

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

“ ___ ” _____ 2021 р.

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

**“Дослідження та програмна реалізація системи моніторингу
мережевих принтерів”**

Виконав здобувач вищої освіти

II курсу, групи КІ-20М-1,4

ОПП «Комп’ютерна інженерія»

спеціальності 123 «Комп’ютерна інженерія»

Немировський Д.В.

« ___ » _____ 2021 р.

Керівник проекту

доктор технічних наук, професор

Олексій СМІРНОВ

« ___ » _____ 2021 р.

Рецензент _____

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ
« 6 » вересня 2021 року

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

Немировському Данилі Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи моніторингу мережесих принтерів

2. Керівник роботи Смірнов Олексій Анатолійович, докт. техн. наук, професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи моніторингу мережесих принтерів

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

<u>1. Призначення та область використання.</u>	<u>7. Економічна ефективність розробленої програми.</u>
<u>2. Перегляд аналогічних існуючих систем.</u>	<u>8. Заходи з охорони праці та техніки безпеки</u>
<u>3. Опис і обґрунтування проектних рішень.</u>	<u>9. Висновки.</u>
<u>4. Етапи програмування системи.</u>	
<u>5. Впровадження системи в промислову експлуатацію</u>	
<u>6. Наукова новизна</u>	
<u>6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)</u>	
<u>Наукова новизна</u>	<u>1 аркуш</u>
<u>Структурна схема системи</u>	<u>1 аркуш</u>
<u>Функціональна схема системи</u>	<u>1 аркуш</u>
<u>Діаграма процесів</u>	<u>1 аркуш</u>
<u>Блок-схема алгоритму роботи додатку</u>	<u>2 аркуша</u>
<u>Показники економічної ефективності</u>	<u>1 аркуш</u>

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 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	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Немировський Д.В. Дослідження та програмна реалізація системи моніторингу мережевих принтерів. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи моніторингу мережевих принтерів.

Метою розробки є дослідження та програмна реалізація системи моніторингу мережевих принтерів.

Об'єктом дослідження є процес моніторингу мережевих принтерів.

Предметом дослідження є методи моніторингу мережевих принтерів.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи моніторингу мережевих принтерів.

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

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

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Delphi.

Ключові слова: комп'ютерна інженерія, моніторинг, мережеві принтери

ABSTRACT

Nemyrovskiy D.V. Research and software implementation of network printer monitoring system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work on the second (master's) level of higher education the software which is intended for system of monitoring of network printers is developed.

The purpose of development is research and software implementation of network printer monitoring system.

The object of research is the process of monitoring network printers.

The subject of research is methods of monitoring network printers.

The research methods are based on the methods of computer network theory, methods of mathematical statistics, methods of software development.

The result is a software implementation of a network printer monitoring system.

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

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

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

The program is developed in the Delphi environment.

Keywords: computer engineering, monitoring, network printers

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.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи.....	26
3.2 Розробка структурної схеми	32
3.3 Розробка функціональної схеми.....	35
3.4 Розробка діаграми процесів	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	42
4.2 Захист розробленого програмного забезпечення	67
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	72
6 НАУКОВА НОВИЗНА	75

ВКРМ-123.21.0013.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Немировський Д.Е.			Дослідження та програмна реалізація системи моніторингу мережевих принтерів	Лім.	Аркуш	Аркушів
Перев.		Смірнов О.А.				М	1	117
Н.контр.		Гермак В.С.			ЦНТУ КІ-20М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	76
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	76
7.2 Розрахунок трудомісткості розробки програмної продукції	78
7.3 Визначення чисельності виконавців і планового фонду зарплати	80
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	85
7.5 Визначення собівартості розробки та ціни програмної продукції.	89
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	93
7.7 Визначення експлуатаційних витрат.....	94
7.8 Визначення економічної ефективності програмної продукції.....	95
7.9 Висновок.	97
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	98
8.1 Вступ	98
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	99
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	100
8.4 Розробка заходів з умов поліпшення охорони праці.....	103
8.5 Розрахункова частина	104
8.6 Висновки до розділу.....	106
9 ОСНОВНІ ВИСНОВКИ.....	107
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	109

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	- база даних
БДРП	– багатофункціональні друкуючі пристрої
БСА	– блок-схема алгоритму
ВКРМ	– випускна кваліфікаційна робота магістра
ДРП	– друкуючі пристрої
ДСТУ	– Державні стандарти України
ДС	– диск С
ЗОТ	– засоби обчислювальної техніки
LAN	– локальна мережа типу LAN
МЕТОЗ	- методичне забезпечення
НСВ	– несанкціоноване використання
НСД	– несанкціонований доступ
ОС	– операційна система
ОРЗ	– організаційне забезпечення
ПЗ	– програмне забезпечення
ПК	– персональний комп'ютер
СП	– системна подія
СУБД	– система управління базами даних
ТЗ	– технічне завдання

ВСТУП

Актуальність теми. Мабуть самими неконтрольованими та невидимими витратами в організаціях, які використовують сучасні комп'ютерні технології та електронний документообіг, є витрати на друк документів (включаючи витрати на папір, апаратне обладнання та тонер). Аналіз цього питання, проведений в період переддипломної практики, виявив наступні результати: вартість виготовлення документів складає біля 40 % трудових витрат відповідних відділів фірми, які використовують документообіг в електронному вигляді та роздруковують документи на паперових носіях інформації.

Багато адміністраторів і керівників неоднократно зіткнулися з необхідністю контролю ефективності використання принтерів в організації і намагалися провести такий контроль власними силами. Але дуже непросто відстежити тих співробітників, які постійно витрачають запаси паперу і тонера, призначені для всього відділу, на друкування паперів для власних потреб.

Якщо організація надає обчислювальні послуги по роздрукуванню текстових і інших документів, то перед нею неминуче встане проблема тарифікації, безпосередньо пов'язана з автоматичним обліком ресурсів. Особливо організація питання моніторингу ускладнюється, якщо принтер не один і більшість з них знаходиться в мережі, локальній чи корпоративній.

Проте на сьогодні організація моніторингу роботи принтерів все ще залишається на етапі мінімальної автоматизації. Про це свідчать результати проведеного під час переддипломної практики експрес-аналізу програмного ринку країни. Тому розробка програмного забезпечення системи, яка відстежувала б, хто, коли і скільки надрукував, яку до того ж можна буде інтегрувати в систему контролю організації, є задачею дійсно актуальною. Адже це не тільки позбавило б адміністраторів від зайвого діалогу з власними співробітниками та клієнтами, але і надало б фірмі сучаснішого вигляду.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Таким чином, враховуючи вищезначене, розробка автоматизованих комп'ютерних систем, які забезпечать моніторинг роботи принтерів в мережі на основі використання сучасних технологій, на сьогодні є питанням надзвичайно перспективним та нагальним. Адже такі системи можуть будуть корисними та незамінними в будь-яких галузях народного господарства країни та будь-якій сфері діяльності людини.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи моніторингу мережевих принтерів.

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

- Огляд існуючих систем моніторингу мережевих принтерів.
- Дослідження системи моніторингу мережевих принтерів.
- Програмна реалізація системи моніторингу мережевих принтерів.

Об'єктом дослідження є процес моніторингу мережевих принтерів.

Предметом дослідження є методи моніторингу мережевих принтерів.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод моніторингу мережевих принтерів.
- Розроблено вітчизняний продукт моніторингу мережевих принтерів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі моніторингу мережевих принтерів.

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.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи моніторингу мережевих принтерів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Предметом реалізації випускної кваліфікаційної роботи магістра (ВКРМ), згідно технічного завдання (ТЗ), є розробка програмного забезпечення моніторингу мережевих принтерів (в подальшому – система). Розглянемо призначення системи, що підлягає розробці, та визначимо область її використання.

На сьогоднішній день всебічне скорочення витрат стоїть у ряді пріоритетних завдань менеджменту будь-якого суб'єкта економічної діяльності, незалежно від форми власності. Це один із засадничих способів оптимізації бізнес-процесів, підтримка конкурентоспроможності продукту або послуги і збільшення прибутків.

Аналітики компанії Al I Associates провели оцінку витрат на виготовлення документів з використанням принтерів в організаціях різних галузей, в результаті були отримані приголомшуючі дані. В середньому витрати на друк різних документів складають близько 1% від загальнокорпоративного обороту. Інші дослідження дають ще значиміші показники. Так, експерти IDC і Gartner вважають, що згадані процеси «з'їдають» в середньому від 3 до 5% обороту, що призводить до досить великих втрат компаній навіть з багатомільйонними операційними об'ємами. Проаналізувавши структуру внутрішньоофісних витрат, аналітики зробили висновок, що їх можна скоротити майже на 30%, якщо вибрати розумне рішення проблеми.

Серйозна економія коштів досяжна за рахунок застосування спеціальних технологій моніторингу і контролю, використання відповідних технологій в офісах з великим штатом співробітників забезпечує найкращий результат. Якщо дрібні фірми в змозі «в ручному режимі» організувати контроль витрат в цьому

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Таким чином, до функціоналу системи, яка підлягає розробці, можна віднести наступне:

- дозволить централізовано контролювати всі ДРП організації;
- допоможе відстежувати і присікати спроби використання офісних ДРП в особистих цілях співробітників;
- реально допоможе скоротити витрати на придбання матеріалів: паперу, тонера;
- буде працювати в автоматичному режимі і не вимагатиме додаткових зусиль та спеціальних знань з боку користувача;
- буде гранично простою у використанні, мати високу ступінь мобільності та адаптивності і тому дозволить почати моніторинг роботи ДРП вже через невеликий проміжок часу після установки;
- система дозволить формувати велику кількість звітів і діаграм, що показують ефективність використання контролюємих ДРП за будь-який термін часу;
- наявність спеціальних засобів для автоматизації процесу створення і відправлення звітів про використання ДРП дозволить відповідальному персоналу одержати реальну картину роботи ДРП в будь-який термін часу; при цьому буде надаватись можливість перегляду образів роздрукованих документів;
- програма забезпечить точне визначення числа роздрукованих сторінок і кількості копій.

1.2 Область застосування

Враховуючи вищезначене, можемо визначити майбутню систему, як складний багатофункціональний програмно-апаратний комплекс, відмінними особливостями якого будуть:

- багатоланкова архітектура, що дозволяє установку всіх модулів програми на різних комп'ютерах;
- віддалене адміністрування програми з будь-якого комп'ютера в мережі;

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- можливість адміністрування і моніторингу відразу з декількох робочих станцій в мережі;
- віддалена установка і видалення агентів;
- два режими роботи агентів: на комп'ютерах користувачів і на принт-серверах;
- підтримка SQLite, MS Access, MS SQL, MYSQL, POSTGRESQL, Oracle як сховища даних;
- перегляд образів роздрукованих документів;
- велика кількість зумовлених звітів, що дозволяють аналізувати зібрану інформацію в різних розрізах;
- підтримка Active Directory;
- сумісність з Terminal Services;
- сумісність з Microsoft Cluster Server.

Тому будемо орієнтувати систему, яка підлягає розробці в процесі виконання ВКРМ, на впровадження в будь-якій сфері діяльності людини, не залежно від форми власності, де використовується:

- друкування паперових носіїв документів для власних потреб;
- друкування паперових носіїв документів в комерційних цілях на замовлення клієнтів;
- контроль ефективності використання матеріалів для ДРП: паперу, тонера;
- контроль ефективності використання ДРП співробітниками організації/установи/тощо;
- виявлення співробітників, які постійно витрачають запаси паперу і тонера, призначені для всього відділу, на власні потреби.

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

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

Згідно технічного завдання, в процесі виконання кваліфікаційної магістерської роботи передбачена розробка програмного забезпечення моніторингу мережевих ДРП. Система може використовуватись в якості незалежного програмного забезпечення, а при необхідності – може ввійти складовим модулем до більш потужної системи. Проте для розробки дійсно дієспроможної системи, яка буде виконувати всі означені ТЗ функції, необхідно виявити та проаналізувати наявні системи – аналоги.

Проведений в процесі переддипломної практики аналіз ринку програмних продуктів країни показав наступне: на ринку програмних продуктів країни системи аналогічного спрямування фактично відсутні; розрізнені програми не дозволяють користувачу проводити одночасний контроль та обробку інформації, що не є бажаним фактором. Тому розглянемо найбільш поширені системи – аналоги закордонного виробника.

Система Printer Activity Monitor – це програмний продукт, призначений для контролю над ефективністю використання принтерів. За допомогою Printer Activity Monitor можна легко визначити:

- які документи роздруковувалися на кожному з принтерів, що є предметом контролю системного адміністратора/відповідальної особи/керівника;
- хто з користувачів найактивніше друкує;
- кількість надрукованих сторінок;
- проглядати образи роздрукованих документів і так далі.

Програма дозволяє визначити такі атрибути роздрукованих документів:

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- кількість сторінок;
- кількість копій;
- ім'я і модель принтера;
- розмір паперу;
- колір паперу;
- орієнтацію листа паперу;
- назву додатку, з якого здійснювався друк.

До переваг розглянутої системи віднесемо наступне:

- програма дозволяє централізовано контролювати всі принтери організації;
- допомагає відстежувати і присікати спроби використання офісних принтерів в особистих цілях співробітників;
- реально допомагає скорочувати витрати організації на друк документів;
- працює в автоматичному режимі і не вимагає додаткових зусиль з боку користувача;
- програма гранично проста у використанні і дозволяє почати контроль принтерів вже через хвилину після установки;
- дозволяє будувати велику кількість звітів і діаграм, що показують ефективність використання кожного принтера в мережі;
- має спеціальні засоби для автоматизації процесу створення і відправки звітів про використання принтерів відповідальному персоналу;
- дозволяє проглядати образи роздрукованих документів;
- точно визначає число роздрукованих сторінок і кількість копій.

Усі вищезначені функції доцільно ввести до функціоналу нашої системи, адже вони дійсно забезпечать ефективність її роботи.

Щодо структури побудови системи Printer Activity Monitor, слід відзначити багатоланкову архітектуру, що дозволяє установку всіх модулів програми на різних комп'ютерах та віддалене адміністрування програми з будь-якого комп'ютера в мережі. Окрім цього, система Printer Activity Monitor

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

забезпечує можливість адміністрування і моніторингу (в разі потреби) відразу з декількох робочих станцій в мережі та надає два режими роботи агентів: на комп'ютерах користувачів і на принт-серверах.

Підтримка SQLite, MS Access, MS SQL, MYSQL, POSTGRESQL, Oracle, як сховища даних; перегляд образів роздрукованих документів; забезпечення формування та роздрукування великої кількості зумовлених звітів, які дозволяють аналізувати зібрану інформацію в різних розрізах; підтримка Active Directory; сумісність з Terminal Services та з Microsoft Cluster Server надають розглянутій системі особливої привабливості та характеризують її, як дійсно багатофункціональну систему.

Разом з тим, система Printer Activity Monitor не пред'являє особливих функціональних вимог, адже для її успішного функціонування достатньо: ОС Windows; 512Mb RAM; 10000 Кб дискового простору.

Програма Printer Activity Monitor складається з наступних модулів:

– Агент – системний сервіс, який постійно знаходиться в пам'яті і збирає інформацію про роздруковані завдання. Агент встановлюється або на принт-сервері, або на робочій станції користувачів.

– Центр зберігання даних (Дата – Центр) – системний сервіс, який отримує дані від агентів і акумулює їх в базу даних (рисунок 2.1).

– Консоль управління – оснащення для Microsoft Management Console, призначена для адміністрування програми і аудиту друку.

Всі компоненти можуть встановлюватися на різних комп'ютерах в локальній мережі і конфігуруватися дистанційно. Декілька фахівців можуть адмініструвати одну програму з різних комп'ютерів в залежності від прав доступу.

Центр зберігання даних може бути встановлений на будь-якому комп'ютері в локальній мережі.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

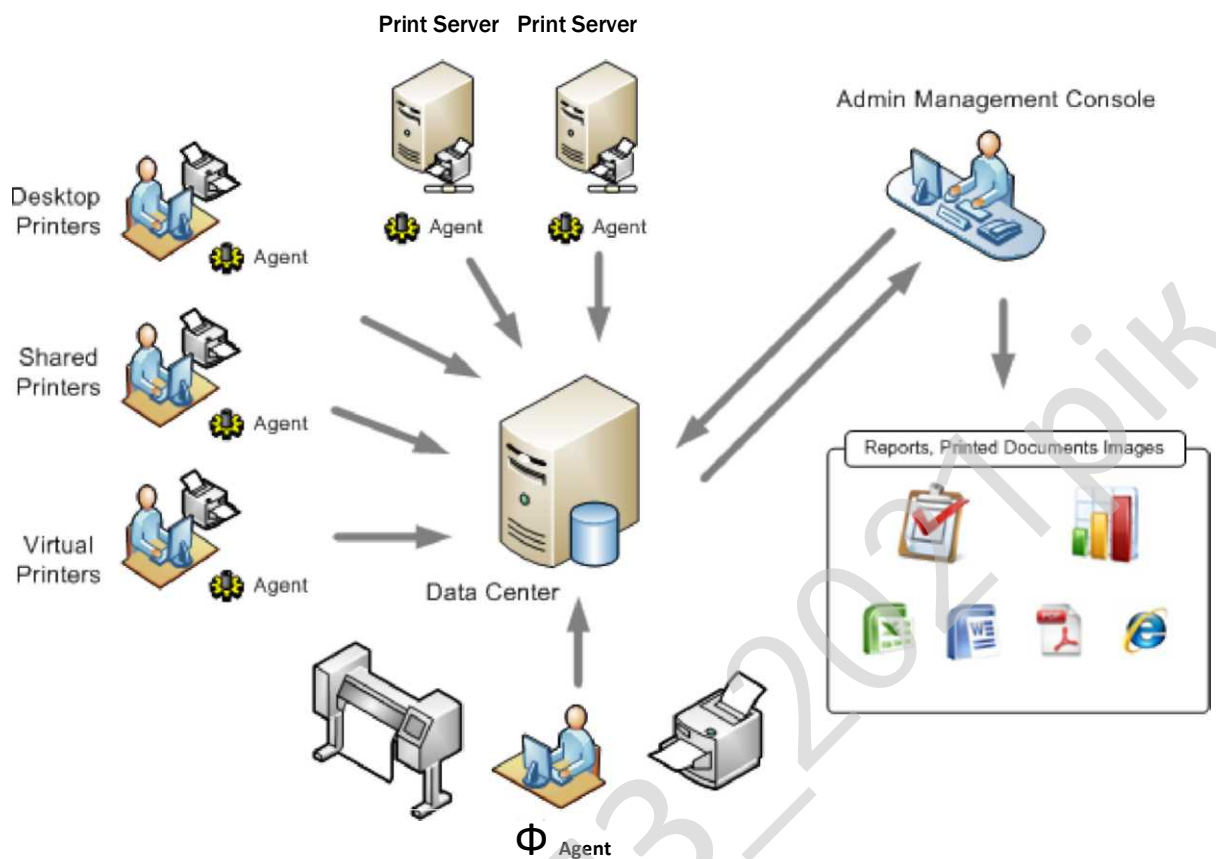


Рисунок 2.1 – Структурна схема центру зберігання даних (Дата – Центр)

Агенти моніторингу встановлюються на комп'ютери у вигляді сервісу, що займається збиранням інформації про всі завдання, відправлені на друк. Отриману інформацію агент пересилає центру зберігання даних.

Управління Центром зберігання даних відбувається за допомогою консолі управління, яка є оснащенням для Microsoft Management Console. Консоль управління може бути встановлена як на одному комп'ютері з Центром зберігання даних, так і на віддаленій машині адміністратора.

Система Print Manager Plus дозволяє користувачу квотувати і відстежувати використання ДРП в локальній мережі. Тепер у бюджеті організації, яка використовує цю систему, не буде втрат із-за неконтрольованого друку. А можна заощадити, відстежуючи ці витрати, дуже багато (про це свідчать дані

Впровадження PrintMonitor у будь-якій організації дозволить отримати консолідовану звітність по всіх принтерах і принт-серверах. Це набагато зручніше, ніж використовувати вбудовані в принтери лічильники.

Звіти PrintMonitor дозволять скоротити витрати і планувати бюджет по витратних матеріалах. Персонал стане свідомішим – люди думатимуть, перш, ніж що-небудь друкувати для власних потреб, завдаючи цим збитки власнику організації.

Завдяки зібраній статистиці можна провести оптимізацію навантаження на друкуючі пристрої. У керівництва буде повна інформація про те, скільки сторінок надрукував кожен користувач на кожному з ДРП, включаючи і інформацію по кількості копій одного документа. І, звичайно ж, буде детальна статистика по кожному роздрукованому документу.

Третьою системою, розгляд та аналіз якої ми проведемо, буде **професійна (Pro), стандартна (Standard) і безкоштовна Lite-версія.**

В процесі експлуатації вона забезпечує збирання:

- детальної статистики, включаючи розмір паперу;
- ім'я користувача і ім'я комп'ютера, з якого друкували;
- точний підрахунок копій (при друці декількох копій документа);
- контроль/моніторинг плотерів, мережевих "розшарених" і локальних принтерів;
- повний набір звітів (є вбудований редактор звітів);
- можливість інтеграції із зовнішнім програмним забезпеченням.

Програма Print Censor призначена для ведення статистики використання принтерів.

Ця програма створена для полегшення управління всіма принтерами. З її допомогою можна задавати обмеження для принтерів і для окремих користувачів, наприклад – по числу надрукованих сторінок і користувачі не матимуть можливості друкувати більше заданого обмеження.

За допомогою Print Censor (розробник UsefulSoft) відповідальна

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

особа/системний адміністратор завжди знатимете необхідну інформацію про кожне надруковане завдання:

- ім'я користувача;
- ім'я комп'ютера;
- ім'я документа;
- дату і час друку;
- число сторінок і число копій;
- об'єм документа;
- пріоритет друку;
- розмір паперу, кольоровий або чорно-білий;
- чи був задіяний режим дуплексу (друк на обох сторонах листа).

Більш того, програма надає доступ до перегляду вмісту завдань друку як в черзі, так і для всіх вже надрукованих завдань. Є можливість копіювати завдання з одного принтера на інший, визначити вартість всіх надрукованих завдань і багато чого іншого.

Проте усі розглянуті багатофункціональні програмні продукти мають один спільний, але досить суттєвий недолік: вони коштують досить дорого і тому, враховуючи економічний стан нашої країни, є недосяжними для багатьох потенціальних користувачів.

Зважаючи на ці фактори, робимо наступний висновок: в нашій країні існує великий потенціал для розробки систем даного класу та спрямування. Тому розробка, запланована ТЗ на виконання ВКРМ, є доцільною та своєчасною.

Одночасно слід зазначити, що для побудови нашої системи необхідно запозичити з розглянутих систем-аналогів наступне:

- структуру алгоритмів для розробки власних альтернативних алгоритмів;
- принцип модульної структури побудови системи;
- введення до складу ПЗ системи потужного українського інтерфейсу користувача;
- забезпечення системи високою ступінню інформативності.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.2 Обґрунтування вибору засобів побудови системи та мови програмування

Система, яка підлягає розробці в процесі виконання ВКРМ, буде спроектована у відповідності з вимогами концепції побудови систем аналогічного класу та спрямування, та забезпечить:

- вирішення задач моніторингу в мережі роботи наявних друкуючих пристроїв;
- обробку одержаних в результаті проведеного моніторингу даних;
- збереження результату моніторингу в відповідних спеціальних log-файлах та базі даних (БД).

По суті, система буде багаторівневою системою реального часу, оскільки обробку даних необхідно здійснювати за позначками реального часу.

При розробці програмного забезпечення системи використаємо наступні принципи побудови систем аналогічного класу та спрямування:

- Основні принципи організації збирання і обробки інформації. Основним є вимога загального перетину інформаційного простору для всіх суб'єктів мережі. Реалізуємо це положення у вигляді єдиної інтегрованої мережі збирання, накопичення і обробки інформації про результати проведення моніторингу мережевих ДРП.

Всі відповідальні особи підприємства/організації/установи (в подальшому – замовник) будуть мати авторизований доступ до вихідної інформації завдяки організованому нами режиму “Перегляд звіту”. При побудові мереж використаємо стандартні методи технології «Клієнт-Сервер».

- Основні принципи організації контролю. При розробці ПЗ будуть введені процедури моніторингу кожного ДРП мережі (чи вибірково – на розсуд керівництва замовника) безперервно чи з заданою користувачем періодичністю. Це важливо не тільки з технічної точки зору (мається на увазі сто відсоткове забезпечення реалізації функціональних вимог до системи), але також і з точки

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

зору економічних і правових взаємовідносин працедавця та робітників.

Окрім цього, розроблена в процесі виконання ВКРМ система, буде:

- виконувати задачу контролю в автоматичному режимі;
- забезпечить власну невидимість при роботі в мережі;
- виконання, окрім основних, наступних додаткових функцій: наявність функції реєстрації користувачів для входу в систему; формування звітів;
- передача звітів;
- пошук/роздрукування звітів з БД сервера по заданих реквізитах та за означений користувачем термін часу.

Таке представлення системи є зручним для виконання визначених згідно ТЗ функцій роботи системи.

Для побудови системи необхідно визначитись з переліком засобів, які дозволять вирішити проектну задачу. До них віднесемо:

- вибір типу операційної системи;
- сучасних інформаційних технологій;
- мови програмування;
- технології програмування.

Вибір типу операційної системи напрямки пов'язаний з рішенням задачі керування робочим обладнанням в режимі жорсткого реального часу. Це вимушує робити вибір між спеціалізованими ОС реального часу і деякими ОС загального призначення, які надають аналогічний сервіс (наприклад – Unix System Release 4).

Наявність серверів БД, які будуть складовими структури майбутньої системи, вимагає високої надійності і продуктивності, що відкидає системи класу DOS.

Тому в якості платформи для розробки системи ми обираємо ОС Windows, яка задовольняє вищезначені вимоги і, в порівнянні з іншими ОС, має наступні переваги:

- простоту в експлуатації;

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

- звичність в роботі для переважної кількості користувачів;
- надійність, підтримку переважної кількості сучасних комп'ютерних технологій;
- економію системних ресурсів (комірки пам'яті, які не використовуються, переміщуються в файл підкачки), тощо.

Програмне забезпечення системи планується розробити на мові програмування DELPHI. Цей вибір обумовлений наступними причинами:

- об'єктно-орієнтована мова програмування DELPHI дозволяє зручно абстрагувати об'єкт проектування – систему та її елементи і організувати класи загального призначення із контролем доступу до даних за рахунок введення захищених та відкритих членів класу, що ефективно моделюють проектування складних систем керування і роботу реальних об'єктів з використанням передачі інформації між ними;

- у мові DELPHI представлені зручні засоби для роботи як на високому рівні, так і на низькому в той час, як інші сучасні мови надають перевагу одному напрямку (або високий рівень із створенням зручного інтерфейсу, або низький рівень із програмуванням мережного зв'язку з недостатнім забезпеченням функцій інтерфейсу).

Таким чином, вибір мови програмування DELPHI є доцільним, оскільки вона має засоби дуже зручного опису елементів об'єкта та зв'язків між ними та зовнішнім середовищем.

При побудові ПЗ будуть використані стандартні модулі, які виконують наступні функції:

- Menus – модуль, що забезпечує (обслуговує) всі об'єкти та процедури для системи меню DELPHI, а також випадючі меню та активні елементи рядка статусу, роботу з меню;
- Messages, Dialog – модулі, що забезпечують процедури і функції виклику діалогових вікон;
- SysUtils – модуль, що забезпечує процедури та функції нескладних

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

математичних функцій, функції перетворення з одного типу в інший; деякі тригонометричні та гіперболічні функції; процедури та функції перетворення дат і часу, роботи з рядками та інше;

- Classes – процедури та функції для роботи з класами;

- Variants – процедури та функції перетворення типу Variant в заданий тип;

- Forms – процедури та функції по роботі з формами, що підключають та забезпечують можливість використання стандартних бібліотечних функцій, а саме: підключення діалогових вікон, меню користувача, організація функції натиснення кнопок для отримання з клавіатури режиму роботи системи в поточний момент часу, виведення на екран файлу з інформацією про режим роботи системи, підготовки екрану до діалогу з користувачем, тощо.

Використання стандартних бібліотек також дозволить полегшити побудову деяких частин програми та скоротити обсяг програмного коду.

Організація процесу керування обробкою даних повинна відповідати та проводитись згідно діючих державних стандартів, які пред'являють ряд загальних та специфічних вимог, а саме:

- повинні використовуватись сучасні методи організації та керування, включаючи системно-структурний аналіз та моделювання процесів, економічно-математичні методи;

- повинні використовуватись раціональні параметричні та типорозмірні ряди виробів електронної техніки та засобів контролю;

- повинна бути орієнтація на оптимальний, для конкретних умов, рівень автоматизації поставленої задачі та процесу його контролю: збирання, підготовки, обробки, передача та представлення інформації.

Для зручності розробки та впровадження, процес побудови системи доцільно, згідно діючих стандартів (ДСТУ 23501.0-92... ДСТУ 23501.4-92, ДСТУ 23501.5-94, ДСТУ 23501.9-94, ДСТУ 25501.97 – ДСТУ 23501.6-97), розподілити на три етапи:

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- виділення основних функцій майбутньої системи;
- визначення задач кожної функції;
- визначення підзадач в кожній задачі.

При розробці програмного забезпечення планується організація архітектури ПЗ системи згідно модульної структури: основна (керуюча) програма та ряд підпрограм загального та спеціального призначення.

Підпрограми загального призначення необхідні для реалізації в системі функцій, які вирішує ПК:

- інформаційний пошук;
- кодування;
- контроль;
- перетворення інформації;
- формування та представлення вихідних даних для програм керування та контролю;
- оформлення визначеного пакету звітної документації.

Підпрограми спеціального призначення будуть виконувати функції контролю некоректних дій користувача та оптимізації нестандартних ситуацій в роботі системи. Структурними елементами підпрограм другого типу є модулі програми.

Побудова системи повинна відповідати шести основним принципам:

- композиції;
- єдності;
- відкритості;
- максимальної універсальності;
- інформаційної єдності;
- інваріантності.

Розглянемо більш детально кожен з них.

Принцип композиції складається з того, що система, яка підлягає розробці, створюється на базі вже існуючих систем, які були розглянуті в цьому розділі,

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

систем.

Принцип інваріантності означає, що система, яка підлягає розробці, з точки зору її використання на інших ЗОТ, повинна бути універсальною або типовою. Вирішення цієї задачі планується введенням в ПЗ універсальних або спеціальних модулів.

Наявність достатньої кількості такого типу модулів та ППП дозволить забезпечити закінченість рішення задач системою.

З методологічної точки зору, задачу розробки системи доцільно, використавши методику декомпозиції, розділити на ряд взаємозалежних частин, виділивши основні компоненти. Таких частин доцільно організувати п'ять.

1. Головна програма (Операції високого рівня).

2. Частина 1: Моніторинг ДРП в мережі.

3. Частина 2: Читання, керування процесом обробки та обробка результатів моніторингу роботи ДРП в мережі. Запис одержаних результатів в локальну БД.

4. Частина 3: Передача інформації від модуля моніторингу на сервер.

5. Частина 4: Читання, обробка та запис в серверну БД.

Виконана декомпозиція проектної задачі дозволить значно полегшити задачу побудови структури майбутньої системи, визначення функціональних блоків та їх взаємодії.

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 моніторингу мережевих принтерів.

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

						ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			24

негативних якостей. Результати аналізу врахувати в подальших розробках;

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Предметом розробки ВКРМ є програмне забезпечення моніторингу мережевих принтерів.

Як результат роботи очікується система моніторингу мережевих друкуючих пристроїв, яка може бути використана як незалежна система або може бути інтегрованою для роботи в іншій системі аналогічного класу та спрямування.

Основною задачею розробки ПЗ є створення програмної моделі системи, яка в процесі промислової експлуатації буде виконувати наступні функції:

- проводити моніторинг мережевих ДРП в фоновому режимі з заданою періодичністю 500 мс (інтервал моніторингу може бути змінений замовником);
- визначити необхідну інформацію про кожне надруковане завдання:
 - ім'я користувача;
 - ім'я комп'ютера;
 - ім'я документа;
 - дату і час друку;
 - число сторінок і число копій;
 - об'єм документа;
 - пріоритет виконання завдання;
 - розмір паперу;
 - папір кольоровий чи чорно-білий;
 - чи був задіяний режим дуплексу (друк на обох сторонах листа);
 - надавати можливість відповідальній особі/системному адміністратору проглянути вміст завдань друку як в черзі, так і для всіх вже надрукованих завдань;

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

(клієнтська частина) та в мережевій БД (серверна частина). Враховуючи специфіку задач, що буде виконувати система, найбільш доцільно використати реляційні БД. В такій БД об'єкти та взаємозв'язок між ними представляються у вигляді плоских прямокутних таблиць з рядів та стовпчиків.

Для розробки ПЗ системи нами обрано мову програмування DELPHI. DELPHI не має власного формату таблиць, але має засоби, які дозволяють працювати з багатьма зовнішніми форматами, тому необхідно вибрати формат таблиць бази даних. Найбільш розповсюдженим та розвиненим є формат Paradox.

Таблиці Paradox утримують достатньо велике число полів, підтримують цілісність посилань, автоматично перевіряють дані, що вводяться, на сумісність по типу і підтримують парольний захист даних. Основні розширення таблиць БД Paradox, що будуть використані при розробці ПЗ, наводимо в таблиці 3.1.

Таблиця 3.1 – Файли таблиць БД Paradox

Розширення файла	Зміст файла
*.db	Дані таблиці.
*.mb	Великі двійкові дані (BL.OB, BinaryLargeObject).
*.px	Ключ (головний індекс).
*.xq*i*.uq*	Індекси.
*.val	Параметри для перевірки типів даних, що вводяться, цілісності посилань.
*.net	Використовується для контролю паролічного доступу.

Додаток, створений в процесі реалізації дипломного проекту за допомогою DELPHI, здійснює доступ до БД через спеціальний процесор баз даних ВДЕ. ВДЕ – це набір драйверів і динамічно приєднаних бібліотек (файл *.dll), які забезпечують доступ до даних. Процесор ВДЕ дозволить найбільш ефективно організувати роботу з таблицями БД Paradox.

Для успішної роботи з БД необхідні програмні засоби, які б забезпечували

доступ до потрібної інформації та виконання будь-яких дій з нею. Для рішення цієї задачі використаємо СУБД. Всі СУБД поділяються на дві групи: локальні та мережні.

В нашому випадку більш доцільно використати мережну СУБД, оскільки в процесі роботи система буде використовуватись режим мультидоступа користувачів, що будуть постійно взаємодіяти з однією БД за допомогою технології “клієнт-сервер”. В розробці СУБД немає необхідності, оскільки можна використати одну з стандартних СУБД: InterBase, Oracle, Microsoft SOL Server, тощо.

Враховуючи, що система буде мати складну ієрархічну структуру, при розробці ПЗ необхідно забезпечити організацію та виконання наступних системних програмних задач:

- реалізація роботи з мережею за допомогою протоколу TCP/IP;
- реалізація роботи з зовнішнім периферійним обладнанням: принтери, плотери, багатофункціональні друкуючі пристрої (БДРП), тощо;
- реалізація введення/виведення інформації;
- реалізація роботи з файловою системою БД Paradox;
- реалізація функцій часу;
- реалізація масштабування, виведення керуючих сигналів, перетворення кодів.

Багатофункціональність системи, що підлягає розробці, забезпечить локалізацію різноманітних задач, починаючи з налаштування і конфігурування, комунікаційних функцій і задач контролю, і закінчуючи системою розробки користувачем своїх елементів інтерфейсу і методик розрахунків в рамках системи, застосування у всіх цих задачах єдиного підходу, єдиної інформаційної бази і інструментів розробки.

В комп'ютерних системах моніторингу мережних друкуючих пристроїв, як це було визначено при розгляді існуючих систем-аналогів, апаратні засоби та програмне забезпечення існують в формі неподільного апартно-програмного

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

комплексу, який виконує комплекс визначених функцій. Тому одною з самих складних задач, що підлягають вирішенню в процесі розробки програмного забезпечення системи, є задача оптимального розподілу функцій контролера між апаратними засобами та програмним забезпеченням.

Враховуючи зазначене вище, весь цикл розробки ПЗ та організації системи доцільно розподілити, використавши методику системного проектування та декомпозиції, на послідовність трьох фаз проектування:

- а) аналіз задачі і вибір апаратних засобів;
- б) розробка системного та прикладного програмного забезпечення;
- в) комплектування апаратних засобів та ПЗ в прототипі системи та її налагодження.

Визначимо вимоги до архітектури майбутньої системи. Вибір виконуваних функцій і керування режимами відображення плануємо здійснювати через меню і кнопки керування. До функціоналу системи необхідно ввести наступні режими роботи:

– моніторингу – для відображення параметрів, які контролюються і враховуються системою по точках моніторингу (ПК користувачів, що працюють з ДРП); відображення буде відбуватись у вигляді таблиці;

– автоматичне формування звітів одержаних результатів моніторингу роботи ДРП по заданих шаблонах, редагування та виведення звітів на екран (або на друк);

– адміністратора – для визначення списку користувачів системи і списку задач, з якими користувачі можуть працювати, налаштування IP-адреси сервера, очищення БД, архівація БД (в разі необхідності);

– служба експорту-імпорту для організації обміну даними по моніторингу роботи мережних ДРП;

– інтерактивний режим – надання можливості системному адміністратору/відповідальній особі/користувачу одержати результати проведеного моніторингу та, зробивши відповідні висновки, виконати необхідні

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

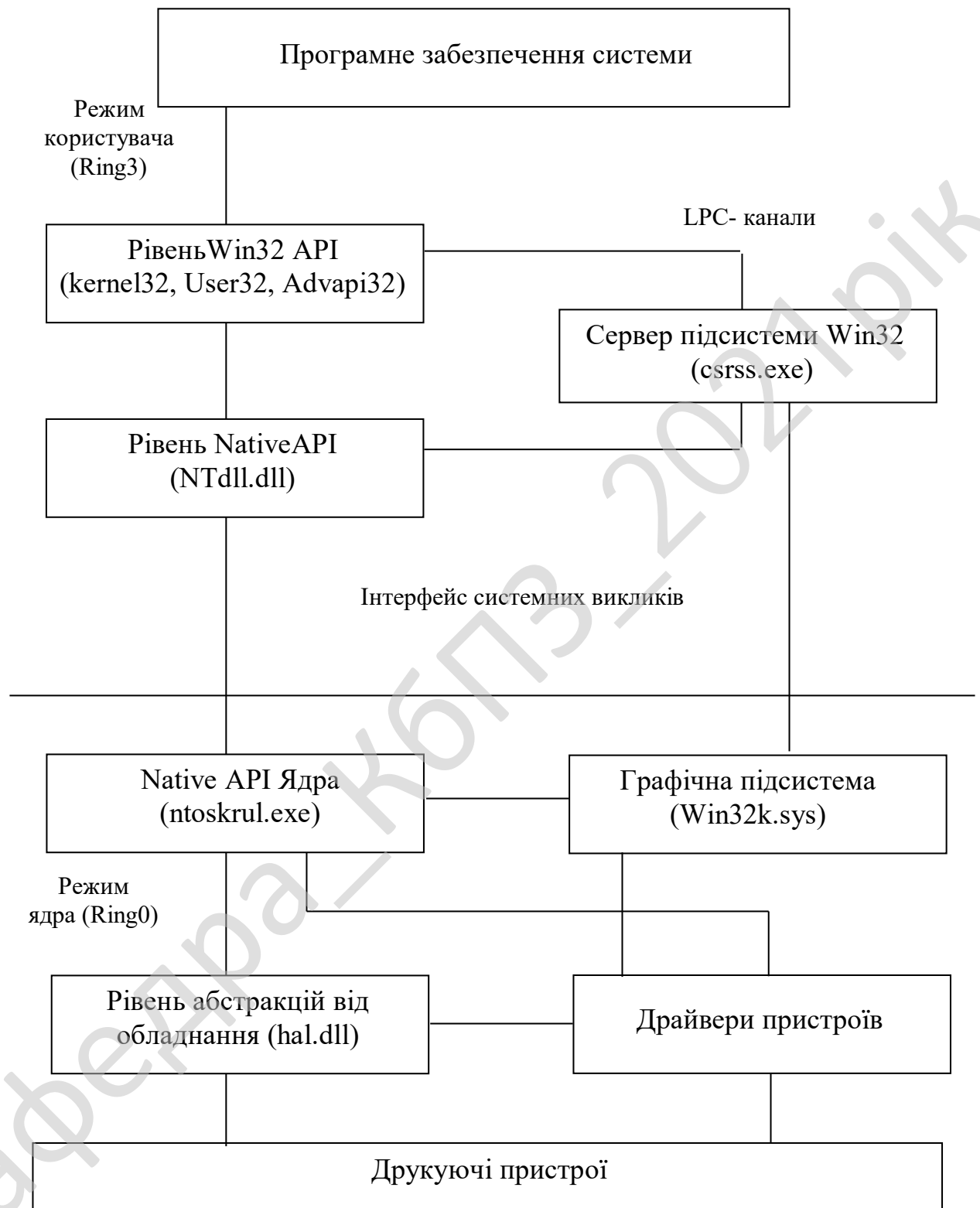


Рисунок 3.1 – Структурна схема системи

Передача даних проводиться не однооментно, а дейтаграмами (блоками, що проходять один за одним) по протоколу TCP/IP. Обробник подій OnRead компоненти TClientSocket активізується при передачі кожного блоку даних, тому йому необхідна інформація про те, що іде приймання даних першого блоку чи другого, чи всіх останніх блоків. В першому випадку обробник створює тимчасовий файл для збереження даних, в другому – відкриває цей файл та приєднує ці дані в його кінець.

Низькорівневі API-функції сокетів дозволять БД серверу одержати прапорці пакету, серед яких є прапорець, що сигналізує про закінчення передачі даних. Однак, на компонентному рівні DELPHI аналіз цих прапорців недоступний.

Тому першу проблему вирішимо за рахунок використання глобальної змінної логічного типу, значення якої не руйнується після завершення роботи обробника. Для визначення кінця передачі використаємо програмно реалізований таймер, який запускається перед завершенням роботи обробника і зупиняється на початку роботи. Після закінчення встановленого на ньому часу – кінець передачі даних. Враховуючи вищезначене, проведемо визначення додаткових функцій, виконання яких повинна забезпечити система:

- зв'язок з зовнішніми приладами і керування ними;
- самодіагностика та діагностика системи;
- формування та ведення архівів (БД) в автоматичному режимі;
- формування звітів за будь-який термін часу та їх передача по мережі від клієнтської частини до серверної частини;
- контроль роботи користувача: виявлення його некоректних дій з послідуною забороною на роботу з системою (в разі необхідності);
- контроль повноти введення інформації на рівні моніторингу;
- організація та функціонування діалогового режиму, інтерфейсу користувача, режиму мультидоступу.

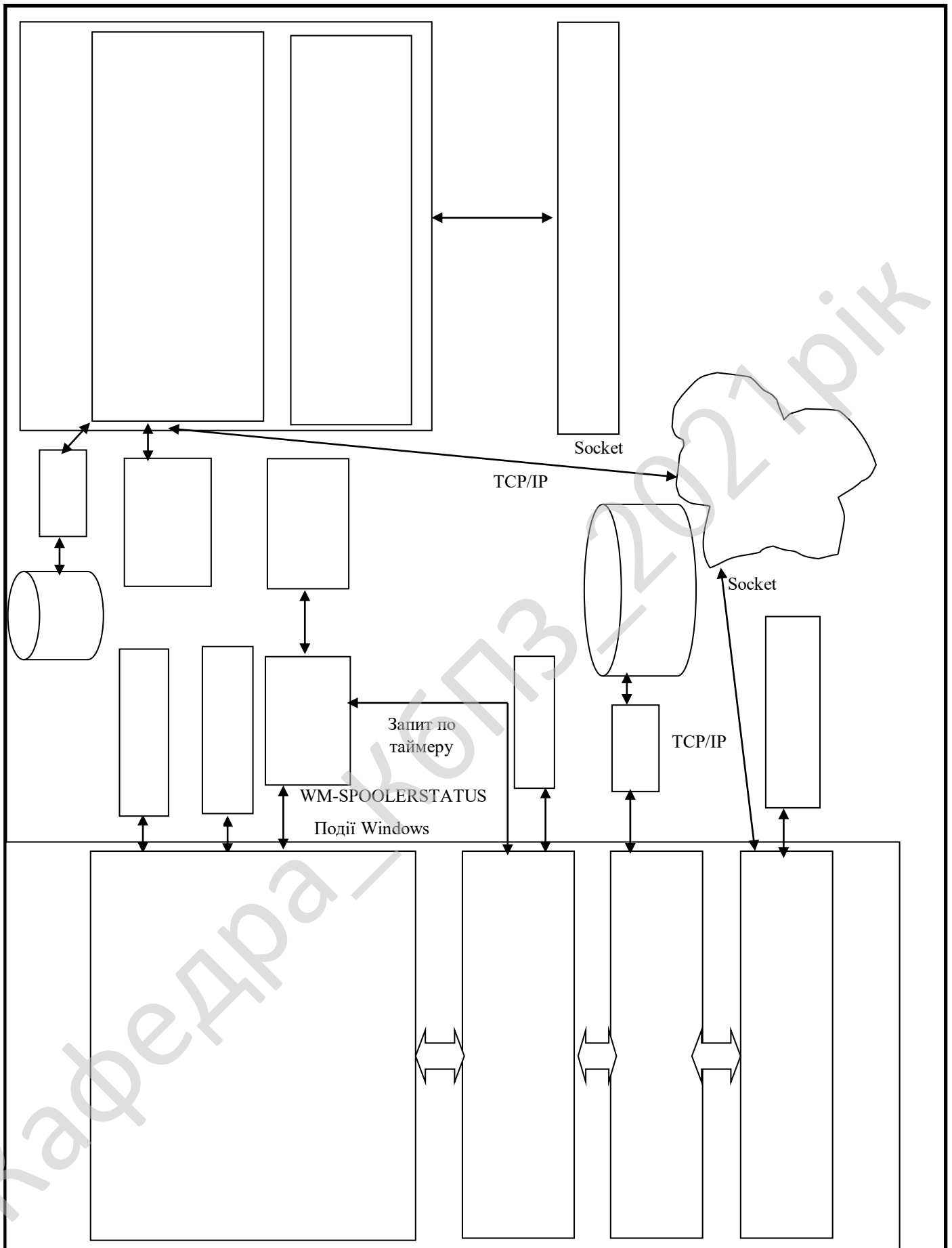


Рисунок 3.3 – Функціональна схема системи

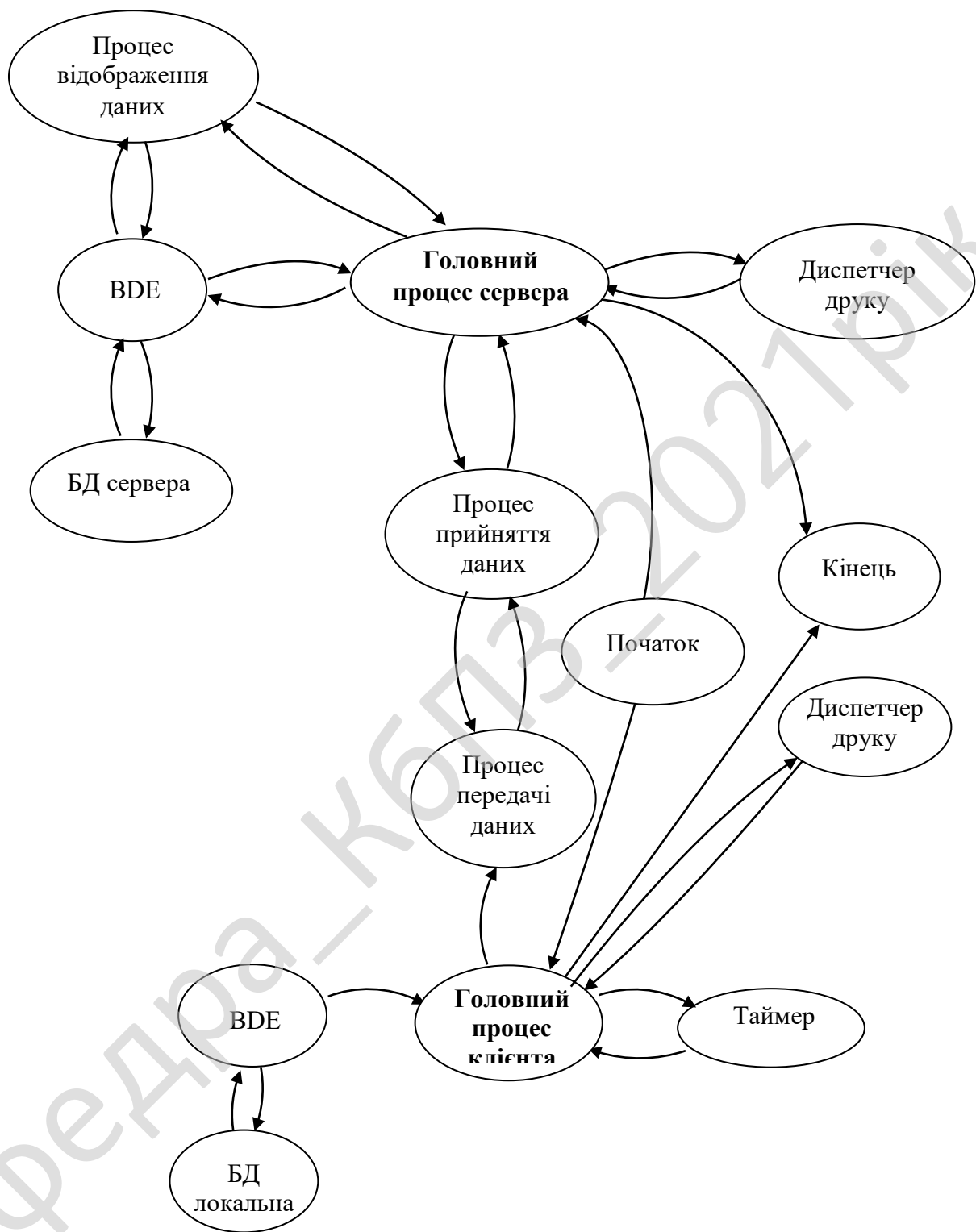


Рисунок 3.3 – Діаграма взаємодії процесів в системі

Головний процес клієнтської частини забезпечить виконання моніторингу вибраних друкуючих пристроїв за допомогою таймеру та системних подій ОС

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0013.00.00.ПЗ

Арк.

40

Windows. Використавши графічний інтерфейс, користувач може в будь-який час і за будь-який термін часу вивести додаткові дані про друкуючий пристрій/друкуючі пристрої, які підлягають моніторингу: список шрифтів, які підтримує друкуючий пристрій; метрику пристрою; драйвера пристрою і його версію; можливостей пристрою.

Моніторинг ДРП та запис даних моніторингу відбувається в автоматичному режимі. Передача даних на сервер виконується в двох режимах: інтерактивному і автоматичному (по запису сервера). Передача/прийняття даних здійснюється по сокетній технології за допомогою відповідних процесів.

Процес ВДЕ, як на стороні сервера, так і на стороні клієнта, забезпечує організацію доступу до БД: локальної, серверної. Цей процес забезпечить читання процедур читання/запису/пошуку/перевірки даних на відповідність при роботі з БД.

Процес відображення даних призначений для виведення на графічну форму даних моніторингу мережевих ДРП та їх фільтрацію (при необхідності).

Процес диспетчера друку забезпечить виведення даних моніторингу на друкуючий пристрій.

Визначивши і обґрунтувавши:

- проектні рішення щодо побудови системи та розробки її програмного забезпечення;
- побудову структурної, функціональної схем системи та діаграми процесів;
- обмеження, вимоги, умови та критерії, яких повинна дотримуватись система при виконанні визначених функцій;
- комплекс технічних засобів, які забезпечить експлуатацію системи;
- перелік вхідних даних, необхідних для організації моніторингу мережевих ДРП, маємо всі необхідні критерії та дані для розробки програмного забезпечення системи.

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-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

При розробці програмного забезпечення моніторингу мережевих принтерів використане програмування під ОС Windows з використанням подій Windows та елементів ActiveX, що надасть програмній моделі більш наглядного вигляду та зрозумілості для сучасного користувача, який звик працювати в цій операційній системі.

Під час роботи системи буде працювати досить потужний графічний інтерфейс користувача, який забезпечить реалізацію діалогу з користувачем та керування модулями системи. Використовуючи інтерактивний режим, користувач зможе виконувати функції спостереження та контролю за роботою системи в цілому та проводити корегування/підтримку режимів роботи системи. Але інтерфейс буде організований таким чином, що він не зможе впливати на сам хід процесу роботи системи, його присутність необхідна тільки на початковому етапі ініціалізації системи та, як це було вже означено, виконання функцій спостереження/контролю в інтерактивному режимі.

У теорії моніторинг роботи мережевих ДПР не є чимось неймовірно складним – ті ж драйвери не просто здійснюють моніторинг, але і надають можливості по управлінню їх роботою. Проте це лише теорія, тому на даному етапі роботи ми визначимося з побудовою власних алгоритмів відстежування посланих на друк завдань. Для цього в арсеналі ОС Windows передбачено два методи.

Перший – використання відстеження повідомлення WM_SPOOLERSTATUS, яке система посилає всякий раз при додаванні в чергу

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

нового завдання або ж при його видаленні.

Другий спосіб – використання функцій FindFirstPrinterChangeNotification, FindPrinterChangeNotification і FindClosePrinterChangeNotification.

Ми використовуємо, як основу розробки власних алгоритмів системи, перший метод. Як і більшість повідомлень ОС Windows, WM_SPOOLERSTATUS повертає структурі TMessage деяку змістовну інформацію, яку користувач може використовувати в своїх потребах. Але, на жаль, ніякої дійсно важливої і потрібної інформації (крім кількості завдань, що залишилися в черзі) ця структура не несе.

Проте, в ОС Windows є додаткові методи для визначення необхідної інформації. Так, серед інших, в модулі WinSpool присутня функція EnumJobs, що повертає кількісні показники і характеристики завдань на друк, в яких міститься необхідна інформація, – від назви документа, ПК, ДРП і імені користувача, що відправив цей документ на друк, до часу, коли це було зроблено і кількості сторінок, посланих на друк.

Синтаксис цієї функції виглядає наступним чином:

```
function EnumJobs(hPrinter: THandle; FirstJob, NoJobs, Level: DWORD; pJob: Pointer; cbBuf: DWORD; var pcbNeeded, pcReturned: DWORD): BOOL; stdcall,
```

де:

- hPrinter – ідентифікатор принтера, до якого проводиться запит;
- FirstJob – номер першого запитаного завдання;
- NoJobs – кількість запитаних завдань;
- Level – тип структури, яка передається в pJob;
- pJob – покажчик на масив змінних типу JOB_INFO_1 при Level=1 і JOB_INFO_2 при Level=2 (таблиця 4.1, таблиця 4.2);
- CbBuf – розмір буфера даних;

– Name – назва принт – об'єкта; значення цього параметра використовується в парі з Level (таблиця 4.4);

– Level – індекс рівня; використовується разом з Name;

– PPrinterEnum – покажчик на масив елементів одного з типів TPrinterInfo1, TPrinterInfo2, TPrinterInfo4, TPrinterInfo5;

– CbBuf – розмір буфера pPrinterEnum; ·PcbNeeded – змінна, у якій міститься кількість байт, скопійованих при вдалому завершенні операції, або ж розмір бракованої пам'яті, якщо значення cbBuf дуже мале;

– PcReturned – кількість актуальних записів, занесених в буфер.

Таблиця 4.3 – Прапорці для EnumPrinters

PRINTER_ENUM_LOCAL	Ігнорує Name і перераховує всі локальні ДРП.
PRINTER_ENUM_NAME	Перераховує ДРП, задані параметром Name. Якщо Name = nil, перераховує всі доступні принт-провайдери.
PRINTER_ENUM_SHARED	Перераховує розшарені принтери; використовується в парі з іншим прапорцем.
PRINTER_ENUM_DEFAULT	Тільки в Win 95; повертає інформацію з принтера по замовченню.
PRINTER_ENUM_CONNECTIONS	Тільки в NT; повертає інформацію про принтери, до яких відбулось звернення.
PRINTER_ENUM_NETWORK	Тільки в NT; перераховує мережні принтери в домені даного ПК (використовується при Level = 1).
PRINTER_ENUM_REMOTE	Тільки в NT; перераховує мережні принтери і принт-сервери в домені даного ПК (використовується при Level = 1).


```

        ListBox1.Items.Add('Jobs Left:'+IntToStr(Msg.WParamLo));
        if not EnumJobs(PH,0,1024,2,@Job2,SizeOf(Job2),Needed,Returned) then
        ShowMessage(SysErrorMessage(GetLastError));
        For i:=0 to Returned-1 do
        With Job2[i] do
        ListBox1.Items.Add(Format('%s %s %s',[pPrinterName,pDocument,pUserName]))
        end;

```

Після запуску додатку на екрані з'являється вікно, в компоненті TreeView якого присутній список доступних локальних принтерів. Відкриваємо документ (подвійне клацання на назві) і одержимо необхідну на даний відлік часу інформацію.

Отже, нами фактично визначені шляхи реалізації функції моніторингу стану черги на друк. І не просто моніторингу, а ще і автоматичної обробки результатів моніторингу у корисних цілях.

Програмне забезпечення системи, розроблене в процесі виконання ВКРМ, має модульну структуру побудови. Тобто, головний модуль та ряд допоміжних модулів, що забезпечують виконання функцій головним модулем.

Вхідні та вихідні дані забезпечимо часовою міткою у для фіксації моменту їх передачі/приймання, адже дані будуть оброблятися в динамічному режимі на протязі всього процесу моніторингу.

Мінімальна глибина зберігання даних відповідає нормативним документам. Система забезпечить їх архівацію та надасть можливість користувачу самому приймати рішення по цьому питанню.

Спеціально розроблені алгоритми забезпечать захист ПЗ від несанкціонованого доступу (НСД) на введення інформації та обробку даних.

До складу функцій системи буде введений контроль на обмеження дій користувача при його роботі з системою. В разі виявлення некоректних дій користувача, система заблокує його роботу та надасть доступ до режиму HELP, тобто: теоретичних відомостей, інструкцій користувача, правил роботи з системою, тощо.

Головна програма є, по суті, диспетчером системи. Вона забезпечує:

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- виклик, контроль роботи та завершення роботи основних модулів системи;
- санкціонований доступ та реєстрацію користувачів в системі;
- забезпечення роботи графічного інтерфейсу;
- обробку помилок;
- формування та виведення на екран монітору користувача інтерактивних довідок;
- організацію та контроль взаємодії програмних модулів системи.

Враховуючи, що при побудові архітектури системи використана технологія «Клієнт-Сервер» і до складу системи входить 2 змістовні частини: частина 1 «Клієнт» і частина 2: «Сервер», відповідно – кожна з них буде мати свою головну програму. Проте функції, які вони будуть виконувати, багато в чому схожі.

Головна програма клієнтської частини в процесі експлуатації системи буде забезпечувати керування відповідними модулями частини 1 і виконання наступних функцій:

- моніторинг принтерів (локальних/мережевих): кількості заданих на друк та надрукованих сторінок; імені користувача; імені ПК; назви додатку, з якого відбувся друк;
- виведення даних по драйверу принтера;
- виведення метрики принтера;
- виведення списку шрифтів, які підтримуються ДРП;
- виведення можливостей принтера;
- запис даних про друк в проміжну структуру;
- збереження даних з проміжної структури до БД.

Розглянемо блок-схему алгоритму роботи головної програми клієнтської частини (рисунок 4.1).

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

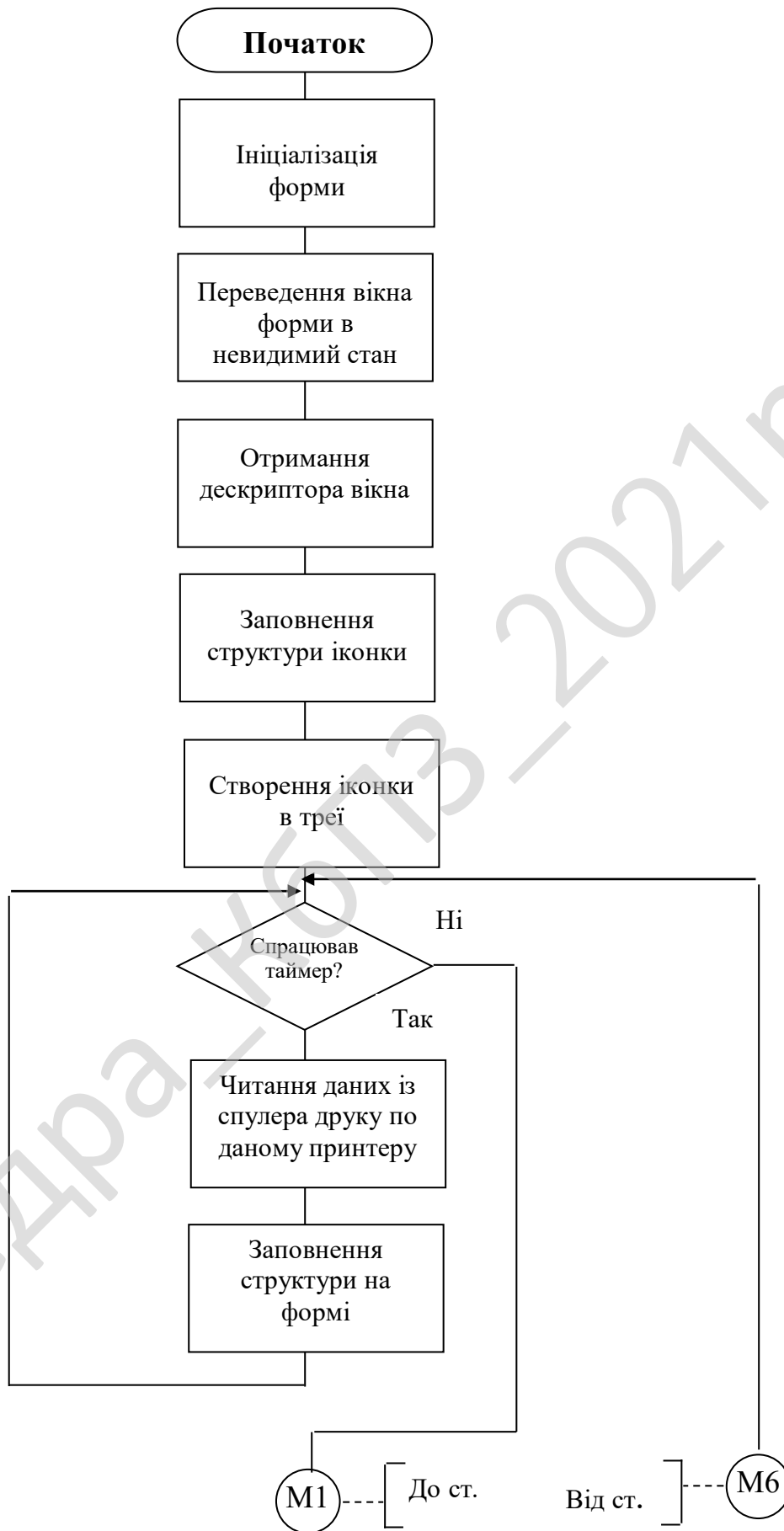


Рисунок 4.1 – Блок-схема алгоритму роботи головної програми клієнтської частини

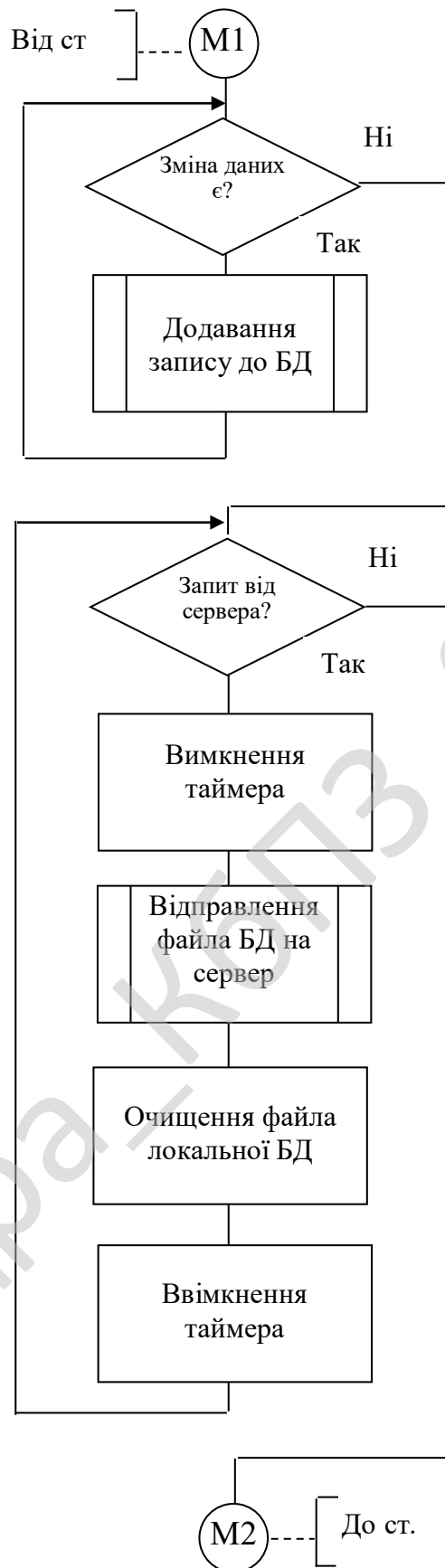


Рисунок 4.1, аркуш 2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0013.00.00.ПЗ

Арк.

52

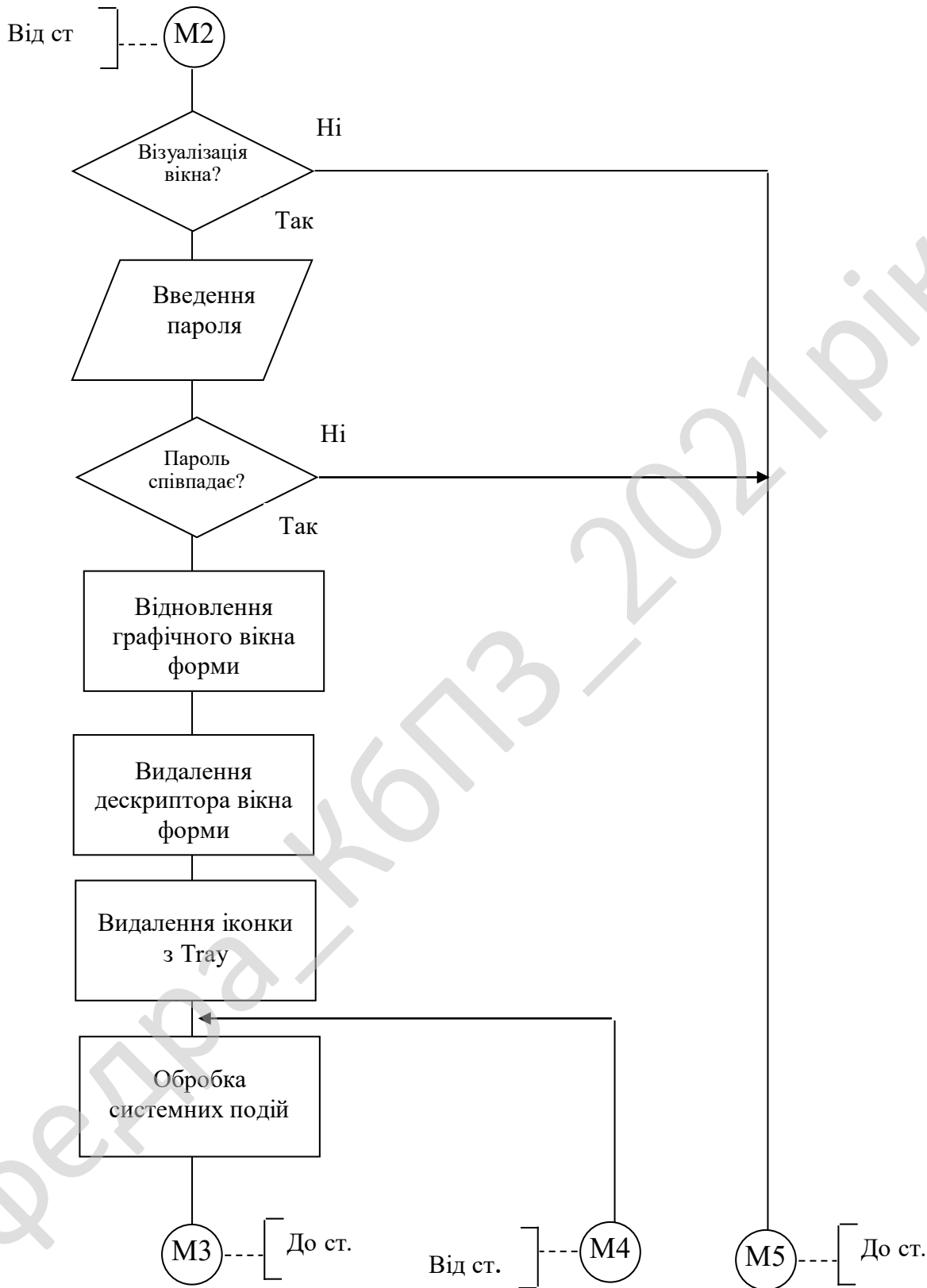


Рисунок 4.1, аркуш 3

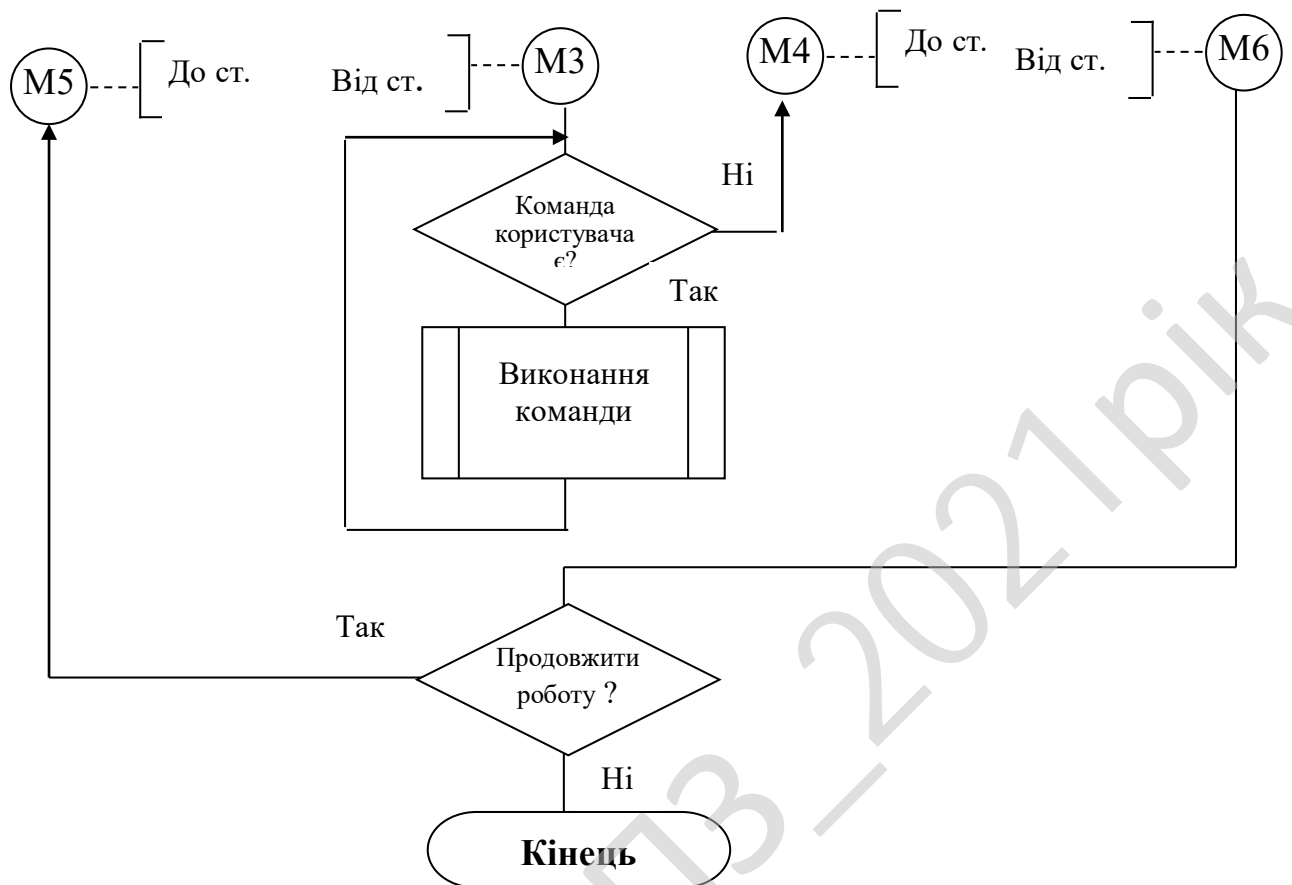


Рисунок 4.1, аркуш 4

Наводимо алгоритм роботи головної програми клієнтської частини:

- Крок 1: початок роботи головної програми. Перехід на крок 2.
- Крок 2: ініціалізація головної форми графічного інтерфейсу. Перехід на крок 3.
- Крок 3: переведення вікна форми в невидимий стан. Перехід на крок 4.
- Крок 4: отримання дескриптора вікна. Перехід на крок 5.
- Крок 5: заповнення структури іконки. Перехід на крок 6.
- Крок 6: створення іконки в треї. Перехід на крок 7.
- Крок 7: очікування спрацювання таймера (періодичність спрацювання 500 мс). Якщо таймер спрацював – перехід на крок 8, якщо ні – перехід на крок 10.

– Крок 8: читання даних із спулера друку по заданому локальному/мережному ДРП. Перехід на крок 9.

– Крок 9: заповнення структури на формі одержаними даними моніторингу. Повернення на крок 7 і знову очікування спрацювання таймера через 500 мс. Час спрацювання таймера може бути змінений по бажанню користувача.

– Крок 10: контроль зміни даних структури на формі. Якщо зміна даних відбулась – перехід на крок 11, якщо ні – перехід на крок 12.

– Крок 11: відкриття локальної бази даних клієнтської частини, додавання одержаного запису до бази даних, закриття локальної БД і повернення на крок 10. Очікування зміни даних в структурі на формі.

– Крок 12: очікування запиту від сервера, якщо запит надійшов – перехід на крок 13, якщо ні – на крок 17.

– Крок 13: вимкнення таймера. Перехід на крок 14.

– Крок 14: відправлення файлу локальної бази даних на сервер. Перехід на крок 15.

– Крок 15: очищення файлу локальної БД клієнтської частини. Перехід на крок 16.

– Крок 16: ввімкнення таймера і повернення на крок 12.

– Крок 17: якщо від користувача надійшла команда на розгортання вікна головної форми з іконки Тгеу – перехід на крок 18. Якщо ні – перехід на крок 26.

– Крок 18: введення паролю. Перехід на крок 19.

– Крок 19: перевірка введеного паролю на співпадання з еталоном, який знаходиться в базі даних. Якщо перевірка пройшла успішно – перехід на крок 20, якщо ні – перехід на крок 26.

– Крок 20: відновлення графічного вікна форми. Перехід на крок 21.

– Крок 21: видалення дескриптора вікна форми. Перехід на крок 22.

– Крок 22: видалення іконки з Тгеу. Перехід на крок 23.

– Крок 23: обробка системних подій. Перехід на крок 24.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

– Крок 24: перевірка надходження команди користувача на виведення даних моніторингу. Якщо команда надійшла – перехід на крок 25, якщо ні – повернення на крок 23.

– Крок 25: виконання команди користувача: виведення даних характеристики друкуючого пристрою; даних персонального комп'ютера, з якого надійшло завдання, тощо. Повернення на крок 24.

– Крок 26: якщо система продовжує роботу – повернення на крок 7, якщо ні – вихід на кінець роботи програми.

Головна програма серверної частини системи в процесі експлуатації забезпечить виклик, запуск та закриття (після виконаної роботи) підлеглих їй модулів системи для виконання наступних функцій:

– отримання даних моніторингу від друкуючого пристрою з локальної чи глобальної мережі;

– перевірка коректності одержаних даних;

– запис даних до бази даних моніторингу компанії /організації/ установи/тощо;

– перегляд даних моніторингу;

– режим фільтрації по заданих критеріях;

– друк вибраних з черги даних на друкуючому пристрої: принтері, плотері тощо.

Розглянемо алгоритм роботи головної програми серверної частини системи, наведений на рисунку 4.2.

Обробник події OnRead компоненти TClientSocket активізується при передачі кожного блоку даних, тому йому потрібна інформація про те, що іде приймання першого блоку даних/другого блоку даних/інших блоків даних.

В першому випадку, обробник створить тимчасовий файл для збереження даних, а в другому – відкриє цей файл і приєднає до нього ці дані в кінець. Окрім цього, сервер ніяк не сповіщає клієнта про передачу останнього блоку даних, але низькорівневі API-функції сокетів дозволяють клієнту одержати прапорці пакету,

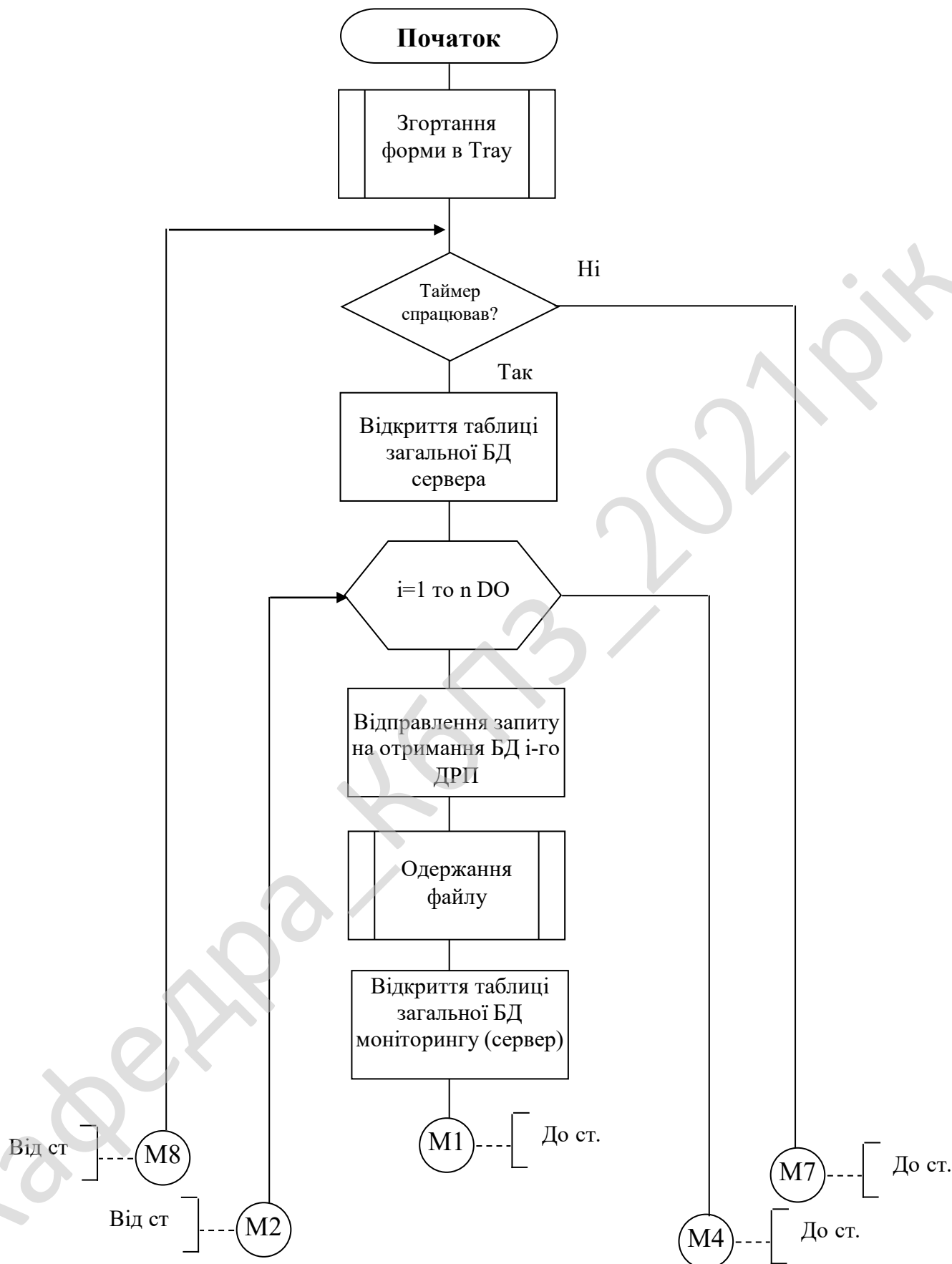


Рисунок 4.2 – Блок-схема роботи алгоритму роботи головної програми серверної частини

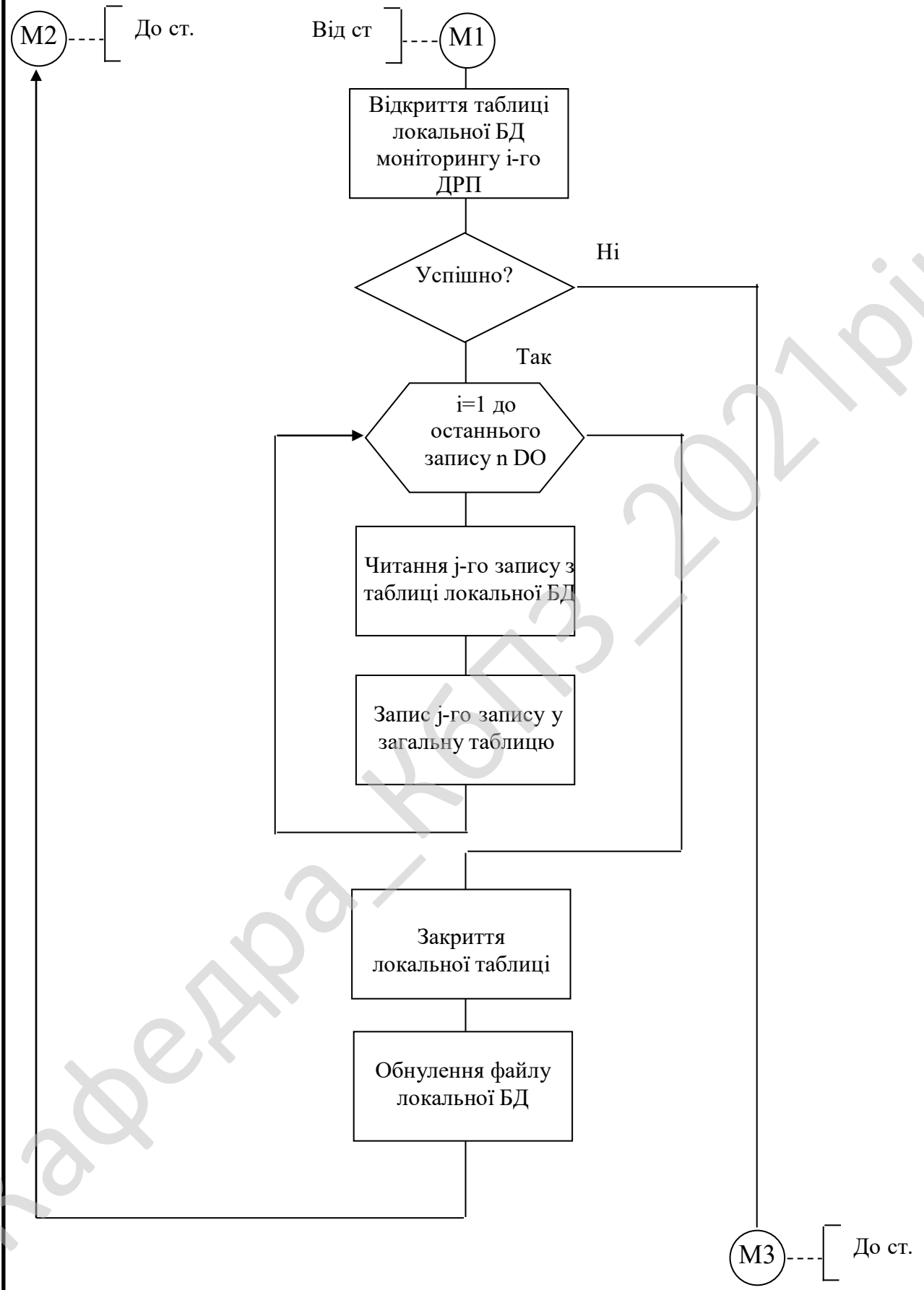


Рисунок 4.2, аркуш 2

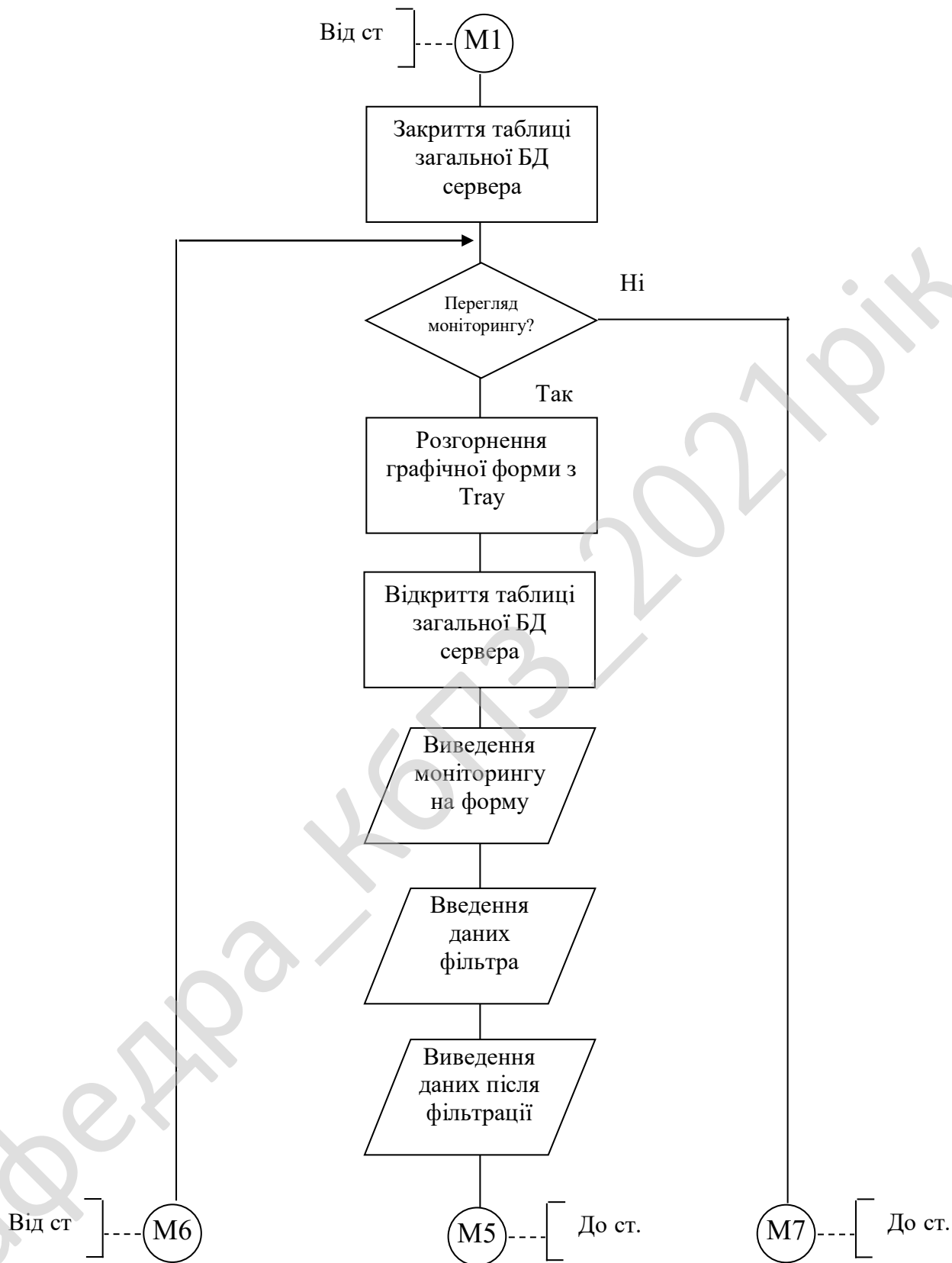


Рисунок 4.2, аркуш 3

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0013.00.00.П3

Арк.

59

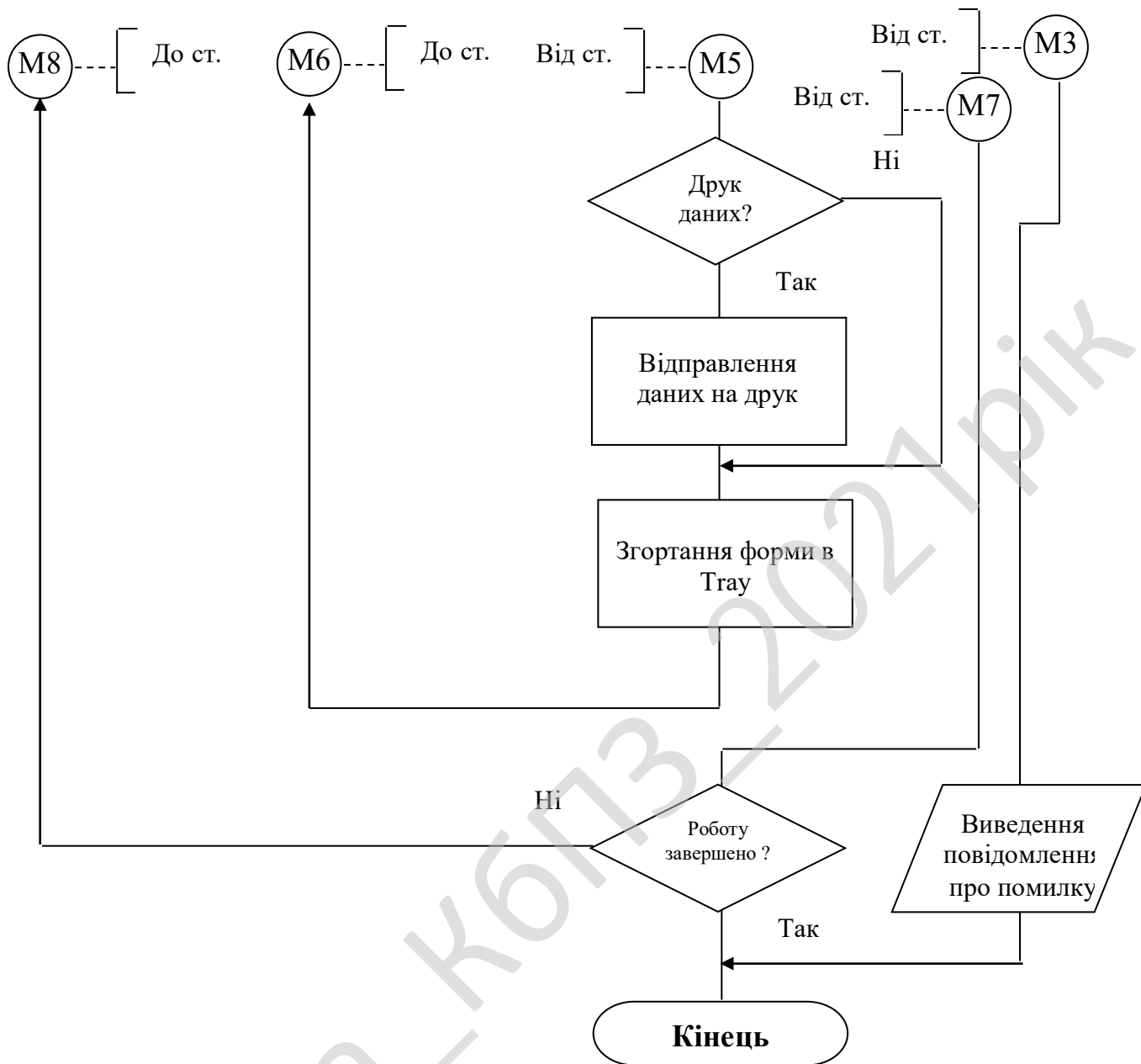


Рисунок 4.3, аркуш 4

серед яких є прапорець, який вказує на те, що передача даних закінчена. Однак, на компонентному рівні DELPHI аналіз цих прапорців недоступний. Тому цю задачу теж слід означити та вирішити.

Перша означена проблема вирішена досить просто – за рахунок використання глобальної змінної логічного типу, значення якої не порушується після завершення роботи обробника. Щоб мати дані про завершення передачі блоків, використаємо при написанні ПЗ таймер, який запускається перед

завершенням роботи обробника і зупиняється на початку роботи. По витоку встановленого в ньому часу можна стверджувати, що передача закінчена.

Установимо відлік часу таймера 100 мс. Якщо таймер спрацював – значить, на протязі останніх 100 мс від сервера не поступало ні одного блока даних і, відповідно, передача закінчилась. Відлік часу періодичності спрацювання таймера може бути змінений на будь-яке інше значення по бажанню користувача.

Розглянемо блок-схему роботи алгоритму модуля зв'язку з сервером зі сторони клієнта, наведену на рисунку 4.3.

Наводимо алгоритм роботи модуля зв'язку з сервером зі сторони клієнта:

- Крок 1: початок роботи програми. Перехід на крок 2.
- Крок 2: встановлення зв'язку з сервером. Перехід на крок 3.
- Крок 3: встановлення прапорця приймання нового файлу. Перехід на крок 4.
- Крок 4: якщо пакет від сервера надійшов – перехід на крок 5, якщо ні – повернення на крок 3.
- Крок 5: перевірка стану таймера 1. Якщо таймер спрацював – повернення на крок 3, якщо ні – перехід на крок 6.
- Крок 6: вимикаємо таймер 1. Перехід на крок 7.
- Крок 7: одержання довжини пакета даних. Перехід на крок 8.
- Крок 8: резервування пам'яті, перехід на крок 9.
- Крок 9: одержання пакетів даних в буфер, перехід на крок 10.
- Крок 10: підготовка файлу до прийняття пакетів даних. Перехід на крок 11.
- Крок 11: визначення напрямку роботи. Якщо одержуємо новий файл – перехід на крок 12, якщо файл уже існує і необхідно зробити черговий запис даних до нього – відкриваємо файл для запису, переходимо на його кінець і переходимо на крок 14.
- Крок 12: відкриття файлу для перезапису, перехід на крок 13.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

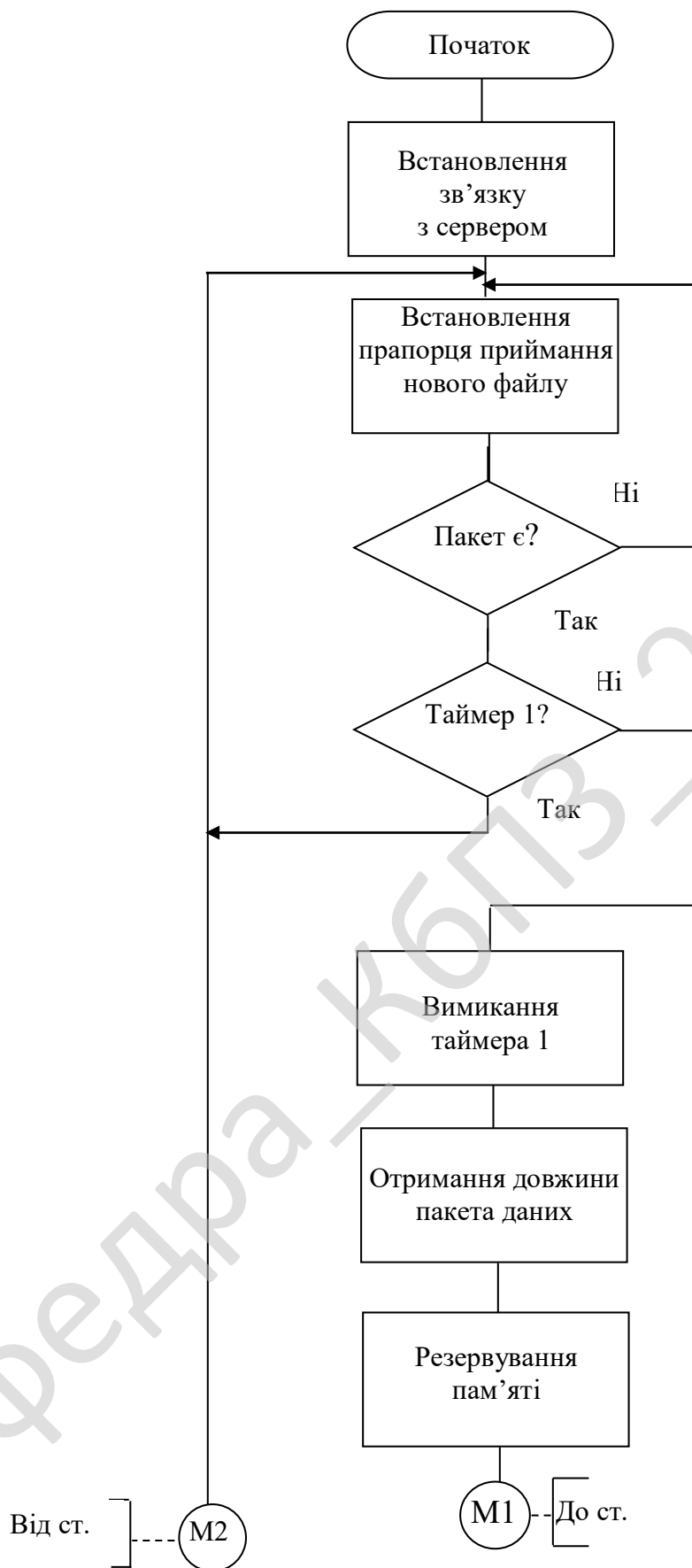


Рисунок 4.3 – Блок-схема роботи алгоритму модуля зв'язку з сервером зі сторони клієнта

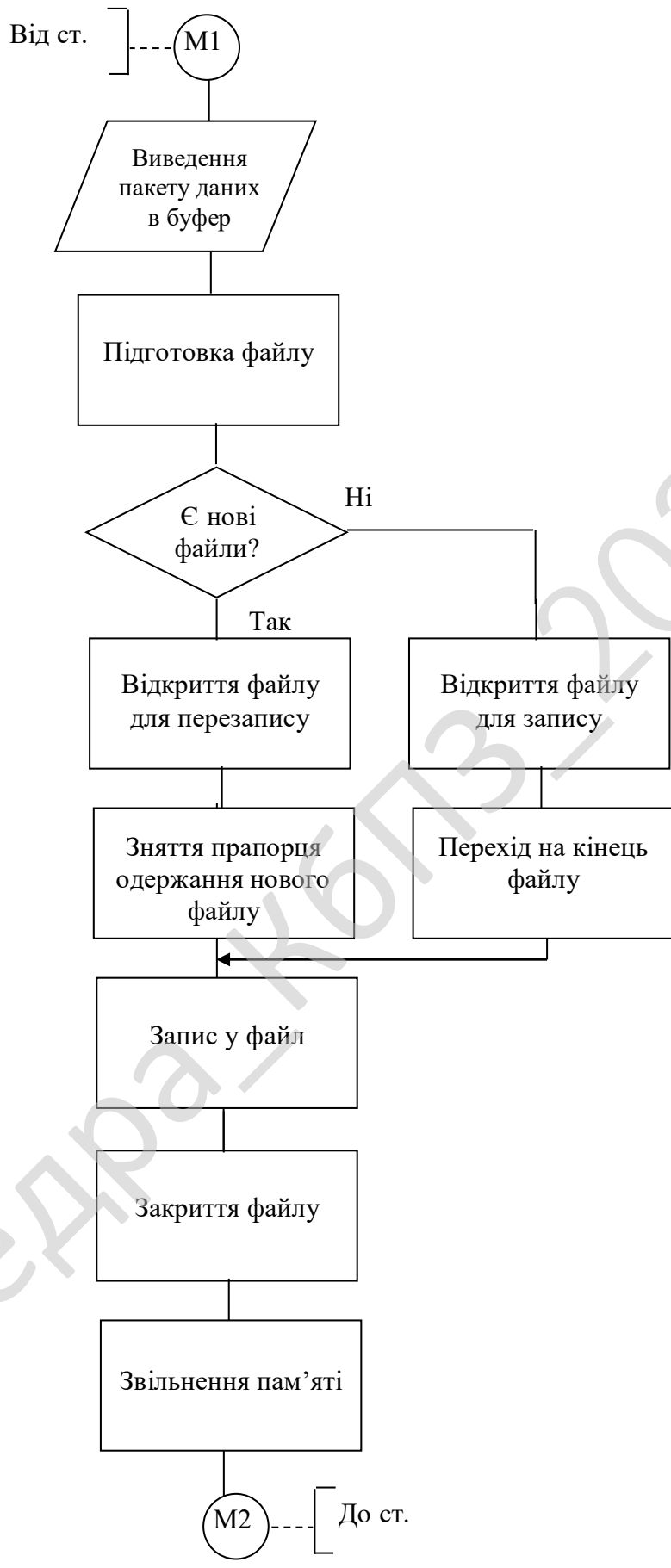


Рисунок 4.3, аркуш 2

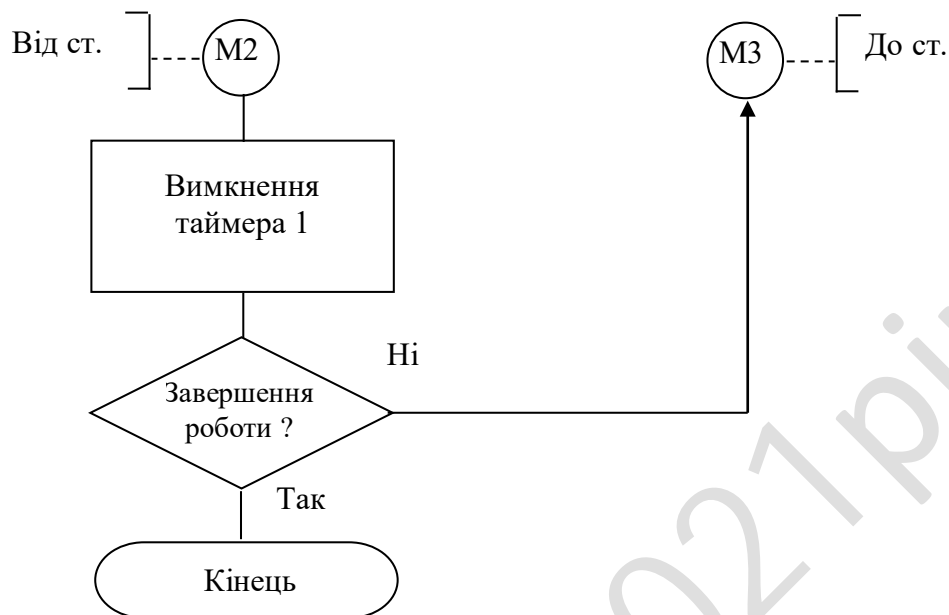


Рисунок 4.3, аркуш 3

- Крок 13: знімаємо прапорець прийняття нового файлу, перехід на крок 14.
- Крок 14: запис пакета даних у файл, перехід на крок 15.
- Крок 15: закриття файлу, перехід на крок 16.
- Крок 16: звільнення зарезервованої пам'яті, перехід на крок 17.
- Крок 17: вимкнення таймера, перехід на крок 18.
- Крок 18: визначення напрямку роботи. Якщо роботу програми завершено – перехід на крок 19, якщо ні – повернення на крок 3.
- Крок 19: кінець роботи програми.

При розробці вищезначеної програми для класу форми TfmC1 в секцію rab1ic встановлені 2 глобальні змінні – IsRecStart та VmpFile.

Перша глобальна змінна використовується в якості прапорця початку передачі даних. В обробнику Button2Click вона одержує значення True. Це значення аналізується в обробнику ClientSocket 1 Read: якщо True, файл VmpFile створюється, в нього розміщується перша порція даних; при цьому прапорець IsRecStart одержує значення False; якщо False – файл відкривається і в його кінець дописується чергова порція даних.

комп'ютерах, які, як показало час, не за горами[4]. У цьому ж році, математики Jeffrey Hoffstein, Jill Pipher і Joseph H. Silverman, що розробили систему разом із засновником компанії NTRU Cryptosystems, Inc. (пізніше перейменованої в SecurityInnovation), Даніелем Ліманом (Daniel Lieman) запатентували свій винахід.[5]

Кільця усічених багаточленів

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

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

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

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

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

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

Генерація відкритого ключа

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

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

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

Використовуючи ці поліноми, зашифроване повідомлення виходить за формулою:

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

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

Приклад:

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

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

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

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

Тоді шифротекст e , готовий для передачі Бобові буде:

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

Розшифрування

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

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

Якщо розписати шифротекст, то одержимо ланцюжок:

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

і остаточно:

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

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

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Згідно з ТЗ в процесі роботи над ВКРМ було розроблене ПЗ моніторингу роботи мережевих принтерів. Розроблена система може бути впроваджена в промислову експлуатацію в структурах будь-якої сфери діяльності людини, де використовується робота ПК в локальних та глобальних мережах.

Впровадження любого побічного інструментарію, в тому числі – і програмних засобів моніторингу мережевих принтерів, супроводжується складностями. Вони пов'язані з тим, що необхідно розширити коло обов'язків супровідників ПЗ та служби технічної підтримки.

Персоналу, що буде супроводжувати ПЗ, необхідно вивчити функції системи, представлені наявною технічною документацією на рівні, що буде достатнім для реалізації інтеграції системи моніторингу в наявний комплекс програмного забезпечення замовника. Це досить складне питання, оскільки потребує вбудови ПЗ в вихідні тексти програм наявного ПЗ, що потягне за собою витрати на додаткове тестування і появу нової версії продукту. Але ця задача середньої складності і підлягає вирішенню в термін 3-4 дні. З ціллю здеш евлення загальної вартості впровадження, доцільно провести пробне навчання спеціалістів Замовника та провести впровадження системи в експлуатацію.

Впровадження системи може бути ефективним при умові наявності методичного (МЕТОЗ) та організаційного забезпечень (ОРЗ) системи. МЕТОЗ безпосередньо взаємодіє з інформаційним, програмним та технічним забезпеченням системи на етапі промислової експлуатації. Найбільше МЕТОЗ взаємодіє з інформаційним забезпеченням. Адже інформаційне забезпечення визначає склад і уміст інформації, що використовується й отримується в процесі проходження проекту. Зрозуміло, що методичний та інформаційний бік процесу моніторингу взаємопов'язані, адже вони використовують однакові джерела

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

початкової інформації та однакові форми її подання. Саме від якості методичного забезпечення залежить якість початкової інформації, а відповідно і якість організації самого процесу моніторингу роботи ПК в мережі.

Після запуску програми моніторингу, програма згортається в трей і не загромаджує своїм робочим вікном робочий простір екрана, як це ілюструється скріншотами (рисунок 5.1, 5.2, 5.3).

Дата і час	Ім'я принтера	Ім'я ПК	Ім'я користувача	Найменування документа	Метод друку	Статус	Статус 2	Кількість сторінок
2:08:48_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:49_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:49_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:50_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:50_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:51_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:51_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:52_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:52_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:53_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:53_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:54_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:54_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:55_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:55_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:56_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:56_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:57_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:57_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:58_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:58_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:59_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	
2:08:59_03.06.2	Canon LASER SHOT LBF \\\\NOYT		Admin	Microsoft Word - Інф	RAW		JOB_POSITI 2	

Рисунок 5.1 – Скріншот графічного інтерфейсу серверної частини

Авторське право

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 моніторингу мережевих принтерів

ОПП «Комп'ютерна інженерія»
 спеціальності 123 «Комп'ютерна інженерія»
 Виконав: Немировський Д.В.
 Науковий керівник: Олексій СМІРНОВ
 Кропивницький - 2021

OK

Рисунок 5.2 – Форма авторського права

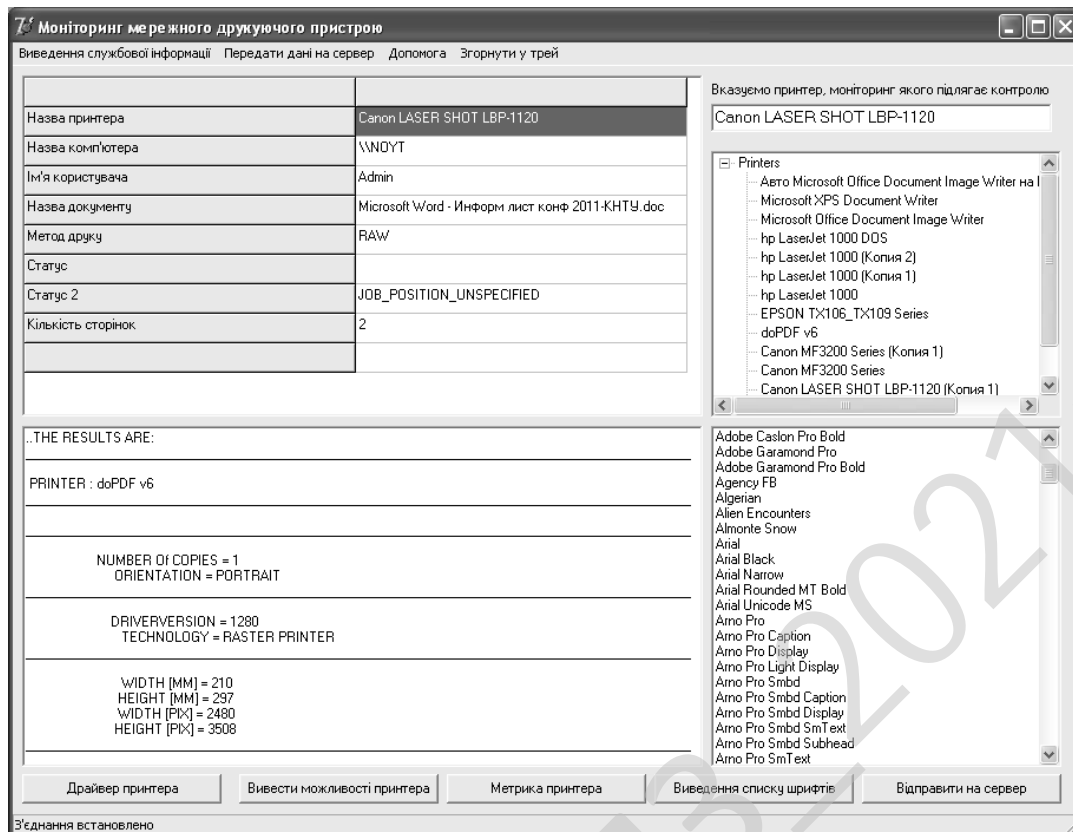


Рисунок 5.3 – Скріншот графічного інтерфейсу клієнтської частини

МЕТОЗ на стадії впровадження системи в промислову експлуатацію – це комплекс документів, що описують принципи організації її впровадження і утримують правила та інструкції по використанню ПЗ, а саме: інструкції по використанню програмного забезпечення; правила та інструкції по забезпеченню виконання вимог ТБ та промсанітарії при експлуатації системи; інструкції користувача, який буде працювати з системою; посадові інструкції обслуговуючого персоналу.

Організаційне забезпечення – це перелік документів, що регламентують взаємодію та функції підрозділів підприємства та матеріально-технічного забезпечення системи.

Методичне та організаційне забезпечення забезпечують виконання плану впровадження.

План впровадження передбачає організацію підготовки кадрів, експлуатаційного персоналу та аналіз функціонування системи після виконання всіх робіт із її впровадження.

У плані впровадження зазначаються організації (підрозділи), які відповідають за виконання робіт. Тому для впровадження розробленого програмного забезпечення в експлуатацію потрібна взаємодія програміста – розробника з користувачем, ІТР по обслуговуванню технічного обладнання, керівниками різних рівнів та підрозділів, які будуть його експлуатувати.

Програма контролю має графічний інтуїтивно зрозуміли інтерфейс та є візуальним модулем. Крім даного візуального модуля для роботи системи моніторингу необхідні і не візуальні (невидимі модулі).

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи моніторингу мережевих принтерів.

Метою розробки є дослідження та програмна реалізація системи моніторингу мережевих принтерів.

Об'єктом дослідження є процес моніторингу мережевих принтерів.

Предметом дослідження є методи моніторингу мережевих принтерів.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод моніторингу мережевих принтерів.
- Розроблено вітчизняний продукт моніторингу мережевих принтерів, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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 моніторингу мережевих принтерів. 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	№	130 (2 ост. цифри № зал*10 ¹)
3. Запланований термін розробки, днів	Frq	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

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0013.00.00.ПЗ

Арк.

78

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 (2 ост. цифри № зал*10 ³)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	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.:

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де F_{pq} – плановий фонд робочого часу одного спеціаліста, днів,

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{209 \cdot 1}{48 - 5} = 4,8 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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 обладнан ня	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м. п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

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:

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

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки}$$

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 (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server, серверу доступу АДСЛ (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	0,2	0,1
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів 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:

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	10000	20000
Продакт-менеджер	0,4	9000	7200
Інженер-програміст	4,8	8500	81600
Інженер-електронщик	1,2	7500	18000
Інженер-системотехнік	0,1	7500	1500
Адміністратор мережі	0,1	7500	1500
Системний програміст	0,1	7500	1500
Дизайнер WEB	0,1	7500	1500
Інженер-верстальник	0,1	7500	1500
Бухгалтер-економіст	0,1	7500	1500
Всього за період розробки	$R_{cn}=8$	-	$\Phi_{роб}=135800$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{135800}{8 \cdot 48} = 354 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

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 ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає

1000...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 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:

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

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

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

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

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

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 фірми Brain за 29.10.21 – джерело <http://brain.com.ua>

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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	Подаруно

Продовження таблиці 7.5

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Монітор BenQ GL2450HM Black	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складено таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	12721	10176,8	111944,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	133576,3

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>, що враховуючи курс 25 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.

$$Z_o = 354 \cdot 209 / 130 = 569 \text{ грн}$$

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-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Згідно виданих викладачем норм приймаємо 0,5 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n=70$ грн., визначаємо вартість паперу за період розробки $N_m=2$ міс:

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

$$Z_{M1} = 70 \cdot 2 \cdot 0,5 = 70 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d., \quad (7.17)$$

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 7 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 12 грн/шт.

$$Z_{M2} = 130 \cdot 7 + 12 = 922 \text{ грн.}$$

Згідно виданих викладачем 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.)

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 148544 \cdot 2 / (130 \cdot 12) = 190 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 569 + 57 + 138 + 85 + 21 + 85 + 190 = 1145 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де P_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 1145 = 572,5 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	569
2. Додаткова зарплата виконавців	Z_d	57
3. Відрахування на соціальні потреби	C_{oc}	138
4. Загальногосподарські витрати	Γ_{ocn}	85
5. Витрати на матеріали	Z_m	21
6. Освоєння нових операційних систем, мов програмування	O_n	85

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	190
8. Повна собівартість програмного забезпечення	C_n	1145
9. Плановий прибуток	P_p	572,5
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1717,5
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	343,5
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	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

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування системи	Z_p	10065	2684
2. Витрати на амортизацію	$Z_{ам}$	0	515,25
Всього витрат за рік	I	10065	3199,25

Витрати на технічне обслуговування і підтримку працездатності системи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 75 годин на рік до 20 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 75 \cdot 100 \cdot 1,1 \cdot 1,22 = 10065 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 20 \cdot 100 \cdot 1,1 \cdot 1,22 = 2684 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

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.	6351
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Роки	0,3

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

$$E_{cn} = (10065 - 3199,25) - 0,25 \cdot 2061 = 6351 \text{ грн.}$$

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

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

$$T_{cn} = \frac{2061}{10065 - 3199,25} = 0,3 \text{ роки}$$

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-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя [1]. Охорона праці є складовою частиною безпеки життя. Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності.

Законом України “Про охорону праці” [2] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста,

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат;

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	8,5*
Довжина	13*
Висота	2,9

* вказано загальні розміри поєданого приміщення, де загалом працюють 16 людей, а фактично у наявності є дві кімнати, розділених перестінком.

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6,9
Обсяг, V	м ³	не менше 20.0	20

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працює 16 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких:

У приміщенні знаходяться наступні джерела шуму: принтер *HP Laser 135a*, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р. 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 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 поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 8,5 м, довжина 13 м, висота 2,9 м.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = ESKZ/n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S = 8,5 \times 13 = 110,5$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h(A+B)),$$

де:

S – площа приміщення, $S = 110,5 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 2,9 \text{ м}$ (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 8,5 \text{ м}$;

B – довжина приміщення, $B = 13 \text{ м}$.

Підставимо всі значення у формулу та визначемо індекса приміщення:
 $i = 1,77$.

Знаючи індекс приміщення (i), знаходимо $n = 0,57$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників [3]). Підставимо всі значення у формулу, визначимо світловий потік: $F = 95960 \text{ Лм}$.

Для штучного освітлення приміщення використовуються світильники $LED 54W-5000K$, світловий потік яких $F_{л} = 5940 \text{ Лм}$.

Число світильників визначається по формулі:

$$N = F / F_{л}$$

де:

F – світловий потік,

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

F_l – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення: $N=95960/5940=16,15$ шт.

Для забезпечення нормованої мінімальної освітленості приймаємо необхідну кількість світильників 17 шт.

8.6 Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи моніторингу мережевих принтерів.

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 моніторингу мережевих принтерів.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем моніторингу мережевих принтерів.
- Досліджена система моніторингу мережевих принтерів.
- На основі отриманих результатів досліджень створена програмна реалізація системи моніторингу мережевих принтерів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання моніторингу мережевих принтерів.

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

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

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

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. 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 XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм NTRU.

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

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 6351 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,3 роки.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Немировський Д.В. Дослідження та програмна реалізація системи моніторингу мережевих принтерів // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.

2. Неберекутін Александр. Printer Activity Monitor – афективне використання оргтехніки. [Електронний ресурс]. Режим доступу: <http://www.1csoft.ru/publications/8144/6082470/>.

1. А.А.Тімченко Основа системного проектування та системного аналізу складних об'єктів: Підручник для студентів ВНЗ: У 2-хкн. Кн.1 (А.А.Тімченко; під ред.: В.І.Биков). – К.: Либідь, 2000. – 270 с.

2. Хавьер Пашеку. Програмування Borland Delphi для професіоналів Delphi for. NET DEVELOPER`S Guide. – М.: Вільямс, 2006. – 944 с.

3. Глонь О.В., Дубовой В.М., Мітюшкін Ю.І. Комп'ютеризовані системи керування: Навчальний посібник. – Вінниця: ВНТУ, 2005. – 157с.

4. Савеленко О.К., Колодочкіна А.В., Лисенко І.А. CASE-ТЕХНОЛОГІЇ У ПРОЕКТУВНІ ІНФОРМАЦІЙНИХ СИСТЕМ. – Кропивницький: ЦНТУ, 2017.- 194 с.

5. Береза А.М. Основи створення інформаційних систем: Навч. посібник. 2 видання,перероблене і доповнене – К.: КНЕУ, 2001. – 270 с.

6. Савеленко О.К., Якименко Н.М., Колодочкіна А.В., Сорокін В.В.

7. Технології проектування комп'ютерних систем: Навчальний посібник. – Кіровоград: КНТУ, 2016. – 237 с.

8. Линда И. Шафер. Роберт Т. Фатрелл, Дональд Ф. Шафер Керування програмними проектами: досягнення оптимальної якості при мінімумі затрат., Видавничий дім «Вільямс». 2003 г. -1136 с.

9. А.В. Колодочкіна, О.К. Савеленко Технології проектування комп'ютерних систем: конспект лекцій для студентів денної та заочної форм

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

навчання напряму підготовки 123 “Комп’ютерна інженерія” – Кіровоград: КНТУ, 2014. – 220 с.

10. Printer Activity Monitor. System Administrator’s Guide. [Електроний ресурс]. Режим доступу: http://www.redline-software.com/eng/support/docs/_data/pamHelp.pdf.

11. Наумчук О.М. Основи систем автоматизованого проектування.Інтерактивний комплекс навчально-методичного забезпечення. – Рівне: НУВГП, 2008. – 136 с.

12. Козлов А.П., Кринецький М.І. Основи систем автоматизованого проектування. Конспект лекцій. .К: НАУ, 2003. – 86 с.

13. Онищенко О.А. Конспект лекций по курсу Основы САПР. Одесса: Одесская Государственная Академия Холода, 2002. – 47 с.

14. Саєнко С.Ю., Нечипоренко І.В. Основи САПР. Х.: ХДУХТ, 2017. – 119 с.

15. Смирнов А.А. Разработка математической GERT-модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / А.А.Смирнов, Мохамад Гани Абу Таам // Информационные системы в управлении, образовании, промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – 498 с.

16. Смирнов А.А. Метод управления доступом в интеллектуальных узлах коммутации / Мохамад Гани Абу Таам, А.А.Смирнов // Информационные технологии и защита информации в информационно-коммуникационных системах: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – 486 с.

17. Смирнов А.А. Математическая GERT-модель технологии передачи метаданных в облачные антивирусные системы / В.В.Босько, А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Збірник наукових праць "Системи обробки інформації". – Випуск 1(117). – Х.: ХУПС – 2014. – С. 137-141.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

18. Смирнов А.А. Структурно-логическая GERT-модель технологии распространения компьютерных вирусов / А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Системи управління, навігації та зв'язку. – Випуск 1(29). – П.: ПНТУ. – 2014. – С. 120-125.

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

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

21. Смирнов А.А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 4(17). – Харків: ХУПС. – 2014. – С.90-95.

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

23. Смирнов А.А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Системи озброєння і військова техніка. – Випуск 3(43) – Х.: ХУПС – 2015. – С. 100-107.

24. Смирнов А.А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 3(19). – Х.: ХУПС. – 2015. – С. 134-141.

25. Mohamad Abou Taam Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

26. Смирнов А.А. GERT-модель технологии передачи данных в облачные антивирусные системы / А.А. Смирнов, В.В. Босько, Мохамад Гани Абу Таам // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 18-19.

27. Смирнов А.А. Математическое моделирование технологии передачи сигнатур в облачные антивирусные системы / Мохамад Гани Абу Таам, А.А. Смирнов // Збірник тез VI міжнародної науково-практичної конференції «Проблеми і перспективи розвитку ІТ-індустрії». м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 260.

28. Смирнов А.А. Анализ требований к качеству обслуживания в информационно-телекоммуникационных системах / А.А. Смирнов, Мохамад Гани Абу Таам // Збірник тез XVI міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 11-12 квітня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 124-126.

29. Смирнов А.А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / Мохамад Гани Абу Таам, С.А. Смирнов // Збірник тез науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія». м. Кіровоград. 4 грудня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 168.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

30. Смирнов А.А. Исследование математических моделей технологии распространения компьютерных вирусов / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник наукових праць міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 25-28 лютого 2015 р. – Київ: Європейський університет. – 2015. – С. 90-91.

31. Смирнов А.А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез всеукраїнської науково-практичної конференції «Інформаційна безпека держави, суспільства та особистості». м. Кіровоград. 16 квітня 2015. – Кіровоград: КНТУ. – 2015. – С. 50-52.

32. Смирнов А.А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник тез VII міжнародної науково-практичної конференції «Проблеми і перспективи розвитку ІТ-індустрії». м. Харків. 17-18 квітня 2015 р. – Харків: ХНЕУ. – 2015. – С. 14.

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

34. Смирнов А.А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез II Міжнародної науково-практичної Інтернет-конференції «Інформаційна та економічна безпека» (INFECO-2015)». м. Харків. 21-22 травня 2015 р. – Харків: ХІБС УБС НБУ. – 2015. – С. 20-24.

35. Смирнов А.А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

телекоммуникационных сетях / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Сборник тезисов XI международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 01 – 06 июня 2015 г – Варна. ТУВ. – 2015. – С. 488-491

36. Смирнов А.А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Комп'ютерні технології та інформаційна безпека». м. Кіровоград. 2-3 липня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 4-5.

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

38. Кочуров В.А. Инновационные технологии в системах автоматизированного проектирования / В.А. Кочуров, А.В. Бородуля, И.Л. Ковалёва, В.В. Напрасников, С.Е. Пекарчик/. – Учебно-методическое пособие. – Минск: БНТУ, 2017. – 111 с. – ISBN 978-985-550-703-2.

39. Якубовская Е.С. Методология проектирование и основы САПР. Курс лекций. – Минск: БГАТУ, 2004. – 205 с.

40. CAD-CAM & Rapid Prototyping Application Evaluation. Bookboon, 2010. – 174 p. – ISBN: 978-87-7681-676-6

41. Davidson J.K. (Ed.) Models for Computer Aided Tolerancing in Design and Manufacturing. Springer, 2005. – 354 p. – ISBN 978-1402054389.

42. He X. Hua E. Lin Y. Liu X. (Eds.) Computer-Aided Design, Manufacturing, Modeling and Simulation. Verlag C.H.Beck, 2011. – 780 p. – ISBN 978-3319509389.

43. Kurdila A.J., Pardalos P.M., Zabarankin M. (ed.) Robust Optimization-Directed Design . Springer, 2006. – 284 p. – ISBN-10: 0387282637

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

44. Leon-Rovira N. (Ed.) Trends in Computer Aided Innovation 2007. Springer, 2007. – 229 p. – ISBN 978-0387754550.
45. Luo Y. (Ed.) Cooperative Design Visualization and Engineering, CDVE 2017. Springer, 2017. – 300 p. – ISBN 978-3319668048.
46. Shaler Stidham Jr. Optimal Design of Queueing Systems. Chapman and Hall/CRC, 2009. 384 p. – ISBN-10: 1584880767
47. Астахова, И.Ф. Компьютерные науки. Деревья, операционные системы, сети / И.Ф. Астахова и др. – М.: Физматлит, 2013. – 88 с.
48. Астахова, И.Ф. Компьютерные науки. Деревья, операционные системы, сети / И.Ф. Астахова, И.К. Астанин и др. – М.: Физматлит, 2013. – 88 с.
49. Баринов, В.В. Компьютерные сети: Учебник / В.В. Баринов, И.В. Баринов, А.В. Пролетарский. – М.: Academia, 2018. – 192 с.
50. Баринов, В.В. Компьютерные сети: Учебник / В.В. Баринов. – М.: Академия, 2015. – 256 с.
51. Кузин, А.В. Компьютерные сети: Учебное пособие / А.В. Кузин.. – М.: Форум, НИЦ Инфра-М, 2013. – 192 с.
52. Кузин, А.В. Компьютерные сети: Учебное пособие / А.В. Кузин, Д.А. Кузин. – М.: Форум, 2018. – 704 с.
53. Кузьменко, Н.Г. Компьютерные сети и сетевые технологии / Н.Г. Кузьменко. – СПб.: Наука и техника, 2013. – 368 с.
54. Куроуз, Д. Компьютерные сети. Нисходящий подход / Д. Куроуз, К. Росс. – М.: Эксмо, 2016. – 912 с.
55. Куроуз, Дж. Компьютерные сети: Нисходящий подход / Дж. Куроуз. – М.: Эксмо, 2018. – 800 с.
56. Луганцев, Л.Д. Компьютерные сети / Л.Д. Луганцев. – М.: МГУИЭ, 2001. – 452 с.
57. Максимов, Н.В. Компьютерные сети: Учебное пособие / Н.В. Максимов, И.И. Попов. – М.: Форум, 2017. – 320 с.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

58. Максимов, Н.В. Компьютерные сети: Учебное пособие для студентов учреждений среднего профессионального образования / Н.В. Максимов, И.И. Попов. – М.: Форум, НИЦ Инфра-М, 2013. – 464 с.

59. Новожилов, Е.О. Компьютерные сети: Учебное пособие / Е.О. Новожилов. – М.: Academia, 2017. – 288 с.

60. Новожилов, Е.О. Компьютерные сети / Е.О. Новожилов. – М.: Academia, 2016. – 352 с.

61. Новожилов, Е.О. Компьютерные сети: Учебное пособие / Е.О. Новожилов. – М.: Academia, 2016. – 288 с.

62. Новожилов, Е.О. Компьютерные сети: Учебное пособие / Е.О. Новожилов. – М.: Академия, 2018. – 176 с.

63. Новожилов, Е.О. Компьютерные сети. Учебное пособие / Е.О. Новожилов. – М.: Academia, 2016. – 288 с.

64. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы: Учебник / В. Олифер, Н. Олифер. – СПб.: Питер, 2016. – 176 с.

65. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы: Учебник / В. Олифер, Н. Олифер. – СПб.: Питер, 2016. – 318 с.

66. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы: Учебник для ВУЗов / В. Олифер. – СПб.: Питер, 2012. – 944 с.

67. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. Стандарт третьего поколения / В.Г. Олифер, Н.А. Олифер.. – СПб.: Питер, 2013. – 944 с.

68. Попов, И.И. Компьютерные сети / И.И. Попов, Н.В. Максимов. – М.: Форум, 2004. – 336 с.

69. Прончев, Г.Б. Компьютерные коммуникации. Простейшие вычислительные сети: Учебное пособие / Г.Б. Прончев. – М.: КДУ, 2009. – 64 с.

70. Смелянский, Р.Л. Компьютерные сети. В 2 т.Т. 2. Сети ЭВМ / Р.Л. Смелянский. – М.: Academia, 2016. – 448 с.

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

71. Столлингс, В. Компьютерные сети, протоколы и технологии Интернета / В. Столлингс. – СПб.: ВHV, 2005. – 832 с.
72. . Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – СПб.: Питер, 2013. – 960 с.
73. Таненбаум, Э. Компьютерные сети / Э. Таненбаум. – СПб.: Питер, 2019. – 960 с.
74. Таненбаум, Э.С. Компьютерные сети / Э.С. Таненбаум, Д. Уэзеролл. – СПб.: Питер, 2018. – 512 с.
75. Шелухин, О.И. Обнаружение вторжений в компьютерные сети (сетевые аномалии): Учебное пособие для вузов / О.И. Шелухин, Д.Ж. Сакалема, А.С. Филинова. – М.: Гор. линия-Телеком, 2013. – 220 с.
76. Шелухин, О.И. Обнаружение вторжений в компьютерные сети (сетевые аномалии) / О.И. Шелухин. – М.: ГЛТ, 2013. – 220 с.
77. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
78. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>
79. Центр післядипломної освіти та підвищення кваліфікації. Режим доступу до ресурсу: <https://cpo.stu.cn.ua/>

					ВКРМ-123.21.0013.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

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

					ВКРМ-123.21.0013.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Немировський Д.В.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.						
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М-1,4		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи моніторингу мережевих принтерів.

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. № 42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи моніторингу мережевих принтерів.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.21.0013.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи моніторингу мережевих принтерів;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.21.0013.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi.

					ВКРМ-123.21.0013.00.00.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 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті Шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.21.0013.00.00.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 – 117 аркушів.

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 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2021 р.

					ВКРМ-123.21.0013.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Смірнов О.А.

*Дослідження та програмна реалізація
системи моніторингу мережевих принтерів*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 28

Літера: РП

Кропивницький – 2021 року

printer_monitoring.dpr

```
program printer_monitoring;
//файл проекта программного агента
uses
  Forms,
  main in 'main.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Кафедра КБПЗ – 2021 рік

Main.pas

```

unit main;
//реалізація модуля програмний агент
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, Printers, WinSpool, ExtCtrls, Grids, Menus,
  DB, DBTables, ScktComp, ShellApi;
const
  MY_MESSAGE = WM_USER + 4243; //(5267)

type
  JOB_INFO_1_ARRAY = array of JOB_INFO_1 ;
  TForm1 = class(TForm)
    ListBox1: TListBox;
    TreeView1: TTreeView;
    Memo1: TMemo;
    Button1: TButton;
    PrintDialog1: TPrintDialog;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    StringGrid1: TStringGrid;
    Timer1: TTimer;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    Edit1: TEdit;
    Label1: TLabel;
    Table1: TTable;
    DataSource1: TDataSource;
    Button5: TButton;
    ServerSocket1: TServerSocket;
    StatusBar1: TStatusBar;
    PopupMenu1: TPopupMenu;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    procedure FormCreate(Sender: TObject);
    procedure TreeView1DbClick(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure ServerSocket1ClientConnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientRead(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure PopupMenu1Popup(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure PopupRestore1Click(Sender: TObject);
    procedure PopupClose1Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
  end;

```

```

procedure N3Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N5Click(Sender: TObject);
private
  PrintersRoot:TTreeNode;
  PH:Cardinal;
  j: Integer;
  procedure btPrintMemoWithHeader(Memo: TCustomMemo; Printer: TPrinter;
    PrintDialog: TPrintDialog; HeaderText:
string;
                                Margin, BottomMargin, LeftMargin: Integer);
  //функції отримання списку шрифтів і їх розмір для поточного принтера
  function EnumFontFamilyProc(var lf: TLogFont; var tm: TNewTextMetric;
    FontType: integer; var Memo: TMemo): integer
    {$IFDEF WIN32} stdcall;
    {$ELSE} export;
    {$ENDIF}

  function EnumFontFamiliesProc(var lf: TLogFont; var tm: TNewTextMetric;
    FontType: integer; var Memo: TMemo): integer
    {$IFDEF WIN32} stdcall;
    {$ELSE} export;
    {$ENDIF}

  procedure WMPrinterStatus(var Msg:TMessage); message WM_SPOOLERSTATUS;
  procedure GetDeviceSettings(DevCtrl: TMemo);

  function GetSpoolerJobs(sPrinterName: string): JOB_INFO_1_ARRAY;

  { Private declarations }
public
  procedure WndProc (var Message: TMessage);

  { Public declarations }
end;
const MAX_PATH = 144;
var
  Form1: TForm1;
  MS: TMemoryStream;
  Ico_Message: integer=wm_User; {повідомлення}
  noIconData: TNotifyIconData; {дескриптор структури}
  HIcon1: HICON; {дескриптор іконки}
  FHandle: HWND; {дескриптор вікна (форми)}

  mayClose: boolean = False;

implementation

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
//обробка дій при відкритті форми
var PI:array[0..1023] of PRINTER_INFO_1;
    Needed,
    Returned:Cardinal;
    i, ii:integer;

begin
  j:=1;
  //відкриваємо сокет
  ServerSocket1.Open;
  //формування структури відображення
  StringGrid1.Cells[0, 1] :='Назва принтера';
  StringGrid1.Cells[0, 2] :='Назва комп'ютера';
  StringGrid1.Cells[0, 3] :='Ім'я користувача';
  StringGrid1.Cells[0, 4] :='Назва документу';
  StringGrid1.Cells[0, 5] :='Метод друку';
  StringGrid1.Cells[0, 6] :='Статус';

```

```

StringGrid1.Cells[0, 7] := 'Статус 2';
StringGrid1.Cells[0, 8] := 'Кількість сторінок';

for i := 0 to StringGrid1.ColCount - 1 do
for ii := 0 to StringGrid1.RowCount - 1 do
  StringGrid1.Cells[i, ii] := '';
//очистка даних компоненти
ListBox1.Clear;
//створення дерева принтерів
PrintersRoot:=TreeView1.Items.AddFirst(nil, 'Printers');
if not EnumPrinters(PRINTER_ENUM_LOCAL, nil, 1, @PI, SizeOf(PI), Needed, Returned)
  then
  ListBox1.Items.Add(SysErrorMessage(GetLastError));
For i:=0 to Returned-1 do
TreeView1.Items.AddChild(PrintersRoot, PI[i].pName);

end;

procedure TForm1.TreeView1DbClick(Sender: TObject);
//перевіряємо принтер на наявність при виборі його на формі
begin
if not OpenPrinter(PChar(TreeView1.Selected.Text), PH, nil) then
  ListBox1.Items.Add(SysErrorMessage(GetLastError));

end;

procedure TForm1.FormDestroy(Sender: TObject);
//при закритті форми
begin
  ClosePrinter(PH);
  //закриваємо сокет
  ServerSocket1.Close;
end;

procedure TForm1.WMPrinterStatus(var Msg: TMessage);
//перевіряємо статус принтерів
var i:integer;
    Job2:Array[0..1023] of JOB_INFO_2;
    Needed,
    Returned:Cardinal;

begin
Msg.Result:=0;
ListBox1.Items.Add('Jobs Left:'+IntToStr(Msg.WParamLo));
if not EnumJobs(PH, 0, 1024, 2, @Job2, SizeOf(Job2), Needed, Returned) then
  ShowMessage(SysErrorMessage(GetLastError));

For i:=0 to Returned-1 do
  With Job2[i] do
    ListBox1.Items.Add(Format('%s %s %s', [pPrinterName, pDocument, pUserName]))
end;

procedure TForm1.Button1Click(Sender: TObject);
//виведення інформації про драйвер принтера
var
  pDevice: pChar;
  pDriver: pChar;
  pPort: pChar;
  hDMode: THandle;
begin
//очистка поля виводу даних
Memo1.Clear;
//вибір принтера через діалог з користувачем
if PrintDialog1.Execute then begin
//виділяємо пам'ять
  GetMem(pDevice, cchDeviceName);

```

```

GetMem(pDriver, MAX_PATH);
GetMem(pPort, MAX_PATH);
Printer.GetPrinter(pDevice, pDriver, pPort, hDMode);
  if lStrLen(pDriver) = 0 then begin
    //зчитуємо з WIN.INI - файлу
    //дані про драйвер
    GetProfileString('devices', pDevice, '', pDriver, MAX_PATH);
    pDriver[pos(',', pDriver) - 1] := #0;
  end;
  //дані про порт
  if lStrLen(pPort) = 0 then begin
    GetProfileString('devices', pDevice, '', pPort, MAX_PATH);
    lStrCpy(pPort, @pPort[lStrLen(pPort) + 2]);
  end;
  //виводимо дані в компоненту відображення на формі
  Memol.Lines.Add('Друкуючий пристрій := ' + StrPas(pDevice));
  Memol.Lines.Add('Драйвер := ' + StrPas(pDriver));
  Memol.Lines.Add('Порт := ' + StrPas(pPort));
  //очистка пам'яті
  FreeMem(pDevice, cchDeviceName);
  FreeMem(pDriver, MAX_PATH);
  FreeMem(pPort, MAX_PATH);
end;

end;

procedure TForm1.Button2Click(Sender: TObject);
//виведення даних про метрику принтера
begin
  //Margin - відстань від верхньої частини до рядка, що друкується
  //LeftMargin - відступ зліва
  btPrintMemoWithHeader(Memol, printer, PrintDialog1,
    'Виводимо метрику вибраного принтера', 10, 1, 15);
end;

procedure TForm1.btPrintMemoWithHeader(Memo: TCustomMemo;
  Printer: TPrinter; PrintDialog: TPrintDialog; HeaderText: string; Margin,
  BottomMargin, LeftMargin: Integer);
//метрика принтера
var
  FirstPage: Boolean;
  i,
  LinesPerPage,
  CurrentLine,
  Line: Integer;
  PrintText: System.Text;
  LeftMarginString,
  Header: string;
begin
  if PrintDialog1.Execute then
    begin
      with Printer do
        begin
          AssignPrn(PrintText);
          Rewrite(PrintText);
          { Заповнюємо ліву частину рядка певною кількістю пропусків.}
          LeftMarginString := '';
          for i := 0 to LeftMargin do
            LeftMarginString := LeftMarginString + ' ';
          { Призначаємо принтеру такий же шрифт, як і в компоненті Memo.:}
          Canvas.Font := (Memol as TMemo).Font;
          { Обчислюємо кількість рядків на сторінці.}
          LinesPerPage := PageHeight div Canvas.TextHeight('X');
          LinesPerPage := LinesPerPage - 8 - Margin - BottomMargin;
          CurrentLine := LinesPerPage;
          FirstPage := True;
          { Друк Memo.}
          //Memol.Lines.Append(IntToStr(Memol.Lines.Count));
          for Line := 0 to Memol.Lines.Count - 1 do

```

```

begin
{ Якщо кінець сторінки, починаємо нову.}
  if CurrentLine >= LinesPerPage then
    begin
    { Друкуємо "Form Feed", якщо це не нова сторінка принтера.}
    if not FirstPage then Write(PrintText, #12);
    { Якщо не перша сторінка, то міняємо лист}
    FirstPage := False;
    { Друкуємо певну кількість порожніх рядків для верхнього поля.}
    for i := 0 to Margin do
      Writeln(PrintText, '');
    { Форматуємо і друкуємо рядок заголовка.}
    Header := Format('Сторінка %s %s %s %s'#13#10,
    [IntToStr(Printer.PageNumber), DateToStr(Date),
    TimeToStr(Time), HeaderText]);
    Write(PrintText, LeftMarginString);
    Writeln(PrintText, Header);
    { Скидаємо номер поточного рядка на 1 для наступної сторінки.}
    CurrentLine := 1;
    end;
    { Друкуємо рядок з Memo.}
    Write(PrintText, LeftMarginString);
    Writeln(PrintText, Memo1.Lines[Line]);
    Inc(CurrentLine);
    end;
  CloseFile(PrintText);
end;
end;

function TForm1.EnumFontFamilyProc(var lf: TLogFont;
var tm: TNewTextMetric; FontType: integer; var Memo: TMemo): integer;
//виведення даних на форму
begin
Memo.Lines.Add(StrPas(@lf.lfFaceName) + #32 + IntToStr(lf.lfHeight));
result := 1;
end;

function TForm1.EnumFontFamiliesProc(var lf: TLogFont;
var tm: TNewTextMetric; FontType: integer; var Memo: TMemo): integer;
begin
if FontType = TRUETYPE_FONTTYPE then
begin
Memo.Lines.Add(StrPas(@lf.lfFaceName) + #32 + 'All Sizes');
end
else
//запит на список шрифтів
EnumFontFamilies(Printer.Handle,
@lf.lfFaceName,
@TForm1.EnumFontFamilyProc,
LongInt(@Memo));
result := 1;
end;

procedure TForm1.Button3Click(Sender: TObject);
//виведення списку шрифтів, що підтримує принтер
var
tm: TTextMetric;
i: integer;
begin
//для функції виводу шрифтів
Listbox1.Items.Assign(Screen.Fonts);
if PrintDialog1.Execute then
begin
EnumFontFamilies(Printer.Handle, nil, @TForm1.EnumFontFamiliesProc,
LongInt(@Memo1));

```

```

    end;
end;

procedure TForm1.Button4Click(Sender: TObject);
//можливості принтера - обробка події натиснення кнопки
begin
    GetDeviceSettings(Mem01);
end;

procedure TForm1.GetDeviceSettings(DevCtrl: TМемо);
//список можливостей вибраного принтера
var
    Sep: string;
//-----
procedure MakeInt(S: string; key: Integer);
begin
    S := UpperCase(S);
    DevCtrl.Lines.Add(UpperCase(Format(' %36S = %d ',
    [s, GetDeviceCaps(Printer.Handle, Key)])));
end;
//-----
function StringToBits(S: string): string;
var
    H: string;
    i: Integer;
//-----
function SubStr(C: Char): string;
//переводимо тип Char в string
begin
    if c = '0' then SubStr := '0000';
    if c = '1' then SubStr := '0001';
    if c = '2' then SubStr := '0010';
    if c = '3' then SubStr := '0011';
    if c = '4' then SubStr := '0100';
    if c = '5' then SubStr := '0101';
    if c = '6' then SubStr := '0110';
    if c = '7' then SubStr := '0111';
    if c = '8' then SubStr := '1000';
    if c = '9' then SubStr := '1001';
    if c = 'A' then SubStr := '1010';
    if c = 'B' then SubStr := '1011';
    if c = 'C' then SubStr := '1100';
    if c = 'D' then SubStr := '1101';
    if c = 'E' then SubStr := '1110';
    if c = 'F' then SubStr := '1111';
end;
//-----
begin
    StringToBits := '';
    S := UpperCase(s);
    H := '';
    if Length(S) = 0 then Exit;
    if Length(S) = 1 then S := '0000' + S;
    if Length(S) = 2 then S := '000' + S;
    if Length(S) = 3 then S := '00' + S;
    if Length(S) = 4 then S := '0' + S;
    for i := 1 to Length(s) do
        H := H + ' ' + SubStr(S[i]);
    StringToBits := H;
end;
//-----
procedure MakeHex(S: string; key: Cardinal);
//переводимо string в Hex
var
    h: string;
begin
    S := UpperCase(S);
    h := Format('%X', [GetDeviceCaps(Printer.Handle, Key)]);
    if Length(H) = 0 then Exit;

```

```

if Length(H) = 1 then H := '0000' + H;
if Length(H) = 2 then H := '000' + H;
if Length(H) = 3 then H := '00' + H;
if Length(H) = 4 then H := '0' + H;
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(SEP);
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(Format('%37S = Flags(%S) Key(%S)',
[s, h, StringToBits(H)]
));
// (( GetDeviceCaps(Printer.Handle,Key),
end;
//-----
procedure MakeFlag(S: string; key, subKey: Cardinal);
//перевіряємо значення прапорця
var
i: Cardinal;
begin
S := UpperCase(S);
i := GetDeviceCaps(Printer.Handle, Key);
if i and SubKey <> 0 then
DevCtrl.Lines.Add(Format(' %34S = Flag(%4S) Key(%6D,%S)',
[s, 'ON ', SubKey, StringToBits(Format('%x', [SubKey]))]))
else
DevCtrl.Lines.Add(Format(' %34S = Flag(%4S) Key(%6D,%S)',
[s, 'OFF', SubKey, StringToBits(Format('%x', [SubKey]))]))
end;
//-----
function TechnoToStr(i: Integer): string;
//переводимо тип обладнання в рядкову константу
begin
TechnoToStr := '#ERROR# is Unknwon';
case i of
DT_PLOTTER: TechnoToStr := 'Vector Plotter';
DT_RASDISPLAY: TechnoToStr := 'Raster Display';
DT_RASPRINTER: TechnoToStr := 'Raster Printer';
DT_RASCAMERA: TechnoToStr := 'Raster Camera';
DT_CHARSTREAM: TechnoToStr := 'Character Stream';
DT_METAFILE: TechnoToStr := 'Metafile';
DT_DISPFILE: TechnoToStr := 'Display file';
end;
end;
//--реалізація процедури
//-----
begin
DevCtrl.SetFocus;
DevCtrl.Visible := False;
if Printer.PrinterIndex < 0 then Exit;
// отримання пристрою
try
if not (GetMapMode(Printer.Handle) = MM_TEXT) then
SetMapMode(Printer.Handle, MM_Text);
//формування документа
DevCtrl.Clear;
Sep :=
'
';
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(' PRINTER : ' + Printer.Printers[Printer.PrinterIndex]);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(Format('%36S = %D', ['NUMBER of COPIES', Printer.Copies]));
if Printer.Orientation = poLandscape then
DevCtrl.Lines.Add(Format('%36S = LANDSCAPE', ['ORIENTATION']));
if Printer.Orientation = poPortrait then
DevCtrl.Lines.Add(Format('%36S = PORTRAIT', ['ORIENTATION']));

```

```

//отримання і виведення отриманих даних на компонент форми
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
//версія драйвера
MakeInt('DRIVERVERSION', DRIVERVERSION);
DevCtrl.Lines.Add(Format(' %36S = %S', ['TECHNOLOGY',
UpperCase(TechnoToStr(GetDeviceCaps(Printer.Handle, Technology)))]));
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
//розміри листа
MakeInt('WIDTH [mm]', HORZSIZE);
MakeInt('HEIGHT [mm]', VERTSIZE);
MakeInt('WIDTH [pix]', HORZRES);
MakeInt('HEIGHT [pix]', VERTRES);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeInt('Physical Width [pix]', PHYSICALWIDTH);
MakeInt('Physical Height [pix]', PHYSICALHEIGHT);
MakeInt('Physical Offset X [pix]', PHYSICALOFFSETX);
MakeInt('Physical Offset Y [pix]', PHYSICALOFFSETY);
MakeInt('SCALING FACTOR X', SCALINGFACTORX);
MakeInt('SCALING FACTOR Y', SCALINGFACTORY);
//розподільна здатність друку
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeInt('horizontal [DPI]', LOGPIXELSX);
MakeInt('vertical [DPI]', LOGPIXELSY);
MakeInt('BITS PER PIXEL', BITSPIXEL);
MakeInt('COLOR PLANES', PLANES);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeInt('NUMBER OF BRUSHES', NUMBRUSHES);
MakeInt('NUMBER OF PENS', NUMPENS);
MakeInt('NUMBER OF FONTS', NUMFONTS);
MakeInt('NUMBER OF COLORS', NUMCOLORS);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeInt('ASPECT Ratio X [DPI]', ASPECTX);
MakeInt('ASPECT Ratio Y [DPI]', ASPECTY);
MakeInt('ASPECT Ratio XY [DPI]', ASPECTXY);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeInt('SIZE OF PALETTE', SIZEPALETTE);
MakeInt('RESERVED TO SYSTEM PALETTE **', NUMRESERVED);
MakeInt('ACTUAL RASTER RESOLUTION **', COLORRES);
DevCtrl.Lines.Add('');
DevCtrl.Lines.Add(' **...only true if KEY RASTERCAPS(RC_PALETTE)= ON');
MakeFlag('... KEY RASTERCAPS (RC_PALETTE)', RasterCaps, RC_PALETTE);
DevCtrl.Lines.Add('');
MakeHex('Clipping Capabilities ', ClipCaps);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('No Support ', ClipCaps, CP_NONE);
MakeFlag('Support Rectangles', ClipCaps, CP_RECTANGLE);
MakeFlag('Support PolyRegion 32 Bit', ClipCaps, CP_REGION);
MakeHex('Raster Printing Flags ', RasterCaps);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('Support Bitmap Transfer', RasterCaps, RC_BITBLT);
MakeFlag('Support Banding', RasterCaps, RC_BANDING);
MakeFlag('Support Scaling', RasterCaps, RC_SCALING);
MakeFlag('Support Bitmaps > 64 kByte', RasterCaps, RC_BITMAP64);
MakeFlag('Support features of Win 2.0', RasterCaps, RC_GDI20_OUTPUT);
MakeFlag('Support Set~/GetDIBITS()', RasterCaps, RC_DI_BITMAP);
MakeFlag('Support Palette Devices', RasterCaps, RC_PALETTE);
MakeFlag('Support SetDIBitsToDevice()', RasterCaps, RC_DIBTODEV);
MakeFlag('Support Floodfill', RasterCaps, RC_FLOODFILL);
MakeFlag('Support StretchBlt()', RasterCaps, RC_STRETCHBLT);
MakeFlag('Support StretchBID()', RasterCaps, RC_STRETCHDIB);

```

```

MakeFlag('Support DIBS', RasterCaps, RC_DEVBITS);
MakeHex('Curve Printion Flages', CurveCaps);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('No Curve support', CurveCaps, CC_NONE);
MakeFlag('Support Circles', CurveCaps, CC_Circles);
MakeFlag('Support Pie', CurveCaps, CC_PIE);
MakeFlag('Support Arces', CurveCaps, CC_CHORD);
MakeFlag('Support Ellipses', CurveCaps, CC_ELLIPSEs);
MakeFlag('Support WIDE FRAMES', CurveCaps, CC_WIDE);
MakeFlag('Support STYLED FRAMES', CurveCaps, CC_STYLED);
MakeFlag('Support WIDE&STYLED FRAMES', CurveCaps, CC_WIDESTYLED);
MakeFlag('Support INTERIORS', CurveCaps, CC_INTERIORS);
MakeFlag('Support ROUNDRECT', CurveCaps, CC_ROUNDRECT);
MakeHex('Line & Polygon Printing Flags', LineCaps);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('No Line Support', LineCaps, LC_NONE);
MakeFlag('Support Polylines', LineCaps, LC_PolyLine);
MakeFlag('Support Marker', LineCaps, LC_Marker);
MakeFlag('Support PolyMarker', LineCaps, LC_PolyMarker);
MakeFlag('Support Wide Lines', LineCaps, LC_WIDE);
MakeFlag('Support STYLED Lines', LineCaps, LC_STYLED);
MakeFlag('Support WIDE&STYLED Lines', LineCaps, LC_WIDESTYLED);
MakeFlag('Support Lines Interiors', LineCaps, LC_INTERIORS);
MakeHex('Polygon (Areal) Printing Flags', POLYGONALCAPS);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('No Polygon Support', PolygonalCaps, PC_NONE);
MakeFlag('Filling Alternate Polygons', PolygonalCaps, PC_POLYGON);
MakeFlag('Drawing Rectangles', PolygonalCaps, PC_RECTANGLE);
MakeFlag('Filling Winding Polygons', PolygonalCaps, PC_WINDPOLYGON);
MakeFlag('Drawing Trapezoid (??Flag)', PolygonalCaps, PC_Trapezoid);
MakeFlag('Drawing a ScanLine', PolygonalCaps, PC_SCANLINE);
MakeFlag('Drawing Wide Border', PolygonalCaps, PC_WIDE);
MakeFlag('Drawing Styled Border', PolygonalCaps, PC_STYLED);
MakeFlag('Drawing WIDE&STYLED Border', PolygonalCaps, PC_WIDESTYLED);
MakeFlag('Drawing Interiors', PolygonalCaps, PC_INTERIORS);
MakeHex('Text Printing Flags', TEXTCAPS);
DevCtrl.Lines.Add(sep);
DevCtrl.Lines.Add('');
MakeFlag('Support Character Output Precision', TextCaps, TC_OP_CHARACTER);
MakeFlag('Support Stroke Output Precision', TextCaps, TC_OP_STROKE);
MakeFlag('Support Stroke Clip Precision', TextCaps, TC_CP_STROKE);
MakeFlag('Support 90° Character Rotation', TextCaps, TC_CR_90);
MakeFlag('Support any Character Rotaion', TextCaps, TC_CR_ANY);
MakeFlag('Support Character Scaling in X&Y', TextCaps, TC_SF_X_YINDEP);
MakeFlag('Support Character Scaling REAL', TextCaps, TC_SA_DOUBLE);
MakeFlag('Support Character Scaling RATIONAL', TextCaps, TC_SA_INTEGER);
MakeFlag('Support Character Scaling EXACT', TextCaps, TC_SA_CONTIN);
MakeFlag('Support Character Weight REAL', TextCaps, TC_EA_DOUBLE);
MakeFlag('Support Character Italic', TextCaps, TC_IA_ABLE);
MakeFlag('Support Character Underline', TextCaps, TC_UA_ABLE);
MakeFlag('Support Character Strikeout', TextCaps, TC_SO_ABLE);
MakeFlag('Support Character as RASTER FONT', TextCaps, TC_RA_ABLE);
MakeFlag('Support Character as VECTOR FONT', TextCaps, TC_VA_ABLE);
MakeFlag('Reserved Flag ???', TextCaps, TC_Reserved);
MakeFlag('DEVICE NOT USE a SCROLLBIT BLOCK ?', TextCaps, TC_SCROLLBLT);
DevCtrl.Lines.Insert(0, '..THE RESULTS ARE:');
except
  // MessageDlg('The Current Printer is not valid ! ',
  // mtError,[mbok],0);
  Printer.PrinterIndex := -1;
  DevCtrl.Lines.Add(' ! Принтер не выбраный!');
end;
DevCtrl.Visible := True;
DevCtrl.SetFocus;
end;

```

```

function TForm1.GetSpoolerJobs(sPrinterName: string): JOB_INFO_1_ARRAY;
//отримати завдання з вибраного принтера
var
  i: Integer;
  hPrinter: THandle;
  bResult: Boolean;
  cbBuf: DWORD;
  pcbNeeded: DWORD;
  pcReturned: DWORD;
  aJobs: array[0..99] of JOB_INFO_1;
begin
  cbBuf := 1000;
  bResult := OpenPrinter(PChar(sPrinterName), hPrinter, nil);
  if not bResult then
  begin
    ShowMessage('Не можливо відкрити принтер! ');
    exit;
  end;
  bResult := EnumJobs(hPrinter, 0, Length(aJobs), 1, @aJobs, cbBuf, pcbNeeded,
    pcReturned);
  if not bResult then
  begin
    ShowMessage('Помилка при отриманні завдання, відправленого на принтер');
    exit;
  end;
  ClosePrinter(hPrinter);
  for i := 0 to pcReturned - 1 do
  begin
    if aJobs[i].pDocument <> nil then
    begin
      SetLength(Result, Length(Result) + 1);
      Result[Length(Result) - 1] := aJobs[i];
    end;
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
//обробляється при спрацюванні таймера
var
  i, ii: Integer;
  aJobs: JOB_INFO_1_ARRAY;

begin
  //отримуємо завдання
  aJobs := GetSpoolerJobs(Edit1.Text);
  //виводимо завдання на форму
  for i := 0 to Length(aJobs)-1 do
  begin
    StringGrid1.Cells[i+j, 1] := aJobs[i].pPrinterName;
    StringGrid1.Cells[i+j, 2] := aJobs[i].pMachineName;
    StringGrid1.Cells[i+j, 3] := aJobs[i].pUserName;
    StringGrid1.Cells[i+j, 4] := aJobs[i].pDocument;
    StringGrid1.Cells[i+j, 5] := aJobs[i].pDatatype;
    StringGrid1.Cells[i+j, 6] := aJobs[i].pStatus;
    StringGrid1.Cells[i+j, 7] := IntToStr(aJobs[i].Status);
    StringGrid1.Cells[i+j, 8] := IntToStr(aJobs[i].TotalPages);
    //визначення статусу
    case aJobs[i].Status of
      JOB_STATUS_PAUSED: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_PAUSED';
      JOB_STATUS_ERROR: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_ERROR';
      JOB_STATUS_DELETING: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_DELETING';
      JOB_STATUS_SPOOLING: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_SPOOLING';
      JOB_STATUS_PRINTING: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_PRINTING';
      JOB_STATUS_OFFLINE: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_OFFLINE';
      JOB_STATUS_PAPEROUT: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_PAPEROUT';
      JOB_STATUS_PRINTED: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_PRINTED';
      JOB_STATUS_DELETED: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_DELETED';
      JOB_STATUS_BLOCKED_DEVQ: StringGrid1.Cells[i+j, 7] :=

```

```

'JOB_STATUS_BLOCKED_DEVO';
JOB_STATUS_USER_INTERVENTION: StringGrid1.Cells[i+j, 7] :=
'JOB_STATUS_USER_INTERVENTION';
JOB_STATUS_RESTART: StringGrid1.Cells[i+j, 7] := 'JOB_STATUS_RESTART';
JOB_POSITION_UNSPECIFIED: StringGrid1.Cells[i+j, 7] :=
'JOB_POSITION_UNSPECIFIED';
else
StringGrid1.Cells[i+j, 7] := 'Невідомий статус...';
end;
//поновлення структури відображення
end;
StringGrid1.Refresh;
StringGrid1.Cells[0, 1] := 'Назва принтера';
StringGrid1.Cells[0, 2] := 'Назва комп'ютера';
StringGrid1.Cells[0, 3] := 'Ім'я користувача';
StringGrid1.Cells[0, 4] := 'Назва документу';
StringGrid1.Cells[0, 5] := 'Метод друку';
StringGrid1.Cells[0, 6] := 'Статус';
StringGrid1.Cells[0, 7] := 'Статус 2';
StringGrid1.Cells[0, 8] := 'Кількість сторінок';

if (StringGrid1.Cells[1, 1]<>'') and (StringGrid1.Cells[1, 8]<>'') then
begin
//запис даних у БД локального моніторингу
Table1.Open;
Table1.AppendRecord([nil, TimeToStr(SysUtils.Time)+'_'+
DateToStr(SysUtils.Date),
StringGrid1.Cells[1, 1], StringGrid1.Cells[1,
2],StringGrid1.Cells[1, 3],
StringGrid1.Cells[1, 4],StringGrid1.Cells[1,
5],StringGrid1.Cells[1, 6],
StringGrid1.Cells[1, 7],StringGrid1.Cells[1, 8]]);
Table1.Close;
end;
end;

procedure TForm1.Button5Click(Sender: TObject);
//відправити на сервер в інтерактивному режимі
var
Size: Integer;
P: ^Byte;
begin
//зупиняємо таймер
Timer1.Enabled:=False;
//створюємо буфер для файлу
MS:=TMemoryStream.Create;
//завантажуємо файл в буфер
MS.LoadFromFile('LocalBD.db');

//відправляємо інформацію про файл на приймаючу сторону
ServerSocket1.Socket.Connections[0].SendText('file#'+
LowerCase(ExtractFileName('LocalBD.db'))+'#'+IntToStr(MS.Size)+'#');
//переходимо на початок файлу
MS.Position:=0;
//завантажуємо у змінну файл
P:=MS.Memory;
//відправляємо файл
Size:=ServerSocket1.Socket.Connections[0].SendBuf(P^,ms.Size);
//виводимо статистику в рядок статусу
StatusBar1.SimpleText:='Відправлено'+IntToStr(Size)+'з'+IntToStr(MS.Size)+'
байт'
end;

procedure TForm1.ServerSocket1ClientConnect(Sender: TObject;
Socket: TCustomWinSocket);
//виведення службової інформації в рядок статусу
begin
StatusBar1.SimpleText:='З'єднання встановлено';
end;

```

```

procedure TForm1.ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
//виведення службової інформації в рядок статусу
begin
  StatusBar1.SimpleText:='З'єднання відсутнє';
end;

procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  //якщо клієнт прийняв файл, то
  if Socket.ReceiveText='end' then
    begin
      //виведення службової інформації в рядок статусу
      StatusBar1.SimpleText:='Файл відправлений успішно';
      //очистка буфера
      MS.Free;
      //очищуємо дані в таблиці БД
      Table1.EmptyTable;
      //запускаємо таймер
      Timer1.Enabled:=True;
    end;
end;

procedure TForm1.PopupMenu1Popup(Sender: TObject);
begin
  //видалити меню автоматично при втраті фокуса
  SetForegroundWindow(Form1.Handle);
end;

procedure TForm1.WndProc(var Message: TMessage);
//обробка користувацьких повідомлень
begin
  if Message.Msg = Ico_Message then
    begin
      //якщо була відпущена кнопка
      if Message.LParam = WM_LBUTTONDOWN then
        begin
          mayClose:= False;
          //відновлюємо вікно
          Form1.Show;
          //розташування поверх усіх вікон
          Application.BringToFront;
          DeallocateHWnd(FHandle);
          //видалення із пам'яті дескриптора вікна
          Shell_NotifyIcon(NIM_DELETE, @noIconData);
          //обробляємо системні події
          Application.ProcessMessages;
        end;
      end;
      //обробляємо подію натиснення правої кнопки мишки
      if Message.LParam = WM_RBUTTONDOWN then
        //контекстне меню
        Form1.PopupMenu1.Popup(Screen.Width-32, Screen.Height-32);
    end;

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
//обробка подій при закритті форми
var st: String;
begin
  if mayClose = False then
    begin
      //не закриваємо програму
      CanClose:= False;
      //при повторному запиті закриваємо
      mayClose:= True;
      //дескриптор вікна
      FHandle:= AllocateHWnd(WndProc);

```

```

//дескриптор іконки
HIcon1:= CopyIcon(Application.Icon.Handle);

with noIconData do
begin
  //визначаємо розмір структури
  cbSize:=SIZEOF(TNotifyIconData);
  //дескриптор вікна
  Wnd:= FHandle;
  //одна іконка
  uID:= 0;
  //активність усім прапорцям
  UFlags:= NIF_MESSAGE or NIF_ICON or NIF_TIP;
  //впливаюча підказка
  szTip:= 'Моніторинг друкуючих пристроїв';
  //дескриптор іконки
  hIcon:= HIcon1;
  //повідомлення, визначене користувачем
  uCallbackMessage:= Ico_Message;
end;
  //створити іконку
Shell_NotifyIcon(NIM_ADD,@noIconData);
  //сховати вікно
Form1.Hide;
end else
begin
  //знищити дескриптор вікна
  DeallocateHWnd(FHandle);
  //видалити іконку
  Shell_NotifyIcon(NIM_DELETE, @noIconData);
  CanClose:= True;
end;
end;

procedure TForm1.PopupClose1Click(Sender: TObject);
begin
  //закрити форму
  Form1.Close;
end;

procedure TForm1.PopupRestore1Click(Sender: TObject);
begin
  //не можна закривати програму
  mayClose:= False;
  //показати вікно
  Form1.Show;
  //розмістити поверх усіх вікон
  Application.BringToFront;
  //знищити дескриптор
  DeallocateHWnd(FHandle);
  //знищити іконку
  Shell_NotifyIcon(NIM_DELETE, @noIconData);
  //обробка системних подій
  Application.ProcessMessages;
end;

procedure TForm1.N11Click(Sender: TObject);
  //закриття програми
begin
  mayClose:= True;
  Close;
end;

procedure TForm1.N8Click(Sender: TObject);
  //згорнути у трей
begin
  Close;
end;

```

```
procedure TForm1.N2Click(Sender: TObject);
//обробка командного меню - виведення інформації про драйвер принтера
begin
  Button1.Click;
end;

procedure TForm1.N3Click(Sender: TObject);
//обробка командного меню - виведення інформації про можливості принтера
begin
  Button4.Click;
end;

procedure TForm1.N4Click(Sender: TObject);
//обробка командного меню - виведення інформації про метрику принтера
begin
  Button2.Click;
end;

procedure TForm1.N5Click(Sender: TObject);
//обробка командного меню - виведення інформації про шрифти
begin
  Button3.Click;
end;

end.
```

Кафедра КБПЗ – 2021 рік

//програмна реалізація графічного інтерфейсу програмного агента моніторингу друкуючих мережевих пристроїв

```
object Form1: TForm1
  Left = 219
  Top = 28
  Width = 913
  Height = 705
  Caption = 'Моніторинг мережевого друкуючого пристрою'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  Menu = MainMenu1
  OldCreateOrder = False
  OnCloseQuery = FormCloseQuery
  OnCreate = FormCreate
  OnDestroy = FormDestroy
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 592
    Top = 13
    Width = 286
    Height = 13
    Caption = 'Вказуємо принтер, моніторинг якого підлягає контролю'
  end
  object ListBox1: TListBox
    Left = 592
    Top = 305
    Width = 297
    Height = 289
    ItemHeight = 13
    TabOrder = 0
  end
  object TreeView1: TTreeView
    Left = 592
    Top = 72
    Width = 297
    Height = 226
    Indent = 19
    TabOrder = 1
    OnDbClick = TreeView1DbClick
  end
  object Memo1: TMemo
    Left = 8
    Top = 305
    Width = 577
    Height = 289
    Lines.Strings = (
      '')
    TabOrder = 2
  end
  object Button1: TButton
    Left = 8
    Top = 600
    Width = 169
    Height = 25
    Caption = 'Драйвер принтера'
    TabOrder = 3
    OnClick = Button1Click
  end
  object Button2: TButton
    Left = 368
```

```
Top = 600
Width = 169
Height = 25
Caption = 'Метрика принтера'
TabOrder = 4
OnClick = Button2Click
end
object Button3: TButton
Left = 544
Top = 600
Width = 169
Height = 25
Caption = 'Виведення списку шрифтів'
TabOrder = 5
OnClick = Button3Click
end
object Button4: TButton
Left = 192
Top = 600
Width = 169
Height = 25
Caption = 'Вивести можливості принтера'
TabOrder = 6
OnClick = Button4Click
end
object StringGrid1: TStringGrid
Left = 8
Top = 8
Width = 577
Height = 289
ColCount = 2
DefaultColWidth = 280
RowCount = 10
TabOrder = 7
RowHeights = (
  24
  24
  24
  24
  24
  24
  24
  24
  24
  24)
end
object Edit1: TEdit
Left = 592
Top = 32
Width = 289
Height = 24
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -13
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 8
Text = 'Canon LASER SHOT LBP-1120'
end
object Button5: TButton
Left = 720
Top = 600
Width = 169
Height = 25
Caption = 'Відправити на сервер'
TabOrder = 9
OnClick = Button5Click
end
```

```
object StatusBar1: TStatusBar
  Left = 0
  Top = 632
  Width = 905
  Height = 19
  Panels = <>
  SimplePanel = True
end
object PrintDialog1: TPrintDialog
  Left = 16
  Top = 16
end
object Timer1: TTimer
  Interval = 500
  OnTimer = Timer1Timer
  Left = 16
  Top = 48
end
object MainMenu1: TMainMenu
  Left = 48
  Top = 16
  object N1: TMenuItem
    Caption = 'Виведення службової інформації'
    object N2: TMenuItem
      Caption = 'Інформація про драйвер принтера'
      OnClick = N2Click
    end
    object N3: TMenuItem
      Caption = 'Вивести можливості принтера'
      OnClick = N3Click
    end
    object N4: TMenuItem
      Caption = 'Вивести метрику принтера'
      OnClick = N4Click
    end
    object N5: TMenuItem
      Caption = 'Вивести список підтримуваних шрифтів друкувачим пристроєм'
      OnClick = N5Click
    end
  end
  object N6: TMenuItem
    Caption = 'Передати дані на сервер'
  end
  object N7: TMenuItem
    Caption = 'Допомога'
  end
  object N8: TMenuItem
    Caption = 'Згорнути у трей'
    OnClick = N8Click
  end
  object N11: TMenuItem
    Caption = 'Вихід'
    OnClick = N11Click
  end
end
object Table1: TTable
  TableName = 'LocalBD.db'
  Left = 80
  Top = 16
end
object DataSource1: TDataSource
  DataSet = Table1
  Left = 80
  Top = 48
end
object ServerSocket1: TServerSocket
  Active = False
  Port = 4444
  ServerType = stNonBlocking
```

```
OnClientConnect = ServerSocket1ClientConnect
OnClientDisconnect = ServerSocket1ClientDisconnect
OnClientRead = ServerSocket1ClientRead
Left = 112
Top = 48
end
object PopupMenu1: TPopupMenu
  OnPopup = PopupMenu1Popup
  Left = 48
  Top = 48
  object N9: TMenuItem
    Caption = 'Розгорнути'
    OnClick = PopupRestore1Click
  end
  object N10: TMenuItem
    Caption = 'Закрити'
    OnClick = PopupClose1Click
  end
end
end
```

Кафедра_КБПЗ_2021 рік

ServerMain.dpr

```
program ServerMain;
//файл проекта. Реалізація збереження даних моніторинга в мережевій БД
uses
  Forms,
  Server_main in 'Server_main.pas' {Form1},
  Druk in 'Druk.pas' {TGridReport};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TTGridReport, TGridReport);
  Application.Run;
end.
```

Кафедра КБПЗ – 2021 рік

Server_main.pas

```

unit Server_main;
//Реалізація збереження даних моніторинга в мережевій БД
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ScktComp, ExtCtrls, ComCtrls, StdCtrls, DB, DBTables, Grids,
  DBGrids, Menus, ShellApi;
const
  MY_MESSAGE = WM_USER + 4242; //(5266)

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    ServerSocket1: TServerSocket;
    ClientSocket1: TClientSocket;
    Label1: TLabel;
    Label2: TLabel;
    StatusBar1: TStatusBar;
    Table1: TTable;
    DataSource1: TDataSource;
    Table2: TTable;
    DataSource2: TDataSource;
    DBGrid1: TDBGrid;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    PopupMenu1: TPopupMenu;

    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;      procedure ClientSocket1Read(Sender: TObject; Socket:
TCustomWinSocket);
    procedure writing(Text:string);
    procedure writingBD;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure PopupMenu1Popup(Sender: TObject);
    procedure PopupRestore1Click(Sender: TObject);
    procedure PopupClose1Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
  private
    temp: String;
    Name: string; //ім'я файлу
    Size: Integer;//Розмір файлу
    Receive: Boolean;//Режим роботи - стан прапорця
    MS: TMemoryStream;//буфер для файлу

    { Private declarations }
  public
    IsRecStart: Boolean; // Ознака початку отримання даних
    BmpFile: File; // Файл для передачі
    procedure WndProc (var Message: TMessage);
    { Public declarations }
  end;

var
  Form1: TForm1;
  Ico_Message: integer=wm_User; {повідомлення}
  noIconData: TNotifyIconData; {дескриптор структури}
  HIcon1: HICON; {дескриптор іконки}

```

```

FHandle: HWND; {дескриптор вікна (форми)}

mayClose: boolean = False;

implementation

{$R *.dfm}

procedure TForm1.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);
  //обробка пакетів інформації, що надходить по мережі
var
  Rtext: string;
begin
  //заносимо у змінну данні, що прийшли
  Rtext:=Socket.ReceiveText;
  //перевіряємо прапорець
  if Receive then
    //якщо прийом пакету даних, то перехід на процедуру запису
    Writing(RText)
  else
    //якщо нові дані, то
    //визначаємо тип даних
    //якщо передача файлу, то
    if Copy(Rtext,0,Pos('#',Rtext)-1)='file' then
      begin
        //створюємо потік
        MS:=TMemoryStream.Create;
        //видаляємо службові маркери
        Delete(Rtext,1,Pos('#',Rtext));
        //читаємо ім'я файлу
        Name:= Copy(Rtext,0,Pos('#',Rtext)-1);
        Delete(Rtext,1,Pos('#',Rtext));
        //читаємо дані про розмір файлу
        Size:=StrToInt(Copy(Rtext,0,Pos('#',Rtext)-1));
        Delete(Rtext,1,Pos('#',Rtext));
        Label1.Caption:='Розмір файлу: '+IntToStr(Size) +' байт';
        Label2.Caption:='Ім'я файлу: '+ Name;
        //прапорець переводимо в режим прийому пакетів даних
        Receive:=True;
        //процедура запису
        Writing(RText);
      end;
end;

procedure TForm1.Writing(Text: string);
//процедура запису
begin
  if MS.Size<Size then
    //якщо прийнято байт менше розміру файлу, то
    //записуємо у буфер
    MS.Write(Text[1],Length(Text));
    //виводимо дані в рядок статусу
    StatusBar1.SimpleText:='Прийнято'+IntToStr(MS.Size)+'з'+IntToStr(Size);
    //якщо файл прийнятий
    if MS.Size=Size then
      begin
        //встановлюємо прапорець в False (кінець прийому пакетів передачі даних)
        Receive:=False;
        //на початок буфера
        MS.Position:=0;
        //збереження файлу
        MS.SaveToFile(Name);
        //відправка успішності прийняття файлу
        ClientSocket1.Socket.SendText('end');
        //очистка буфера
        MS.Free;
        //виведення в рядок статусу інформації

```

```

StatusBar1.SimpleText:='Файл одержаний';
//деактивація сокета
ClientSocket1.Active:=False;
//виклик процедури запису звіту
writingBD;
//видаляємо звіт
if DeleteFile('LocalBD.db') then
begin
  StatusBar1.SimpleText:='Сервер готовий до прийому нових звітів';
  //чекаємо видалення
  sleep(50);
  //якщо видалений успішно, то
  if not FileExists('LocalBD.db') then
  //активація сокета
  ClientSocket1.Active:=True;
end;
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
//відкриття сокетів при відкритті форми
begin
  ClientSocket1.Active:=True;
  if FileExists('LocalBD.db') then DeleteFile('LocalBD.db');
end;

procedure TForm1.FormDestroy(Sender: TObject);
//закриття сокетів при закритті програми
begin
  ClientSocket1.Active:=False;
end;

procedure TForm1.writingBD;
//процедура запису даних з БД звіту клієнта
var i:Integer;
begin
  if FileExists('LocalBD.db') then
  begin
    //відкриваємо таблиці БД
    Table1.Open;
    Table2.Open;
    //якщо в файлі-звіті є дані
    if Table1.RecordCount>0 then
    begin
      //від 1 до останнього рядка
      for i:=0 to Table1.RecordCount-1 do
      begin
        Table1.RecNo:=i;
        //додаємо дані звіту у загальну(мережеву) таблицю БД
        Table2.AppendRecord([nil,Table1.FieldByName('Data_Time').AsString,
        Table1.FieldByName('Name_printer').AsString,
        Table1.FieldByName('Name_PC').AsString,
        Table1.FieldByName('Name_User').AsString,
        Table1.FieldByName('Name_DOC').AsString,
        Table1.FieldByName('Metod_Druk').AsString,
        Table1.FieldByName('Status1').AsString,
        Table1.FieldByName('Status2').AsString,Table1.FieldByName('Kil_storinok').AsString]);
      end;
      //закрити таблицю звіту
      Table1.Close;
    end;
  end;
end;

procedure TForm1.N2Click(Sender: TObject);
//очистка таблиці БД моніторингу

```

```

begin
  ClientSocket1.Active:=False;
  Table2.Close;
  if FileExists('ServerBD.db') then Table2.EmptyTable;
  Table2.Open;
  ClientSocket1.Active:=True;
end;

procedure TForm1.WndProc(var Message: TMessage);
//обробка користувацьких повідомлень
begin
  if Message.Msg = Ico_Message then
    begin
      //якщо була відпущена кнопка
      if Message.LParam = WM_LBUTTONDOWN then
        begin
          mayClose:= False;
          //відновлюємо вікно
          Form1.Show;
          //розташування поверх усіх вікон
          Application.BringToFront;
          DeallocateHWnd(FHandle);
          //видалення із пам'яті дескриптора вікна
          Shell_NotifyIcon(NIM_DELETE, @noIconData);
          //обробляємо системні події
          Application.ProcessMessages;
        end;
      end;
      //обробляємо події натиснення правої кнопки мишки
      if Message.LParam = WM_RBUTTONDOWN then
        //контекстне меню
        Form1.PopupMenu1.Popup(Screen.Width-32, Screen.Height-32);
    end;

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
var st: String;
begin
  if mayClose = False then
    begin
      //не закриваємо програму
      CanClose:= False;
      //при повторному запиті закриваємо
      mayClose:= True;
      //дескриптор вікна
      FHandle:= AllocateHWnd(WndProc);
      //дескриптор іконки
      HIcon1:= CopyIcon(Application.Icon.Handle);

      with noIconData do
        begin
          //визначаємо розмір структури
          cbSize:=SIZEOF(TNotifyIconData);
          //дескриптор вікна
          Wnd:= FHandle;
          //одна іконка
          uID:= 0;
          //активність усім прапорцям
          UFlags:= NIF_MESSAGE or NIF_ICON or NIF_TIP;
          //впливаюча підказка
          szTip:= 'Моніторинг друкуючих пристроїв';
          //дескриптор іконки
          hIcon:= HIcon1;
          //повідомлення, визначене користувачем
          uCallbackMessage:= Ico_Message;
        end;
      //створити іконку
      Shell_NotifyIcon(NIM_ADD, @noIconData);
      //сховати вікно
      Form1.Hide;
    end;

```

```

end else
begin
    //знищити дескриптор вікна
    DeallocateHWnd(FHandle);
    //видалити іконку
    Shell_NotifyIcon(NIM_DELETE, @noIconData);
    CanClose:= True;
end;
end;

procedure TForm1.PopupMenu1Popup(Sender: TObject);
begin
    //видалити меню автоматично при втраті фокуса
    SetForegroundWindow(Form1.Handle);
end;

procedure TForm1.PopupClose1Click(Sender: TObject);
begin
    //закрити форму
    Form1.Close;
end;

procedure TForm1.PopupRestore1Click(Sender: TObject);
begin
    //не можна закривати програму
    mayClose:= False;
    //показати вікно
    Form1.Show;
    //розмістити поверх усіх вікон
    Application.BringToFront;
    //знищити дескриптор
    DeallocateHWnd(FHandle);
    //знищити іконку
    Shell_NotifyIcon(NIM_DELETE, @noIconData);
    //обробка системних подій
    Application.ProcessMessages;
end;

procedure TForm1.N4Click(Sender: TObject);
//закриття програми
begin
    mayClose:= True;
    Close;
end;

procedure TForm1.N7Click(Sender: TObject);
//згорнути у трей
begin
    Close;
end;
end.

```

```

unit Druk;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, QuickRpt, DBGrids, Grids, DB, QRCtrls, QRPrntr;

type
  TTGridReport = class(TForm)
    GridRep: TQuickRep;
  private
    { Private declarations }
  public
    { Public declarations }
    procedure Preview (Grid: TDBGrid);
  end;

var
  TGridReport: TTGridReport;

implementation

uses main;

{$R *.dfm}

{ TTGridReport }

procedure TTGridReport.Preview(Grid: TDBGrid);
//процедура створення звіту з послідовним друком
//через компоненту QuickRep
var
  i, CurrentLeft, CurrentTop : integer;
  BMark: TBookmark;
begin
  GridRep.Dataset:=Grid.DataSource.Dataset;

  if not GridRep.Bands.HasColumnHeader then
    GridRep.Bands.HasColumnHeader:=true;

  if not GridRep.Bands.HasDetail then
    GridRep.Bands.HasDetail:=true;

  GridRep.Bands.ColumnHeaderBand.Height:=Abs(Grid.TitleFont.Height) + 10;
  GridRep.Bands.DetailBand.Height:=Abs(Grid.Font.Height) + 10;
  CurrentLeft := 12;
  CurrentTop := 6;

  {Запис, на якому користувач зупинився в DBGrid}
  BMark:=Grid.DataSource.Dataset.GetBookmark;
  {Заборона оновлення ґрида у процесі роботи звіта}
  Grid.DataSource.Dataset.DisableControls;
  try
    for i:=0 to Grid.FieldCount - 1 do
      begin
        if (CurrentLeft + Canvas.TextWidth(Grid.Columns[i].Title.Caption)) >
          (GridRep.Bands.ColumnHeaderBand.Width) then
          begin
            CurrentLeft := 12;
            CurrentTop := CurrentTop + Canvas.TextHeight('A') + 6;
            GridRep.Bands.ColumnHeaderBand.Height :=
              GridRep.Bands.ColumnHeaderBand.Height + (Canvas.TextHeight('A') + 10);
            GridRep.Bands.DetailBand.Height :=
              GridRep.Bands.DetailBand.Height + (Canvas.TextHeight('A') + 10);
          end;
        {Створимо заголовок звіту за допомогою QRLabels}
      end;
    end;
  end;
end;

```

```

with TQRLabel.Create(GridRep.Bands.ColumnHeaderBand) do
begin
    Parent := GridRep.Bands.ColumnHeaderBand;
    Color := GridRep.Bands.ColumnHeaderBand.Color;
    Left := CurrentLeft;
    Top := CurrentTop;
    Caption:=Grid.Columns[i].Title.Caption;
end;
{Створимо тіло звіту за допомогою QRDBText}
with TQRDbText.Create(GridRep.Bands.DetailBand) do
begin
    Parent := GridRep.Bands.DetailBand;
    Color := GridRep.Bands.DetailBand.Color;
    Left := CurrentLeft;
    Top := CurrentTop;
    Alignment:=Grid.Columns[i].Alignment;
    AutoSize:=false;
    AutoStretch:=true;
    Width:=Grid.Columns[i].Width;
    Dataset:=GridRep.Dataset;
    DataField:=Grid.Fields[i].FieldName;
    CurrentLeft:=CurrentLeft + (Grid.Columns[i].Width) + 15;
end;
end;

{Викликаємо режим перегляду перед друком з QuickRep}
GridRep.PreviewModal; {за допомогою PreviewModal}

finally
//зачистка
with Grid.DataSource.DataSet do
begin
    GotoBookmark(BMark);
    FreeBookmark(BMark);
    EnableControls;
end;
end;
end;

end.

```