

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

”Допущено до захисту”

Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор

\_\_\_\_\_ Олексій СМІРНОВ  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
за першим (бакалаврським) рівнем вищої освіти  
на тему

**“Програмне забезпечення системи дотримання відповідності  
SLA для сервісів IaaS”**

КБГЗ-2024

Виконав здобувач вищої освіти  
IV курсу, групи КМ-21-3СК  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Пушкаренко В.О.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
канд. фіз.-мат. наук, доцент  
\_\_\_\_\_ Якименко Н.М.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Освітній ступінь бакалавр  
Галузь знань . 12 “Інформаційні технології”  
Спеціальність 123 “Комп’ютерна інженерія”  
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.  
Олексій СМІРНОВ  
« 17 » січня 2024 року

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

Пушкаренку Віталію Олеговичу

(прізвище, ім'я, по батькові)

- Тема роботи Програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS
- Керівник роботи Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  
затверджені наказом вищого навчального закладу № 134-02 від 01.04.2024 року
- Строк подання студентом роботи до захисту 23.05.2024 р.
- Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи дотримання відповідності SLA для сервісів IaaS
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
  - Призначення та область використання.
  - Перегляд аналогічних існуючих систем.
  - Опис і обґрунтування проектних рішень.
  - Етапи програмування системи.
  - Впровадження системи в промислову експлуатацію.
  - Висновки
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

<u>Структурна схема системи</u>	<u>1 аркуш</u>
<u>Функціональна схема системи</u>	<u>1 аркуш</u>
<u>Діаграма процесів</u>	<u>1 аркуш</u>
<u>Блок-схема алгоритму роботи додатку</u>	<u>2 аркуша</u>

7. Дата видачі завдання « 17 » січня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання  
« 17 » січня 2024 р.

Підпис керівника

Якименко Н.М.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2024 р.

Підпис здобувача

Пушкаренко В.О.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Пушкаренко В.О. Програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи дотримання відповідності SLA для сервісів IaaS.

Метою розробки є програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS.

Результат роботи – програмна реалізація системи дотримання відповідності SLA для сервісів IaaS.

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

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

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

Програму розроблено в середовищі Delphi 10.4.1.

**Ключові слова:** комп'ютерна інженерія, SLA, IaaS

## ABSTRACT

**Pushkarenko V.O. SLA compliance software for IaaS services. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the first (bachelor) level of higher education, software was developed, which is intended for the SLA compliance system for IaaS services.

The goal of the development is the SLA compliance system software for IaaS services.

The result of the work is the software implementation of the SLA compliance system for IaaS services.

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

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

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

The program was developed in the Delphi 10.4.1 environment.

**Keywords:** computer engineering, SLA, IaaS

## 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.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	43
2.3 Розгорнута постановка завдання .....	49
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	51
3.1 Опис функціонування системи .....	51
3.2 Розробка структурної схеми.....	59
3.3 Розробка функціональної схеми .....	67
3.4 Розробка діаграми процесів.....	69
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	71
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	71
4.2 Захист розробленого програмного забезпечення.....	82
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	85
6 ОСНОВНІ ВИСНОВКИ.....	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	91

					ВКРБ-123.24.0003.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Лушкаренко В.О.				Програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS	Літ.	Аркуш	Аркушів
Перев.	Якименко Н.М.					Б	1	97
Н.контр.	Коваленко А.С.				ЦНТУ КМ-21-ЗСК			
Затв.	Смірнов О.А.							

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БІС	–	банківські інформаційні системи
ПЗ	–	програмне забезпечення
ТВС	–	технології відкритих систем

КБПЗ – 2024

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ВСТУП

**Актуальність теми.** Послуги «ІТ-інфраструктура як сервіс», IaaS, стають усе популярніше в корпоративних клієнтів, причому їх використовують уже й для критично важливих завдань. Прийшов час розібратися, що гарантують постачальники цих послуг і яку відповідальність несуть у тих випадках, коли віртуальна ІТ-інфраструктура гальмує роботу або зовсім стає недоступною.

Опитавши ведучих постачальників інфраструктурних сервісів IaaS корпоративного рівня, ми провели аналіз їхніх пропозицій. При цьому під «корпоративним рівнем» розуміється наступне: хмарна платформа розгорнута в ЦОД, що відповідає вимогам Tier III (наявність сертифіката від Uptime Institute не обов'язково), і забезпечує високий рівень відказостійкості за рахунок механізмів High Availability (HA) і переїзду віртуальних машин у випадку аварії.

Основні параметри сервісу IaaS, які звичайно вказують в угоді SLA, – це рівень його доступності, час реакції на різні інциденти й тривалість їхнього дозволу, а також схема й параметри компенсації у випадку простою.

Вирішивши скористатися віртуальною ІТ-інфраструктурою, можна сміло розраховувати на доступність 99,5% і вище. Принаймні, меншу цифру не назвав жоден з опитаних нами провайдерів. Причому представники багатьох компаній підкреслили, що зазначене в їхніх відповідях значення є типовим і за запитом замовника рівень доступності може бути збільшений за допомогою різних технічних засобів.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS.

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

- Огляд існуючих систем дотримання відповідності SLA для сервісів IaaS.
- Дослідження системи дотримання відповідності SLA для сервісів IaaS.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– Програмна реалізація системи дотримання відповідності SLA для сервісів IaaS.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі дотримання відповідності SLA для сервісів IaaS.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ – 2024

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Звичайно платформи для надання послуг IaaS корпоративного рівня розміщуються в центрах обробки даних (власних або зовнішніх), що відповідають рівню відказостійкості Tier III, що, як відомо, припускає доступність 99,98%. Зазначені провайдерами значення доступності віртуальних інфраструктур IaaS не перевищують відповідну характеристику фізичної площадки, що цілком природно.

У принципі, постачальник може вказати в SLA як завгодно високу доступність, хоч 100%, але тоді ризикує більше втратити, чим заробити, адже будь-який розсудливий покупець зажадає включити в договір тверду схему компенсації за невиконання погоджених умов. Поки якої-небудь типової схеми ще не вироблений – кожний постачальник пропонує щось своє, так що покупець повинен оцінити запропоновану компенсацію з обліком можливих фінансових втрат у випадку простою ІТ-сервісів.

Багато компаній пропонують певне відшкодування щомісячного платежу (у процентному співвідношенні) за кожний додатковий (понад застережений в SLA) година неприступності сервісу. Наприклад, при зазначеному в SLA рівні доступності 99,95% (простий не більше 1 години на місяць) за кожну додаткову годину відключення від сервісу компанія Inoventica готова відшкодувати 2% від щомісячного платежу. Cloud4U у стандартному варіанті компенсує 1% за 1 годину простою (при розрахунках використовується загальна вартість послуги за повний календарний місяць, що передує даному), але не більше 50% вартості послуги. Ряд провайдерів надали докладні розрахунки того, як розмір компенсації міняється залежно від рівня доступності. У випадку значного зниження цього рівня пропонується дуже істотна компенсація. Наприклад, при значенні менш 95% «Онланта» (ГК «Ланіт») допускає зниження рівня оплати послуги до 40%. А

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

компанія «ІТ-Град», якщо рівень доступності опуститься нижче 96,71%, обіцяє компенсацію 50%. Ясно, що подібне погіршення якості послуг провайдери вважають малоімовірним.

Ми ввели два самостійних принципи компенсації: за порушення цільових показників параметрів послуги й цільових показників по обробці звернень. Порушення цільових показників параметрів послуги компенсується по прогресивній шкалі. Залежно від фактичного рівня доступності розраховується показник компенсації, що виражається у відсотках від суми рахунку за користування послугою. Компенсація за порушення цільових показників по обробці звернень вираховується виходячи із тривалості очікування клієнта з точністю до хвилини.

Відповідно до практики види звернень клієнтів, а також загальні цільові показники за максимальним часом реакції на звернення й максимальний час рішення проблеми описані в регламенті сервісної взаємодії. А в самому договорі SLA ці показники уточнюються для конкретної послуги.

Відповідно до договору, замовник може одержувати в нас кілька послуг. Саме тому в регламенті описуються загальні показники з позначкою: «Цільові показники, визначені в SLA на конкретну послугу, перекривають показники, зазначені в регламенті. Це зроблено для того, щоб при необхідності можна було уточнити (розширити або зменшити) час реакції й час рішення. Ми зобов'язані відреагувати на звернення будь-якого виду протягом 15 хв. Максимальний час рішення, залежно від типу й пріоритету обігу, становить від 1 год (для інцидентів із пріоритетом № 1) до 48 год (для звернень, по яких потрібна повне пророблення інформаційного запиту замовника – наприклад, надання інформації з тарифів і інших послуг, різні уточнення й інструктажі).

Час реакції на заявку звичайно залежить від її пріоритетності. От, наприклад, які рівні пріоритету:

– Critical – сервіс недоступний повністю, необхідно вжити термінових заходів по відновленню, час реакції 15 хв, час відновлення не більше 4 год.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

– High – сервіс недоступний частково, час реакції до 1 год, підвищений пріоритет.

– Normal – уточнення по параметрах сервісу, що течуть нетермінові питання, час реакції до 1 год, на підготовку відповіді приділяється 24 год.

## 1.2 Область застосування

Мало вказати в договорі SLA привабливий рівень доступності й тверді схеми компенсацій, треба ще надати клієнтові зручний і ефективний інструмент контролю. І тут підходи постачальників істотно розрізняються.

Клієнти більше зацікавлені в одержанні від оператора прозорості й точної звітності, чим в освоєнні якихось особливих інструментів для самостійного моніторингу. Як правило щомісяця надаються звіти по погодженому набору параметрів, але, за бажанням клієнта, контрактом може передбачатися й більше часта звітність. Багато компаній, за замовчуванням, надають звіти про стан працездатності сервісу раз на місяць, але можуть і частіше – за запитом клієнтів. Більшість сервісів-провайдерів обходяться статистикою за рівнем доступності віртуальних машин. Ряд компаній пропонують клієнтам скористатися консоллю самообслуговування в онлайн-режимі. У кожного споживача послуги IaaS є доступ до такої консолі з убудованою можливістю онлайн-моніторингу функціонування тих або інших складових. Наприклад, у випадку віртуальних машин ІТ-фахівці замовника можуть проконтролювати, наскільки завантажений процесор, як працює уведення-вивід, скільки пам'яті зайняте та ін. Ці дані доступні в режимі реального часу, а також – за запитом – у вигляді статистики за будь-який період.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Гарна платформа для систем керування корпоративними мережами повинна мати наступні якості:

- масштабованістю;
- розподіленістю відповідно до концепції клієнт/сервер;
- відкритістю, що дозволяє впоратися з різномірним – від настільних комп'ютерів до мейнфреймів – устаткуванням.

Перші дві властивості тісно зв'язані. Гарна масштабованість досягається за рахунок розподіленості системи керування. Розподіленість тут означає, що система може включати кілька серверів і клієнтів. Сервери (називані також менеджерами) збирають дані про поточний стан мережі від агентів (SNMP, CMIP або RMON), убудованих в устаткування мережі, і накопичують їх у своїй базі даних. Клієнти являють собою графічні консолі, за яких працюють адміністратори мережі. Програмне забезпечення клієнта системи керування приймає запити на виконання яких-небудь дій від адміністратора (наприклад, побудова докладної карти частини мережі) і звертається за необхідною інформацією до сервера. Якщо сервер має потрібну інформацію, то він відразу ж передає її клієнтові, якщо немає – те намагається зібрати її від агентів.

Ранні версії систем керування сполучали всі функції в одному комп'ютері, за яким працював адміністратор. Для невеликих мереж або мереж з невеликою кількістю керованого встаткування така структура виявляється цілком задовільною, але при великій кількості керованого встаткування єдиний комп'ютер, до якого стікається інформація від всіх пристроїв мережі, стає

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

вузьким місцем. І мережа не справляється з більшим потоком даних, і сам комп'ютер не встигає їх обробляти. Крім того, великою мережею управляє звичайно не один адміністратор, тому, крім декількох серверів у великій мережі повинне бути кілька консолей, за яких працюють адміністратори мережі, причому на кожній консолі повинна бути представлена специфічна інформація, що відповідає поточним потребам конкретного адміністратора.

Підтримка різноманітного встаткування – скоріше бажане, чим реально існуюче властивість сьгоднішніх систем керування. До числа найбільш популярних продуктів мережного керування відносяться чотири системи: Spectrum компанії CabletronSystems, OpenView фірми Hewlett-Packard, NetView корпорації IBM і Solstice виробництва SunSoft – підрозділу SunMicrosystems. Три компанії із чотирьох самі випускають комунікаційне встаткування. Природно, що система Spectrum найкраще управляє встаткуванням компанії Cabletron, OpenView – устаткуванням компанії Hewlett-Packard, а NetView- устаткуванням компанії IBM.

При побудові карти мережі, що складається з устаткування інших виробників, ці системи починають помилятися й приймати одні пристрою за інші, а при керуванні цими пристроями підтримують тільки їхні основні функції, а багато корисних додаткових функцій, які відрізняють даний пристрій від інших, система керування просто не розуміє й, тому, не може ними скористатися.

Для виправлення цього недоліку розроблювачі систем керування включають підтримку не тільки стандартних баз MIB1, MIB2 і RMONMIB, але й численних часток MIB фірм-виробників. Лідер у цій області – система Spectrum, що підтримує близько 1000 баз MIB різних виробників.

Іншим способом більше якісної підтримки конкретної апаратури є використання на основі якої-небудь платформи керування додатка тої фірми, що випускає це встаткування. Провідні компанії – виробники комунікаційного встаткування – розробили й поставляють досить складні й багатифункціональні системи керування для свого встаткування. До найбільш відомих систем цього

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

класу відносяться Optivity компанії BayNetworks, CiscoWorks компанії CiscoSystems, Transcend компанії 3Com. Система Optivity, наприклад, дозволяє робити моніторинг і управляти мережами, що складаються з маршрутизаторів, комутаторів і концентраторів компанії BayNetwork, повністю використовуючи всі їхні можливості й властивості. Устаткування інших виробників підтримується на рівні базових функцій керування. Система Optivity працює на платформах OpenView компанії Hewlett-Packard і SunNetManager (попередник Solstice) компанії SunSoft. Однак, робота на основі якої-небудь платформи керування з декількома системами, такими як Optivity, занадто складна й вимагає, щоб комп'ютери, на яких все це буде працювати, мали дуже потужні процесори й велику оперативну пам'ять.

Проте, якщо в мережі переважає встаткування від якого-небудь одного виробника, то наявність додатків керування цього виробника для якої-небудь популярної платформи керування дозволяє адміністраторам мережі успішно вирішувати багато завдань. Тому розроблювачі платформ керування поставляють разом з ними інструментальні засоби, що спрощують розробку додатків, а наявність таких додатків і їхня кількість вважаються дуже важливим фактором при виборі платформи керування.

Відкритість платформи керування залежить також від форми зберігання зібраних даних про стан мережі. Більшість платформ-лідерів дозволяють зберігати дані в комерційних базах даних, таких як Oracle, Ingres або Informix. Використання універсальних СУБД знижує швидкість роботи системи керування в порівнянні зі зберіганням даних у файлах операційної системи, але зате дозволяє обробляти ці дані будь-якими додатками, що вміють працювати із цими СУБД.

### **Система керування мережами Optivity**

Система керування Optivity компанії BayNetworks випускається в різних варіантах, що відрізняються набором функціональних властивостей і програмно-апаратних платформ.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Версія **OptivityEnterprise** працює на RISC-комп'ютерах у середовищах SunNetManager, HPOpenViewNetworkNodeManager і IBMNetViewAIX/6000. Ця версія призначена для великих корпоративних мереж з кількістю вузлів більше 1000, має високий ступінь масштабованості й найбільш повним набором функцій. Складається з наступних підсистем:

– OptivityLAN для керування локальними мережами, комутаторами й концентраторами.

– OptivityInternetnetwork для керування мережами із застосуванням маршрутизаторів.

– OptivityDesignandAnalysis підтримує функції планування й аналізу мережі.

– Крім цих підсистем склад OptivityEnterprise може бути доповнений підсистемами ATMNetworkManagementApplication для керування мережами, побудованими на основі АТМ-комутаторів. Ця підсистема розташовується в середовищі SunNetManager разом з іншими компонентами Optivity і дозволяє контролювати й управляти пристроями LattisCell і EtherCell, а також створювати віртуальні мережі.

Версія **OptivityCampus** працює на персональних комп'ютерах із процесором Intel у середовищах HPOpenViewforWindows і Novell Open Enterprise ServerManagementSystem. Ця версія призначена для керування мережами середніх розмірів (від 150 до 1000), що складаються з концентраторів, комутаторів і маршрутизаторів.

Версія **OptivityWorkgroup** працює в середовищі MSWindows на персональних комп'ютерах із процесором Intel і призначена для керування невеликими мережами (до 200 вузлів), що складаються з концентраторів, комутаторів і маршрутизаторів.

У своїй роботі система Optivity опирається на функціональні можливості **агентів SNMP**, убудованих у комунікаційні пристрої.

Існує три версії агентів – Standard, Advanced і AdvancedAnalyzer.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Агенти **AdvancedAnalyzer** реалізують найбільш розвинену на сьогоднішній день промислову технологію убудованих агентів, включаючи повну підтримку всіх груп стандарту RMON, а також засобу SuperRMON. Засоби SuperRMON розширюють можливості стандарту RMON на 1 рівень семирівневої моделі (для контролю портів) і на 3 рівень.

Агенти **Advanced** підтримують розвинені властивості убудованого керування – пороги, захист доступу, автотопологію, а також чотири групи змінних RMON.

Агенти **Standard** забезпечують тільки базові засоби керування й збору статистики для концентраторів.

Розглянемо докладніше властивості версії середнього класу – **OptivityforHPOpenView/Windows**.

Як і інші версії Optivity, дана версія надає повний набір засобів для керування транспортними функціями мережі як єдиною, погодженою системою, а не набором незв'язаних пристроїв. Система Optivity дає загальну картину корпоративної мережі за рахунок відбиття й керування взаємозв'язками між концентраторами, комутаторами, маршрутизаторами, мостами й кінцевими станціями.

Optivity легко інтегрується із платформою HPOpenView. У цій системі об'єднані засоби керування маршрутизаторами й підтримка стандарту RMON, що дозволяє користувачам збирати деталізовану інформацію про відмови, помилки, продуктивності й діагностиці в будь-якому місці мережі. Динамічне відображення стану мережі дозволяє легко одержувати точну інформацію з кожного порту.

OptivityforOpenViewforWindows підтримує всю лінію продуктів BayNetworks.

### **Динамічне виявлення конфігурації мережі**

У системі Optivity реалізована запатентована технологія динамічного виявлення конфігурації мережі, називана Autotopology. Ця технологія добре

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

інтегрується з поданням мережі платформою OpenView, у результаті чого адміністратор може користуватися серією вкладених подань мережі зі зростаючим ступенем деталізації.

Мережні менеджери можуть подорожувати по поданнях мережі з метою збору й збереження таких даних, як діагностика на MAC-рівні, коефіцієнти використання сегментів і портів, дані про помилки й інші. Будь-які зміни конфігурації мережі автоматично відбиваються в поданнях мережі, постачаючи адміністратора точною інформацією.

– Плоске подання мережі (flatnetworkview) показує з'єднання між собою сегментів Ethernet або кілець TokenRing, здійснене за допомогою мостів, комутаторів або маршрутизаторів.

– Сегментне подання (segmentview) відображає фізичні з'єднання між концентраторами в сегментах Ethernet або TokenRing, зв'язаних мостами або комутаторами.

### **Програмне конфігурування мережі**

До складу Optivity входить додаток LANarchitect, що істотно спрощує процес конфігурування мережі. Цей додаток працює із пристроями сімейств System, Distributed і LattisSwitchSystem і дозволяє приписувати визначені порти, слоти або кластери портів різним робочим групам (тобто сегментам або кільцям), поза залежністю від їхнього фізичного розташування. У результаті утвориться гнучка, програмно керована конфігурація мережі.

Додаток LANarchitect пов'язане з іконками пристроїв System, Distributed і LattisSwitchSystem на карті мережі й при вказівці на них представляє всі робочі групи, що відносяться до даного пристрою у формі каталогу файлів. При відкритті каталогу показуються всі порти й кластери, що втримуються в даній робочій групі, і адміністратор, використовуючи техніку drag-and-drop, може додавати нові порти або кластери, переміщати їх між робочими групами або створювати нові робочі групи. При цьому створюються нові з'єднання в керованих пристроях, які підтримуються убудованим у них програмним

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

забезпеченням, і тим самим виключається потреба фізичних перекомутацій у мережі.

За допомогою компонента LANarchitect адміністратор може передати вказівку модулю керування DCE (DataCollectionEngine), що може встановлюватися в пристроях System і Distributed, проводити детальний аналіз і обробку пакетів у певних сегментах Ethernet або кільцях TokenRing.

### **Інтегроване керування маршрутизаторами**

Система Optivity включає інтегрований додаток RouterMan для контролю й керування маршрутизаторами BayNetworks, Cisco і іншу, підтримуючу базу керуючої інформації MIB-II. Даний додаток має простий графічний інтерфейс із кольоровим кодуванням, що спрощує операції по стандартах SNMP і MIB. Автоматично відображаються всі протоколи й інтерфейси, підтримувані маршрутизатором, у тому числі IP, IPX, DECnetIV, XNS, AppleTalk, BanyanVINES, Ethernet, TokenRing, FDDI і WAN-протоколи на послідовних лініях.

Адміністратор може встановити умови попереджень і тривог для раннього оповіщення про виникаючі або потенційні проблеми.

За допомогою RouterMan можна конфігурувати маршрутизатори BayNetworks і Cisco, а також відслідковувати зміни конфігурації, порівнювати різні версії конфігураційних файлів і робити реконфігурацію за протоколом SNMP.

### **Аналіз і керування продуктивністю на основі стандарту RMON**

До складу Optivity входять додатки, що підтримують передовий промисловий стандарт RMON, що забезпечує для мереж Ethernet і TokenRing засобу виявлення відмов і збоїв, аналізу продуктивності й потенційних проблем.

– *DecodeMan* – утиліта, розроблена спільно BayNetworks і NetworkGeneral, що дозволяє декодувати і аналізувати трафік у віддалених локальних мережах. DecodeMan заснований на продукті FoundationManager компанії NetworkGeneral (одного з лідерів у виробництві мережних аналізаторів) і дозволяє повне

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

декодування пакетів на всіх 7 рівнях для більшості основних протоколів (включаючи IP, IPX, AppleTalk, DECnet, VINES, OSI, SNA і NetBIOS). За допомогою цієї утиліти можна аналізувати й розв'язувати проблеми, що виникають у віддалених сегментах, сидячи за центральною консоллю системи керування.

DecodeMan використовує можливості "зонд-в-концентраторі", надавані агентами типу AdvancedAnalyzer, убудованими в пристрої BayNetworks.

– *NodalView* – засіб для збору детальної RMON-статистики про помилки й продуктивність всіх вузлів сегмента Ethernet або кільця TokenRing. Кінцеві станції відображаються в компактному виді у формі списку, дозволяючи адміністраторові швидко визначити їхній статус і визначити потенційні проблемні вузли. Функції сортування дають можливість упорядкувати вузли за різними показниками – рівню трафіку, інтенсивності помилок, IP-адресам і іншим, дозволяючи швидко виявити вузли, що створюють найбільший трафік, що частіше генерують помилкові пакети, "діри" у послідовності IP-адрес і інші проблеми. Інформація про продуктивність сегмента або окремого вузла може бути представлена у вигляді графіка, круговий або східчастої гістограмми.

#### **Аналіз помилок і проблем, що попереджає**

Система Optivity включає засоби, що дозволяють виявляти умови виникнення проблем, попереджаючи їхній негативний вплив на продуктивність мережі.

Граничні значення дозволяють адміністраторові встановити відповідні рівні для специфічних помилок, завантаженості й активності сегментів і вузлів мереж Ethernet і TokenRing. При досягненні порога система автоматично генерує повідомлення про цю подію для адміністратора. Ці RMON-події можуть також ініціювати визначені адміністратором дії, такі як ізоляція порту, подача звукового сигналу тривоги, відправлення повідомлення по електронній пошті або дзвінок по телефону.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

## Керування пристроями в реальному масштабі часу

Optivity включає також розвинені засоби для керування й контролю концентраторами на рівні портів у реальному масштабі часу. Запатентований графічний інтерфейс ExpandedView забезпечує точне поточне зображення обраного концентратора, з усіма встановленими в нього модулями й активними світлодіодами. Адміністратор може одержати різноманітну інформацію за допомогою пунктів меню Fault, Performance або Configuration до рівня порту. Можна також дозволити або заборонити з'єднання, або ж тимчасово ізолювати порт для більше точного мережного керування.

### Додаткові керуючі засоби й утиліти

– *FindNodes* – ця утиліта допомагає швидко знайти користувачів мережі за неповним даними, наприклад, по його MAC-адресі або по псевдоніму. FindNodes негайно надає інформацію про те, до якого концентратора, слоту й порту підключений даний користувач.

– *AgentManager* – автоматизує процес поширення й завантаження нових і оновлених версій програмних агентів по корпоративній мережі. Це додаток також проводить автоматичну інвентаризацію агентів, убудованих у концентратори, постачаючи адміністратора повним списком типів модулів керування, версій агентів і версій мікропрограм.

– *AllowedNodes* – забезпечує засоби захисту мережі від несанкціонованого доступу. Якщо станція з несанкціонованою адресою намагається ввійти в мережу, то систему можна запрограмувати таким чином, що вона пошле повідомлення на консоль адміністратора або ж автоматично відключить порт.

### Продукти моніторингу й аналізу мереж компанії NetworkGeneral

Продукти, що випускаються компанією NetworkGeneral, призначені для роботи в трьох секторах ринку засобів моніторингу й аналізу мереж:

– Сектор недорогих систем для не дуже критичних до збоїв мереж широкого класу. Компанія Network General випускає для цього сектору продукти Foundation Agent, Foundation Probe, Foundation Manager.

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– Сектор дорогих систем вищого класу, призначених для аналізу й моніторингу мереж, що пред'являють максимально можливі вимоги по забезпеченню надійності й продуктивності. Такі системи звичайно є розподіленими. У цьому секторі позиціонується сімейство DistributedSnifferSystem.

– Сектор переносних систем аналізу й моніторингу: NotebookSnifferNetworkAnalyzer і ExpertSnifferNetworkAnalyzer.

### **Foundation Agent, Foundation Probe, Foundation Manager**

Це сімейство являє собою закінчену систему мережного моніторингу на базі RMON і містить у собі два типи агентів-моніторів FoundationAgent, FoundationProbe і консоль оператора FoundationManager. Агент FoundationAgent являє собою програмний продукт, установлюваний на стандартний IBM-сумісний комп'ютер, і здатний збирати інформацію в сегментах Ethernet або TokenRing.

FoundationAgent підтримує всі можливості стандартного SNMP-агента, всі 9 (або 10 для TokenRing) груп RMON, розвинену систему збору й фільтрації даних. Відносно недавно з'явився новий представник цього класу агентів – FoundationAgentMulti-Port, що дозволяє за допомогою одного комп'ютера збирати інформацію із чотирьох сегментів Ethernet або трьох сегментів TokenRing.

FoundationProbe – сертифікований комп'ютер із сертифікованою мережною платою й передвстановленим програмним забезпеченням FoundationAgent відповідного типу.

FoundationAgent і FoundationProbe звичайно функціонують у безмоніторном і безклавіатурному режимі, оскільки управляються програмним забезпеченням FoundationManager. У принципі, не виключена можливість виконання на агентах RMON ще якої-небудь діяльності, крім аналізу сегментів, однак робити це настійно не рекомендується через необхідний агентіві запасу обчислювальної потужності.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Програмне забезпечення консолі FoundationManager поставляється у двох варіантах – для Windows і для Unix.

Консоль FoundationManager дозволяє відображати в графічному виді статистику по всіх контрольованих сегментах мережі, автоматично визначати усереднені параметри мережі й реагувати на перевищення припустимих меж параметрів (наприклад, запускати програму-оброблювач, ініціювати SNMP-trap і SNA-alarm), будувати за зібраним даними RMON графічну динамічну карту трафіку між станціями.

Засоби аналізу трендів і автоматичної генерації звітів дозволяють значно полегшити аналіз тенденцій росту мережі за тривалий час. Надаються й деякі унікальні можливості, наприклад, аналізу використання мережі різними протоколами й нагромадження статистики в агенті в плинні місяця, що дозволяє значно підвищити автономність агента. Прозора інтеграція з іншим програмним забезпеченням від NetworkGeneral дозволяє передавати захоплені пакети даних для більше глибокого аналізу підсистемі ExpertAnalysis або використовувати розвинені засоби аналізу протоколів підсистеми Protocol Interpreter.

FoundationManager здатний працювати разом з кожним з RMON-агентів, однак при цьому будуть використані тільки стандартні можливості агента. Деякі з перерахованих вище можливостей є специфічними й реалізуються тільки при роботі разом з агентами від NetworkGeneral.

### **Сімейство продуктів Distributed Sniffer System**

DistributedSnifferSystem (DSS) – являє собою систему, що складається з декількох розподілених по мережі апаратних компонентів і програмного забезпечення, необхідного для безперервного аналізу всіх, включаючи віддалені, сегментів мережі.

Система DSS будується з компонентів двох типів – Sniffer Server (SS) і SniffMasterConsole (SM).

Пристрої типу Sniffer Server являють собою спеціалізований програмно-апаратний комплекс, побудований на базі комп'ютера, спеціалізованих мережних

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

карт і додаткових інтерфейсів для взаємодії з консоллю. На сьогодні доступні Sniffer Server для аналізу наступних мережних технологій LAN і WAN:

- Ethernet (10 Base-T, 10 Base-2, 10 Base-5);
- Token Ring (UTP, STP);
- FDDI (multimode fiber);
- Fast Ethernet (100 Base-TX, 100 Base-T4);
- ATM (OC-3a multi-mode fiber, OC-3z copper, DS-3 coax, E-3 coax);
- глобальних мереж (RS-232/P5-449/Ч.35, X.25, framerelay, ISDNBRI і PRI до швидкостей E1 і T1).

Як інтерфейси для взаємодії з консоллю можуть бути використані карти Ethernet, TokenRing або послідовний порт. Таким чином, істи можливість контролювати сегмент практично будь-якої мережної топології й використовувати різні середовища взаємодії з консоллю, включаючи з'єднання по модему.

SniffMasterConsole – програмне забезпечення, що виконує функції керування всією системою DSS. SniffMaster випускається у варіантах для роботи з MS Windows і для роботи з різними варіантами Unix і систем керування мережами (HP-UX із HP OpenView, AIX з NetView, SunOS або Solaris з SunNetManager). Система SniffMaster надає користувачеві розвинутий графічний інтерфейс керування серверами Sniffer Server. Одна єдина консоль SniffMaster здатна управляти будь-якою кількістю серверів Sniffer Server будь-яких мережних топологій. Крім того, можлива установка декількох консолей для керування одним сервером Sniffer Server або їхньою групою, що дозволяє створювати запасні пункти контролю мережі й дозволяє декільком експертам-адміністраторам спільно вирішувати виникаючі завдання.

Система DSS загалом повторює типову схему побудови розподіленої системи аналізу мереж. Однак є кілька особливостей, які вивели саме цю систему в лідери ринку.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

По-перше, це – концепція взаємодії сервера й консолі аналізу. Розвиваючи концепцію RMON, сервер аналізу Sniffer Server діє як повністю незалежний пристрій, не тільки збираючи інформацію про функціонування мережі (подібно агентіві SNMP) або проводячи її первинну обробку (подібно агентіві RMON), але й виконуючи її повний аналіз на всіх семи рівнях мережної моделі ISO/OSI. Більше того, сервер бере на себе всі функції по відображенню інформації, формуючи якийсь віртуальний екран з інформацією про функціонування конкретного сегмента мережі. Далі цей віртуальний екран передається на консоль, де й відображається в окремому вікні. Для керування сервером аналізу є можливість пересилання команд із консолі на сервер. Будь-які обміни даними між сервером і консоллю оптимізуються, наприклад, при передачі віртуального екрана реально передаються тільки зкопресійовані дані, що представляють собою зміна вмісту цього екрана в порівнянні з попереднім пересиланням.

Другою особливістю є використання для зв'язку між консоллю й сервером фірмового протоколу передачі даних NGCP (Network General Communication Protocol) замість SNMP. Протокол NGCP базується на протоколі TCP і, на відміну від SNMP, є захищеним, тобто всі передані за допомогою NGCP дані передаються в зашифрованому виді. З огляду на, що при роботі систем типу Sniffer Server будь-яка циркулююча в мережі інформація, включаючи адреси станцій, запити до серверів баз даних і відповіді від них і навіть паролі доступу, можуть бути легко перехоплені й піддані аналізу, можливість використання захищених методів зв'язку виявляється дуже доречною. NGCP може бути використаний як для зв'язку по локальній мережі, так і по що комутируються й виділеним каналам (у цьому випадку використовується протокол NGCP-serial, подібний PPP).

Програмне забезпечення Sniffer Server складається із трьох підсистем – моніторингу, інтерпретації протоколів і експертного аналізу. Підсистема моніторингу являє собою систему відображення поточного стану мережі, що дозволяє одержувати статистику по кожній зі станцій і сегментів мереж по

кожному з використовуваних протоколів. Дві інші підсистеми заслуговують окремого обговорення.

### **Підсистема Protocol Interpreter**

У функції цієї підсистеми входить аналіз захоплених пакетів і як можна більше повна інтерпретація кожного з полів заголовків пакетів і його вмісту. Компанія NetworkGeneral створила саму потужну підсистему подібного типу – Protocol Interpreter здатний повністю декодувати більше 200 протоколів всіх семи рівнів моделі ISO/OSI (TCP/IP, IPX/SPX, NCP, DECnetSunNFS, X-Windows, сімейство протоколів SNA/IBM, AppleTalk, BanyanVINES, OSI, XNS, X.25, різні протоколи міжмережної взаємодії). При цьому відображення інформації можливо в одному із трьох режимів – загальному, деталізованому й шістнадцятковому.

Загальний режим передбачає відображення лише найбільш важливої інформації про пакет – адреси відправника й одержувача, назва вищого по ієрархії ISO/OSI протоколу, використаного в пакеті, коротка характеристика вмісту пакета (наприклад, читання даних). У цьому режимі кожний з пакетів займає один рядок у вікні інтерпретатора протоколів.

Деталізований режим передбачає відображення повної розшифровки всієї ієрархії протоколів, при цьому кожний з рівнів ієрархії відображається своїм кольором, дається розшифровка на близькому до природного англійського, кожного з полів пакетів всіх рівнів ієрархії й докладно описуються дані пакета.

При роботі в шістнадцятковому режимі пакети відображаються в шістнадцятковому виді, а також у вигляді символів кодувань ASCII або EBCDIC.

Для розроблювачів поставляється спеціальний інструментарій – PDK (ProtocolDevelopmentKit), що дозволяє створювати модулі підтримки нових протоколів для Protocol Interpreter.

### **Підсистема Expert Analysis**

Серцем серверів Sniffer Server є експертна система аналізу мережі ExpertAnalysis. В основі системи лежить унікальна база знань, накопичена фахівцями компанії NetworkGeneral з 1986 року й заснована на досвіді роботи з

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

користувачами різних мереж і розробках груп Станфордського й Массачусетського університетів, а також компанії NipponTelephoneandTelegraph (NTT).

Основне призначення системи – скорочення часу простою мережі й ліквідація вузьких місць мережі за допомогою автоматичної ідентифікації аномальних явищ і автоматичної генерації методів їхнього дозволу. Система експертного аналізу надає діагностичну інформацію трьох категорій:

– Симптом – подія в мережі, якому адміністратор мережі повинен приділити додаткову увагу (наприклад, фізична помилка при звертанні до вузла мережі або одинична повторна передача файлу). Не обов'язково означає виникнення проблеми, однак при високому рівні періодичності вимагає уваги адміністратора.

– Діагноз – кількаразове повторення симптому, підмет обов'язковому розслідуванню з боку адміністратора мережі. Звичайно діагноз описує ситуації, що характеризують серйозні несправності в мережі (наприклад, дублюєма мережна адреса).

– Пояснення – контекстно-контекстно-залежний експертний висновок системи аналізу для кожного симптому або діагнозу. Пояснення містить опис декількох можливих причин сформованої ситуації, обґрунтування подібного висновку й рекомендації з їхнього усунення.

У системі є можливості доповнення існуючої бази знань специфічними даними, отриманими адміністратором мережі в процесі її використання.

Система автоматичного аналізу ExpertAnalysis заснована на унікальній багатозадачній технології аналізу пакетів, що кількома словами можна описати в такий спосіб:

– Циркулюючі в мережі пакети безупинно захоплюються й містяться в кільцевий буфер захвата (перше завдання).

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– Одночасно із цим кілька завдань-аналізаторів протоколів (по однієї на кожне із сімейств протоколів) сканують буфер захвата й генерують інформацію в єдиному внутрішньому форматі.

– Ця стандартизована інформація надходить на групу завдань-експертів. Кожний з експертів є таким лише у своїй вузькій області, наприклад, у знанні протоколу взаємодії клієнта із сервером NetWare. Якщо експерт знаходить подію, пов'язане з його областю інтересів, він генерує деякий відповідний об'єкт (наприклад, "користувач Guest сервера IBSO") в базі даних про мережу, називаної BlackboardKnowledgeBase, і зв'язує його з відповідними об'єктами більшого низького рівня (у нашій випадку – з адресою IPX станції користувача Guest, пов'язаним з MAC-адресою плати цієї станції, і із сервером IBSO, пов'язаним з адресами IPX і IP, а також з MAC-адресами мережних плат сервера, а також з усіма вже, що ввійшли на сервер користувачами, принт-серверами, і т.д.). У результаті виникає деяка складна структура, що відображає всі об'єкти мережі, що відносяться до деякого протоколу й всі можливі зв'язки між ними на всіх 7 рівнях моделі ISO/OSI.

– Існує друга група завдань-експертів, що постійно аналізує стан бази даних і повідомлення, що видають, про ненормальне функціонування мережі (симптоми або діагнози). У цілому система ExpertAnalysis знайома з більш ніж 200 різними проблемами функціонування мережі.

Останній елемент цієї системи – експерти, що генерують докладний опис проблеми й методи її виправлення. При цьому ці експерти сканують базу даних і підставляють у видавані рекомендації реальні об'єкти мережі – MAC-адреси, назви серверів, імена завдань і т.д.

Подібна багатозадачна система аналізу є унікальною на ринку аналізаторів. Деякі із продуктів компаній-конкурентів також пропонують системи з елементами штучного інтелекту, однак принципи побудови їх зовсім інші. Принцип побудови аналізатора на основі "круглого стола експертів", реалізованого на основі спеціалізованого багатозадачного ядра, дозволяє

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

проводити аналіз із дуже високою продуктивністю. Завдання-експерти одночасно ведуть обробку інформації з мері її надходження, а використовувані евристичні правила дозволяють швидко локалізувати коло експертів по кожному з конкретних випадків і тимчасово призупинити роботу інших експертів.

Принцип, що вимагає послідовного застосування всіх евристичних правил, веде до зниження продуктивності аналізу на тих же потужностях процесора й до необхідності використання могутнішого процесора для забезпечення порівнянної продуктивності.

Система ExpertAnalysis забезпечує те, що компанія NetworkGeneral називає активним аналізом. Для розуміння цієї концепції розглянемо обробку того самого помилкової події в мережі системами традиційного пасивного аналізу, і системою активного аналізу.

Нехай у мережі в 3:00 ночі відбувся ширококомовний шторм, що викликав в 3:05 збій системи створення архівних копій баз даних. До 4:00 шторм припиняється й параметри системи входять у норму. У випадку роботи в мережі системи пасивного аналізу трафіку які прийшли на роботу до 8:00 адміністратори не мають для аналізу нічого, крім інформації про другий збій і, у найкращому разі, загальної статистики по трафіку за ніч – розмір будь-якого буфера захвата не дозволить зберігати весь трафік, що пройшов по мережі за ніч. Імовірність ліквідації причини, яка привела до ширококомовного шторму в такій ситуації вкрай мала.

А тепер розглянемо реакцію на ті ж події системи активного аналізу. В 3:00, відразу послу початку ширококомовного шторму, система активного аналізу фіксує настання нестандартної ситуації, активує відповідний експерт і фіксує видану їм інформацію про подію і його причини в базі даних. В 3:05 фіксується нова нестандартна ситуація, зв'язана зі збоєм системи архівування, і фіксується відповідна інформація. У результаті в 8:00 адміністратори одержують повний опис виниклих проблем, їхніх причин і рекомендації з усунення цих причин.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## **Портативні аналізатори**

Портативний варіант аналізатора, майже аналогічний по своїх можливостях DSS, реалізований у продуктах серії ExpertSnifferAnalyzer (ESA), відомий також як TurboSnifferAnalyzer. При значно меншій, чим продукти серії DSS, вартості, ESA надають адміністраторові ті ж можливості, що й повномасштабна DSS, але тільки для того сегмента мережі, до якої ESA підключений у цей момент. Існуючі версії забезпечують повний аналіз, інтерпретацію протоколів, а так само моніторинг підключеного сегмента мережі або лінії міжсегментного зв'язку. При цьому підтримуються всі ті ж мережні топології, що й для систем DSS. Як правило, ESA використовуються для періодичної перевірки некритичних сегментів мережі, на яких недоцільно постійно використовувати агент-аналізатор.

Існує й ще більш компактна версія аналізатора – Notebook Sniffer Analyzer (NSA), реалізований на базі портативного комп'ютера класу notebook, спеціальної карти стандарту PCMCIA Турі II і програмного забезпечення, аналогічного продуктам серії ESA з можливістю підрахунку числа колізій. Здатний виконувати всі функції по аналізі мереж на базі Ethernet і TokenRing. Є гарним рішенням для активно, що переміщається фахівця, що використовує notebook як портативний комп'ютер загального призначення.

Відносно недавно вийшла більше "полегшена" версія Notebook Sniffer Analyzer – Notebook Sniffer Analyzer Lite, що володіє всіма можливостями повноцінного Notebook Sniffer Analyzer, але тільки відносно мережних середовищ на базі Novell Open Enterprise Server.

## **Додаткові продукти**

Компанія Network General пропонує ще кілька продуктів, що дозволяють вирішити ті або інші завдання аналізу мереж більш ефективно.

Потужний засіб, призначений для аналізу систем з використанням баз даних типу клієнт/сервер – Sniffer Network Analyzer Database Module (for Oracle). Додавання цього продукту до стандартного інтерпретатора протоколів дозволяє

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

аналізувати пакети протоколів SQL\*Netv.2, що виконує функції обміну між сервером баз даних Oracle і клієнтом. Можливості продукту по глибині аналізу аналогічні можливостям Protocol Interpreter. Як видно з назви, даний продукт призначений для роботи в системах на базі Oracle, причому поки підтримується лише нижній рівень SQL\*Net – TransportNetworkSubstrate (TNS). Однак, що навіть існує в даний момент версія дозволяє вимірювати часи відповідей на запити, робити декодування мережних протоколів разом із протоколом TNS, відображати файли конфігурації Oracle і робити трасування пакетів. Всі ці можливості різко знижують часи локалізації помилок і вузьких місць у системах клієнт-сервер. Найближчим часом очікується вихід версій для Sybase, MicrosoftSQLServer, InformixOnLine і IBMDB2, а також введення в систему експертних можливостей у стилі ExpertAnaLyzer.

Адміністраторові великої мережі також може бути корисний NGCSnifferReporter – спеціалізований засіб складання звітів на основі даних, зібраних іншими додатками аналізу даних. Дані, зібрані в різних крапках мережі різними системами аналізу (ESA, NSA, DSS, FoundationManager), за допомогою SnifferReporter можуть бути піддані одночасному аналізу, що дозволяє виявити додаткові закономірності функціонування мережі. NGCSnifferReporter являє собою програмне забезпечення, що функціонує під MS Windows.

### **Аналізатор протоколів LANalyser**

LANalyser поставляється у вигляді мережної плати й програмного забезпечення, які необхідно встановлювати на персональному комп'ютері, або у вигляді ПК, із уже встановленими платою й програмним забезпеченням.

LANalyser має розвитий зручний інтерфейс із користувачем, за допомогою якого встановлюються обраний режим роботи. Меню **ApplicationLANalyser** є основним засобом настроювання режиму перехоплення й містить варіанти вибору набору протоколів, фільтрів, ініціаторів, аварійних сигналів і т.д. Даний аналізатор може працювати із протоколами: NetBIOS, SMB, NCP, NCPBurst, TCP/IP, DECnet, BanyanVINES, AppleTalk, XNS, SunNFS, ISO, EGP, NIS, SNA і

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

деякими іншими. З меню Application можна або вибрати й зконфігурувати спеціальні тестові комплекти додатків, або вибрати один із заздалегідь певних тестових комплектів додатків для TokenRing або Ethernet.

За допомогою LANalyser можуть бути визначені до 9 каналів прийому й до 6 каналів передачі. Канал прийому – це в сутності фільтри для всієї інформації, що користувач бажає одержувати в ході сеансу аналізу протоколу. Канали передачі дозволяють згенерувати в мережі потоки даних заданої структури.

Є можливість динамічно модифікувати параметри тестового комплекту додатка. Використовуючи зручний інтерфейс ви можете, наприклад, указати дисковий файл для реєстрації основних статистичних параметрів даних, що збираються, або режим роздруківки. LANalyser не має у своєму розпорядженні які-небудь засоби генерації звітів, але файли статистики можна імпортувати в різні додатки.

В LANalyser передбачені наступні режими відображення результатів аналізу мережі:

– *Режим глобального відображення* надає в розпорядження користувача статистичну інформацію про мережу в цілому – загальна кількість пакетів для кожного типу протоколів, процентне співвідношення трафіков різного виду, у тому числі ширококомовного, трафіку помилкових пакетів і т.п. На цьому ж екрані розміщені діаграми інтенсивностей трафіку різного виду.

– *Режим роздільного відображення* забезпечує статистичну інформацію з окремих пакетів, захопленим по каналах прийому.

– *Режим відображення використання* надає в розпорядження користувача універсальну картину використання всіма активними каналами смуги пропускання мережі.

– *Режим відображення станцій* виводить статистику по взаємодії окремих станцій.

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Функція **DisplayPacketTrace** дозволяє переглядати перехоплені пакети в загальному хронологічному переліку або в детальному поданні пакетів. Перебуваючи в цьому режимі, можна переглянути поточне трасування або завантажити раніше виконані трасування з диска.

Функція **TestNetworkCabling** дозволяє виконати серію мережних тестів, у тому числі базовий тест кабелю, тести з'єднань і стану кільця (для TokenRing).

Функція **Utilities** активізує підменю, що включає наступні утиліти, які служать для адаптації того або іншого прикладного тестового набору до потреб користувача:

- *Name* – дозволяє привласнювати імена конкретним адресам Ethernet і TokenRing;
- *Genname* – автоматично генерує файл найменувань для ряду конкретних адрес вузлів мережі;
- *Stats* – дозволяє переглядати збережений раніше файл, отриманий у результаті виконання певного тесту в основних режимах відображення;
- *Template* – визначає шаблони фільтрів для завдання каналів передачі й прийому.

LANalyser має деякі можливості, що підвищують ефективність його роботи в мережах Ethernet і TokenRing. У складі LANalyser поставляється тестовий комплект, названий ERRMON, що настраюється таким чином, щоб канали прийому могли автоматично фіксувати ту або іншу типову помилку мережі Ethernet або TokenRing. Інший тестовий набір, названий SEGMENTS, призначений для аналізу мереж, побудованих на основі мостів і комутаторів.

В останню версію LANalyser включена також експертна система, що робить користувачеві допомога в пошуку несправностей.

### **Багатофункціональний пристрій Comras**

На відміну від кабельного сканера, пристрій Comras дозволяє швидко вирішити більшість проблем, що виникають перед адміністратором мережі, наприклад, не тільки виявити місце й причину порушення роботи кабельної

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

системи, але й визначити ділянки мережі з найбільш напруженим трафіком, ступінь завантаженості процесора сервера й деякі інші параметри.

Досить натиснути одну кнопку "DIAGNOSE" і Compas проведе серію необхідних тестів, не тільки визначить причину несправності, але й укаже можливі способи її усунення.

Compas можна підключати в будь-якому місці мережі. Він сам визначає місце включення й запускає відповідні тести. За допомогою унікальної функції NetTap можна підключити Compas між будь-якими двома мережними пристроями, наприклад, між робочою станцією й концентратором або файл-сервером і концентратором і за допомогою функції NetTap аналізувати трафік між будь-яким мережним пристроєм і концентратором. Дана функція дозволяє тестувати роботу концентраторів, що використовують технологію SwitchedEthernet.

Як аналізатор протоколів Compas дозволяє проводити моніторинг мережного трафіку й визначати несправності на рівні протоколів. Compas визначає трафік, кількість помилок, мережні пристрої, що створюють основний трафік, джерела помилок і ширококомовних пакетів. Можна переглядати піки завантаження й помилок протягом тривалого періоду. Compas розпізнає всі протоколи, використовувані в даному сегменті, у тому числі: NovellIPX, IP, DECLAVC, DECnet, AppleTalkII (APP2), XeroxXNS, BanyanVINES, ISO і ARP і визначає сукупний відсоток утилізації для кожного протоколу.

Як кабельний сканер Compas дозволяє проводити діагностику кабельної мережі. Compas вимірює наступні параметри кабелів: NEXT, імпеданс, рівень електромагнітних шумів і схему розведення кабелю. Маючи два рознімання RJ-45, Compas може тестувати навіть кроссировочні кабелі, що часто є причиною несправності мереж.

Compas показує детальну інформацію об файл-серверах з операційною системою NetWare з використанням CompasNetWareLoadableModule (NLM). Даний тест дозволяє переглядати значення утилізації процесора, переповнення

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

кеш-буферів, утилізації сервера, використовуваний фрейм і ін. Можна використовувати Comras для емуляції файл-сервера або робочої станції. Comras також дозволяє тестувати черги друку й роздруковувати результати всіх тестів на мережному принтері. Comras має один BNC і два RJ-45 рознімання. Прилад автоматично визначає, до якого рознімання підключений кабель.

### **Кабельні сканери**

Сімейство моделей PentaScanner компанії Microtest призначено для проведення сертифікації кабельних систем.

Модель кабельного сканера **PentaScannerCableAdmin** забезпечує сертифікацію кабельних систем категорії 5 рівнів точності I. Цей сканер призначений для пошуку несправностей кабельної системи адміністраторами ЛОМ і являє собою порівняно дешевий і простий у використанні прилад, що дозволяє швидко визначити несправність кабельної системи.

Кабельний сканер **PentaScanner+** призначений, головним чином, для фахівців компаній мережних інтеграторів або співробітників відділів автоматизацій підприємств, яким необхідно встановлювати й сертифікувати кабельні системи категорії 5. Стандарт TSB-67 вимагає виміру NEXT з обох кінців лінії. Використовуючи PentaScanner+ разом із двонаправленим інжектором – 2-WayInjector+, виміру NEXT можна робити з обох кінців лінії одночасно. При використанні Penta-Scanner+ спільно зі стандартним інжектором – SuperInjector+, необхідно міняти місцями PentaScanner+ і SuperInjector+ для проведення повної сертифікації лінії.

PentaScanner+ проводить всі необхідні тести для сертифікації кабельних мереж, включаючи визначення NEXT, загасання, відносини сигнал-шум, імпедансу, ємності й активного опору.

PentaScanner+ містить кілька частотних генераторів і вузькополосних приймачів, графічний дисплей на рідких кристалах і флеш-пам'ять для запису результатів тестування й нових версій програмного забезпечення. Як елемент живлення PentaScanner використовує акумуляторні батареї, що працюють без

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

підзарядки до 10 годин. Прилад містить рознімання для прямого приєднання до кабелю без використання додаткових адаптерів.

Для виміру перехресних наведень між крученими парами (NEXT) джерело сигналів – SuperInjector+, прилад поставляється в комплекті з PentaScanner+ – приєднується до передавальної пари й починає передавати в неї сигнали різної частоти. Приймач сигналів підключається до прийомної пари й вимірює сигнал, наведений у ній, порівнюючи його зі стандартними величинами. Перевагою вузькополосного приймача в PentaScanner+ є вимір "чистого" NEXT з відфільтровуванням всіх наведень і електричного шуму. Для виміру загасання PentaScanner+ використовує SuperInjector+ як віддалене джерело сигналів, що генерує серію сигналів різної частоти. PentaScanner+ у цей момент вимірює амплітуду цих сигналів на іншому кінці кабелю.

Остання модель сканерів сімейства PentaScanner – є сканером нового покоління, призначеного для тестування кабельних систем категорії 5 на частоті до 350 МГц. PentaScanner 350 являє собою найбільш прецизійний на сьогоднішній день кабельний сканер, що повністю відповідає Рівню точності II стандарту TSB-67. У пам'яті сканера PentaScanner 350 можуть зберігатися результати до 500 різних тестів.

Описані вище пристрої призначені для тестування кабельних систем на основі мідного кабелю. Для діагностики волоконно-оптичних кабелів компанія Microtest пропонує комплект FiberSolutionKit, що складається із двох приладів: вимірника оптичної потужності **FiberEye** і каліброваного світлового джерела **FiberLight**.

Ці прилади дозволяють тестувати мережі стандартів Ethernet, TokenRing і FiberDistributedDataInterface (FDDI).

FiberEye вимірює потужність світлового пучка, що входить або виходить із волоконно-оптичної лінії. Точний вимір оптичної потужності й втрати оптичного сигналу необхідні при інсталяції, технічному обслуговуванні й пошуку несправностей у волоконно-оптичних мережах. За допомогою FiberEye можна

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

також перевірити правильність роботи різних волоконно-оптичних компонентів, таких, як волоконно-оптичні концентратори, повторювачі й мережні адаптери. Дані про втрату сигналу допомагають визначити дефектні ділянки кабелю, несправні рознімання й коннектори.

FiberLight – каліброване світлове джерело, може бути використаний разом з FiberEye для забезпечення ефективності діагностики волоконно-оптичної мережі. FiberLight складається із двох джерел світлових імпульсів, кожний з яких має своє зовнішнє рознімання для підключення до кабелю. Одне джерело використовується для мереж Ethernet і TokenRing, а іншої для мереж FDDI.

### **Засоби моніторингу й аналізу компанії Fluke**

Компанія Fluke є одним із провідних виробників засобів діагностики й сертифікації кабельних систем, а також засобів моніторингу й аналізу мереж.

Повний спектр портативних вимірювальних засобів компанії Fluke включає:

- 610 CableMapper – пристрій для відображення кабельних з'єднань.
- 620 CableMeter – пристрій для тестування кабельних систем.
- DSP-100 DigitalCableMeter – пристрій для сертифікації кабелів категорії

5.

- OneTouchNetworkAssistant – пристрій для комплексної перевірки кабельної проводки, концентраторів, мережних адаптерів і моніторингу мережного трафіку.

- 67XLANMeter – сімейство приладів, що сполучають функції аналізатора протоколу, генератора трафіку й кабельного тестера.

- 68X EnterpriseLANMeter – сімейство приладів, призначене для аналізу протоколів у корпоративній мережі, що включає підтримку SNMP, RMON і аналіз трафіку у віддалених мережах.

Розглянемо докладно серію портативних приладів EnterpriseLANMeter, що володіють найбільш широкими функціональними можливостями по дослідженню мережі.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

## **Особливості 68X Enterprise LANMeter**

EnterpriseLANMeter являє собою портативний засіб, що підтримує протокол SNMP, що дозволяє йому виявляти причини відмови в мережі за межами локального сегмента, до якого підключений цей прилад. Підтримуються бази керуючої інформації MIB1, MIB2 і RMON.

Інструментальні засоби EnterpriseLANMeter підтримують стек протоколів TCP/IP, а також інші протоколи, використовувані в мережних операційних системах.

LANMeter зводить до мінімуму можливі відмови внесені новими компонентами мережі перевіряючи працездатність мережних інтерфейсних карт, кабелів, концентраторів і пристроїв мережного доступу (MAU – MediaAccessUnit) перед їхньою установкою в мережу. Він дозволяє також за допомогою ряду спеціальних тестів переконатися в правильності з'єднань і наявності доступу до мережних ресурсів через локальні й корпоративні підключення.

LANMeter швидко виявляє причини виникнення найпоширеніших для мереж Ethernet і TokenRing видів несправностей і визначає місцезнаходження ушкоджень кабельної проводки або некоректно працюючих пристроїв.

LANMeter надає користувачеві зручний і інтуїтивно зрозумілий інтерфейс, заснований на системі меню. Графічний інтерфейс користувача використовує 10-рядковий рідкокристалічний дисплей і індикатори стану на світлодіодах, що сповіщають користувача про найбільш загальні проблеми спостережуваних мереж. Є великий файл підказок операторові з рівневим доступом відповідно до контексту. Інформація про стан мережі представляється таким чином, що користувачі будь-якої кваліфікації можуть її швидко зрозуміти.

### **Функціональні можливості**

Як більшість найбільш дорогих аналізаторів мережних протоколів, EnterpriseLANMeter дозволяє провести аналіз у режимі реального часу функціонування мережі, виконуючи спеціалізовані тести. Одночасно виконується

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

тестування по двох групах випробувань: збір статистики про трафік в мережі в цілому, збір статистики про трафік окремих вузлів. У першу групу входять функції NetworkStatistics, ErrorStatistics і CollisionAnalysis (які виміряють, відповідно, загальний трафік, трафік помилкових кадрів і інтенсивність колізій у мережі), а в другу групу – функції визначення вузлів, що відправляють найбільшу кількість кадрів (Top Senders), що одержують найбільшу кількість кадрів (TopReceivers) і ширококомовних кадрів, що генерують найбільшу кількість, (Top Broadcasters).

Розглянемо функціональні можливості EnterpriseLANMeter на прикладі аналізу мереж Ethernet.

### **NetworkStatistics**

Ця функція дозволяє спостерігати загальний стан мережі за допомогою статистичної обробки й подання результатів за основними показниками працездатності мережі. До них відносяться ступінь використання (Utilization), рівень колізій (Collisions), рівень помилок і ширококомовного трафіку. EnterpriseLANMeter представляє результати вимірів у числовій і графічній формі.

### **ErrorStatistics**

Ця функція дозволяє відслідковувати всі типи й причини помилок. Результати представляються в числовій формі й у вигляді кругової діаграми, що показує відносний розподіл типів відмов по загальній їхній кількості. Типи відмов, яким передують маркер \*, можна виділити підсвічуванням на дисплеї, а потім клавішею з функцією "Zoomin" (збільшення) можна викликати на дисплей список станцій, що є джерелом цих збоїв.

### **Collisionanalysis**

Забезпечує інформацію про кількість і види колізій, відзначених на сегменті мережі, дозволяє визначити наявність і місцезнаходження проблеми. У режимі аналізу колізій на дисплей виводяться всі зареєстровані колізії, включаючи колізії заголовків і енергетичні "примари" (energyghosts) – наведення в кабелі, які займають частину смуги пропускання, заважаючи вузлам мережі

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

станціям передавати інформацію. Більшість аналізаторів мережних протоколів не мають можливість реєструвати кадри-примари.

### **Розподіл використовуваних мережних протоколів (Protocol Mix)**

На дисплеї відображається список основних протоколів в убутному порядку щодо процентного співвідношення кадрів, що містять пакети даного протоколу до загального числа кадрів у мережі. Підсвітивши протокол, що цікавить, і нажавши клавішу "ZoomIn", можна одержати перелік основних станцій в убутному порядку, що використовують цей протокол. Перелік всіх протоколів і їхніх станцій, що використовують, можна роздрукувати або викликати на екран.

### **Основні відправники (Top Senders)**

Функція дозволяє відслідковувати найбільш активні передавальні вузли локальної мережі. LANMeter можна настроїти на фільтрацію по єдиній адресі й виявити список основних відправників кадрів для даної станції. Дані відбиваються на дисплеї у вигляді кругової діаграми разом з переліком основних відправників кадрів. Можливий вивід на печатку або дисплей переліку всіх випробувань станцій, що віщали в плин періоду.

### **Основні одержувачі (Top Receivers)**

Функція дозволяє стежити за найбільш активними вузлами-одержувачами мережі. Інформація відображається у вигляді, аналогічному наведеному вище.

### **Основні генератори широкомовного трафіку (Top Broadcasters)**

Функція може використовуватися для ідентифікації неправильно зконфігурованих станцій. LANMeter аналізує й виявляє розходження між широкомовними, груповими (multi-cast) і унікальними (non-broadcast) адресами кадрів у мережі Ethernet.

Список всіх станцій, що передають широкомовні кадри, може бути роздрукований або виведений на дисплей.

## Генерування трафіку (TrafficGeneration)

LANMeter створює трафік для перевірки компонентів мережі при роботі з підвищеним навантаженням. Додатковий трафік виявляє проблеми, зв'язані із середовищем і іншими проблемами на фізичному рівні. Трафік може генеруватися в мережах Ethernet паралельно з активізованими функціями NetworkStatistics, ErrorStatistics і CollusionAnalysis.

Користувач може задати параметри генеруємого трафіку, такі як інтенсивність і розмір кадрів. Для тестування мостів і маршрутизаторів LANMeter автоматично створює заголовки пакетів IP і IPX: всі що потрібно від оператора – це внести адресу джерела й одержувача.

У ході випробувань користувач може збільшити на ходу розмір і частоту проходження кадрів за допомогою клавіш керування курсором. Це особливо коштовно при пошуку джерела проблем продуктивності мережі й умов виникнення відмов.

LANMeter забезпечує аналіз протоколів:

- TCP/IP;
- Novell NetWare;
- NetBIOS мережах: MS Windows, IBM LAN Server;
- BanyanVINES.

### Засоби аналізу протоколів стека Novell Open Enterprise Server

Ця частина тестів проводить діагностику мереж з архітектурою Novell, перевіряючи правильність підключення клієнтів і сервера через IPX-маршрутизатори, а також реєструючи інформацію про основні джерела трафіку в мережі.

### Список серверів

Відбиває перелік серверів, доступних конкретному учасникові мережі. Ця можливість подібна функції DisplayServers, убудованої в NovellServers. На результуючому екрані відображаються привласнене ім'я, мережна адреса IPX і час відгуку.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

### **Опитування учасників мережі (NetWarePing)**

Дозволяє перевірити правильність підключення клієнта або сервера мережі Novell. Привласнене адресне ім'я цільової станції може бути обране з переліку станцій або уведено в шістнадцятирозрядному вирахованні. На дисплей виводяться отримані IPX адреса, MAC-адреса, число й час відгуку.

### **Пакетна статистика (PacketStatistics)**

Ця функція заснована на зборі статистичних даних для оцінки загального стану трафіку NetWare у локальному сегменті. Безперервне спостереження за трафіком включає реєстрацію пакетів затримки (Delays), маршрутизований трафік (Routed) і пакетний (Burstmode) трафік. Мережні станції, що генерують основний трафік у кожній категорії, можуть бути ідентифіковані за допомогою функції "Zooming" у бажаній категорії.

### **Аналіз маршрутизації (RoutingAnalysis)**

LANMeter може допомогти збалансувати маршрутизацію IPX трафіку. На дисплей виводиться інформація з категорій трафіку, з розпізнаванням адрес мережі призначення й мережі-джерела:

- локальний – локальний;
- локальний – віддалений;
- віддалений – локальний.

### **Основні IPX відправники (Top IPXSenders)**

Функція допомагає відслідковувати вузли, що найбільше інтенсивно передають IPX-пакети. LANMeter може бути зконфігурован таким чином, щоб спостерігати за десятьма основними відправниками інформації конкретної мережної станції. Дані виводяться на дисплей у вигляді кругової діаграми разом з переліком основних відправників, розташованих у порядку убывання. Адреса станції може бути зазначений в одному із трьох форматів – привласнене ім'я, MAC-адреса або IPX-мережний номер.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

## **Основні IPX одержувачі (TopIPXReceivers)**

Функція дозволяє вести безперервне спостереження за найбільш завантаженими вузлами-одержувачами на локальному сегменті. Інформація виводиться на дисплей у вигляді, аналогічному описаному вище.

## **Засоби аналізу протоколів стекаTCP/IP**

Ця група тестів забезпечує діагностику причин відмов у мережах TCP/IP, перевіряючи правильність з'єднань і безупинно відслідковуючи рівень трафіку й ICMP-активність.

## **Моніторинг ICMP (ICPM Monitor)**

Функція дозволяє визначити перевантажені пристрої й неправильно зконфігуровані маршрутизатори й центральні EOM шляхом моніторингу ключових ICMP-пакетів. У цьому режимі LANMeter відслідковує й виводить на дисплей інформацію про ряд основних IP- подій. До числа спостережуваних відносяться кадри: адресат недосяжний (DestinationUnreachable), переадресування (Redirect), обривши джерела повідомлення SourceQuench), перевищення часу (TimeExceeded) і відхилення параметра (Parameter Problem). Мережні станції генеруючі основний трафік у кожній категорії, можуть бути ідентифіковані за допомогою функції "Zooming" у бажаній категорії.

## **ICMPping**

Функція перевіряє наявність зв'язку з конкретної IP-станцією. Символьне ім'я, або IP-адреса станції, може бути обране з переліку або уведено із клавіатури у вигляді десятиричної розділеної крапками запису. На дисплей виводиться інформація, що містить IP-адресу, MAC-адрес, число відгуків і час відгуку.

## **Основні IP-відправники (Top IPSenders)**

Функція допомагає відслідковувати вузли, що найбільше інтенсивно передають IP-пакети в даному локальному сегменті. LANMeter може бути зконфігурован таким чином, щоб відфільтрувати інформацію з однієї адреси й показати основних відправників інформації для конкретної мережної станції. Дані виводяться на дисплей у вигляді кругової діаграми разом з переліком

основних відправників, розташованим у порядку убубання інтенсивностей генеруємих ними трафіків. Адреса станції відображається у вигляді привласненого ім'я або IP-адреси.

Список всіх станцій, що заявили про себе як про відправників протягом періоду випробувань, може бути виведений на дисплей або роздрукований.

### **Основні IP-одержувачі (Top IPReceivers)**

Функція дозволяє вести безперервне спостереження за найбільш завантаженими вузлами – одержувачами в локальному сегменті. Інформація виводиться на дисплей у вигляді, аналогічному описаному вище.

### **Функція відстеження маршруту (TraceRoute)**

Функція TraceRoute повідомляє про кожний маршрутизатор, зустрінутому при посилці IP-пакета на конкретний вузол. Якщо сервер DNS доступний, то в цьому випадку LANMeter відшукує ім'я кожного зустрінутого пристрою. Залежно від конфігурації тестуємої мережі функція TraceRoute покаже випадки використання множинних маршрутів, маршрут, по якому треба пакет при досягненні їм мети, і інформацію про останній маршрутизатор, що відправив пакет, якщо він не досяг мети.

### **Додаткові функції аналізу стека TCP/IP**

Fluke значно поліпшив підтримку мереж TCP/IP. Протокол TCP/IP швидко став найбільш важливим мережним протоколом для багатьох корпоративних і глобальних мереж. У минулому мережі з подібним протоколом підтримувалися лише невеликою групою експертів. Сьогодні багато організацій створюють спеціальні підрозділи для підтримки й керування швидко, що розширюються мережами, TCP/IP. EnterpriseLANMeter має убудовані функції, що дозволяють одержати ключову інформацію про конфігурацію, необхідної для діагностики й усунення несправностей в IP мережах. Використовуючи протокол SNMP, EnterpriseLANMeter може одержувати інформацію від інтелектуальних пристроїв у локальних і віддалених сегментах мережі, а також аналізувати робочі характеристики ланок корпоративних мереж.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

## **Автоматична конфігурація IP (IPAutoConfiguration)**

Функція дозволяє автоматизувати IP конфігурування EnterpriseLANMeter. Вона здійснює пошук IP-адреси, призначеного LANMeter, правильної мережної маски (subnetmask), IP-адреси маршрутизатора за замовчуванням і IP-адреси будь-якого доступного DNS сервера. Функція може використовуватися в режимі Assisted, а також використовувати сервери протоколу динамічного конфігурування DHCP або протоколу завантаження.

## **Дослідження сегментів (SegmentDiscovery)**

Прилад аналізує найбільш важливі параметри IP-мережі, до якої він підключений. Автоматично визначається наявність таких проблем, як некоректна мережна маска, дублювання IP-адрес або відсутність у мережі заявлених серверів. Функція SegmentDiscovery здійснює також пошук у мережі IP-маршрутизаторів, DHCP- і BOOTP-серверів, DNS-серверів, агентів SNMP (SNMPagents) і хостів.

## **Сканування хостів (ScanHost)**

Функція дозволяє перевіряти конфігурацію комп'ютерів мережі, повідомляючи MAC-адресу й мережну маску комп'ютера, а також IP-адреса використовуваного за замовчуванням маршрутизатора. Якщо сервер DNS сконфігурован, то повідомляється також ім'я хосту.

## **Комплект засобів перевірки InternetToolkit**

Цей комплект використовує протокол SNMP для одержання інформації про стан вузлів мережі й дозволяє:

- Збирати інформацію з інтелектуальних пристроїв завдяки функції опитування змінних системної групи MIB – SystemGroupQuery.
- Перевіряти таблиці маршрутів, використовуючи функцію запити таблиць – Route TableQuery.
- Перевіряти стан інтерфейсу за допомогою опитування змінні групи InterfaceStatistics.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

– Виявляти проблеми, використовуючи інформацію про помилки інтерфейсу – InterfaceError.

– Перевіряти дані з DNS-сервера, запитуючи в нього IP-адресу або ім'я.

– Перевіряти можливість доступу до хосту за допомогою функції PingTests і одержувати інформацію з віддалених сегментів мережі завдяки функції RMONStatisticsStudies.

### **Засоби аналізу протоколу NetBIOS**

У цей час спостерігається постійний ріст числа мереж, що використовують протокол NetBIOS, оскільки все більше застосування знаходять операційні системи MicrosoftWindowsNT, Windows 95 і IBMOS/2 Warp. EnterpriseLANMeter діагностує основні причини відмов у мережах NetBIOS: неправильно сконфігуровані вузли, дублювання імен NetBIOS, надмірно високий рівень трафіку NetBIOS.

### **Автоматичне конфігурування стека IP (IP AutoConfiguration)**

Функція призначена для мереж, у яких використовується протокол NetBIOSoverTCP/IP (NBT). Вона здійснює автоматичне IP-конфігурування EnterpriseLANMeter.

### **Дослідження сегмента (SegmentDiscovery)**

Дозволяє аналізувати найбільш важливі параметри сегмента NetBIOS, до якого підключений прилад. Автоматично виявляє проблеми типу дублювання імен, помилок реєстрації.

### **NetBIOSPing**

Дозволяє перевірити досяжність вузла по ім'ю. Виклик NetBEUI є локальним стосовно сегмента підключення й не може трасуватися. Виклик NetBIOS поверх IP або IPX/SPX може перетинати IP і IPX маршрутизатори.

### **Основні відправники пакетів NetBIOS (Top NetBIOSSenders)**

Функція здійснює спостереження за найбільш активними передавальними вузлами NetBIOS у локальному сегменті. EnterpriseLANMeter може бути

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

зконфігурований для фільтрації інформації з певної адреси й індикації основних відправників, що звертаються до конкретної станції мережі.

### **Основні одержувачі пакетів NetBIOS (TopNetBIOSReceivers)**

Функція відслідковує вузли, найбільш завантаженим прийомом пакетів NetBIOS у локальному сегменті.

### **Функції перевірки апаратури й кабелів**

LANMeter сполучить найбільше часто використовувани на практиці функції аналізаторів мережних протоколів і кабельних сканерів з рядом зовсім нових можливостей тестування:

### **Комплексна автоматична перевірка (Expert-TAutotest)**

Цей комплексний тест дозволяє послідовно підключити LANMeter між кінцевим вузлом мережі й концентратором. Цей тест дозволяє автоматично визначити місцезнаходження джерела несправності – кабель, концентратор, мережний адаптер або програмне забезпечення станції.

### **Автоматична перевірка мережних інтерфейсних плат (NICAutotest)**

Перевіряє правильність функціонування знову встановлених або "підозрілих" мережних інтерфейсних карт. Для мереж Ethernet за підсумками перевірки повідомляються: MAC-адреса, рівень напруги сигналів (а також присутність і полярність тестових імпульсів на 10 BASE-T). Якщо сигнал не виявлений на інтерфейсній карті, то тест автоматично сканує сполучне рознімання й кабель для їхньої діагностики.

### **Сканування кабелю**

Функція дозволяє вимірювати довжину кабелю, відстань до самого серйозного дефекту й розподіл імпедансу по довжині кабелю. При перевірці неекранованої кручений пари можуть бути виявлені наступні помилки: розщеплена пара, обриви, коротке замикання й інші види порушення з'єднання.

Для мереж Ethernet на коаксіальному кабелі ці перевірки можуть бути здійснені на працюючій мережі.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

## **Функція визначення розподілу кабельних жил – Wire Map**

Здійснює перевірку правильності приєднання жив, наявність проміжних розривів і перемичок на кручених парах. На дисплей виводиться перелік зв'язаних між собою контактних груп.

## **Функція визначення типу кабелю**

Використовується для складання карти основних кабелів і кабелів, що відгалужуються від центрального приміщення.

## **Автоматична перевірка кабелю – CableAutotest**

Залежно від конфігурації можливо визначити довжину, імпеданс, схему підключення жив, загасання й параметр NEXT на частоті до 100 МГц. Різні індивідуальні тести кабелів виконуються залежно від конфігурації мережі.

Автоматична перевірка виконується для:

- коаксіальних кабелів;
- екранованої кручений пари з імпедансом 150 Ом;
- неекранованої кручений пари з опором 100 Ом.

## **Цілісність ланцюга при перевірці постійним струмом (DCContinuity)**

Ця функція використовується при перевірці коаксіальних кабелів для верифікації правильності використовуваних термінаторів і їхньої установки.

## **Визначення номінальної швидкості поширення NVP (NominalVelocityofPropagation)**

Функція обчислює номінальну швидкість поширення по кабелі відомої довжини й додатково зберігає отримані результати у файл для обумовленому користувачем типу кабелю (UserDefinedcabletype) або стандартного кабелю.

## **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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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++.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

– 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 4к 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 проектах. 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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

підвищеною швидкодією. 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.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

### **Розширена підтримка бібліотек C++**

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCl, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

### **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

### **Підвищення якості й швидкодії інструментів**

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Stake.
- Велика кількість виправлень для підвищення стабільності і якості.

– Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

### **Змінені стилі VCL для High DPI**

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

### **Нові High DPI стилі й стилізація окремих VCL компонент**

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

### **Поліпшена кроссплатформеність**

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

- 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 випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи дотримання відповідності SLA для сервісів IaaS.

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

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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Із чого починається контроль орендованих послуг зв'язку? Багато хто вважає, що із придбання й розгортання на мережі інструментів для організації такого контролю, але насправді – із придбання самої послуги зв'язку, а саме з формування вимог до її якості й продуктивності.

Діяльність сучасної компанії найчастіше виявляється недостатньо ефективною через низьку продуктивність або неприступність корпоративних ІТ-сервісів, оскільки своєчасне виконання важливих завдань різними співробітниками й підрозділами компанії виявляється утруднене або зовсім неможливо. Тому швидка локалізація й усунення несправностей стають найбільш важливими для ІТ-відділу показниками КРІ.

Проблеми доступу до корпоративних сервісів можуть підстерігати на різних рівнях ланцюжка надання послуги (канал зв'язку в офісі, доступність центра обробки даних, помилки в самому додатку). Однак контроль за станом каналу зв'язку як базової послуги доступу до ресурсів має першорядне значення. Він дозволяє чітко розмежувати відповідальність і виявити, чи винуватий у тому, що відбулося оператор зв'язку або неполадки виникають через особливості роботи корпоративного додатка. У результаті значно скорочується час локалізації проблем, підвищується продуктивність корпоративних додатків, забезпечується гарантована якість сеансів відео-конференц-зв'язку й передачі голосового трафіку.

Із чого починається контроль орендованих послуг зв'язку? Багато хто вважають, що із придбання й розгортання на мережі інструментів для організації такого контролю, але насправді – із придбання самої послуги зв'язку, а саме з формування вимог до її якості й продуктивності.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Від правильності складання вимог до якості послуги залежить багато чого: її вартість, рівень компенсацій за порушення показників якості, точність і релевантність вимірів і визначення зон відповідальності. Набір вимог для конкретної послуги може бути унікальним, тому що залежить від тих користувальницьких і корпоративних сервісів верхнього рівня, для яких вона буде транспортом. Наприклад, при забезпеченні роботи складу пред'являються одні вимоги, а у випадку центрального операційного офісу банку – інші.

Коли користувача не задовольняє якість послуги, у першу чергу встає питання про наявність відповідних законодавчих вимог. У цілому вони носять інформативний характер і задають нижню границю рівня обслуговування (мінімально припустима якість надаваних послуг). Однак на практиці виконання цих норм не гарантує очікуваної якості послуги.

Під час відсутності нормативної бази врегулювання взаємин оператора й клієнта, нормування показників якості й реалізація політики в області контролю й забезпечення якості можуть вироблятися тільки на основі договору про рівень обслуговування Service Level Agreement (SLA) між постачальником послуги й споживачем. Договір SLA не є нормативним документом, що однозначно визначає рівень якості виконання послуги конкретного виду, тому що цей рівень визначається суб'єктивним сприйняттям сервісу клієнтом.

Структура типової угоди SLA докладно описана в рекомендації МСЄ.Т М.3342 (07/06) – Guidelines for the definition of SLA representation templates – і повинна містити:

- загальну інформацію про бізнес і опис ділових процедур, що відносяться до послуги, що надає постачальник послуги (ділова частина);
- докладну інформацію про надавану послугу, де конкретизується зміст договірної послуги й погоджений рівень її надання (частина, що ставиться до послуги);
- детальний перелік параметрів якості обслуговування, найважливіших характеристик і деякої інфраструктури технічної підтримки (технічна частина);

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

– дані, що входять у звіт про якість обслуговування, на підставі якого клієнт і постачальник оцінюють рівень виконання послуги, закріплений в SLA (опис звіту SLA).

Згідно із цими рекомендаціями, вимоги до якості послуги є обов'язковим додатком до договору про надання послуг і являють собою технічну частину угоди про рівень обслуговування (SLA).

Висування вимог до якості послуги оператора зв'язку повинне підкріплюватися трьома ключовими пунктами договору:

- перелік показників якості й нормованих значень;
- методика й засоби контролю показників;
- штрафні санкції за порушення показників якості.

### **Перелік показників якості послуги**

Для кожної послуги зв'язку повинні бути визначені ключові показники якості. Це обов'язкова й головна умова вимог до якості орендованої послуги. Показники повинні бути об'єктивні, вимірювані й обчислювальні, і крім того, необхідно визначити їхні граничні значення.

Перелік показників якості для каналів зв'язку представлений у рекомендації Міжнародного союзу електрозв'язку Y.1540 «Internet protocol data communication service – IP packet transfer and availability performance parameters». У даній рекомендації виділяються наступні основні показники продуктивності IP-З'єднання:

- коефіцієнт втрати пакетів (IP Packet Loss Ratio, IPLR);
- затримка передачі пакетів по мережі (IP Packet Transfer Delay, IPTD);
- зміна (варіація) затримки пакетів (IP Packet Delay Variation, IPDV, Jitter);
- доступна пропускна здатність (throughput).

Це вимірювані показники – метрики (KPI), звичайно одержувані за допомогою вимірювального встаткування. На їхній підставі розраховуються показники якості – Key Quality Indicators (KQI), які й визначають властиво якість послуги зв'язку, а аж ніяк не метрики, які оператори найчастіше намагаються продати в договорах на послуги зв'язку.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Для визначення підсумкової якості обслуговування, відповідно до стандарту TM Forum GB917 – SLA Management Handbook v3.2, за допомогою вимірюваних метрик розраховуються підсумкові показники якості послуги:

- коефіцієнт готовності послуги (Service Availability, SA);
- час, протягом якого метрики не відповідали граничним значенням (Service Degradation Period);
- середній час між відмовами послуги (MTBF);
- середній час відновлення послуги (MTTR).

Найчастіше при висновку SLA оператори зв'язку йдуть на хитрість, указуючи в договорі граничні значення для показників якості за весь звітний період – наприклад, середнє значення часу затримки передачі пакета за місяць 100 мс, при середній затримці передачі пакета по даному напрямку 20 мс.

Для наочності розглянемо наступну ситуацію. На деякому напрямку використовуються чутливі до затримки передачі пакетів корпоративні додатки, які вважаються непрацездатними при затримках більше 300 мс. Протягом стандартного восьмигодинного робочого дня затримка регулярно перевищує припустиме граничне значення, а середнє (за 10-хвилинний оцінний інтервал) становить 375 мс, тоді як в інший час – 20 мс. У результаті за календарний місяць (~720 годин) сервіс буде непрацездатним більше 140 годин, тобто понад 87,5% робітника часу. При цьому середня затримка за місяць при 10-хвилинному оцінному інтервалі складе  $(1400 \cdot 375 + 5800 \cdot 20) / 7200 = 89$  мс і формальна угода SLA не буде порушено.

При подібному підході виявляється практично безглуздим одне з найважливіших понять SLA – деградація якості послуги. Деградацією вважається такий стан послуги, коли значення параметрів якості перебувають за межами граничних значень, але умови відмови ще не наступають.

Відмовою же послуги вважається повна або часткова непрацездатність протягом контрольного строку – наприклад, за півгодини губиться більше 50% пакетів. Саме із цих інтервалів складається загальний час неготовності послуги й формується значення підсумкової доступності за звітний період.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Щоб уникнути подібних ситуацій західні постачальники послуг указують, скільки часу (у відсотковому відношенні) показники якості повинні перебувати в межах нормальних значень. У випадку порушення цієї вимоги призначаються штрафні санкції.

Одним із ключових моментів для визначення якості послуги зв'язки є правила розрахунку параметра готовності послуги як основного показника виконання зобов'язань оператором зв'язку. При неправильному підході може скластися ситуація, аналогічна зазначеній вище (із середнім показником параметра за звітний період), коли послуга, якою неможливо користуватися, вважається наданою в повному обсязі, тобто без порушення SLA.

У вимогах до якості послуги повинні враховуватися й планово-профілактичні заходи. Їхня тривалість, як і кількість, повинне бути строго регламентовано. Крім того, варто вказати й припустиме число відмов.

Нерідко замовники відносять запитання про способи визначення граничних значень параметрів якості каналів зв'язку. З переходом від технології передачі даних TDM до технології Ethernet, через величезну розмаїтість пакетних сервісів, безлічі сімейств протокольних стеків пакетної комутації й незліченного числа виробників устаткування, норми якості на канали зв'язку й методологія їхнього виміру перестали відповідати єдиним стандартам.

Як показують результати досліджень, у ході яких вивчався вплив якості каналів на функціонування корпоративного відео-конференц-зв'язку (ВКЗ), для досягнення найкращих показників необхідно контролювати рівень втрат пакетів. Оскільки системи відео-конференц-зв'язку мають великий буфер обміну, затримка і її варіація (jitter) фактично не роблять впливу на якість ВКЗ, як і затримка пакетів. Разом з тим при затримці пакетів у буфері довше граничного значення пакети вважаються загубленими, що теж погіршує якість послуги. Інакше кажучи, навіть у випадку невиконання вимог до затримки якість відеозв'язку може сприйматися користувачем як прийнятне. І навпаки, для інших сервісів, наприклад сервісів з гарантованою доставкою даних, втрата пакетів має менше значення, чим затримка і її варіація.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

У перших версіях вимог до послуги зв'язку важливий сам факт наявності показників якості (KQI) і методів їхнього контролю. А в наступних договорах, після нагромадження певного обсягу статистичних даних, норми можуть бути більш жорсткими. Для встановлення в SLA адекватних – стосовно до кожного замовника – показників KPI необхідне проведення випробувань у спеціально обладнаних метрологічних лабораторіях.

### **Методики й засоби контролю якості**

На даний момент відсутні єдині стандарти на методологію виміру показників продуктивності (метрик) зв'язку в пакетних мережах передачі даних. Приміром, для розрахунку показника варіації затримки передачі пакета (Jitter) застосовуються чотири основні методики: MEF 10.2, Y.1731, RFC 3550 і Y.1541. Відсутні і єдині методики розрахунку показників якості послуг зв'язку.

Неможливо сказати, яка з методик точніше або в який результат більше релевантний – еталона тут немає. Однак дуже важливо чітко формалізувати конкретну методику в договорі SLA, тому що при відсутності визначеності результати вимірів, виконаних клієнтом, буде практично неможливо корелювати з результатами вимірів, проведених оператором із застосуванням іншої методики.

Відповідно, вибір і узгодження конкретної методики виміру метрик є однією з головних завдань складання SLA і формулювання вимог до якості послуги. Ідеально, якщо оператор запропонує використовувати сертифіковану й найпоширенішу (яка є стандартом де факто) систему моніторингу й вимірів.

Для кожної контрольованої метрики повинен бути зазначений інтервал усереднення показників продуктивності послуги. Щоб забезпечити коректне порівняння результатів, рекомендація Y.1541: Network performance objectives for IP-based services пропонує робити його не занадто коротким – на практиці близько 5 хв.

Вимір метрик повинне здійснюватися в безперервному «наскрізному» режимі моніторингу якості клієнтських послуг (end-to-end, «від порту до порту»). При такому підході тестовий трафік підмішується в активне з'єднання без шкоди

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56



## Знижки за порушення показників якості

Відповідно до рекомендацій M.3342 (07/06) – Guidelines for the definition of SLA representation templates – і стандарту TM Forum GB917 – SLA Management Handbook, обов'язковим пунктом SLA є визначення правил розрахунку компенсацій у випадку порушення гарантованого рівня якості послуги.

В Україні в основному застосовується простий лінійний підхід до розрахунку компенсацій за порушення гарантованої якості, при якому за кожну повну годину неприступності послуги з підсумкового рахунку віднімається 1/720-а частина щомісячної абонентської плати.

Даний підхід відрізняється простотою й прозорістю, однак він не завжди відбиває інтереси клієнта, тому що відмова на 15 хвилин і на 5 годин представники бізнесу сприймають по-різному. Спеціально для таких випадків ряд західних операторів зв'язку застосовують модель прогресивної знижки, коли рівень компенсації зростає зі збільшенням тривалості аварії.

Нерідко для особливо важливих для діяльності підприємства крапок в SLA передбачають додаткові санкції, якщо послуга перебуває в деградованому стані.

Широке поширення SLA дало новий імпульс бізнесу операторів у корпоративному й державному сегментах. Завдяки можливості гарантувати рівень обслуговування й моніторингу якості надаваної послуги, вони одержали шанс додати ще 10-30% до тієї ціни, по якій вони продають свої звичайні послуги VPN. Однак через недостатню поінформованість клієнтів підписані угоди SLA не приводять до реального підвищення якості послуги й не дозволяють вимагати компенсації за невиконання зобов'язань. Тому до висновку таких угод доцільно залучати незалежних експертів, які допоможуть підготувати необхідні вимоги й погодити фінальний текст документа.

Більше того, перед публікацією вимог доцільно провести аудит і визначити дійсні значення показників, які зараз пропонує оператор (для існуючих послуг). Такий аудит не займає багато часу, але дозволяє істотно розширити знання про рівень якості вже надаваних послуг і правильно визначити границі припустимих значень для нових. Аудит варто доручати компаніям, що володіють

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

необхідною експертизою, і виконувати його із застосуванням сертифікованих вимірювальних засобів. В ідеалі результати цього аудита повинні прийматися операторами зв'язку.

### 3.2 Розробка структурної схеми

Очевидно, що при росту навантаження на IaaS-платформу провайдеру можлива деградація рівня продуктивності віртуальної машини. Постачальники послуг усіляко прагнуть не допустити такого розвитку подій. У цьому солідарні всі компанії. Однак деякі включають параметри продуктивності в SLA, а інші вважають подібну міру непотрібною.

Ми не спостерігаємо деградації [продуктивності] навіть при росту навантаження, тому що вчасно робимо розширення й модернізацію потужностей дата-центрів. Окремо в SLA дані параметри (продуктивність VM і СЗД) не відбиті, оскільки їхнє дотримання є нашим першорядним обов'язком, незалежно від звернень клієнтів. Фахівці здійснюють постійний моніторинг всіх основних параметрів орендованих інфраструктурних потужностей, що дозволяє їм оперативно одержувати інформацію про потенційні проблеми й вчасно їх прогнозувати.

Якщо клієнт вимагає, щоб для його віртуальних машин гарантовано виділялися визначені обчислювальні ресурси, ми застосовуємо стандартні засоби сучасних платформ віртуалізації, які при виникненні конкуренції за ресурси дозволяють перемістити віртуальні машини на інші сервери.

Чимало й компаній, які думають, що внесення в SLA характеристик продуктивності доцільно. Самим вузьким місцем віртуалізованого ІТ-середовища багато експертів вважають продуктивність системи зберігання, тому більшість постачальників приділяють найбільше пильну увагу таким характеристикам СЗД, як кількість операцій уведення-виводу в секунду (IOPS) і час доступу до диска (latency).

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Значення IOPS вимірюються системою моніторингу, а час доступу до диска – штатними засобами ПЗ віртуалізації. У випадку виникнення проблем з віртуальною машиною чергова зміна й інженери групи віртуалізації одержують відповідне попередження. Крім того, забезпечується моніторинг різних параметрів на рівні операційної системи й запущених у ній сервісів. Якщо клієнт користується сервісом компанії по адмініструванню ОС і сервісів, такий моніторинг здійснюється за замовчуванням.

Для недопущення деградації продуктивності віртуальних машин фахівці застосовують комплекс мір. Так, для кластера використовується механізм Distributed Resource Scheduler (DRS), що відслідковує завантаження фізичних серверів по основних параметрах, – у випадку досягнення певного навантаження на сервер частина віртуальних машин автоматично переміщається на іншій. У кластері підтримується надмірність серверів з таким розрахунком, щоб навантаження на весь кластер становили не більше 70%. У рамках укладених сервісних контрактів з постачальниками встаткування ресурсні потужності кластерів можна нарощувати по плані-графіку.

Ми гарантуємо забезпечення продуктивності віртуальних машин по нижній границі, не обмежуючи її зверху. Таким чином, якщо на сервері, де розташована віртуальна машина, є вільні обчислювальні ресурси понад гарантований, вони будуть доступний замовник». Що стосується СЗД, то в цей час всі клієнти користуються загальним каналом зв'язку із системами зберігання. Довгий час це не викликало проблем, але зараз, щоб задовольнити зростаючі потреби замовників, компанія переводить хмарні СЗД із дисків Fibre Channel і SATA на флеш-накопичувачі із прямим доступом до них з віртуальних машин через мережу. Паралельно впроваджується ПЗ для забезпечення гарантованої пропускної здатності системи зберігання даних у хмарі. Відповідні зміни в SLA будуть внесені вже цією осінню.

За узгодженням із замовником фіксується в SLA кожного проекту показники продуктивності окремих компонентів хмарної платформи. Крім того, в

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

угоді вказуються способи виміру цих показників і періодичність проведених вимірів. «Написати «гарантується 100 500 OPs на 1 Гбайт дискового простору» може будь-який оператор, але далеко не всі здатні довести, що цей критерій витриманий. Ми за максимально прозорі відносини між оператором хмарної платформи і її споживачем. Продуктивність віртуальних машин і СЗД визначається в SLA показниками IOPS і Latency.

Пікові показники продуктивності регламентуються таким чином, щоб завантаження пропускної здатності уведення-виводу й мережі не перевищувало 80%. Для різних систем важливі різні показники: для Web-сайтів – час відгуку, для розміщення IT-інфраструктур – показники пікового завантаження процесора, пам'яті, віртуальній мережі й т.д. У свій SLA ця компанія включає також параметри гарантованого резервного копіювання, способи й строки надання й зберігання користувальницьких даних («Чесне розставання»).

### **Наскрізнi SLA**

Як би не була висока надійність самої платформи IaaS, розміщеної в відказостійкому ЦОД, вузьким для замовника місцем можуть стати канали доступу до цієї платформи. Гарною новиною є те, що багато хто із провайдерів практикують висновок наскрізних SLA, що охоплюють як сам сервіс IaaS, так і канали доступу. При цьому, по їхньому твердженню, при правильній організації й резервуванні каналів рівень доступності зв'язку виявляється не нижче, ніж у платформи SLA, а тому в наскрізних SLA ця важлива характеристика не знижується.

Втім зниження або збереження рівня доступності залежить від способу організації каналів зв'язку – якщо канал зарезервований, доступність не погіршується. В іншому випадку рівень доступності в наскрізному SLA знижується до рівня доступності каналу.

Ті постачальники послуг IaaS, які є одночасно й операторами зв'язку, використовують цю перевагу.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Постачальники послуг IaaS, що не є операторами зв'язку, активно співробітничать із такими. Умови цих SLA компанія транслює в договорах із клієнтами, які користуються послугами зв'язку. А контроль за дотриманням SLA забезпечують технічні служби ЦОД. Ми вказуємо в наших контрактах ті ж параметри SLA, що й в операторів, – тобто беремо на себе відповідальність за якість їхньої роботи й безперебійне надання каналів зв'язку.

Для надання клієнтам каналів зв'язку практикує використання послуг телекомунікаційних операторів за схемою субпідряду. При такій схемі компанія контролює якість у рамках свого договору з оператором, клієнт же одержує від її комплексну послугу й має справу тільки з одним контрагентом. Рівень доступності такої комплексної послуги не знижується.

Для тих організацій, для яких критична фіксована пропускна здатність каналу зв'язку, компанія реалізує індивідуальне підключення до хмари по окремих каналах з гарантованою пропускною здатністю або навіть по «темній» оптиці. Таке підключення найчастіше оснащується індивідуальними засобами шифрування, у тому числі сертифікованими.

Отже, послуги IaaS пропонуються в Україні досить більшим числом компаній, причому по цілком зрозумілим і документованим (в SLA) правилам. У галузі ще не прийшли до згоди щодо того, чи треба регламентувати в SLA характеристики продуктивності віртуальних IT-інфраструктур, але гарантуємі показники доступності виглядають цілком прийнятними навіть для самих вимогливих корпоративних замовників. До того ж провайдери розуміють потреба замовників у наскрізних SLA і працюють над їхнім удосконалюванням.

На початковому етапі дослідження було виконано структурно-функціональне моделювання розподілу елементів IaaS і інформаційні потоки між ними на різних рівнях.

Реалізуємо програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS на прикладі банківського середовища.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

На першому етапі побудована модель розміщення елементів банківського середовища, що також ураховує взаємозв'язок рівнів ієрархії (центр, філія, відділення) і груп банківських систем (Back office, Middle office, Front office). Результати аналізу моделі відображені на структурній схемі (рисунк 3.1)

На структурній схемі наведені основні блоки IT-інфраструктури як сервіс (IaaS), яка досліджується. Тобто IT-інфраструктури як сервіс (IaaS) з розподіленими ресурсами, ефективність якої оцінюють (SLA). Така система наведена на прикладі філії банку.

Front-office – це комплекс програмно-апаратних засобів, що підвищують ефективність спілкування. Це спеціалізовані IT-інфраструктури як сервіс (IaaS), що автоматизують роботу співробітників, які спілкуються з "зовнішнім миром", які допомагають їм у повсякденній діяльності. Іншою стороною взаємодії учасників «телефонних» відносин є IT-інфраструктури як сервіс (IaaS), що автоматизують внутрішні взаємодії в компанії. Back-office, – це "каркас" підприємства. Для ефективної роботи всієї фірми обидві сторони (Front-office і Back-office) повинні взаємодіяти один з одним по оптимальних алгоритмах.

Front-office відповідає за відносини із клієнтами й ухвалює рішення щодо здійсненні угод, він не повинен мати можливість здійснювати фінансовий облік або аналіз результативності своїх операцій. Middle-office відповідає, зокрема, за моніторинг і аналіз ризику, а також розробку мер активного керування. Back-office здійснює рутинні операції ведення внутрішнього й зовнішнього обліку, розрахунку податків і складання звітності. Незважаючи на необхідність строгого поділу функцій front-office, middle-office і back-office, першочерговою задачею є забезпечення доступу до інформації всім учасникам процесу керування ризиками.

### **Автоматизація Front-офісних операцій**

При автоматизації Front-офісних операцій сучасні IT-інфраструктури як сервіс (IaaS) дозволяють:

– Оптимізувати роботу з різними видами Front-офісних операцій. У системі передбачений готові бізнес процеси обробки й обліку різних операцій,

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

наприклад валютні контракти, включаючи валютні опціони; операції по залученню/розміщенню ресурсів, включаючи короткострокове кредитування/фінансування, застави й так далі.

– Можливість використання відкритих інтерфейсів для автоматичного імпорту операцій з будь-якої зовнішньої ІТ-інфраструктури як сервіс (IaaS) – наявність відкритих інтерфейсів для кожного типу операції дають можливість імпортувати інформацію про угоду з будь-якого зовнішнього джерела, включаючи текстовий файл.

– Можливість починати обробку Front-офісної операції на підставі заявки.

– Можливість використовувати різні методи (документообіг) обробки заявки на операцію. Існує можливість організувати документообіг заявки в системі. Електронний документообіг має на увазі автоматичне формування повідомлень уповноваженим співробітникам, використання різної бізнес логіки.

– Ефективна інтеграція з функціями Middle-офісу(наприклад, установка лімітів) і Back-офісу (організація бухгалтерських проводок) – передбачена можливість інтеграції Front-офісних функцій з функціями Middle-офісу й Back-офісу, наприклад відсутня необхідність повторного введення інформації про угоду при формуванні платіжних документів.

– Формування різних операційних звітів – будь-які необхідні операційні звіти, наприклад, Trade Slips, журнали угод і так далі, можуть автоматично формуватися й роздруковуватися в системі.

– Формування різних аналітичних звітів у розрізі дилерів, портфелів, валюти, угод і так далі – у системі передбачена можливість використання різних екранних форм або звітів для проведення аналізу ведення Front-офісних операцій з використанням різних аналітичних зрізів, наприклад по контрагентах, по дилерах, та інше.

### **Автоматизація Middle-офісних операцій**

На жаль, у більшості українських банків діяльність відділів Middle-офісу дотепер майже не автоматизована. Основна аналітична звітність складається за

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

допомогою не призначених для цього додатків. Відсутність єдиної інтегрованої ІТ-інфраструктури як сервіс (IaaS) для обліку всіх Front-офісних операцій значно обмежує можливості співробітників Middle-офісу по керуванню й аналізу діяльності банку.

Сучасні ІТ-інфраструктури як сервіс (IaaS) дозволяють:

– Оптимізувати внутрибанківські фінансові потоки між центрами виникнення прибутку й витрат.

– Управляти параметризацією всіх банківських операцій, у тому числі існує можливість додавання необмеженого числа аналітичних параметрів по кожній угоді.

– Управляти й здійснювати моніторинг кредитно-інвестиційного портфеля банку в режимі реального часу.

– Становити різні аналітичні звіти для оцінки й прогнозування діяльності банку.

– Становити різні аналітичні звіти для оцінки й прогнозування діяльності банків-конкурентів.

– Проектувати й управляти необмеженим числом лімітів, нормативів, у тому числі існує можливість аналізувати використання лімітів і нормативів у режимі реального часу.

– Переоцінювати відкриті позиції.

– Управляти ризиками: VAR-Аналіз, стратегії хеджування, інші звіти.

### **Автоматизація Back-офісу**

Автоматизація Back-офісу дозволить уникнути рутинного дублювання введення інформації. Використання сучасної ІТ-інфраструктури як сервіс (IaaS) дозволить автоматизувати наступні функції:

– Автоматичне формування бухгалтерських проводок по кожній операції на підставі раніше створених шаблонів.

– Можливість проектувати різні типи документообігу для тверджень операцій.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

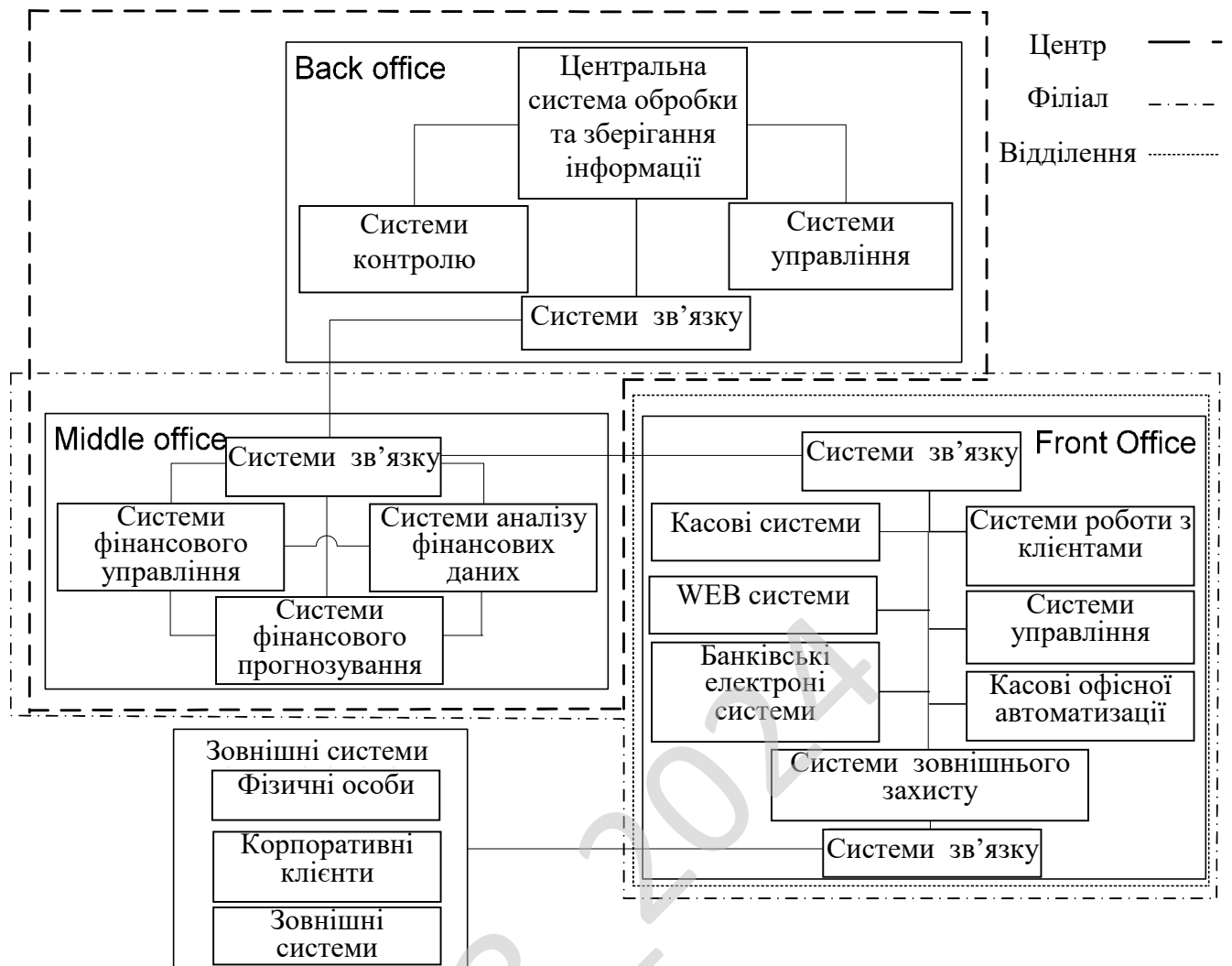


Рисунок 3.1 – Структурна схема системи

- Автоматичне формування розрахункових інструкцій для генерації SWIFT файлів(підтримка множини SWIFT форматів).
- Проектування й формування пакетів документації по кожному типі операції.
- Автоматичне формування підтверджень по кожній угоді, облік отриманих підтверджень від контрагентів.

### 3.3 Розробка функціональної схеми

На другому етапі, для оцінки ефективності мережного трафіку, формованого кожною банківською функцією, побудована модель функціональної взаємодії елементів структури філії банку.

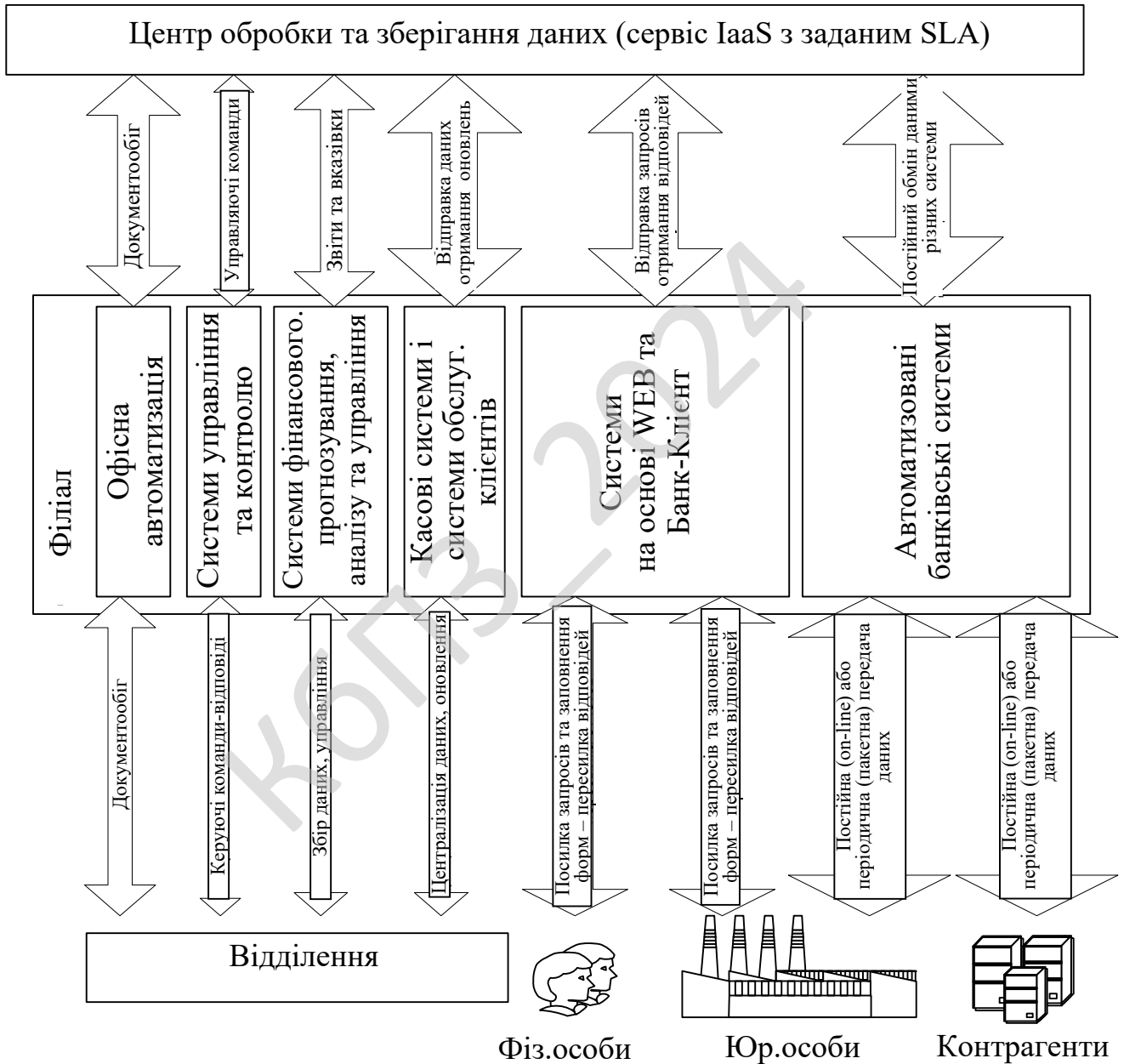


Рисунок 3.2 – Функціональна схема системи

На зовнішній моделі зображені функціональні заявки між блоками, які взаємодіють або з клієнтами, або з іншими відділеннями банків, які є елементами розподіленої оцінюваної мережі.

У проекті оцінюється ефективність зв'язків у мережі між різними блоками мережі. Ця мережа зв'язує центральний офіс з філіалами та відділеннями.

У центральному офісі знаходиться центр обробки та зберігання даних, які надходять по мережі з усіх філій.

У філіях досліджувана комп'ютерна мережа зв'язує наступні функціональні блоки:

- Офісна автоматизація.
- IT-інфраструктури як сервіс (IaaS) управління та контролю.
- IT-інфраструктури як сервіс (IaaS) фінансового прогнозування, аналізу та управління.
- Касові IT-інфраструктури як сервіс (IaaS) і IT-інфраструктури як сервіс (IaaS) обслуговування клієнтів.
- IT-інфраструктури як сервіс (IaaS) на основі WEB та Банк-Клієнт.
- Автоматизовані банківські IT-інфраструктури як сервіс (IaaS).

Ці IT-інфраструктури як сервіс (IaaS) діють у рамках Front-office, Middle-office та back-office, а також працюють з зовнішніми системами.

При цьому ці IT-інфраструктури як сервіс (IaaS) призначені для обігу наступної інформації:

- Документообіг.
- Управляючі команди.
- Звіти та вказівки.
- Відправка даних та отримання оновлень.
- Відправка запитів та отримання відповідей.
- Постійний обмін даними різних систем.

Філіали з відділеннями по мережі передають наступну інформацію:

- Документообіг.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

- Керуючі команди та відповіді.
- Збір даних, управління.
- Централізація даних оновлення.

Філіали при роботі з клієнтами по мережі передають наступну інформацію:

- Посилка запитів та заповнення форм – пересилка відповідей.
- Постійна (on-line) або періодична (пакетна) передача даних.

Як ми бачимо, даних дуже велика кількість, тому оцінка ефективності розподіленої комп'ютерної банківської мережі, та оптимізація, на основі проведеної оцінки, цих систем є дуже актуальною задачею.

При розгляді функціональної моделі внутрішньої взаємодії філій банку, ми застосуємо такий же самий підхід, як й при зовнішній. Тобто розглянемо усі структурні елементи досліджуємої ІТ-інфраструктури як сервіс (IaaS) та функціональні зв'язки між ним. Оцінимо об'єм потоків та види топологій й протоколів передачі даних між логами, й на основі цих досліджень можемо надати рекомендації по оптимізації внутрішньої комп'ютерної мережі.

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

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>69</b>

процесів та потоків даних з метою показати систему що розробляється. Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

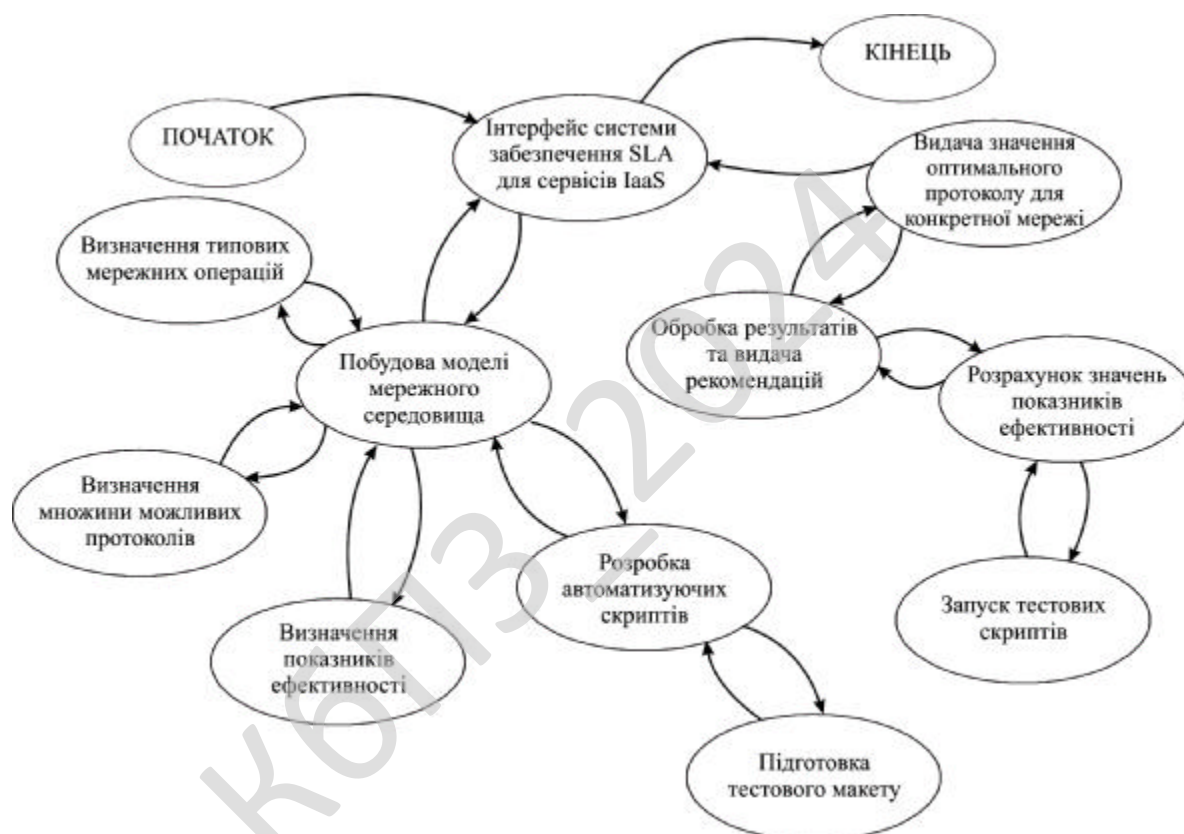


Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис ІТ-інфраструктури як сервіс (IaaS), структурну, функціональну схеми ІТ-інфраструктури як сервіс (IaaS), та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації ІТ-інфраструктури як сервіс (IaaS).

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

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

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

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

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

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

#### Опис алгоритмів функціонування системи

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи дотримання відповідності SLA для сервісів IaaS.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

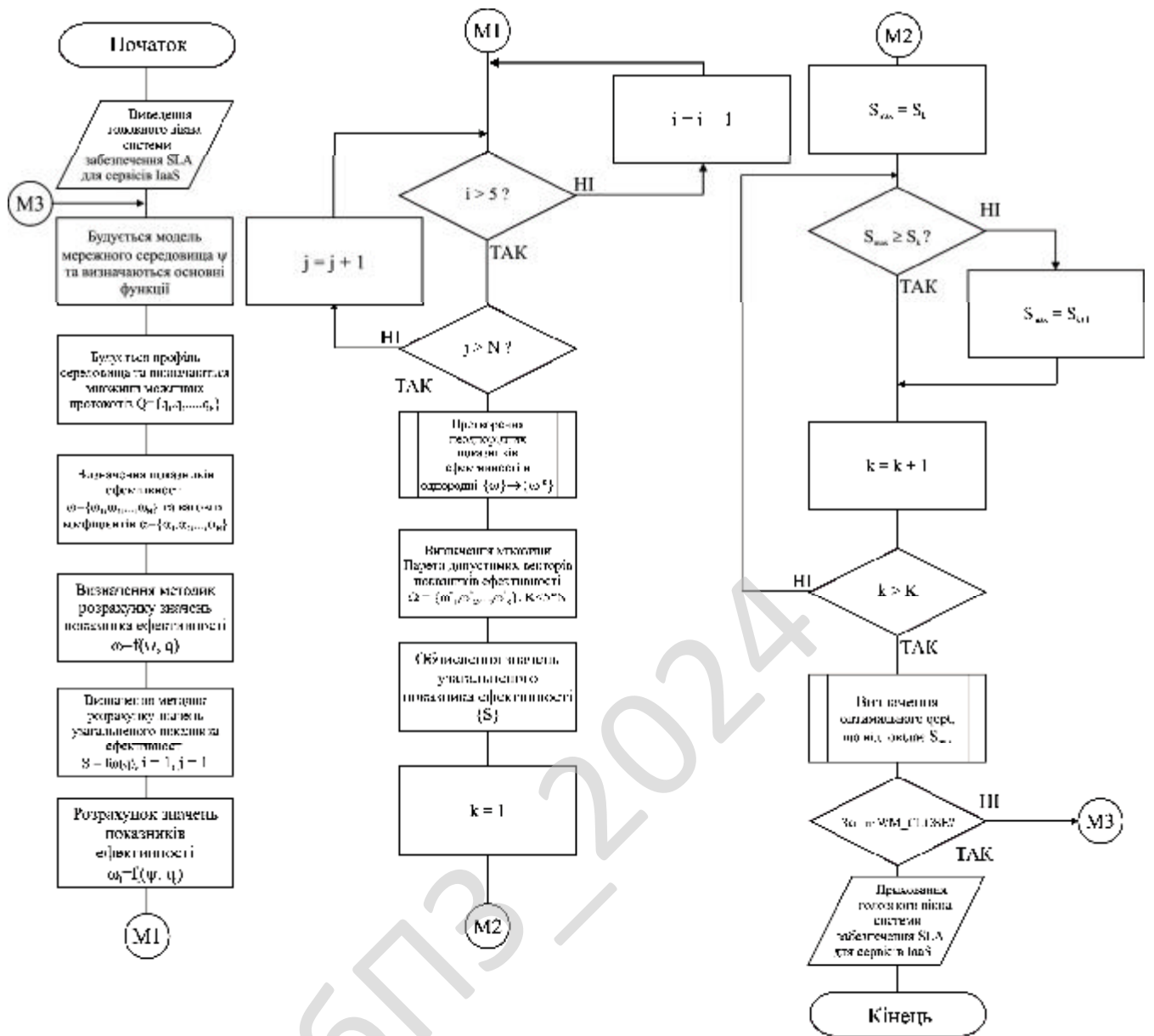


Рисунок 4.1 – Блок-схема основної програми

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

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки

програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, названої UML-моделлю.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

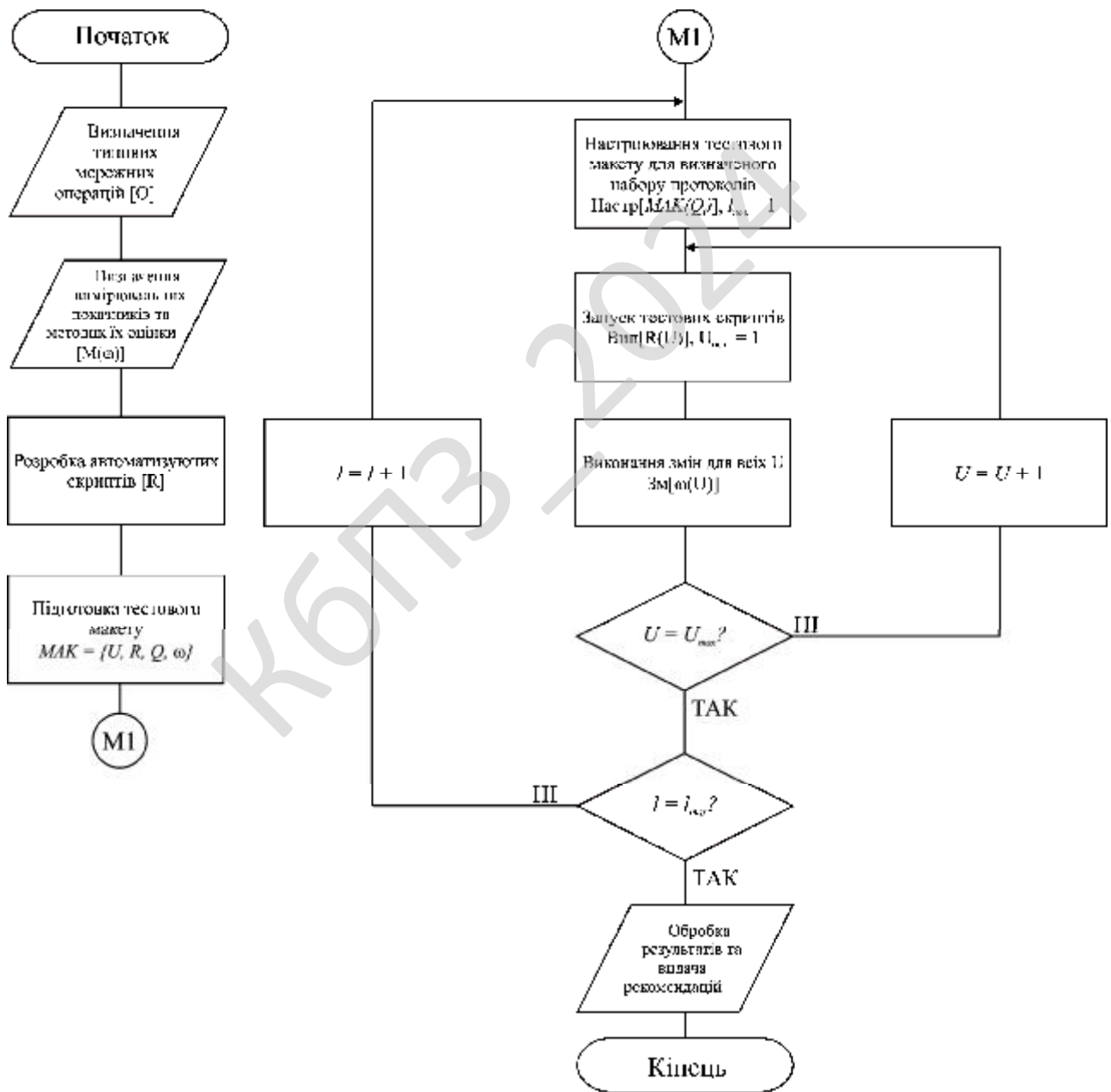


Рисунок 4.2 – Блок-схема роботи підпрограми

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

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

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

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

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

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

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

(прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

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

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

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

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

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

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

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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

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

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

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

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

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

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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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

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

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

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

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

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

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

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

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

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

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

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

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

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

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Нижче наведемо функцію, яка дозволяє побудувати схему мережі.

```
function TMainfm.Scheme(X: Integer) : integer;
begin
  Inc(X);
  //додаємо наступний елемент
  if KolScheme < X then
    begin
      PaintScheme := false;
    //схема не побудована
      if Manager.ElementsCount > Manager.CountVakMest then
        begin
          ShowMessage('Некоректно з'єднані пристрої!');
          Manager.DeleteAllSchemea;
        end;
      end;
    else
      begin
```

```
PaintScheme := true;// схема коректно побудована
end;
Status.Panels[3].Text := IntToStr(Manager.CountVakMest);
Result := X;
end;
```

### Наведемо функцію оцінки ефективності розподіленої мережі.

```
procedure TEfficForm.btnCloseSharesClick(Sender: TObject);
var
    OS:Boolean;
    FLibHandle : THandle;
    Name9x:array [0..12] of Char;
    NameXP:PWChar;
    i:Integer;
    ShareName: String;
begin
    if not IsXP(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 XP
        FLibHandle := LoadLibrary('NETAPI32.DLL');
        if FLibHandle = 0 then Exit;
        @NetShareDelXP := GetProcAddress(FLibHandle, 'NetShareDel');
        if not Assigned(NetShareDelXP) then
//Перевірка
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        i:= SizeOf(WideChar)*256;
        GetMem(NameXP, i);
//Виділяємо пам'ять під змінну
        StringToWideChar(ShareName, NameXP, i);
//Перетворимо в PWideChar
        NetShareDelXP(nil, NameXP, 0);
```

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<b>81</b>

```

//Видаляємо ресурс
    FreeMem (NameXP);
//Звільняємо пам'ять
end else begin //Код для 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDel := GetProcAddress (FLibHandle, 'NetShareDel');
    if not Assigned (NetShareDel) then
//Перевірка
        begin
            FreeLibrary (FLibHandle);
            Exit;
        end;
        FillChar (Name9x, SizeOf (Name9x), #0);
// Очищуємо масив
        move (ShareName [1], Name9x [0], Length (ShareName));
// Заповнюємо масив
        NetShareDel (nil, @Name9x, 0);
// Видаляємо ресурс
        end;
        FreeLibrary (FLibHandle);
end;

```

## 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-1, який заснований на перетвореннях алгоритму SHA-1. SHACAL-1 шифрує 160-бітний блок даних з використанням 512-бітного ключа шифрування. Допускається використання більше коротких ключів шифрування (не коротше 128 біт), які перед виконанням розширення ключа (процедура розширення ключа також успадкована від SHA-1 і буде описана нижче) повинні бути доповнені нульовими бітами для досягнення 512-бітного розміру.

В алгоритмі SHACAL-1 передбачено 80 раундів шифрування. Шифруєме повідомлення представляється у вигляді п'яти 32-бітних субблоків *A*, *B*, *C*, *D* і *E*, над якими в кожному раунді виконуються наступні дії:

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>82</b>



Етап 2. Інші фрагменти розширеного ключа  $K_{16} \dots K_{79}$  обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = (K_{i-3} \oplus K_{i-8} \oplus K_{i-14} \oplus K_{i-16}) \lll 1.$$

Раунди розшифрування виконуються у зворотній послідовності; кожний з них має на увазі виконання наступних операцій:

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

$$B_i = C_{i+1} \lll 2,$$

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

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

$$E_i = K'_i + (B_{i+1} \lll 5)' + f'_i(C_{i+1} \lll 2, D_{i+1}, E_{i+1}) + A_{i+1} + M'_i + 4.$$

Тут запис  $f'(x)$  позначає побітовий комплемент результату виконання операції  $f(x)$ .

КБПЗ - 2024

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання бакалаврської дипломної роботи. Розроблене програмне забезпечення системи дотримання відповідності SLA для сервісів IaaS складається з наступних функціональних блоків:

- Навігаційне меню: Файл; Побудова розподіленої ІС; Довідка.
- Функціональних кнопок ПЗ: Додати; Видалити; Показники ефективності; Запустити тестування; Збереження результатів; Типові мережні операції; Множина можливих протоколів.

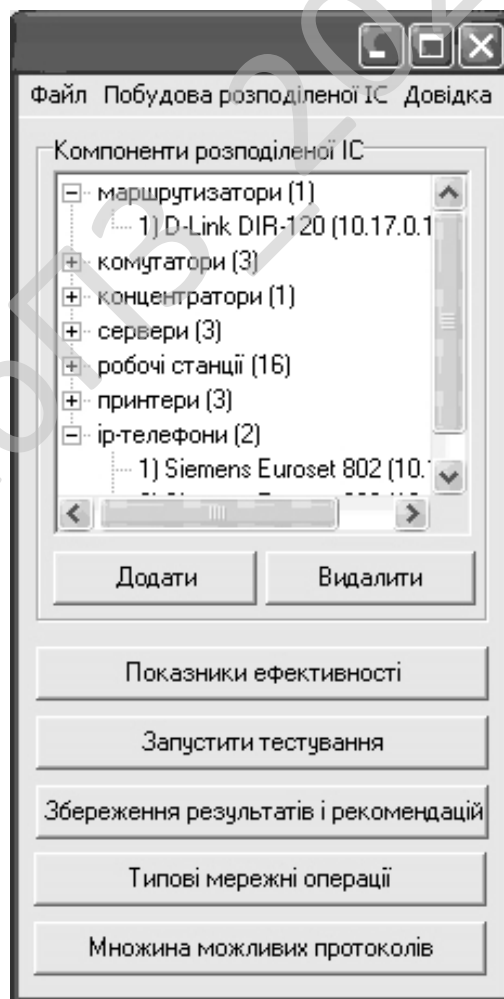


Рисунок 5.1 – Головне вікно розробленого ПЗ

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

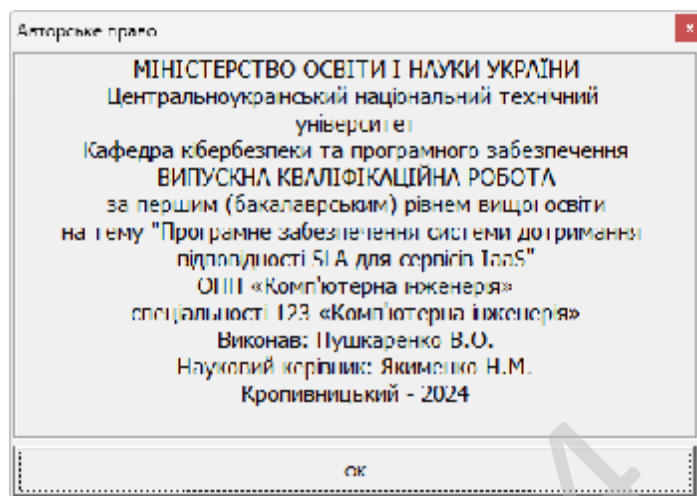


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись. Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

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

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

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів. Проводилось тестування форматом чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе. Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок. Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок. Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

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

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму.

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

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи дотримання відповідності SLA для сервісів IaaS.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем дотримання відповідності SLA для сервісів IaaS.
- Досліджена система дотримання відповідності SLA для сервісів IaaS.
- На основі отриманих результатів досліджень створена програмна реалізація системи дотримання відповідності SLA для сервісів IaaS.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання дотримання відповідності SLA для сервісів IaaS.

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 10.4.1. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи дотримання відповідності SLA для сервісів IaaS. 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

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHACAL-1.

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.

КБПЗ\_2024

					ВКРБ-123.24.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
2. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
3. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
4. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
5. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
6. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
7. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
8. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
9. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.
10. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

12. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.

13. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

14. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

15. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

16. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

18. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

19. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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

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

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

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

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

25. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

26. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobayev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

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

28. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

29. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

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

31. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

32. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». *Проблеми телекомунікацій*. № 1(26). С. 83-96. 2020.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

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

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

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

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

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

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

39. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Електронний Журнал]. Georgia. Tbilisi: SCSA – 2018.

40. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian

Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

41. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

42. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

43. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.

44. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). - Полтава: ПолтНТУ. - 2017. - С. 112-115.

45. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

46. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). - Полтава: ПолтНТУ. - 2016. - С. 98-103.

47. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). - Харків: ХУПС. - 2016. - С.96-100.

48. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". - Випуск 8 (145). - Х.: ХУПС - 2016. - С. 77-80.

49. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". - Випуск 6 (143). - Х.: ХУПС - 2016. - С. 216-220.

50. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод кількісної оцінки ризиків розроблення програмного забезпечення. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). - Харків: ХУПС. - 2016. - С. 128-133.

51. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод якісного аналізу ризиків розроблення програмного забезпечення. Наука і техніка Збройних Сил України. – Випуск 2(23). - Харків: ХУПС. - 2016. - С. 150-158.

52. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Проблеми аналізу та оцінки ризиків інформаційної діяльності. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 40-42.

53. Смірнов О.А., Коваленко А.С., Коваленко О.В., Доренський О.П. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи. Системи озброєння і військова техніка. – Випуск 2(46) – Х.: ХУПС – 2016. – С. 103-107.

					<b>ВКРБ-123.24.0003.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

					<b>ВКРБ-123.24.0003.00.00.ТЗ</b>		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Пушкаренко В.С.				Літ.	Аркуш	Аркушів
Перевірів	Якименко Н.М.			Б			
Н. Контр.	Коваленко А.С.				ЦНТУ КМ-21-3СК		
Затв.	Смірнов О.А.						

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи дотримання відповідності SLA для сервісів IaaS.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 134-02 від 01.04.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи дотримання відповідності SLA для сервісів IaaS.

## 4 Джерела розробки

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

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.24.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи дотримання відповідності SLA для сервісів IaaS;
- 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.

					ВКРБ-123.24.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					<b>ВКРБ-123.24.0003.00.00.ТЗ</b>	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 – 97 аркушів.

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					<b>ВКРБ-123.24.0003.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

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.

## 9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 10.06.2024 р.

					ВКРБ-123.24.0003.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

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

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
першим (бакалаврським) рівнем вищої освіти

\_\_\_\_\_ Якименко Н.М.

*Програмне забезпечення системи дотримання відповідності SLA для  
сервісів IaaS*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2024 року

## Файл fmMain.pas основної програми

```

unit fmMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Connector, Db, Search, Effic, About
  Dialogs, ActnList, Menus, inifiles, StdCtrls, Datamodule, ClassManager, ExtCtrls,
  ComCtrls, ClassInformation, ClassDraw, ToolWin, ClassGrid, ClassGenAlg, ClassTrass;

type
  TMainFm = class(TForm)
    ActionList1: TActionList;
    actExit: TAction;
    actFormReadFile: TAction;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    actAddElement: TAction;
    actAddElement1: TMenuItem;
    Panell1: TPanel;
    Panel2: TPanel;
    Tree: TTreeView;
    PaintBox: TImage;
    actOptionsOpen: TAction;
    N5: TMenuItem;
    N6: TMenuItem;
    GroupBox1: TGroupBox;
    actFormManagerBD: TAction;
    N7: TMenuItem;
    N8: TMenuItem;
    PopupMenu1: TPopupMenu;
    N9: TMenuItem;
    TreeUzel: TTreeView;
    PopupMenu2: TPopupMenu;
    N10: TMenuItem;
    Panel3: TPanel;
    Button1: TButton;
    N11: TMenuItem;
    N12: TMenuItem;
    Status: TStatusBar;
    btRazm: TButton;
    Time: TTimer;
    btRazmCont: TButton;
    Panel4: TPanel;
    Panel5: TPanel;
    Prog: TProgressBar;
    Button2: TButton;
    N13: TMenuItem;
    btKomp: TButton;
    Time2: TTimer;
    btKompCont: TButton;
    Button3: TButton;
    N14: TMenuItem;
    ScrollBox1: TScrollBox;
    N15: TMenuItem;
    Save: TSaveDialog;
    procedure N15Click(Sender: TObject);
    procedure N14Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure btKompContClick(Sender: TObject);
    procedure Time2Timer(Sender: TObject);
    procedure btKompClick(Sender: TObject);
  end;

```

```

procedure N13Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure btRazmContClick(Sender: TObject);
procedure TimeTimer(Sender: TObject);
procedure btRazmClick(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure TreeClick(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure PaintBoxMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure PaintBoxClick(Sender: TObject);
procedure PaintBoxMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure N8Click(Sender: TObject);
procedure actFormManagerBDEExecute(Sender: TObject);
procedure PaintBoxMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure actOptionsOpenExecute(Sender: TObject);
procedure actAddElementExecute(Sender: TObject);
procedure actExitExecute(Sender: TObject);
procedure actFormReadFileExecute(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormCreate(Sender: TObject);
function DrawScheme (X : integer) : integer;
private
  { Private declarations }
public
  { Public declarations }
  PX,PY : integer;
  Index,KolScheme,isKolScheme : integer;
  Info : TInformation;
  PointScheme : Tpoint;
  AlgStart,IsStopAlg : boolean;
  CheckObj,Move,PaintScheme,isPaintScheme : boolean;
  Trass : TTrass;
  procedure CreateChildForm(form: TForm; TForms: TFormClass);
protected
  //Draw : TDraw;
  GenAlg : TGenAlg;
  // Метод у якому необхідно створювати всі об'єкти
  procedure CreateObject;
  // Метод у якому необхідно знищувати всі об'єкти
  procedure FreeObject;
end;

var
  MainFm: TMainFm;
  Draw : TDraw;

implementation

{$R *.dfm}

uses
  FormReadFile,ClassMatr,FormAddElement,ClassOptions,FormOptions,FormMAnagerBD;

function TMainfm.DrawScheme(X: Integer) : integer;
begin
  Inc(X);//додаємо наступний елемент
  if KolScheme < X then
  begin
    PaintScheme := false;//схема не побудована
  end;
end;

```

```

    if Manager.ElementsCount > Manager.CountVakMest then
    begin
        ShowMessage('Некоректно з'єднані пристрої!');
        Manager.DeleteAllSchemea;
    end;
end
else
begin
    PaintScheme := true;// схема коректно побудована
end;
Status.Panels[3].Text := IntToStr(Manager.CountVakMest);
Result := X;
end;

procedure TMainfm.CreateChildForm(form: TForm; TFormClass: TFormClass);
begin
    form:=TForms.Create(Application);
    form.ShowModal;
    form.Free;
    form:=nil;
end;

procedure TMainFm.FormCreate(Sender: TObject);
begin
    CreateObject;
end;

procedure TMainFm.FormDestroy(Sender: TObject);
begin
    FreeObject;
end;

procedure TMainfm.FreeObject;
begin
    FreeAndNil(Manager);
    FreeAndNil(Info);
    FreeAndnil(Draw);
end;

procedure TMainFm.N10Click(Sender: TObject);
begin
    if (Manager.UzelsCount > 0) and (TreeUzel.Selected.Text <> 'Вузли') then
    begin
        try
            Draw.DrawAll;
            Draw.DrawMoreZone(TreeUzel.Selected.Text);
        except
        end;
    end;
end;

procedure TMainFm.N11Click(Sender: TObject);
begin
    VisibleGrid := N11.Checked;
end;

procedure TMainFm.N12Click(Sender: TObject);
begin
    IsSnap := N12.Checked;
end;

procedure TMainFm.N13Click(Sender: TObject);
begin
    if Index > -1 then
        Manager.Elements[Index].ChangeZakr;
    end;
end;

```

```

procedure TMainFm.N14Click(Sender: TObject);
begin
if (Manager.UzelsCount > 0)and(TreeUzel.Selected.Text <> 'Вузли') then
begin
try
Draw.DrawAllnoLine;
Trass.ShowTrass (Manager.FindUzel (TreeUzel.Selected.Text));
except

end;

end;
end;

procedure TMainFm.N15Click(Sender: TObject);
begin
if Save.Execute then
begin
if Info.Save(Save.FileName) then ShowMessage('Файл '+Save.FileName+' успішно
збережений!');
end;
end;

procedure TMainFm.N8Click(Sender: TObject);
begin
if ShowMiniElements then
begin
N8.Checked := false;
end;
Panell.Visible := N8.Checked;
Draw.DrawGrid;
end;

procedure TMainFm.N9Click(Sender: TObject);
begin
if Index > -1 then
begin
Manager.Elements[Index].ChangeRotate;
Draw.DrawAll;
Draw.DrawZone (Index);
end;
end;

procedure TMainFm.PaintBoxClick(Sender: TObject);
begin
{ Index := Manager.FindByPoint (PX, PY);
if Index > -1 then
begin
Draw.DrawZone (Index);
end;
}
end;

procedure TMainFm.PaintBoxMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
ShowMessage (intToStr (PaintBox.Width));
Draw.DrawAll;
Index := -1;
Index := Manager.FindByPoint (PX, PY);
CheckObj := false;
if PaintScheme then
begin
PointScheme.X := X;
PointScheme.Y := Y;
PaintScheme := false;
isPaintScheme := true;
Index := -1;
end;
end;

```

```

    if Index > -1 then
    begin
        Draw.DrawZone(Index);
        CheckObj := true;
    end;
end;

procedure TMainFm.PaintBoxMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
    PX := X;
    PY := Y;
    Move := false;
    if isPaintScheme then
    begin
        Draw.DrawScheme(PointScheme.X, PointScheme.Y, X, Y);
    end;
    if CheckObj then
    begin
        Draw.ChangePoz(Index, X, Y);
        Move := true;
    end;
end;

procedure TMainFm.PaintBoxMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var P, P2 : Tpoint;
    TempKolScheme : integer;
begin
    P.X := PX;
    P.Y := PY;
    CheckObj := false;
    if isPaintScheme then
    begin
        P2.X := X;
        P2.Y := Y;
        Manager.AddSchemea(PointScheme, P2);
        isPaintScheme := false;
        isKolScheme := DrawScheme(isKolScheme);

    end;
    if (Index > -1) and (Move) then
    begin
        Manager.Elements[Index].VerhPoint := P;
        if IsSnap then Draw.SnapToGrid(Manager.Elements[Index]);
    end;
    if Manager.ElementsCount > 0 then Draw.DrawAll;
    if Index > -1 then Draw.DrawZone(Index);
end;

procedure TMainFm.Time2Timer(Sender: TObject);
begin
    if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen, genAlg.FHrom[0].GetGen) then
    begin
        GenAlg.Iteration := 0;
        Prog.Position := 0;
    end
    else
    begin
        Inc(GenAlg.Iteration);
        Prog.StepIt;
    end;
    GenAlg.SetCelFunc;
    Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
    if IsAlgDraw then Draw.DrawAll;
    if (GenAlg.Iteration > genAlg.MinPop) or (isStopAlg) then
    begin
        Draw.DrawAll;
        Time2.Enabled := false;
    end;
end;

```

```

    Panel4.Visible := false;
  end;
end;

procedure TMainFm.TimeTimer(Sender: TObject);
begin
  if GenAlg.Crossing(genAlg.FHrom[0].GetGen, genAlg.FHrom[0].GetGen) then
  begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
  end
  else
  begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
  end;
  GenAlg.SetCelFunc;
  Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
  if IsAlgDraw then Draw.DrawAll;
  if (GenAlg.Iteration > genAlg.MinPop) or (isStopAlg) then
  Begin
    Draw.DrawAll;
    Time.Enabled := false;
    Panel4.Visible := false;
  end;
end;

procedure TMainFm.TreeClick(Sender: TObject);
begin
  if (Manager.ElementsCount > 0) and (Tree.Selected.Text <> 'Пристрої') then
  begin
    try
      Draw.DrawAll;
      Draw.DrawZone(Manager.FindElement(Tree.Selected.Text));
    except
    end;
  end;
end;

procedure TMainFm.actAddElementExecute(Sender: TObject);
begin
  CreateChildForm(fmAddElement, TfmAddElement);
end;

procedure TMainFm.actExitExecute(Sender: TObject);
begin
  Close;
end;

procedure TMainFm.actFormManagerBdExecute(Sender: TObject);
begin
  CreateChildForm(fmManagerBd, TfmManagerBd);
end;

procedure TMainFm.actFormReadFileExecute(Sender: TObject);
begin
  ShagSetk := 0;
  CreateChildForm(fmReadFile, TfmReadFile);
  if Manager.ElementsCount > 0 then
  begin
    if not ShowMiniElements then
    begin
      Panell1.Visible := true;
      n8.Checked := true;
    end;
    btRazmCont.Enabled := false;
    btKompCont.Enabled := false;
  end;
end;

```

```

Info.ShowElements(Tree,TreeUzel);
Tree.Items[0].Expand(False);
Tree.Items[0].Selected:=true;
Tree.SetFocus;
TreeUzel.Items[0].Expand(False);
TreeUzel.Items[0].Selected:=true;
TreeUzel.SetFocus;
if ShowMiniElements then
begin
  Panell.Visible := false;
  n8.Checked := false;
end;
Manager.DeleteAllSchemea;
Draw.DrawFirst;
Draw.DrawAll;
AlgStart := false;
end;
Status.Panels.Items[1].Text := IntToStr(Manager.ElementsCount);
end;

procedure TMainFm.actOptionsOpenExecute(Sender: TObject);
begin
  CreateChildForm(fmOptions,TfmOptions);
end;

procedure TMainFm.Button1Click(Sender: TObject);
var D : string;
    // X : integer;
begin
  if Manager.ElementsCount > 0 then
  begin
    if InputQuery('Введіть кількість пристроїв','',D) then
    begin
      manager.DeleteAllSchemea;
      Draw.DrawAll;
      KolScheme := StrToInt(D);
      isKolScheme := DrawScheme(0);
    end;
  end
  else
  begin
    ShowMessage('Необхідно експортувати пристрої!');
  end;
end;

procedure TMainFm.Button2Click(Sender: TObject);
begin
  IsStopAlg := true;
end;

procedure TMainFm.Button3Click(Sender: TObject);
begin
  if manager.UzelsCount > 0 then
  begin
    Trass.Prepare;
  end;
end;

procedure TMainFm.btKompContClick(Sender: TObject);
begin
  if AlgStart then
  begin
    GenAlg.PrepareSecond;
    GenAlg.Inversion := optInversion;
    GenAlg.MinPop := OptMinPop;
    GenAlg.Mutation := OptMutation;
    AlgStart := true;
    Draw.DrawAll;
    Panel4.Visible := true;
  end;
end;

```

```

    isStopAlg := false;
    Prog.Max := GenAlg.MinPop;
    Prog.Position := 0;
    if isAlgDraw then
begin
    Time2.Enabled := true;
end
else
begin
    while GenAlg.Iteration < GenAlg.MinPop do
begin
    if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen) then
begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
end
else
begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
end;
// GenAlg.SetCelFunc;
    Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
//if IsAlgDraw then Draw.DrawAll;
end;
    Draw.DrawAll;
    Time2.Enabled := false;
    Panel4.Visible := false;
end;
end;
end;

procedure TMainFm.btKompClick(Sender: TObject);
begin
    if Manager.SchemesCount > 1 then
begin
    GenAlg.PrepareSecond;
    IsStopAlg := false;
    GenAlg.Invertion := optInvertion;
    GenAlg.MinPop := OptMinPop;
    GenAlg.Mutation := OptMutation;
    AlgStart := true;
    Draw.DrawAll;
    Panel4.Visible := true;
    Prog.Max := GenAlg.MinPop;
    btKompCont.Enabled := true;
    Prog.Position := 0;
    if isAlgDraw then
begin
    Time2.Enabled := true;
end
else
begin
    while GenAlg.Iteration < GenAlg.MinPop do
begin
    if GenAlg.CrossingKomp(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen) then
begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
end
else
begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
end;
// GenAlg.SetCelFunc;
    Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
//if IsAlgDraw then Draw.DrawAll;
end;

```

```

    Draw.DrawAll;
    Time2.Enabled := false;
    Panel4.Visible := false;
end;
end
else
begin
    ShowMessage('Для прогнозування необхідно не менш 2 пристроїв!');
end;
end;

procedure TMainFm.btRazmClick(Sender: TObject);
begin
    if Manager.SchemesCount > 0 then
    begin
        GenAlg.PrepareSecond;
        IsStopAlg := false;
        GenAlg.Inversion := optInversion;
        GenAlg.MinPop := OptMinPop;
        GenAlg.Mutation := OptMutation;
        AlgStart := true;
        Draw.DrawAll;
        Panel4.Visible := true;
        Prog.Max := GenAlg.MinPop;
        btRazmCont.Enabled := true;
        Prog.Position := 0;
        if isAlgDraw then
        begin
            Time.Enabled := true;
        end
        else
        begin
            while GenAlg.Iteration < GenAlg.MinPop do
            begin
                if GenAlg.Crossing(genAlg.FHrom[0].GetGen, genAlg.FHrom[0].GetGen) then
                begin
                    GenAlg.Iteration := 0;
                    Prog.Position := 0;
                end
                else
                begin
                    Inc(GenAlg.Iteration);
                    Prog.StepIt;
                end;
            // GenAlg.SetCelFunc;
                Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
            //if IsAlgDraw then Draw.DrawAll;
            end;
            Draw.DrawAll;
            Time.Enabled := false;
            Panel4.Visible := false;
            end;
        end

end;

end;

procedure TMainFm.btRazmContClick(Sender: TObject);
begin
    if AlgStart then
    begin
        GenAlg.PrepareSecond;
        GenAlg.Inversion := optInversion;
        GenAlg.MinPop := OptMinPop;
        GenAlg.Mutation := OptMutation;
        AlgStart := true;
        Draw.DrawAll;
        Panel4.Visible := true;
        isStopAlg := false;
        Prog.Max := GenAlg.MinPop;
    end;
end;

```

```

    Prog.Position := 0;
    if isAlgDraw then
begin
    Time.Enabled := true;
end
else
begin
    while GenAlg.Iteration < GenAlg.MinPop do
begin
    if GenAlg.Crossing(genAlg.FHrom[0].GetGen,genAlg.FHrom[0].GetGen) then
begin
    GenAlg.Iteration := 0;
    Prog.Position := 0;
end
else
begin
    Inc(GenAlg.Iteration);
    Prog.StepIt;
end;
end;
// GenAlg.SetCelFunc;
    Status.Panels[5].Text := IntToStr(GenAlg.CelFunc);
//if IsAlgDraw then Draw.DrawAll;
end;
    Draw.DrawAll;
    Time.Enabled := false;
    Panel4.Visible := false;
end;
end;
end;

procedure TMainFm.CreateObject;
var
server_name: string;
database_name: string;
begin
    Save.FileName := GetCurrentDir + '\SAVE';
    Trass := TTrass.Create;
    Manager := TManager.Create;
    Info := TInformation.Create;
    ShagSetk := 0;
    AlgStart := false;
    GenAlg := TgenAlg.Create;
    PaintScheme := false;
    isPaintScheme := false;
    Draw := TDraw.Create(PaintBox);
    try
        ini:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'Prof.ini');
        server_name:=ini.ReadString('options','server_name','');
        database_name:=ini.readString('options','database_name','');
        if (server_name='') or (database_name='') then
begin
            server_name:='localhost';
            if not InputQuery('Уведіть ім'я сервера','',server_name) then
begin
                Application.Terminate;
                Exit;
            end;
            if not InputQuery('Введіть повний шлях до файлу БД','',database_name) then
begin
                Application.Terminate;
                Exit;
            end;
            ini.WriteString('options','server_name',server_name);
            ini.WriteString('options','database_name',database_name);
        end;
    try
        Data.Database.DatabaseName := database_name;
        Data.DataBase.Connected := true;
        Data.Transaction.Active := true;

```

```
except
    ShowMessage('Неправильні параметри підключення');
    Application.Terminate;
    Exit;
end;
finally
    LoadOptions(ini);
    //FreeAndNil(ini);
end;
end;

end.
```

КБПЗ\_2024

## Файл Connector.pas – створення з'єднань між мережними пристроями

```

unit Connector;

interface

type
  TConnection = record
    CStart: record
      x,y: integer;
    end;
    CEnd: record
      x,y: integer;
    end;
    Num: Integer;
  end;
  TField = array [0..200,0..200] of Integer;

var
  MField, Field: TField;

procedure Connect;

implementation

uses windows, SysUtils, math, Unit1, Unit3, Types;

procedure Connect;
var i,j,k,l,m: Integer;
    temp: TField;
    cur: Integer;
    x,y: Integer;
    napr: Integer;
    naprd: Integer;
    Och: array [1..1000] of record
      x,y:byte;
    end;

    uk1,uk2: integer;
    tocon: Integer;
    ppp: TConnection;

const dd: array [1..4,1..3] of record
      dx,dy: integer;
    end =
  ((dx:0;dy:-1), (dx:-1;dy:0), (dx:1;dy:0)),
  ((dx:1;dy:0), (dx:0;dy:-1), (dx:0;dy:1)),
  ((dx:0;dy:1), (dx:1;dy:0), (dx:-1;dy:0)),
  ((dx:-1;dy:0), (dx:0;dy:1), (dx:0;dy:-1)));
{
  tt: array [0..5,0..3] of byte =
  (10,9,10,7),
  (8,10,6,10),
  (9,10,10,6),
  (10,10,9,8),
  (7,6,10,10),
  (10,8,7,10));}

function Get(var x,y:integer):Boolean;
begin
  Result:=true;
  if uk1 = uk2 then begin Result:=false; exit; end;
  x:=och[uk1].x;
  y:=och[uk1].y;
  uk1:=uk1+1;
  if uk1>1000 then uk1:=1;
end;
procedure Put(x,y:integer);
begin
  och[uk2].x:=x;

```

```

    och[uk2].y:=y;
    uk2:=uk2+1;
    if uk2>1000 then uk2:=1;
end;
function Test(x,y: Integer):boolean;
begin
    Result:=false;
    if (x<1) or (x>199) or (y<1) or (y>199) then exit;
    if (x=ppp.CStart.x) and (y=ppp.CStart.y) then
        begin Result:=true; temp[x,y]:=napr*100000+cur+1; exit; end;
    if (MField[x,y]=tocon) and (tocon<>-1) then
        begin Result:=true; temp[x,y]:=napr*100000+cur+1; exit; end;
    if temp[x,y]=9999999 then exit;
//    if (napr mod 2 = 1) and ((temp[x-1,y]=9999999) or (temp[x+1,y]=9999999))
then exit;
    if (temp[x,y]>10001) and (temp[x,y]<10011) then exit;
    if (temp[x,y]=10000) and (napr mod 2 = 0) then exit;
    if (temp[x,y]=10001) and (napr mod 2 = 1) then exit;
    if (temp[x,y]<>0) and (temp[x,y] mod 100000 < cur) then exit;
    temp[x,y]:=napr*100000+cur+1;
    Put(x,y);
end;
function Find(var x,y: Integer): boolean;
var t: integer;
    xx,yy: integer;
begin
    Result:=true;
    xx:=x; yy:=y;
    t:=temp[x,y] mod 100000;
    napr:=temp[x,y] div 100000 + 2; if napr>4 then napr:= napr-4;
    if (temp[x+dd[napr,1].dx,y+dd[napr,1].dy] mod 100000 < t) and
(temp[x+dd[napr,1].dx,y+dd[napr,1].dy]<>0) then begin xx:=x+dd[napr,1].dx;
yy:=y+dd[napr,1].dy; t:=temp[xx,yy] mod 100000; end;
    if (temp[x+dd[napr,2].dx,y+dd[napr,2].dy] mod 100000 < t) and
(temp[x+dd[napr,2].dx,y+dd[napr,2].dy]<>0) then begin xx:=x+dd[napr,2].dx;
yy:=y+dd[napr,2].dy; t:=temp[xx,yy] mod 100000; end;
    if (temp[x+dd[napr,3].dx,y+dd[napr,3].dy] mod 100000 < t) and
(temp[x+dd[napr,3].dx,y+dd[napr,3].dy]<>0) then begin xx:=x+dd[napr,3].dx;
yy:=y+dd[napr,3].dy; t:=temp[xx,yy] mod 100000; end;
    if (x=xx) and (y=yy) then Result:=False;
    x:=xx;
    y:=yy;
    if (x=ppp.CEnd.x) and (y=ppp.CEnd.y) then Result:=false;
end;
label 1,2,3,4;
begin
    for i:=0 to 200 do for j:=0 to 200 do begin Field[i,j]:=0; MField[i,j]:=-1;
end;
    for k:=1 to High(Elements) do
        for i:=0 to Elements[k].Size.x do
            for j:=0 to Elements[k].Size.y do
Field[Elements[k].Coord.x+i,Elements[k].Coord.y+j]:=9999999;
        for k:=0 to High(Cons) do
            for m:=1 to High(Cons[k]) do
                begin
                    ppp.CStart.x:=Cons[k][0].X;
                    ppp.CStart.y:=Cons[k][0].Y;
                    ppp.CEnd.x:=Cons[k][m].X;
                    ppp.CEnd.y:=Cons[k][m].Y;
                    temp:=Field;
                    cur:=1;
                    x:=ppp.CEnd.x;
                    y:=ppp.CEnd.y;
                    tocon:=MField[x,y];
                {
                    if tocon<>-1 then begin
                        ppp.CEnd.x:=ppp.CStart.x;
                        ppp.CEnd.y:=ppp.CStart.y;
                        ppp.CStart.x:=x;
                        ppp.CStart.y:=y;

```

```

                                x:=ppp.CEnd.x;
                                y:=ppp.CEnd.y;
                                if MField[x,y]<>-1 then goto 2;
                                end;}
                                if MField[ppp.CStart.x,ppp.CStart.y]<>-1 then
tocon:=MField[ppp.CStart.x,ppp.CStart.y];
                                uk1:=1;
                                uk2:=1;
                                if (temp[x+1,y]=0) or (temp[x+1,y]=10000) or (temp[x+1,y]=10001) then
napr:=2;
                                if (temp[ x-1,y]=0) or (temp[ x-1,y]=10000) or (temp[ x-1,y]=10001) then
napr:=4;
                                if (temp[x,y+1]=0) or (temp[x,y+1]=10000) or (temp[x,y+1]=10001) then
napr:=3;
                                if (temp[x, y-1]=0) or (temp[x, y-1]=10000) or (temp[x, y-1]=10001) then
napr:=1;
                                temp[x,y]:=1+napr*100000;
                                Put(x,y);
                                while true do begin
                                    if Get(i,j) then
                                        begin
                                            cur:=temp[i,j] mod 100000;
                                            naprd:=temp[i,j] div 100000;
                                            napr:=naprd;
                                            if Test(i+dd[naprd,1].dx,j+dd[naprd,1].dy) then goto 1;
                                            if MField[i,j]=-1 then begin
                                                cur:=cur+5;
                                                napr:= naprd-1; if napr=0 then napr:=4;
                                                if Test(i+dd[naprd,2].dx,j+dd[naprd,2].dy) then goto 1;
                                                napr:=naprd+1; if napr=5 then napr:=1;
                                                if Test(i+dd[naprd,3].dx,j+dd[naprd,3].dy) then goto 1;
                                                cur:= cur-5;
                                            end;
                                        end else goto 2;
                                    MessageBox(0,pchar(inttostr(cur)),'',0);
                                    if cur>300 then goto 2;
                                end;
1:
                                if tocon=-1 then begin
                                    x:=ppp.CStart.x;
                                    y:=ppp.CStart.y;
                                    l:=k;
                                end else begin
                                    x:=i;
                                    y:=j;
                                    l:=tocon;
                                    Field[x+dd[napr,1].dx,y+dd[napr,1].dy]:=10010;
                                    if (x=ppp.CEnd.x) and (y=ppp.CEnd.y) then goto 3;
                                    napr:=napr+2; if napr>4 then napr:= napr-4;
                                    naprd:=napr;
                                    goto 4;
                                end;
                                naprd:=temp[x,y] div 100000 + 2; if naprd>4 then naprd:= naprd-4;
                                napr:=napr+2; if napr>4 then napr:= napr-4;
                                while Find(x,y) do
                                    begin
4:
                                        if napr=naprd then
                                            begin
                                                if MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]=-1 then
                                                    Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10000+naprd mod 2
                                                    else if Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]<>10010 then
                                                        Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10006;
                                                    end else begin
                                                        if ((naprd=2) and (napr=1)) or ((naprd=3) and (napr=4)) then
                                                            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10002;
                                                        if ((naprd=1) and (napr=4)) or ((naprd=2) and (napr=3)) then
                                                            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10003;
                                                        if ((naprd=3) and (napr=2)) or ((naprd=4) and (napr=1)) then

```

```

        Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10004;
        if ((naprd=1) and (napr=2)) or ((naprd=4) and (napr=3)) then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10005;
        end;
        naprd:=napr;
        MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=1;
    end;
if napr=naprd then
    begin
        if MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]=-1 then
            Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10000+naprd mod 2
            else Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10006;
        end else begin
            if ((naprd=2) and (napr=1)) or ((naprd=3) and (napr=4)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10002;
            if ((naprd=1) and (napr=4)) or ((naprd=2) and (napr=3)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10003;
            if ((naprd=3) and (napr=2)) or ((naprd=4) and (napr=1)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10004;
            if ((naprd=1) and (napr=2)) or ((naprd=4) and (napr=3)) then
                Field[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=10005;
            end;
            naprd:=napr;
            MField[ x-dd[napr,1].dx, y-dd[napr,1].dy]:=1;
        3:
            Field[x,y]:=10000+naprd mod 2;
            MField[x,y]:=1;
        2:
            end;
        end;
    end.

```

## Файл Db.pas - створення бази даних мережних пристроїв

```

unit Db;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Grids, DBGrids, DB, DBTables, ExtCtrls, StdCtrls,
  Mask, DBCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    GroupBox4: TGroupBox;
    Panel2: TPanel;
    Image1: TImage;
    GroupBox5: TGroupBox;
    Label3: TLabel;
    Button2: TButton;
    DBEdit5: TDBEdit;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    DBEdit6: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    Panel3: TPanel;
    TabSheet3: TTabSheet;
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses Main1, Main2, Db_elements;

{$R *.dfm}

```

```
procedure TForm3.FormCreate(Sender: TObject);
begin
  Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
  Table1.TableName := 'microchips.db';
  Table1.Active := true;
end;

procedure TForm3.Button5Click(Sender: TObject);
begin
  Close;
  Table1.Edit;
  Table1.Post;
end;

procedure TForm3.Button4Click(Sender: TObject);
begin
  if Table1.RecordCount > 0 then
  begin
    begin
      if Application.MessageBox('Ви дійсно хочете видалити поточний мережний пристрій', 'Увага', MB_YESNO) = IDYES then
        Table1.Delete;
    end;
  end;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
  form11.ShowModal;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  form12.ShowModal;
end;

procedure TForm3.DBGrid1TitleClick(Column: TColumn);
begin
  if column.Index = 0 then begin Table1.IndexFieldNames := 'Mark';
  DBGrid1.Columns[0].Title.Caption := 'Назва'; DBGrid1.Columns[1].Title.Caption := 'Тип'; end;;
  if column.Index = 1 then begin Table1.IndexFieldNames := 'Type';
  DBGrid1.Columns[0].Title.Caption := 'Назва'; DBGrid1.Columns[1].Title.Caption := 'Тип*'; end;
end;

procedure TForm3.Table1AfterScroll(DataSet: TDataSet);
var i: integer;
begin
  Form7.Table1.First;
  for i := 0 to form7.Table1.RecordCount - 1 do
  begin
    if form7.Table1.fieldbyname('Name').AsString =
    table1.FieldName('Elem_name').AsString then break;
    form7.Table1.Next;
  end;
end;

end.
```

## Файл Search.pas - вікно пошуку мережних пристроїв

```

unit Search;

interface

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

type
  TForm4 = class(TForm)
    LabeledEdit1: TLabeledEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

uses Db;

{$R *.dfm}

procedure TForm4.FormActivate(Sender: TObject);
begin
  LabeledEdit1.SetFocus;
  LabeledEdit1.SelStart := 0;
  LabeledEdit1.SelLength := 255;
end;

procedure TForm4.Button1Click(Sender: TObject);
var i : integer;
begin
  Form3.Table1.first;
  for i := 0 to Form3.Table1.RecordCount - 1 do
    begin
      if pos (LabeledEdit1.Text, Form3.Table1.fieldbyname('Mark').asString) <> 0
      then begin close; {Form3.DBGrid1.SetFocus;} exit end;
      Form3.Table1.Next;
    end;
  Form3.Table1.First;
  Application.MessageBox('Пристрій не знайдено', 'Увага!', MB_OK);
  Close;
  Exit; end; end.

```

**Файл Effic.PAS - оцінка ефективності спроектованої розподіленої інформаційної системи**

```

unit Effic;

interface

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

type
  TEfficForm = class(TForm)
    gbxCShares: TGroupBox;
    lbxCShares: TListBox;
    gbxCSessions: TGroupBox;
    lvCSessions: TListView;
    bvlCSessions: TBevel;
    gbxCFiles: TGroupBox;
    btnCGetShares: TButton;
    btnCCloseShares: TButton;
    btnCAddShares: TButton;
    btnCCloseSession: TButton;
    btnCGetSessions: TButton;
    bvlCTopSessions: TBevel;
    plCButtonFiles: TPanel;
    btnCGetFiles: TButton;
    btnCCloseFile: TButton;
    bvlCLeftFiles: TBevel;
    plCFiles: TPanel;
    lvCFiles: TListView;
    bvlCTopFiles: TBevel;
    gbxCTraffic: TGroupBox;
    lvCTraffic: TListView;
    bvlCTraffic: TBevel;
    tmrCTraffic: TTimer;
    function IsXP(var Value: Boolean): Boolean;
    procedure btnCGetSharesClick(Sender: TObject);
    procedure btnCCloseSharesClick(Sender: TObject);
    function SelectDirectory: String;
    procedure btnCAddSharesClick(Sender: TObject);
    function CardinalToTimeStr(Value: Cardinal): String;
    procedure btnCGetSessionsClick(Sender: TObject);
    procedure btnCCloseSessionClick(Sender: TObject);
    procedure btnCGetFilesClick(Sender: TObject);
    procedure btnCCloseFileClick(Sender: TObject);
    procedure tmrCTrafficTimer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    SessionCloseKey: array [0..512] of SmallInt;
  end;

type
  TShareInfo2 = packed record
    shi2_netname : 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_netname : 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;
    dwInNUCastPkts : DWORD;
    dwInDiscards    : DWORD;
    dwInErrors      : DWORD;
    dwInUnknownProtos : DWORD;
    dwOutOctets     : DWORD;
    dwOutUcastPkts : DWORD;
    dwOutNUCastPkts : 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
  NetShareEnumXP: function (   servername:PWChar;
                             level:DWORD;
                             bufptr:Pointer;
                             pefmaxlen:DWORD;
                             entriesread,
                             totalentries,
                             resume_handle:LPDWORD): DWORD; stdcall;

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

var
  NetShareDelXP: function (servername: PWideChar;
                           netname: PWideChar;
                           reserved: DWORD): LongInt; stdcall;

var
  NetShareDel: function ( pszServer,
                           pszNetName:PChar;
                           usReserved:Word): DWORD; stdcall;

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

var
  NetShareAdd: function ( pszServer:Pchar;
                           sLevel:Cardinal;

```

```

        pbBuffer:PChar;
        cbBuffer:Word):DWORD; stdcall;

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

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

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

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

var
NetFileEnumXP:function ( servername,
        basepath,
        username:PWChar;
        level:DWORD;
        bufptr:Pointer;
        pefmaxlen:DWORD;
        entriesread,
        totalentries,
        resume_handle:LPDWORD):DWORD; stdcall;

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

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

var
NetFileClose2:function ( pszServer:PChar;
        ulFileId:LongWord):DWORD; stdcall;

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

var
    EfficForm: TEfficForm;

implementation

```

```

{$R *.dfm}

{ TEfficForm }

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

function TEfficForm.IsXP(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.dwSchemeformId of //визначаємося
        VER_SCHEMEFORM_WIN32_XP      : Value := True; //Windows XP - підходить
        VER_SCHEMEFORM_WIN32_WINDOWS : Value := False; //Windows 9x - підходить
        VER_SCHEMEFORM_WIN32s       : Result := False //Windows 3.x - не підходить
    end;
end;

//
// Одержання всіх відкритих загальних ресурсів
//

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

    if OS then begin //Код для XP
        FLibHandle := LoadLibrary('NETAPI32.DLL'); //Завантажуємо бібліотеку
        if FLibHandle = 0 then Exit;
        //Зв'язуємо функцію
        @NetShareEnumXP := GetProcAddress(FLibHandle, 'NetShareEnum');
        if not Assigned(NetShareEnumXP) then //Перевірка
        begin
            FreeLibrary(FLibHandle);
            Exit;
        end;
        ShareXP := nil; //Очищаємо покажчик на масив структур
        //Виклик функції
        if NetShareEnumXP(nil, 2, @ShareXP, 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(ShareXP^[i].shi2_netname));
    end;
end;

```

```

end else begin //Код для 9x
  FLibHandle := LoadLibrary('SVRAPI.DLL'); //Завантажуємо бібліотеку
  if FLibHandle = 0 then Exit;
  //Зв'язуємо функцію
  @NetShareEnum := GetProcAddress(FLibHandle, 'NetShareEnum');
  if not Assigned(NetShareEnum) then //Перевірка
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  //Виклик функції
  if NetShareEnum(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_netname));
  end;
  FreeLibrary(FLibHandle); //Не забуваємо вивантажити бібліотеку
end;

//
// Закриття загального ресурсу
//

procedure TEfficForm.btnCloseSharesClick(Sender: TObject);
var
  OS:Boolean;
  FLibHandle : THandle;
  Name9x:array [0..12] of Char;
  NameXP:PWChar;
  i:Integer;
  ShareName: String;
begin
  if not IsXP(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 XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDelXP := GetProcAddress(FLibHandle, 'NetShareDel');
    if not Assigned(NetShareDelXP) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    i:= SizeOf(WideChar)*256;
    GetMem(NameXP, i); //Виділяємо пам'ять під змінну
    StringToWideChar(ShareName, NameXP, i); //Перетворимо в PWideChar
    NetShareDelXP(nil, NameXP, 0); //Видаляємо ресурс
    FreeMem(NameXP); //Звільняємо пам'ять
  end else begin //Код для 9x
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetShareDel := GetProcAddress(FLibHandle, 'NetShareDel');
    if not Assigned(NetShareDel) then //Перевірка
    begin
      FreeLibrary(FLibHandle);
      Exit;
    end;
    FillChar(Name9x, SizeOf(Name9x), #0); //Очищаємо масив
  end;
end;

```

```

        move(ShareName[1],Name9x[0],Length(ShareName)); //Заповнюємо масив
        NetShareDel(nil,@Name9x,0); //Видаляємо ресурс
    end;
    FreeLibrary(FLibHandle);
end;

//
// Показу діалогу вибору директорії
//

function TEfficForm.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 := '';
    Result := String(TempPath);
end;

//
// Додавання загального ресурсу
//

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

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

    if OS then begin //Код для XP
        FLibHandle := LoadLibrary('NETAPI32.DLL');
        if FLibHandle = 0 then Exit;
        @NetShareAddXP := GetProcAddress(FLibHandle,'NetShareAdd');
        if not Assigned(NetShareAddXP) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        TmpLength := SizeOF(WideChar)*256; //Визначаємо необхідний розмір

        GetMem(TmpNameXP, TmpLength); //Конвертуємо в PWChar
        StringToWideChar(TmpName, TmpNameXP, TmpLength);
        ShareXP.shi2_netname := TmpNameXP; //Ім'я
    end;
end;

```

```

ShareXP.shi2_type := STYPE_DISKTREE; //Тип ресурсу
ShareXP.shi2_remark := ''; //Коментар
ShareXP.shi2_permissions := ACCESS_ALL; //Доступ
ShareXP.shi2_max_uses := DWORD(-1); // Кількість максим. підключень
ShareXP.shi2_current_uses := 0; // Кількість поточних підключень

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

ShareXP.shi2_passwd := ''; //Пароль

NetShareAddXP(nil, 2, @ShareXP, nil); //Додаємо ресурс
FreeMem (TmpNameXP); //звільняємо пам'ять
FreeMem (TmpDirXP);
end else begin
  FLibHandle := LoadLibrary('SVRAPI.DLL');
  if FLibHandle = 0 then Exit;
  @NetShareAdd := GetProcAddress(FLibHandle, 'NetShareAdd');
  if not Assigned(NetShareAdd) then
  begin
    FreeLibrary(FLibHandle);
    Exit;
  end;
  FillChar(Share9x.shi50_netname, SizeOf(Share9x.shi50_netname), #0);
  move(TmpName[1], Share9x.shi50_netname[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
  NetShareAdd(nil, 50, @Share9x, SizeOf(Share9x));
end;
FreeLibrary(FLibHandle);
end;

//
// Помітьте що активний і неактивний час сесій буде даватися нам
// у вигляді кількості секунд (тип Cardinal). Пропоную написати невелику
// функцію, завдання якої буде перетворювати кількість секунд у більше
// звичну форму відображення.
//

function TEfficForm.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;
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
//

procedure TEfficForm.btnGetSessionsClick(Sender: TObject);
var
    OS: Boolean;
    FLibHandle : THandle;
    SessionInfo50: array [0..512] of TSessionInfo50;
    SessionInfo502 : PSessionInfo502Array;
    TotalEntries,EntriesReadXP: DWORD;
    EntriesRead,TotalAvial: Word;
    i:integer;
begin
    lvSessions.Items.Clear;

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

    if OS then begin //Код для XP
        FLibHandle := LoadLibrary('NETAPI32.DLL');
        if FLibHandle = 0 then Exit;
        @NetSessionEnumXP := GetProcAddress(FLibHandle, 'NetSessionEnum');
        if not Assigned(NetSessionEnumXP) then
            begin
                FreeLibrary(FLibHandle);
                Exit;
            end;
        SessionInfo502 := nil;
        if NetSessionEnumXP(nil,nil,nil,502,@SessionInfo502,DWORD(-
1),@entriesreadXP, @totalentries, nil)=0 then
            for i:=0 to EntriesReadXP-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
активне
                            SubItems.Add(CardinalToTimeStr(SessionInfo502^[i].sesi502_idle_time));
//Час не активне
                        end;
                    end;
                end else begin //Код для Windows 9x
                    FLibHandle := LoadLibrary('SVRAPI.DLL');
                    if FLibHandle = 0 then Exit;
                    @NetSessionEnum := GetProcAddress(FLibHandle, 'NetSessionEnum');
                    if not Assigned(NetSessionEnum) then
                        begin
                            FreeLibrary(FLibHandle);
                            Exit;
                        end;
                    if NetSessionEnum
(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
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
активне
SubItems.Add(CardinalToTimeStr(SessionInfo50[i].sesi50_idle_time));
//Час не активне
SessionCloseKey[i]:= SessionInfo50[i].sesi50_key; //Унікальний
ідентифікатор для закриття
end;
end;
end;
FreeLibrary(FLibHandle);
end;

//
// Завершення обраної сесії
//

procedure TEfficForm.btnCloseSessionClick(Sender: TObject);
var
  OS: Boolean;
  FLibHandle : THandle;
  CNameXP: PWideChar;
  CName9x: PAnsiChar;
  Key: SmallInt;
  i: Integer;
begin
  if not IsXP(OS) then Close; //З'ясуємо тип системи

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

  if OS then begin
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetSessionDelXP := GetProcAddress(FLibHandle, 'NetSessionDel');
    if not Assigned(NetSessionDelXP) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    //Перетворимо дані в необхідний вид
    CNameXP := PWChar(WideString('\'+lvSessions.Items.Item[i].Caption));
    NetSessionDelXP(nil, CNameXP, nil);
  end else begin
    FLibHandle := LoadLibrary('SVRAPI.DLL');
    if FLibHandle = 0 then Exit;
    @NetSessionDel := GetProcAddress(FLibHandle, 'NetSessionDel');
    if not Assigned(NetSessionDel) 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
    NetSessionDel(nil, CName9x, Key);
  end;
  FreeLibrary(FLibHandle);
end;

//
// Одержання списку відкритих файлів
//

procedure TEfficForm.btnGetFilesClick(Sender: TObject);
var

```

```

OS: Boolean;
FLibHandle : THandle;
FileInfoXP: PFileInfo3Array;
FileInfo9x: array [0..512] of TFileInfo50;
TotalEntries,EntriesReadXP: DWORD;
EntriesRead,TotalAvial: Word;
i:integer;
begin
  lvfiles.Items.Clear;

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

  if OS then begin //Код для XP
    FLibHandle := LoadLibrary('NETAPI32.DLL');
    if FLibHandle = 0 then Exit;
    @NetFileEnumXP := GetProcAddress(FLibHandle, 'NetFileEnum');
    if not Assigned(NetFileEnumXP) then
      begin
        FreeLibrary(FLibHandle);
        Exit;
      end;
    FileInfoXP := nil;
    if NetFileEnumXP(nil,nil,nil,3,@FileInfoXP,DWORD(-1),@entriesreadXP,
@totalentries, nil)=0 then
      for i:=0 to EntriesReadXP-1 do
        begin
          with lvFiles.Items.Add do //Заповнення даними зі структури
            begin
              Caption := string(IntToStr(FileInfoXP^[i].fi3_id)); //Ідентифікатор
              SubItems.Add(FileInfoXP^[i].fi3_pathname); //Шлях до файлу
              SubItems.Add(FileInfoXP^[i].fi3_username); //Ім'я користувача
            end;
          end;
        end else begin //Код для Windows 9x
          FLibHandle := LoadLibrary('SVRAPI.DLL');
          if FLibHandle = 0 then Exit;
          @NetFileEnum := GetProcAddress(FLibHandle, 'NetFileEnum');
          if not Assigned(NetFileEnum) then
            begin
              FreeLibrary(FLibHandle);
              Exit;
            end;
          if NetFileEnum (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 TEfficForm.btnCloseFileClick(Sender: TObject);
          var
            OS: Boolean;
            FLibHandle : THandle;
            i: Integer;
          begin
            if not IsXP(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 XP
  FLibHandle := LoadLibrary('NETAPI32.DLL');
  if FLibHandle = 0 then Exit;
  @NetFileClose := GetProcAddress(FLibHandle, 'NetFileClose');
  if not Assigned(NetFileClose) then
  begin
    FreeLibrary(FLibHandle);
    Close;
  end;
  NetFileClose(nil, StrToInt(lvFiles.Items.Item[i].Caption)); //Закриваємо файл
end else begin //Код для Windows 9x
  FLibHandle := LoadLibrary('SVRAPI.DLL');
  if FLibHandle = 0 then Exit;
  @NetFileClose2 := GetProcAddress(FLibHandle, 'NetFileClose2');
  if not Assigned(NetFileClose2) then
  begin
    FreeLibrary(FLibHandle);
    Close;
  end;
  NetFileClose2(nil, StrToInt(lvFiles.Items.Item[i].Caption));
end;
FreeLibrary(FLibHandle);
end;

//
// Визначаємо вхідний - вихідний трафік
//

procedure TEfficForm.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 про всякий випадок 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
      SubItems.Add(IntToStr(Table.Table[i].dwOutOctets)); //Усього відправлено
байт
    end;
  end;
  lvTraffic.Items.EndUpdate;
  FreeLibrary(FLibHandle);
  tmrTraffic.Enabled := true; //Не забуваємо активувати таймер
end;

end.
```

КБПЗ\_2024

## Файл About.PAS - довідка про програму

```
unit About;

interface

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

type
  TForm2 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
  Form2.Close;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  Memo1.Clear;
  Memo1.Lines.Add('БАКЛАВРСЬКА ДИПЛОМНА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Програмне забезпечення системи дотримання відповідності SLA для
сервісів IaaS');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Пушкаренко Віталій Олегович');
  Memo1.Lines.Add('                гр. КМ-21-ЗСК');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Якименко Н.М');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('м. Кропивницький 2024');
end;
end.
```