хмарами, підтримує такі з'єднання. Цей повнофункціональний комутатор рівня підприємства, відповідний до вимог рівнів 2 і 3, підтримує комутування на швидкості каналу з високою пропускнуою здатністю, фільтрацію трафіку й організацію черг трафіку без яких-небудь затримок при передачі даних.

– Для прискорення обміну трафіком застосовуються буфери даних високої ємності, відповідні до потреб дата-центрів. Блоки живлення й вентилятори з підтримкою резервування, укупі із численними функціями забезпечення високої доступності, сприяють відказостійкості при передачі трафіку, важливого для діяльності підприємства. Підключення зовнішніх мережних пристроїв реалізовано за допомогою шести Ethernet-портів пропускнуою здатністю до 100 Гбіт/с.

– Хмарна операційна система Lenovo Cloud Network Operating System (CNOS) для стійкового комутатора NE2572 Rackswitch тепер підтримує протоколи Radius-Accounting/Mscharp2 і інфраструктуру відкритих ключів (PKI), включаючи розширення для журналювання користувацьких транзакцій і застосування списків керування доступом (ACL) при передачі Ір-пакетів через VLAN інтерфейси третього рівня для інтеграції з Microsoft Azure Stack.

- Рішення Microsoft WSSD для Storage Spaces Direct.
- 2U.
- 2 Intel Xeon Scalable.
- До 1,5 ТБ в 24 розмірностях TruDDR4.
- 14 дисків LFF.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватиме довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.
 - Підтримка Metal Driver GPU для macOS і iOS.
 - Вбудований Fmxlinux.
 - Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
 - Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).
 - Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.
 - Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services
 - У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey
- RAD Studio 10.4 Короткий огляд:**
- Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проєктах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.
 - Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

підвищеною швидкістю. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису `custom managed records`. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи керування гібридною IT-інфраструктурою.

В процесі розробки кваліфікаційної бакалаврської роботи 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

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

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.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

ІТ-технології йдуть за бізнесом, щораз закриваючи нові його потреби. Бізнес прагне більше гнучкості й надійності? Будь ласка! Потрібно будь-яка кількість ресурсів на вимогу? Не питання. Інша справа, що для виконання всіх бізнес-бажань іноді доводиться йти на компроміси. Це добре прослідковується в сфері хмарних послуг. Багато компаній прагнуть одержати переваги, які дає хмара, але не все готові перенести туди 100 % ІТ-інфраструктури. А це й не потрібно. Залишити всі робочі навантаження в локальному середовищі або в тому ж обсязі віддати їх у хмару – крайності. Ефекту буде більше, якщо бізнес знайде між ними розумний баланс – те, що сьогодні називають гібридною інфраструктурою.

Як побудована гібридна інфраструктура

Гібридна інфраструктура дозволяє вибирати, які навантаження зручно, вигідно й безпечно обслуговувати локально, а які – у хмарі. Бізнес може набудувати виконання завдань таким чином, щоб швидко одержати ефект і скористатися перевагами хмарного середовища. Критичні процеси й ті, які пов'язані з обробкою конфіденційних даних, краще залишити в локальному середовищі, менш значимі (поштовий сервер, файлообмінники) – розмістити в хмарі. Такий підхід допоможе раціонально використовувати обчислювальні потужності й заощадити на змісті ІТ-інфраструктури.

Стратегія як основа

Гра в шахи – не єдина ситуація, коли потрібно мислити стратегічно. Бізнесу доводиться прораховувати багато варіантів і вибирати ті, які дозволять оптимально розподілити робочі навантаження. Від цього буде залежати, які результати одержить або, навпаки, не одержить компанія.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Стратегія розгортання гібридної інфраструктури – річ специфічна. Те, що підійде одному бізнесу, виявиться провальним для іншого, тому що звичайно відрізняються цілі, сфера, навантаження й доступні ресурси. Навіть усередині галузі компанії одного рівня можуть по-різному використовувати гібридну інфраструктуру у своїх інтересах. Тому перед корпоративними клієнтами коштують завдання, у яких немає універсального рішення: знайти золоту середину між локальною й хмарною інфраструктурою.

Логіка або інтуїція?

Можна подивитися на конкурентів, піти їхньому прикладу, віддавши 100 % робочих навантажень у хмару й... програти: втратити час і гроші. А можна проаналізувати підходящі варіанти розподілу ІТ-інфраструктури між локальним середовищем і хмарою й виявитися у вигазі. Експерти аналітичної компанії IDC акцентують на цьому увагу й застерігають: навіть лідери ІТ-сектору розуміють, що повномасштабний переїзд у хмару відріже дорога назад і не дасть змінити обрану стратегію. Замість цього їм доведеться прискореними темпами пристосовуватися до того, як тепер будуть організовані робочі процеси.

В Moor Insights & Strategy згодні з колегами й затверджують: компаніям, які прагнуть перенести в хмару ІТ-інфраструктуру, слід розглядати модель Cloud If, але ніяк не Cloud Always. На їхню думку, хмара повинна бути інструментом, а не самоціллю. Перенос робочих навантажень у хмару не розв'яже проблеми й не закрий всі потреби бізнесу, а стане свого роду випробуванням з певною часткою ризику.

Технічно будь-яка компанія може перейти в хмару вже сьогодні. Але на практиці не кожна до цього готова. Людині спокійніше, коли гроші зберігаються поруч, «під подушкою». З бізнесом те ж саме, тільки він зберігає й обробляє дані, а їх цілісність і безпека – це ті ж гроші.

Бізнесу подобаються переваги хмар, тому вони згодні на компроміс – гібридну модель, коли в хмару переносять менш важливі дані й тим самим розвантажують локальний майданчик. Плюс хмара можна використовувати для

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

зберігання резервних копій або для розгортання середовищ тестування. Досить у парі кліків підняти віртуальну машину із шаблону, обрати сервер з певною кількістю процесорів і пам'яті, а після тестування відмовитися від цієї послуги й більше за неї не платити.

Великі компанії можуть розгорнути приватна хмара у своєму дата-центрі й працювати з ним віддалено. Так бізнес не буде переживати про те, як організовано живлення, охолодження, як забезпечується безпека й надійність ЦОД. Це оптимальний рішення для держсектора й фінансової сфери. Невелика компанія такого собі дозволити не може, тому звертається до провайдера хмарних послуг. Їй зручно залишити файловий сервер у себе, а в хмару перенести пошту й бекапи на випадок стихійного лиха.

3.2 Розробка структурної схеми

Традиційні методи IT-Адміністрування й керування IT-інфраструктурою активно переглядаються й поступово реформуються. Використання штучного інтелекту (ШІ) в управлінні IT-інфраструктурою малює найцікавіші перспективи.

Термін AIOps (Artificial Intelligence for IT Operations) позначає застосування штучного інтелекту до керування експлуатацією IT. AIOps використовує методи й алгоритми штучного інтелекту як для моніторингу даних, так і для скорочення часу й числа простоїв внаслідок збоїв. Бурхливий розвиток цифрових компаній приводить до технологічних змін і в організації їх діяльності, спрямованим на оптимізацію й усуненню витрат, пов'язаних із традиційними способами керування інфраструктурою.

Цифрові компанії оперують з величезними потоками даних, багатих різною інформацією. Їхній аналіз і моніторинг потрібно робити без яких-небудь затримок. Від збору даних до зберігання, від обробки до аналізу – усі операції повинні повинні проводитися злагоджено, швидко й безвідмовно, щоб використовувати добуту інформацію максимально ефективним образом.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Шлях до АІОрs починається з усвідомлення необхідності й реалізації глибоких цифрових перетворень. Впровадження новітніх технологій приводить до серйозного відновлення ІТ. Машинне навчання, обробка великих даних, Інтернет речей, інтерфейси прикладного програмування, граничні обчислення й т.п. неминуче проникають усередину ІТ-організації й торкаються керування ІТ-інфраструктурою, яке зазнає істотним змінам, що реформують її традиційні підходи до організації робіт.

Головні фактори, що виявляють вплив на цифрову трансформацію:

– складна ІТ-інфраструктура. Ручні операції (збір даних, обробка даних, керування хмарними сховищами, оптимізація баз даних, використання сторонніх інструментів і утиліт керування й інтеграції SaaS) стає усе сутужніше виконувати через величезні масштаби.

– обробка величезних потоків даних. Технології Інтернету речей (ІоТ) швидко виходять на рівень промислового використання. ІоТ-пристрою генерують велика кількість даних. Обробка такої кількості даних уже не під силу для окремих фахівців. Необхідно використовувати алгоритми ШІ, за допомогою яких дані інтерпретуються й аналізуються.

– збільшення обчислювальних потужностей. Мережна інфраструктура стала надійною, але, проте, ІТ усі також маємо гостру потребу в підвищенні швидкості обробки даних. Розгортаючи ІТ-рішення, що використовують технології штучного інтелекту й машинного навчання, можна різко побільшати швидкість обробки даних.

Два основні стовпи АІОрs, які прискорюють інновації в керуванні ІТ-інфраструктурою, – машинне навчання й великі дані. Замість того щоб витратити людські зусилля й час на підготовку, обробку й аналіз сирих даних, узятих з різних платформ, можна виконувати ті ж завдання, використовуючи алгоритми й методи машинного навчання, аналізувати структуру й природу даних і витягати з них необхідну інформацію.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

AIOps поліпшує й автоматизує керування IT-інфраструктурою, використовуючи наступні підходи:

– Використання алгоритмів машинного навчання й нейромереж для ефективного й швидкого аналізу зібраних даних. Сам збір даних здійснюється за допомогою пристроїв і утиліт Інтернету речей, що використовують технології хмарного зберігання.

– Автоматична інтерпретація подій: якщо щось десь в інфраструктурі не так, то помилки оперативно виявляються, діагностуються й усуваються без надання впливу на швидкість виконання складнозв'язаних і залежних друг від друга операцій з даними.

– Зменшення впливу людського фактора шляхом автоматизації процесів, що використовували раніше діяльність окремих фахівців.

Використання AIOps виявляє величезний позитивний вплив на роботу складних і великих IT-компаній, людських ресурсів яких не вистачає на виконання всіх завдань. Серед них можна виділити:

– Команди розроблювачів. Компанії, які використовують підходи DevOps у своїй роботі, використовують ШІ й інші технології для вдосконалювання своїх внутрішніх процесів. Завдяки цьому підвищується швидкість зворотного зв'язку, гнучкість і прозорість у керуванні IT-інфраструктурою.

– Компанії, що широко використовують хмарні обчислення. У хмарних обчислень велике майбутнє. Організації зберігають величезні дані в хмарних сховищах, і всі операції, пов'язані з доступом і керуванням хмарними даними, необхідно оптимізувати, тому що з ними можуть виникати різні проблеми. Щоб дані процеси виконувалися без помилок, необхідна автоматизація.

– Цифрові компанії. Компанії по усьому світу активно впроваджують технологічні рішення. Фінтех, туризм, охорона здоров'я, електронна комерція – компанії із цих і інших сфер автоматизують свої внутрішні процеси, надаючи своїм клієнтам можливість використовувати застосунки, засновані на нових

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

запропонувати щось купити або цікавого подивитися або почитати – це теж приклад машинного навчання. Ці методи надійно й широко використовуються в середовищах, де потрібна висока швидкість реакції на динамічно мінливі умови й очікування користувачів. Прийшла черга й ІТ зробити крок у майбутнє.



Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На даному етапі різні виробники пропонують надлишкові або часткові рішення AIOps, тому визначимо 9 функціональних елементів, які реалізуються в AIOps платформа, розробленої в даній роботі:

- Керування накопиченими даними – програмне забезпечення або апаратний комплекс, які дозволяють записувати, індексувати й зберігати напівструктуровані дані, які надходять у більших обсягах і з високою швидкістю. По своїй суті це керування більшими даними.

- Потокове керування даними – програмне забезпечення або апаратний комплекс, які дозволяють здійснювати захвата, можливу нормалізацію й індексацію, а також вистава в реальному часі одного або декількох типів даних,

представлених нижче. Для керування даними в режимі реального часу, програмне забезпечення повинне не тільки представляти вхідні дані в часі, які користувач буде вважати реальними, але й фактично надавати дані користувачеві безпосередньо в момент приймання, так сказати на лету, не вимагаючи доступу до бази даних.

- Приймання логів – програмне забезпечення, яке дозволяє ухвалювати текстові лог файли від різних джерел з метою подальшої обробки й індексування для зберігання.

- Приймання пакетних даних – програмне забезпечення, яке дозволяє здійснювати захват пакетних даних прямо із дзеркальних портів або розгалужувачів трафіку. Усі пакети або записи про потоки (flow) повинні бути підготовлені до запису й проіндексовані.

- Приймання цифрових показників – програмне забезпечення, яке дозволяє ухвалювати будь-які цифрові показники, до яких може бути застосована обробка шляхом математичних операцій або агрегації за часом.

- Приймання документів – програмне забезпечення, яке дозволяє ухвалювати документи й здійснювати семантичний і синтаксичний аналіз документів. По своїй суті це обробка природньої мови – natural-language processing (NLP).

- Автоматизоване виявлення й прогнозування шаблонів – програмне забезпечення, яке на основі історичних або потокових даних одного або декількох зі згаданих вище типів створює математичні або структурні шаблони, що описують нові кореляції, які можуть бути виведені з наборів даних, але не безпосередньо присутні в них. Потім ці закономірності можуть використовуватися для прогнозування інцидентів з різним ступенем імовірності.

- Виявлення аномалій – програмне забезпечення, яке використовує різні шаблони, виявлені попередніми компонентами, щоб спочатку визначити базову продуктивність або стан елемента або сервісу, а потім визначає будь-які відхилення і їх ступінь важливості.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3.

Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

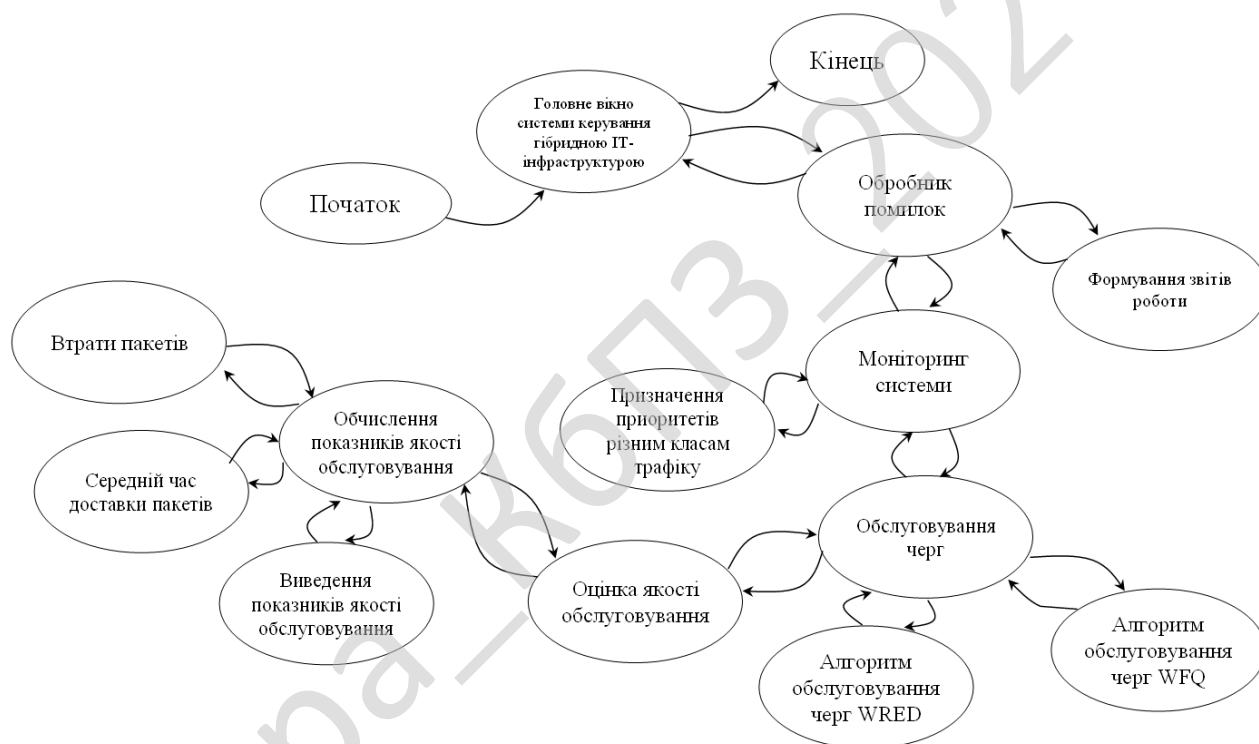


Рисунок 3.3 – Діаграма взаємодії процесів

Основні складові елементи діаграми взаємодії процесів це потоки даних:

- Репозиторії, потік сховища даних.
- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1 після запуску програми відбуваються наступні дії:

- Виведення системи керування гібридною ІТ- інфраструктурою;
- З'єднання з ЛМ інфраструктури;
- Виведення інформації про мережу;
- Запит встановлення параметрів з викликом підпрограми;
- Моніторинг вхідного TCP/IP трафіку;
- Класифікація вхідного TCP/IP трафіку;
- Організація черг системи;
- Виклик підпрограми статистичної оцінки;
- Запит дослідження WRED;
- Запит дослідження WFQ.

Робота підпрограми зображено на рисунку 4.2, з рисунку видно, що відбуваються наступні дії:

- Запит встановлення пріоритетів;
- Аналіз та класифікація мережних додатків за типом трафіку;
- Надання керуючій інформації першого пріоритету;
- Надання даним, що передаються за протоколом FTP, третього пріоритету;
- Надання текстовій інформації четвертого пріоритету;
- Встановлення пріоритетів відповідно до типу трафіку;
- Запит встановлення пріоритетів вхідного трафіку;

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- Запуск системи у тестовому режимі;
- Визначення показників QoS для різних мережних додатків;
- Оптимізація параметрів QoS під наявні мережні програми;
- Встановлення пріоритетів відповідно до розрахунків.

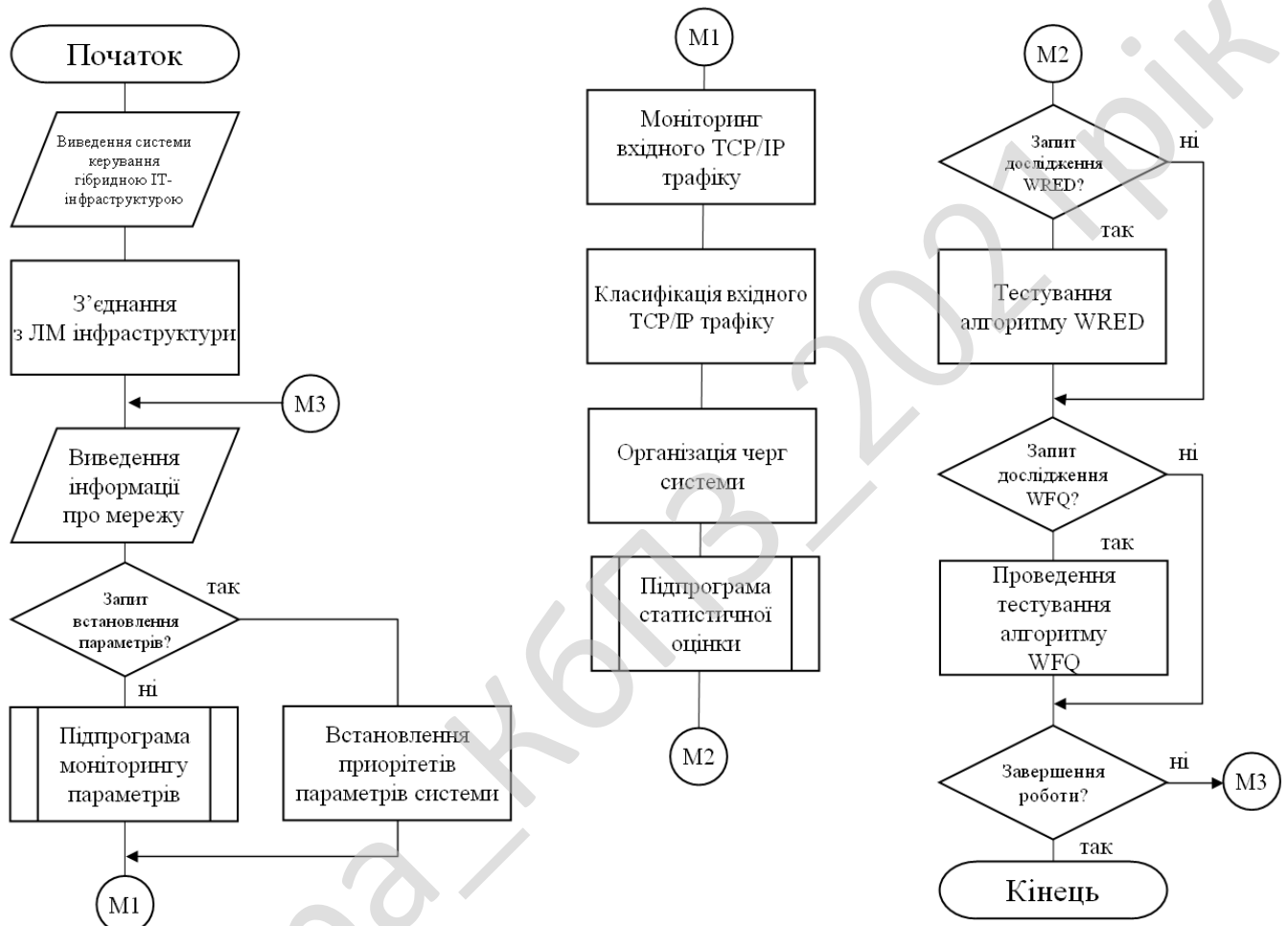


Рисунок 4.1 – Блок-схема основної програми

Дані з таблиці маршрутизації досліджуємої системи інтелектуального управління IT-інфраструктурою отримуються за допомогою наступних підпрограм:

```

procedure Get_ForwardT( List: TStrings );
var
    IPForwRow      : TMibIPForwardRow;
    TableSize      : DWORD;

```

```

ErrorCode      : DWORD;
i              : integer;
pBuf          : PChar;
NumEntries    : DWORD;

```

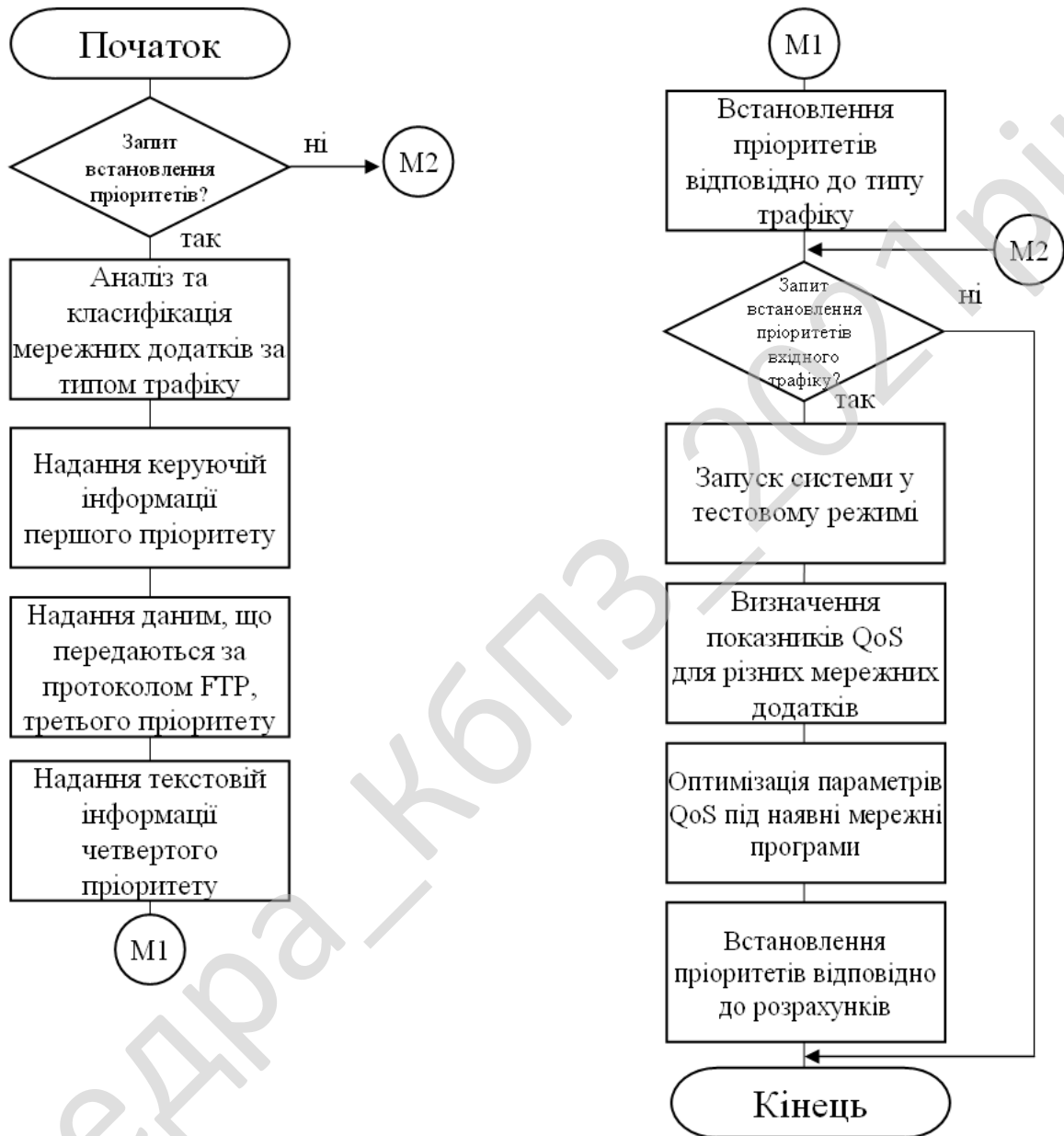


Рисунок 4.2 – Блок-схема роботи підпрограми

```

begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    TableSize := 0;
    // перший виклик: беремо довжину таблиці

```

```

NumEntries := 0 ;
ErrorCode := GetIpForwardTable( Nil, @TableSize, true);
if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
    EXIT;
// беремо таблицю
GetMem( pBuf, TableSize );
ErrorCode := GetIpForwardTable( PTMibIPForwardTable( pBuf ), @TableSize, true);
if ErrorCode = NO_ERROR then
begin
    NumEntries := PTMibIPForwardTable( pBuf )^.dwNumEntries;
    if NumEntries > 0 then
    begin
        inc( pBuf, SizeOf( DWORD ) );
        for i := 1 to NumEntries do
        begin
            IPForwRow := PTMibIPForwardRow( pBuf )^;
            with IPForwRow do
            begin
                if (dwForwardType < 1)
                or (dwForwardType > 4) then
                    dwForwardType := 1 ; // дані
                List.Add( Format(
                    ' %15s | %15s | %15s | %8.8x | %7s | %5.5d | %7s | %2.2d' ,
                    [IPAddr2Str( dwForwardDest ),
                    IPAddr2Str( dwForwardMask ),
                    IPAddr2Str( dwForwardNextHop ),
                    dwForwardIFIndex,
                    IPForwTypes[dwForwardType],
                    dwForwardNextHopAS,
                    IPForwProtos[dwForwardProto],
                    dwForwardMetric1
                    ] ) );
                end ;
                inc( pBuf, SizeOf( TMibIPForwardRow ) );
            end;
        end
        else
            List.Add( ' немає даних.' );
        end
        else
            List.Add( SysErrorMessage( ErrorCode ) );
        dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibIPForwardRow ) );
    end
end

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0036.00.00.ПЗ

Арк.

45


```

else
    List.Add( SysErrorMessage( ErrorCode ) );
end;
function IpHlpIPStatistics (var IPStats: TMibIPStats): integer ;
// дані
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetIPStatistics( @IPStats ) ;
end ;
//-----
procedure Get_UdpStatistics( List: TStrings ) ;
var
    UdpStats      : TMibUDPStats;
    ErrorCode      : integer;
begin
    if not Assigned( List ) then EXIT;
    ErrorCode := GetUDPStatistics( @UdpStats );
    if ErrorCode = NO_ERROR then
        begin
            List.Clear;
            with UdpStats do
                begin
                    List.add( ' Датаграми (In)      : ' + inttostr( dwInDatagrams ) );
                    List.add( ' Датаграми (Out)     : ' + inttostr( dwOutDatagrams ) );
                    List.add( ' Немає портів       : ' + inttostr( dwNoPorts ) );
                    List.add( ' Помилка (In)       : ' + inttostr( dwInErrors ) );
                    List.add( ' UDP список портів  : ' + inttostr( dwNumAddrs ) );
                end;
            end;
        end
    else
        List.Add( SysErrorMessage( ErrorCode ) );
    end;
//-----*//
function IpHlpUdpStatistics (UdpStats: TMibUDPStats): integer ;
// дані
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetUDPStatistics ( @UdpStats ) ;
end ;
//-----

```

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

```

procedure Get_ICMPStats( ICMPIn, ICMPOut: TStrings );
var
  ErrorCode      : DWORD;
  ICMPStats      : PTMibICMPInfo;
begin
  if ( ICMPIn = nil ) or ( ICMPOut = nil ) then EXIT;
  ICMPIn.Clear;
  ICMPOut.Clear;
  New( ICMPStats );
  ErrorCode := GetICMPStatistics( ICMPStats );
  if ErrorCode = NO_ERROR then
  begin
    with ICMPStats.InStats do
    begin
      ICMPIn.Add( ' Прийнято повідомлень      : ' + IntToStr( dwMsgs ) );
      ICMPIn.Add( ' Помилка                    : ' + IntToStr( dwErrors ) );
      ICMPIn.Add( ' Розташування недотягнуто : ' + IntToStr( dwDestUnreachs ) );
      ICMPIn.Add( ' Час перевищений           : ' + IntToStr( dwTimeEcxcds ) );
      ICMPIn.Add( ' Проблеми з параметрами      : ' + IntToStr( dwParmProbs ) );
      ICMPIn.Add( ' Джерело відключено           : ' + IntToStr( dwSrcQuenchs ) );
      ICMPIn.Add( ' Переназначено                : ' + IntToStr( dwRedirects ) );
      ICMPIn.Add( ' Ехо запит                      : ' + IntToStr( dwEchos ) );
      ICMPIn.Add( ' Ехо відповідь                 : ' + IntToStr( dwEchoReps ) );
      ICMPIn.Add( ' Запит мітки часу                : ' + IntToStr( dwTimeStamps ) );
      ICMPIn.Add( ' Відповідь мітки часу             : ' + IntToStr( dwTimeStampReps));
      ICMPIn.Add( ' Запит маски адрес           : ' + IntToStr( dwAddrMasks));
      ICMPIn.Add( ' Відповідь маски адрес        : ' + IntToStr( dwAddrReps ) );
    end;
  //
    with ICMPStats.OutStats do
    begin
      ICMPOut.Add( ' Повідомлення вправлено          : ' + IntToStr( dwMsgs ) );
      ICMPOut.Add( ' Помилка                    : ' + IntToStr( dwErrors ) );
      ICMPOut.Add( ' Розташування недотягнуто : ' + IntToStr( dwDestUnreachs));
      ICMPOut.Add( ' Час перевищений           : ' + IntToStr( dwTimeEcxcds ) );
      ICMPOut.Add( ' Проблеми з параметрами      : ' + IntToStr( dwParmProbs ) );
      ICMPOut.Add( ' Джерело відключено           : ' + IntToStr( dwSrcQuenchs ) );
      ICMPOut.Add( ' Переназначено                : ' + IntToStr( dwRedirects ) );
      ICMPOut.Add( ' Ехо запит                      : ' + IntToStr( dwEchos ) );
      ICMPOut.Add( ' Ехо відповідь                 : ' + IntToStr( dwEchoReps ) );
      ICMPOut.Add( ' Запит мітки часу                : ' + IntToStr( dwTimeStamps ) );
      ICMPOut.Add( ' Відповідь мітки часу             : ' + IntToStr( dwTimeStampReps ) );
    end;
  end;
end;

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0036.00.00.ПЗ

Арк.

48

```

    ICMPOut.Add( 'Запит маски адрес: ' + IntToStr( dwAddrMasks ) );
    ICMPOut.Add( 'Відповідь маски адрес : ' + IntToStr( dwAddrReps ) );
end;
end
else
    IcmpIn.Add( SysErrorMessage( ErrorCode ) );
    Dispose( ICMPStats );
end;
//-----
procedure Get_RecentDestIPs( List: TStrings );
begin
    if Assigned( List ) then
        List.Assign( RecentIPs );
end;
initialization
    RecentIPs := TStringList.Create;
finalization
    RecentIPs.Free;
end;

```

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ИС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

– Німецька V-модель була розроблена аерокосмічної компанією IAVG в Оттобрунні поряд з Мюнхеном у співпраці з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проектів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери.

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. У середині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що прийнятно-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектурі, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

відхилення в проєкті і ризику на ранніх стадіях і покращує якість управління проєктом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проєкту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проєкту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проєкти.

4. Підвищення якості комунікації між учасниками проєкту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проєкту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проєкт і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проєкту V-подібна модель може бути адаптована під цей проєкт, так як ця модель не залежить від типів організацій та проєктів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

Обмеження. Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

– Не регулюється розміщення контрактів на обслуговування.

– Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

– V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібної моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПЗ – перед розробкою компонентів.

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дані повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

– Модель не передбачає роботу з паралельними подіями.

– У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

Фрагменти розширеного ключа $K_0...K_{63}$ обчислюються в процесі процедури розширення ключа, що виконується в такий спосіб:

Етап 1. 512-бітний вихідний ключ шифрування ділиться на 16 фрагментів по 32 біта $K_0...K_{15}...$

Етап 2. Інші фрагменти розширеного ключа $K_{16}...K_{63}$ обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = O_1(K_{i-2}) + K_{i-7} + O_0(K_{i-15}) + K_{i-16},$$

де функції O_0 і O_1 визначені так:

$$O_0(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3),$$

$$O_1(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10),$$

де \gg – операція побітового зрушення (не циклічного) вправо.

Раунди розшифрування алгоритму виконуються у зворотній послідовності:

$$T = A_{i+1} + S_0'(B_{i+1}) + Maj'(B_{i+1}, C_{i+1}, D_{i+1}) + 2,$$

$$H_i = T + S_1'(F_{i+1}) + Ch'(F_{i+1}, G_{i+1}, H_{i+1}) + M_i' + K_i' + 4,$$

$$G_i = H_{i+1},$$

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

$$F_i = G_{i+1},$$

$$E_i = F_{i+1},$$

$$D_i = E_{i+1} + T + 1,$$

$$C_i = D_{i+1},$$

$$B_i = C_{i+1},$$

$$A_i = B_{i+1}.$$

Кафедра КБПЗ – 2021 рік

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

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

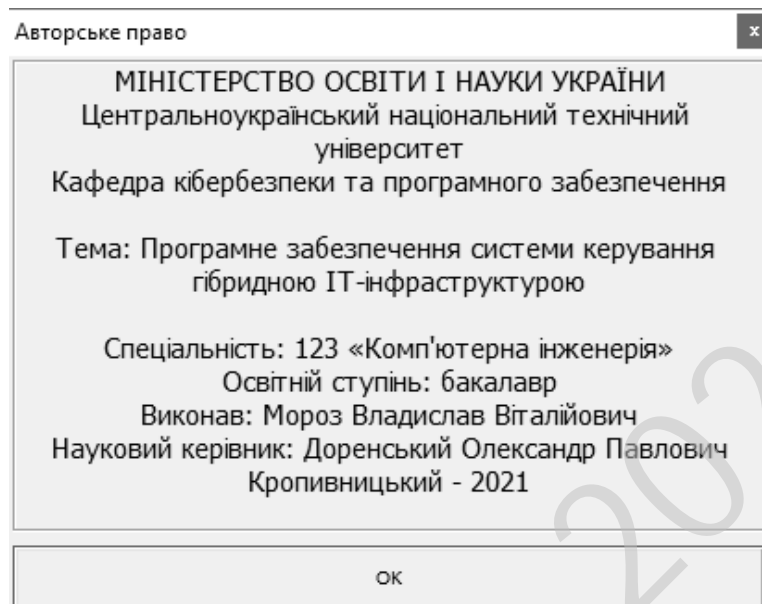


Рисунок 5.2 – Авторське право

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

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи керування гібридною ІТ-інфраструктурою.

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

Рішення завдання полягало у вирішенні наступних задач:

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

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання керування гібридною ІТ-інфраструктурою.

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 Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи керування гібридною ІТ-інфраструктурою. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

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.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHACAL-2.

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.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Семенов С.Г. Исследования вероятностно-временных характеристик мультисервисного канала связи с использованием математического аппарата GERT-сети / С.Г. Семенов, В.В. Босько, І.А. Березюк // Системи обробки інформації. – Х.: ХУ ПС, 2012. – Т. 1., Вип. 3(101). – С. 139-142.
2. Семенов С.Г. Моделирование защищенного канала связи с использованием экспоненциальной GERT-сети / С.Г. Семенов, А.А. Можаяев // Информатика, математическое моделирование, экономика. – Смоленськ.: Смоленский филиал АНО ВПО ЦС РФ "Российский университет кооперации". – 2012. – Том.1. – С. 152-160.
3. Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов // Вісник Національного технічного університету «Харківський політехнічний інститут». – Х.:НТУ «ХПІ». – 2012. –№62 (968). – С 173-181.
4. Семенов С.Г. Защита данных в компьютеризированных управляющих системах / С.Г. Семенов, В.В. Давыдов, С.Ю. Гавриленко. – LAP Lambert Academic Publishing GmbH & Co. KG (Саарбрюккен, Германия), 2014. – 236 с.
5. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.
6. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

7. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

8. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

9. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

10. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.

11. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

12. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam,

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

13. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.

14. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. – Х.: ХУПС, 2015. – № 3(20). – С. 134-141.

15. Смирнов С. А. Комплекс GERT-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

16. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.

17. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.

18. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.

19. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев,

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.

20. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.

21. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных вирусов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.

22. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.

23. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.

24. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

25. Смирнов С. А. Технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.

26. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.

27. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.

28. Смирнов С. А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.

29. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.

30. Смирнов С. А. Разработка комплекса GERT-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

взаємодії» (IT & I): зб. тез II міжнар. наук. -практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.

31. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

32. Смирнов С. А. GERT-модели технологии облачной антивирусной защиты / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Безпека українського суспільства в концепції вступу в постіндустріальне суспільство ЄС: зб. тез Круглого столу, м. Київ, 16 грудня 2015 р. – К.: Європейський університет, 2015. – С.41-43.

33. Смирнов С. А. Алгоритмы формирования множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. п раць II Міжнар. наук.-практ. конф., м. Київ, 24-27 лютого 2016 р. – К.: Європейський університет, 2016. – С. 140-142.

34. Смирнов С. А. Разработка и реализация метода безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Securitea informationala 2015-2016: Conferenta internationala (editia a XII-a), Chisinau, Moldova, 3 martie 2016. – Chisinau: ADSEM, 2016. – С. 90-96.

35. Смирнов С. А. Алгоритм формирования базового множества маршрутов передачи метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформатика та системні науки (ICN-2016): зб. тез VII всеукр. наук.-практ. конф., м. Полтава, 10-12 березня 2016 р. – Полтава: ПУЕТ, 2016. – С. 261-263.

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

36. Смирнов С. А. Система обработки и формирования начального состояния маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблемы кібербезпеки інформаційно-телекомунікаційних систем: зб. тез наук. -практ. конф., м. Київ, 10-11 березня 2016 р. – К.: КНУ ім. Тараса Шевченка, 2016. – С. 81-82.

37. Смирнов С. А. Алгоритм безопасной маршрутизации на базовом множестве путей передачи метаданных в программный сервер облачной антивирусной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційна безпека та комп'ютерні технології (IS&CT): зб. тез міжнар. наук. -практ. конф., м. Кіровоград, 24-25 березня 2016 р. – Кіровоград: КНТУ, 2016. – С. 73.

38. Смирнов С. А. Исследование способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016), м. Харків, 30 березня – 1 квітня 2016 р. – Х.: НТУ «ХП», 2016. – С. 14.

39. Смирнов С. А. Разработка способа контроля линий связи телекоммуникационной системы для облачных антивирусов / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Матеріали XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» (м. Кіровоград, 15-16 квітня 2016 р.). –Кіровоград: КНТУ, 2016. – С. 182-186.

40. Смирнов С. А. Разработка и исследование способа контроля линий связи телекоммуникационных сетей для облачных антивирусных систем / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Проблеми і перспективи розвитку ІТ-індустрії: VIII міжнар. наук.-практ. конф., м. Харків, 28-29 квітня 2016 р.: зб. тез. – Х.: ХНЕУ, 2016. – С. 48.

41. Смирнов С. А. Модель системы нейросетевых экспертов безопасной маршрутизации для облачных антивирусных систем / А. А. Смирнов,

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

50. Elwalid, D. Mitra, I. Saniee, and I. Widjaja. Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs// Journal of lightwave technology. – 2003. – №21(11), P. 2828-28-38.

51. A.B. Bagula, M. Botha, and A.E Krzesinski. Online Traffic Engineering: The Least Interference Optimization Algorithm // IEEE Communications Society – 2004, P. 1232-1236.

52. Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Evaluating the Impact of Stale Link State on Quality-of-Service Routing // IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.

53. Basabi Chakraborty. Simultaneous Search for Multiple Routes using Genetic Algorithm / IEEE International Conference on Computational Intelligence for Measurement System and Applications Boston, MA, USA, 14-16, July 2004, P. 77-80/

					КБР-123.21.0036.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					КБР-123.21.0036.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Мороз В.В.				<i>Програмне забезпечення системи керування гібридною ІТ- інфраструктурою</i>		
Перевірів	Доренський О.П.						
Н. Контр.	Гермак В.С.				Літ.	Аркуш	Аркушів
Затв.	Смірнов О.А.				Б	1	6
					ЦНТУ КІ-18-ЗСК		

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи керування гібридною ІТ-інфраструктурою.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 204-02 від 28.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи керування гібридною ІТ-інфраструктурою.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					КБР-123.21.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи керування гібридною IT-інфраструктурою;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					КБР-123.21.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.

					КБР-123.21.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 68 аркушів.

					КБР-123.21.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2020 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 3.06.2020 р.

					КБР-123.21.0036.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Доренський О.П.

Програмне забезпечення системи керування гібридною ІТ-інфраструктурою

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2021 року

```

unit Main;

interface

// опис бібліотек

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, Stat,
  ShellAPI, ShlObj, ImgList, TCP_IP, About;

//опис типів

type
  TMainForm = class(TForm)
    gbxShares: TGroupBox;
    lbxShares: TListBox;
    gbxSessions: TGroupBox;
    lvSessions: TListView;
    bvlSessions: TBevel;
    gbxFiles: TGroupBox;
    btnGetShares: TButton;
    btnCloseShares: TButton;
    btnAddShares: TButton;
    btnCloseSession: TButton;
    btnGetSessions: TButton;
    bvlTopSessions: TBevel;
    plButtonFiles: TPanel;
    btnGetFiles: TButton;
    btnCloseFile: TButton;
    bvlLeftFiles: TBevel;
    plFiles: TPanel;
    lvFiles: TListView;
    bvlTopFiles: TBevel;
    gbxTraffic: TGroupBox;
    lvTraffic: TListView;
    bvlTraffic: TBevel;
    tmrTraffic: TTimer;
    Button1: TButton;
    rgScope: TRadioGroup;
    GroupBox1: TGroupBox;
    cbUsageAll: TCheckBox;
    cbUsageConnectable: TCheckBox;
    cbUsageContainer: TCheckBox;
    GroupBox2: TGroupBox;
    cbTypeAny: TCheckBox;
    cbTypeDisk: TCheckBox;
    cbTypePrint: TCheckBox;
    Net_IIM_Tree: TTreeView;
    ImageList1: TImageList;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    function IsNT(var Value: Boolean): Boolean;
    procedure btnGetSharesClick(Sender: TObject);
    procedure btnCloseSharesClick(Sender: TObject);
    function SelectDirectory: String;
    procedure btnAddSharesClick(Sender: TObject);
    function CardinalToTimeStr(Value: Cardinal): String;
    procedure btnGetSessionsClick(Sender: TObject);
    procedure btnCloseSessionClick(Sender: TObject);
    procedure btnGetFilesClick(Sender: TObject);
    procedure btnCloseFileClick(Sender: TObject);
    procedure tmrTrafficTimer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  end;

```

```

procedure Net_IIM_TreeCustomDrawItem(Sender: TCustomTreeView;
  Node: TTreeNode; State: TCustomDrawState; var DefaultDraw: Boolean);
procedure Net_IIM_TreeDbClick(Sender: TObject);
procedure Net_IIM_TreeGetImageIndex(Sender: TObject; Node: TTreeNode);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);

//опис типів та записів

private
  { Private declarations }
public
  { Public declarations }
  SessionCloseKey: array [0..512] of SmallInt;
  procedure Open_Do_Close_Enum(const ParentNode: TTreeNode;
    ResScope, ResType, ResUsage: DWORD; const Net_IIM_ContainerToOpen:
PNet_IIM_Resource);
  // function OpenEnum(const Net_IIM_ContainerToOpen: PNet_IIM_Resource;
  //   ResScope, ResType, ResUsage: DWORD): THandle;
  // function EnumResources(const ParentNode: TTreeNode;
  //   ResScope, ResType, ResUsage: DWORD; hNet_IIM_Enum: THandle): UINT;
end;

type
  TShareInfo2 = packed record
    shi2_net_IIM_name : PWChar;
    shi2_type: DWORD;
    shi2_remark :PWChar;
    shi2_permissions: DWORD;
    shi2_max_uses : DWORD;
    shi2_current_uses : DWORD;
    shi2_path : PWChar;
    shi2_passwd : PWChar;
end;
PShareInfo2 = ^ TShareInfo2;
TShareInfo2Array = array [0..512] of TShareInfo2;
PShareInfo2Array = ^ TShareInfo2Array;

type
  TShareInfo50 = packed record
    shi50_net_IIM_name : array [0..12] of Char;
    shi50_type : Byte;
    shi50_flags : Word;
    shi50_remark : PChar;
    shi50_path : PChar;
    shi50_rw_password : array [0..8] of Char;
    shi50_ro_password : array [0..8] of Char;
end;

type
  TSessionInfo502 = packed record
    Sesi502_cname: PWideChar;
    Sesi502_username: PWideChar;
    Sesi502_num_opens: DWORD;
    Sesi502_time: DWORD;
    Sesi502_idle_time: DWORD;
    Sesi502_user_flags: DWORD;
    Sesi502_cltype_name: PWideChar;
    Sesi502_transport: PWideChar;
End;
PSessionInfo502 = ^TSessionInfo502;
TSessionInfo502Array = array[0..512] of TSessionInfo502;
PSessionInfo502Array = ^TSessionInfo502Array;

type
  TSessionInfo50 = packed record
    Sesi50_cname : PChar;

```

```

    Sesi50_username      : PChar;
    sesi50_key           : Cardinal;
    sesi50_num_conns     : Word;
    sesi50_num_opens     : Word;
    sesi50_time          : Cardinal;
    sesi50_idle_time     : Cardinal;
    sesi50_protocol      : Byte;
    pad1                 : Byte;
end;

```

```
type
```

```

TFileInfo3 = packed record
    fi3_id              : DWORD;
    fi3_permissions     : DWORD;
    fi3_num_locks       : DWORD;
    fi3_pathname        : PWChar;
    fi3_username        : PWChar;
end;
PFileInfo3 = ^TFileInfo3;
TFileInfo3Array = array[0..512] of TFileInfo3;
PFileInfo3Array = ^TFileInfo3Array;

```

```
type
```

```

TFileInfo50 = packed record
    fi50_id             : Cardinal;
    fi50_permissions    : WORD;
    fi50_num_locks      : WORD;
    fi50_pathname       : PChar;
    fi50_username       : PChar;
    fi50_sharename      : PChar;
end;

```

```
type
```

```

TMibIfRow = packed record
    wszName              : array[0..255] of WideChar;
    dwIndex              : DWORD;
    dwType               : DWORD;
    dwMtu                : DWORD;
    dwSpeed              : DWORD;
    dwPhysAddrLen       : DWORD;
    bPhysAddr            : array[0..7] of Byte;
    dwAdminStatus       : DWORD;
    dwOperStatus        : DWORD;
    dwLastChange        : DWORD;
    dwInOctets          : DWORD;
    dwInUcastPkts       : DWORD;
    dwInNUCastePkts     : DWORD;
    dwInDiscards        : DWORD;
    dwInErrors          : DWORD;
    dwInUnknownProtos   : DWORD;
    dwOutOctets          : DWORD;
    dwOutUcastPkts      : DWORD;
    dwOutNUCastePkts    : DWORD;
    dwOutDiscards       : DWORD;
    dwOutErrors         : DWORD;
    dwOutQLen           : DWORD;
    dwDescrLen          : DWORD;
    bDescr              : array[0..255] of Char;
end;
TMibIfArray = array [0..512] of TMibIfRow;
PMibIfRow = ^TMibIfRow;
PMibIfArray = ^TMibIfArray;

```

```
type
```

```

TMibIfTable = packed record
    dwNumEntries        : DWORD;
    Table               : TMibIfArray;
end;
PMibIfTable = ^TMibIfTable;

```

```

var
Net_IIM_ShareEnumNT:function (      servername:PWChar;
                                level:DWORD;
                                bufptr:Pointer;
                                prefmaxlen:DWORD;
                                entriesread,
                                totalentries,
                                resume_handle:LPDWORD): DWORD; stdcall;

var
Net_IIM_ShareEnum:function ( pszServer   : PChar;
                             sLevel     : Cardinal;
                             pbBuffer   : Pchar;
                             cbBuffer   : Cardinal;
                             pcEntriesRead,
                             pcTotalAvail: Pointer):DWORD; stdcall;

var
Net_IIM_ShareDelNT:function (servername: PWideChar;
                             net_IIM_name: PWideChar;
                             reserved: DWORD): LongInt; stdcall;

var
Net_IIM_ShareDel:function ( pszServer,
                             pszNet_IIM_Name:PChar;
                             usReserved:Word): DWORD; stdcall;

var
Net_IIM_ShareAddNT: function(servername: PWideChar;
                             level: DWORD;
                             buf: Pointer;
                             parm_err: LPDWORD): DWORD; stdcall;

var
Net_IIM_ShareAdd: function ( pszServer:Pchar;
                             sLevel:Cardinal;
                             pbBuffer:PChar;
                             cbBuffer:Word):DWORD; stdcall;

Var
Net_IIM_SessionEnumNT:function(servername,
                                UncClientName,
                                username:PWChar;
                                level:DWORD;
                                bufptr:Pointer;
                                prefmaxlen:DWORD;
                                entriesread,
                                totalentries,
                                resume_handle:LPDWORD):DWORD; stdcall;

var
Net_IIM_SessionEnum:function(pszServer:PChar;
                             sLevel: DWORD;
                             pbBuffer:Pointer;
                             cbBuffer:DWORD;
                             pcEntriesRead,
                             pcTotalAvial:Pointer):integer; stdcall;

var
Net_IIM_SessionDelNT:function(ServerName,
                                UncClientName,
                                username:PWChar):DWORD; stdcall;

var
Net_IIM_SessionDel:function( pszServer:PChar;
                             pszClientName: PChar;
                             sReserved: SmallInt):DWORD; stdcall;

var

```

```

Net_IIM_FileEnumNT:function( servername,
                             basepath,
                             username:PWChar;
                             level:DWORD;
                             bufptr:Pointer;
                             prefmaxlen:DWORD;
                             entriesread,
                             totalentries,
                             resume_handle:LPDWORD):DWORD; stdcall;

var
Net_IIM_FileEnum:function(      pszServer,
                               pszBasePath:PChar;
                               sLevel:DWORD;
                               pbBuffer:Pointer;
                               cbBuffer:DWORD;
                               pcEntriesRead,
                               pcTotalAvail:pointer):integer; stdcall;

var
Net_IIM_FileClose:function(  ServerName:PWideChar;
                             FileId:DWORD):DWORD; stdcall;

var
Net_IIM_FileClose2:function( pszServer:PChar;
                              ulFileId:LongWord):DWORD; stdcall;

var
GetIfTable:function(      pIfTable      : PMibIfTable;
                          pdwSize      : PULONG;
                          bOrder      : Boolean ): DWORD; stdcall;

var
  MainForm: TMainForm;

implementation

{$R *.dfm}

{ TMainForm }

////////////////////////////////////
//
// Спочатку нам потрібно визначитися, під якою системою ми працюємо,
// щоб довідатися яку частину коду (для NT чи ні) використовувати в цей момент.
// Для цього напишемо невелику функцію, що і буде визначати тип системи.
//

function TMainForm.IsNT(var Value: Boolean): Boolean;
var Ver: TOSVersionInfo;
    BRes: Boolean;
begin
  Ver.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  BRes := GetVersionEx(Ver);
  if not BRes then //Перевірка
  begin
    Result := False; //Інформація не отримана
    Exit; //ідемо
  end else
    Result := True; //Інформація отримана

  case Ver.dwPlatformId of //визначаємося
    VER_PLATFORM_WIN32_NT      : Value := True; //Windows NT тьа вище -
    VER_PLATFORM_WIN32_WINDOWS : Value := False; //Windows 9 x-Me- підходить
    VER_PLATFORM_WIN32s       : Result := False //Windows 3.x- не підходить
  end;
end;
end;

```

```

////////////////////////////////////
//
// Одержання всіх відкритих загальних ресурсів досліджуємої мережі
//

procedure TMainForm.btnGetSharesClick(Sender: TObject);
var
  i:Integer;
  FLibHandle : THandle;
  ShareNT : PShareInfo2Array; //<= Змінні
  entriesread,totalentries:DWORD; //<= для Windows NT
  Share : array [0..512] of TShareInfo50; //<= Змінні
  pcEntriesRead,pcTotalAvail:Word; //<= для Windows 9 x-Me
  OS: Boolean;
begin
  lbxShares.Items.Clear;
  if not IsNT(OS) then Close; //Визначаємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NET_IIM_API32.DLL' ); //Завантажуємо бібліотеку
    if FLibHandle = 0 then Exit;
    //Зв' язуємо функцію
    @Net_IIM_ShareEnumNT := GetProcAddress(FLibHandle,' Net_IIM_ShareEnum' );
    if not Assigned(Net_IIM_ShareEnumNT) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    ShareNT := nil; //Очищаємо покажчик на масив структур
    //Виклик функції
    if Net_IIM_ShareEnumNT(nil,2,@ShareNT,DWORD(-1),
      @entriesread,@totalentries,nil) <> 0 then
    begin //Якщо виклик невдалий вивантажуємо бібліотеку
      FreeLibrary(FLibHandle);
      Exit;
    end;
    if entriesread > 0 then //Обробка результатів
    for i:= 0 to entriesread- 1 do
      lbxShares.Items.Add(String(ShareNT^[i].shi2_net_IIM_name));
    end else begin //Код для 9 x-me
      FLibHandle := LoadLibrary(' SVRAPI.DLL' ); //Завантажуємо бібліотеку
      if FLibHandle = 0 then Exit;
      //Зв' язуємо функцію
      @Net_IIM_ShareEnum := GetProcAddress(FLibHandle,' Net_IIM_ShareEnum' );
      if not Assigned(Net_IIM_ShareEnum) then //Перевірка
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
      //Виклик функції
      if Net_IIM_ShareEnum(nil,50,@Share,SizeOf(Share),
        @pcEntriesRead,@pcTotalAvail)<> 0 then
      begin //Якщо виклик невдалий вивантажуємо бібліотеку
        FreeLibrary(FLibHandle);
        Exit;
      end;
      if pcEntriesRead > 0 then //Обробка результатів
      for i:= 0 to pcEntriesRead- 1 do
        lbxShares.Items.Add(String(Share[i].shi50_net_IIM_name));
      end;
      FreeLibrary(FLibHandle); //Не забуваємо вивантажити бібліотеку
    end;

  //////////////////////////////////////
  //
  // Закриття загального ресурсу
  //

```

```

procedure TMainForm.btnCloseSharesClick(Sender: TObject);
var
  OS:Boolean;
  FLibHandle : THandle;
  Name9x:array [0..12] of Char;
  NameNT:PWChar;
  i:Integer;
  ShareName: String;
begin
  if not IsNT(OS) then Close; //Визначаємо тип системи

  if lbxShares.Items.Count = 0 then Exit;
  for i:= 0 to lbxShares.Items.Count-1 do
    if lbxShares.Selected[i] then Break; //Шукаємо обраний елемент
  if not lbxShares.Selected[i] then Exit; //Якщо не знайдений ідемо
  ShareName := lbxShares.Items.Strings[i];

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary( ' NET_IIM_API32.DLL' );
    if FLibHandle = 0 then Exit;
    @Net_IIM_ShareDelNT := GetProcAddress(FLibHandle, ' Net_IIM_ShareDel' );
    if not Assigned(Net_IIM_ShareDelNT) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    i:= SizeOf(WideChar)*256;
    GetMem(NameNT,i); //Виділяємо пам' ять під змінну
    StringToWideChar(ShareName,NameNT,i); //Перетворимо в PWideChar
    Net_IIM_ShareDelNT(nil,NameNT,0); //Видаляємо ресурс
    FreeMem(NameNT); //Звільняємо пам' ять
  end else begin //Код для 9 x-ме
    FLibHandle := LoadLibrary( ' SVRAPI.DLL' );
    if FLibHandle = 0 then Exit;
    @Net_IIM_ShareDel := GetProcAddress(FLibHandle, ' Net_IIM_ShareDel' );
    if not Assigned(Net_IIM_ShareDel) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FillChar(Name9x, SizeOf(Name9x), #0); //Очищаємо масив
    move(ShareName[1],Name9x[0],Length(ShareName)); //Заповнюємо масив
    Net_IIM_ShareDel(nil,@Name9x,0); //Видаляємо ресурс
  end;
  FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Показ діалогу вибору директорії
//

function TMainForm.SelectDirectory: String;
var
  lpItemID : PItemIDList;
  BrowseInfo : TBrowseInfo;
  DisplayName : array[0..MAX_PATH] of Char;
  TempPath : array[0..MAX_PATH] of Char;
begin
  FillChar(BrowseInfo, sizeof(TBrowseInfo), #0);
  BrowseInfo.hwndOwner := Handle;
  BrowseInfo.pszDisplayName := @DisplayName;
  BrowseInfo.lpszTitle := ' Specify a directory' ;
  BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;
  lpItemID := SHBrowseForFolder(BrowseInfo);
  if Assigned(lpItemID) then begin
    SHGetPathFromIDList(lpItemID, TempPath);
    GlobalFreePtr(lpItemID);
  end else Result := ' ' ;
end;

```

```

    Result := String(TempPath);
end;

////////////////////////////////////
//
// Додавання загального ресурсу
//

procedure TMainForm.btnAddSharesClick(Sender: TObject);
const
    STYPE_DISKTREE = 0;
    ACCESS_ALL = 258;
    SHI50F_FULL = 258;
var
    FLibHandle : THandle;
    Share9x : TShareInfo50;
    ShareNT : TShareInfo2;
    TmpDir, TmpName: String;
    TmpDirNT, TmpNameNT: PWChar;
    OS: Boolean;
    TmpLength: Integer;
begin
    TmpDir := SelectDirectory; //Визначаємо шлях до наступного ресурсу
    TmpName := InputBox(' Share name' , ' Enter name' , ' Test' ); //Визначаємо ім'
я під яким він буде видний у мережі
    if TmpDir = ' ' then Exit;

    if not IsNT(OS) then Close; //З' ясовуємо тип системи

    if OS then begin //Код для NT
        FLibHandle := LoadLibrary(' NET_IIM_API32.DLL' );
        if FLibHandle = 0 then Exit;
        @Net_IIM_ShareAddNT := GetProcAddress(FLibHandle, ' Net_IIM_ShareAdd' );
        if not Assigned(Net_IIM_ShareAddNT) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        TmpLength := SizeOf(WideChar)*256; //Визначаємо необхідний розмір

        GetMem(TmpNameNT, TmpLength); //Конвертуємо в PWChar
        StringToWideChar(TmpName, TmpNameNT, TmpLength);
        ShareNT.shi2_net_iim_name := TmpNameNT; //Ім' я

        ShareNT.shi2_type := STYPE_DISKTREE; //Тип ресурсу
        ShareNT.shi2_remark := ' '; //Коментар
        ShareNT.shi2_permissions := ACCESS_ALL; //Доступ
        ShareNT.shi2_max_uses := DWORD(-1); // Кількість максим. підключ.
        ShareNT.shi2_current_uses := 0; // Кількість тік підкл.

        GetMem(TmpDirNT, TmpLength);
        StringToWideChar(TmpDir, TmpDirNT, TmpLength);
        ShareNT.shi2_path := TmpDirNT; //Шлях до ресурсу

        ShareNT.shi2_passwd := ' '; //Пароль

        Net_IIM_ShareAddNT(nil, 2, @ShareNT, nil); //Додаємо ресурс
        FreeMem (TmpNameNT); //звільняємо пам' ять
        FreeMem (TmpDirNT);
    end else begin
        FLibHandle := LoadLibrary(' SVRAPI.DLL' );
        if FLibHandle = 0 then Exit;
        @Net_IIM_ShareAdd := GetProcAddress(FLibHandle, ' Net_IIM_ShareAdd' );
        if not Assigned(Net_IIM_ShareAdd) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
    end;
end;

```

```

    FillChar(Share9x.shi50_net_IIM_name, SizeOf(Share9x.shi50_net_IIM_name),
#0);
    move(TmpName[1],Share9x.shi50_net_IIM_name[0],Length(TmpName)); //Ім'я
    Share9x.shi50_type := STYPE_DISKTREE; //Тип ресурсу
    Share9x.shi50_flags := SHI50F_FULLL; //Доступ
    FillChar(Share9x.shi50_remark,
        SizeOf(Share9x.shi50_remark), #0); //Коментар
    FillChar(Share9x.shi50_path,
        SizeOf(Share9x.shi50_path), #0);
    Share9x.shi50_path := PAnsiChar(TmpDir); //Шлях до ресурсу
    FillChar(Share9x.shi50_rw_password,
        SizeOf(Share9x.shi50_rw_password), #0); //Пароль повного доступу
    FillChar(Share9x.shi50_ro_password,
        SizeOf(Share9x.shi50_ro_password), #0); //Пароль для читання
    Net_IIM_ShareAdd(nil,50,@Share9x,SizeOf(Share9x));
end;
FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Помітьте що активний і неактивний час сесій буде даватися нам
// у вигляді кіл-ті секунд (тип Cardinal). Напишемо невелику
// функцію, задача якої буде перетворювати кількість секунд у більше
// звичну форму відображення.
//

function TMainForm.CardinalToTimeStr(Value: Cardinal): String;
var d,h,m,s: Real;
begin
    d:=0;
    h:=0;
    m:=0;
    s:=Value;
    if s > 59 then begin
        m:=int(s / 60);
        s:= s-s-(m*60);
    end;
    if m > 59 then begin
        h:=int(m/60);
        m:= m-m-(h*60);
    end;
    if h > 23 then begin
        d:=int(h/24);
        h:= h-h-(d*24);
    end;
    Result:='';
    if (d>0) then Result:=Result+floattostr(d)+' d. ';
    if (h<9) then Result:=Result+' 0' +floattostr(h)+' :' else
Result:=Result+floattostr(h)+' :';
    if (m<9) then Result:=Result+' 0' +floattostr(m)+' :' else
Result:=Result+floattostr(m)+' :';
    if (s<9) then Result:=Result+' 0' +floattostr(s) else
Result:=Result+floattostr(s);
end;

////////////////////////////////////
//
// Одержання списку сесій досліджуємої мережі для оцінки якості обслуговування
(QoS) системи інтелектуального управління IT-інфраструктурою
//

procedure TMainForm.btnGetSessionsClick(Sender: TObject);
var
    OS: Boolean;
    FLibHandle : THandle;
    SessionInfo50: array [0..512] of TSessionInfo50;
    SessionInfo502 : PSessionInfo502Array;
    TotalEntries,EntriesReadNT: DWORD;

```

```

EntriesRead, TotalAvial: Word;
i: integer;
begin
  lvSessions.Items.Clear;

  if not IsNT(OS) then Close; //З'ясовуємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary('NET_IIM_API32.DLL');
    if FLibHandle = 0 then Exit;
    @Net_IIM_SessionEnumNT := GetProcAddress(FLibHandle, 'Net_IIM_SessionEnum');
  );
  if not Assigned(Net_IIM_SessionEnumNT) then
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  SessionInfo502 := nil;
  if Net_IIM_SessionEnumNT(nil, nil, nil, 502, @SessionInfo502, DWORD(-1), @entriesreadNT, @totalentries, nil) = 0 then
  for i:=0 to EntriesReadNT-1 do
  begin
    with lvSessions.Items.Add do //Заповнення даними зі структури
    begin
      Caption := string(SessionInfo502^[i].sesi502_cname); //Ім'я комп'ютера
      SubItems.Add(SessionInfo502^[i].sesi502_username); //Ім'я користувача
      SubItems.Add(IntToStr(SessionInfo502^[i].sesi502_num_opens));
    //Відкритих ресурсів
      SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].Sesi502_Time)); //Час активний
      SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
    //Час не активний
    end;
  end;
end else begin //Код для Windows 9 x-Me
  FLibHandle := LoadLibrary('SVRAPI.DLL');
  if FLibHandle = 0 then Exit;
  @Net_IIM_SessionEnum := GetProcAddress(FLibHandle, 'Net_IIM_SessionEnum');
  if not Assigned(Net_IIM_SessionEnum) then
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  if Net_IIM_SessionEnum
  (nil, 50, @SessionInfo50, SizeOf(SessionInfo50), @EntriesRead, @TotalAvial) = 0 then
  for i:=0 to EntriesRead-1 do
  begin
    with lvSessions.Items.Add do //Заповнення даними зі структури
    begin
      Caption := string(SessionInfo50[i].Sesi50_cname); //Ім'я комп'ютера
      досліджуємої мережі для оцінки якості обслуговування (QoS) системи
      інтелектуального управління IT-інфраструктурою
      SubItems.Add(SessionInfo50[i].Sesi50_username); //Ім'я користувача
      SubItems.Add(IntToStr(SessionInfo50[i].sesi50_num_opens)); //Відкритих
      ресурсів
      SubItems.Add(CardinalToTimeStr(SessionInfo50[i].Sesi50_Time)); //Час
      активний
      SubItems.Add(CardinalToTimeStr(SessionInfo50[i].sesi50_idle_time));
    //Час не активний
      SessionCloseKey[i] := SessionInfo50[i].sesi50_key; //Унікальний
      ідентифікатор для закриття
    end;
  end;
end;
  FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//

```

```

// Завершення обраної сесії
//

procedure TMainForm.btnCloseSessionClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  CNameNT: PWideChar;
  CName9x: PAnsiChar;
  Key:SmallInt;
  i: Integer;
begin
  if not IsNT(OS) then Close; //3' ясовуємо тип системи

  if not Assigned(lvSessions.Selected) then Exit;
  i:= lvSessions.Selected.Index; //Визначаємо номер обраної сесії

  if OS then begin
    FLibHandle := LoadLibrary(' NET_IIM_API32.DLL' );
    if FLibHandle = 0 then Exit;
    @Net_IIM_SessionDelNT := GetProcAddress(FLibHandle, ' Net_IIM_SessionDel' );
    if not Assigned(Net_IIM_SessionDelNT) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    //Перетворимо дані в необхідний вид
    CNameNT := PWChar(WideString('\ ' +lvSessions.Items.Item[i].Caption));
    Net_IIM_SessionDelNT(nil,CNameNT,nil);
  end else begin
    FLibHandle := LoadLibrary(' SVRAPI.DLL' );
    if FLibHandle = 0 then Exit;
    @Net_IIM_SessionDel := GetProcAddress(FLibHandle, ' Net_IIM_SessionDel' );
    if not Assigned(Net_IIM_SessionDel) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    //Перетворимо дані в необхідний вид
    CName9x := PAnsiChar(lvSessions.Items.Item[i].Caption);
    key := SessionCloseKey[i]; //Беремо ключ із масиву
    Net_IIM_SessionDel(nil,CName9x,Key);
  end;
  FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Одержання списку відкритих файлів досліджуємої мережі для оцінки якості
// обслуговування (QoS) системи інтелектуального управління IT-інфраструктурою
//

procedure TMainForm.btnGetFilesClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  FileInfoNT: PFileInfo3Array;
  FileInfo9x: array [0..512] of TFileInfo50;
  TotalEntries,EntriesReadNT: DWORD;
  EntriesRead,TotalAvial: Word;
  i:integer;
begin
  lvfiles.Items.Clear;

  if not IsNT(OS) then Close; //3' ясовуємо тип системи

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NET_IIM_API32.DLL' );
    if FLibHandle = 0 then Exit;

```

```

@Net_IIM_FileEnumNT := GetProcAddress(FLibHandle, ' Net_IIM_FileEnum' );
if not Assigned(Net_IIM_FileEnumNT) then
begin
  FreeLibrary(FLibHandle);
  Exit;
end;
FileInfoNT := nil;
if Net_IIM_FileEnumNT(nil, nil, nil, 3, @FileInfoNT, DWORD(-1), @entriesreadNT,
@totalentries, nil)=0 then
for i:=0 to EntriesReadNT-1 do
begin
  with lvFiles.Items.Add do //Заповнення даними зі структури
  begin
    Caption := string(IntToStr(FileInfoNT^[i].fi3_id)); //Ідентифікатор
    SubItems.Add(FileInfoNT^[i].fi3_pathname); //Шлях до файлу
    SubItems.Add(FileInfoNT^[i].fi3_username); //Ім'я користувача
  end;
end;
end else begin //Код для Windows 9 x-Me
FLibHandle := LoadLibrary(' SVRAPI.DLL' );
if FLibHandle = 0 then Exit;
@Net_IIM_FileEnum := GetProcAddress(FLibHandle, ' Net_IIM_FileEnum' );
if not Assigned(Net_IIM_FileEnum) then
begin
  FreeLibrary(FLibHandle);
  Exit;
end;
if Net_IIM_FileEnum (nil,
nil, 50, @FileInfo9x, SizeOf(FileInfo9x), @EntriesRead, @TotalAvial)= 0 then
for i:=0 to EntriesRead-1 do
begin
  with lvFiles.Items.Add do //Заповнення даними зі структури
  begin
    Caption := string(IntToStr(FileInfo9x[i].fi50_id)); //Ідентифікатор
    SubItems.Add(FileInfo9x[i].fi50_pathname); //Шлях до файлу
    SubItems.Add(FileInfo9x[i].fi50_username); //Ім'я користувача
  end;
end;
end;
FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
// Закриття файлу
//

procedure TMainForm.btnCloseFileClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  i: Integer;
begin
  if not IsNT(OS) then Close; //З'ясовуємо тип системи
  if not Assigned(lvFiles.Selected) then Exit;
  i:= lvFiles.Selected.Index; //Визначаємо номер обраного файлу

  if OS then begin //Код для NT
    FLibHandle := LoadLibrary(' NET_IIM_API32.DLL' );
    if FLibHandle = 0 then Exit;
    @Net_IIM_FileClose := GetProcAddress(FLibHandle, ' Net_IIM_FileClose' );
    if not Assigned(Net_IIM_FileClose) then
    begin
      FreeLibrary(FLibHandle);
      Close;
    end;
    Net_IIM_FileClose(nil, StrToInt(lvFiles.Items.Item[i].Caption)); //Закриваємо
    файл
  end;
end;

```

```

end else begin //Код для Windows 9 x-Me
  FLibHandle := LoadLibrary( ' SVRAPI.DLL' );
  if FLibHandle = 0 then Exit;
  @Net_IIM_FileClose2 := GetProcAddress(FLibHandle, ' Net_IIM_FileClose2' );
  if not Assigned(Net_IIM_FileClose2) then
  begin
    FreeLibrary(FLibHandle);
    Close;
  end;
  Net_IIM_FileClose2(nil, StrToInt(lvFiles.Items.Item[i].Caption));
end;
FreeLibrary(FLibHandle);
end;

////////////////////////////////////
//
//  Визначаємо вхідний / вихідний трафік досліджуємої мережі для оцінки якості
//  обслуговування (QoS) системи інтелектуального управління IT-інфраструктурою
//

procedure TMainForm.tmrTrafficTimer(Sender: TObject);
// Допоміжна функція, що перетворить MAC адресу до "нормального" виду
//Визначаємо спеціальний тип, щоб можна було передати у функцію масив
type TMAC = array [0..7] of Byte;
//Як перше значення масив, друге значення, розмір даних у масиві
function GetMAC(Value: TMAC; Length: DWORD): String;
var
  i: Integer;
begin
  if Length = 0 then Result := ' 00-00-00' else
  begin
    Result := ' ';
    for i:= 0 to Length-2 do
      Result := Result + IntToHex(Value[i],2)+' -' ;
    Result := Result + IntToHex(Value[ Length-1],2);
  end;
end;

//Сама процедура
var
  FLibHandle : THandle;
  Table: TMibIfTable;
  i : integer;
  Size : integer;
begin
  tmrTraffic.Enabled := false; //Припиняємо таймер
  lvTraffic.Items.BeginUpdate;
  lvTraffic.Items.Clear; //Очищаємо список
  FLibHandle := LoadLibrary( ' IPHLPAPI.DLL' ); //Завантажуємо бібліотеку
  if FLibHandle = 0 then Exit;
  @GetIfTable := GetProcAddress(FLibHandle, ' GetIfTable' );
  if not Assigned(GetIfTable) then
  begin
    FreeLibrary(FLibHandle);
    Close;
  end;

  Size := SizeOf(Table);
  if GetIfTable(@Table, @Size, false ) = 0 then //Виконуємо функцію
  for i:= 0 to Table.dwNumEntries-1 do begin
    with lvTraffic.Items.Add do begin //Виводимо результати
      Caption := String(Table.Table[i].bDescr); //Найменування інтерфейсу
      SubItems.Add(GetMAC(TMAC(Table.Table[i].bPhysAddr),
        Table.Table[i].dwPhysAddrLen)); //MAC адреса
      SubItems.Add(IntToStr(Table.Table[i].dwInOctets)); //Усього прийнято
      байт з досліджуємої мережі для оцінки якості обслуговування (QoS) системи
      інтелектуального управління IT-інфраструктурою
    end;
  end;
end;

```

```

        SubItems.Add(IntToStr(Table.Table[i].dwOutOctets)); //Усього відправлено
байт у досліджуєму мережу для оцінки якості обслуговування (QoS) системи
інтелектуального управління IT-інфраструктурою
        end;
    end;
    lvTraffic.Items.EndUpdate;
    FreeLibrary(FLibHandle);
    tmrTraffic.Enabled := true; //Не забуваємо активувати таймер
end;

function OpenEnum(const Net_IIM_ContainerToOpen: PNet_IIM_Resource; ResScope,
ResType, ResUsage: DWORD): THandle;
var
    hNet_IIM_Enum: THandle;
begin
    Result:=0;
    if (NO_ERROR<>WNet_IIM_OpenEnum(ResScope, ResType, ResUsage,
        Net_IIM_ContainerToOpen, hNet_IIM_Enum))
    then ShowMessage(' Помилка!' )
    else Result:=hNet_IIM_Enum;
end;

function EnumResources(const ParentNode: TTreeNode;
ResScope, ResType, ResUsage: DWORD; hNet_IIM_Enum: THandle): UINT;
function ShowResource(const ParentNode: TTreeNode; Res: TNet_IIM_Resource):
TTreeNode;
begin
    Result:=MainForm.Net_IIM_Tree.Items.AddChild(ParentNode,
string(Res.lpRemoteName));
end;

const
    RESOURCE_BUF_ENTRIES = 2000;

var
    ResourceBuffer: array[1..RESOURCE_BUF_ENTRIES] of TNet_IIM_Resource;
    i, ResourceBuf, EntriesToGet: dword;
    NewNode: TTreeNode;
begin
    Result:=0;
    while true do
        begin
            ResourceBuf:=sizeof(ResourceBuffer);
            EntriesToGet:=RESOURCE_BUF_ENTRIES;
            if (NO_ERROR<>WNet_IIM_EnumResource(hNet_IIM_Enum, EntriesToGet,
                @ResourceBuffer, ResourceBuf))
            then
                begin
                    case GetLastError() of
                        NO_ERROR: // проход буферу без перемикання
                            Break;
                        ERROR_NO_MORE_ITEMS:
                            // Повертає 0 у тому випадку, коли останов
                            // RESOURCE_BUF_ENTRIES дані на попередньому виклику, щоб
                            // WNet_IIM_EnumResource, та були точно
                            // RESOURCE_BUF_ENTRIES дані в запису на момент
                            // попереднього виклику
                            Exit;
                    else ShowMessage(Помилка!' );
                    Result:=1;
                    Exit;
                end;
            end;
        end;
    for i:=1 to EntriesToGet do
        begin
            NewNode:=ShowResource(ParentNode, ResourceBuffer[i]);
            if (ResourceBuffer[i].dwUsage and RESOURCEUSAGE_CONTAINER)<>0

```

```

        then MainForm.Open_Do_Close_Enum(NewNode, ResScope, ResType, ResUsage,
@ResourceBuffer[i]);
        Application.ProcessMessages;
    end;
end;
end;

procedure TMainForm.Open_Do_Close_Enum(const ParentNode: TTreeNode; ResScope,
ResType, ResUsage: DWORD; const Net_IIM_ContainerToOpen: PNet_IIM_Resource);
var
    hNet_IIM_Enum: THandle;
begin
    hNet_IIM_Enum:=OpenEnum(Net_IIM_ContainerToOpen, ResScope, ResType, ResUsage);
    if (hNet_IIM_Enum=0)
    then Exit;
    EnumResources(ParentNode, ResScope, ResType, ResUsage, hNet_IIM_Enum);
    if (NO_ERROR<>WNet_IIM_CloseEnum(hNet_IIM_Enum))
    then ShowMessage(' WNet_IIM_CloseEnum Помилка' );
end;

procedure TMainForm.Button1Click(Sender: TObject);
var
    ResScope, ResType, ResUsage: dword;
begin
    Button1.Caption:=' Пошук мережних ресурсів. Чекайте...' ;
    Button1.Enabled:=false;
    //
    Net_IIM_Tree.Items.Clear;
    case rgScope.ItemIndex of
        1: ResScope:=RESOURCE_GLOBALNET_IIM_;
        2: ResScope:=RESOURCE_REMEMBERED;
        else ResScope:=RESOURCE_CONNECTED;
    end;
    ResType:=0;
    if cbTypeAny.Checked
    then ResType:=ResType or RESOURCETYPE_ANY;
    if cbTypeDisk.Checked
    then ResType:=ResType or RESOURCETYPE_DISK;
    if cbTypePrint.Checked
    then ResType:=ResType or RESOURCETYPE_PRINT;
    ResUsage:=0;
    if cbUsageConnectable.Checked
    then ResUsage:=ResUsage or RESOURCEUSAGE_CONNECTABLE;
    if cbUsageContainer.Checked
    then ResUsage:=ResUsage or RESOURCEUSAGE_CONTAINER;
    Open_Do_Close_Enum(Net_IIM_Tree.Items.Add(nil, ' Net_IIM_work Resources' ),
        ResScope, ResType, ResUsage, nil);
    //
    Button1.Caption:=' Обновити список ресурсів' ;
    Button1.Enabled:=true;
end;

procedure TMainForm.Net_IIM_TreeCustomDrawItem(Sender: TCustomTreeView;
Node: TTreeNode; State: TCustomDrawState; var DefaultDraw: Boolean);
begin
    if cdsSelected in State
    then Sender.Canvas.Font.Style:=Sender.Canvas.Font.Style+[fsUnderline];
end;

procedure TMainForm.Net_IIM_TreeDbClick(Sender: TObject);
begin
    ShellExecute(0, ' open' , PChar(Net_IIM_Tree.Selected.Text), ' \ , ' \ , SW_SHOW);
end;

procedure TMainForm.Net_IIM_TreeGetImageIndex(Sender: TObject; Node: TTreeNode);
begin
    if Node.HasChildren
    then Node.ImageIndex:=1

```

```
    else Node.ImageIndex:=0;
end;

procedure TMainForm.Button4Click(Sender: TObject);
begin
Form1.Show;
end;

procedure TMainForm.Button2Click(Sender: TObject);
begin
Form2.Show;
end;

procedure TMainForm.Button3Click(Sender: TObject);
begin
Form3.Show;
end;

end.
```

Кафедра КБПЗ – 2021 рік

Основна програма

Файл IT_infrastructure_management.dpr основної програми

```
program IT_infrastructure_management;

uses
  Forms,
  Main in 'Main.pas' {MainForm},
  About in 'About.pas' {Form1},
  TCP_IP in 'TCP_IP.pas' {Form2},
  Stat in 'Stat.pas' {Form3};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TMainForm, MainForm);
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.Run;
end.
```

Кафедра_КБПЗ_2021_рік

Файл IPHLPAPI.pas- обробка API функцій

```

unit IPHLPAPI;

interface
uses
  Windows, winsock;

const
  VERSION      = ' 1.5' ;

//----- Заголовок з Microsoft IPTYPES.H-----

const
  ANY_SIZE      = 1;
  MAX_ADAPTER_DESCRIPTION_LENGTH = 128; // arb.
  MAX_ADAPTER_NAME_LENGTH = 256; // змінна
  MAX_ADAPTER_ADDRESS_LENGTH = 8; // змінна
  DEFAULT_MINIMUM_ENTITIES = 32; // змінна
  MAX_HOSTNAME_LEN = 128; // змінна
  MAX_DOMAIN_NAME_LEN = 128; // змінна
  MAX_SCOPE_ID_LEN = 256; // змінна

  // Вузлові типи ( NET_IIM_BIOS)
  BROADCAST_NODETYPE = 1;
  PEER_TO_PEER_NODETYPE = 2;
  MIXED_NODETYPE = 4;
  HYBRID_NODETYPE = 8;

  NET_IIM_BIOSTypes : array[0..8] of string[20] =
    ( ' Невизначений' , ' Передача' , ' Рівень до рівня' , ' ' , '
Змішаний' , ' ' , ' ' , ' ' , ' гібрид'
    );

  // Типи адаптеру
  { v1.4-> 1.5
  IF_OTHER_ADAPTERTYPE = 0;
  IF_ETHERNET_IIM_ADAPTERTYPE = 1;
  IF_TOKEN_RING_ADAPTERTYPE = 2;
  IF_FDDI_ADAPTERTYPE = 3;
  IF_PPP_ADAPTERTYPE = 4;
  IF_LOOPBACK_ADAPTERTYPE = 5;
  IF_SLIP_ADAPTERTYPE = 6;

  Знайдено у ipifcons.h :
  #define MIB_IF_TYPE_OTHER 1
  #define MIB_IF_TYPE_ETHERNET_IIM_ 6
  #define MIB_IF_TYPE_TOKENRING 9
  #define MIB_IF_TYPE_FDDI 15
  #define MIB_IF_TYPE_PPP 23
  #define MIB_IF_TYPE_LOOPBACK 24
  #define MIB_IF_TYPE_SLIP 28
  }
  IF_OTHER_ADAPTERTYPE = 1;
  IF_ETHERNET_IIM_ADAPTERTYPE = 6;
  IF_TOKEN_RING_ADAPTERTYPE = 9;
  IF_FDDI_ADAPTERTYPE = 15;
  IF_PPP_ADAPTERTYPE = 23;
  IF_LOOPBACK_ADAPTERTYPE = 24;
  IF_SLIP_ADAPTERTYPE = 28;

  // AdaptTypes : array[0..6] of string[10] =
  // ( ' інший' , ' ethernet_IIM_' , ' tokenring' , ' FDDI' , ' PPP' , '
loopback' , ' SLIP' );
  AdaptTypes : array[1..28] of string[10] =

```

```

( 'інший' , ' ' , ' ' , ' ' , ' ' , ' ' , ' ethernet_IIM_' , ' ' , ' ' , '
tokenring' , ' ' , ' ' , ' ' , ' ' , ' ' , ' FDDI' , ' ' , ' ' , ' ' , ' ' , '
' ' , ' ' , ' PPP' , ' ' , ' ' , ' ' , ' ' , ' SLIP' );
// Кінець змін в типі адаптерів

//-----для інших MS заготовочних файлів-----

MAX_INTERFACE_NAME_LEN = 256; { mrapi.h }
MAXLEN_PHYSADDR = 8; { iprtmib.h }
MAXLEN_IFDESCR = 256; {"--      }

//-----

type
  TMacAddress = array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte;

//---IP адресні структури-----

PTIP_ADDRESS_STRING = ^TIP_ADDRESS_STRING;
TIP_ADDRESS_STRING = array[0..15] of char; // IP рядок
//
PTIP_ADDR_STRING = ^TIP_ADDR_STRING;
TIP_ADDR_STRING = packed record // для використання у зв'язних списках
  Next: PTIP_ADDR_STRING;
  IpAddress: TIP_ADDRESS_STRING;
  IpMask: TIP_ADDRESS_STRING;
  Context: DWORD;
end;

//-----Fixed Info структура-----

PTFixedInfo = ^TFixedInfo;
TFixedInfo = packed record
  HostName: array[1..MAX_HOSTNAME_LEN + 4] of char; // дані
  DomainName: array[1..MAX_DOMAIN_NAME_LEN + 4] of char; // дані
  CurrentDNSServer: PTIP_ADDR_STRING;
  DNSServerList: TIP_ADDR_STRING;
  NodeType: UINT;
  ScopeID: array[1..MAX_SCOPE_ID_LEN + 4] of char; // дані
  EnableRouting: UINT;
  EnableProxy: UINT;
  EnableDNS: UINT;
end;

//-----структура мережного інтерфейсу досліджуємої мережі для оцінки
якості обслуговування (QoS) системи інтелектуального управління IT-
інфраструктурою-----
////////////////////////////////////
//
//
// Наступне є діючими станами для WAN да LAN інтерфейсів. //
// Порядок станів створений для визначення. Для //
// стану >= CONNECTED можливо передавати дані зразу. Стан >= DISCONNECTED
//
// може передавати деякі дані. Стан < DISCONNECTED може //
// не передавати дані.
//
// карта з поміткою UNREACHABLE якщо DIM викликає InterfaceUnreachable для
//
// причин. Крім невдачі з'єднання. //
//
//
// NON_OPERATIONAL- Перевірка для LAN інтерфейсу. Позначає карту що не
працює //
// або не з'єднується з картою. //

```



```

    dwInErrors: DWORD;
    dwInUnknownProtos: DWORD;
    dwOutOctets: DWORD;
    dwOutUCastPkts: DWORD;
    dwOutNUCastPkts: DWORD;
    dwOutDiscards: DWORD;
    dwOutErrors: DWORD;
    dwOutQLen: DWORD;
    dwDescrLen: DWORD;
    bDescr: array[1..MAXLEN_IFDESCR] of char; //byte;
end;

//
PTMibIfTable = ^TMIBIfTable;
TMibIfTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIfRow;
end;

//---ADAPTER INFO структура-----

PTIP_ADAPTER_INFO = ^TIP_ADAPTER_INFO;
TIP_ADAPTER_INFO = packed record
    Next: PTIP_ADAPTER_INFO;
    ComboIndex: DWORD;
    AdapterName: array[1..MAX_ADAPTER_NAME_LENGTH + 4] of char; //
дані
    Description: array[1..MAX_ADAPTER_DESCRIPTION_LENGTH + 4] of char;
// дані
    AddressLength: UINT;
    Address: array[1..MAX_ADAPTER_ADDRESS_LENGTH] of byte; // дані
    Index: DWORD;
    aType: UINT;
    DHCPEnabled: UINT;
    CurrentIPAddress: PTIP_ADDR_STRING;
    IPAddressList: TIP_ADDR_STRING;
    GatewayList: TIP_ADDR_STRING;
    DHCPServer: TIP_ADDR_STRING;
    HaveWINS: BOOL;
    PrimaryWINSServer: TIP_ADDR_STRING;
    SecondaryWINSServer: TIP_ADDR_STRING;
    LeaseObtained: LongInt ; // UNIX час, секунди з 1970
    LeaseExpires: LongInt; // UNIX час, секунди з 1970
    SpareStuff: array [1..200] of char ; // дані- простір для списку IP
адрес досліджуємої мережі для оцінки якості обслуговування (QoS) системи
інтелектуального управління IT-інфраструктурою
end;

//-----TCP структура-----

PTMibTCPRow = ^TMibTCPRow;
TMibTCPRow = packed record
    dwState: DWORD;
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
    dwRemoteAddr: DWORD;
    dwRemotePort: DWORD;
end;
//
PTMibTCPTable = ^TMibTCPTable;
TMibTCPTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..0] of TMibTCPRow;
end;
//
PTMibTCPStats = ^TMibTCPStats;
TMibTCPStats = packed record
    dwRTOAlgorithm: DWORD;
    dwRTOMin: DWORD;

```

```

    dwRTOMax: DWORD;
    dwMaxConn: DWORD;
    dwActiveOpens: DWORD;
    dwPassiveOpens: DWORD;
    dwAttemptFails: DWORD;
    dwEstabResets: DWORD;
    dwCurrEstab: DWORD;
    dwInSegs: DWORD;
    dwOutSegs: DWORD;
    dwRetransSegs: DWORD;
    dwInErrs: DWORD;
    dwOutRsts: DWORD;
    dwNumConns: DWORD;
end;

//-----UDP CTPVKTYPA -----

PTMibUDPRow = ^TMibUDPRow;
TMibUDPRow = packed record
    dwLocalAddr: DWORD;
    dwLocalPort: DWORD;
end;
//
PTMibUDPTable = ^TMIBUDPTable;
TMIBUDPTable = packed record
    dwNumEntries: DWORD;
    UDPTable: array[0..ANY_SIZE- 1] of TMibUDPRow;
end;
//
PTMibUdpStats = ^TMIBUdpStats;
TMIBUdpStats = packed record
    dwInDatagrams: DWORD;
    dwNoPorts: DWORD;
    dwInErrors: DWORD;
    dwOutDatagrams: DWORD;
    dwNumAddrs: DWORD;
end;

//-----IP CTPVKTYPA -----

//
PTMibIPNet_IIM_Row = ^TMibIPNet_IIM_Row;
TMibIPNet_IIM_Row = packed record
    dwIndex: DWord;
    dwPhysAddrLen: DWord;
    bPhysAddr: TMacAddress;
    dwAddr: DWord;
    dwType: DWord;
end;
//
PTMibIPNet_IIM_Table = ^TMibIPNet_IIM_Table;
TMibIPNet_IIM_Table = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPNet_IIM_Row;
end;
//
PTMibIPStats = ^TMibIPStats;
TMibIPStats = packed record
    dwForwarding: DWORD;
    dwDefaultTTL: DWORD;
    dwInReceives: DWORD;
    dwInHdrErrors: DWORD;
    dwInAddrErrors: DWORD;
    dwForwDatagrams: DWORD;
    dwInUnknownProtos: DWORD;
    dwInDiscards: DWORD;
    dwInDelivers: DWORD;
    dwOutRequests: DWORD;
    dwRoutingDiscards: DWORD;

```

```

    dwOutDiscards: DWORD;
    dwOutNoRoutes: DWORD;
    dwReasmTimeOut: DWORD;
    dwReasmReqds: DWORD;
    dwReasmOKs: DWORD;
    dwReasmFails: DWORD;
    dwFragOKs: DWORD;
    dwFragFails: DWORD;
    dwFragCreates: DWORD;
    dwNumIf: DWORD;
    dwNumAddr: DWORD;
    dwNumRoutes: DWORD;
end;
//
PTMibIPAddrRow = ^TMibIPAddrRow;
TMibIPAddrRow = packed record
    dwAddr: DWORD;
    dwIndex: DWORD;
    dwMask: DWORD;
    dwBCastAddr: DWORD;
    dwReasmSize: DWORD;
    Unused1,
    Unused2: WORD;
end;
//
PTMibIPAddrTable = ^TMibIPAddrTable;
TMibIPAddrTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPAddrRow;
end;

//
PTMibIPForwardRow = ^TMibIPForwardRow;
TMibIPForwardRow = packed record
    dwForwardDest: DWORD;
    dwForwardMask: DWORD;
    dwForwardPolicy: DWORD;
    dwForwardNextHop: DWORD;
    dwForwardIFIndex: DWORD;
    dwForwardType: DWORD;
    dwForwardProto: DWORD;
    dwForwardAge: DWORD;
    dwForwardNextHopAS: DWORD;
    dwForwardMetric1: DWORD;
    dwForwardMetric2: DWORD;
    dwForwardMetric3: DWORD;
    dwForwardMetric4: DWORD;
    dwForwardMetric5: DWORD;
end;
//
PTMibIPForwardTable = ^TMibIPForwardTable;
TMibIPForwardTable = packed record
    dwNumEntries: DWORD;
    Table: array[0..ANY_SIZE- 1] of TMibIPForwardRow;
end;

//----- ICMP-CTPYKTYPA -----

PTMibICMPStats = ^TMibICMPStats;
TMibICMPStats = packed record
    dwMsgs: DWORD;
    dwErrors: DWORD;
    dwDestUnreachs: DWORD;
    dwTimeEcxcds: DWORD;
    dwParmProbs: DWORD;
    dwSrcQuenchs: DWORD;
    dwRedirects: DWORD;
    dwEchos: DWORD;
    dwEchoReps: DWORD;

```

```

    dwTimeStamps: DWORD;
    dwTimeStampReps: DWORD;
    dwAddrMasks: DWORD;
    dwAddrReps: DWORD;
end;

PTMibICMPInfo = ^TMibICMPInfo;
TMibICMPInfo = packed record
    InStats: TMibICMPStats;
    OutStats: TMibICMPStats;
end;

//-----импорт до IPHLPAPI.DLL-----

var

GetAdaptersInfo: function ( pAdapterInfo: PTIP_ADAPTER_INFO;
    pOutBufLen: PULONG ): DWORD; stdcall;

GetNet_IIM_workParams: function ( FixedInfo: PTFixedInfo; pOutPutLen:
PULONG ):
    DWORD; stdcall;

GetTcpTable: function ( pTCPTable: PTMibTCPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD; stdcall;

GetTcpStatistics: function ( pStats: PTMibTCPStats ): DWORD; stdcall;

GetUdpTable: function ( pUdpTable: PTMibUDPTable; pDwSize: PDWORD;
    bOrder: BOOL ): DWORD; stdcall;

GetUdpStatistics: function ( pStats: PTMibUdpStats ): DWORD; stdcall;

GetIpStatistics: function ( pStats: PTMibIPStats ): DWORD; stdcall;

GetIpNet_IIM_Table: function ( pIpNet_IIM_Table: PTMibIPNet_IIM_Table;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdcall;

GetIpAddrTable: function ( pIpAddrTable: PTMibIPAddrTable;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdcall;

GetIpForwardTable: function ( pIPForwardTable: PTMibIPForwardTable;
    pdwSize: PULONG; bOrder: BOOL ): DWORD; stdCall;

GetIcmpStatistics: function ( pStats: PTMibICMPInfo ): DWORD; stdCall;

GetRTTAndHopCount: function ( DestIPAddress: DWORD; HopCount: PULONG;
    MaxHops: ULONG; RTT: PULONG ): BOOL; stdCall;

GetIfTable: function ( pIfTable: PTMibIfTable; pdwSize: PULONG;
    bOrder: boolean ): DWORD; stdCall;

GetIfEntry: function ( pIfRow: PTMibIfRow ): DWORD; stdCall;

// попередження - недокументована функція, можливі помилки при
використанні
GetFriendlyIfIndex: function (var IfIndex: DWORD): DWORD; stdcall;

const
    IpHlpDLL = ' IPHLPAPI.DLL' ;
var
    IpHlpModule: THandle;

    function LoadIpHlp: Boolean;

implementation

function LoadIpHlp: Boolean;

```

```

begin
    Result := True;
    if IpHlpModule <> 0 then Exit;

    // відкрити DLL
    IpHlpModule := LoadLibrary (IpHlpDLL);
    if IpHlpModule = 0 then
    begin
        Result := false;
        exit ;
    end ;
    GetAdaptersInfo := GetProcAddress (IpHlpModule, ' GetAdaptersInfo' ) ;
    GetNet_IIM_workParams := GetProcAddress (IpHlpModule, '
GetNet_IIM_workParams' ) ;
    GetTcpTable := GetProcAddress (IpHlpModule, ' GetTcpTable' ) ;
    GetTcpStatistics := GetProcAddress (IpHlpModule, ' GetTcpStatistics' )
;
    GetUdpTable := GetProcAddress (IpHlpModule, ' GetUdpTable' ) ;
    GetUdpStatistics := GetProcAddress (IpHlpModule, ' GetUdpStatistics' )
;
    GetIpStatistics := GetProcAddress (IpHlpModule, ' GetIpStatistics' ) ;
    GetIpNet_IIM_Table := GetProcAddress (IpHlpModule, '
GetIpNet_IIM_Table' ) ;
    GetIpAddrTable := GetProcAddress (IpHlpModule, ' GetIpAddrTable' ) ;
    GetIpForwardTable := GetProcAddress (IpHlpModule, ' GetIpForwardTable'
) ;
    GetIcmpStatistics := GetProcAddress (IpHlpModule, ' GetIcmpStatistics'
) ;
    GetRTTAndHopCount := GetProcAddress (IpHlpModule, ' GetRTTAndHopCount'
) ;
    GetIfTable := GetProcAddress (IpHlpModule, ' GetIfTable' ) ;
    GetIfEntry := GetProcAddress (IpHlpModule, ' GetIfEntry' ) ;
    GetFriendlyIfIndex := GetProcAddress (IpHlpModule, '
GetFriendlyIfIndex' ) ;
end;

initialization
    IpHlpModule := 0 ;
finalization
    if IpHlpModule <> 0 then
    begin
        FreeLibrary (IpHlpModule) ;
        IpHlpModule := 0 ;
    end ;
end.

```

Файл IPHelper.pas- функції роботи з IP-протоколом

```

unit IPHelper;

interface

uses
  Windows, Messages, SysUtils, Classes, Dialogs, IpHlpApi;

const
  NULL_IP      = ' 0.0. 0.0' ;

//---перетворення добре відомих номерів портів до імен сервісів-----
type
  TWellKnownPort = record
    Prt: DWORD;
    Srv: string[20];
  end;

const
  // тільки найбільш популярні сервіси...
  WellKnownPorts: array[1..32] of TWellKnownPort
  = (
  //   ( Prt: 0; Srv:  ' RESRVED' ),      {Зарезервовано}
    ( Prt: 7; Srv:  ' ECHO  ' ),      {Ping      }
    ( Prt: 9; Srv:  ' DISCARD' ),
    ( Prt: 13; Srv: ' DAYTIME' ),
    ( Prt: 17; Srv: ' QOTD  ' ),      {Показчик на день}
    ( Prt: 19; Srv: ' CHARGEN' ),     {Генератор символів}
    ( Prt: 20; Srv: ' FTPDATA' ),     { File Transfer Protocol- дані}
    ( Prt: 21; Srv: ' FTPCTRL' ),     { File Transfer Protocol- управління}
    ( Prt: 22; Srv: ' SSH    ' ),
    ( Prt: 23; Srv: ' TELNET_IIM_ ' ),
    ( Prt: 25; Srv: ' SMTP   ' ),      { Simple Mail Transfer Protocol}
    ( Prt: 37; Srv: ' TIME   ' ),      { Часовий протокол }
    ( Prt: 43; Srv: ' WHOIS  ' ),      { Сервіс - Кто це }
    ( Prt: 53; Srv: ' DNS    ' ),      { Domain Name Service }
    ( Prt: 67; Srv: ' BOOTPS ' ),      { BOOTP Сервер }
    ( Prt: 68; Srv: ' BOOTPC ' ),      { BOOTP Клієнт }
    ( Prt: 69; Srv: ' TFTP   ' ),      { стандартний FTP }
    ( Prt: 70; Srv: ' GOPHER ' ),      { Протокол Gopher }
    ( Prt: 79; Srv: ' FINGER ' ),      { Протокол Finger }
    ( Prt: 80; Srv: ' HTTP   ' ),      { Протокол HTTP }
    ( Prt: 88; Srv: ' KERBROS' ),      { Протокол Kerberos }
    ( Prt: 109; Srv: ' POP2   ' ),      { Протокол Post Office Protocol Version
2 }
    ( Prt: 110; Srv: ' POP3   ' ),      { Протокол Post Office Protocol Version
3 }
    ( Prt: 111; Srv: ' SUN_RPC' ),      { Протокол SUN Remote Procedure Call }
    ( Prt: 119; Srv: ' NNTP   ' ),      { Протокол Net_IIM_work News Transfer
Protocol }
    ( Prt: 123; Srv: ' NTP    ' ),      { Протокол Net_IIM_work Time protocol
}
    ( Prt: 135; Srv: ' DCOMRPC' ),      { Протокол Location Service
}
    ( Prt: 137; Srv: ' NBNAME ' ),      { NET_IIM_BIOS сервіс імен }
    ( Prt: 138; Srv: ' NBDGRAM' ),      { NET_IIM_BIOS сервіс датаграм }
    ( Prt: 139; Srv: ' NBSESS ' ),      { NET_IIM_BIOS сервіс сесій }
    ( Prt: 143; Srv: ' IMAP   ' ),      { Протокол Internet_IIM_ Message Access
Protocol }
    ( Prt: 161; Srv: ' SNMP   ' ),      { Протокол Simple Net_IIM_w. Management
Protocol }
    ( Prt: 169; Srv: ' SEND   ' )
  )

```

```

);

//-----перетворення ICMP кодів помилок до рядків-----

const
  ICMP_ERROR_BASE = 11000;
  IcmpErr : array[1..22] of string =
  (
    ' IP_BUFFER_TOO_SMALL' , ' IP_DEST_NET_IIM_UNREACHABLE' , '
IP_DEST_HOST_UNREACHABLE' ,
    ' IP_PROTOCOL_UNREACHABLE' , ' IP_DEST_PORT_UNREACHABLE' , ' IP_NO_RESOURCES'
  ,
    ' IP_BAD_OPTION' , ' IP_HARDWARE_ПОМИЛКА' , ' IP_PACKET_TOO_BIG' , '
IP_REQUEST_TIMED_OUT' ,
    ' IP_BAD_REQUEST' , ' IP_BAD_ROUTE' , ' IP_TTL_EXPIRED_TRANSIT' ,
    ' IP_TTL_EXPIRED_REASSEM' , ' IP_PARAMETER_PROBLEM' , ' IP_SOURCE_QUENCH' ,
    ' IP_OPTION_TOO_BIG' , ' IP_BAD_DESTINATION' , ' IP_ADDRESS_DELETED' ,
    ' IP_SPEC_MTU_CHANGE' , ' IP_MTU_CHANGE' , ' IP_UNLOAD'
  );

//-----Перетворення різних перерахованих величин у рядки-----

  ARPEntryType : array[1..4] of string = ( ' інший' , ' неправильний' ,
    ' динамічний' , ' статичний'
  );
  TCPConnState :
    array[1..12] of string =
    ( ' closed' , ' listening' , ' syn_sent' ,
      ' syn_rcvd' , ' established' , ' fin_wait1' ,
      ' fin_wait2' , ' close_wait' , ' closing' ,
      ' last_ack' , ' time_wait' , ' delete_tcb'
    );
  TCPToAlgo : array[1..4] of string =
    ( ' Const.Timeout' , ' MIL-STD-1778' ,
      ' Van Jacobson' , ' інший' );
  IPForwTypes : array[1..4] of string =
    ( ' інший' , ' invalid' , ' local' , ' remote' );
  IPForwProtos : array[1..18] of string =
    ( ' інший' , ' LOCAL' , ' NET_IIM_MGMT' , ' ICMP' , ' EGP' ,
      ' GGP' , ' HELLO' , ' RIP' , ' IS_IS' , ' ES_IS' ,
      ' CISCO' , ' BBN' , ' OSPF' , ' BGP' , ' BOOTP' ,
      ' AUTO_STAT' , ' STATIC' , ' NOT_DOD' );

type
  // для IpHlpNet_IIM_workParams
  TNet_IIM_workParams = record
    HostName: string ;
    DomainName: string ;
    CurrentDnsServer: string ;
    DnsServerTot: integer ;
    DnsServerNames: array [0..9] of string ;
    NodeType: UINT;
    ScopeID: string ;
    EnableRouting: UINT;
    EnableProxy: UINT;
    EnableDNS: UINT;
  end;

  TIfRows = array of TMibIfRow ; // динамічний масив колонок

  // для IpHlpAdaptersInfo
  TAdaptorInfo = record
    AdapterName: string ;

```



```

//-----Основні функції-----

{ отримання наступного "токена" з рядка }
function NextToken( var s: string; Separator: char ): string;
var
  Sep_Pos      : byte;
begin
  Result := ' ';
  if length( s ) > 0 then begin
    Sep_Pos := pos( Separator, s );
    if Sep_Pos > 0 then begin
      Result := copy( s, 1, Pred( Sep_Pos ) );
      Delete( s, 1, Sep_Pos );
    end
    else begin
      Result := s;
      s := ' ';
    end;
  end;
end;

//-----
{ перетворення числового MAC-адреса до ww-xx-yy-zz рядка }
function MacAddr2Str( MacAddr: TMacAddress; size: integer ): string;
var
  i      : integer;
begin
  if Size = 0 then
    begin
      Result := ' 00-00-00' ;
      EXIT;
    end
  else Result := ' ';
  //
  for i := 1 to Size do
    Result := Result + IntToHex( MacAddr[i], 2) + ' -' ;
    Delete( Result, Length( Result ), 1 );
  end;
end;

//-----
{ перетворення IP-адреси в мережний байт типу DWORD }
function IpAddr2Str( IPAddr: DWORD ): string;
var
  i      : integer;
begin
  Result := ' ';
  for i := 1 to 4 do
    begin
      Result := Result + Format( ' %3d.' , [IPAddr and $FF] );
      IPAddr := IPAddr shr 8;
    end;
    Delete( Result, Length( Result ), 1 );
  end;
end;

//-----
{ перетворення крапкової десяткової IP-адреси в мережний байт типу DWORD}
function Str2IpAddr( IPStr: string ): DWORD;
var
  i      : integer;
  Num    : DWORD;
begin
  Result := 0;
  for i := 1 to 4 do
    try
      Num := ( StrToInt( NextToken( IPStr, ' .' ) ) ) shl 24;
      Result := ( Result shr 8 ) or Num;
    except
      Result := 0;
    end;
  end;
end;

```

```

end;

//-----
{ перетворення номеру порту в мережний байт типу DWORD }
function Port2Wrd( nwoPort: DWORD ): DWORD;
begin
  Result := Swap( WORD( nwoPort ) );
end;

//-----
{ перетворення номеру порту в мережний байт типу string }
function Port2Str( nwoPort: DWORD ): string;
begin
  Result := IntToStr( Port2Wrd( nwoPort ) );
end;

//-----
{ перетворення номеру порту в сервіс ID }
function Port2Svc( Port: DWORD ): string;
var
  i          : integer;
begin
  Result := Format( '%4d' , [Port] ); // у випадку, якщо порт не знайдено
  for i := Low( WellKnownPorts ) to High( WellKnownPorts ) do
    if Port = WellKnownPorts[i].Prt then
      begin
        Result := WellKnownPorts[i].Srv;
        BREAK;
      end;
  end;
end;

//-----
{ голова частина, фіксація мережних параметрів досліджуємої мережі для оцінки
якості обслуговування (IT_infrastructure_management) }

procedure Get_Net_IIM_workParams( List: TStrings );
var
  Net_IIM_workParams: TNet_IIM_workParams ;
  I, ErrorCode: integer ;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  ErrorCode := IpHlpNet_IIM_workParams (Net_IIM_workParams) ;
  if ErrorCode <> 0 then
    begin
      List.Add (SysErrorMessage (ErrorCode));
      exit;
    end ;
  with Net_IIM_workParams do
    begin
      List.Add( ' Ім'я хосту           : ' + HostName );
      List.Add( ' Домен              : ' + DomainName );
      List.Add( ' NET_IIM_BIOS тип   : ' + NET_IIM_BIOSTypes[NodeType] );
      List.Add( ' DHCP область       : ' + ScopeID );
      List.Add( ' ROUTING визначено  : ' + IntToStr( EnableRouting ) );
      List.Add( ' PROXY визначено    : ' + IntToStr( EnableProxy ) );
      List.Add( ' DNS визначено      : ' + IntToStr( EnabledDNS ) );
      if DnsServerTot <> 0 then
        begin
          for I := 0 to Pred (DnsServerTot) do
            List.Add( ' DNS адреса сервєру : ' + DnsServerNames [I] );
          end ;
        end ;
    end ;
end ;

//-----//
function IpHlpNet_IIM_workParams (var Net_IIM_workParams: TNet_IIM_workParams):
integer ;

```

```

var
  FixedInfo      : PTFixedInfo;          // дані
  InfoSize      : Longint;
  PDnsServer    : PTIP_ADDR_STRING ;    // дані
begin
  InfoSize := 0 ; // дані
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  result := GetNet_IIM_workParams( Nil, @InfoSize ); // дані
  if result <> ERROR_BUFFER_OVERFLOW then exit ; // дані
  GetMem (FixedInfo, InfoSize) ; // дані
  try
    result := GetNet_IIM_workParams( FixedInfo, @InfoSize ); // дані
    if result <> ERROR_SUCCESS then exit ;
    Net_IIM_workParams.DnsServerTot := 0 ;
    with FixedInfo^ do
      begin
        Net_IIM_workParams.HostName := trim (HostName) ;
        Net_IIM_workParams.DomainName := trim (DomainName) ;
        Net_IIM_workParams.ScopeId := trim (ScopeID) ;
        Net_IIM_workParams.NodeType := NodeType ;
        Net_IIM_workParams.EnableRouting := EnableRouting ;
        Net_IIM_workParams.EnableProxy := EnableProxy ;
        Net_IIM_workParams.EnableDNS := EnabledDNS ;
        Net_IIM_workParams.DnsServerNames [0] := DNSServerList.IPAddress ; //
дані
        if Net_IIM_workParams.DnsServerNames [0] <> '\ ' then
          Net_IIM_workParams.DnsServerTot := 1 ;
        PDnsServer := DnsServerList.Next;
        while PDnsServer <> Nil do
          begin
            Net_IIM_workParams.DnsServerNames [Net_IIM_workParams.DnsServerTot]
:=
            PDnsServer^.IPAddress ; // дані
            inc (Net_IIM_workParams.DnsServerTot) ;
            if Net_IIM_workParams.DnsServerTot >=
              Length (Net_IIM_workParams.DnsServerNames) then exit
;
            PDnsServer := PDnsServer.Next ;
          end;
        end ;
      finally
        FreeMem (FixedInfo) ; // дані
      end ;
    end;
  end;

//-----

function ICMPErr2Str( ICMPErrCode: DWORD) : string;
begin
  Result := ' UnknownError : ' + IntToStr( ICMPErrCode );
  dec( ICMPErrCode, ICMP_ERROR_BASE );
  if ICMPErrCode in [Low(ICMPerr)..High(ICMPerr)] then
    Result := ICMPerr[ ICMPErrCode];
end;

//-----

// включення байтів у/з для кожного адаптера

function IpHlpIfTable(var IfTot: integer; var IfRows: TIfRows): integer ;
var
  I,
  TableSize : integer;
  pBuf, pNext : PChar;
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;

```

```

SetLength (IfRows, 0) ;
IfTot := 0 ; // дані
TableSize := 0;
// перший виклик: необхідно отримати розмір пам' яті
result := GetIfTable (Nil, @TableSize, false) ; // дані
if result <> ERROR_INSUFFICIENT_BUFFER then exit ;
GetMem( pBuf, TableSize );
try
    FillChar (pBuf^, TableSize, #0); // очищаємо буфер з W98 не беремо
крапку таблиці
    result := GetIfTable (PTMibIfTable (pBuf), @TableSize, false) ;
    if result <> NO_ERROR then exit ;
    IfTot := PTMibIfTable (pBuf)^.dwNumEntries ;
    if IfTot = 0 then exit ;
    SetLength (IfRows, IfTot) ;
    pNext := pBuf + SizeOf(IfTot) ;
    for i := 0 to Pred (IfTot) do
    begin
        IfRows [i] := PTMibIfRow (pNext )^ ;
        inc (pNext, SizeOf (TMibIfRow)) ;
    end;
finally
    FreeMem (pBuf) ;
end ;
end;

procedure Get_IfTable( List: TStrings );
var
    IfRows      : TIfRows ;
    Error, I     : integer;
    NumEntries  : integer;
    sDescr, sIfName: string ;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    SetLength (IfRows, 0) ;
    Error := IpHlpIfTable (NumEntries, IfRows) ;
    if (Error <> 0) then
        List.Add( SysErrorMessage( GetLastError ) )
    else if NumEntries = 0 then
        List.Add( ' даних немає ' )
    else
    begin
        for I := 0 to Pred (NumEntries) do
        begin
            with IfRows [I] do
            begin
                if wszName [1] = #0 then
                    sIfName := ' '
                else
                    sIfName := WideCharToString (@wszName) ; // конвертуємо Юнікод
                sIfName := trim (sIfName) ;
                sDescr := bDescr ;
                sDescr := trim (sDescr);
                List.Add (Format (
                    ' %0.8x |%3d | %16s |%8d |%12d |%2d |%2d |%10d |%10d | %-s| %-s'
                    ,
                    [dwIndex, dwType, MacAddr2Str( TMacAddress( bPhysAddr ) ,
                    dwPhysAddrLen ), dwMTU, dwSpeed, dwAdminStatus,
                    dwOperStatus, Int64 (dwInOctets), Int64 (dwOutOctets), //
                    конвертуємо до 32-біт
                    sIfName, sDescr] ) // дані, додані в/з
                );
            end;
        end ;
    end ;
    SetLength (IfRows, 0) ; // вільна пам' ять
end ;

```

```

function IpHlpIfEntry(Index: integer; var IfRow: TMibIfRow): integer ;
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  FillChar (IfRow, SizeOf (TMibIfRow), #0); // очищаємо буфер з W98 не беремо
  IfRow.dwIndex := Index ;
  result := GetIfEntry (@IfRow) ;
end ;

//-----
{ інформація про інсталювані адаптери }

function IpHlpAdaptersInfo(var AdpTot: integer; var AdpRows: TAdaptorRows):
integer ;
var
  BufLen      : DWORD;
  AdapterInfo : PTIP_ADAPTER_INFO;
  PIPAddr     : PTIP_ADDR_STRING;
  PBuf        : PCHAR ;
  I           : integer ;
begin
  SetLength (AdpRows, 4) ;
  AdpTot := 0 ;
  BufLen := 0 ;
  result := GetAdaptersInfo( Nil, @BufLen ) ;
  if (result <> ERROR_INSUFFICIENT_BUFFER) and (result = NO_ERROR) then exit ;
  GetMem( pBuf, BufLen ) ;
  try
    FillChar (pBuf^, BufLen, #0); // очищуємо буфер
    result := GetAdaptersInfo( PTIP_ADAPTER_INFO (PBuf), @BufLen ) ;
    if result = NO_ERROR then
      begin
        AdapterInfo := PTIP_ADAPTER_INFO (PBuf) ;
        while ( AdapterInfo <> nil ) do
          begin
            AdpRows [AdpTot].IPAddressTot := 0 ;
            SetLength (AdpRows [AdpTot].IPAddressList, 2) ;
            SetLength (AdpRows [AdpTot].IPMaskList, 2) ;
            AdpRows [AdpTot].GatewayTot := 0 ;
            SetLength (AdpRows [AdpTot].GatewayList, 2) ;
            AdpRows [AdpTot].DHCPTot := 0 ;
            SetLength (AdpRows [AdpTot].DHCPSTotal, 2) ;
            AdpRows [AdpTot].PrimWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].PrimWINSServer, 2) ;
            AdpRows [AdpTot].SecWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].SecWINSServer, 2) ;
            AdpRows [AdpTot].CurrIPAddress := NULL_IP;
            AdpRows [AdpTot].CurrIPMask := NULL_IP;
            AdpRows [AdpTot].AdapterName := Trim( string(
AdapterInfo^.AdapterName ) ) ;
            AdpRows [AdpTot].Description := Trim( string(
AdapterInfo^.Description ) ) ;
            AdpRows [AdpTot].MacAddress := MacAddr2Str( TMacAddress(
AdapterInfo^.AddressLength ) ;
            AdpRows [AdpTot].Index := AdapterInfo^.Index ;
            AdpRows [AdpTot].aType := AdapterInfo^.aType ;
            AdpRows [AdpTot].DHCPEnabled := AdapterInfo^.DHCPEnabled ;
            if AdapterInfo^.CurrentIPAddress <> Nil then
              begin
                AdpRows [AdpTot].CurrIPAddress :=
AdapterInfo^.CurrentIPAddress.IpAddress ;
                AdpRows [AdpTot].CurrIPMask :=
AdapterInfo^.CurrentIPAddress.IpMask ;
              end ;

            // беремо список IP адрес та конвертуємо в IPAddressList
            I := 0 ;

```

```

PipAddr := @AdapterInfo^.IPAddressList ;
while (PipAddr <> Nil) do
begin
  AdpRows [AdpTot].IPAddressList [I] := PipAddr.IpAddress ;
  AdpRows [AdpTot].IPMaskList [I] := PipAddr.IpMask ;
  PipAddr := PipAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].IPAddressList) <= I then
  begin
    SetLength (AdpRows [AdpTot].IPAddressList, I -2) ;
    SetLength (AdpRows [AdpTot].IPMaskList, I -2) ;
  end ;
end ;
AdpRows [AdpTot].IPAddressTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PipAddr := @AdapterInfo^.GatewayList ;
while (PipAddr <> Nil) do
begin
  AdpRows [AdpTot].GatewayList [I] := PipAddr.IpAddress ;
  PipAddr := PipAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].GatewayList) <= I then
    SetLength (AdpRows [AdpTot].GatewayList, I -2) ;
end ;
AdpRows [AdpTot].GatewayTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PipAddr := @AdapterInfo^.DHCPServer ;
while (PipAddr <> Nil) do
begin
  AdpRows [AdpTot].DHCPServer [I] := PipAddr.IpAddress ;
  PipAddr := PipAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].DHCPServer) <= I then
    SetLength (AdpRows [AdpTot].DHCPServer, I -2) ;
end ;
AdpRows [AdpTot].DHCPSTot := I ;

// беремо список IP адрес для PrimaryWINSServer
I := 0 ;
PipAddr := @AdapterInfo^.PrimaryWINSServer ;
while (PipAddr <> Nil) do
begin
  AdpRows [AdpTot].PrimWINSServer [I] := PipAddr.IpAddress ;
  PipAddr := PipAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].PrimWINSServer) <= I then
    SetLength (AdpRows [AdpTot].PrimWINSServer, I -2) ;
end ;
AdpRows [AdpTot].PrimWINSTot := I ;

// беремо список IP адрес для SecondaryWINSServer
I := 0 ;
PipAddr := @AdapterInfo^.SecondaryWINSServer ;
while (PipAddr <> Nil) do
begin
  AdpRows [AdpTot].SecWINSServer [I] := PipAddr.IpAddress ;
  PipAddr := PipAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].SecWINSServer) <= I then
    SetLength (AdpRows [AdpTot].SecWINSServer, I -2) ;
end ;
AdpRows [AdpTot].SecWINSTot := I ;

AdpRows [AdpTot].LeaseObtained := AdapterInfo^.LeaseObtained ;
AdpRows [AdpTot].LeaseExpires := AdapterInfo^.LeaseExpires ;

```

```

        inc (AdpTot) ;
        if Length (AdpRows) <= AdpTot then
            SetLength (AdpRows, AdpTot -2) ; // більше пам' яті
        AdapterInfo := AdapterInfo^.Next;
    end ;
    SetLength (AdpRows, AdpTot) ;
end ;
finally
    FreeMem( pBuf );
end ;
end ;

procedure Get_AdaptersInfo( List: TStrings );
var
    AdpTot: integer;
    AdpRows: TAdaptorRows ;
    Error: DWORD ;
    I: integer ;
    //J: integer ;
    //S: string ;      id.
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    SetLength (AdpRows, 0) ;
    AdpTot := 0 ;
    Error := IpHlpAdaptersInfo(AdpTot, AdpRows) ;
    if (Error <> 0) then
        List.Add( SysErrorMessage( GetLastError ) )
    else if AdpTot = 0 then
        List.Add( ' даних немає ' )
    else
        begin
            for I := 0 to Pred (AdpTot) do
                begin
                    with AdpRows [I] do
                        begin
                            //List.Add(AdapterName + ' |' + Description ); // jpt : не
                            використовується
                            List.Add( Format( ' %8.8x | %6s | %16s | %2d | %16s | %16s | %16s' ,
                                [Index, AdaptTypes[aType], MacAddress, DHCPEnabled,
                                    GatewayList [0], DHCPSTServer [0], PrimWINSSServer [0]] ) );
                            {if IPAdressTot <> 0 then // jpt : не використовується
                                begin
                                    S := ' ' ;
                                    for J := 0 to Pred (IPAdressTot) do
                                        S := S + IPAdressList [J] + ' /' + IPMaskList [J] + '
| ' ;
                                    List.Add(IntToStr (IPAdressTot) + ' IP Adresse(s): ' + S);
                                end ;
                            List.Add( ' ' ); }
                        end ;
                    end ;
                end ;
            SetLength (AdpRows, 0) ;
        end ;

//-----
{ моніторимо час доступу до IP досліджуємої мережі для оцінки якості
обслуговування (IT_infrastructure_management)}
function Get_RTTAndHopCount( IPAddr: DWORD; MaxHops: Longint; var RTT: Longint;
    var HopCount: Longint ): integer;
begin
    if not GetRTTAndHopCount( IPAddr, @HopCount, MaxHops, @RTT ) then
        begin
            Result := GetLastError;
            RTT :=-1; // Расположення BAD_HOST_NAME,etc...
            HopCount :=-1;
        end
end

```

```

else
    Result := NO_ERROR;
end;

//-----
{ ARP-таблиця включає відношення між віддаленим IP та віддаленим MAC-адресом.
}
procedure Get_ARPTable( List: TStrings );
var
    IPNet_IIM_Row      : TMibIPNet_IIM_Row;
    TableSize         : DWORD;
    NumEntries        : DWORD;
    ErrorCode         : DWORD;
    i                 : integer;
    pBuf              : PChar;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    // перший виклик: беремо довжину таблиці
    TableSize := 0;
    ErrorCode := GetIPNet_IIM_Table( Nil, @TableSize, false ); // дані
    //
    if ErrorCode = ERROR_NO_DATA then
    begin
        List.Add( ' ARP-кеш пустий.' );
        EXIT;
    end;
    // беремо таблицю
    GetMem( pBuf, TableSize );
    NumEntries := 0 ;
    try
        ErrorCode := GetIpNet_IIM_Table( PTMIBIPNet_IIM_Table( pBuf ), @TableSize,
        false );
        if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMIBIPNet_IIM_Table( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
            begin
                inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
                for i := 1 to NumEntries do
                begin
                    IPNet_IIM_Row := PTMIBIPNet_IIM_Row( PBuf )^;
                    with IPNet_IIM_Row do
                        List.Add( Format( ' %8x | %12s | %16s | %10s' ,
                        [dwIndex, MacAddr2Str( bPhysAddr, dwPhysAddrLen ),
                        IPAddr2Str( dwAddr ), ARPEntryType[dwType]
                        ]));
                    inc( pBuf, SizeOf( IPNet_IIM_Row ) );
                end;
            end;
        end;
    else
        List.Add( ' ARP-кеш пустий.' );
    end;
    else
        List.Add( SysErrorMessage( ErrorCode ) );
    // необхідно відновити показник!
    finally
        dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( IPNet_IIM_Row ) );
        FreeMem( pBuf );
    end ;
end;

//-----
procedure Get_TCPTable( List: TStrings );
var
    TCPRow           : TMIBTCPRow;
    i,

```

```

    NumEntries : integer;
    TableSize  : DWORD;
    ErrorCode   : DWORD;
    DestIP     : string;
    pBuf       : PChar;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    RecentIPs.Clear;
    // перший виклик: беремо довжину таблиці
    TableSize := 0;
    NumEntries := 0 ;
    ErrorCode := GetTCPTable( Nil, @TableSize, false ); // дані
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    // беремо розмір пам'яті, викликаємо знову
    GetMem( pBuf, TableSize );
    // беремо таблицю
    ErrorCode := GetTCPTable( PTMIBTCPTable( pBuf ), @TableSize, false );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMIBTCPTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
                    for i := 1 to NumEntries do
                        begin
                            TCPRow := PTMIBTCPRow( pBuf )^; // беремо останній запис
                            with TCPRow do
                                begin
                                    if dwRemoteAddr = 0 then
                                        dwRemotePort := 0;
                                    DestIP := IPAddr2Str( dwRemoteAddr );
                                    List.Add(
                                        Format( ' %15s : %-7s | %15s : %-7s | %-16s' ,
                                            [IpAddr2Str( dwLocalAddr ),
                                              Port2Svc( Port2Wrd( dwLocalPort ) ),
                                              DestIP,
                                              Port2Svc( Port2Wrd( dwRemotePort ) ),
                                              TCPConnState[dwState]
                                            ] ) );
                                    //
                                    if (not ( dwRemoteAddr = 0 ))
                                        and ( RecentIPs.IndexOf( DestIP ) = -1 ) then
                                        RecentIPs.Add( DestIP );
                                end;
                            inc( pBuf, SizeOf( TMIBTCPRow ) );
                        end;
                    end;
                end
            else
                List.Add( SysErrorMessage( ErrorCode ) );
            dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibTCPRow ) );
            FreeMem( pBuf );
        end;

    //-----
    procedure Get_TCPStatistics( List: TStrings );
    var
        TCPStats      : TMibTCPStats;
        ErrorCode     : DWORD;
    begin
        if not Assigned( List ) then EXIT;
        List.Clear;
        if NOT LoadIpHlp then exit ;
        ErrorCode := GetTCPStatistics( @TCPStats );
        if ErrorCode = NO_ERROR then

```

```

with TCPStats do
begin
    List.Add( ' Алгоритм повторної передачі : ' + TCPToAlgo[dwRTOAlgorithm]
);
    List.Add( ' Мінімальний час виходу          : ' + IntToStr( dwRTOMin ) + '
ms' );
    List.Add( ' Максимальний час виходу          : ' + IntToStr( dwRTOMax ) +
' ms' );
    List.Add( ' Максимальне число підключень : ' + IntToStr( dwRTOAlgorithm )
);
    List.Add( ' Активні підключення              : ' + IntToStr( dwActiveOpens
) );
    List.Add( ' пасивні підключення              : ' + IntToStr( dwPassiveOpens
) );
    List.Add( ' Невдала спроба відкриття          : ' + IntToStr( dwAttemptFails )
);
    List.Add( ' Скидання встановленого підключення : ' + IntToStr(
dwEstabResets ) );
    List.Add( ' Поточне встановлене підключення.: ' + IntToStr( dwCurrEstab )
);
    List.Add( ' Отримані сегменти                : ' + IntToStr( dwInSegs ) );
    List.Add( ' Передані сегменти                : ' + IntToStr( dwOutSegs ) );
    List.Add( ' Перепідключені сегменти          : ' + IntToStr( dwReTransSegs ) );
    List.Add( ' помилка входу                    : ' + IntToStr( dwInErrs ) );
    List.Add( ' Перезавантаження вихідних        : ' + IntToStr( dwOutRsts
) );
    List.Add( ' Сумарні зв'язки                  : ' + IntToStr( dwNumConns ) );
end
else
    List.Add( SysErrorMessage( ErrorCode ) );
end;

function IpHlpTCPStatistics (var TCPStats: TMibTCPStats): integer ;
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetTCPStatistics( @TCPStats );
end;

//-----
procedure Get_UDPTable( List: TStrings );
var
    UDPRow          : TMIBUDPRow;
    i,
    NumEntries      : integer;
    TableSize       : DWORD;
    ErrorCode        : DWORD;
    pBuf            : PChar;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;

    // перший виклик: беремо довжину таблиці
    TableSize := 0;
    NumEntries := 0 ;
    ErrorCode := GetUDPTable( Nil, @TableSize, false );
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    // виділяємо пам'ять, викликаємо знову
    GetMem( pBuf, TableSize );

    // беремо таблицю
    ErrorCode := GetUDPTable( PTMIBUDPTable( pBuf ), @TableSize, false );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMIBUDPTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin

```

```

inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
for i := 1 to NumEntries do
begin
  UDPRow := PTMIBUDPRow( pBuf )^; // беремо останній запис
  with UDPRow do
    List.Add( Format( ' %15s : %-6s' ,
      [IpAddr2Str( dwLocalAddr ),
      Port2Svc( Port2Wrd( dwLocalPort ) )
      ] ) );
    inc( pBuf, SizeOf( TMIBUDPRow ) );
  end;
end
else
  List.Add( ' немає даних.' );
end
else
  List.Add( SysErrorMessage( ErrorCode ) );
dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibUDPRow ) );
FreeMem( pBuf );
end;

//-----
procedure Get_IPAddrTable( List: TStrings );
var
  IPAddrRow      : TMibIPAddrRow;
  TableSize      : DWORD;
  ErrorCode      : DWORD;
  i              : integer;
  pBuf           : PChar;
  NumEntries     : DWORD;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  TableSize := 0; ;
  NumEntries := 0 ;
  // перший виклик: беремо довжину таблиці
  ErrorCode := GetIpAddrTable( Nil, @TableSize, true ); // дані
  if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
    EXIT;

  GetMem( pBuf, TableSize );
  // беремо таблицю
  ErrorCode := GetIpAddrTable( PTMibIPAddrTable( pBuf ), @TableSize, true );
  if ErrorCode = NO_ERROR then
    begin
      NumEntries := PTMibIPAddrTable( pBuf )^.dwNumEntries;
      if NumEntries > 0 then
        begin
          inc( pBuf, SizeOf( DWORD ) );
          for i := 1 to NumEntries do
            begin
              IPAddrRow := PTMIBIPAddrRow( pBuf )^;
              with IPAddrRow do
                List.Add( Format( ' %8.8x | %15s | %15s | %15s | %8.8d' ,
                  [dwIndex,
                  IPAddr2Str( dwAddr ),
                  IPAddr2Str( dwMask ),
                  IPAddr2Str( dwBCastAddr ),
                  dwReasmSize
                  ] ) );
                inc( pBuf, SizeOf( TMIBIPAddrRow ) );
              end;
            end
          else
            List.Add( ' немає даних.' );
          end
        else
          List.Add( SysErrorMessage( ErrorCode ) );
        end
      end
    end
  end
end

```

```

// відновлюємо показчик!
dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( IPAddrRow ) );
FreeMem( pBuf );
end;

//-----
{ отримуємо дані з таблиці маршрутизації досліджуємої мережі для оцінки якості
обслуговування (IT_infrastructure_management); }
procedure Get_IPForwardTable( List: TStrings );
var
  IPForwRow      : TMibIPForwardRow;
  TableSize      : DWORD;
  ErrorCode      : DWORD;
  i              : integer;
  pBuf           : PChar;
  NumEntries     : DWORD;
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  TableSize := 0;

  // перший виклик: беремо довжину таблиці
  NumEntries := 0 ;
  ErrorCode := GetIpForwardTable( Nil, @TableSize, true);
  if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
    EXIT;

  // беремо таблицю
  GetMem( pBuf, TableSize );
  ErrorCode := GetIpForwardTable( PTMibIPForwardTable( pBuf ), @TableSize,
true);
  if ErrorCode = NO_ERROR then
    begin
      NumEntries := PTMibIPForwardTable( pBuf )^.dwNumEntries;
      if NumEntries > 0 then
        begin
          inc( pBuf, SizeOf( DWORD ) );
          for i := 1 to NumEntries do
            begin
              IPForwRow := PTMibIPForwardRow( pBuf )^;
              with IPForwRow do
                begin
                  if (dwForwardType < 1)
                    or (dwForwardType > 4) then
                    dwForwardType := 1 ; // дані
                  List.Add( Format(
                    '%15s | %15s | %15s | %8.8x | %7s | %5.5d | %7s | %2.2d' ,
                    [IPAddr2Str( dwForwardDest ),
                    IPAddr2Str( dwForwardMask ),
                    IPAddr2Str( dwForwardNextHop ),
                    dwForwardIFIndex,
                    IPForwTypes[dwForwardType],
                    dwForwardNextHopAS,
                    IPForwProtos[dwForwardProto],
                    dwForwardMetric1
                    ] ) );
                end ;
              inc( pBuf, SizeOf( TMibIPForwardRow ) );
            end;
          end
        else
          List.Add( ' немає даних.' );
        end
      else
        List.Add( SysErrorMessage( ErrorCode ) );
      dec( pBuf, SizeOf( DWORD ) + NumEntries -SizeOf( TMibIPForwardRow ) );
      FreeMem( pBuf );
    end;
end;

```

```

//-----
procedure Get_IPStatistics( List: TStrings );
var
  IPStats      : TMibIPStats;
  ErrorCode    : integer;
begin
  if not Assigned( List ) then EXIT;
  if NOT LoadIpHlp then exit ;
  ErrorCode := GetIPStatistics( @IPStats );
  if ErrorCode = NO_ERROR then
  begin
    List.Clear;
    with IPStats do
    begin
      if dwForwarding = 1 then
        List.add( ' Розблокована пересилка      : ' + ' так' )
      else
        List.add( ' Розблокована пересилка      : ' + ' ні' );
      List.add( ' Любий TTL                      : ' + inttostr( dwDefaultTTL ) );
      List.add( ' Датаграма прийнята           : ' + inttostr( dwInReceives ) );
      List.add( ' Помилка заголовку (In)       : ' + inttostr( dwInHdrErrors )
    );
      List.add( ' Помилка адреси (In)          : ' + inttostr( dwInAddrErrors ) );
      List.add( ' Датаграма переслана         : ' + inttostr( dwForwDatagrams ) );
    // дані
      List.add( ' Невизначений протокол (In)   : ' + inttostr( dwInUnknownProtos
    ) );
      List.add( ' Датаграма відмовлена         : ' + inttostr( dwInDiscards ) );
      List.add( ' Датаграма встановлена        : ' + inttostr( dwInDelivers ) );
      List.add( ' Зовнішній запит              : ' + inttostr( dwOutRequests )
    );
      List.add( ' Маршрутизація не виконана       : ' + inttostr(
dwRoutingDiscards ) );
      List.add( ' Немає маршрутів (Out)        : ' + inttostr( dwOutNoRoutes )
    );
      List.add( ' Перебраний час                : ' + inttostr( dwReasmTimeOut ) );
      List.add( ' Запит перебору                : ' + inttostr( dwReasmReqds ) );
      List.add( ' Повний перебор                : ' + inttostr( dwReasmOKs ) );
      List.add( ' Помилка перебору             : ' + inttostr( dwReasmFails ) );
      List.add( ' Повна фрагментація:          : ' + inttostr( dwFragOKs ) );
      List.add( ' Помилка фрагментації         : ' + inttostr( dwFragFails ) );
      List.add( ' Датаграма фрагментована      : ' + inttostr( dwFRagCreates )
    );
      List.add( ' Кількість інтерфейсів         : ' + inttostr( dwNumIf ) );
      List.add( ' Кількість IP-адрес          : ' + inttostr( dwNumAddr ) );
      List.add( ' Маршрут в таблиці маршрутизатора : ' + inttostr( dwNumRoutes
    ) );
    end;
  end
  else
    List.Add( SysErrorMessage( ErrorCode ) );
  end;
function IpHlpIPStatistics (var IPStats: TMibIPStats): integer ;      // дані
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  result := GetIPStatistics( @IPStats );
end ;

//-----
procedure Get_UdpStatistics( List: TStrings );
var
  UdpStats     : TMibUDPStats;
  ErrorCode    : integer;
begin
  if not Assigned( List ) then EXIT;
  ErrorCode := GetUDPStatistics( @UdpStats );

```

```

if ErrorCode = NO_ERROR then
begin
  List.Clear;
  with UDPStats do
  begin
    List.add( \ Датаграми (In)      : \ + inttostr( dwInDatagrams ) );
    List.add( \ Датаграми (Out)    : \ + inttostr( dwOutDatagrams ) );
    List.add( \ Немає портів      : \ + inttostr( dwNoPorts ) );
    List.add( \ Помилка (In)      : \ + inttostr( dwInErrors ) );
    List.add( \ UDP список портів : \ + inttostr( dwNumAddrs ) );
  end;
end
else
  List.Add( SysErrorMessage( ErrorCode ) );
end;

//-----*//
function IpHlpUdpStatistics (UdpStats: TMibUDPStats): integer ; // дані
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  result := GetUDPStatistics (@UdpStats) ;
end ;

//-----
procedure Get_ICMPStats( ICMPIn, ICMPOut: TStrings );
var
  ErrorCode      : DWORD;
  ICMPStats      : PTMibICMPInfo;
begin
  if ( ICMPIn = nil ) or ( ICMPOut = nil ) then EXIT;
  ICMPIn.Clear;
  ICMPOut.Clear;
  New( ICMPStats );
  ErrorCode := GetICMPStatistics( ICMPStats );
  if ErrorCode = NO_ERROR then
  begin
    with ICMPStats.InStats do
    begin
      ICMPIn.Add( \ Прийнято повідомлень      : \ + IntToStr( dwMsgs ) );
      ICMPIn.Add( \ Помилка                  : \ + IntToStr( dwErrors ) );
      ICMPIn.Add( \ Розташування недосягнено  : \ + IntToStr( dwDestUnreachs
    ) );
      ICMPIn.Add( \ Час перевищений          : \ + IntToStr( dwTimeExcds ) );
      ICMPIn.Add( \ Проблеми з параметрами    : \ + IntToStr( dwParmProbs
    );
      ICMPIn.Add( \ Джерело відключено       : \ + IntToStr( dwSrcQuenchs ) );
      ICMPIn.Add( \ Переназначено           : \ + IntToStr( dwRedirects ) );
      ICMPIn.Add( \ Ехо запит                : \ + IntToStr( dwEchos ) );
      ICMPIn.Add( \ Ехо відповідь           : \ + IntToStr( dwEchoReps ) );
      ICMPIn.Add( \ Запит мітки часу         : \ + IntToStr( dwTimeStamps ) );
      ICMPIn.Add( \ Відповідь мітки часу     : \ + IntToStr( dwTimeStampReps
    );
      ICMPIn.Add( \ Запит маски адрес        : \ + IntToStr( dwAddrMasks ) );
      ICMPIn.Add( \ Відповідь маски адрес    : \ + IntToStr( dwAddrReps ) );
    end;
    //
    with ICMPStats.OutStats do
    begin
      ICMPOut.Add( \ Повідомлення вправлено   : \ + IntToStr( dwMsgs ) );
      ICMPOut.Add( \ Помилка                  : \ + IntToStr( dwErrors ) );
      ICMPOut.Add( \ Розташування недосягнено : \ + IntToStr( dwDestUnreachs
    ) );
      ICMPOut.Add( \ Час перевищений          : \ + IntToStr( dwTimeExcds ) );
      ICMPOut.Add( \ Проблеми з параметрами    : \ + IntToStr( dwParmProbs
    );
      ICMPOut.Add( \ Джерело відключено       : \ + IntToStr( dwSrcQuenchs ) );
      ICMPOut.Add( \ Переназначено           : \ + IntToStr( dwRedirects ) );
      ICMPOut.Add( \ Ехо запит                : \ + IntToStr( dwEchos ) );

```

```
        ICMPOut.Add( 'Ехо відповідь          : ' + IntToStr( dwEchoReps ) );
        ICMPOut.Add( 'Запит мітки часу      : ' + IntToStr( dwTimeStamps ) );
        ICMPOut.Add( 'Відповідь мітки часу  : ' + IntToStr( dwTimeStampReps )
);
        ICMPOut.Add( 'Запит маски адрес: ' + IntToStr( dwAddrMasks ) );
        ICMPOut.Add( 'Відповідь маски адрес : ' + IntToStr( dwAddrReps ) );
    end;
end
else
    IcmpIn.Add( SysErrorMessage( ErrorCode ) );
    Dispose( ICMPStats );
end;

//-----
procedure Get_RecentDestIPs( List: TStrings );
begin
    if Assigned( List ) then
        List.Assign( RecentIPs )
    end;

initialization
    RecentIPs := TStringList.Create;

finalization
    RecentIPs.Free;

end.
```

Кафедра_КБПЗ_2021 рік

Файл TCP_IP.pas- монітор TCP/IP з'єднань досліджуємої мережі гібридної IT-інфраструктури

```

unit TCP_IP;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IPHelper, IpHlpApi, Buttons;

type
  TForm2 = class(TForm)
    StaticText2: TStaticText;
    StaticText3: TStaticText;
    TCPMemo: TMemo;
    UDPMemo: TMemo;
    Timer1: TTimer;
    cbTimer: TCheckBox;
    btRTTI: TSpeedButton;
    SpeedButton1: TSpeedButton;
    edtRTTI: TEdit;
    procedure Timer1Timer(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure btRTTIClick(Sender: TObject);
    procedure cbRecentIPsClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    procedure DOIpStuff;
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Timer1Timer(Sender: TObject);
begin
  if cbTimer.State = cbCHECKED then
  begin
    Timer1.Enabled := false;
    DoIPStuff;
    Timer1.Enabled := true;
  end;
end;

procedure TForm2.DOIpStuff;
begin
  Get_TCPTable( TCPMemo.Lines );
  Get_UDPTable( UDPMemo.Lines );

end;

procedure TForm2.SpeedButton1Click(Sender: TObject);
begin
  Speedbutton1.Enabled := false;
  DoIPStuff;
  Speedbutton1.Enabled := true;
end;

procedure TForm2.btRTTIClick(Sender: TObject);

```

```

var
  IPadr      : dword;
  Rtt, HopCount : longint;
  Res       : integer;
begin
  btRTTI.Enabled := false;
  Screen.Cursor := crHOURLASS;
  IPadr := Str2IPAddr( edtRTTI.Text );
  Res := Get_RTTAndHopCount( IPadr, 128, RTT, HopCount );
  if Res = NO_ERROR then
    ShowMessage( ' Час запиту '
      + inttostr( rtt ) + ' ms, '
      + inttostr( HopCount )
      + ' hops to : ' + edtRTTI.Text
    )
  else
    ShowMessage( ' Відбулася помилка:' + #13
      + ICMPErr2Str( Res ) );
  btRTTI.Enabled := true;
  Screen.Cursor := crDEFAULT;

end;

procedure TForm2.cbRecentIPsClick(Sender: TObject);
begin
  //edtRTTI.Text := cbRecentIPs.Items[cbRecentIPs.ItemIndex];
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  if LoadIpHlp then
    begin
      DOIpStuff;
      Timer1.Enabled := true;
    end
  else
    ShowMessage( ' Інтернет помічник DLL не є доступним, або не підтримується'
  ) ;
end;

end.

```

Файл Stat.pas- статистика мережі гібридної IT-інфраструктури

```

unit Stat;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IPHelper, IpHlpApi, Buttons;

type
  TForm3 = class(TForm)
    StaticText7: TStaticText;
    TCPStatMemo: TMemo;
    StaticText5: TStaticText;
    IPStatsMemo: TMemo;
    StaticText12: TStaticText;
    ICMPInMemo: TMemo;
    ICMPOutMemo: TMemo;
    StaticText4: TStaticText;
    UDPStatsMemo: TMemo;
    Timer1: TTimer;
    cbTimer: TCheckBox;
    btRTTI: TSpeedButton;
    edtRTTI: TEdit;
    procedure Timer1Timer(Sender: TObject);
    procedure btRTTIClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    procedure DOIpStuff;
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.DOIpStuff;
begin

  Get_TCPStatistics( TCPStatMemo.Lines );
  Get_IPStatistics( IPStatsMemo.Lines );
  Get_UDPStatistics( UDPStatsMemo.Lines );
  Get_ICMPStats( ICMPInMemo.Lines, ICMPOutMemo.Lines );

end;

procedure TForm3.Timer1Timer(Sender: TObject);
begin
  if cbTimer.State = cbCHECKED then
  begin
    Timer1.Enabled := false;
    DoIPStuff;
    Timer1.Enabled := true;
  end;
end;

procedure TForm3.btRTTIClick(Sender: TObject);
var
  IPadr      : dword;
  Rtt, HopCount : longint;
  Res        : integer;

```

```
begin
  btRTTI.Enabled := false;
  Screen.Cursor := crHOURLASS;
  IPadr := Str2IPAddr( edtRTTI.Text );
  Res := Get_RTTAndHopCount( IPadr, 128, RTT, HopCount );
  if Res = NO_ERROR then
    ShowMessage( ' Час запиту '
      + inttostr( rtt ) + ' ms, '
      + inttostr( HopCount )
      + ' hops to : ' + edtRTTI.Text
    )
  else
    ShowMessage( ' Помилка:' + #13
      + ICMPErr2Str( Res ) ) ;
  btRTTI.Enabled := true;
  Screen.Cursor := crDEFAULT;

end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  if LoadIpHlp then
  begin
    DOIpStuff;
    Timer1.Enabled := true;
  end
  else
    ShowMessage( 'Інтернет помічник DLL не є доступним, або не підтримується'
  ) ;
end;

end.
```

Файл About.pas - довідка

```
unit About;

interface

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

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Button1: TButton;
    Image2: TImage;
    Image1: TImage;
    Image3: TImage;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Form1.Close;
end;

end.
```