

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

“ \_\_\_ ” \_\_\_\_\_ 2022 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
за другим (магістерським) рівнем вищої освіти  
на тему

**“Дослідження та програмна реалізація системи бюджетування  
хмарної інформаційної системи для визначення ефективності її  
впровадження”**

Виконав здобувач вищої освіти

II курсу, групи КН-21М-1,4

ОПП «Комп’ютерні науки»

спеціальності 122 «Комп’ютерні науки»

Поляруш Б.С.

« \_\_\_ » \_\_\_\_\_ 2022 р.

Керівник проекту

кандидат технічних наук, доцент

Доренський О.П.

« \_\_\_ » \_\_\_\_\_ 2022 р.

Рецензент \_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 "Інформаційні технології"  
Спеціальність 122 "Комп'ютерні науки"  
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

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

Полярушу Богдану Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження

2. Керівник роботи Доренський Олександр Павлович, канд. техн. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 18-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту 10.12.2022 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання  
« 6 » вересня 2022 р.

Підпис керівника

Доренський О.П.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2022 р.

Підпис здобувача

Поляруш Б.С.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Поляруш Б.С. Дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Метою розробки є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Об'єктом дослідження є процес бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Предметом дослідження є методи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

Результат роботи – програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

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

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

Програму розроблено в середовищі Delphi 10.4.

**Ключові слова:** комп'ютерні науки, бюджетування, хмарна інформаційна система, ефективність

## ABSTRACT

**Poliarush B.S. Research and software implementation of the budgeting system of the cloud information system to determine the effectiveness of its implementation. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.**

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the budgeting system of the cloud information system to determine the effectiveness of its implementation.

The purpose of the development is the research and software implementation of the budgeting system of the cloud information system to determine the effectiveness of its implementation.

The object of the research is the budgeting process of the cloud information system to determine the effectiveness of its implementation.

The subject of the study is the budgeting methods of the cloud information system to determine the efficiency of its implementation.

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

The result of the work is the software implementation of the budgeting system of the cloud information system to determine the effectiveness of its implementation.

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 PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4 environment.

**Keywords:** computer science, budgeting, cloud information system, efficiency

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання .....	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	27
3.1 Опис функціонування системи .....	27
3.2 Розробка структурної схеми.....	42
3.3 Розробка функціональної схеми .....	46
3.4 Розробка діаграми процесів.....	50
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	52
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	60
6 НАУКОВА НОВИЗНА .....	63

						ВКРМ-122.22.0012.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження	Літ.	Аркуш	Аркушів
Розроб.	Поляруш Б.С.					М	1	104
Перев.	Доренський О.П.							
Н.контр.	Гермак В.С.							
Затв.	Смірнов О.А.						ЦНТУ КН-21М-1,4	

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	64
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	64
7.2 Розрахунок трудомісткості розробки програмної продукції.....	66
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	68
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	73
7.5 Визначення собівартості розробки та ціни програмної продукції.....	77
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	80
7.7 Визначення експлуатаційних витрат.....	81
7.8 Визначення економічної ефективності програмної продукції.....	82
7.9 Висновок.....	84
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	85
8.1 Аналіз умов праці на робочому місці програміста .....	85
8.2 Розробка заходів з умов поліпшення охорони праці.....	88
8.3 Дослідження інформаційного навантаження на програміста.....	90
8.4 Висновки до розділу.....	93
9 ОСНОВНІ ВИСНОВКИ.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	96

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

Кафедра \_ КБПЗ \_ 2022 рік

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3



використовувати фінансові інвестиції на розвиток і супровід інформаційної системи підприємства (банку).

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

- Огляд існуючих систем бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- Дослідження системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- Програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

*Об'єктом дослідження* є процес бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

*Предметом дослідження* є методи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

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

- Удосконалено метод бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- Розроблено вітчизняний продукт бюджетування хмарної інформаційної системи для визначення ефективності її впровадження, який має більш широкі можливості, на відміну від існуючих аналогів.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

Робота апробована на LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>6</b>

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Основне призначення системи полягає в створенні методики аналізу ефективності компонентів інформаційних систем для збільшення загальної конкурентоспроможності підприємства (банку) на прикладі мережного середовища філії банку.

Для досягнення поставленої мети в роботі вирішувалися наступні задачі:

- аналіз банківської інформаційної системи;
- структурно-функціональне й імітаційне моделювання БІС філії;
- побудова профілю середовища відкритої БІС філії;
- розробка методики й алгоритму комплексної оцінки ефективності стеків протоколів локальних мереж;
- проведення теоретичного й практичного аналізу мережного середовища типової банківської філії на основі розробленої методики;
- визначення найбільш ефективна стека протоколів локальних мереж БІС філії й параметрів налаштування протоколів, що входять у даний стек.

У якості бажаних результатів є досягнення наступних положень:

- розробка моделі банківської інформаційної системи;
- розробка профілю середовища відкритої системи філії банку;
- методика й алгоритм комплексної оцінки ефективності мережного середовища банку;
- методика й результати тестування ефективності стеків протоколів локальних мереж.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

## 1.2 Область застосування

Ефективність побудови й використання корпоративних інформаційних систем стала надзвичайно актуальною задачею, особливо в умовах недостатнього фінансування інформаційних технологій на підприємствах.

Критеріями оцінки ефективності можуть служити зниження вартості реалізації інформаційної системи, відповідність поточним вимогам і вимогам найближчого часу, можливість і вартість подальшого розвитку й переходу до нових технологій. Основу інформаційної системи становить обчислювальна система, що включає такі компоненти, як кабельна мережа й активне мережне устаткування, комп'ютерне й периферійне устаткування, устаткування зберігання даних (бібліотеки), системне програмне забезпечення (операційні системи, системи керування базами даних), спеціальне ПЗ (системи моніторингу й керування мережами) і в деяких випадках прикладне ПЗ. Найпоширенішим підходом до проектування інформаційних систем у цей час є використання експертних оцінок. Відповідно до цього підходу фахівці в області обчислювальних засобів, активного мережного устаткування й кабельних мереж на підставі наявного в них досвіду й експертних оцінок здійснюють проектування обчислювальної системи, що забезпечує рішення конкретної задачі або класу задач. Цей підхід дозволяє мінімізувати витрати на етапі проектування, швидко оцінити вартість реалізації інформаційної системи. Однак рішення, отримані з використанням експертних оцінок, носять суб'єктивний характер, вимоги до устаткування й програмного забезпечення також грішать суб'єктивністю, як і оцінка гарантій працездатності й розвиненості запропонованого проекту системи.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

#### Спеціалізовані системи імітаційного моделювання обчислювальних мереж

Існують спеціальні, орієнтовані на моделювання обчислювальних мереж програмні системи, у яких процес створення моделі спрощений. Такі програмні системи самі генерують модель мережі на основі вихідних даних про її топологію й використовувані протоколи, про інтенсивності потоків запитів між комп'ютерами мережі, довжини ліній зв'язку, про типи використовуваного устаткування й додатків. Програмні системи моделювання можуть бути вузько спеціалізованими й досить універсальними, що дозволяють імітувати мережі всіляких типів. Якість результатів моделювання в значній мірі залежить від точності вихідних даних про мережу, переданих у систему імітаційного моделювання.

Програмні системи моделювання мереж – інструмент, що може пригодитися будь-якому адміністраторові корпоративної мережі, особливо при проектуванні нової мережі або внесенні кардинальних змін у вже існуючу. Продукти даної категорії дозволяють перевірити наслідку впровадження тих або інших рішень ще до оплати устаткування, що здобувається. Звичайно, більшість із цих програмних пакетів коштують досить дорого, але й можлива економія може бути теж досить відчутною.

Програми імітаційного моделювання мережі використовують у своїй роботі інформацію про просторове розташування мережі, числі вузлів, конфігурації зв'язків, швидкості передачі даних, використовувані протоколи і

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

типи устаткування, а також про виконувані в мережі додатки. Звичайно імітаційна модель будується не з нуля. Існують готові імітаційні моделі основних елементів мереж: найпоширеніших типів маршрутизаторів, каналів зв'язку, методів доступу, протоколів і т.п. Ці моделі окремих елементів мережі створюються на підставі різних даних: результатів тестових випробувань реальних пристроїв, аналізу принципів їхньої роботи, аналітичних співвідношень. У результаті створюється бібліотека типових елементів мережі, які можна набудувати за допомогою заздалегідь передбачених у моделях параметрів. Системи імітаційного моделювання звичайно включають також набір засобів для підготовки вихідних даних про досліджувану мережу – попередньої обробки даних про топологію мережі й обмірюваний трафік. Ці засоби можуть бути корисні, якщо моделюєма мережа являє собою варіант існуючої мережі і є можливість провести в ній виміру трафіку й інших параметрів, потрібних для моделювання. Крім того, система забезпечується засобами для статистичної обробки отриманих результатів моделювання.

Систем динамічного моделювання обчислювальної системи досить багато, вони розробляються в різних країнах. Удалося виявити такі системи, зроблені в Румунії й інших країнах, що не є лідерами комп'ютерно-інформаційної індустрії. Крім того, найчастіше розвинені системи діагностування встановленої обчислювальної системи (інтелектуальні кабельні тестери, сканери, аналізатори протоколів) також зараховують до систем моделювання, що не відповідає дійсності. Класифікуємо системи по двох зв'язаних критеріях: ціна й функціональні можливості. Як і слід було сподіватися, функціональні можливості систем моделювання жорстко пов'язані з їхньою ціною. Аналіз пропонованих на ринку систем показує, що динамічне моделювання обчислювальних систем – справа досить дорога. Хочете одержати реальну картину в обчислювальній системі – платіть гроші. Всі системи динамічного моделювання можуть бути розбиті на дві цінові категорії:

– Дешеві (сотні й тисячі доларів).

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– High-end (десятки тисяч доларів, у повному варіанті – сто й більше тисяч доларів).

На жаль, знайти системи середнього цінового діапазону не вдалося, однак багато які з них являють собою набір пакетів і розкид у ціні однієї й тої ж системи визначається комплектом поставки, тобто об'ємом виконуваних функцій. Дешеві системи відрізняються від дорогих тим, наскільки докладно вдається в них описати характеристики окремих частин моделюємої системи. Вони дозволяють одержати лише "прикидочні" результати, не дають статистичних характеристик і не надають можливості проведення докладного аналізу системи. Системи класу high-end дозволяють збирати вичерпну статистику по кожному з компонентів мережі при передачі даних по каналах зв'язку й проводити статистичну оцінку отриманих результатів. По функціональності системи моделювання, використовувані при дослідженні обчислювальних систем, можуть бути розбиті на два основних класи:

- Системи, що моделюють окремі елементи (компоненти) системи.
- Системи, що моделюють обчислювальну систему цілком.

### **Побудова пілотних проектів проєктованих мереж**

Якщо для завдання інформації про топологію мережі не потрібно мати реальну мережу, то для збору вихідних даних про інтенсивність джерел мережного трафіку можуть знадобитися виміри на пілотних мережах, що представляють собою натурну модель проєктованої мережі. Ці виміри можуть бути виконані різними засобами, у тому числі й за допомогою аналізаторів протоколів.

Крім одержання вихідних даних для імітаційного моделювання пілотна мережа може використовуватися для рішення самостійних важливих задач. Вона може дати відповіді на питання, що стосуються принципової працездатності того або іншого технічного рішення або сумісності устаткування. Натурні експерименти можуть зажадати значних матеріальних витрат, але вони компенсуються високою вірогідністю отриманих результатів.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Пілотна мережа повинна бути як можна більше схожа на ту мережу, що створюється, для вибору параметрів якої й створюється пілотна мережа. Для цього необхідно в першу чергу виділити ті особливості створюваної мережі, які можуть вплинути на її працездатність і продуктивність. Якщо є сумніви в сумісності продуктів різних виробників, наприклад, комутаторів, що підтримують віртуальні мережі або інші поки не стандартизовані можливості, то в пілотній мережі повинні перевірятися на сумісність саме ці пристрої й саме в тих режимах, які викликають найбільші сумніви.

Що ж стосується використання пілотній мережі для прогнозування пропускної здатності реальної мережі, то тут можливості цього виду моделювання досить обмежені. Сама по собі пілотна мережа навряд чи зможе дати гарну оцінку продуктивності мережі, що включає набагато більше вузлів підмереж і користувачів, тому що не ясний спосіб екстраполяції результатів, отриманих у невеликій мережі, на мережа набагато більших розмірів.

Тому пілотну мережа доцільно використовувати в цьому випадку разом з імітаційною моделлю, що може використовувати зразки трафіку, затримок і пропускної здатності пристроїв, отриманих у пілотній мережі, для завдання характеристик моделей частин реальної мережі. Потім, ці приватні моделі можуть бути об'єднані в повну модель створюваної мережі, робота якої буде імітуватися.

### **NetMaker XA**

Обчислювальне ядро моделювання, використовуване в NetMaker XA від Make Systems, – одне з найбільш потужних на ринку, і це зіграло немаловажну роль у тому, що продукт зарекомендував себе настільки добре. За що не візьмися – все працює в повній відповідності з описами. У нас не виникло ніяких проблем ні з моделюванням спроектованої нами невеликої мережі, ні з удосконаленням системи, наведеної виробником як приклад. Крім того, генеруємі програмою звіти містили всю необхідну інформацію. Головні недоліки NetMaker XA – необхідність серйозного навчання користувача й висока вартість. Якщо до

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

ціни базової конфігурації виробу додати вартість додаткових модулів, вийде досить значна сума.

Оснoву продукту становлять модулі Visualizer, Planner і Designer. Кожен з них виконує якусь одну функцію; щоб змоделювати роботу мережі, необхідні всі три. Visualizer служить для одержання інформації про мережу і її перегляд. У його склад входять SNMP-модулі автоматичного розпізнавання, які опитують мережні пристрої й створюють відповідні їм об'єкти. Інформацію про ці об'єкти можна потім редагувати за допомогою Visualizer.

Planner – це бібліотека пристроїв, що допомагає проаналізувати, що вийде при установці в мережі нового пристрою (наприклад, додаткового маршрутизатора). Make Systems поставляє вбудовуються модулі, (plug-in), що містять об'єкти з даними про продукти різних виробників. У таких об'єктах утримується повний опис різних моделей пристроїв (від числа мережних інтерфейсів до типу процесора); вся інформація засвідчується виробником. За допомогою Planner користувач може самостійно будувати свої власні об'єкти для опису мережних пристроїв і каналів зв'язку, не включених у бібліотеку. Designer потрібний для побудови схем мереж. Даний засіб дозволяє легко й швидко створювати моделі й аналізувати альтернативи. Якщо користуватися ним разом з Planner, можна одержувати інформацію про те, як буде працювати мережа заданої конфігурації.

Якщо потрібно піти трохи далі, прийдеться придбати ще три модулі: Accountant, Interpreter і Analyzer. До складу Accountant входить тарифікаційна база даних; цей модуль допомагає проаналізувати витрати, пов'язані з використанням тих або інших мереж загального доступу. Нам здався дуже корисним модуль Interpreter, призначений для збору даних від засобів аналізу трафіку. Потім дані автоматично імпортувалися в нашу модель, що дозволяло використовувати їх майже в режимі реального часу, а не будувати гіпотези щодо роботи мережі. Нарешті, Analyzer і призначений для нього вбудовується модуль, що, "виживаності" допомагають розробляти плани відновлення після аварій, а також

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

домагатися того, щоб жодна несправність (після її локалізації) не могла привести до відмови мережі в цілому.

Коштує все це багатство функцій дуже дорого – від 37 тис. дол. за базовий комплект плюс доплати за модули, що вбудовуються. Тому, хто захоче придбати модулі Accountant, Interpreter і Analyzer, прийде розщедритися ще на 30 тис. дол.

Установити NetMaker ХА можна тільки на SPARCstation від Sun Microsystems. До цього треба додати вартість навчання, оскільки без нього у вас просто нічого не вийде. В Make Systems усвідомлюють, що користуватися їхнім продуктом не так-то просто; під час випробувань до нас надіслали фахівця, що навчив нас роботі з пакетом.

Проте для власника великої мережі на кілька тисяч вузлів NetMaker ХА – те що потрібно.

#### **SES/Strategizer – альтернативний підхід**

Той, хто не збирається включати можливість росту у свою модель мережі, цілком задовольниться значно менш дорогим продуктом SES/Strategizer від Scientific and Engineering Software (ціна 9995 дол.). SES/Strategizer пропраховує моделі дуже швидко. Ми встановили цей продукт на робочій станції на базі Pentium IV, і всього за 2 мс. програма розрахувала, як буде працювати досить складна мережа протягом 24 годин. Можна також збирати тонкі статистичні дані про якийсь один конкретний елемент моделі, наприклад, стежити за ступенем завантаження центрального процесора з розбивкою по процесах, користувачам і моделям поведіння.

Один із серйозних недоліків програми – необхідність перезапуску моделі при кожному внесенні яких-небудь змін. Інші продукти дозволяють вставляти в модель різні змінні (наприклад, що враховують ріст мережі); у результаті можна випробувати кілька варіантів у ході одного прогону програми.

Установка не викликала ніяких утруднень.

Як та інші пакети, SES/Strategizer дозволяє без праці задавати й модифікувати значення параметрів, таких як пропускна здатність. Крім того,

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

продукт видає запит на підтвердження ("Застосувати" або "Скасування"), якщо користувач намагається закрити діалогове вікно, клацнувши мишею на хрестик у правому верхньому куті. Така функція не передбачена в інших продуктах, що незручно, оскільки з ними ніколи не можна бути впевненим, яке дія буде почато за замовчуванням.

І все-таки окремі сторони SES/Strategizer мають потребу в доробці. Наприклад, для перегляду результатів моделювання на тому же ПК, де працює сама програма, потрібно запустити Microsoft Excel; дані він повинен брати зі створених SES/Strategizer файлів, де для поділу числових полів використовуються знаки табуляції. Якщо Excel не встановлений, користувач одержує дивне повідомлення про помилку, що вказує на зовсім іншу причину збою. Треба просто інформувати користувача, що йому варто встановити Excel, або забезпечувати можливість перегляду засобами який-небудь іншого додатка.

Розходження між SES/Strategizer і Predictor аж ніяк не так великі, як дозволяє припустити різниця в їхніх цінах (19 тис. дол.). Predictor гарний тим, що розрахунки можуть охоплювати тривалий період існування мережі, а користувач – урахувати ріст трафіку із часом. По частині функцій SES/Strategizer відстає зовсім не так сильно – користувачеві просто прийдеться упокоритися з необхідністю постійно прораховувати модель заново. І все-таки NetMaker ХА залишається королем. Він – для тих, хто може розщедритися на чималу суму й хоче роздобути кращий засіб моделювання мережі.

Основні вимоги, пропоновані до систем моделювання обчислювальних систем:

- відсутність необхідності програмування;
- можливість імпорту інформації з існуючих систем керування мережами й засобів моніторингу;
- наявність розширеної бібліотеки об'єктів; інтуїтивно-зрозумілий інтерфейс;
- просте налаштування на об'єкти реального миру;

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- гнучка система побудови сценаріїв моделювання;
- зручне подання результатів моделювання;
- анімація процесу моделювання;
- автоматичний контроль моделі на внутрішню несуперечність.

### **Використання моделювання для оптимізації продуктивності мережі**

Аналізатори протоколів незамінні для дослідження реальних мереж, але вони не дозволяють одержувати кількісні оцінки характеристик для ще не існуючих мереж, що перебувають у стадії проектування. У цих випадках проектувальники можуть використовувати засоби моделювання, за допомогою яких розробляються моделі, що відтворюють інформаційні процеси, що протікають у мережах.

### **Методи аналітичного, імітаційного й натурального моделювання**

Моделювання являє собою потужний метод наукового пізнання, при використанні якого досліджуваній об'єкт заміняється більш простим об'єктом, названим моделлю. Основними різновидами процесу моделювання можна вважати два його види – математичне й фізичне моделювання. При фізичному (натурному) моделюванні досліджувана система заміняється відповідною їй іншою матеріальною системою, що відтворює властивості досліджуваної системи зі збереженням їхньої фізичної природи. Прикладом цього виду моделювання може служити пілотна мережа, за допомогою якої вивчається принципова можливість побудови мережі на основі тих або інших комп'ютерів, комунікаційних пристроїв, операційних систем і додатків.

Можливості фізичного моделювання досить обмежені. Воно дозволяє вирішувати окремі задачі при завданні невеликої кількості сполучень досліджуваних параметрів системи. Дійсно, при натурному моделюванні обчислювальної мережі практично неможливо перевірити її роботу для варіантів з використанням різних типів комунікаційних пристроїв – маршрутизаторів, комутаторів і т.п. Перевірка на практиці біля десятка різних типів маршрутизаторів зв'язана не тільки з більшими зусиллями й часовими витратами,

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>16</b>

але й із чималими матеріальними витратами. Але навіть і в тих випадках, коли при оптимізації мережі змінюються не типи пристроїв і операційних систем, а тільки їхні параметри, проведення експериментів у реальному масштабі часу для величезної кількості всіляких сполучень цих параметрів практично неможливо за доступне для огляду час. Навіть проста зміна максимального розміру пакета в якому-небудь протоколі вимагає переконфігурування операційної системи в сотнях комп'ютерів мережі, що жадає від адміністратора мережі проведення дуже великої роботи. Тому, при оптимізації мереж у багатьох випадках кращим виявляється використання математичного моделювання.

Математична модель являє собою сукупність співвідношень (формул, рівнянь, нерівностей, логічних умов), що визначають процес зміни стану системи залежно від її параметрів, вхідних сигналів, початкових умов і часу.

Особливим класом математичних моделей є імітаційні моделі. Такі моделі являють собою комп'ютерну програму, що крок за кроком відтворює події, що відбуваються в реальній системі. Стосовно до обчислювальних мереж їхні імітаційні моделі відтворюють процеси генерації повідомлень додатками, розбивка повідомлень на пакети й кадри певних протоколів, затримки, пов'язані з обробкою повідомлень, пакетів і кадрів усередині операційної системи, процес одержання доступу комп'ютером до поділюваного мережного середовища, процес обробки вступників пакетів маршрутизатором і т.д. При імітаційному моделюванні мережі не потрібно здобувати дороге устаткування – його робота імітується програмами, що досить точно відтворюють всі основні особливості й параметри такого устаткування.

Перевагою імітаційних моделей є можливість підміни процесу зміни подій у досліджуваній системі в реальному масштабі часу на прискорений процес зміни подій у темпі роботи програми. У результаті за кілька мінут можна відтворити роботу мережі протягом декількох днів, що дає можливість оцінити роботу мережі в широкому діапазоні варіюємих параметрів. Результатом роботи імітаційної моделі є зібрані в ході спостереження за подіями, що протікають,

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

статистичні дані про найбільш важливі характеристики мережі: часах реакції, коефіцієнтах використання каналів і вузлів, імовірності втрат пакетів і т.п.

Існують спеціальні мови імітаційного моделювання, які полегшують процес створення програмної моделі в порівнянні з використанням універсальних мов програмування. Прикладами мов імітаційного моделювання можуть служити такі мови, як SIMULA, GPSS, SIMDIS.

Існують також системи імітаційного моделювання, які орієнтовані на вузький клас досліджуваних систем і дозволяють будувати моделі без програмування. Подібні системи для обчислювальних мереж розглядаються нижче.

### **Моделі теорії масового обслуговування**

Використовувані в цей час у локальних мережах протоколи канального рівня використовують методи доступу до середовища, засновані на її спільному використанні декількома вузлами за рахунок поділу в часі. У цьому випадку, як і у всіх випадках поділу ресурсів з випадковим потоком запитів, можуть виникати черги. Для опису цього процесу звичайно використовуються моделі теорії масового обслуговування.

Механізм поділу середовища протоколу Ethernet спрощено описується найпростішою моделлю типу М/М/1 – одноканальною моделлю з пуассоновським потоком заявок і показовим законом розподілу часу обслуговування. Вона добре описує процес обробки випадково вступників заявок на обслуговування системами з одним обслуговуючим приладом з випадковим часом обслуговування й буфером для зберігання вступних заявок на час, поки обслуговуючий прилад зайнятий виконанням іншої заявки.

Передавальне середовище Ethernet представлене в цій моделі обслуговуючим приладом, а пакети відповідають заявкам.

У теорії масового обслуговування для даної моделі отримані наступні результати: середній час очікування заявки в черзі (час очікування пакетом доступу до середовища)  $W$  дорівнює:

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

$$W = \frac{\rho b}{1 - \rho} \quad (2.1)$$

Уведемо позначення:

$l$  – інтенсивність надходження заявок, у цьому випадку це середнє число пакетів, що претендують на передачу в середовищі в одиницю часу,

$b$  – середній час обслуговування заявки (без обліку часу очікування обслуговування), тобто середній час передачі пакета в середовищі з урахуванням паузи між пакетами в 9.6 мкс,

$r$  – коефіцієнт завантаження обслуговуючого приладу, у цьому випадку це коефіцієнт використання середовища,  $r = lb$ .

## 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 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.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19





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 звільнюються з допомоги 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.

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, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

## 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 Cmake.
- 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.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## 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 TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- 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.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## 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 випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

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

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

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

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

д) розробити рекомендації по організаційних та методичних заходах, які

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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.

Кафедра \_ КБПЗ \_ 2022 рік

					VKPM-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

### 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

#### 3.1 Опис функціонування системи

Рішення задачі оцінки й підвищення ефективності функціонування БІС

Математична постановка задачі підвищення ефективності функціонування БІС

Приведемо особливості БІС:

- архітектуру мережі БІС (рисунок 3.1);
- основні задачі й функції, розв'язувані на різних рівнях ієрархії (таблиця 3.1);
- типи даних, переданих у мережах (таблиця 3.2).

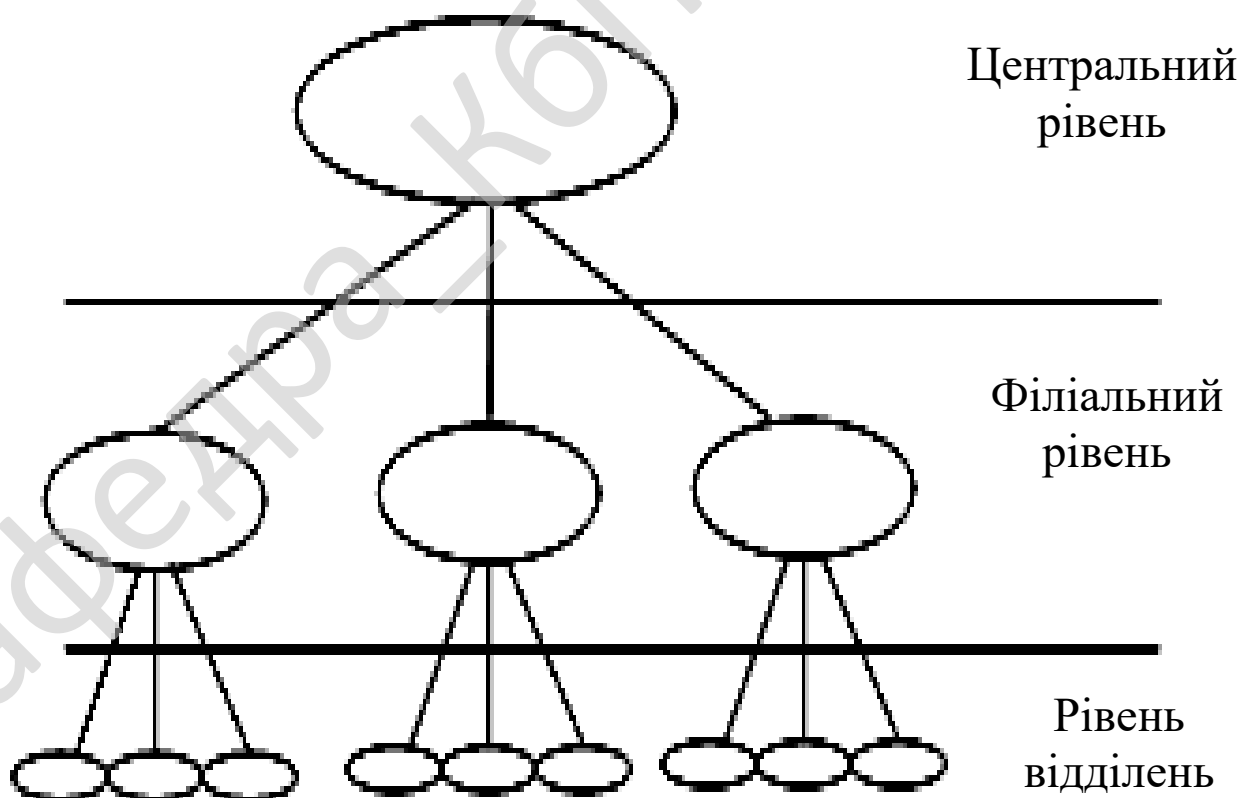


Рисунок 3.1 – Модель БІС



$$\Psi = \{U, D, O, Q\}, \quad (3.1)$$

де:

–  $U = \{u_i\}, i > 2$  – хости, підключених до локальної мережі підприємства (банку).

–  $D = \{d_j\}, j \geq 1$  – типи даних переданих по мережі.

–  $O = \{o_k\}, k > 2$  – типи мережних операцій.

–  $Q = \{q\}, l > 2$  – можливі стеки протоколів, що утворюють мережне середовище.

–  $q = \{x_1, x_2, \dots, x_m\}, 1 \leq m \leq 7$ , – набір протоколів різних рівнів, що утворюють стек.

З огляду на, що замовник інформаційної системи висуває вимоги з використанням різних альтернативних показників, то необхідно використовувати вектор показників ефективності  $\omega$ :

$$\omega = \{\omega_1, \omega_2, \dots, \omega_5\}, \quad (3.2)$$

де:

–  $\omega = F(\Psi, q)$  – вектор-функція й де кожний приватний показник може бути представлений у вигляді залежності  $\omega_h = f_h(\Psi, q)$ ;

–  $h = 1 \dots 5$  для різних стеків протоколів  $Q$ .

Таким чином, ми маємо багатокритеріальну задачу, для рішення якої можуть бути використані лексикографічні методи, методи головного й узагальненого показника, методи послідовних поступок і т.д. Аналіз перерахованих методів показав, що для рішення даної задачі доцільно використовувати метод узагальненого показника, що має наступний вид:

$$S = \sum_{h=1}^5 \alpha_h \omega_h^*, \quad (3.3)$$

$$\sum_{h=1}^5 \alpha_h = 1, \quad (3.4)$$

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

де:

- $\alpha_h$  – ваговий коефіцієнт h-го частки показника;
- $\omega_h^*$  – значення наведеного до однорідної величини показника ефективності.

У результаті багатокритеріальна задача зводиться до однокритеріальної і буде мати наступну постановку: визначити  $q_{opt} \in Q$ , при якому узагальнений показник ефективності приймає максимальне значення:

$$S_{\max} = \sum_{h=1}^5 \alpha_h \omega_h^*(\Psi, q_{opt}) \quad (3.5)$$

Таким чином, процедура рішення задачі дослідження зводиться до алгоритму, представленому на рисунку 4.1.

У результаті структурно-функціонального моделювання БІС філії виділені основні банківські функції й формовані ними мережні потоки. Для оцінки формованого даними функціями мережного трафіку, вибору можливих технічних рішень їхньої реалізації, а також оцінці масштабованості обраних рішень расстрим процедуру побудови профілю середовища відкритої БІС філії.

### **Профіль середовища відкритої БІС філії**

Опишемо побудову профілю середовища відкритої БІС філії відповідно до ТВС. При цьому до БІС застосовні основні вимоги відкритих систем: уніфікація обміну даними між компонентами БІС, забезпечення переносимості рішень між різними системами, а також забезпечення єдиного інтерфейсу для користувачів у різних системах. Даний профіль визначає набір стандартів і протоколів для локальних мереж, за допомогою яких варто реалізувати необхідний набір банківських функцій.

Відповідно до керівництва по проектуванню профілів середовища відкритої системи, проектування проводилося наступними етапами:

- аналіз вимог – декомпозиція функціональних служб на служби інформаційних систем;

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– логічний проект – визначення зв'язку служб інформаційних технологій і інформаційних систем;

– фізичний проект – визначення базових стандартів, які можуть бути використані при побудові мережної структури локальної мережі.

На основі результатів фізичного проекту були проведені декомпозиції основних функцій банківської філії:

– по типах трафіку – визначення взаємозв'язку розглянутих функцій і типів даних, переданих у мережі (таблиця 3.3);

– по вимогах до бізнес-системи – знаходження можливих стеків протоколів локальних мереж для забезпечення даних функцій з урахуванням навантаження (таблиця 3.4).

Таблиця 3.3 – Таблиця розрахунку об'єму трафіку різного типу з урахуванням функціональних служб

Тип трафіку	Функції						Сумарний трафік по типах
	Офісна автоматизація	Системи керування й контролю	Системи фінансового прогнозування, аналізу й керування	Касові системи й системи обслуговування клієнтів	Системи на основі WEB і Банк-Клієнт	Автоматизовані банківські системи	
I	-	50 Мб	50 Мб	200 Мб	3000 Мб	5000 Мб	8300 Мб
II	700 Мб	50 Мб	100 Мб	-	2000 Мб	-	2850 Мб
III	300 Мб	-	50 Мб	-	-	-	350 Мб

Таблиця 3.4 – Матриця взаємозв'язку функціональних служб і

технологічних вимог

Вимоги до бізнес-системі	Функції					
	Офісна автоматизація	Системи керування й контролю	Системи фінансового прогнозування, аналізу й управління	Касові системи й системи обслуж. клієнтів	Системи на основі WEB і Банк-Клієнт	Автоматизовані банківські системи
Число користувачів	20	2	5	50	1000	200
Число одночасних підключень	10	1	3	50	100	50
Об'єм даних у день	1Гб	100Мб	200Мб	200Мб	5Гб	5Гб
Транспортні й мережні протоколи	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS
Канальні протоколи	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring
Протоколи фізичного рівня	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5

Отримані таблиці можуть із успіхом застосовуватися при оцінці масштабування філіальної мережі. Для вибору оптимального стека протоколів з базових стандартів, отриманих на етапі фізичного проекту, надалі пропонується методика аналізу ефективності мережного середовища підприємства (банку).

### **Методика аналізу ефективності мережного середовища підприємства (банку)**

Розробимо методику аналізу ефективності мережного середовища підприємства (банку). На початковому етапі був визначений вектор показників ефективності  $\omega$ , що відбивають основні аспекти поставленої задачі. Вибір показників був проведений методом експертних оцінок (метод анкетного опитування керівників підрозділів різних рівнів і напрямків). Для одержання комплексної оцінки були виділені показники різної природи (економічні, технічні й т.п):

- надмірність – показник ефективності використання пропускну здатності мережі;
- продуктивність – показник ефективності роботи мережі при різних навантаженнях;
- навантаження на устаткування – показник складності реалізації протоколів, що входять у стек;
- вартість рішення – показник ефективності фінансових вкладень;
- перспективність – показник можливості масштабування й розвитку даного рішення.

Розроблена методика розрахунку узагальненого показника ефективності  $S$  включає:

1. Вибір стеків протоколів, для яких проводяться дослідження  $Q$ .
2. Послідовне знаходження значень кожного приватного показника для кожного стека протоколів  $\{\omega\}$  (таблиця 3.5).

3. Приведення отриманих результатів по кожному показнику до безрозмірної величини від 1 до 4 (шляхом порівняння результатів на лінійній шкалі)  $\{\omega\} \rightarrow \{\omega^*\}$ .

4. Знаходження значень узагальненого показника ефективності  $\{S\}$ .

5. Побудова таблиці для знаходження найбільш оптимальна стека  $q_{opt}$ .

Перші три показники є технічними, четвертий відбиває економічну ефективність. Методика розрахунку показника перспективності стосовно до даної предметної області вводиться вперше й відбиває можливості подальшого розвитку й застосування розглянутих технологій. При розрахунку значення узагальненого ефективності кожний показник уводиться з обліком його вагового коефіцієнта. Знаходження вагових коефіцієнтів є складною задачею. Для її рішення в роботі застосовувалися методи експертних оцінок (метод анкетного опитування фахівців інформаційних технологій в області локальних мереж).

Для визначення значень перших трьох показників при теоретичному дослідженні застосовувалися методи імовірнісного моделювання, а при експериментальному дослідженні – прямих і непрямих вимірів. Вартість рішення обчислювалася на основі маркетингового аналізу. Для визначення показника перспективності поряд з маркетинговим аналізом також використовувався аналіз рівня стандартизації розглянутих рішень.

Таблиця 3.5 – Методики оцінки показників ефективності стеків протоколів

Показники	Методики оцінки
Надмірність	$I = \frac{V_{служ.}}{V_{заг.}} = \frac{V_{заг.} - V_{кор.}}{V_{заг.}},$ <p>де:</p> <ul style="list-style-type: none"> <li>– <math>V_{служ.}</math> – об'єм службової інформації;</li> <li>– <math>V_{заг.}</math> – загальний об'єм переданих даних;</li> <li>– <math>V_{кор.}</math> – об'єм корисного навантаження.</li> </ul>

Продовження таблиці 3.5

Показники	Методики оцінки
Продуктивність	$C = T / V ,$ <p>де:</p> <ul style="list-style-type: none"> <li>– <math>V</math> – об'єм переданої інформації.</li> </ul> $T = \frac{\sum_1^N T_i}{N} ,$ <p>де:</p> <ul style="list-style-type: none"> <li>– <math>T</math> – час передачі інформації, обмірюваний на рівні додатку;</li> <li>– <math>N</math> – число станцій.</li> </ul>
Навантаження на устаткування	$L = (P^* + M^*) / 2$ – значення навантаження на устаткування з розрахунку наведених безрозмірних значень $P$ і $M$ . $P = P_{\text{експ.}} - P_{\text{поч.}} ,$ <p>де:</p> <ul style="list-style-type: none"> <li>– <math>P</math> – середнє значення лічильника навантаження на процесор;</li> <li>– <math>P_{\text{експ.}}</math> – значення лічильника навантаження на процесор, обмірюване в експерименті при роботі з мережі;</li> <li>– <math>P_{\text{поч.}}</math> – значення лічильника навантаження на процесор, обмірюване в експерименті при роботі локально.</li> </ul> $M = M_{\text{експ.}} - M_{\text{поч.}}$ – середнє значення лічильника використання пам'яті, <p>де:</p> <ul style="list-style-type: none"> <li>– <math>M_{\text{експ.}}</math> – значення лічильника використання пам'яті, обмірюване в експерименті при роботі з мережі;</li> <li>– <math>M_{\text{поч.}}</math> – значення лічильника використання пам'яті, обмірюване в експерименті при роботі локально.</li> </ul>

Продовження таблиці 3.5

Показники	Методики оцінки
Вартість рішення	$\Phi = \Phi_{CA} + \Phi_{H/S} + \Phi_{наст.},$ <p>де:</p> <ul style="list-style-type: none"> <li>– <math>\Phi</math> – вартість одного мережного підключення;</li> <li>– <math>\Phi_{NA}</math> – вартість мережного адаптера;</li> <li>– <math>\Phi_{H/S}</math> – вартість одного порту концентратора або комутатора;</li> <li>– <math>\Phi_{наст.}</math> – вартість налаштування одного мережного підключення.</li> </ul>
Перспективність	$A_{комп.i} = \frac{1}{4} \sum_1^4 A_j$ – значення перспективності для кожного протоколу в стеці. $A = \frac{1}{n} \sum_1^n A_{комп.i}$ – загальна перспективність стека, <p>де:</p> <ul style="list-style-type: none"> <li>– <math>n</math> – число протоколів у стеці</li> <li>1. <math>A_1</math> – кількість компаній виробників кінцевих рішень: одна (1), менш трьох (2), менш п'яти (3), більше п'яти (4);</li> <li>2. <math>A_2</math> – убудована підтримка в сучасні операційні системи: одна (1), дві (2), три (3), більше трьох (4);</li> <li>3. <math>A_3</math> – відношення до стандартизації: немає стандартів (1), готується до стандартизації (2), стандарт де-факто (3), стандарт де-юре (4);</li> <li>4. <math>A_4</math> – сфера застосування протоколу: скорочується (1), постійна (2), росте (3), швидко росте (4);</li> </ul>
Узагальнений	$S = \sum_{h=1}^5 \alpha_h \omega_h^*, \quad \sum_{h=1}^5 \alpha_h = 1,$ <p>де</p> <ul style="list-style-type: none"> <li>– <math>\alpha_h</math> – ваговий коефіцієнт <math>h</math>-го частки показника,</li> <li>– <math>\omega_h^*</math> – значення наведеного до однорідної величини показника ефективності.</li> </ul>

Для розрахунку узагальненого показника значення кожного часного показника ефективності для всіх стеків протоколів необхідно привести до безрозмірної величини. Для зручності всі показники перетворюються в числа (використання єдиної порядкової шкали)  $\{\omega\} \rightarrow \{\omega^*\}$ , що відповідають місцю на шкалі розподілу від самого гіршого результату до найкращому: поганого (1), нижче середнього (2), середній (3), гарний (4). Приклад даного розрахунку наведений для показника вартості рішення на рисунку 6.

Таким чином, унікальність запропонованої методики полягає у виборі різнобічних показників ефективності, що забезпечують комплексний аналіз, алгоритмі розрахунку значень показників, а також методиці розрахунку узагальненого показника.

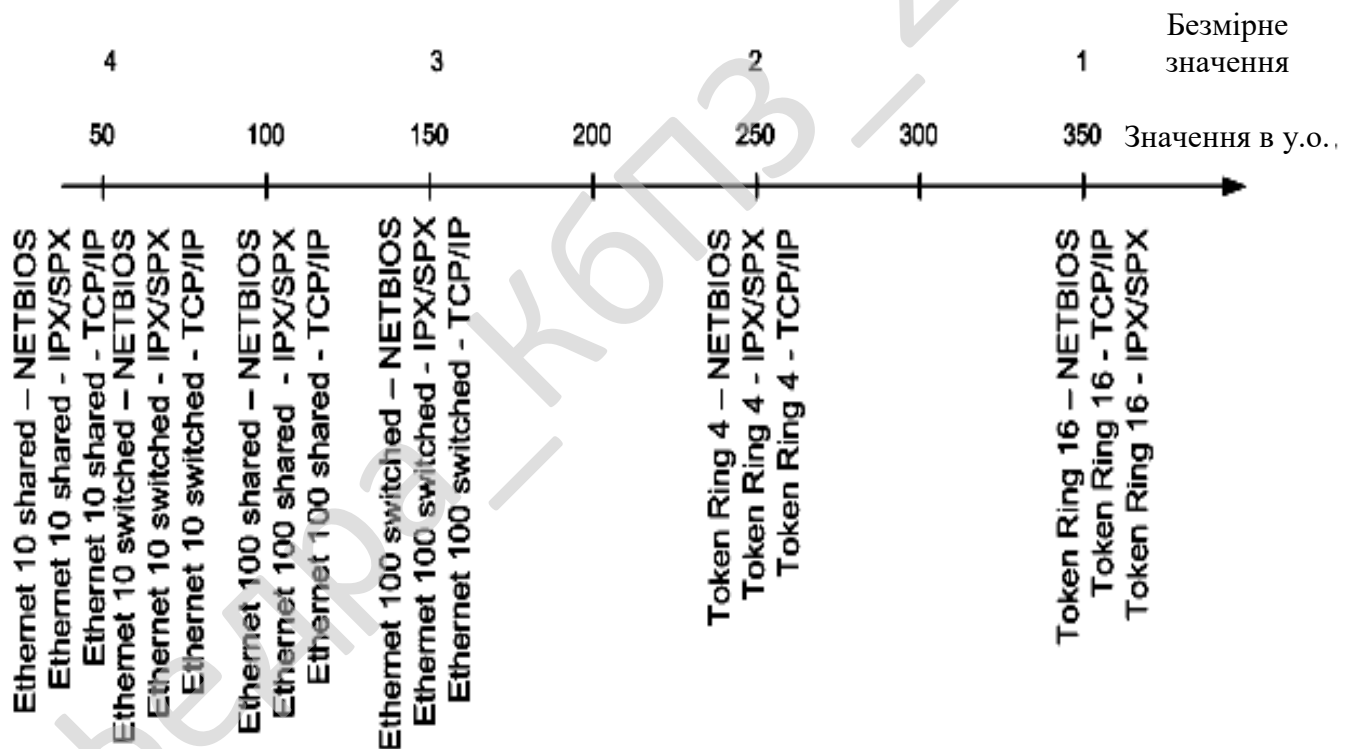


Рисунок 3.2 – Приведення показника вартості рішення до безрозмірної величини

Для перевірки розробленої методики оцінки ефективності стеків протоколів локальних мереж проведемо теоретичне й практичне дослідження сучасних протоколів локальних мереж.

### **Дослідження стеків протоколів локальних мереж**

Дослідимо стеки протоколів локальних мереж типових банківських філій з використанням методики, запропонованої вище. Набір протоколів, обраний для дослідження, характерний саме для локальних мереж банківських філій, і при аналізі інших систем може відрізнитися.

На першому етапі порівняння проводилося для кожного окремого протоколу з урахуванням особливостей філіального мережного трафіку. Це дозволило оцінити й знайти оптимальні параметри кожного протоколу, що входить у стек.

Другим етапом було проведення комплексного аналізу стеків протоколів і знаходження найбільш ефективного рішення для типової банківської філії.

Результати досліджень представлені в таблиці 3.6.

Як показав аналіз існуючих банківських мереж, найбільш ефективними стеками протоколів для локальної мережі банківської філії є: NETBIOS + Ethernet 10Base switched і TCP/IP + Ethernet 10Base switched. Перший набір відрізняє більше високим рівнем продуктивності, другий – низкою вартістю й високою перспективністю.

Таким чином, результати дослідження показали, що перехід до технологій локальних мереж на основі Ethernet 100 Мбіт/с і вище на справжньому етапі розвитку вітчизняного банківського бізнесу є невиправданим, тому що приріст продуктивності буде нівельований високою вартістю модернізації технічних компонентів у філіях. Для перевірки отриманих результатів у п'ятому розділі проведені практичні дослідження сучасних протоколів локальних мереж з обліком основних банківських функцій.

Таблиця 3.6 – Оцінка ефективності протоколів локальних мереж

№ п./ п.	Стек протоколів	Показники					Підсумкова оцінка
		надмірність*	продуктивність*	навантаження на устаткування*	вартість рішення*	перспективність*	
1.	Ethernet 10 shared – NETBIOS**	2	2	4	4	2	2,8
2.	Ethernet 10 shared – TCP/IP	1	2	2	4	3	2,4
3.	Ethernet 10 shared – IPX/SPX	2	2	3	4	2	2,6
4.	Ethernet 10 switched – NETBIOS	4	4	4	3	3	3,6
5.	Ethernet 10 switched – TCP/IP	3	3	3	3	4	3,2
6.	Ethernet 10 switched – IPX/SPX	3	3	3	3	2	2,8
7.	Ethernet 100 shared – NETBIOS	3	3	3	3	2	2,8
8.	Ethernet 100 shared – TCP/IP	3	2	2	3	3	2,6
9.	Ethernet 100 shared – IPX/SPX	3	2	3	3	2	2,6
10.	Ethernet 100 switched – NETBIOS	4	4	3	2	2	3
11.	Ethernet 100 switched – TCP/IP	3	4	2	2	4	3
12.	Ethernet 100 switched – IPX/SPX	3	3	2	2	4	2,8
13.	Token Ring 4 – NETBIOS	3	3	4	1	2	2,6
14.	Token Ring 4 – TCP/IP	2	3	3	1	3	2,4
15.	Token Ring 4 – IPX/SPX	2	3	3	1	2	2,2
16.	Token Ring 16 – NETBIOS	4	4	4	1	2	3
17.	Token Ring 16 – TCP/IP	3	3	2	1	3	2,4
18.	Token Ring 16 – IPX/SPX	3	3	3	1	2	2,4

\* – у даній таблиці зазначені показники, наведені до безрозмірної величини.

\*\* – для позначення протоколів і технологій, що входять у стек застосовуються загальноприйняті позначення.

## Експериментальне дослідження ефективності протоколів локальних мереж

Представимо результати експериментальних досліджень ефективності стеків протоколів локальних мереж.

При проведенні тестування визначалися такі технічні показники, як: продуктивність і навантаження на устаткування. Як об'єкти тестування використовувалися практично всі відомі набори протоколів локальних мереж, використовуваних у банківських філіях. Однак, запропонована методика застосовна до будь-яких інших протоколів, використовуваним у сучасних локальних мережах.

Для проведення експериментальних досліджень була розроблена методика функціонального тестування, заснована на імітації роботи типових мережних додатків (імітаційне моделювання).

Суть даної методики складається у вимірі показників ефективності для різних стеків протоколів залежно від числа одночасно працюючих робочих станцій. При цьому вимір проводиться на рівні додатків. Алгоритм роботи даної методики представлений на рисунку 4.2.

Для проведення тестування були обрані наступні мережні операції *O* типової банківської філії:

1. Копіювання по мережі файлу великого розміру (10Мб) по протоколі SMB (Server Message Block).
2. Копіювання по мережі 100 файлів невеликого розміру (по 2Кб) по протоколі SMB (Server Message Block).
3. Виконання 1000 транзакцій (загальний об'єм 20Кб) до бази даних по протоколі ODBC.

При виконанні даних операцій були обмірювані наступні показники:

- 1) час виконання мережної операції на клієнтській системі;
- 2) навантаження на процесор центральної системи (% Processor Time);

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

3) використання оперативної пам'яті центральної системи (% Committed Bytes In Use).

Відповідно до методики був розгорнутий тестовий стенд, що включає сервер центральної системи, 6 клієнтських систем, вузли реалізації мережного середовища (концентратори Ethernet, комутатори Ethernet, модулі множинного доступу Token Ring), мережні адаптери Ethernet і Token Ring, а також необхідна кількість кабелю UTP п'ятої категорії.

Для автоматизації тестування був розроблений пакет програм мовою VBScript, що імітують роботу кінцевих користувачів на стороні клієнтських систем і здійснюють збір даних і попередню обробку результатів вимірів на стороні центральної системи.

Отримані результати вимірів були оброблені відповідно до методи комплексної оцінки ефективності, розглянутої вище. Результати оцінки по двох технічних показниках представлені в таблиці 3.7.

Практичне впровадження запропонованого стека дозволило підвищити ефективність функціонування локальних мереж даних фінансових установ приблизно на 15% при мінімальних фінансових витратах. Найкращі результати в цих організаціях були досягнуті при наступних параметрах налаштування протоколів: розмір кадру Ethernet – 512 байт, розмір датаграмми NETBIOS – 496 байт. На сучасному етапі розвитку технологій й ринку мережних рішень перехід до більше швидкісних технологій (Ethernet 100 Мбіт/с і вище) став би не вигідним вкладенням коштів, тому що інвестиції в модернізацію мережних компонентів локальної мережі не відбилися б на значному підвищенні ефективності роботи мережі в цілому.

Таблиця 3.7 – Результати експериментальної оцінки ефективності протоколів локальних мереж

Стек протоколів	Продуктивність*				Навантаження на устаткування*				ΣΣ
	Б.Ф.	М.Ф.	Б.Д.	Σ	Б.Ф.	М.Ф.	Б.Д.	Σ	
Ethernet 10 shared-NetBIOS	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 shared-TCP/IP	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 shared-IPX/SPX	3	3	3	3,0	4	3	4	3,7	3,3
Ethernet 10 switched-NetBIOS	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 switched-TCP/IP	3	3	3	3,0	4	4	3	3,7	3,3
Ethernet 10 switched-IPX/SPX	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 100 switched-NetBIOS	4	4	3	3,7	2	3	3	2,7	3,2
Ethernet 100 switched-TCP/IP	4	4	3	3,7	2	4	4	3,3	3,5
Ethernet 100 switched-IPX/SPX	4	4	3	3,7	2	4	3	3,0	3,3
Token Ring 4-NetBIOS	2	2	3	2,3	3	4	3	3,3	2,8
Token Ring 4-TCP/IP	2	3	3	2,7	4	3	3	3,3	3,0
Token Ring 4-IPX/SPX	2	2	2	2,0	3	3	3	3,0	2,5
Token Ring 16-NetBIOS	3	3	4	3,3	2	3	3	2,7	3,0
Token Ring 16-TCP/IP	3	3	4	3,3	2	2	3	2,3	2,8
Token Ring 16-IPX/SPX	3	2	3	2,7	2	3	3	2,7	2,7

### 3.2 Розробка структурної схеми

На початковому етапі дослідження було виконано структурно-функціональне моделювання розподілу елементів БІС і інформаційні потоки між ними на різних рівнях. 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 (рисунок 3.3)

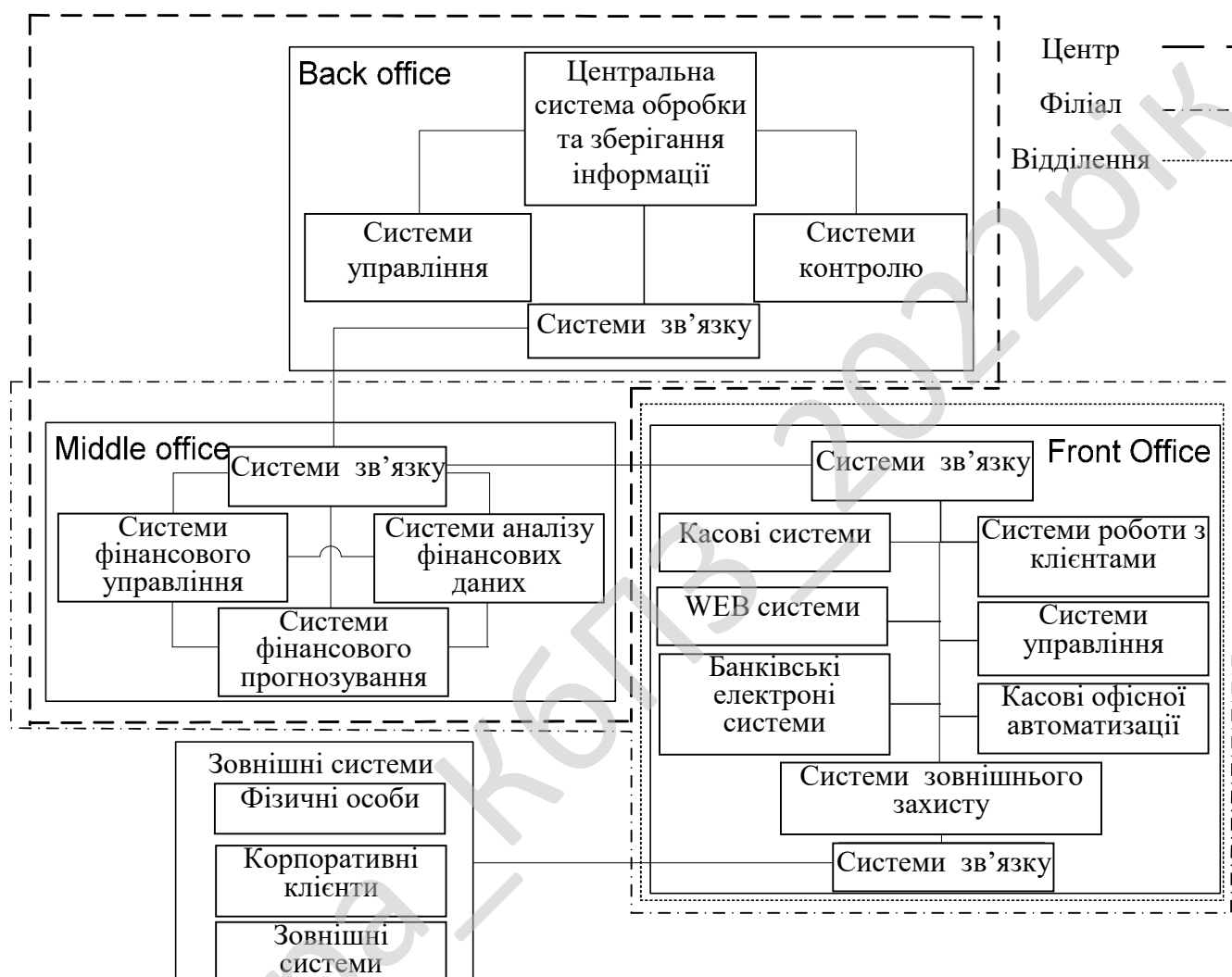


Рисунок 3.3 – Структурна схема розміщення елементів БІС

На структурній схемі наведені основні блоки системи, яка досліджується. Тобто системи з розподіленими ресурсами, ефективність якої оцінюють. Така система наведена на прикладі філії банку.

Front-office – це комплекс програмно-апаратних засобів, що підвищують ефективність спілкування. Це спеціалізовані системи, що автоматизують роботу співробітників, які спілкуються з "зовнішнім миром", які допомагають їм у

повсякденній діяльності. Іншою стороною взаємодії учасників «телефонних» відносин є системи, 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 системи дозволяють:

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

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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 уведення 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 допомогою не призначених для цього додатків. Відсутність єдиної інтегрованої системи для обліку всіх Front-офісних операцій значно обмежує можливості співробітників Middle-офісу по керуванню й аналізу діяльності банку.

Сучасні системи дозволяють:

– Оптимізувати внутрібанківські 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.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

- 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 системи дозволить автоматизувати наступні функції:

- Автоматичне формування бухгалтерських проводок по кожній операції на підставі раніше створених шаблонів.
- Можливість проектувати різні типи документообігу для тверджень операцій.
- 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, побудовані дві моделі

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

функціональної взаємодії елементів структури філії банку: зовнішня (рисунок 3.4) і внутрішня (рисунок 3.5).

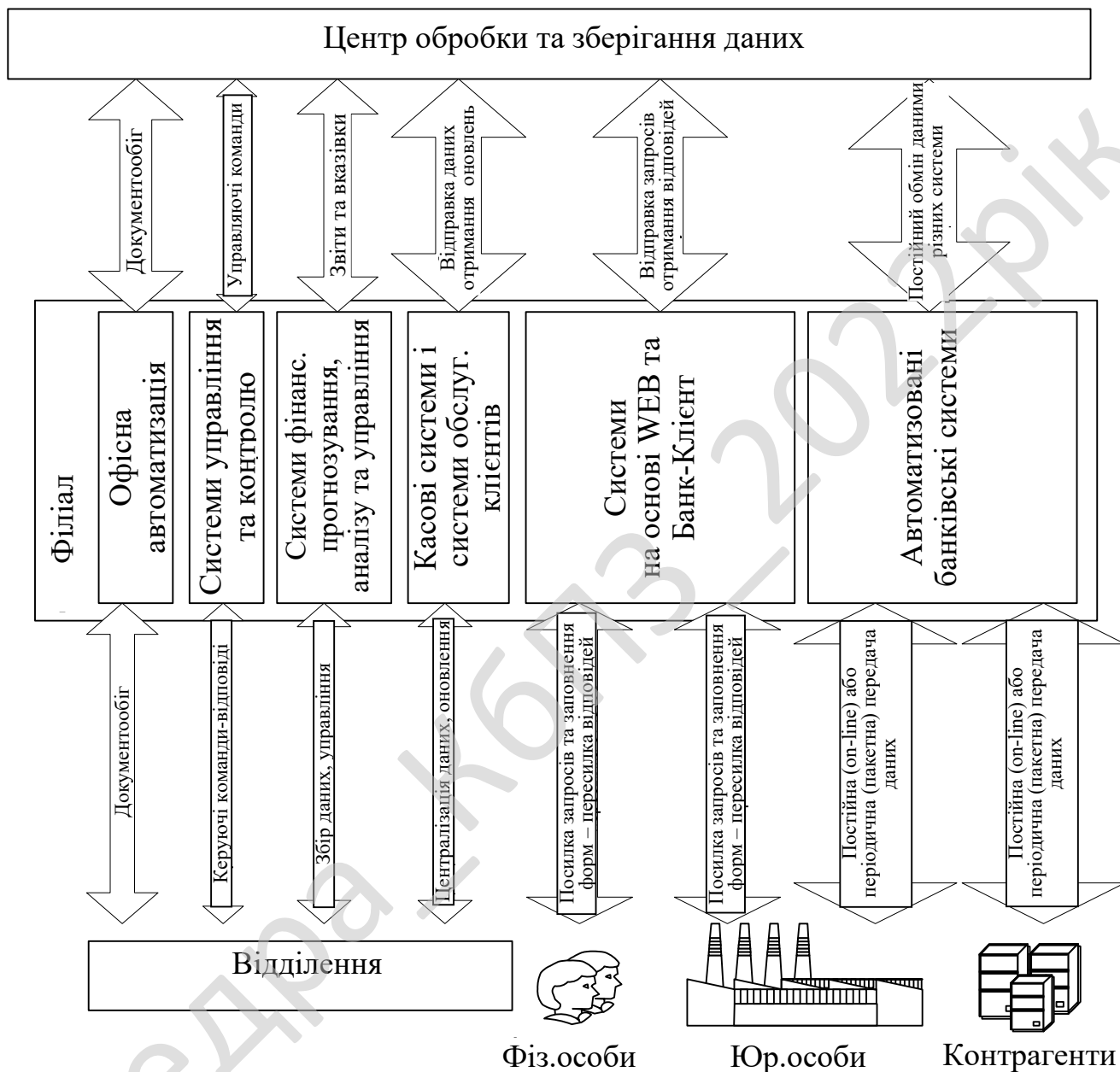


Рисунок 3.4 – Функціональна модель взаємодії елементів філії банку – зовнішня

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

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.22.0012.00.00.ПЗ

Арк.

47

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.
- Системи управління та контролю.
- Системи фінансового прогнозування, аналізу та управління.
- Касові системи і системи обслуговування клієнтів.
- Системи на основі WEB та Банк-Клієнт.
- Автоматизовані банківські системи.

Ці системи діють у рамках Front-office, Middle-office та back-office, а також працюють з зовнішніми системами.

При цьому ці системи 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.
- 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.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

- 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.

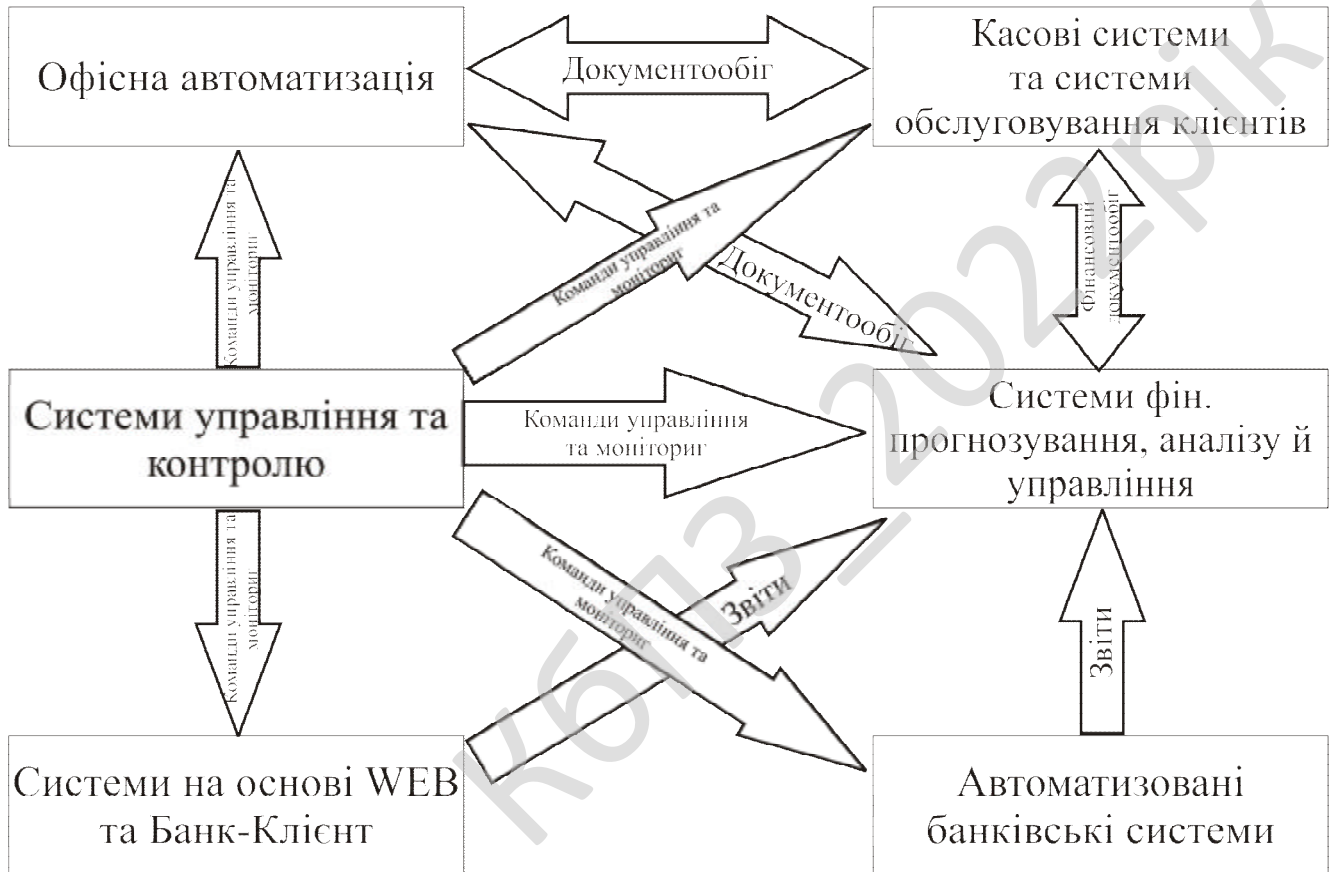


Рисунок 3.5 – Функціональна модель взаємодії елементів філії банку – внутрішня

При розгляді функціональної моделі внутрішньої взаємодії філії банку, ми застосуємо такий же самий підхід, як й при зовнішній.

Тобто розглянемо усі структурні елементи досліджуємої системи та функціональні зв'язки між ним. Оцінимо об'єм потоків та види топологій й протоколів передачі даних між блоками, й на основі цих досліджень можемо надати рекомендації по оптимізації внутрішньої комп'ютерної мережі.

### 3.4 Розробка діаграми процесів

На рисунку 3.6 зображена діаграма процесів, які відбуваються у системі. Програма починає працювати з запуску процесу початку/кінця програми.

Цей процес взаємодіє з наступними процесами:

- Побудова моделі мережного середовища банківської установи.
- Отримання значення оптимального протоколу для конкретної мережі.

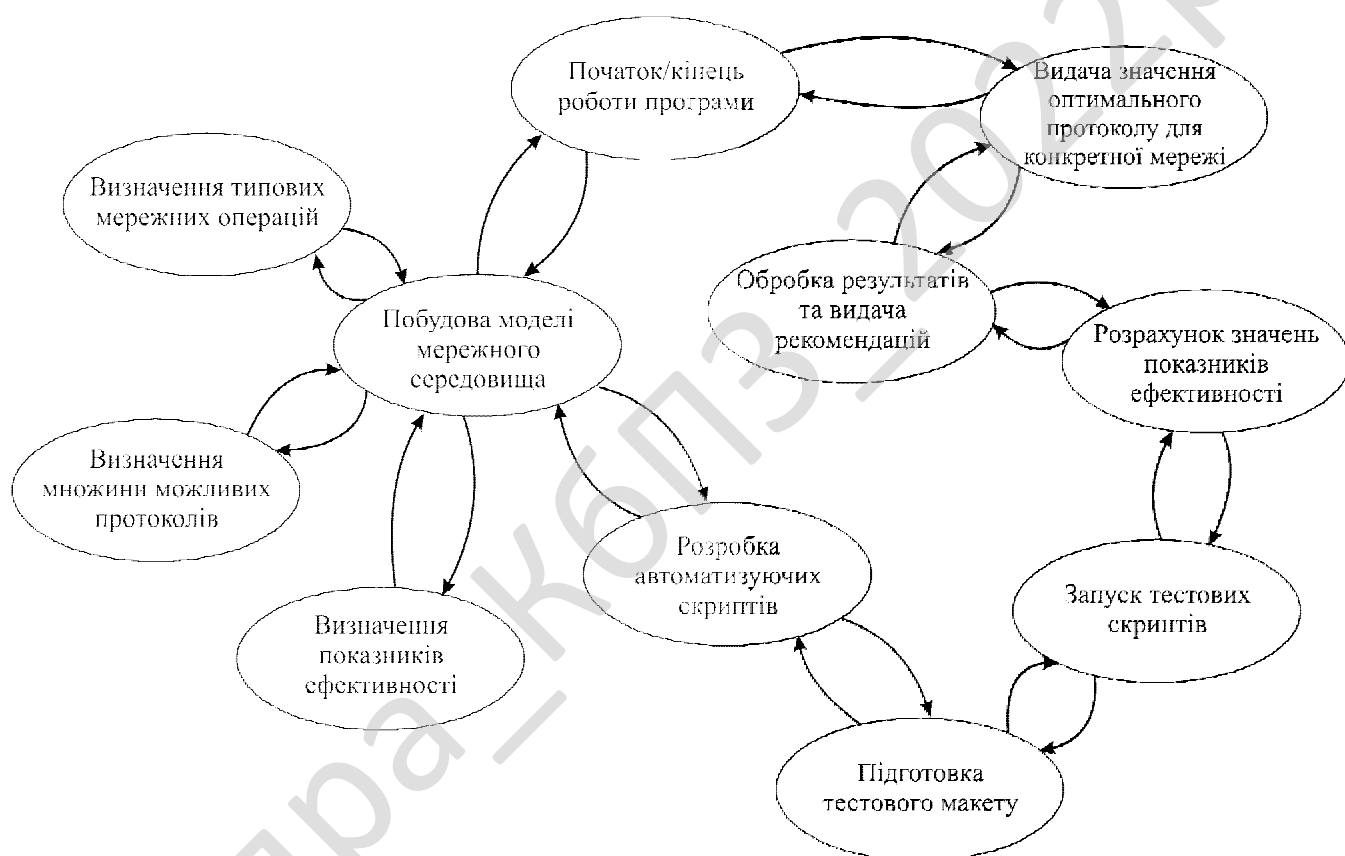


Рисунок 3.6 – Діаграма процесів системи

Процес побудови моделі мережного середовища взаємодіє з наступними процесами:

- процес визначення типових мережних операцій;
- процес визначення множини можливих протоколів;
- процес визначення показників ефективності;

– процес розробки автоматизуючих скриптів.

Останній процес у свою чергу взаємодіє з процесом підготовки тестового макету.

Після підготовки тестового макету відбувається процес запуску тестових скриптів.

Останній процес взаємодіє з розрахунком значень показників ефективності. Після дії цього процесу відбувається обробка результатів та видача рекомендацій.

За результатами обробки результатів видається значення оптимального протоколу для конкретної мережі. Яка розміщена у конкретній банківській установі.

Після цього роботи процесу програма завершує свою роботу.

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

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

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

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

Блок-схема основної програми наведена на рисунку 4.1.

Основна програма відображує алгоритм проведення дослідження ефективності розподіленої комп'ютерної мережі та формування оптимальних параметрів та протоколів для удосконалення цієї системи.

Робота програми відбувається наступним чином:

1. Будується модель мережного середовища та визначаються основні бізнес-функції.
2. Відбувається побудова профілю середовища та визначення множини можливих протоколів.
3. Відбувається визначення показників ефективності та вагових коефіцієнтів.
4. Визначаються методики розрахунку значень показника ефективності.
5. Визначаються методики розрахунку значень узагальненого показника ефективності.
6. Обнуляються індекси масивів можливих протколів, показників ефективності та вагових коефіцієнтів.
7. Розраховуються значення показників ефективності.
8. Відбувається перевірка на скінченність показників ефективності та вагових коефіцієнтів. Якщо скінчились то відбувається перехід на п.10, інакше на п.9.
9. Інкрементуються індекси показників ефективності та вагових коефіцієнтів.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

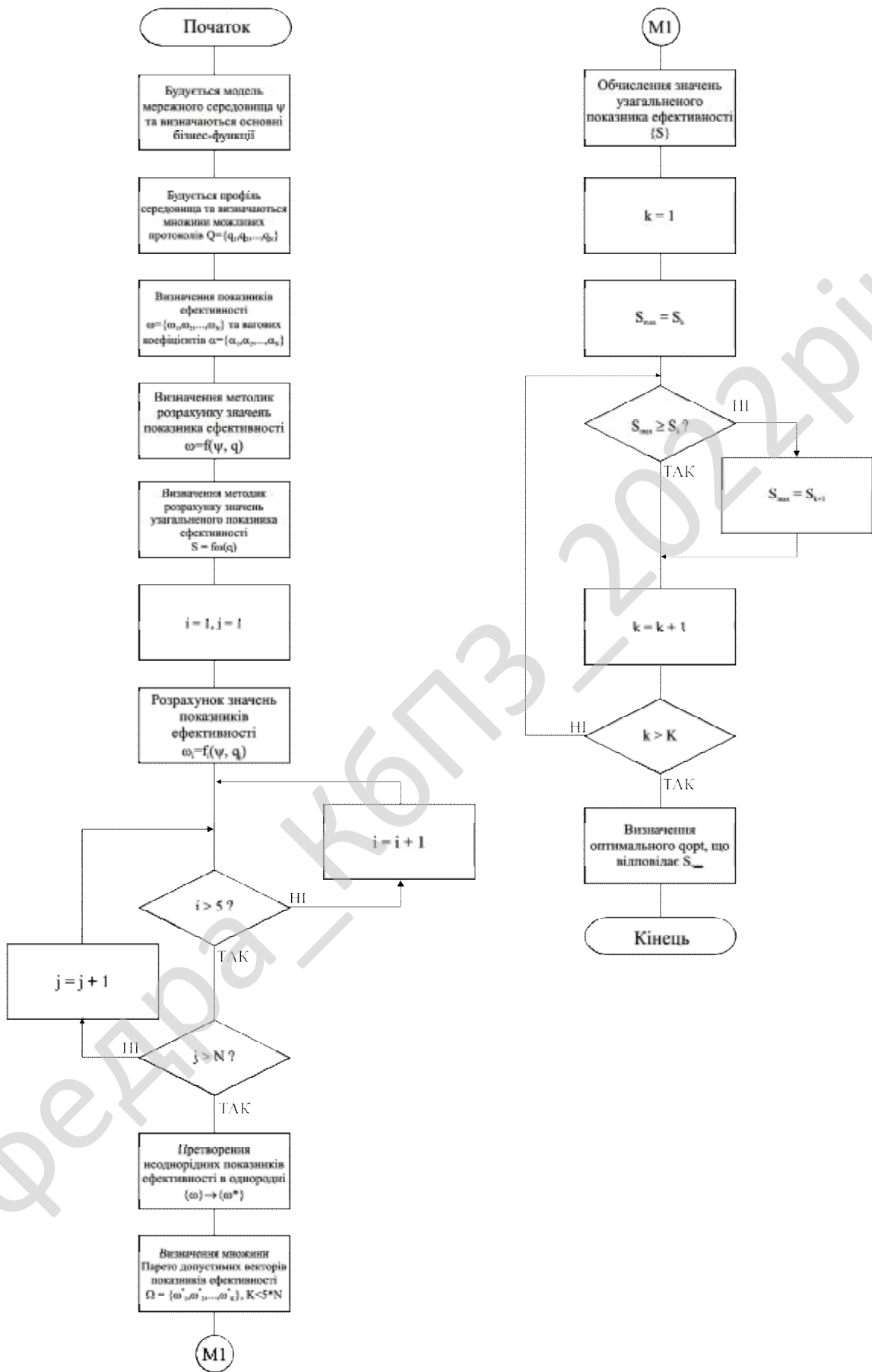


Рисунок 4.1 – Блок-схема програми проведення дослідження

10. Відбувається перевірка закінченості множини протоколів. Якщо вони скінчилися то відбувається перехід на п.12. Інакше відбувається перехід на п.11.

11. Інкрементується індекс масиву мережних протоколів.

12. Відбувається перетворення неоднорідних показників ефективності в однорідні, та будуються відповідні вектори.

13. Проводиться визначення множини припустимих векторів показників ефективності.

14. Розраховуються значення узагальненого показника ефективності.

15. Обнуляється індекс масиву узагальнених показників ефективності.

16. Максимальний узагальнений показник ефективності порівнюється поточному.

17. Порівнюється Максимальний узагальнений показник з наступним узагальненим показником. Якщо він більше або дорівнює то відбувається перехід до п.19. Інакше відбувається перехід до пункту 18.

18. Максимальний узагальнений показник порівнюється наступному.

19. Інкрементується індекс масиву узагальнених показників ефективності.

20. Індекс порівнюється з граничним індексом масиву. Якщо він більший то відбувається перехід до п.21. Інакше до п.17.

21. Видається значення оптимального протоколу для конкретної мережі, виходячи з максимального узагальненого показника ефективності мережі.

На рисунку 4.2 зображена блок-схема підпрограми методики функціонального тестування.

У блок-схемі наведені наступні позначення:

– O – набір типових мережних операцій;

–  $\omega$  – набір вимірюваних показників ефективності для типових мережних операцій;

– M – методики оцінки для кожного  $\omega$ ;

– R – набір скриптів, що реалізує виконання типових мережних операцій на клієнтських і серверних системах, а також провідний вимір обраних показників;

– U – число клієнтських систем;

– Q – набір стеків протоколів локальних мереж;

– МАК – сукупність параметрів, що описують тестовий макет.

Підпрограма методики функціонального тестування працює наступним чином.

1. Відбувається визначення типових мережних операцій.
2. Відбувається визначення вимірювальних показників та методик їх оцінки.
3. Розробляються автоматизуючі скрипти.
4. Проводиться підготовка тестового макету.
5. Відбувається налаштування тестового макету для визначеного набору протоколів.
6. Запускаються тестові скрипти.
7. Відбувається вимірювання для всіх клієнтських систем з набором вимірюваних показників ефективності для типових мережних операцій.
8. Число клієнтських систем порівнюється з максимальним. Якщо дорівнює то відбувається перехід до п.9. Інакше він інкрементується та відбувається перехід до п.6.
9. Набір стеків протоколів локальних мереж порівнюється з максимальним. Якщо дорівнює то відбувається перехід до п.10. Інакше він інкрементується та відбувається перехід до п.5.
10. Відбувається обробка результатів та видача рекомендацій.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

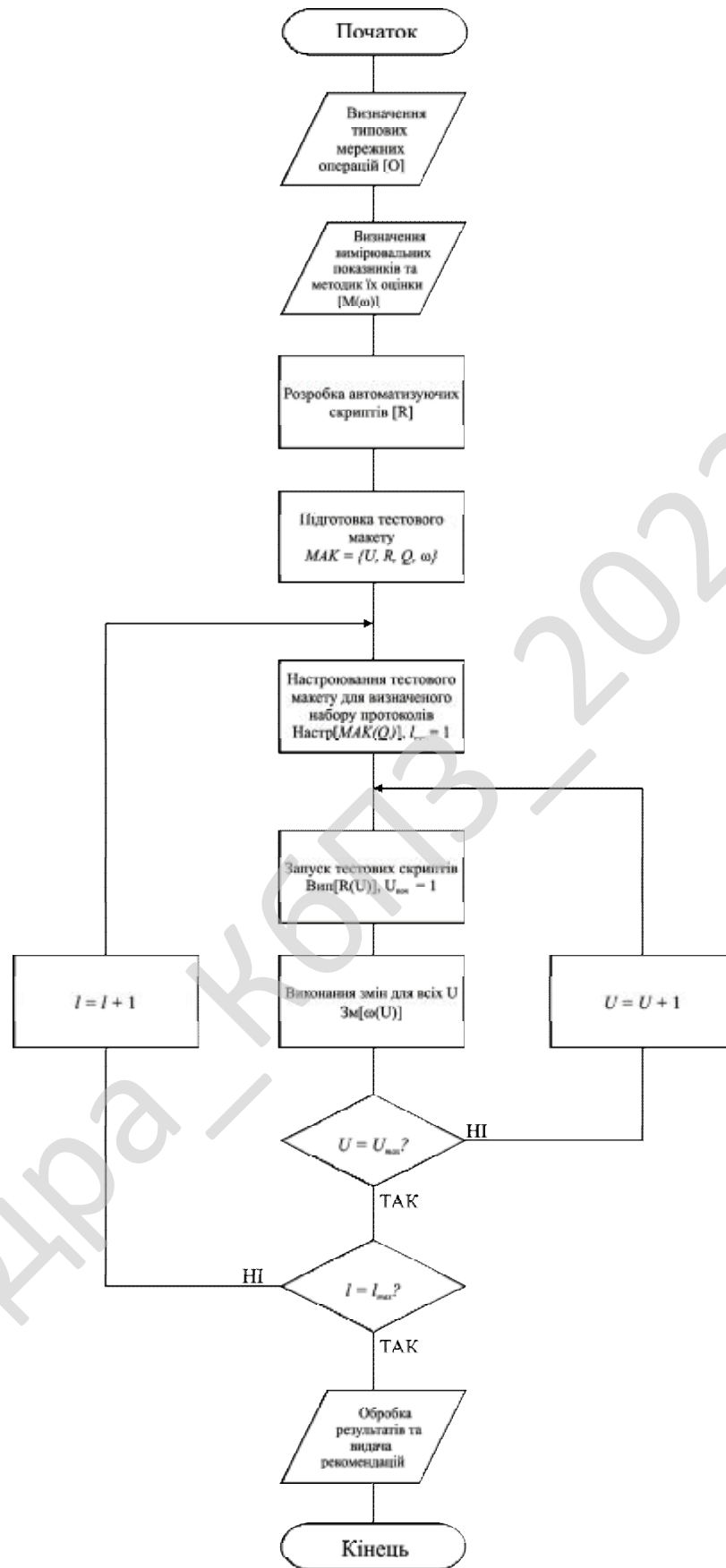


Рисунок 4.2 – Блок-схема підпрограми методики функціонального тестування





таблицею. Наступний етап складається з 16 раундів однієї й тої ж функції, що використовує операції зрушення й підстановки. На третьому етапі ліва й права половини виходу останньої (16-й) ітерації міняються місцями. Нарешті, на четвертому етапі виконується перестановка  $IP^{-1}$  результату, отриманого на третьому етапі. Перестановка  $IP^{-1}$  інверсна початковій перестановці.



Рисунок 4.3 – Загальна схема DES

Праворуч на рисунку показаний спосіб, яким використовується 56-бітний ключ. Спочатку ключ подається на вхід функції перестановки. Потім для кожного з 16 раундів підключ  $K_i$  є комбінацією лівого циклічного зрушення й перестановки. Функція перестановки та сама для кожного раунду, але підключи  $K_i$  для кожного раунду виходять різні внаслідок повторюваного зрушення біт ключа.

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення реалізує системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Програмно-апаратні вимоги:

- Загальний обсяг ОЗП: 256 Мбайт.
- Вільний простір на жорсткому диску: 35 Мбайт.
- Операційна система Microsoft Windows 10/11.

Головне вікно програми зображене на рисунку 5.1.

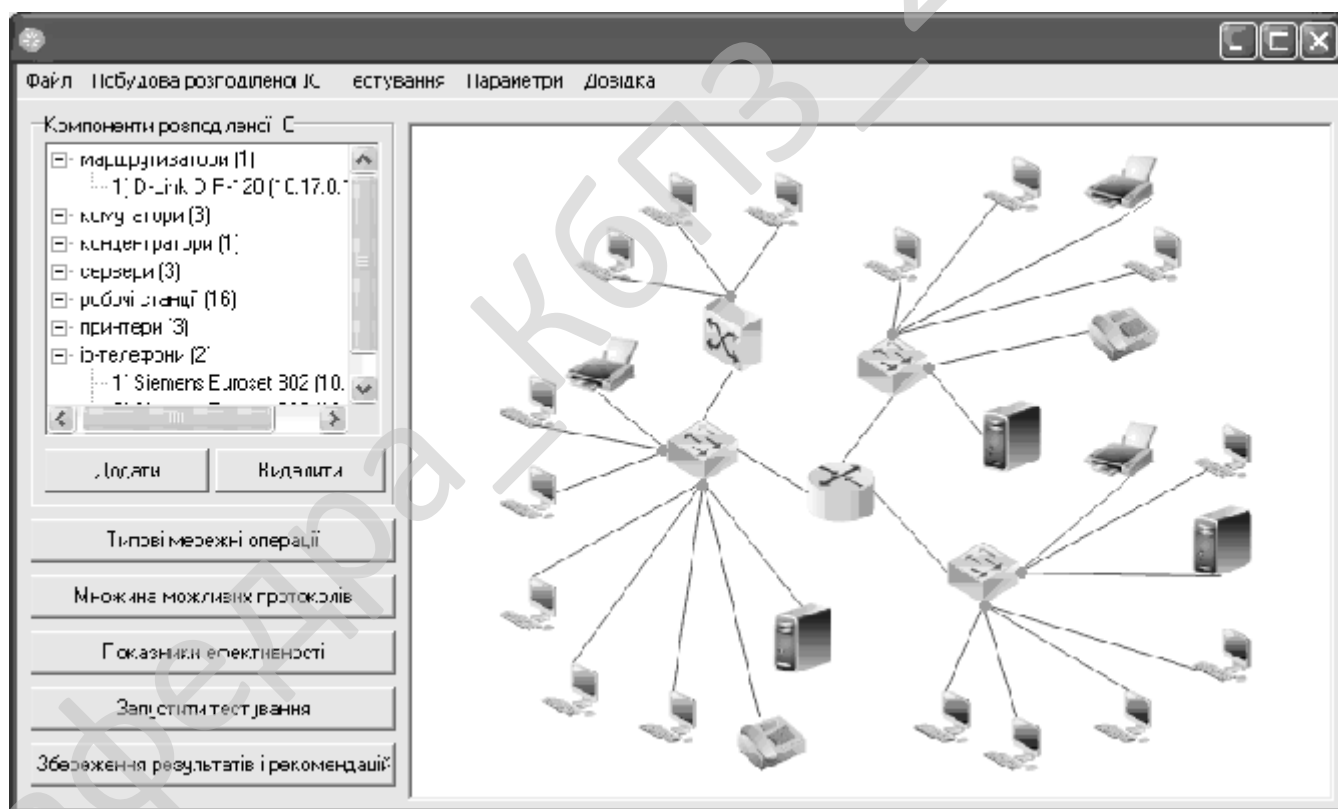


Рисунок 5.1 – Головне вікно програми

Для побудови розподіленої інформаційної системи необхідно додати її компоненти, за допомогою кнопки «Додати», та розмістити їх на схемі, а потім з'єднати.

Щоб з'єднати входи й виходи елементів лінією, необхідно виконати наступні дії.

При наведенні курсору миші на елемент він виділяється, після цього натискаючи ліву кнопку миші треба тягнути до іншого елемента, з яким необхідно з'єднати. Як тільки елемент, з яким ми хочемо з'єднати також виділиться відпускаємо кнопку миші. У результаті буде створена лінії, що з'єднує два елемента.

Для видалення елемента слід виділити його назву у списку та натиснути кнопку «Видалити», або натиснути на елементі на схемі та натиснути клавішу Delete.

Для видалення зв'язку слід клацнути на ньому один раз лівою кнопкою миші та натиснути клавішу Delete, або клацнути правою кнопкою та вибрати у контекстному меню, що з'явиться пункт «Видалити зв'язок».

Після побудови схеми треба задати типові мережні операції, множини можливих протоколів та показники ефективності даної розподіленої інформаційної системи.

Коли побудована схема та задані всі параметри, слід запустити тестування.

Після здійснення тестування програма видасть його результати та рекомендації, а також значення оптимального протоколу для даної мережі.

Коротку довідку про розроблену програму та її автора можна переглянути вибравши підпункт «Про програму...» з меню «Довідка», після чого з'явиться вікно зображене на рисунку 5.2.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Центральноукраїнський національний технічний  
університет  
Кафедра кібербезпеки та програмного забезпечення  
ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА  
за другим (магістерським) рівнем вищої освіти  
на тему: "Дослідження та програмна реалізація системи  
бюджетування хмарної інформаційної системи для визначення  
ефективності її впровадження"  
ОПП «Комп'ютерні науки»  
спеціальності 122 «Комп'ютерні науки»  
Виконав: Поляруш Б.С.  
Науковий керівник: Доренський О.П.  
Кропивницький - 2022

ок

Рисунок 5.2 – Довідка про програму

					VKPM-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

*Метою розробки є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.*

*Об'єктом дослідження є процес бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.*

*Предметом дослідження є методи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.*

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

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

- Удосконалено метод бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- Розроблено вітчизняний продукт бюджетування хмарної інформаційної системи для визначення ефективності її впровадження, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

### 7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці).

В магістерській роботі було проведено дослідження та розроблене програмне забезпечення системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	130
3. Запланований термін розробки, днів	Fpq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	13000
33. Норматив додаткової зарплати, % :	Н <sub>д</sub>	10
34. Норматив відрахувань у соціальні фонди, %	Н <sub>с</sub>	22
35. Норматив загальногосподарських витрат, %	Н <sub>г</sub>	15
36. Норматив витрат на освоєння нових мов програмування, %	Н <sub>п</sub>	15
37. Рівень рентабельності програмної продукції, %	Р <sub>е</sub>	50
38. Ставка податку на додану вартість, %	Н <sub>дв</sub>	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

де  $A$  – коефіцієнт Боєма,  $A=2,45$ ;  $Size$  – загальний об'єм відлагодженого програмного коду, тис. рядків;  $B$  – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де  $W_i$  – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де  $\prod V_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{PI} = 0,3CT_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де  $S$  – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);  $S$  – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{PI} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 80 = 135 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67



Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	8	720	12
Монітор	60	8	480	8
Клавіатура	30	8	240	4
Маніпулятор «мишка»	30	8	240	4
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	4	120	2
Кабельні господарства ЛОМ на 1 м.п.	2,5	280	700	11,67
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 <sub>ч</sub>	47,33

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{op}^c = \frac{47 \cdot 3}{1,2} = 117,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{ор}^c}{F_{ор} \cdot T_{зм}} \quad (7.7)$$

$$Ч_{ел} = 117,5 / (60 \cdot 8) = 0,25 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів – електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	Кількість штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server, серверу доступу АДСЛ (ОС Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	0,2	0,1
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,2	
Всього		0,8	

Продовження табл. 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,4
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	1	
	Контроль взаєморозрахунків з постачальниками	0,2	
Всього		3,2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,2	0,1
	Створення графічних і стилістичних елементів сайту	0,2	
	Оформлення банерів і промо-сторінок	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	0,2	0,1
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	

Складемо штатний розклад виконавців:

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71



$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де  $R_{cn}^1$  – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

$S_y$  – питома площа на одне робоче місце,  $m^2$ ,

$C_{nl}$  – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно  $8 m^2$ . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{nv} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де  $C_m$  – ціна меблів для одного робочого місця, грн.

$$I_{nv} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет-магазину Компбест за 24.10.22 – джерело <https://compbest.com.ua>.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		12721
Системний блок		7721
Процесор	Intel Core i5-4570 (4 ядра по 3.2 – 3.6 GHz); 6 MB Cache Memory	2000
Системна плата	ASUS B85M-K Socket 1150 Intel B85 OEM Refurbished (SATA II – 2 шт, SATA-3 – 4 шт, 4x USB 3.0, 6x USB 2.0, 4x Audio Ports, LAN (RJ-45), 2x PS/2, Com Port, 2x DP, VGA)	1350
Відеокарта	Інтегрована Intel HD Graphics 4600	-
Жорсткий диск	HDD 500 Gb SAMSUNG Barracuda HD502HJ (3.5", 500ГБ, 16МБ, SATA II-300)	1490
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-12800 Kingston, 1600MHz, 512M x 64, CL9-9-9-27, 1.65V, w/heatsink, HyperX	834
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	416
Корпус	Logicpower 8702 – 550w 12cm	1411
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	220
інше	Клавіатура, мишка	Подарунок



Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	137536	-	-
Всього по групі	137536	30	41260,8
Нематеріальні активи			
4. Нематеріальні активи	13000	25	3250
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	121875	20	24375
7. Господарський інвентар	28000	25	7000
Всього по групі	158906	-	33632,75
Разом	$K_p = 1717442$		$A_p = 148543,55$

Примітка: вартість автомобіля ГАЗ Газель взята по даним електронного ресурсу <http://www.avtopoisk.ua> та складає 121875 грн.

## 7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де  $N_e$  – Кількість екземплярів програм, шт.

$$Z_o = 420 \cdot 176 / 130 = 569 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де  $H_q$  – норматив додаткової зарплати, %

$$Z_d = 569 \cdot 10 \cdot 0,01 = 57 \text{ грн}$$

Відрахування на соціальні потреби за нормативом  $H_c = 22\%$  від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де  $H_c$  – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22 (569 + 57) = 138 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом  $H_g = 15\%$  від основної зарплати

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де  $H_g$  – загальногосподарські витрати, %

$$G_{ocn} = 569 \cdot 15 \cdot 0,01 = 85 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де  $Z_{M1}$  – вартість паперу, грн.,  $Z_{M2}$  – вартість запам'ятовуючих пристроїв, грн.,  $Z_{M3}$  – вартість фарби, картриджей, тонеру, грн.,  $N_e$  – кількість екземплярів програм, шт.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Згідно прийнятих норм на підприємстві  $n_{sum}$  приймаємо 0,33 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає  $C_n=210$  грн., визначаємо вартість паперу за період розробки:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 210 \cdot 0,33 = 70 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуваних пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 31):

$$Z_{M2} = \sum C_{\delta}, \quad (7.17)$$

де  $C_{\delta}$  – вартість дисків CD/DVD: CDR box – 22 грн/шт., DVD-R box – 30 грн/шт.

$$Z_{M2} = 30 \cdot 30 + 22 = 922 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де:  $C_z$  – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (70 + 922 + 1702) / 130 = 21 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ( $H_n = 15\%$ ) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де  $H_n$  – норматив витрат на освоєння нових мов програмування, %

$$O_n = 569 \cdot 15 \cdot 0,01 = 85 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 130$  прим.)

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78



Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	$A_M$	190
8. Повна собівартість програмного забезпечення	$C_{\Pi}$	1145
9. Плановий прибуток	$\Pi_p$	572,5
10. Ціна підприємства $C_{\Pi} = C_{\Pi} + \Pi_p$	$C_{\Pi}$	1717,5
11. Податок на додану вартість $\text{ПДВ} = 0.01 \cdot N_{\text{дв}} \cdot C_{\Pi}$	ПДВ	343,5
12. Відпускна ціна програмної продукції $C = C_{\Pi} + \text{ПДВ}$	$C$	2061

### 7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн	
	Базовий	Новий
Вартість програмної продукції	–	2061
Всього капітальних витрат	–	2061





Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	130
2. Повна собівартість розробленої програми	Грн.	1145
3. Ціна розробленої програми	Грн.	1717,5
4. Плановий прибуток від реалізації розробленої програми	Грн.	572,5
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1717442
7. Загальний прибуток від реалізації програмної продукції	Грн.	74425
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	49667,7
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	2061
11. Величина економічного ефекту у користувача програмної продукції	Грн.	5679
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Роки	0,33

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\sigma} - I_n) - E_n(K_n - K_{\sigma}), \quad (7.26)$$

де  $I_{\delta}$ ,  $I_n$  – величина експлуатаційних витрат за базовим и новим варіантом відповідно,  $K_{\delta}$ ,  $K_n$  – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (13420 - 7225,25) - 0,25 \cdot 2061 = 5679 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n} \quad (7.27)$$

$$T_{cn} = \frac{2061}{13420 - 7225,25} = 0,33 \text{ роки.}$$

## 7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Загальна комп'ютеризація суспільства призвела до того, що використання комп'ютерів стало повсюдним у всіх сферах економіки та народного господарства. Застосування персональних комп'ютерів і ЕОМ дозволило значно підвищити продуктивність праці, змінити характер і зміст праці [4]. Комп'ютеризація, поряд з незаперечними перевагами, тягне за собою і багато проблем. Для того, щоб активне застосування комп'ютерних технологій не стало додатковим чинником погіршення здоров'я, вкрай необхідно щоб робоче місце відповідало гігієнічним вимогам. Темою дипломного проекту є розробка та дослідження та реалізація програмного продукту, тому актуально буде розгляд умов праці програміста.

### 8.1. Аналіз умов праці на робочому місці програміста

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 5м×7,2м×2,8м. Одна з її більших стін має шість двостулкових вікон, розмірами 2,2м×1,8м, які виходять на північний схід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1250 мм×850 мм. Висота столів 750 мм. Висота стільців від рівня підлоги становить 430 мм.

Згідно НПАОП 0.00-1.28-10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м<sup>2</sup> на одне робоче місце. Кратність повітрообміну в приміщенні вузла також

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

регламентується ДСанПіН 3.3.2.007 – 98, вона повинна становити 20 м<sup>3</sup>/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5 – 28 – 2006 роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007 – 98 рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи. Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела шуму, як імпульсні. Відповідно паспорта на приміщення рівень звуку, Дб(А), обмірюваний за шкалою (А) шумоміра досяг величини 28,3 Дб(А) при роботі всього встаткування вузла, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах	71	61	54	49	45	42	40	38	50

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормуються НПАОП 0.00 – 1.28 – 10. Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг. Даного місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту

Ріст людини, см	Висота робочої поверхні мм,	Висота простору для ніг, мм	Висота робочого сидіння, мм
175	765(740)	655(600)	450(440)

У дужках зазначені реальні значення параметрів робочого місця; всі вони не відповідають параметрам, зазначеним у стандарті.

Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище.

У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах ДСН 3.3.6.042 – 99 встановлені величини параметрів мікроклімату, що створюють комфортні умови [1]. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (див. табл. 8.3).

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 – 24°C
	Відносна вологість	40 – 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 – 25°C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

## 8.2. Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.



виконуваної роботи, дозволяють понизити навантаження на опорно-руховий апарат людини, що працює за комп'ютером.

### 8.3 Дослідження інформаційного навантаження на програміста

Програміст, у залежності від підготовки і досвіду, вирішує задачі різної складності, але в загальному випадку робота програміста будується по наступному алгоритму:

Таблиця 8.4 – Алгоритм

Етап	Зміст	Витрата часу, %
III	Постановка задачі Вивчення матеріалу за поставленою задачею	6.25
III	Визначення методу рішення задачі	6.25
IV	Складання алгоритму рішення задачі	12.5
V	Програмування	25
VI	Налагодження програми, складання звіту	50

Даний алгоритм відображає загальні дії програміста при рішенні поставленої задачі незалежно від її складності.

Таблиця 8.5 – Алгоритм відображає загальні дії програміста при рішенні поставленої задачі незалежно від її складності

Етап	Член алгоритму	Зміст роботи	Літерне позначення
1	2	3	4
I	1	Одержання першого варіанта технічного завдання	A
	7	Складання і уточнення технічного завдання	Ri
		Одержання остаточного варіанта технічного завдання	Сіл
	4	Складання переліку матеріалів, що існують	НП2
	5	Вивчення матеріалів по тематиці задачі	Ag

Продовження таблиці 8.5

1	2	3	4
	6	Вибір методу рішення	C2J3
	7	Уточнення й узгодження обраного методу	B2
	8	Остаточний вибір методу рішення	CчT4
	9	Аналіз вхідної і вихідної інформації	B2
	10	Вибір мови програмування	C4TS
	11	Визначення структури програми	H3C5q
	12	Складання блок-схеми програми	C6q2
	13	Логічний аналіз програми і коректування її	F1H4W2
	14	Компіляція програми	F2
	15	Виправлення помилок	D1W2
	16	Редагування програми	F2H5B3W
	17	Виконання програми	F3
	18	Аналіз результатів виконання	H6W5
	19	Тестування	C7W6
	20	Підготовка звіту про роботу	F4

Підрахуємо кількість членів алгоритму і їхню частоту (імовірність) щодо загального числа, прийнятого за одиницю. Імовірність повторення  $i$ -ої ситуації визначається по формулі:

$$P_i = k/p, \quad (8.1)$$

де  $k$  – кількість повторень кожного елемента одного типу,  $p$  – сумарна кількість повторень від джерела інформації, одного типу.

Результати розрахунку зведемо в таблицю:

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Таблиця 8.6 – Результати розрахунку

Джерело	Члени алгоритму	Символ	Кількість	Частота повторень
1	Аферентні – усього		6	1,00
	Вивчення технічної	А	2	0,33
	Спостереження	Р	4	0,67
2	Еферентні – усього		18	1,00
	Уточнення	В	3	0,17
	Вибір найкращого	С	8	0,44
	Виправлення	0	1	0.06
	Аналіз отриманих	н	6	0,33
	Виконання	к	0	0
3	Логічні умови		13	1,00
	Прийняття рішень на основі вивчення	І	5	0,39
	Грабічні матеріали	Ч	2	0.15
	Отриманого тексту	V	6	0.46
	Усього:		37	

Кількісні характеристики (Табл. 8.6) дозволяють розрахувати інформаційне навантаження програміста [8]. Ентропія інформації елементів кожного джерела інформації розраховується по формулі, біт/сигн:

$$H_j = \sum_{i=1}^m p_i \log_2 p_i, \quad (8.2)$$

де  $m$  – число однотипних членів алгоритму розглянутого джерела інформації.

$$H_1 = 2 \times 2 + 2 \times 4 = 12$$

$$H_2 = 3 \times 1,585 + 8 \times 3 + 0 + 6 \times 2,585 = 44,265$$

$$H_3 = 5 \times 2,323 + 2 \times 1 + 6 + 2,585 = 29,125$$

Потім визначається загальна ентропія інформації, біт/сигн:

$$H_s = H_1 + H_2 + H_3, \quad (8.3)$$

де  $H_1$ ,  $H_2$ ,  $H_3$  – ентропія аферентних, еферентних елементів і логічних умов відповідно.

$$H_s = 10 + 44,265 + 29,125 = 83,39.$$

Далі визначається потік інформаційного навантаження біт/хв,

$$F = \frac{H_s \cdot N}{t}, \quad (8.4)$$

де  $N$  – сумарне число всіх членів алгоритму;  $t$  – тривалість виконання всієї роботи, хв.

Від кожного джерела в інформації (члена алгоритму) у середньому надходить 3 інформаційних сигнали в годину, час роботи – 225 годин,

$$\Phi = \frac{83,39 \cdot 3 \cdot 225}{13500} = 2,6 \text{ біт/с}$$

Розраховане інформаційне навантаження порівнюється з припустимою. При необхідності приймається рішення про зміни в трудовому процесі.

Умови нормальної роботи виконуються при дотриманні співвідношення:

$$\Phi_{\text{доп.мін}} < \Phi_{\text{расч}} < \Phi_{\text{доп.макс}}, \quad (8.5)$$

де  $\Phi_{\text{доп.мін}}$ , і  $\Phi_{\text{доп.макс}}$  мінімальний і максимальний припустимі рівні інформаційних навантажень (0,8 і 3,2 біт/з відповідно);  $\Phi_{\text{расч}}$  – розрахункове інформаційне навантаження  $0,8 < 2,6 < 3,2$

#### 8.4 Висновки до розділу

У цій частині магістерської роботи були розглянуті вплив факторів трудового і виробничого середовища програмістів, дослідження інформаційного навантаження на програміста. Дотримання умов, що визначають оптимальну організацію робочого місця програміста і навантаження, отримані ним в процесі роботи, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить, як у кількісному, так і в якісному відносінах продуктивність праці програміста.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- Досліджена система бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.
- На основі отриманих результатів досліджень створена програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

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

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

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

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

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DES.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 5679 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,33 роки.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поляруш Б.С. Дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.

2. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

3. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

4. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

5. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

6. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96





19. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

20. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

21. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

22. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

23. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об'єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

24. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (ІТ & І): II між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

25. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко,



34. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.
35. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.
36. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.
37. Манухина С.Ю. Инженерная психология и эргономика: хрестоматия / С.Ю Манухина. – М.: Изд. центр ЕАОИ, 2009. –224 с.
38. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.
39. Мунипов О.В. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / О.В. Мунипов, В.П. Зинченко. – М.: Логос, 2001. – 356 с.
40. Надеев А.И. Математическая модель эксплуатационной надежности интеллектуальных датчиков / А.И. Надеев, Р.А. Юсупов, Ю.К. Свечников, Д.Р. Юсупов // Измерительная техника. – М: Стандартинформ, 2004. – № 1. – С. 8-11.
41. Надійність техніки. Аналіз надійності. Основні положення: ДСТУ 2861-94 – [Чинний від 1997–01–01]. – Київ: Держстандарт України, 1995. – 33 с. – (Національний стандарт України).
42. Надійність техніки. Терміни та визначення: ДСТУ 2860-94 – [Чинний від 1996–01–01]. – Київ: Держстандарт України, 1994. – 36 с. – (Національний стандарт України).
43. Нейлор К. Как построить свою экспертную систему / К. Нейлор. – М.: Энергоатомиздат, 2007. – 242 с.
44. Николаева И. П. Экономический словарь / И.П. Николаева. – Проспект, 2015. – 399 с.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>101</b>

45. Онищук А.Г. Радиоприемные устройства: Учебн. пособ. – 2-е изд., испр. / А.Г. Онищук, И.И. Забеньков, А.М. Амелин. – Минск: Новое знание, 2007. – 240 с.

46. Осипов В. Базы данных и Delphi. Теория и практика / В. Осипов. – БХВ-Петербург, 2011. – 752 с.

47. Павленко М.А. Метод кольорового кодування інформаційних елементів при розробці інформаційних моделей в перспективних АСУ / М.А. Павленко, П.Г. Берднік, Д.В. Прибильнов // Наукова весна – 2008: Матеріали міжнародної наук. – практ. конф. – Х.: МСУ, 2008. – С. 25-27.

48. Павленко М.А. Метод разработки модели деятельности оператора АСУ в системах управления сложными динамическими объектами / М.А. Павленко, О.С. Бодяк, М.Ю. Гусак, С.И. Симонов // Системы обработки информации. – Харьков: ХУПС, 2012. – Вып. 9(107). – С. 196-200.

49. Павленко М.А. Метод разработки системы информационного обеспечения деятельности оператора системы управления интеллектуальной сетью связи / М.А. Павленко, В.М. Руденко, П.Г. Бердник // Наукові записки Українського науково-дослідного інституту зв'язку. –К.: УНДІЗ, 2008. – Вып. 5(7). – С. 33–41.

50. Павленко М.А. Моделирование деятельности оператора АСУ в системах управления сложными динамическими объектами / М.А. Павленко, В.Н. Руденко, П.Г. Бердник, С.В. Сериченко // Наукова весна: міжнар. наук.– практ. конф., м. Харків: матер. – Х.: МСУ, 2009. – С. 10-13.

51. Павленко П.М. Методика синтезу проектних варіантів концептуального проектування складних технічних систем / П.М. Павленко, П.М. Ратушний // XI Міжнародна науково- технічна конференція АВІА-2013: XI Міжнародна науково-технічна конференція, 21-23 травня 2013 р.: тези доп. – Київ: НАУ, 2013. – Т.1. – С. 3.1–3.4.

52. Палий И.А. Линейное программирование. Учебное пособие / И.А. Палий. – М.: Эксмо, 2008. – 256 с.

					<b>ВКРМ-122.22.0012.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

53. Пархоменко П.П. Основы технической диагностики. / П.П. Пархоменко, Е.С. Согомонян. – М: Энергоиздат, 1981. – 642 с.

54. Першин А.В. Анализ процесса взаимодействия пользователя с экспертной системой / А.В. Першин, М.А. Павленко, А. В.Александров // Проблеми інформатики і моделювання: п'ята міжнар. наук.-техн. конф., м. Харків: матер. – Х.: НТУ ХП, 2005. – С. 32.

55. Пицык В.В. Задача прогнозирования точности измерений при допусковом контроле систем / Пицык В.В. // Измерительная техника. – М.: Стандартиформ, 2004. – № 7. – С. 3-6.

56. Повітряний кодекс України: станом на 1 верес. 2011 р., відповідає офіц. тексту. – Х.: Право, 2011. – 48 с.

57. Половко А.М. Основы теории надежности / А.М. Половко, С.В. Гуров. – П.: БХВ-Петербург, 2006. – 702 с.

58. Поморова О.В. Теоретичні основи, методи та засоби інтелектуального діагностування комп'ютерних систем: Автореф. дис. на здобуття наук. ступеня д-ра техн. наук: 05.13.13 / О.В. Поморова; Нац. ун-т "Львів. політехніка". – Л., 2007. – 33 с.

59. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>

60. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

61. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

62. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

63. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час

					ВКРМ-122.22.0012.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103



Додаток А  
(обов'язковий)

Технічне завдання

Зміст

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

					<b>ВКРМ-122.22.0012.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Поляруш Б.С.				<i>Дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження</i>	Літ.	Аркуш	Аркушів
Перевірів	Доренський О.П.					М	1	6
Н. Контр.	Гермак В.С.					ЦНТУ КН-21М-1,4		
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 18-13 від 17.08.2022 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

## 4 Джерела розробки

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

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-122.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження;
- 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.

					<b>ВКРМ-122.22.0012.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 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.

					ВКРМ-122.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці програміста.

					ВКРМ-122.22.0012.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 104 аркуша.

## 10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2022 р.

					<b>ВКРМ-122.22.0012.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

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

ЗАТВЕРДЖУЮ

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

\_\_\_\_\_ Доренський О.П.

*Дослідження та програмна реалізація  
системи бюджетування хмарної інформаційної системи для визначення  
ефективності її впровадження*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2022 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);
  end;

```

```

procedure btKompClick(Sender: TObject);
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; TForms: 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;

```

```

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;
    end;
end;

```

```

    Time2.Enabled := false;
    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;
    end;
end;

```

```

btKompCont.Enabled := false;
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;
  end;
end;

```

```

    Panel4.Visible := true;
    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.Inversion := optInversion;
    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;
            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.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;
    end;
end;

```

```

    Prog.Max := GenAlg.MinPop;
    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;

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.
```

Кафедра \_ КБПЗ \_ 2022 рік

## 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;
    btnGetShares: TButton;
    btnCloseShares: TButton;
    btnAddShares: TButton;
    btnCloseSession: TButton;
    btnGetSessions: TButton;
    bvlTopSessions: TBevel;
    plButtonFiles: TPanel;
    btnGetFiles: TButton;
    btnCloseFile: TButton;
    bvlLeftFiles: TBevel;
    plFiles: TPanel;
    lvFiles: TListView;
    bvlTopFiles: TBevel;
    gbxCTraffic: TGroupBox;
    lvCTraffic: TListView;
    bvlCTraffic: TBevel;
    tmrCTraffic: TTimer;
    function IsXP(var Value: Boolean): Boolean;
    procedure btnGetSharesClick(Sender: TObject);
    procedure btnCloseSharesClick(Sender: TObject);
    function SelectDirectory: String;
    procedure btnAddSharesClick(Sender: TObject);
    function CardinalToTimeStr(Value: Cardinal): String;
    procedure btnGetSessionsClick(Sender: TObject);
    procedure btnCloseSessionClick(Sender: TObject);
    procedure btnGetFilesClick(Sender: TObject);
    procedure btnCloseFileClick(Sender: TObject);
    procedure tmrTrafficTimer(Sender: TObject);
  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;
        prefmaxlen: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;
        prefmaxlen: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 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

```

```

        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

```

```

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'ясовуємо 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;
            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.
```

Кафедра \_ КБПЗ \_ 2022 рік

## 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('МАГІСТЕРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Дослідження та програмна реалізація системи бюджетування  
хмарної інформаційної системи для визначення ефективності її впровадження');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Поляруш Богдан Сергійович');
  Memo1.Lines.Add('                гр. КН-21М-1,4');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Доренський О.П. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('м. Кропивницький 2022');
end;
end.
```