

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

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи комплексних  
рішень для відеонагляду”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Діденко О.О.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
доктор технічних наук, професор  
\_\_\_\_\_ Смірнов О.А.  
« \_\_\_\_ » \_\_\_\_\_ 2024 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 2024 року

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

Діденку Олександр Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи комплексних рішень для відеонагляду

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

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи комплексних рішень для відеонагляду

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

- |  |  |
|--|--|
| <u>1. Призначення та область використання.</u>           | <u>6. Наукова новизна.</u>                                     |
| <u>2. Перегляд аналогічних існуючих систем.</u>          | <u>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</u> |
| <u>3. Опис і обґрунтування проектних рішень.</u>         | <u>8. Заходи з охорони праці та техніки безпеки.</u>           |
| <u>4. Етапи програмування системи.</u>                   | <u>9. Висновки.</u>  |
| <u>5. Впровадження системи в промислову експлуатацію</u> |  |

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

- |  |                 |
|--|-----------------|
| <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	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 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	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Діденко О.О. Дослідження та програмна реалізація системи комплексних рішень для відеонагляду. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комплексних рішень для відеонагляду.

Метою розробки є дослідження та програмна реалізація системи комплексних рішень для відеонагляду.

Об'єктом дослідження є процес комплексних рішень для відеонагляду.

Предметом дослідження є методи комплексних рішень для відеонагляду.

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

Результат роботи – програмна реалізація системи комплексних рішень для відеонагляду.

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

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

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

Програму розроблено в середовищі Python.

**Ключові слова:** комп'ютерна інженерія, відеонагляд

## ABSTRACT

**Didenko O.O. Research and software implementation of a system of complex solutions for video surveillance. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for a system of complex solutions for video surveillance.

The goal of development is research and software implementation of a system of complex solutions for video surveillance.

The object of the study is the process of complex solutions for video surveillance.

The subject of research is the methods of complex solutions for video surveillance.

Research methods are based on the methods of coding theory and the theory of computer networks, methods of mathematical statistics, and methods of software development.

The result of the work is the software implementation of a system of complex solutions for video surveillance.

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

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

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

The program was developed in the Python environment.

**Keywords:** computer engineering, video surveillance

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	12
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	12
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	23
2.3 Розгорнута постановка завдання .....	28
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	29
3.1 Опис функціонування системи .....	29
3.2 Розробка структурної схеми.....	32
3.3 Розробка функціональної схеми .....	35
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	40
4.2 Захист розробленого програмного забезпечення.....	54
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	57
6 НАУКОВА НОВИЗНА .....	62

						ВКРМ-123.24.0008.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Ціденко О.О.				Дослідження та програмна реалізація системи комплексних рішень для відеонагляду	Літ.	Аркуш	Аркушів
Перев.	Смірнов О.А.					М	1	90
Н.контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	63
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	63
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	64
7.3	Вибір методу оцінки вартості ПЗ .....	66
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	67
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	69
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	71
7.7	Визначення ключових факторів успіху конкретного проєкту.....	72
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	74
8.1	Вступ.....	74
8.2	Пожежна безпека .....	75
8.3	Пропозиції щодо підвищення працездатності ІТ-фахівців.....	77
8.4	Розробка заходів з умов поліпшення охорони праці .....	78
8.5	Розрахункова частина .....	79
9	ОСНОВНІ ВИСНОВКИ.....	82
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	84

КБПЗ-2024

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>



## ВСТУП

**Актуальність теми.** Рішення для відеоспостереження вже давно є критично важливим компонентом систем безпеки, надаючи підприємствам і організаціям засоби моніторингу та захисту своїх приміщень. Історично склалося так, що локальні системи відеоспостереження поклалися на мережеві відеореєстратори (NVR) і цифрові відеореєстратори (DVR) для запису та зберігання відеозапису. Однак із появою гіперконвергенції з'явилося більш ефективне та рентабельне рішення, яке зробило революцію у сфері відеоспостереження.

NVR та DVR відіграють значну роль у відеоспостереженні протягом багатьох років. NVR призначені для роботи з камерами IP (Internet Protocol), які знімають цифрове відео та передають його через мережу IP. Ці пристрої зберігають відеодані на жорстких дисках, забезпечуючи легкий пошук і відтворення. З іншого боку, відеореєстратори використовуються з аналоговими камерами та записують відео на внутрішні жорсткі диски.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи комплексних рішень для відеонагляду.

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

- Огляд існуючих систем комплексних рішень для відеонагляду.
- Дослідження системи комплексних рішень для відеонагляду.
- Програмна реалізація системи комплексних рішень для відеонагляду.

*Об'єктом дослідження є процес комплексних рішень для відеонагляду.*

*Предметом дослідження є методи комплексних рішень для відеонагляду.*

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

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Удосконалено метод комплексних рішень для відеонагляду.
- Розроблено вітчизняний продукт комплексних рішень для відеонагляду,

який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі комплексних рішень для відеонагляду.

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

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

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

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Незважаючи на те, що NVR та DVR є ефективними у забезпеченні локального відеоспостереження, вони мають певні обмеження. По-перше, ці системи вимагають спеціального апаратного забезпечення для кожної функції, включаючи відеозапис, зберігання та керування. Це може призвести до складного та дорогого налаштування, оскільки компаніям потрібно інвестувати в декілька пристроїв і керувати їх інтеграцією. Крім того, масштабування цих систем може бути складним завданням, що потребує встановлення додаткового обладнання, оскільки потреби в нагляді розширюються.

Гіперконвергенція для відеоспостереження пропонує більш ефективну та економічно вигідну альтернативу традиційним NVR та DVR. Гіперконвергентна інфраструктура (HCI) поєднує сховище, обчислювальну потужність і мережеві можливості в єдину інтегровану систему. Ця конвергенція усуває потребу в кількох пристроях і спрощує керування та масштабованість рішень відеоспостереження.

Використовуючи гіперконвергенцію, підприємства можуть досягти значної економії коштів. Консолідація апаратного забезпечення зменшує початкові інвестиції та витрати на технічне обслуговування, пов'язані з розгортанням окремих NVR та DVR. Крім того, гіперконвергентні рішення споживають менше фізичного простору, що призводить до зниження енергоспоживання та вимог до охолодження. Ці фактори сприяють зниженню загальної вартості володіння та покращенню рентабельності інвестицій.

Крім того, гіперконвергентні відеорішення пропонують підвищену гнучкість і масштабованість. Інтегрований характер HCI дозволяє легко розширювати його в міру зростання потреб у нагляді. До системи можна

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

безперешкодно додати додаткову пам'ять або обчислювальні ресурси, усуваючи потребу в значній зміні конфігурації або встановленні нового обладнання. Така масштабованість є особливо вигідною для підприємств із постійно зростаючими вимогами до відеоспостереження.

Гіперконвергенція також покращує ефективність відеоспостереження. Завдяки інтегрованому сховищу, обчислювальній потужності та мережі обробку відеоданих можна виконувати ближче до камер, зменшуючи затримку та перевантаження мережі. Це дозволяє здійснювати моніторинг у режимі реального часу та швидше отримувати відеоматеріали, підвищуючи обізнаність про ситуацію та можливості реагування на інциденти.

Крім того, гіперконвергентні відеорішення часто включають розширену аналітику та можливості машинного навчання. Ці технології забезпечують інтелектуальний аналіз відео, наприклад розпізнавання обличчя, виявлення об'єктів і аналіз поведінки. Використовуючи можливості штучного інтелекту та аналітику, компанії можуть отримувати цінну інформацію з даних відеоспостереження, підвищуючи безпеку, операційну ефективність і процеси прийняття рішень.

Хоча NVR та DVR історично служили основою рішень локального відеоспостереження, гіперконвергенція швидко трансформує галузь. Консолідуючи сховище, обчислювальну потужність і мережу в єдину інтегровану систему, гіперконвергентні відеорішення забезпечують більшу ефективність, масштабованість і економічну ефективність. Ці рішення оптимізують процеси розгортання та керування, зменшують витрати на апаратне забезпечення та забезпечують розширені можливості завдяки розширеній аналітиці та машинному навчанню. Оскільки компанії продовжують надавати пріоритет безпеці та шукати інноваційні рішення для відеоспостереження, гіперконвергенція зіграє ключову роль у формуванні майбутнього галузі.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

## 1.2 Область застосування

Перелік потенційних застосувань мережних відеотехнологій практично нескінченний, однак більша їхня частина відноситься до сфери забезпечення безпеки й дистанційного спостереження за людьми, територіями, майном і виробничою діяльністю. Нижче наведені деякі типові можливі застосування в ключових галузевих сегментах.

### Роздрібна торгівля

Системи мережного відеоспостереження можуть значно знизити рівень злочинства, підвищити безпеку персоналу й оптимізувати керування магазинами.

Важливою перевагою мережного відеоспостереження є можливість інтегрувати його із системою електронного спостереження за товарами (EAS) або із системою касових терміналів (POS), одержавши більше повну картину й історію дій, що приводять до втрат товарів. Така система дозволяє швидко виявляти як потенційні інциденти, так і можливі фіктивні тривоги. Системи мережного відеоспостереження відрізняються високим рівнем сумісності й найкоротших строків окупності. Мережні відеотехнології також можуть бути корисні для виявлення найбільш популярних зон магазину й для відстеження дій і купівельного поведіння клієнтів з метою оптимізації планування магазину й компонування вітрин. Воно також дозволяє виявити необхідність поповнення товару на полках або відкрити додаткові каси, якщо утворилися довгі черги.

### Транспорт

Мережне відеоспостереження може підвищити рівень особистої й загальної безпеки в аеропортах, на автомагістралях, на залізничних вокзалах і в інших транспортних системах, а також на самих транспортних засобах – в автобусах, поїздах, на круїзних судах. Мережні відеотехнології також можна застосовувати для контролю транспортних потоків з метою зниження заторів і підвищення ефективності.

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Для установок на транспорті найчастіше потрібні найкращі системи, з найвищою якістю зображення (яке може досягатися за рахунок підтримки камерами режиму прогресивного розгорнення), високою частотою кадрів і тривалим часом зберігання інформації.

Для деяких особливо складних ситуацій, наприклад, для відеоспостереження в автобусах і в поїздах, Axis пропонує мережні камери, здатні витримувати мінливі температури, високу вологість, запиленість, вібрації й вандалізм.

### **Утворення**

В освітніх установах, від дитячого садка до університету, мережні відеотехнології допомагають боротися з вандалізмом і забезпечувати безпека персоналу й що навчаються.

Якщо освітня установа вже володіє IT-інфраструктурою, мережне відеоспостереження може бути переважніше й економічно ефективніше аналогової системи, оскільки в багатьох випадках не потрібна нова кабельна проводка. Крім того, засоби керування подіями в системі мережного відеоспостереження дозволяють видавати сигнали тривоги й у реальному часі надавати операторам точну картину, що відбувається для прийняття рішень. Мережні відеотехнології також можна використовувати для дистанційного навчання – наприклад, для студентів, які не мають можливості відвідувати заняття очно.

### **Промисловість**

Мережні відеотехнології можна застосовувати для моніторингу й підвищення ефективності виробничих ліній, процесів і логістичних систем, а також для охорони складів і систем керування матеріально-технічними запасами.

Мережні відеотехнології дозволяють організовувати віртуальні наради й одержувати технічну підтримку в дистанційному режимі.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## **Безпечні міста**

Мережне відеоспостереження – один із самих корисних інструментів для боротьби зі злочинністю й захисту громадян. Воно відіграє двояку роль, допомагаючи виявляти правопорушення й відлякуючі потенційних порушників.

Поява бездротових мереж відкрило можливість розгортання мережного відеоспостереження в масштабі цілих міст. Дистанційне відеоспостереження за допомогою мережних відеокамер дозволяє органам поліції оперативно реагувати на злочини, що виявляються в реальному часі.

## **Урядові організації**

Устаткування для мережного відеоспостереження застосовується в різних суспільних будинках, від музеїв і офісів до бібліотек і в'язниць.

Камери, установлені на входах і виходах, здатні реєструвати вхідних і вихідних 24 години на добу. Це дозволяє запобігти вандалізму й підвищити рівень безпеки співробітників. У сполученні з інтелектуальними додатками для аналізу відео, що виконують, наприклад, підрахунок відвідувачів, системи мережного відеоспостереження дозволяють одержувати статистичну інформацію – наприклад, про кількість відвідувачів будинку.

## **Охорона здоров'я**

Мережні відеотехнології дозволяють будувати економічно ефективні високоякісні рішення для моніторингу й відеоспостереження пацієнтів, що підвищують безпеку й захищеність персоналу, пацієнтів, відвідувачів і майна.

Наприклад, уповноважені співробітники медичних установ з їхньою допомогою можуть переглядати в реальному часі відео з різних місць, виявляти події, що відбуваються, і надавати допомога в дистанційному режимі.

## **Банки й фінансові організації**

Мережні відеотехнології застосовуються в банківському секторі для забезпечення безпеки філій, центральних офісів і банкоматів.

Відеоспостереження використовується в банках уже багато років, і хоча більшість систем усе ще побудовано на аналогових технологіях, мережні

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

відеотехнології поступово впроваджуються й у цьому секторі, особливо в банках, де потрібна висока якість зображення й можливість ідентифікувати людей по відеозаписі.

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

КБПЗ\_2024

					VKPM-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

## 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

Новинка від Logitech – QuickCam Ultra Vision (рекомендована роздрібна ціна – 149,99 євро) – комплектується оновленим програмним забезпеченням Logitech QuickCam Software із простим і інтуїтивно зрозумілим інтерфейсом, за допомогою якого можна настроїти найпростіші охоронні функції. Крім того, даний пристрій має нову унікальну функцію, призначену для охоронного відеоспостереження: при відкритті кришки об'єктива камера починає передавати відеосигнал через будь-яку програму обміну миттєвими повідомленнями (Windows Live Messenger, Skype Video, Yahoo Messenger або AOL Instant Messenger).

Новинка від Logitech може похвастатися поліпшеною якістю картинки, що досягається за рахунок наявності в об'єктиві з маленьким коефіцієнтом світлопропускання ( $f = 1,6$ ) п'яти лінз великого розміру, частина з яких виконана з мінерального скла – це забезпечує чітку і яскраву відеозйомку. Крім того, фірмова технологія Logitech RightLight2 автоматично адаптує настроювання камери до умов освітлення, а Logitech RightSound – відповідає за чистий звук. Як ми вже згадували, QuickCam Ultra Vision оснащений 1,3-мегапіксельним сенсором, що підтримує формат High Definition (HD) і здатним захоплювати відео в широкоформатному режимі (зі співвідношенням сторін екрана 16:9).

Варто відзначити, що модернізовано й іншої моделі web-камер лінійки QuickCam. У них додані нові технології й функції, призначені для спрощення відеокommунікацій і поліпшення їхньої якості. Ключові продукти лінійки – web-

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

камери Logitech QuickCam Fusion, QuickCam Sphere і QuickCam Pro 5000 – тепер також підтримують RightLight2 і здатні записувати відео у форматі HD.

Безумовно, базова програма для web-камери не може зрівнятися із професійною охоронною системою, що включає ще з десятків різних датчиків. Але елементарні охоронно-попереджувальні функції й інтелектуальний аналіз картинки вона цілком здатна виконувати. При цьому помітимо, що цифрова система з віддаленою камерою невразлива, оскільки її працездатність цілком залежить від роботи персонального комп'ютера, що перебуває поза охоронюваною зоною.

Цікаво, що, по оцінках американських експертів, близько 70% потенційних правопорушень припиняються ще на стадії їхнього планування за рахунок демонстрації наявності охоронної системи. Дійсно, якщо злочинці помітять високий рівень охорони, то можуть відмовитися від первісних планів і перемкнутися на менш ретельно охоронюваний об'єкт.

### **GOTCHA! Multicam**

Що стосується універсальних програм для відеоспостереження від сторонніх виробників, те найбільш популярної з них (у всякому разі по числу згадувань в Інтернеті) залишається програма GOTCHA! Multicam.

Вона дуже проста й доступна, але при цьому дозволяє підключати кілька відеокамер і активізувати їх за бажанням користувача або по початку руху в кадрі, або у випадку припинення всякого руху. Детектор руху – це програмний модуль, призначений для виявлення, що переміщаються в поле зору web-камери об'єктів. Незважаючи на гадану простоту, у цієї програми є всі основні функції, властивим професійним охоронним системам. Результати роботи із вказівкою часу зйомки в кадрі програма GOTCHA! Multicam записує у файли у своєму форматі mv2, які потім програються убудованим плеєром.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13



Рисунок 2.1 – Інтерфейс користувача GOTCHA! Multicam

## Go1984

Програма Go1984 теж є універсальною системою для роботи з web-камерами й була розроблена для відеоспостереження й реєстрації даних. В утиліті go1984 є всі функції, що вимагаються для професійного відеоспостереження, у тому числі передача повідомлення (сигналу тривоги) на мобільний телефон.

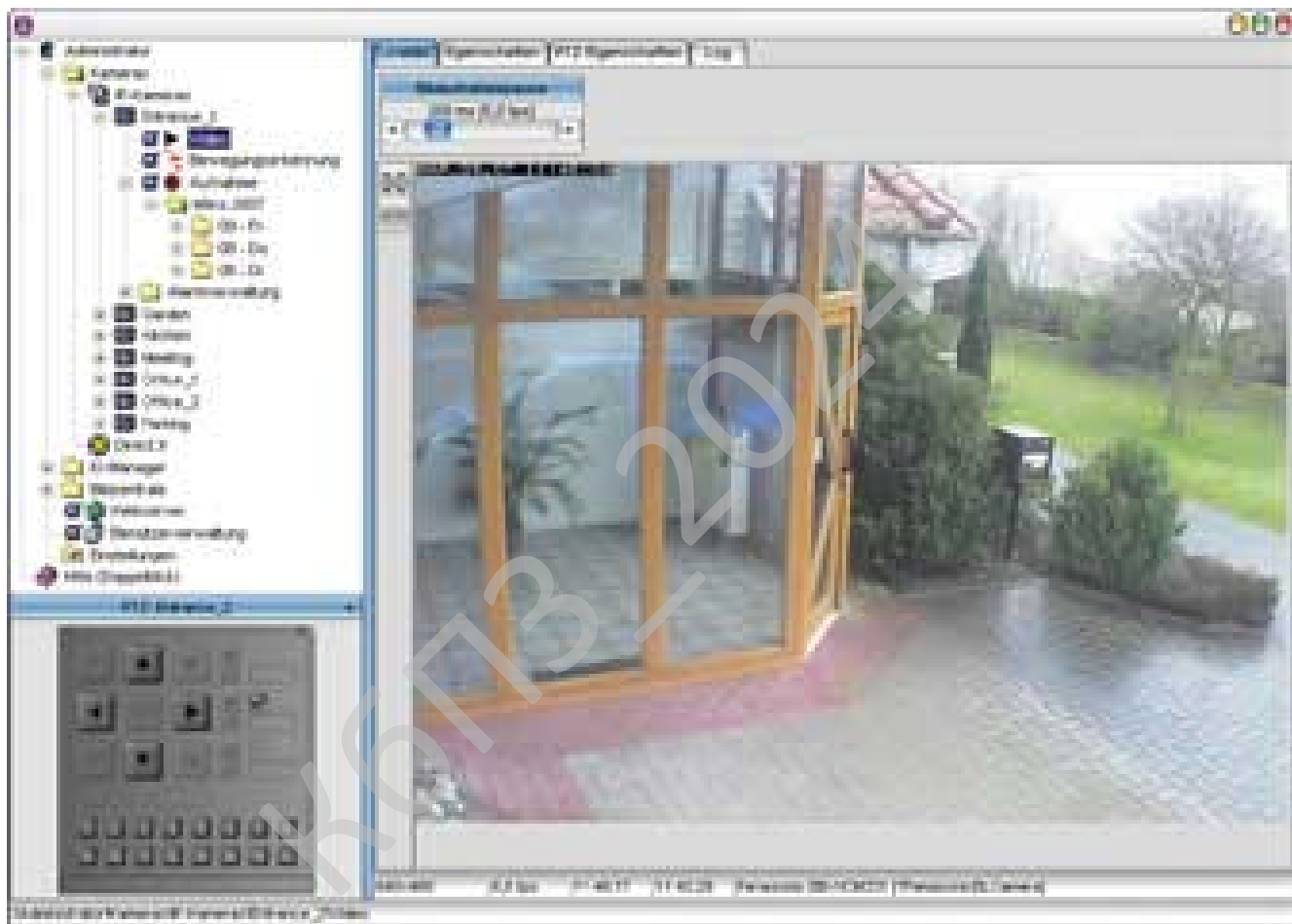


Рисунок 2.2 – Інтерфейс користувача Go1984

## Active WebCam

Програма ActiveWebCam знімає сигнал з web-камер і записує відео на жорсткий диск комп'ютера. Можна зберігати захоплений сигнал як у форматі програми, так і в будь-якому іншому кодеку, установленому в системі. Програма

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15



Інтернету можна організувати трансляцію, настроївши внутрішній сервер. Функція автоматичного завантаження буде із заданою періодичністю робити знімки й відправляти їх на ftp-сервер. Спеціальний Overlay Editor завідує додаванням текстових і графічних блоків, інтерактивних елементів. Різноманітні ефекти й настроювання дозволяють перевернути зображення, включити нічне бачення, накласти розмиття, зробити картинку чорно-білою або інвертувати кольори. І все це відбувається в реальному часі без яких-небудь затримок. Убудований планувальник завдань у випадку відсутності хазяїна на місці в заданий момент запише відео, включить або виключить детектор руху, запустить або зупинить web-сервер.

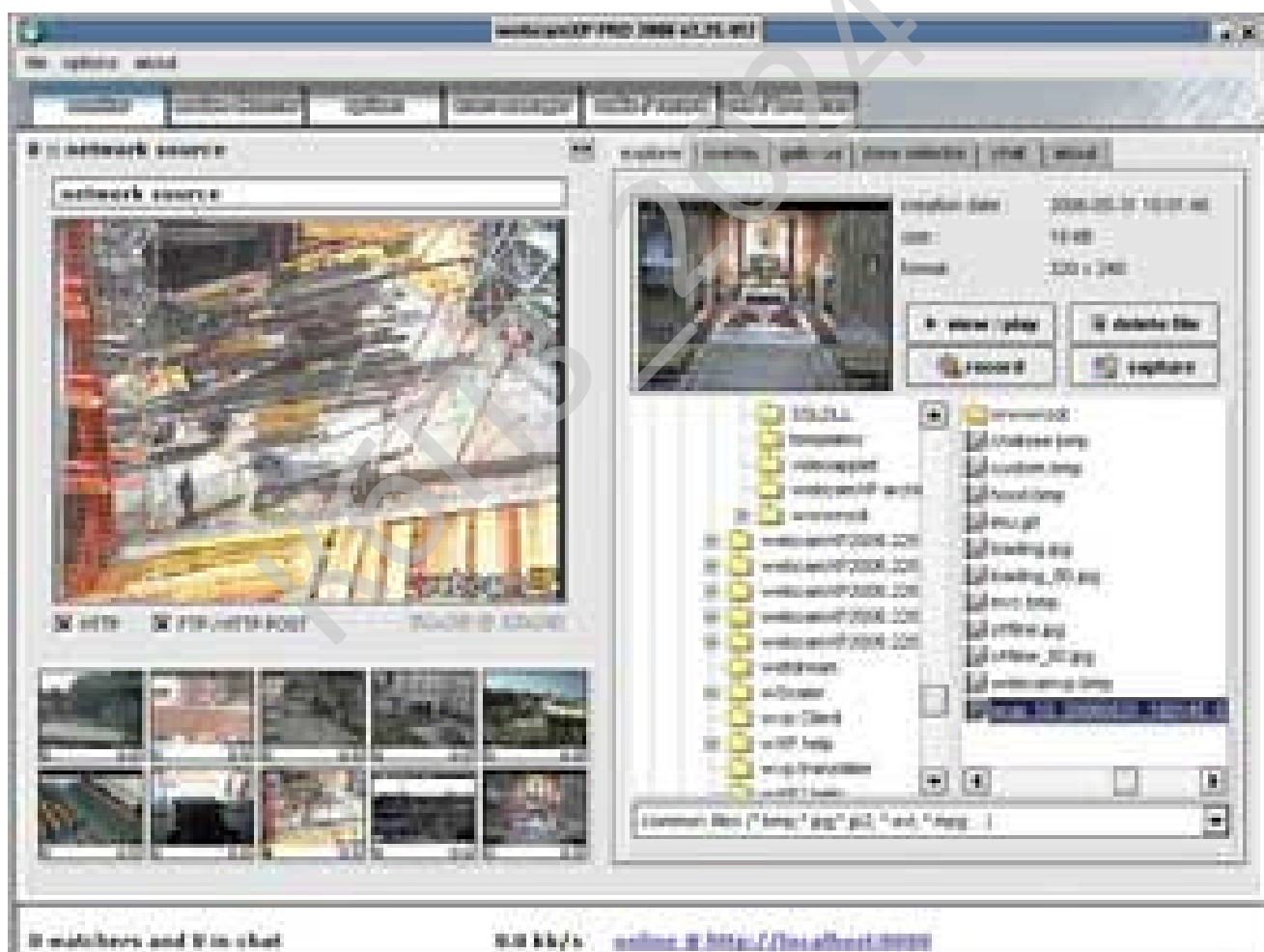


Рисунок 2.4 – Інтерфейс користувача WebcamXP



Для віддаленого відеоспостереження можна використовувати Internet Explorer через web-клієнта. Shareware-версія повністю функціональна, тобто в неї немає обмежень за часом і не відбувається ніяких змін у відеопотоках. Але безкоштовна версія не може працювати як Windows-сервіс, і в неї немає віддаленого доступу через TCP/IP до поточного або вже записаного відео.

### SupervisionCam

Програма SupervisionCam призначена для побудови системи спостереження на базі персонального комп'ютера із застосуванням будь-яких джерел відео.

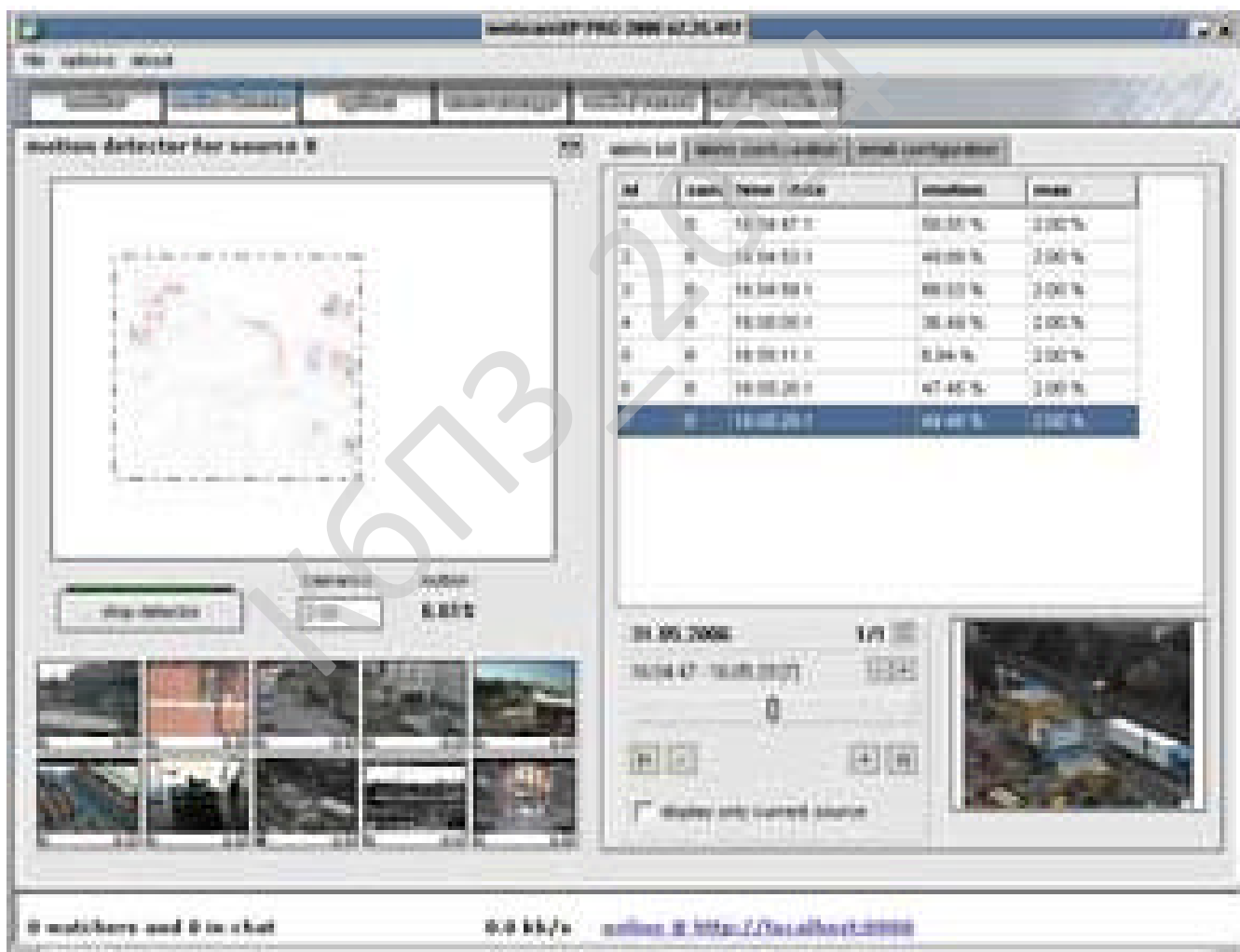


Рисунок 2.6 – Інтерфейс користувача SupervisionCam

Програма займається тим, що порівнює дані, що надходять, з відеокамери, підключеної до комп'ютера або розташованої де-небудь в Інтернеті, і починає захват відео або скріншотів, як тільки виявляє які-небудь зміни зображення. SupervisionCam також може зберігати знімки з екрана безупинно через певні інтервали часу. Дистанційне керування програмою й переглядом зображень можна організувати через звичайний web-браузер. Програма працює з будь-якими відеопристроями – головне, щоб для пристрою були встановлені драйвери й ОС Windows бачила їх як пристрою уведення відео. Заявлено підтримку ТВ-тюнерів, різних плат відеозахвата, USB-камер і інших цифрових камер, підключених по інтерфейсі IEEE-1394, а також мережних відеопристроїв.

### VirtGuard

Програма VirtGuard, або «Віртуальний сторож», призначена для створення системи відеоспостереження на базі недорогої моделі web-камери.



Рисунок 2.7 – Интерфейс користувача VirtGuard

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Як камера може виступати будь-яке джерело відео, підключений до карти відеозахвата або до ТБ-тюнера. Система може бути використана для охорони офісів, квартир і інших об'єктів власності з можливістю підключення сигналізації. Інтерфейс програми виконаний російською мовою.

### Кодос Відео

Програма «Кодосвідео» призначена для побудови охоронної системи відеоспостереження. Для домашнього використання цілком достатньо буде безкоштовної версії. Її відмінності від повної версії мінімальні й вони істотні тільки для серйозної охоронної системи.



Рисунок 2.8 – Інтерфейс користувача КодосВідео

Розроблювачами реалізовані всі можливості, якими володіють сучасні системи охорони й відеоспостереження. Програма має високу функціональність, постачена налаштовувемим детектором руху й веде докладну статистику за подіями, що відбуваються, причому на кожен подію може бути призначена певна дія. Є спеціальний діалог для перегляду статистики, де можна, наприклад, переглядати події за певний проміжок часу.

**Тепер давайте розглянемо деяких провідних постачальників VMS та їхні продукти:**

**Milestone Systems:** Milestone пропонує високо оцінену VMS під назвою "Milestone XProtect". Він надає широкий спектр функцій, включаючи керування відео, аналітику та інтеграцію зі сторонніми системами. XProtect підтримує як невеликі, так і корпоративні розгортання, що робить його придатним для різних програм.

**Genetec:** VMS Genetec під назвою «Центр безпеки» – це комплексна платформа, яка поєднує керування відео, контроль доступу та розпізнавання номерних знаків. Він пропонує розширену аналітику та інтегрується з різними сторонніми системами безпеки. Security Center відомий своєю масштабованістю та гнучкістю.

**Myrasys:** VMS Myrasys – це надійна система керування відео, яка пропонує широкий спектр розширених функцій. Він забезпечує моніторинг у реальному часі, відеозапис і відтворення, а також інтелектуальну аналітику для підвищення безпеки. Програмне забезпечення є масштабованим, зручним для користувача та підтримує інтеграцію з іншими системами безпеки для комплексного управління спостереженням.

**Identiv:** Identiv Velocity Vision – це багатофункціональна система керування відео. Він забезпечує повну функціональність для моніторингу в реальному часі, запису та відтворення відео. Система містить розширену аналітику для інтелектуального аналізу відео, підтримує інтеграцію з іншими системами безпеки та пропонує зручні інтерфейси для ефективного керування відеоспостереженням.

Це лише кілька прикладів провідних постачальників VMS у галузі. Кожен постачальник пропонує унікальний набір функцій і можливостей, що задовольняють різні вимоги клієнтів і розміри системи.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

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

### Швидкість виконання коду Python

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

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

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

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## Використання Python

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

Застосування мови програмування Python

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.
10. ...

## Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з'являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін. Процесор є найважливішим компонентом в комп'ютері. Одна з основних функцій процесора – це обробка даних згідно комп'ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

фрагментами даних. Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп'ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп'ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня. Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора. Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті. Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

### **Отримати**

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

### **Декодувати**

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

### **Виконати**

Останній крок циклу – виконати операцію. Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом. З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди. Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій. Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

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

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

1. Компілятор.
2. Інтерпретатор.

### **Завантаження Python**

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

### **Середовище програмування для Python**

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

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

Текстові редактори та інтегровані середовища програмування для Python:

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

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

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

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

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

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

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

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

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

### 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 комплексних рішень для відеонагляду.

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

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

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

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

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

д) розробити рекомендації по організаційних та методичних заходах, які 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.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		28

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Програмне забезпечення для відеоспостереження відіграє вирішальну роль у сучасних системах безпеки, надаючи розширені можливості для моніторингу, керування та аналізу відеозаписів. Це програмне забезпечення виступає основою для керування відео, дозволяючи підприємствам і організаціям розширювати свої можливості спостереження та ефективно захищати свої приміщення. Давайте заглибимося в те, що надає програмне забезпечення для відеоспостереження, і вивчимо деякі з провідних постачальників систем керування відео (VMS) та їхні продукти.

Програмне забезпечення для відеоспостереження служить центральним центром для керування відеоканалами з кількох камер. Це дозволяє користувачам переглядати прямі відеопотоки, отримувати доступ до записаного матеріалу та дистанційно керувати налаштуваннями камери. Ключові функції програмного забезпечення для відеоспостереження:

**Моніторинг у реальному часі:** програмне забезпечення для відеоспостереження дозволяє відстежувати в режимі реального часу канали камери через централізований інтерфейс. Користувачі можуть одночасно переглядати кілька потоків камер, перемикатися між камерами та спостерігати за певними областями чи інцидентами, коли вони розгортаються.

**Запис і відтворення відео:** програмне забезпечення VMS дозволяє записувати відео з підключених камер. Цей записаний матеріал можна отримати та переглянути пізніше для розслідування, збору доказів або аналізу ситуації. Розширені функції відтворення, такі як очищення відео, покадровий аналіз і синхронізоване відтворення під різними кутами камери, підвищують ефективність перегляду відео.

					VKPM-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

**Попередження та сповіщення:** програмне забезпечення для відеоспостереження може виявляти та запускати сповіщення про певні події, такі як виявлення руху, втручання в камеру або виявлення вторгнення. Сповіщення можна надсилати електронною поштою, SMS або через програмний інтерфейс, що дозволяє своєчасно реагувати на потенційні загрози безпеці.

**Централізоване керування:** програмне забезпечення VMS забезпечує централізовану платформу для керування всіма аспектами системи відеоспостереження. Користувачі можуть налаштовувати параметри камери, визначати розклади запису, встановлювати права доступу користувачів, а також керувати зберіганням і архівуванням відеоматеріалів.

**Інтеграція та масштабованість:** програмне забезпечення для відеоспостереження часто інтегрується з іншими системами безпеки, такими як контроль доступу, сигналізація та аналітичні платформи. Ця інтеграція дозволяє більш комплексно керувати безпекою та створює масштабоване рішення, яке може адаптуватися до мінливих потреб.

#### **Чому хмарне сховище для відеоспостереження?**

Інтеграція хмарного сховища та гіперконвергенції у ваш план відеоспостереження периферійних комп'ютерів може принести численні переваги та підвищити загальну ефективність і ефективність вашої системи. Ось кілька переконливих причин для розгляду цієї інтеграції:

**Масштабованість:** хмарне сховище забезпечує практично необмежену масштабованість, дозволяючи вам легко розширювати свою систему відеоспостереження в міру зростання ваших потреб. Завдяки хмарному сховищу ви можете динамічно регулювати обсяг пам'яті, не турбуючись про фізичні обмеження чи оновлення апаратного забезпечення. Гіперконвергенція доповнює цю масштабованість, забезпечуючи гнучку і легко розширювану інфраструктуру для задоволення зростаючих вимог вашого рішення для відеоспостереження.

**Економічна ефективність:** хмарне сховище усуває потребу у локальній інфраструктурі, такій як виділені сервери та масиви зберігання. Це значно

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30







Системи відеоспостереження висувають особливі вимоги до мережної інфраструктури: вона повинна бути надійною, захищеною, масштабованою й забезпечувати необхідна пропускну здатність.

Комутатори Allied Telesis підтримують QoS і PoE, передбачають використання медіаконвертерів мідь/оптика, мають засобу пошуку несправностей.

Деякі моделі випускаються в промисловому виконанні (із захистом IP30) і підтримують PoE+ з бюджетом до 370 Вт.

Все це дозволяє будувати мережі IP-відеоспостереження в різноманітній конфігурації.

Нові маршрутизатори Allied Telesis здатні обслуговувати до 8 черг QoS і до 4096 VLAN, мають до чотирьох відсіків SFP. Мережею, побудованої на базі комутаторів і маршрутизаторів Allied Telesis, можна управляти централізовано.

Кільцеві топології забезпечують відказостійкість інфраструктури. Цього року Allied Telesis представила десять новинок і до січня планує випустити ще стільки ж нових продуктів.

Важливим компонентом системи відеоспостереження є відеореєстратори (NVR).

У такій якості можуть використовуватися мережні системи зберігання (NAS) тайваньської компанії Synology настільного й стійкового виконання. Вони вміщують до 24 дискових накопичувачів (з модулями розширення – до 106 HDD) і здатні обслуговувати до тисячі користувачів одночасно. До NAS від Synology можна підключати до 90 камер і паралельно транслювати до 90 каналів відео.

Відеореєстратори Synology Surveillance Station підтримують понад 2700 моделей IP-камер більш ніж 40 виробників, забезпечують доступ до відео з мобільних пристроїв, «вирівнюють» зображення з камер Fish Eye, підтримують пошук у відеоархіві за подіями.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

### 3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Цифрові системи відеоспостереження виконують наступні функції:

– Моніторинг. Можливе підключення до 32-х камер, відображення всіх на одному екрані. Можливість поділу камер на два монітори. Використання режиму overlay дозволяє виводити зображення на екран монітора з аналоговою якістю. Можливість цифрового збільшення в 2, 4, 8 і 16 разів. Функція голосового супроводу подій (при підключених аудіоколонках).

– Запис. Стиск зображення здійснюється за алгоритмом Motion-Wavelet (розмір кадру залежить від дозволу, кольоровості камери, ступеня стиску й деталізації зображення) дозволяє досягти значень бітрейта в межах від 2 до 80 Кб. Режими запису: по детекції руху, по команді, постійно, передзапис і дозапис по кожній камері.

– Детекція руху. Відстеження наявності об'єктів, що рухаються. Основні переваги: висока перешкодозахищеність, емпіричні налаштування за розміром й контрастністю зображення, конфігурування області детекції руху незалежно для кожної камери, візуальне виділення контурів об'єктів, що рухаються, перерозподіл ресурсів для тривожної камери.

– Відтворення. Одна камера в один момент часу, перегляд відеоархіву одночасно із записом, індексація відеоархіву при запуску системи для швидкого пошуку, пошук у відеоархіві по даті й часу, відображення щільності запису за добу, покадрове програвання вперед та назад, збільшення швидкості програвання до 25 Fps, утиліта зовнішнього декодування й перегляду відеозаписів зроблених системою, можливість збереження шматків відеоархіву в avi, що б мати можливість перегляду на будь-якому комп'ютері.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35



Рисунок 3.2 – Функціональна схема системи

– Убудований WEB server. Надає можливість контролювати й управляти відеокамерами, переглядати відеоархів, а також здійснювати постановку й зняття з охорони з віддаленого комп'ютера через web-браузер.

– Аудіореєстрація. Дозволяє вводити в систему аналоговий сигнал із зовнішніх джерел: мікрофонів, телефонних каналів і інших. Аудіореєстрація може вестися по незалежних каналах, а може й синхронно. У випадку налаштування аудіореєстрації разом з відеозображенням, архів від цих камер записується спільно зі звуком. Аудіореєстрація необхідна у випадках недостаті тієї інформації, що надають відеокамери. Більше твердий контроль за підлеглими й персоналом. Найбільш часті сфери й місця застосування аудіоконтроля – кімнати переговорів і нарад, місця для паління, спостереження за будинком під час відсутності хазяїна, спостереження за обслуговуючим персоналом, (покоївкою, гувернанткою й іншою прислугою), запис телефонних переговорів секретаря. Для реалізації необхідного ефекту від аудіоконтролю й поліпшення



спеціалізований маніпулятор. Даний маніпулятор дозволяє працювати тільки із програмним забезпеченням.

Цифрова система відеоспостереження, що розроблена:

- забезпечує високу якість відтвореного відеозапису;
- високу швидкість доступу до відеоархіву;
- можливість цифрового збільшення й масштабування будь-якого кадру;
- миттєвий пошук і перегляд відеозапису по камері, даті й часу;
- можливість інтеграції з іншими комп'ютерними системами безпеки;
- легка й недорога трансляція відеоархівів по каналах зв'язку (Інтернет та ін.);
- можливість відправлення тривожних повідомлень по електронній пошті й SMS;
- можливість експорту відеоінформації на сумісні зовнішні носії.

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

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Діаграми потоків даних містять чотири типи елементів:

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

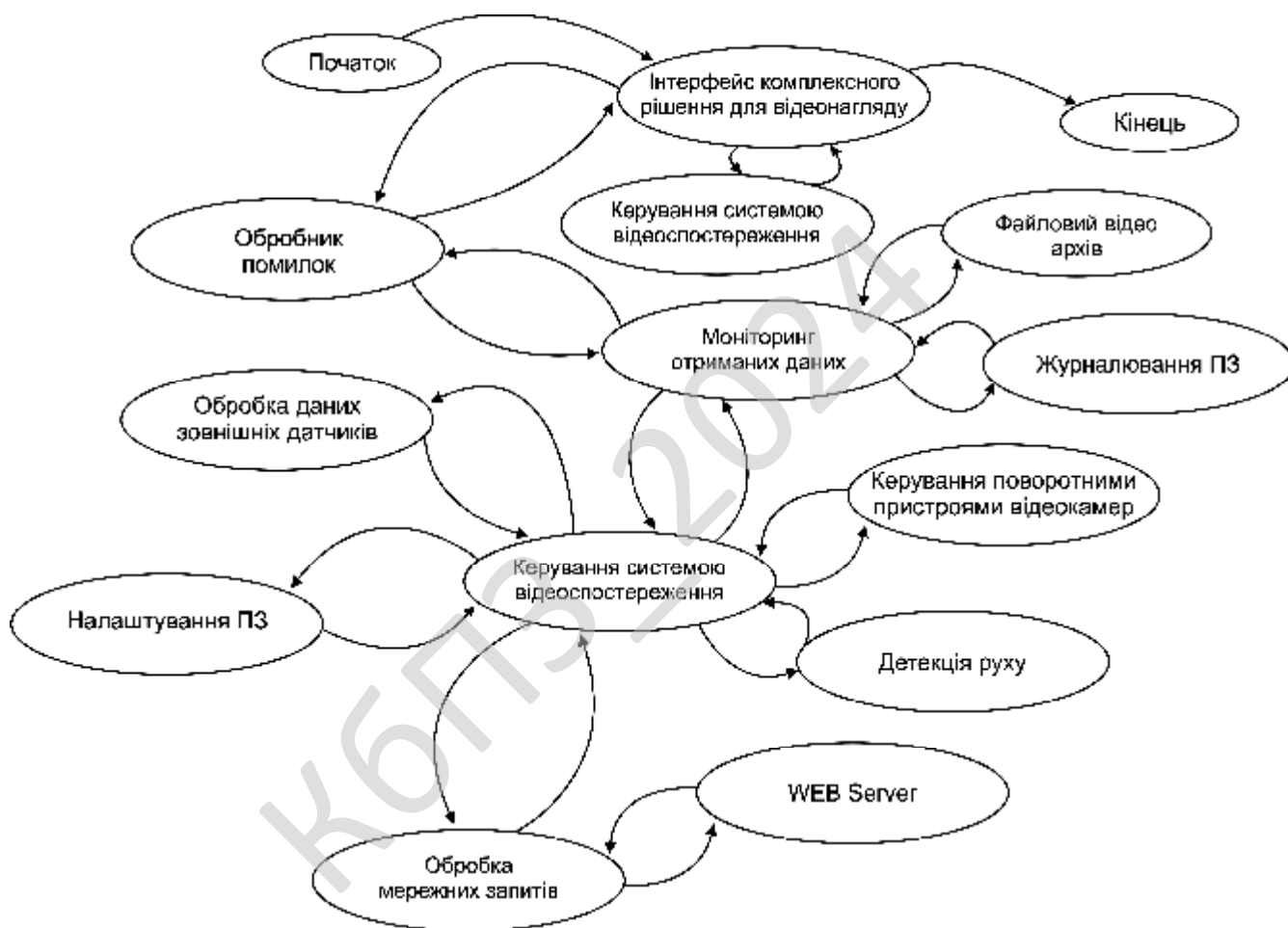


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

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

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

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

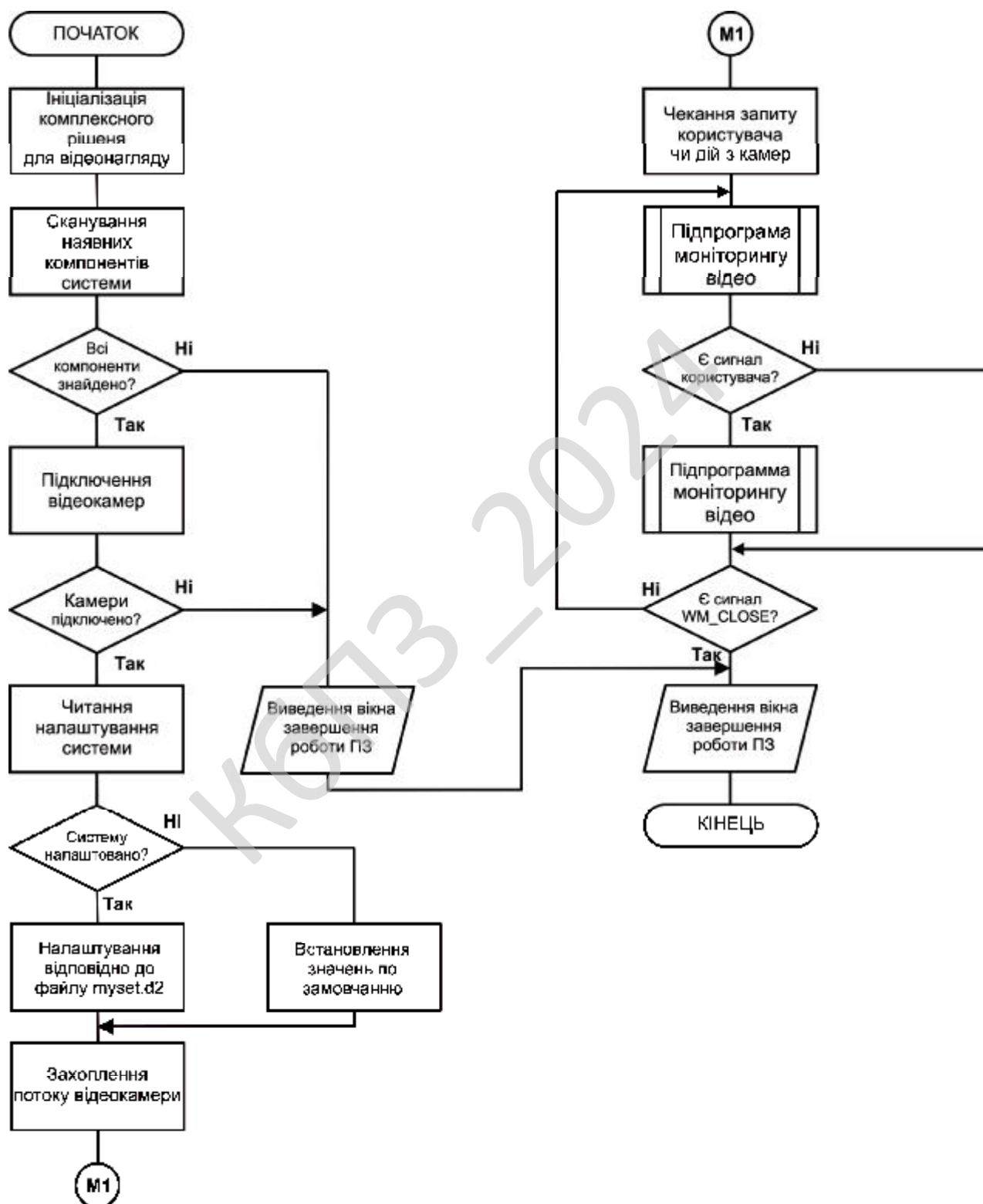


Рисунок 4.1 – Блок-схема основної програми

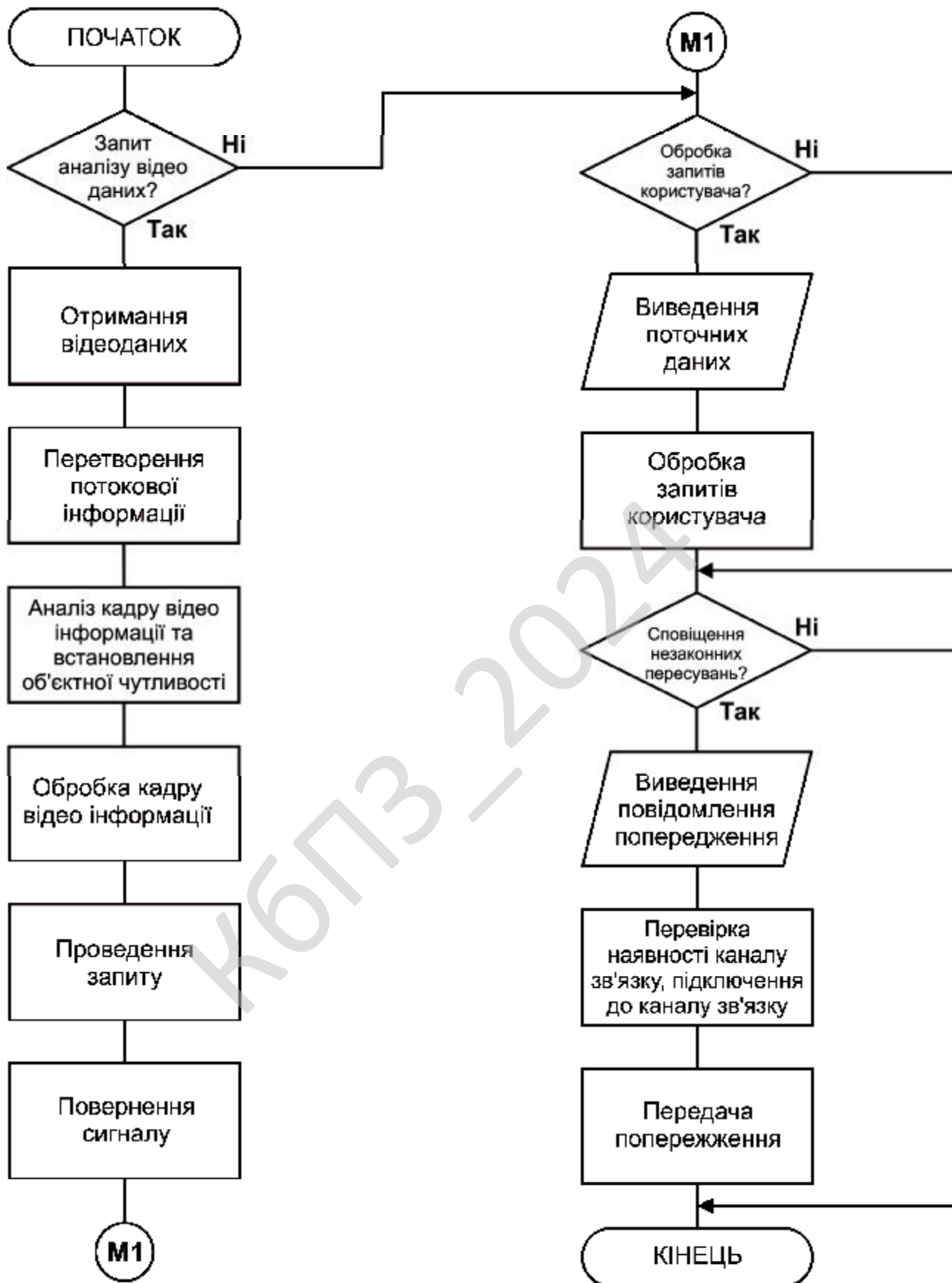


Рисунок 4.2 – Блок-схема роботи підпрограми

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

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

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

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

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

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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

Під час роботи над магістерською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Розглянемо виконані основні напрацювання у вигляді вихідного коду.

```
import cv2
import numpy as np
import datetime
# Клас для відеозахвату з камери
class VideoCapture:
    def __init__(self, source):
        self.source = source
        self.capture = cv2.VideoCapture(self.source)

    def get_frame(self):
        ret, frame = self.capture.read()
        if ret:
            return frame
        else:
            return None

# Клас для запису відео на диск
class VideoRecorder:
    def __init__(self, output_file, frame_size, fps=30):
        fourcc = cv2.VideoWriter_fourcc(*'XVID')
        self.writer = cv2.VideoWriter(output_file, fourcc, fps, frame_size)

    def write_frame(self, frame):
        self.writer.write(frame)

    def release(self):
        self.writer.release()

# Клас для обробки кадрів
class FrameProcessor:
    def __init__(self):
        pass

    def process_frame(self, frame):
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        edges = cv2.Canny(gray, 50, 150)
        return edges

# Клас для обробки подій та детекції руху
class MotionDetector:
    def __init__(self, sensitivity=500):
        self.sensitivity = sensitivity
        self.previous_frame = None
```

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

def detect_motion(self, frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21, 21), 0)
    if self.previous_frame is None:
        self.previous_frame = gray
        return False
    frame_delta = cv2.absdiff(self.previous_frame, gray)
    thresh = cv2.threshold(frame_delta, 25, 255, cv2.THRESH_BINARY)[1]
    thresh = cv2.dilate(thresh, None, iterations=2)
    motion_area = np.sum(thresh)

    self.previous_frame = gray

    return motion_area > self.sensitivity
# Основна система для керування відеонаглядом
class VideoSurveillanceSystem:
    def __init__(self, camera_source, output_file):
        self.camera = VideoCapture(camera_source)
        self.recorder = VideoRecorder(output_file, (640, 480))
        self.processor = FrameProcessor()
        self.motion_detector = MotionDetector()
    def run(self):
        while True:
            frame = self.camera.get_frame()
            if frame is None:
                break
            processed_frame = self.processor.process_frame(frame)
            motion_detected = self.motion_detector.detect_motion(frame)

            if motion_detected:
                timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
                cv2.putText(frame, f"Motion detected {timestamp}", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
                self.recorder.write_frame(frame)
                cv2.imshow("Video Surveillance", frame)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
            self.recorder.release()
            self.camera.capture.release()
            cv2.destroyAllWindows()
# Приклад використання

```

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

```
if __name__ == "__main__":
    system = VideoSurveillanceSystem(0, 'output.avi')
    system.run()
```

Представлені модулі:

- cv2 для захоплення відеопотоку, обробки відео та кадрів.
- numpy для маніпуляцій з даними кадрів.
- datetime для додавання міток часу до кадрів.

Розроблена архітектура:

- VideoCapture: захоплює кадри з камери.
- VideoRecorder записує відео на диск, використовуючи кодек XVID.
- FrameProcessor обробляє кадри для виділення важливих елементів (у цьому прикладі використовується Canny для виділення країв).
- MotionDetector: виявляє рух за допомогою порівняння кадрів.
- VideoSurveillanceSystem інтегрує всі елементи системи і забезпечує основну логіку для запису відео при виявленні руху.

Розглянемо у зжатому форматі проектні запропоновані рішення та розрахунки:

- Для збереження відео обрано формат XVID через його компромісію між якістю та обсягом даних.
- Детекція руху виконується на основі порогу, що контролює чутливість системи.
- Використання сірого та розмитого зображення у процесі детекції руху зменшує вплив шуму та покращує продуктивність

При розгляді розробленого ПЗ можна побачити що програма розбита на декілька важливих блоків, таких як:

- Блок ініціалізації динамічних бібліотек користувача.
- Блок підключення додаткових модулів.
- Блок читання файлів налаштування та керування.
- Блок захоплення потоку.
- Блок очікування дій користувача.

- Блок аналізу даних.
- Блок обробки сигналу.

Обробка відео потоку і виведення на екран в середовищі Windows при застосуванні основних методів виведення відео інформації на екран лінійки операційних систем Windows. Виникає гостра проблема в швидкості обробки потокового кадру, що приводить до уповільнення процесу висновку інформації на екран. Як відомо для перегляду відеопотоку необхідно не менше 24 кадрів в секунду.

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

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

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

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

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

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

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

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

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

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

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

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

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

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

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

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

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

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

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51





діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

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

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

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

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

## 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-2, який шифрує дані 256-бітними блоками з використанням 512-бітного ключа. Допускається використання ключів менших розмірів (не менш 128 біт), які доповнюються бітовими нулями до 512 біт.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Шифруємий блок даних ділиться на 8 фрагментів по 32 біта (які позначені буквами  $A...H$ ). Алгоритм виконує 64 раунду перетворень, у кожному з яких дані фрагменти обробляються в такий спосіб:

$$T = H_i + S_1(E_i) + Ch(E_i, F_i, G_i) + M_i + K_i,$$

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

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

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

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

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

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

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

$$A_{i+1} = T + S_0(A_i) + Maj(A_i, B_i, C_i).$$

де  $T$  – тимчасова змінна.

Використовувані функції визначені в такий спосіб:

$$S_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22),$$

$$S_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25),$$

$$Ch(x, y, z) = (x \& y) \oplus (x' \& z),$$

$$Maj(x, y, z) = (x \& y) \oplus (x \& z) \oplus (y \& z),$$

де  $\ggg$  – операція побітового циклічного зрушення вправо.

Константи, що модифікують,  $M_i$  ( $i = 0...63$ ) наведено нижче (одна за одною від  $M_0$  до  $M_{63}$ ):

428A2F98	71374491	B5C0FBCF	E9B5DBA5
3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967

27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

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

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

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

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

де функції  $O_0$  і  $O_1$  визначені так:

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

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

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

Раунди розшифрування алгоритму виконуються у зворотній послідовності:

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

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

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

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

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

$$D_i = E_{i+1} + T' + 1,$$

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

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

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



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

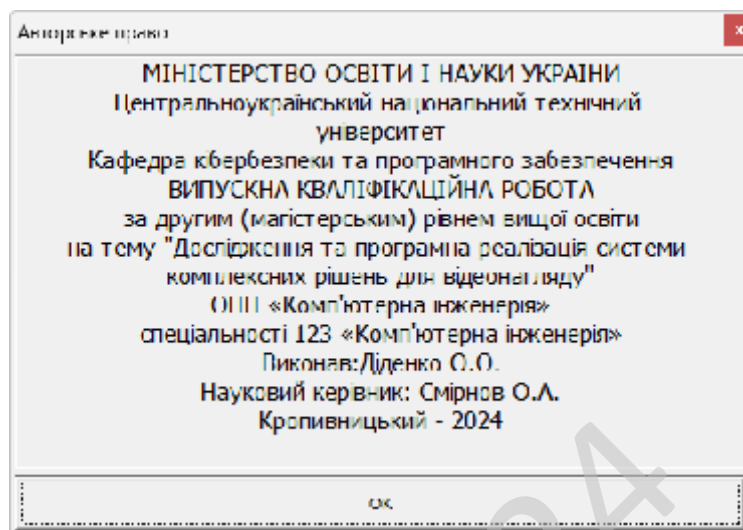


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

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

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58



– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

Обрано умови розповсюдження – commercial software.

Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційних програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації, надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм. Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

КБПЗ - 2024

					VKPM-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комплексних рішень для відеонагляду.

*Метою розробки є дослідження та програмна реалізація системи комплексних рішень для відеонагляду.*

*Об'єктом дослідження є процес комплексних рішень для відеонагляду.*

*Предметом дослідження є методи комплексних рішень для відеонагляду.*

*Методи дослідження базуються на методах теорії кодування та 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 комплексних рішень для відеонагляду.
- Розроблено вітчизняний продукт комплексних рішень для відеонагляду, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи комплексних рішень для відеонагляду можуть бути цікаві різним групам користувачів і організаціям(рисунок 7.1).

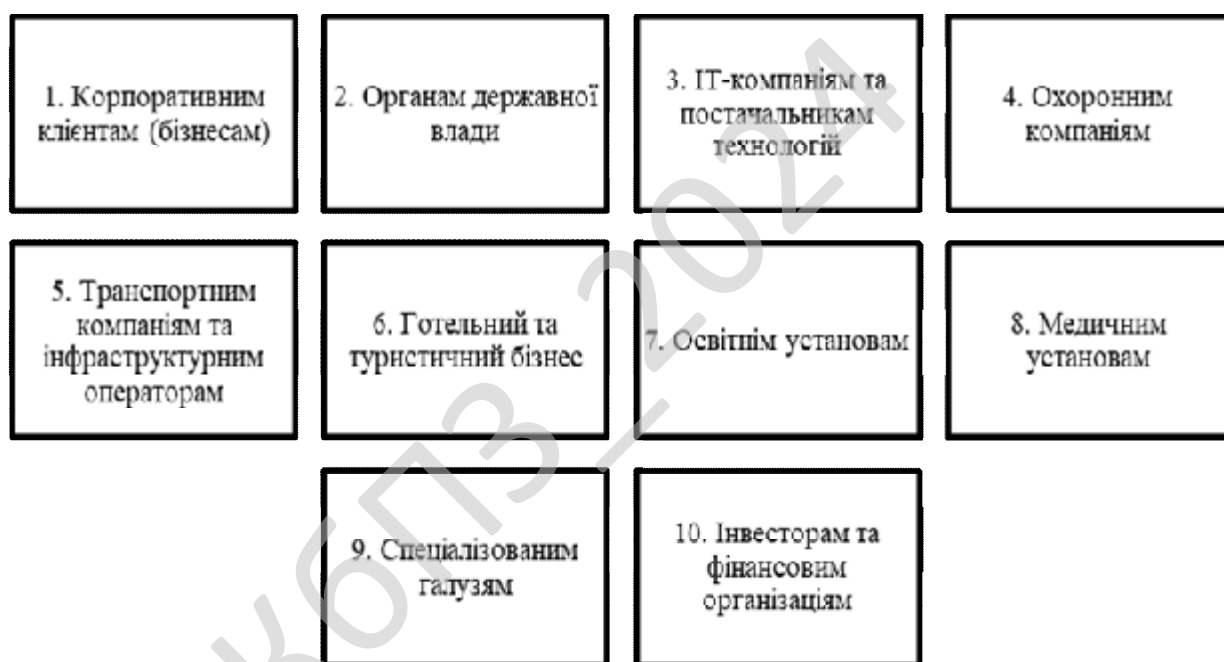


Рисунок 7.1 – Цільова аудиторія

Таким чином, результати дослідження та програмної реалізації системи комплексних рішень для відеонагляду можуть бути корисними для широкого спектра організацій, що займаються забезпеченням безпеки, управлінням інфраструктурою, обробкою даних та впровадженням технологічних інновацій.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи комплексних рішень для відеонагляду за допомогою методів експертних оцінок може здійснюватися шляхом збору думок експертів у різних галузях, таких як безпека, ІТ-технології, управління бізнес-процесами. Для цього можна використовувати різні методи, зокрема методи оцінки з використанням вагових коефіцієнтів, методи парних порівнянь, або ж методи оцінки ризиків.

Оцінити привабливість програмної реалізації системи комплексних рішень для відео нагляду необхідна з точки зору її комерційної вигоди, технологічної ефективності та придатності для різних типів організацій.

Першим кроком є визначення критеріїв, за якими оцінюється привабливість проєкту (рисунок 7.2).

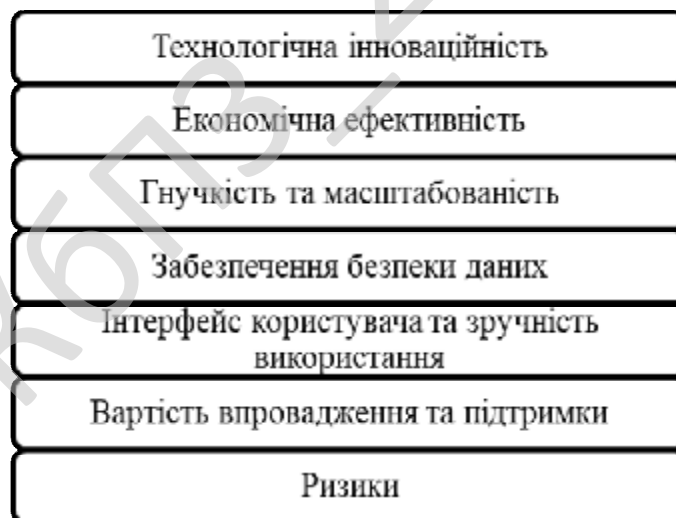


Рисунок 7.2 – Критерії експертної оцінки

Для оцінки привабливості проєкту необхідно залучити групу експертів: ІТ-фахівці (для оцінки технологічних аспектів); бізнес-аналітики (для оцінки



(вага: 0.1), інтерфейс користувача: 7/10 (вага: 0.1), вартість впровадження та підтримки: 6/10 (вага: 0.1).

$$\text{Оцінка привабливості} = (8 \times 0.3) + (7 \times 0.2) + (9 \times 0.2) + (8 \times 0.1) + (7 \times 0.1) + (6 \times 0.1) = 7.$$

8. Це свідчить про те, що проєкт має досить високий рівень привабливості з точки зору експертної оцінки.

Метод експертних оцінок дозволяє отримати об'єктивну та обґрунтовану оцінку привабливості програмної реалізації системи комплексних рішень для відеонагляду. Важливо, щоб оцінки були зібрані від досвідчених та різнопрофільних експертів для забезпечення широкого та збалансованого підходу.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи комплексних рішень для відеонагляду можна використовувати кілька методів, залежно від етапу проєкту та доступних даних. Найбільш підходящими методами є:

1. Метод аналогій. Цей метод передбачає порівняння з іншими схожими проєктами, які вже були реалізовані або оцінені на ринку. Метод є простим і ефективним, коли є достатньо інформації про схожі проєкти.

2. Метод вартості за етапами (поетапна оцінка) передбачає розбиття проєкту на основні етапи (аналіз вимог, розробка, тестування, впровадження, підтримка) і оцінку вартості кожного етапу окремо.

3. Метод оцінки вартості за допомогою експертних оцінок полягає в тому, щоб запросити експертів у сфері розробки програмного забезпечення та відеонагляду для надання їхніх оцінок вартості на основі досвіду та знань. Такі експерти можуть використовувати свої знання про ринок, технології та потенційні ризики для точнішої оцінки вартості проєкту.

4. Метод вартості за обсягом функціоналу використовується для визначення вартості програмної реалізації на основі кількості функціональних одиниць, які необхідно розробити або реалізувати в системі. Цей метод широко

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

застосовується у сфері програмної інженерії та дозволяє оцінити вартість на основі кількості та складності функцій, які необхідно реалізувати.

Підсумовуючи варто зазначити, що залежно від ситуації варто обрати той чи інший спосіб. Якщо проєкт вже реалізувався в аналогічних умовах, варто використовувати метод аналогій. Якщо проєкт складний і має кілька етапів, доцільно використовувати метод вартості за етапами. Для детальної та точнішої оцінки можна комбінувати методи функціональних точок та експертних оцінок.

#### **7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості**

Оцінка економічної ефективності від впровадження системи комплексних рішень для відео нагляду включає витрати на впровадження та очікувані заощадження або доходи внаслідок використання системи. Зведені дані наведено на рисунку 7.1.

Таблиця 7.1 – Основні показники впровадження проєкту

1. інвестиційні витрати на впровадження системи відеонагляду: вартість обладнання та інфраструктури (камери, сервери, мережеве обладнання): камери (100 одиниць) – 50 000 грн сервери та сховище даних – 30 000 грн мережеве обладнання та програмне забезпечення – 20 000 грн інсталяція та налаштування системи – 15 000 грн навчання персоналу – 5 000 грн загальні інвестиційні витрати: 120 000 грн
2. річні експлуатаційні витрати: електроенергія та технічне обслуговування: електроенергія (робота системи відеонагляду) – 5 000 грн/рік технічне обслуговування та підтримка – 10 000 грн/рік оновлення програмного забезпечення – 3 000 грн/рік загальні річні експлуатаційні витрати: 18 000 грн

Продовження таблиці 7.1

3. очікувані економічні вигоди від впровадження системи відеонагляду:

зниження витрат на безпеку:

зниження витрат на найм охоронців:

до впровадження: 5 охоронців, кожен отримує 30 000 грн/рік

витрати на охорону до впровадження: 150 000 грн/рік

після впровадження відеонагляду можна зменшити кількість охоронців до 2-х

(основний контроль, моніторинг та реагування через систему відеонагляду):

витрати на охорону після впровадження: 60 000 грн/рік

економія на витратах на охорону:  $150\,000 - 60\,000 = 90\,000$  грн/рік

зниження витрат на страхування:

зниження рівня злочинності та покращення безпеки може знизити ставки на страхування:

до впровадження страхові внески: 20 000 грн/рік

після впровадження знижка на страхування завдяки покращеній безпеці: 5 000 грн/рік

економія на страхуванні: 5 000 грн/рік

підвищення ефективності:

оптимізація процесів і підвищення продуктивності працівників, оскільки система відеонагляду дозволяє автоматизувати багато функцій, зменшуючи потребу в ручному контролі:

очікувана економія на праці: 15 000 грн/рік

загальні вигоди від впровадження системи:

зниження витрат на охорону: 90 000 грн/рік

економія на страхуванні: 5 000 грн/рік

підвищення ефективності: 15 000 грн/рік

загальні вигоди:  $90\,000 + 5\,000 + 15\,000 = 110\,000$  грн/рік

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

## Продовження таблиці 7.1

### 4. оцінка економічної ефективності (roi):

ROI (повернення на інвестиції) можна розрахувати за допомогою наступної формули:

$$\text{ROI} = \text{чистий виграш від проєкту} \div \text{інвестиційні витрати} \times 100$$

$$\text{чистий виграш} = \text{загальні вигоди} - \text{річні експлуатаційні витрати} = 110\,000 - 18\,000 = 92\,000 \text{ грн/рік}$$

$$\text{ROI} = 92\,000 \div 120\,000 \times 100 = 76.67\%$$

### 5. термін окупності:

термін окупності (payback period) визначається як час, необхідний для повернення інвестицій на основі щорічних економічних вигод.

$$\text{термін окупності} = \text{інвестиційні витрати} \div \text{щорічні вигоди} = 120\,000 \div 92\,000 \approx 1.30 \text{ роки}$$

річна економічна вигода: 110 000 грн

ROI: 76.67%

термін окупності: приблизно 1.3 роки

Цей приклад демонструє високу економічну ефективність від впровадження системи комплексних рішень для відеонагляду. Проєкт окупається менш ніж за два роки завдяки зниженню витрат на охорону, страхування та підвищенню ефективності роботи підприємства.

## 7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проєкту програмної реалізації системи комплексних рішень для відеонагляду може включати кілька ключових етапів, орієнтованих на залучення потенційних клієнтів, підвищення популярності системи та її впровадження в різні сегменти ринку (рисунок 7.3).

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

<p><b>Дослідження потреб ринку:</b></p> <ul style="list-style-type: none"> <li>• Оцінити попит на відеонаглядіві системі серед різних сегментів (бізнеси, організації, державні установи, громадські об'єкти).</li> <li>• Визначити проблеми, які можуть бути вирішені за допомогою системи відеонагляду.</li> </ul>	<p><b>Сегментація клієнтів:</b></p> <ul style="list-style-type: none"> <li>• Визначити потенційних замовників (наприклад, торгові центри, підприємства, розважальні комплекси, школи, банки).</li> <li>• Створити персоналізовані пропозиції для кожного сегмента.</li> </ul>	<p><b>Визначення ключових переваг системи:</b></p> <ul style="list-style-type: none"> <li>• Висока якість відео, зручність інтеграції, можливість моніторингу в реальному часі, покращена безпека завдяки AI та аналізу даних.</li> </ul>
<p><b>Формулювання УТП:</b></p> <ul style="list-style-type: none"> <li>• Створення короткої і зрозумілої пропозиції, яка підкреслює ключові переваги вашої системи, наприклад, "Інтелектуальна система відеонагляду, що автоматизує моніторинг і забезпечує безпеку 24/7".</li> </ul>	<p><b>Створення бренду та брендової стратегії:</b></p> <ul style="list-style-type: none"> <li>• Розробка бренду, який асоціюється з інноваціями, надійністю та високою ефективністю.</li> <li>• Позичування бренду на ринку через інтерв'ю, статті, відео скенерів.</li> </ul>	<p><b>Цифровий маркетинг:</b></p> <ul style="list-style-type: none"> <li>• <b>SEO та контент-маркетинг:</b> Розробка контенту для залучення органічного трафіку через блог, статті, інтерв'ю, кейс-стаді.</li> <li>• <b>SMM (Social Media Marketing).</b></li> <li>• <b>Email-маркетинг</b></li> </ul>
<p><b>Реклама:</b></p> <ul style="list-style-type: none"> <li>• <b>Контекстна реклама</b> через Google Ads та рекламу в соціальних мережах для цільової аудиторії.</li> <li>• <b>Реклама в професійних виданнях та на конференціях:</b> Публікації у спеціалізованих журналах та участь у виставках.</li> </ul>	<p><b>Формування команди продажів:</b></p> <ul style="list-style-type: none"> <li>• Підбір кваліфікованих фахівців, які будуть працювати безпосередньо з потенційними клієнтами, демонструвати переваги системи та надавати консультації.</li> </ul>	<p><b>Побудова довгострокових відносин з клієнтами:</b></p> <ul style="list-style-type: none"> <li>• Забезпечення високоякісної підтримки клієнтів, що дозволить налагодити стабільні взаємини.</li> <li>• Організація гостових запусків або демонстраційних версій системи.</li> </ul>
<p><b>Прямий продаж та партнерські програми:</b></p> <ul style="list-style-type: none"> <li>• Встановлення партнерських відносин з компаніями, які можуть бути потенційними реселерами або інтеграторами вашої системи.</li> <li>• Надання партнерам спеціальних умов або бонусів.</li> </ul>	<p><b>Вибір каналів збуту:</b></p> <ul style="list-style-type: none"> <li>• Продаж через веб-сайт, платформи для підприємств, партнерські організації, інтегратори.</li> </ul>	<p><b>Залучення партнерів для інтеграції системи відеонагляду:</b></p> <ul style="list-style-type: none"> <li>• Створення партнерських програм для компаній, які займаються автоматизацією, технічними рішеннями для підприємств, або IT-компаній.</li> </ul>
<p><b>Інтеграція з іншими системами:</b></p> <ul style="list-style-type: none"> <li>• Система відеонагляду має бути сумісною з іншими IT-рішеннями, наприклад, системами безпеки або автоматизації.</li> </ul>	<p><b>Організація вебінарів та демонстрацій:</b></p> <ul style="list-style-type: none"> <li>• Проведення вебінарів, семінарів або живих демонстрацій на реальних прикладах використання системи відеонагляду.</li> </ul>	<p><b>Публікація успішних кейсів:</b></p> <ul style="list-style-type: none"> <li>• Публікація в Інтернеті результатів успішних реалізованих проєктів, з позитивними відгуками клієнтів.</li> </ul>
<p><b>Збір та аналіз відгуків користувачів:</b></p> <ul style="list-style-type: none"> <li>• Визначення слабких місць та вдосконалення продукту на основі зворотного зв'язку.</li> </ul>	<p><b>Аналіз ефективності маркетингових кампаній:</b></p> <ul style="list-style-type: none"> <li>• Визначення результативності каналів просування (аналітика відвідуваності, ROI, конверсії).</li> </ul>	<p><b>Коригування стратегії на основі результатів:</b></p> <ul style="list-style-type: none"> <li>• Оцінка ефективності різних методів і оптимізація їх для досягнення більш високих результатів.</li> </ul>
<p><b>Підвищення лояльності клієнтів:</b></p> <ul style="list-style-type: none"> <li>• Впровадження програм лояльності, оновлення продуктів, активна комунікація з клієнтами для підтримки довгострокових відносин.</li> </ul>	<p><b>Пропонування фінансових рішень (лізинг, розстрочка, кредитування):</b></p> <ul style="list-style-type: none"> <li>• Надання клієнтам можливості придбати систему через фінансові інструменти, що знижують початкові витрати.</li> </ul>	<p><b>Підтримка на етапах реалізації та післяпродажне обслуговування:</b></p> <ul style="list-style-type: none"> <li>• Пропонування пакету технічної підтримки та обслуговування після впровадження системи для забезпечення стабільної роботи.</li> </ul>

Рисунок 7.3 – Алгоритм просування проєкту

Алгоритм просування проекту програмної реалізації системи комплексних рішень для відеонагляду має бути орієнтований на детальний аналіз ринку, чітке позиціонування продукту, активну маркетингову кампанію, налагодження каналів збуту, а також інтеграцію з іншими рішеннями для забезпечення максимального результату. Це дасть змогу збудувати ефективну стратегію просування та збільшити попит на систему відеонагляду.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи комплексних рішень для відеонагляду є важливим етапом для забезпечення широкого доступу до продукту та його успішного впровадження на ринку (рисунок 7.4).

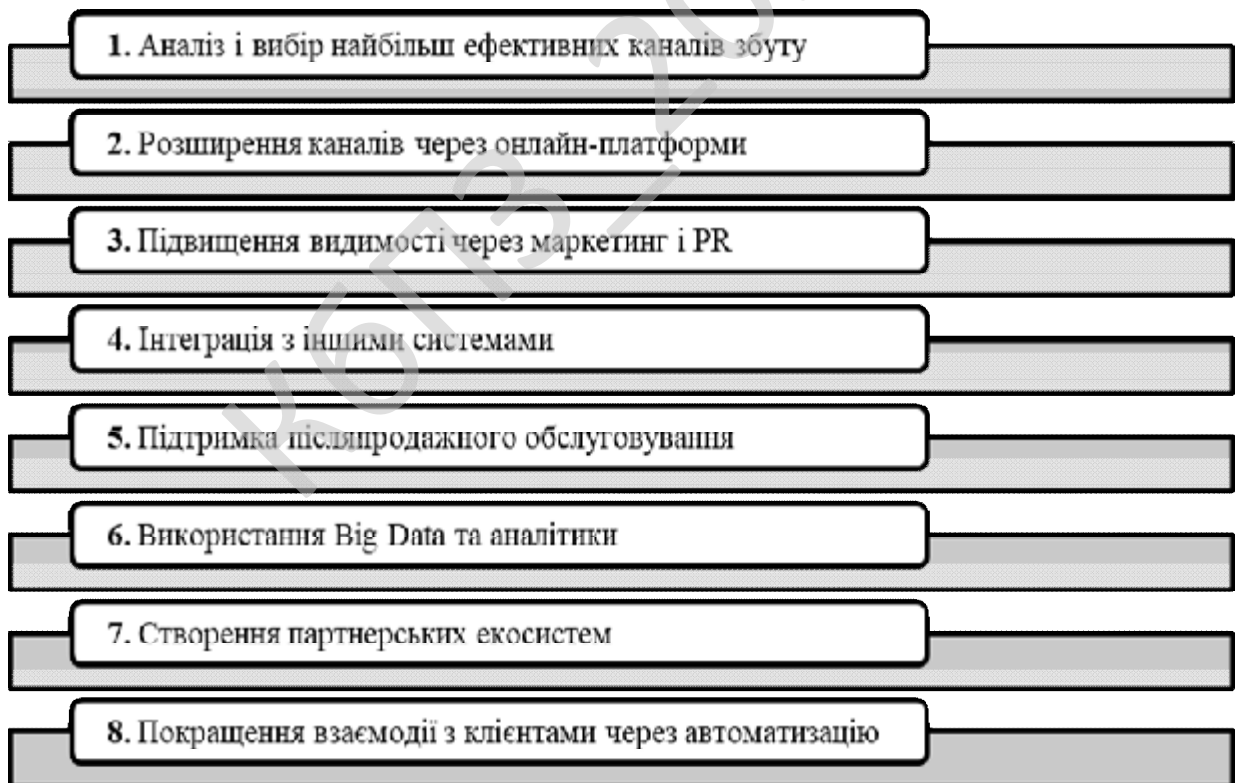


Рисунок 7.4 – Алгоритм просування проекту

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи комплексних рішень для відеонагляду необхідно використовувати багатоканальний підхід, залучати партнерів, активно застосовувати цифрові інструменти та аналітику, а також постійно забезпечувати високий рівень підтримки клієнтів. Це дозволить не тільки залучити нових клієнтів, а й утримувати існуючих, збільшуючи рівень лояльності та продажів.

### 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи (наприклад, система відеоаналітики чи система гіперконвергенції ІТ-інфраструктури) залежать від наведених на рисунку 7.5 важливих аспектів.

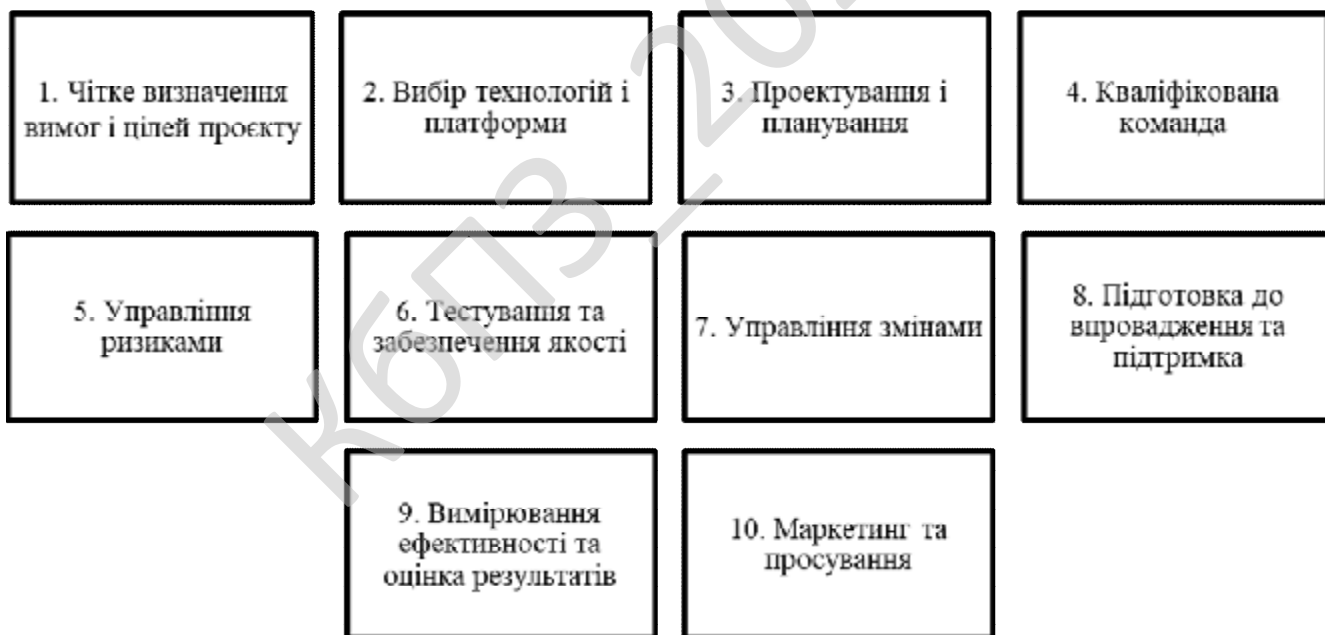


Рисунок 7.5 – Ключові фактори успіху проєкту

Успіх проєкту програмної реалізації системи визначається багатьма факторами, серед яких важливими є чітке планування, ефективне управління командою та ризиками, високий рівень технічної підготовки, а також стратегічне просування і підтримка системи після її впровадження. Збалансований підхід до цих аспектів допоможе досягти максимальних результатів і забезпечить стійкий успіх проєкту.

КБПЗ\_2024

					VKPM-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Охорона праці – система збереження життя і здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційні, технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні та інші заходи.

Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів: інструкцій з охорони праці по кожній професії і загальні; положення про охорону праці; накази з охорони праці; журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] 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» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2].

Науково-технічний прогрес привніс серйозні зміни в умови виробничої діяльності робітників розумової діяльності. Їх праця стала більш інтенсивною, напруженою і вимагає значних витрат розумової, емоційної і фізичної енергії. Це призвело до необхідності у знаходженні комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку. Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

## 8.2 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам. Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах. Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

– евакуаційних шляхів;

					ВКРМ-123.24.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами. Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

### 8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії.

Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які пройшли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, таймбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

#### 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.

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

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

### 8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 35 мм., довжиною  $L=2$  м, та горизонтальний електрод – металева полоса з перетином 35·4 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

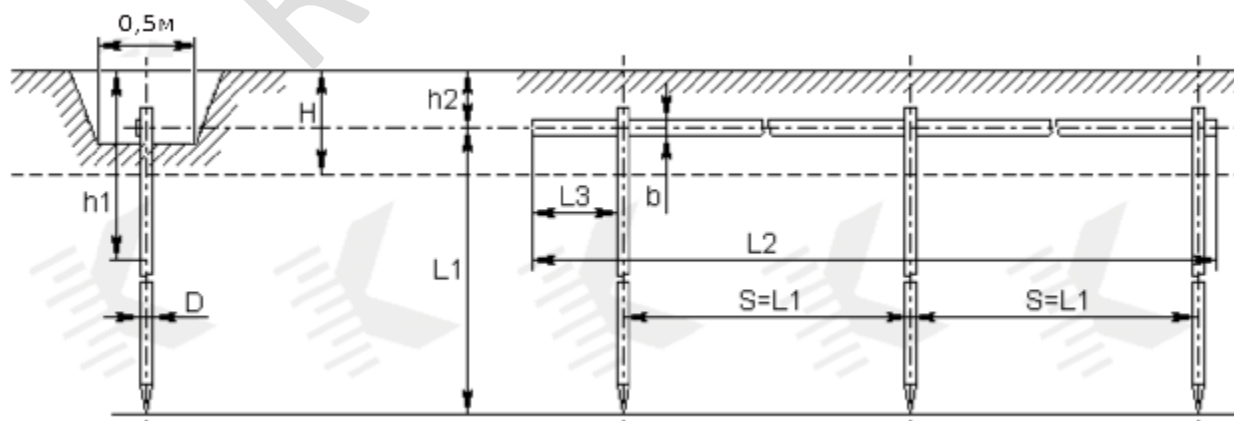


Рисунок 8.1 – Схема штучного заземлення

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79



Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту  $K_{\Pi}$  [10]:

$$R_{\Pi} = 0,366 \cdot (\rho \cdot K_{\Pi} / L_{\Pi}) \cdot \lg(2(L_{\Pi} \cdot L_{\Pi}) / (B \cdot t)) = \\ = 0,366 \cdot (40 \cdot 5 / 40) \cdot \lg((2 \cdot 40^2) / (0,035 \cdot 0,75)) = 11,14 \text{ Ом.}$$

де  $K_{\Pi}=5$  – табличне значення кліматичного коефіцієнта питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [10]:

$B = 35 \text{ мм.} = 0,035 \text{ м.}$  - ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [10]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) = \\ = (21,7 \cdot 11,14) / (21,7 \cdot 0,55 + 10 \cdot 11,14 \cdot 0,53) = 3,4 \text{ Ом.}$$

де  $\eta_{\Pi} = 0,55$  – табличне значення коефіцієнта екранування з'єднуючої полоси [10].

Умова  $R \leq R_{3н}$  виконується ( $3,4 \leq 4$ ).

Так як при 10 вертикальних електродах  $R$  суттєво більше  $R_{3н}$ , зменшимо кількість вертикальних електродів  $N$  до 9 і виконаємо перерахунок. У результаті остаточно отримали:  $R = 3,9 \text{ Ом.}$  при кількості вертикальних електродів  $N = 9$ .

### 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 питань пожежної безпеки, небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи комплексних рішень для відеонагляду.

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 комплексних рішень для відеонагляду.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем комплексних рішень для відеонагляду.

– Досліджена система комплексних рішень для відеонагляду.

– На основі отриманих результатів досліджень створена програмна реалізація системи комплексних рішень для відеонагляду.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комплексних рішень для відеонагляду.

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.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHACAL-2.

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

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>83</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Діденко О.О. Дослідження та програмна реалізація системи комплексних рішень для відеонагляду // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
3. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
4. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
5. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
6. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. – Д.: НГУ, 2016. – 187 с.
7. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
8. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
9. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.
10. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447



International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

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

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

21. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

22. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

23. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

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

25. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and



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

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

35. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

36. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

37. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

38. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

39. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

40. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

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

42. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

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

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

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

46. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» *Комп'ютерні науки та кібербезпека*. № 4. С. 30-37. 2019.

47. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. *Проектування комп'ютерних систем та мереж*. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

					<b>ВКРМ-123.24.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

49. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

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

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

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

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

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

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

					<b>ВКРМ-123.24.0008.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Діденко О.О.				<i>Дослідження та програмна реалізація системи комплексних рішень для відеонагляду</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.					М	1	6
Н. Контр.	Коваленко А.С.				<b>ЦНТУ КІ-23М</b>			
Затв.	Смірнов О.А.							

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи комплексних рішень для відеонагляду.

## 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. № 19-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи комплексних рішень для відеонагляду.

## 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-123.24.0008.00.00.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 комплексних рішень для відеонагляду;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

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

## 5.4 Вимоги до архітектури

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

## 5.5 Вимоги до надійності

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

					<b>ВКРМ-123.24.0008.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0008.00.00.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 маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути виконаний розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					ВКРМ-123.24.0008.00.00.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 – 90 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 Маркетингове та економічне обґрунтування ІТ-проєкту.

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 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2024 р.

					<b>ВКРМ-123.24.0008.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

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

**ЗАТВЕРДЖУЮ**

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

\_\_\_\_\_ Смірнов О.А.

*Дослідження та програмна реалізація  
системи комплексних рішень для відеонагляду*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

### Основна програма

```

import cv2
import numpy as np
import threading
import time
import datetime
import os

# Клас для захоплення відео з декількох камер
class MultiCameraCapture:
    def __init__(self, sources):
        self.captures = [cv2.VideoCapture(src) for src in sources]

    def get_frames(self):
        frames = []
        for capture in self.captures:
            ret, frame = capture.read()
            if ret:
                frames.append(frame)
            else:
                frames.append(None)
        return frames

    def release(self):
        for capture in self.captures:
            capture.release()

# Клас для запису відео з декількох камер
class MultiCameraRecorder:
    def __init__(self, output_dir, frame_size, fps=30):
        self.writers = []
        self.output_dir = output_dir
        for i in range(len(frame_size)):
            fourcc = cv2.VideoWriter_fourcc(*'XVID')
            output_file = os.path.join(self.output_dir,
f'output_camera_{i}.avi')
            writer = cv2.VideoWriter(output_file, fourcc, fps, frame_size[i])
            self.writers.append(writer)

    def write_frames(self, frames):
        for writer, frame in zip(self.writers, frames):
            if frame is not None:
                writer.write(frame)

    def release(self):
        for writer in self.writers:
            writer.release()

# Клас для виявлення руху на основі порівняння попередніх кадрів
class AdvancedMotionDetector:
    def __init__(self, sensitivity=500):
        self.sensitivity = sensitivity
        self.previous_frames = []

    def detect_motion(self, frames):
        motions = []
        for i, frame in enumerate(frames):
            if frame is None:
                motions.append(False)
                continue

            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            gray = cv2.GaussianBlur(gray, (21, 21), 0)

            if len(self.previous_frames) <= i:
                self.previous_frames.append(gray)

```

```

        motions.append(False)
        continue

        frame_delta = cv2.absdiff(self.previous_frames[i], gray)
        thresh = cv2.threshold(frame_delta, 25, 255, cv2.THRESH_BINARY)[1]
        thresh = cv2.dilate(thresh, None, iterations=2)
        motion_area = np.sum(thresh)

        self.previous_frames[i] = gray

        motions.append(motion_area > self.sensitivity)
    return motions

# Клас для обробки подій на основі часу доби
class TimeBasedEventProcessor:
    def __init__(self, start_hour, end_hour):
        self.start_hour = start_hour
        self.end_hour = end_hour

    def is_within_time_range(self):
        current_hour = datetime.datetime.now().hour
        return self.start_hour <= current_hour <= self.end_hour

# Клас для відеонагляду з підтримкою декількох камер та подій
class MultiCameraSurveillanceSystem:
    def __init__(self, camera_sources, output_dir):
        self.camera = MultiCameraCapture(camera_sources)
        frame_sizes = [(640, 480) for _ in camera_sources]
        self.recorder = MultiCameraRecorder(output_dir, frame_sizes)
        self.motion_detector = AdvancedMotionDetector()
        self.event_processor = TimeBasedEventProcessor(8, 18)

    def run(self):
        while True:
            frames = self.camera.get_frames()
            motions = self.motion_detector.detect_motion(frames)

            for i, (frame, motion_detected) in enumerate(zip(frames, motions)):
                if frame is not None:
                    if motion_detected and
self.event_processor.is_within_time_range():
                        timestamp = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
                        cv2.putText(frame, f"Motion detected at camera {i}
{timestamp}", (10, 30),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255),
                                2)

                        self.recorder.write_frames(frames)

                        cv2.imshow(f"Camera {i}", frame)

                        if cv2.waitKey(1) & 0xFF == ord('q'):
                            break

            self.recorder.release()
            self.camera.release()
            cv2.destroyAllWindows()

# Клас для відправки повідомлень про події на електронну пошту
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

class EmailNotifier:
    def __init__(self, sender_email, receiver_email, password):
        self.sender_email = sender_email
        self.receiver_email = receiver_email
        self.password = password

```

```

def send_motion_alert(self, camera_index, timestamp):
    subject = f"Motion detected at camera {camera_index}"
    body = f"Motion was detected at camera {camera_index} on {timestamp}"

    msg = MIMEMultipart()
    msg['From'] = self.sender_email
    msg['To'] = self.receiver_email
    msg['Subject'] = subject

    msg.attach(MIMEText(body, 'plain'))

    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(self.sender_email, self.password)
        text = msg.as_string()
        server.sendmail(self.sender_email, self.receiver_email, text)
        server.quit()
    except Exception as e:
        print(f"Failed to send email {str(e)}")

# Клас для інтеграції всіх систем у одну платформу
class IntegratedSurveillanceSystem:
    def __init__(self, camera_sources, output_dir, email_info):
        self.surveillance_system = MultiCameraSurveillanceSystem(camera_sources,
        output_dir)
        self.email_notifier = EmailNotifier(email_info['sender_email'],
        email_info['receiver_email'], email_info['password'])

    def run(self):
        while True:
            frames = self.surveillance_system.camera.get_frames()
            motions =
self.surveillance_system.motion_detector.detect_motion(frames)

            for i, (frame, motion_detected) in enumerate(zip(frames, motions)):
                if frame is not None:
                    if motion_detected and
self.surveillance_system.event_processor.is_within_time_range():
                        timestamp = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
                        cv2.putText(frame, f"Motion detected at camera {i
{timestamp}", (10, 30),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
                        self.surveillance_system.recorder.write_frames(frames)
                        self.email_notifier.send_motion_alert(i, timestamp)

                        cv2.imshow(f"Camera {i}", frame)

                        if cv2.waitKey(1) & 0xFF == ord('q'):
                            break

            self.surveillance_system.recorder.release()
            self.surveillance_system.camera.release()
            cv2.destroyAllWindows()

# Основна система для запуску відеонагляду
if __name__ == "__main__":
    camera_sources = [0, 1] # Можна додати інші джерела відео
    output_dir = 'video_output'
    email_info = {
        'sender_email': 'example@gmail.com',
        'receiver_email': 'receiver@gmail.com',
        'password': 'yourpassword'
    }

    system = IntegratedSurveillanceSystem(camera_sources, output_dir,
    email_info)
    system.run()

```

## Файл cloud\_storage.py

```
import os
import boto3
from botocore.exceptions import NoCredentialsError

# Клас для завантаження відео в хмару Amazon S3
class CloudStorageUploader:
    def __init__(self, aws_access_key, aws_secret_key, bucket_name):
        self.s3 = boto3.client('s3', aws_access_key_id=aws_access_key,
                               aws_secret_access_key=aws_secret_key)
        self.bucket_name = bucket_name

    def upload_file(self, file_name, object_name=None):
        if object_name is None:
            object_name = os.path.basename(file_name)

        try:
            self.s3.upload_file(file_name, self.bucket_name, object_name)
            print(f"File {file_name} uploaded to {self.bucket_name}")
        except FileNotFoundError:
            print(f"The file {file_name} was not found")
        except NoCredentialsError:
            print("Credentials not available")

# Приклад використання
if __name__ == "__main__":
    uploader = CloudStorageUploader('AWS_ACCESS_KEY', 'AWS_SECRET_KEY',
    'mybucket')
    uploader.upload_file('video_output/output_camera_0.avi')
```

## Файл access\_control.py

```
# Клас для симуляції інтеграції з системою управління доступом
class AccessControlSystem:
    def __init__(self):
        self.authorized_ids = ["1234", "5678", "91011"] # ID авторизованих осіб

    def check_access(self, person_id):
        if person_id in self.authorized_ids:
            print(f"Access granted to ID {person_id}")
            return True
        else:
            print(f"Access denied to ID {person_id}")
            return False

# Приклад використання
if __name__ == "__main__":
    access_system = AccessControlSystem()
    access_system.check_access("1234")
```

КБПЗ\_2024

## Файл face\_recognition.py

```
import cv2
import face_recognition

# Клас для розпізнавання облич
class FaceRecognitionSystem:
    def __init__(self, known_faces):
        self.known_faces_encodings = [face_recognition.face_encodings(face)[0]
for face in known_faces]
        self.known_faces_names = ["Person1", "Person2", "Person3"]

    def recognize_faces(self, frame):
        face_locations = face_recognition.face_locations(frame)
        face_encodings = face_recognition.face_encodings(frame, face_locations)

        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(self.known_faces_encodings,
face_encoding)
            name = "Unknown"
            if True in matches:
                first_match_index = matches.index(True)
                name = self.known_faces_names[first_match_index]
            face_names.append(name)
        return face_locations, face_names

# Приклад використання
if __name__ == "__main__":
    # Завантажуємо відомі обличчя
    known_face_images = [face_recognition.load_image_file("person1.jpg"),
        face_recognition.load_image_file("person2.jpg")]

    face_rec_system = FaceRecognitionSystem(known_face_images)
    frame = cv2.imread("test_image.jpg")
    locations, names = face_rec_system.recognize_faces(frame)
    print(names)
```

## Файл video\_streaming.py

```
from flask import Flask, render_template, Response
import cv2

# Клас для передачі відеопотоку на веб-сайт
class VideoStreaming:
    def __init__(self, source=0):
        self.camera = cv2.VideoCapture(source)

    def generate_frames(self):
        while True:
            success, frame = self.camera.read()
            if not success:
                break
            else:
                ret, buffer = cv2.imencode('.jpg', frame)
                frame = buffer.tobytes()
                yield (b'--frame\r\n'
                    b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

# Ініціалізація Flask-додатку
app = Flask(__name__)

video_stream = VideoStreaming()

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/video_feed')
def video_feed():
    return Response(video_stream.generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == "__main__":
    app.run(debug=True)
```

## Файл high\_res\_camera.py

```
import cv2

# Клас для роботи з камерами високої роздільної здатності
class HighResolutionCamera:
    def __init__(self, source=0):
        self.camera = cv2.VideoCapture(source)
        self.camera.set(cv2.CAP_PROP_FRAME_WIDTH, 1920) # Встановлення ширини
        self.camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080) # Встановлення висоти

    def get_frame(self):
        ret, frame = self.camera.read()
        if ret:
            return frame
        else:
            return None

# Приклад використання
if __name__ == "__main__":
    high_res_cam = HighResolutionCamera()
    frame = high_res_cam.get_frame()
    cv2.imshow("High Resolution Camera", frame)
    cv2.waitKey(0)
    high_res_cam.camera.release()
    cv2.destroyAllWindows()
```

КБПЗ\_2024

## Файл mobile\_notification.py

```
import requests

# Клас для відправки push-повідомлень на мобільні додатки
class MobileNotifier:
    def __init__(self, api_key, device_token):
        self.api_key = api_key
        self.device_token = device_token

    def send_notification(self, title, message):
        url = "https://fcm.googleapis.com/fcm/send"
        headers = {
            "Authorization": f"key={self.api_key}",
            "Content-Type": "application/json"
        }
        payload = {
            "to": self.device_token,
            "notification": {
                "title": title,
                "body": message
            }
        }
        response = requests.post(url, json=payload, headers=headers)
        if response.status_code == 200:
            print("Notification sent successfully")
        else:
            print("Failed to send notification")

# Приклад використання
if __name__ == "__main__":
    notifier = MobileNotifier('API_KEY', 'DEVICE_TOKEN')
    notifier.send_notification("Motion Alert", "Motion detected at Camera 1")
```

## Файл object\_tracking.py

```
import cv2

# Клас для відстеження об'єктів у відео
class ObjectTracker:
    def __init__(self):
        self.tracker = cv2.TrackerCSRT_create()
        self.initialized = False

    def initialize_tracker(self, frame, bbox):
        self.tracker.init(frame, bbox)
        self.initialized = True

    def update(self, frame):
        if self.initialized:
            success, bbox = self.tracker.update(frame)
            if success:
                p1 = (int(bbox[0]), int(bbox[1]))
                p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))
                cv2.rectangle(frame, p1, p2, (255, 0, 0), 2, 1)
            return frame
        else:
            return frame

# Приклад використання
if __name__ == "__main__":
    cap = cv2.VideoCapture(0)
    tracker = ObjectTracker()
    ret, frame = cap.read()

    bbox = cv2.selectROI(frame, False)
    tracker.initialize_tracker(frame, bbox)

    while True:
        ret, frame = cap.read()
        if not ret:
            break
        frame = tracker.update(frame)
        cv2.imshow('Tracking', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

## Файл video\_with\_audio.py

```

import cv2
import pyaudio
import wave
import threading

# Клас для запису аудіо
class AudioRecorder:
    def __init__(self, output_file):
        self.chunk = 1024
        self.format = pyaudio.paInt16
        self.channels = 2
        self.rate = 44100
        self.output_file = output_file
        self.p = pyaudio.PyAudio()
        self.stream = self.p.open(format=self.format, channels=self.channels,
                                   rate=self.rate, input=True,
                                   frames_per_buffer=self.chunk)

        self.frames = []
        self.recording = True

    def record_audio(self):
        while self.recording:
            data = self.stream.read(self.chunk)
            self.frames.append(data)

    def stop(self):
        self.recording = False
        self.stream.stop_stream()
        self.stream.close()
        self.p.terminate()

    def save_audio(self):
        wf = wave.open(self.output_file, 'wb')
        wf.setnchannels(self.channels)
        wf.setsampwidth(self.p.get_sample_size(self.format))
        wf.setframerate(self.rate)
        wf.writeframes(b''.join(self.frames))
        wf.close()

# Клас для запису відео та аудіо одночасно
class VideoWithAudioRecorder:
    def __init__(self, video_output, audio_output):
        self.video_output = video_output
        self.audio_output = audio_output
        self.audio_recorder = AudioRecorder(self.audio_output)
        self.video_recorder = None

    def record(self):
        self.audio_thread =
threading.Thread(target=self.audio_recorder.record_audio)
        self.audio_thread.start()

        self.video_recorder = cv2.VideoWriter(self.video_output,
cv2.VideoWriter_fourcc(*'XVID'), 20.0, (640, 480))
        cap = cv2.VideoCapture(0)

        while True:
            ret, frame = cap.read()
            if ret:
                self.video_recorder.write(frame)
                cv2.imshow("Recording", frame)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break

        self.audio_recorder.stop()
        self.video_recorder.release()

```

```
cap.release()
cv2.destroyAllWindows()
self.audio_recorder.save_audio()

# Приклад використання
if __name__ == "__main__":
    recorder = VideoWithAudioRecorder('output_video.avi', 'output_audio.wav')
    recorder.record()
```

КБПЗ\_2024

## Файл data\_encryption.py

```
from cryptography.fernet import Fernet

# Клас для шифрування та дешифрування файлів
class DataEncryptor:
    def __init__(self, key=None):
        if key:
            self.key = key
        else:
            self.key = Fernet.generate_key()
            self.cipher = Fernet(self.key)

    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as file:
            file_data = file.read()
            encrypted_data = self.cipher.encrypt(file_data)
            with open(f'{file_name}.enc', 'wb') as file:
                file.write(encrypted_data)
            print(f"File {file_name} encrypted")

    def decrypt_file(self, file_name):
        with open(file_name, 'rb') as file:
            encrypted_data = file.read()
            decrypted_data = self.cipher.decrypt(encrypted_data)
            with open(f'decrypted_{file_name}', 'wb') as file:
                file.write(decrypted_data)
            print(f"File {file_name} decrypted")

# Приклад використання
if __name__ == "__main__":
    encryptor = DataEncryptor()
    encryptor.encrypt_file('output_video.avi')
    encryptor.decrypt_file('output_video.avi.enc')
```

## Файл time\_based\_recording.py

```
import cv2
import datetime

# Клас для запису відео на основі часу доби
class TimeBasedVideoRecorder:
    def __init__(self, start_hour, end_hour, video_output):
        self.start_hour = start_hour
        self.end_hour = end_hour
        self.video_output = video_output
        self.video_recorder = None

    def within_time_range(self):
        current_hour = datetime.datetime.now().hour
        return self.start_hour <= current_hour <= self.end_hour

    def start_recording(self):
        cap = cv2.VideoCapture(0)
        self.video_recorder = cv2.VideoWriter(self.video_output,
        cv2.VideoWriter_fourcc(*'XVID'), 20.0, (640, 480))

        while True:
            ret, frame = cap.read()
            if not ret:
                break

            if self.within_time_range():
                self.video_recorder.write(frame)
                cv2.imshow("Recording", frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        cap.release()
        self.video_recorder.release()
        cv2.destroyAllWindows()

# Приклад використання
if __name__ == "__main__":
    recorder = TimeBasedVideoRecorder(8, 18, 'time_based_output.avi')
    recorder.start_recording()
```

## Файл sms\_notification.py

```
from twilio.rest import Client

# Клас для відправки SMS-повідомлень через Twilio
class SMSNotifier:
    def __init__(self, account_sid, auth_token, sender_phone, receiver_phone):
        self.client = Client(account_sid, auth_token)
        self.sender_phone = sender_phone
        self.receiver_phone = receiver_phone

    def send_sms(self, message):
        message = self.client.messages.create(
            body=message,
            from_=self.sender_phone,
            to=self.receiver_phone
        )
        print(f"Message sent: {message.sid}")

# Приклад використання
if __name__ == "__main__":
    notifier = SMSNotifier('ACCOUNT_SID', 'AUTH_TOKEN', '+1234567890',
        '+0987654321')
    notifier.send_sms("Motion detected at Camera 1")
```

КБПЗ\_2024

## Файл web\_interface.py

```
from flask import Flask, render_template, request, jsonify
import cv2

app = Flask(__name__)

# Клас для управління камерами через веб-інтерфейс
class WebCameraControl:
    def __init__(self, source=0):
        self.camera = cv2.VideoCapture(source)

    def get_frame(self):
        success, frame = self.camera.read()
        if success:
            ret, buffer = cv2.imencode('.jpg', frame)
            return buffer.tobytes()
        return None

camera_control = WebCameraControl()

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/get_frame', methods=['GET'])
def get_frame():
    frame = camera_control.get_frame()
    if frame:
        return Response(frame, mimetype='image/jpeg')
    return jsonify({'error': 'Frame not available'})

@app.route('/start_camera', methods=['POST'])
def start_camera():
    return jsonify({'message': 'Camera started'})

@app.route('/stop_camera', methods=['POST'])
def stop_camera():
    camera_control.camera.release()
    return jsonify({'message': 'Camera stopped'})

if __name__ == "__main__":
    app.run(debug=True)
```

## Файл file\_cleanup.py

```
import os
import time

# Клас для автоматичного видалення файлів, старших за певний період
class FileCleanup:
    def __init__(self, directory, max_age_days):
        self.directory = directory
        self.max_age = max_age_days * 86400 # днів у секундах

    def cleanup(self):
        now = time.time()
        for filename in os.listdir(self.directory):
            file_path = os.path.join(self.directory, filename)
            if os.path.isfile(file_path):
                file_age = now - os.path.getmtime(file_path)
                if file_age > self.max_age:
                    os.remove(file_path)
                    print(f"Deleted {file_path}")

# Приклад використання
if __name__ == "__main__":
    cleaner = FileCleanup('video_output', 30)
    cleaner.cleanup()
```

КБПЗ\_2024

## Файл license\_plate\_recognition.py

```
import cv2
import pytesseract

# Клас для розпізнавання номерних знаків
class LicensePlateRecognizer:
    def __init__(self, tesseract_cmd):
        pytesseract.pytesseract.tesseract_cmd = tesseract_cmd

    def recognize_plate(self, frame):
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        plates = pytesseract.image_to_string(gray, config='--psm 8 --oem 3')
        return plates

# Приклад використання
if __name__ == "__main__":
    recognizer = LicensePlateRecognizer(r'C:\Program Files\Tesseract-OCR\tesseract.exe')
    frame = cv2.imread('car_image.jpg')
    plate_text = recognizer.recognize_plate(frame)
    print(f"License Plate: {plate_text}")
```

КБПЗ\_2024