

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 керування**  
**гучномовцями в системах відеоспостереження”**

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

Керівник проекту  
доктор філософії (PhD)  
\_\_\_\_\_ Усік П.С.  
« \_\_\_\_ » \_\_\_\_\_ 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

Осуському Артему Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження

2. Керівник роботи Усік Павло Сергійович, доктор філософії (PhD)

(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 керування гучномовцями в системах відеоспостереження

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування ІТ-проєкту.

3. Опис і обґрунтування проєктних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Показники економічної ефективності 1 аркуш

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	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

**Осуський А.В. Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження. 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 керування гучномовцями в системах відеоспостереження.

Метою розробки є дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження.

Об'єктом дослідження є процес керування гучномовцями в системах відеоспостереження.

Предметом дослідження є методи керування гучномовцями в системах відеоспостереження.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи керування гучномовцями в системах відеоспостереження.

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 Visual C#.

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

## ABSTRACT

**Osuskii A.V. Research and software implementation of a loudspeaker control system in video surveillance systems. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for a loudspeaker control system in video surveillance systems.

The purpose of the development is the research and software implementation of a loudspeaker control system in video surveillance systems.

The object of the research is the process of loudspeaker control in video surveillance systems.

The subject of the research is methods of loudspeaker control in video surveillance systems.

The research methods are based on methods of coding theory, methods of mathematical statistics, and methods of software development.

The result of the work is a software implementation of a loudspeaker control system in video surveillance systems.

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 the software are provided.

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

The program was developed in the Visual C# 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.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання .....	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	19
3.1 Опис функціонування системи .....	19
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми .....	38
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	53
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	56
6 НАУКОВА НОВИЗНА .....	63

						ВКРМ-123.24.0029.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Осуський А.В.				Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження	Літ.	Аркуш	Аркушів
Перев.	Усік П.С.					М	1	91
Н.контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	64
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	64
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	65
7.3	Вибір методу оцінки вартості ПЗ .....	67
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	68
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	70
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	72
7.7	Визначення ключових факторів успіху конкретного проєкту.....	73
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	74
8.1	Вступ.....	74
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	75
8.3	Аналіз умов праці на робочому місці програміста.....	76
8.4	Розрахункова частина .....	79
8.5	Висновки до розділу.....	82
9	ОСНОВНІ ВИСНОВКИ.....	83
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	85

КБПЗ-2024

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- ДКП – дискретне косинусне перетворення
- ПЗ – програмне забезпечення
- ЛОМ – локальна обчислювальна мережа
- ССТV – система замкнутого телебачення
- IP – інтернет протокол
- USB – універсальна серійна шина

КБПЗ\_2024

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Оснащення системи відеоспостереження гучномовцями дозволяє досить ефективно запобігати здійсненню небажаних дій. З їхньою допомогою оператор системи може віддалено давати вказівки й задавати питання як відвідувачам, так і незваним гостям. Тому гучномовцям у системах відеоспостереження приділяється досить важлива роль. Якщо поведження людини, що попала в поле зору камери, викликає підозру (наприклад він проникає на територію з обмеженим доступом), охоронець, що перебуває за пультом системи відеоспостереження, може негайно застерегти його від можливих неправомірних дій по голосному зв'язку. У багатьох випадках цього цілком достатньо для того, щоб уникнути небажаних інцидентів. Таким чином, використання системи відеоспостереження допомагає вжити які попереджують заходам і запобіганню правопорушення. Перелік потенційних застосувань мережевих відеотехнологій практично нескінченний, однак більша їхня частина відноситься до сфери забезпечення безпеки й дистанційного спостереження за людьми, територіями, майном і виробничою діяльністю.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження.

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

- Огляд існуючих систем керування гучномовцями в системах відеоспостереження.
- Дослідження системи керування гучномовцями в системах відеоспостереження.
- Програмна реалізація системи керування гучномовцями в системах відеоспостереження.

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4



# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Гучномовці – це пристрої, що перетворюють електричні сигнали в чутні механічні хвилі тиску (звук). Існують різні типи гучномовців, але в системах відеоспостереження вибрати доводиться, як правило, між динамічними й рупорними.

### Динамічні гучномовці.

Динамічний гучномовець має полегшену діафрагму – конус, з'єднаний механічним способом з голосовою котушкою. Пристрою такого типу застосовуються в багатьох додатках. Як приклад можна привести стельові й настінні динаміки, а також системи Hi-Fi. Недоліком динамічних гучномовців є їх низький ККД: тільки 1% електричної енергії, переданої на динамік з підсилювача, перетвориться в акустичну енергію.

### Гучномовці рупорного типу

Завдяки акустичному рупору загальна ефективність цієї системи набагато вище. Вхідний до складу пристрою компресійний привод підключається безпосередньо до рупора. Оскільки така конструкція в десять разів збільшує потужність вихідного сигналу, що надходить із підсилювача, рупорні гучномовці часто використовуються в системах оповіщення. У порівнянні з іншими типами динаміків трансльований ними звук чутний далі.

Недолік же рупорних гучномовців полягає в тому, що їх оптимальний частотний діапазон обмежений. Тому вони малопригодні для відтворення музики. Зате для трансляції мови (особливо при необхідності охопту більших просторів) рупорні гучномовці підходять як не можна краще.

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 1.2 Область застосування

Нижче наведені деякі типові можливі застосування в ключових галузевих сегментах.

### Роздрібна торгівля

Системи мережевого відеоспостереження можуть значно знизити рівень злочинства, підвищити безпека персоналу й оптимізувати керування магазинами.

Важливою перевагою мережевого 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 мережевого відеоспостереження відрізняються високим рівнем сумісності й найкоротших строків окупності.

Мережеві 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

Мережеве 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. Мережеві відеотехнології також можна застосовувати для контролю транспортних потоків з метою зниження заторів і підвищення ефективності. 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

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

складних ситуацій, наприклад, для відеоспостереження в автобусах і в поїздах, пропонуються мережеві камери, здатні витримувати мінливі температури, високу вологість, запиленість, вібрації й вандалізм.

### **Утворення**

В освітніх установах, від дитячого садка до університету, мережеві відеотехнології допомагають боротися з вандалізмом і забезпечувати безпеку персоналу й що навчаються. Якщо освітня установа вже розташовує ІТ-інфраструктурою, мережеве відеоспостереження може бути переважніше й економічно ефективніше аналогової системи, оскільки в багатьох випадках не потрібно нова кабельна проводка. Крім того, засоби керування подіями в системі мережевого відеоспостереження дозволяють видавати сигнали тривоги й у реальному часі надавати операторам точну картину, що відбувається для прийняття рішень. Мережеві відеотехнології також можна використовувати для дистанційного навчання – наприклад, для студентів, які не мають можливості відвідувати заняття очно.

### **Промисловість**

Мережеві 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.

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

### **Безпечні міста**

Мережеве 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 мережевого відеоспостереження в масштабі цілих міст. Дистанційне відеоспостереження за допомогою мережевих відеокамер дозволяє органам поліції оперативно реагувати на злочини, що виявляються в реальному часі.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## **Урядові організації**

Устаткування для мережевого 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 мережевого відеоспостереження дозволяють одержувати статистичну інформацію – наприклад, про кількість відвідувачів будинку.

## **Охорона здоров'я**

Мережеві 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**

Мережеві 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, мережеві відеотехнології поступово впроваджуються й у цьому секторі, особливо в банках, де потрібне висока якість зображення й можливість ідентифікувати людей по відеозапису.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## 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

Універсальне ПЗ для централізованого керування засобами IP-відеоспостереження, розраховане на роботу з камерами різних виробників, як правило, коштує чималих грошей. Але в той же час існують і безкоштовні програмні продукти, хоча вони, звичайно ж, мають певні обмеження.

Безкоштовний сир буває не тільки в мишоловці. У сучасному комерційному суспільстві – це найчастіше спосіб залучити клієнта, викликавши в нього апетит. Принцип рекламної годівлі активно застосовують і маркетологи IT-сфери. Інтернет під зав'язку набитий усілякими «безкоштовними», «пробними» і «тестовими» версіями програмного забезпечення. Більшість розроблювачів, особливо із числа тих, які пропонують ПЗ для корпоративної сфери, щиро сподіваються, що спробувавши зручність і якість тестового зразка, покупець заплатить гарну ціну за повну версію. У теж час у багатьох випадках пробної версії може цілком вистачити для того, щоб повністю вирішити завдання потенційного клієнта. Трохи безкоштовних, але досить функціональних програмних розробок пропонується й у сфері IP-відеоспостереження.

#### **Milestone XProject Smart Client**

XProject Go, як і всі розробки Milestone, підтримує біля дев'ятисот моделей камер більш ніж вісімдесятьох різних виробників (повний список можна побачити на сайті компанії) незалежно від того, у який альянс вони входять – ONVIF або PSIA. Також є можливість використання відеокамер, що підключаються по USB. Крім того, до системи можна приєднати також аналогові камери за допомогою спеціального IP-декодера; XProject Go буде працювати

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

також і з ними. Незважаючи на те що пробна версія продукту є безкоштовної, у ній підтримуються сучасні формати стиску відео: MPEG4 ASP, MxPEG, H.264. До убудованого «інтелектуальним» функціям можна віднести програмний детектор руху, що працює незалежно від камери.



Рисунок 2.1 – Вікно керування ПЗ Milestone XProject Smart Client

Пробну версію можна використовувати протягом місяця, після чого вартість ліцензії складе від тридцяти до п'ятисот британських фунтів (залежно від кількості підключених камер).

Система підтримує необмежене число відеокамер (аналогових, IP, USB), серверів і обсяг архіву. Використовуються кодеки MJPEG і MPEG4. Є убудований веб-сервер, функції детектування руху, оповіщення відповідальних осіб по e-mail, автоматичного завантаження даних на зазначений FTP-сервер.

Цікава функція – буфер запису, у який автоматично зберігається відео, зняте за деяке, досить обмежене час, скажемо, за останні десять секунд. Це зроблено для того, щоб у випадку спрацьовування детектора руху можна було побачити не тільки тривожну подію, але й те, що йому передувало. В SecuritySpy є також можливості пошуку подій в архіві й автоматичній його оптимізації (яка зводиться головним чином до видалення застарілих даних). Підтримуються безліч камер, у тому числі роботизованих, таких виробників, як Axis, JVC, Panasonic, Pixord, D-Link і багатьох інших.

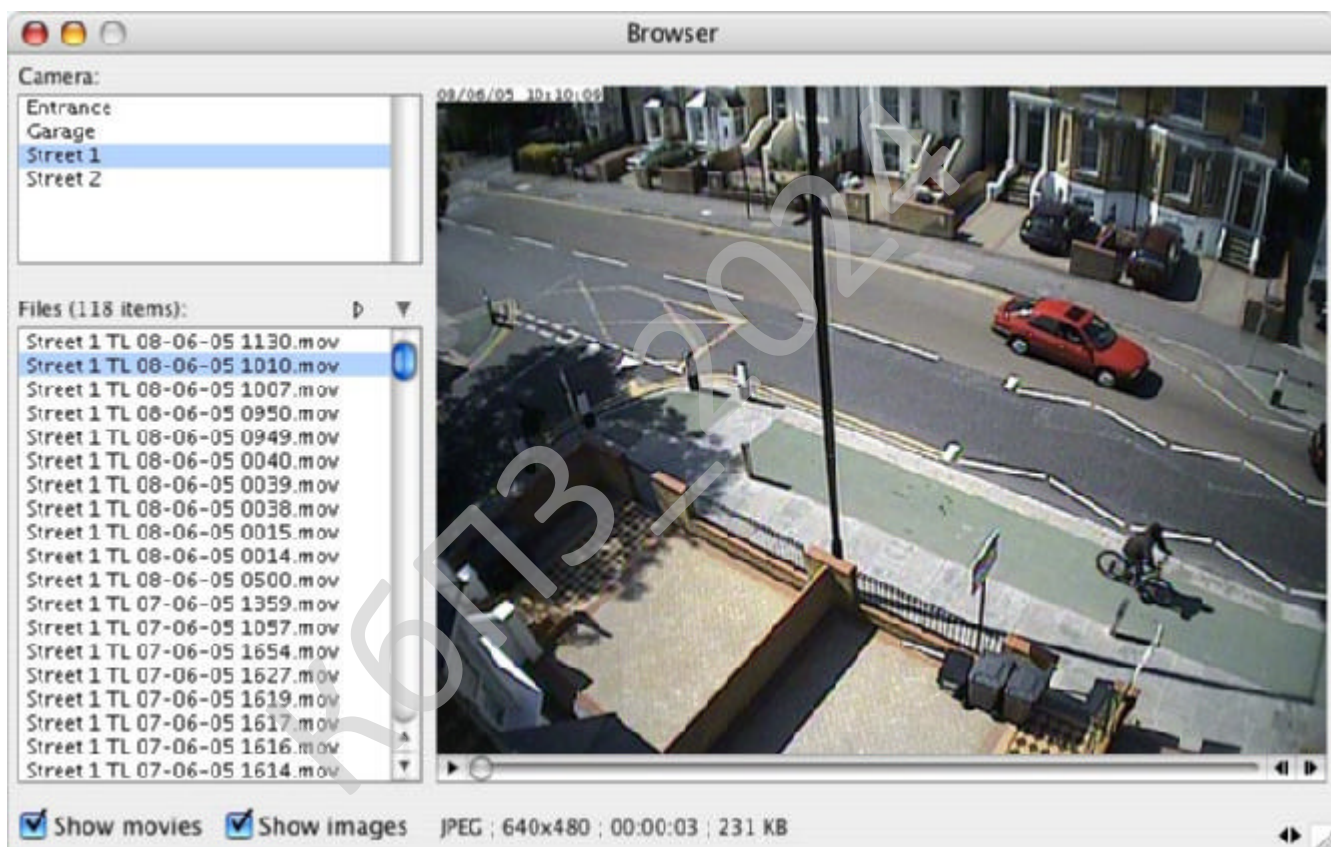


Рисунок 2.2 – Програма BenSoftware SecuritySpy призначена для комп'ютерів Macintosh

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

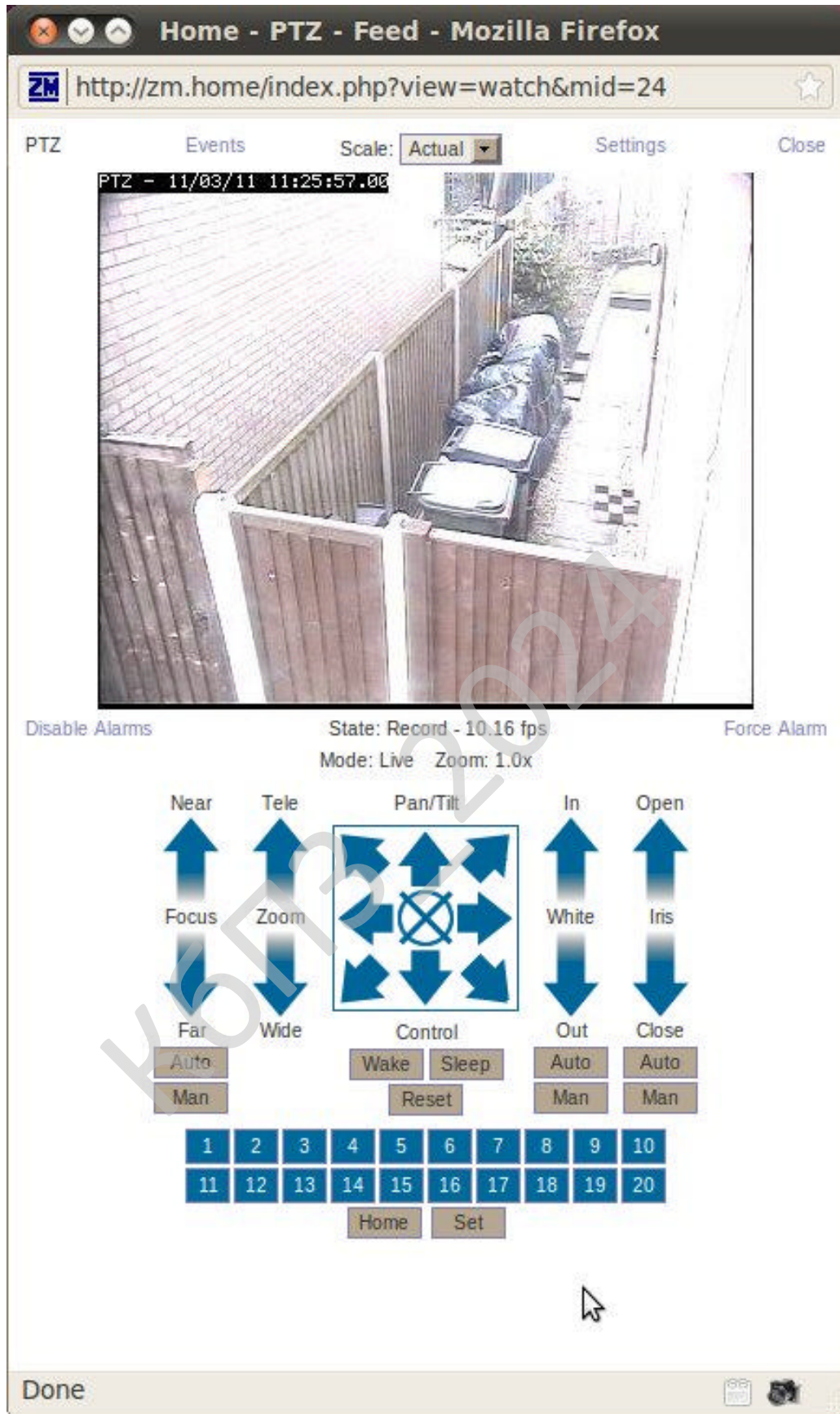


Рисунок 2.3 – Інтерфейс контролю над PTZ-камерою за допомогою ПЗ ZoneMinder

Операційна система Linux теж не залишилася осторонь. Щоб безкоштовно управляти IP-камерами в цьому середовищі, можна використовувати програму ZoneMinder (рисунок 2.3).

Величезною перевагою даного продукту є його безкоштовність, оскільки весь проект існує на добровільні пожертвування. Тут немає обмежень на число відеокамер, користувачів або розмір архіву – все залежить тільки від можливостей ПК або сервера, що обробляє й зберігає відеопотоки. У той же час тут, як і у двох вищезгаданих системах, є підтримка IP-, USB- і аналогових камер різних виробників, детектор руху (з функцією автоматичного спрацьовування запису й відправлення повідомлення по e-mail або SMS), веб-клієнт, багатомовна підтримка й т.д. Оскільки програма ZoneMinder задумана як вільна платформа (побудована на базі C++, Perl і PHP), вона здатна інтегруватися із програмними розробками сторонніх виробників. Також передбачена можливість завантаження відео на FTP-сервер. Це програмне рішення позиціонується як безкоштовна альтернатива не тільки для невеликих компаній і кінцевих користувачів (наприклад, у випадку спостереження за периметром частки будинку), але й для великих організацій.

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

Програмне забезпечення написано мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних додатків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські додатки Windows, веб-служби XML, розподілені компоненти, додатки типу “ сервер-клієнт”, додатки баз даних і багато яких інших. В Visual C# 2012 є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів,

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14



- Властивості, що виступають у ролі методів доступу для закритих змінних-членів.
- Атрибути з декларативними метаданими про типи під час виконання.
- Вбудовані коментарі XML-документації.
- LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів у різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, у мові Visual C# можна використовувати процес, що називається "Interop". Процес Interop дозволяє програмам на Visual C# виконувати практично будь-які дії, які може виконувати вихідний додаток на C++. Мова Visual C# підтримує навіть покажчики й поняття "небезпечного" коду для тих випадків, коли прямий доступ до пам'яті має вкрай важливе значення.

Процес побудови Visual C# у порівнянні з C і C++ простий і є більше гнучким, чим в Java. Немає окремих файлів заголовка, а методи й типи не потрібно повідомляти в певному порядку. У вихідному файлі Visual C# може бути визначене будь-яке число класів, структур, інтерфейсів і подій.

### **Архітектура платформи .NET Framework**

Програма мовою Visual C# виконується в середовищі .NET Framework – інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів. Середовище CLR являє собою комерційну реалізацію корпорацією Майкрософт інфраструктури CLI, що є міжнародним стандартом, який лежить в основі створення середовищ виконання й розробки, у яких забезпечується тісна взаємодія між мовами й бібліотеками.

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, названому складанням, з розширенням EXE або DLL у більшості випадків. Складання

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

При виконанні програми на Visual C# складання завантажується в середовище CLR залежно від відомостей у маніфесті. Далі, якщо вимоги безпеки дотримані, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, що відносяться до автоматичного збору сміття, обробки виключень і керуванню ресурсами. Код, виконуваний середовищем CLR, іноді називають "керованим кодом" у протиставлення "некерованому коду", що компілюється в машинний код, призначений для певної системи. Далі показані відносини під час компіляції й час виконання між файлами з вихідним кодом Visual C#, бібліотеками класів .NET Framework, складаннями й середовищем CLR.

Взаємодія між мовами є ключовою особливістю .NET Framework. Оскільки код IL, створюваний компілятором Visual C# відповідає специфікації CTS, код IL на основі Visual C# може взаємодіяти з кодом, створюваним версіями мов Visual Basic, Visual C++, Visual J# платформи .NET Framework і ще більш ніж 20 CTS-сумісних мов. В одному складанні може бути кілька модулів, написаних на різних мовах платформи .NET Framework, і типи можуть посилатися один на одного, як якби вони були написані на одній мові.

Крім служб часу виконання, в .NET Framework також є велика бібліотека, що складається з більш ніж 4000 класів, організованих по просторах імен, які забезпечують різноманітні корисні функції для будь-яких дій, починаючи від введення й виведення файлів для керування рядками для розбивки XML, і закінчуючи елементами керування Windows Forms. У звичайному додатку мовою Visual C# бібліотека класів .NET Framework інтенсивно використовується для "устрою" коду.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування гучномовцями в системах відеоспостереження.

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-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Організація системи відеоспостереження

Система відеоспостереження може бути організована різними способами, для цього необхідно визначитися з декількома моментами: площею й периметрами охоронюваного об'єкта, кількістю й типами відеокамер і прийомних моніторів, способом передачі, запису, зберігання відеоінформації, наявністю нічної відеозйомки й аудіозапису, вибором централізованої або розподіленої архітектури, інтеграцією з іншими програмними комплексами, припустимою вартістю системи.

#### Проектування

Після того як замовник визначився з рядом ключових моментів, він представляє проектувальникові складене відповідно до керівного документа (РД 25.952-90) технічне завдання, а той проводить передпроектне обстеження, попереднє проектування, становить проектно-кошторисну документацію, затверджує із замовником робочий проект.

При проектуванні проводиться цілий ряд робіт, серед них:

- Визначення архітектури системи відеоспостереження.
- Вибір її ємності з обліком можливого подальшого розвитку системи, включаючи місткість сховища даних.
- Підбор сумісного встаткування, що відповідає побажанням замовника й вимогам регулюючих норм.

#### Розрахунок вартості системи відеоспостереження

Скільки коштує система відеоспостереження? Від 20 тисяч до декількох мільйонів гривень – тут багато чого залежить від замовника, а також від застосовності певних систем для конкретних цілей. Тому, щоб обговорювати

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

вартість комплексу відеоспостереження, потрібно врахувати всі побажання клієнта: обрані їм бренди, масштаб і тип системи відеоспостереження й т.д.

Вартість проектування у випадку замовлення повного комплексу робіт з поставки й інсталяції системи відеоспостереження в середньому становить:

- розраховуючи на 4 відеокамери – від 12 000 гривень;
- розраховуючи на 16 відеокамер – від 30 000 гривень.

Якщо підрядникові поручається тільки проектування, а покупку, монтаж і обслуговування будуть здійснювати інші організації, то ціна питання зростає й становить близько 90 000 гривень і вище.

Вартість монтажних робіт визначається їхнім обсягом і місцем розташування об'єкта, але в середньому воно розраховується в такий спосіб:

- Установка однієї відеокамери в приміщенні – від 1400 гривень.
- Установка однієї відеокамери на вулиці – від 2000 гривень.
- Установка однієї купольної роботизованої камери – від 3500 гривень.
- Установка й настроювання відеосервера – від 2000 гривень.
- Установка й настроювання відеореєстратора – від 1800 гривень.
- Інсталяція й програмування комп'ютерної системи – від 2500 гривень.
- Прокладка кабелів за один метр (залежно від способу прокладки) – від 40 до 300 гривень і т.д.

Вище викладений лише орієнтовний і неповний приклад, у якому не можуть бути враховані всі можливі варіанти й особливості, однак ясно, що якщо купувати систему відеоспостереження, проектувати її й робити монтаж в однієї перевіреної організації, те це завжди виявляється дешевше, ніж розподіляти роботи між декількома фірмами. З наростанням сукупного масштабу інстальної системи відеоспостереження, що діють знижки на встаткування й роботи здешевляють покупку в цілому.

А також, солідні фірми-інсталятори пропонують значні строки гарантійного обслуговування – до 3 років і потім здатні обслуговувати комплекс відеоспостереження по більше низьким, чим у конкурентів цінам.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20





## **Зона покриття**

Після вибору типу динаміка необхідно встановити зону покриття, після чого визначаються рівень звукового тиску, дальність оповіщення й область покриття (сектор), а також кількість необхідних гучномовців.

## **Звуковий тиск**

Гучність звуку залежить від рівня звукового тиску. Кожні додаткові 10 Дб приводять до дворазового збільшення гучності. Наприклад, 120 Дб у два рази голосніше, ніж 110 Дб. Рівні звукового тиску можна проілюструвати на наступних прикладах: мотоцикл (98 Дб), голосний рок-концерт (115 Дб), гуркіт грому поблизу (120 Дб), реактивний літак на зльоті (133 Дб). Дискомфортним для людського вуха прийнято вважати рівень 120 Дб, а больовий поріг перебуває в районі 130 Дб. Значення звукового тиску для конкретної зони буде залежати від рівня фонового шуму й необхідної дальності. Звичайно рекомендується вибирати ледве могутніший динамік, оскільки гучність може знижуватися.

## **Дальність дії й зона покриття**

Різні динаміки забезпечують різні дальність дії й зону покриття. У загальному випадку чим ширше розчин кута покриття, тим менше дальність дії. Звук, що виходить із динаміка з вузькою зоною покриття, чутний далі. Зона покриття варіюється для різних частот і завжди визначається приблизно. Інтенсивність звуку над усе посередині й знижується в міру видалення від центра. Те ж саме спостерігається й у джерел світла: яскравість максимальна в центрі й зменшується по краях.

Варто врахувати й те, що звук відбивається від більшості поверхонь. І в цьому випадку теж можна провести аналогію зі світлом, що відбивається від висячих на стінах дзеркал. У деяких ситуаціях це негативно позначається на якості звуку, утрудняючи сприйняття. Особливо сильно знижується якість мови, що іноді стає зовсім нерозбірливою. На відкритому просторі гарного звуку домогтися простіше, ніж у місцях, де є багато перешкод.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## **Спрямованість динаміка**

Після визначення типу динаміка потрібно продумати його орієнтацію в просторі при установці. Динамік повинен бути спрямований на ту область, на якій фокусується увага, а його центр – у місце передбачуваного знаходження людей. При розгортанні систем відеоспостереження динамік монтується звичайно на висоті 3-8 м над землею й направляється на зону, що перебуває під спостереженням. Як ми вже відзначали раніше, звук поширюється так само, як світло лампи: він найбільш інтенсивний у центрі й слабшає до країв. Чим вище динамік піднятий над землею, тим ширше буде зона його покриття й нижче рівень звукового тиску.

## **Розширення зони покриття**

Для збільшення зони покриття може знадобитися установка декількох динаміків. У кожному разі вони повинні бути повернені в різні сторони. На рисунку зверху приводиться приклад використання трьох динаміків у великому дворі будинку, по периметрі якого організоване спостереження.

Звук, що виходить від динаміків, повинен бути чутний у тих же місцях, які контролюються камерами. Це дозволить операторам звертатися до кожного, хто потрапив у зону їхньої уваги, і здійснювати перемикання між різними камерами й динаміками.

Використання гучномовців у системах відеоспостереження є ефективним способом запобігання правопорушень. Динаміки можуть бути інтегровані в систему відеоспостереження по-різному. Для створення ефективного рішення, що буде відповідати пропонованим до нього вимогам, важливо вибрати правильний варіант. При установці динаміка необхідно насамперед брати до уваги зону покриття й потрібний рівень звукового тиску, а також особливості монтажу й обслуговування. Тестування пристрою перед покупкою, коли це можливо, дозволить загодя розставити точки над «і» і визначитися з вибором.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

### 3.2 Розробка структурної схеми

Уявімо такий випадок: у торговому центрі вибухнула бомба. Як виявити зловмисників, а ще краще – як запобігти подібній ситуації? Скажімо, на кожному куті будівлі встановлено неймовірну кількість камер, які працюють цілий день. Але яке значення мають ці камери для заходів безпеки? Навіть якщо пара операторів сидить у своїй диспетчерській і спостерігає за 8 годин на день, немає ніяких гарантій, що можливий інцидент можна буде виявити в момент його виникнення. Тому багато інцидентів виявляють лише після того, як про них повідомляють, і можуть знадобитися дні за днями, щоб перевірити всі дані камер спостереження, щоб зловити лиходія.

У такому сценарії аналіз відеовмісту в реальному часі (VCA) може бути дуже корисним. Він може допомогти впоратися з різними соціальними, політичними, безпековими, виробничими та багатьма іншими проблемами, здійснюючи моніторинг подій у реальному часі та використовуючи багатогранні алгоритми для виявлення проблеми.

Очікується, що до кінця 2027 року ринок відеоспостереження, також відомого як CCTV (системи замкнутого телебачення), зросте до 18 890 мільйонів доларів США. Уявлення, які можна отримати з даних, оброблених із відео, є величезними та вражаючими завдяки своїм можливостям покращити бізнес-розвідку. Завдяки штучному інтелекту відеомоніторинг у реальному часі став достатньо складним, щоб розпізнавати непередбачувані та нетрадиційні моделі з величезних обсягів даних. Це чудове рішення для підприємств, які намагаються покращити свою діяльність з різних точок зору.

Коротке пояснення того, як працює відеомоніторинг, дивіться у відео нижче:

#### **Що таке відеоспостереження в реальному часі і як воно працює?**

Відеоспостереження в режимі реального часу виходить далеко за рамки встановлення камер відеоспостереження: це передова технологія, яка включає

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

аналіз відеоконтенту (VCA) і може автоматично аналізувати відеоконтент, виявляти об'єкти та події на основі попередньо визначених алгоритмів. Зважаючи на те, що відеокамери створюють нескінченну кількість відеоматеріалів, працівникам служби безпеки чи операторам неможливо постійно переглядати кожну камеру та виявляти підозрілу поведінку. Тому найчастіше інциденти розслідуються після їх виявлення. Це займає багато часу, утомлює та не дає результатів, керованих даними.

Щоб покращити традиційні засоби віддаленого відеоспостереження, технологія відеомоніторингу в реальному часі, технології відстеження об'єктів і людей, а також рішення для відстеження активів вступають у гру, щоб легко та швидко ідентифікувати об'єкти та аналізувати їх за лічені хвилини. На основі атрибутів об'єкта, руху та поведінки можна виявити шаблони розпізнавання облич, підозрілих осіб, раптовий спалах полум'я чи будь-які інші ознаки. Таким чином, **моніторинг у режимі реального часу може стримувати злочинну діяльність ще до її початку**.

Технологія інтелектуального аналізу відео допомагає покращити ситуаційну обізнаність за допомогою індивідуальних сповіщень у реальному часі. Вони спрацьовують після виявлення незвичної діяльності. Оператори відеоспостереження встановлюють оповіщення на основі конкретних задалегідь визначених критеріїв.

### **Критерії оповіщення відеоспостереження**

Критерії програмного забезпечення для відеоаналітики включають:

– **Схожість зовнішнього вигляду предмета.** На основі цих критеріїв активуються сповіщення, наприклад, коли є предмети (велосипеди, автомобілі тощо) або люди, які не повинні перебувати в певному місці, або коли хтось живе в цьому місці. Сповіщення можна активувати навіть на основі критеріїв кольору одягу. Наприклад, якщо є закрита територія лише для приватних співробітників у формі, інші люди в одязі різних кольорів будуть відстежуватись і повідомлятися як підозрілі. Завдяки цій функції, якщо більше людей, ніж очікувалося, перебуває

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

або тиняється, наприклад, біля магазину, камера активує сповіщення, щоб персонал служби безпеки швидко реагував.

– **Сповіщення на основі кількості.** Сповіщення, які базуються на підрахунку кількості об'єктів у певних зонах, можуть допомогти контролювати трафік або натовп. Наприклад, якщо виявлені великі черги в магазині або на реєстраціях в аеропорту, менеджери можуть швидко вирішити цю проблему завдяки відеомоніторингу в реальному часі. Крім того, за допомогою функції виявлення присутності можна вимкнути світло, якщо в кімнаті нікого немає.

– **Сповіщення про розпізнавання обличчя.** Завдяки цифровим зображенням, отриманим із відеозаписів, співробітники служби безпеки можуть легко визначити підозрюваних у режимі реального часу.

## **Відеоналітика в реальному часі для інтелектуального відеоспостереження**

Давайте розглянемо випадок, коли наш клієнт, страхова компанія, зіткнувся з необхідністю зібрати більше даних про поведінку водіїв і дослідити середовище, в якому відбуваються аварії. Завданням команди програмного забезпечення було розробити хмарне IoT-рішення, яке б оптимізувало поліси страхування транспортних засобів. Фаза відкриття проекту показала більше перспектив для клієнта. Завдяки зібраним даним клієнт міг не тільки побачити аварії чи їх провісники, але й вжити запобіжних заходів.

### **Технічна архітектура**

#### **1. Дорожній агрегат**

Відеокамера розрізана в обох напрямках – до дороги та до водія. Він підключається до джерела живлення транспортного засобу та автомобільних аксесуарів для отримання інформації про автомобіль. Встановлений біля дзеркала заднього виду, він забезпечує хороший огляд дороги та водія. Разом з камерами йдуть мікрофон, динамік і слот для micro SD.

#### **2. Хмарний центр**

Як засіб зв'язку між хмарою і камерою використовувався модуль LTE. Він

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27



## 5. Візуалізація аналітики

Ефективність обладнання, що використовується для відеоспостереження, підвищується програмним забезпеченням, розробленим відповідно до потреб і вимог клієнта. У даному випадку зібрані дані служили багатьом цілям. До нього мали доступ три типи користувачів:

- Оператор центру моніторингу
- Власники автопарків
- Адміністратори клієнтів

Для зручності користування всім зацікавленим сторонам були надані інтерактивні панелі зі статистикою кожного водія, розділені за категоріями відео та карта.

### Чого вдалося досягти?

Завдання клієнта полягало в моніторингу поведінки водія та виявленні ризиків на дорозі. Такі дані безпосередньо вплинули на фінансові результати страхової компанії. Завдяки платформі відеомоніторингу та аналітики в реальному часі клієнт може легко мінімізувати необґрунтовані витрати. Крім того, як профілактичний захід можна запровадити та посилити інструктаж водіїв. Масштабованість програмного забезпечення дозволяє клієнту отримати більше від телематики та аналітики, і процес його вдосконалення триває.

### 5G і штучний інтелект: нова ера віддаленого відеоспостереження?

#### Віддалений відеоспостереження з 5G

Незважаючи на те, що мережі 4G вже успішно охопили більшість користувальницьких програмних рішень IoT, хвилювання щодо використання 5G у віддаленому відеоспостереженні в режимі реального часу не вщухає.

За прогнозами, різке зменшення затримки з поточних 50-100 мілісекунд у мережах 4G до 1-4 мілісекунд у 5G кардинально змінить швидкість реагування камери та моніторинг відео в реальному часі. Ще одна важлива перевага зв'язку 5G – це стрибок у якості зображення камер безпеки до підтримки відео 4K та 8K. Це забезпечить плавну та миттєву відеозйомку в прямому ефірі, даючи персоналу

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

служби безпеки можливість швидше ідентифікувати злочинців та/або реагувати на інциденти та надзвичайні ситуації.

Ще одна перевага, яку 5G може запропонувати для відеоспостереження, – широкі можливості підключення в містах або інших перевантажених місцях, де кілька користувачів намагаються підключитися одночасно. Це призводить до непробачних затримок у зменшенні ризиків. З 5G цю проблему вирішено, оскільки ця мережа забезпечує більш швидке та безперебійне підключення для більшої кількості пристроїв. Для порівняння ємності, 4G може вмістити 1 мільйон пристроїв на площі понад 500 квадратних кілометрів, а 5G підійде для такої ж кількості пристроїв в одній мережі на площі понад 1 квадратний кілометр.

Завдяки високій швидкості, наднизькій затримці, безпечному з'єднанню та розширеним можливостям мобільного потокового передавання записів відеоспостереження технологія 5G відкриє нові можливості для віддаленого моніторингу відео. Щоб дізнатися більше про мережі IoT, прочитайте цей посібник .

### **Відеоспостереження в реальному часі за допомогою ШІ**

На відміну від традиційних систем керування відео (VMS), які є хмарними рішеннями для відеоспостереження, технологія аналізу відеоконтенту на основі штучного інтелекту зробила величезний крок вперед завдяки використанню глибоких нейронних мереж (DNN). З розвитком глибокого навчання ідентифікувати конкретні об'єкти та розпізнавати певні шаблони стало легше, ніж будь-коли.

Системи відеоспостереження з підтримкою штучного інтелекту працюють на основі певних алгоритмів, які дозволяють аналізувати дані в режимі реального часу для перетворення їх у розумні та практичні висновки. Ці статистичні дані додатково агрегуються для представлення історичних діаграм, графіків, теплових карт. Головне, що всі процеси та аналізи виконуються без втручання людини.

#### **– Як працює моніторинг у реальному часі на основі ШІ?**

– Початкове дослідження питань виконується з багатьох точок зору для

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

визначення потенційних технологічних підходів .

- На основі дослідження інженери-техніки створюють і тестують алгоритми машинного навчання для ідентифікації людей і об'єктів у відеокадрах.
- ШІ-система навчена розпізнавати конкретні шаблони.
- Система розгорнута.

### **Безпечніше водіння з моніторингом дорожнього руху**

Система моніторингу дорожнього руху на основі IoT з автоматичним виявленням аварій.

### **Кейс з дослідження**

### **Переваги системи відеоспостереження в реальному часі**

Сьогодні технологія аналізу відеовмісту розвивається, і щоб підвищити рентабельність інвестицій, оптимізувати системи безпеки та покращити взаємодію з клієнтами, компаніям доводиться покращувати корисність мереж камер відеоспостереження та максимізувати цінність існуючої інфраструктури. Сповіщення в режимі реального часу, які запускаються вдосконаленим програмним забезпеченням для аналізу відеовмісту, допомагають покращити обізнаність про ситуацію та прискорити заходи щодо безпеки, реагування на надзвичайні ситуації, виявлення незвичайних дій, краще розуміння потреб клієнтів і багато іншого. Давайте класифікуємо основні переваги систем VCA, щоб ви могли зважити всі «за» і «проти» та зрозуміти, чи дійсно ваш бізнес потребує цього інноваційного підходу.

– **Високий рівень захисту.** Ця перевага стосується не лише громадської безпеки, а й моніторингу зон високого ризику. Поточе відео в режимі реального часу допомагає виявляти бездіяльність, ідентифікувати підозрілі об'єкти, які можуть вплинути на вашу організацію, або своєчасно забезпечувати безпеку в місцях, де є ризики аварій. Крім того, якщо ваш об'єкт містить небезпечні хімічні речовини, наприклад, відеоспостереження в режимі реального часу допомагає усунути ризик перебування людей поблизу. На відміну від звичайних підходів до відеоспостереження, відеоспостереження в реальному часі

за допомогою алгоритмів штучного інтелекту здатне відстежувати та виявляти кожен об'єкт, що з'являється. Завдяки вдосконаленій аналітиці відео ви можете створити структуровану базу даних, отриманих із необроблених відеоданих, що забезпечує інтелектуальне сповіщення, детальний пошук і ефективні звіти. У той час як звичайні відеокамери не можуть повністю охопити рухомі об'єкти через кольорові обмеження, великі відстані, погодні умови, аналіз відеоконтенту на основі нейронних мереж, навчених розпізнавати різні шаблони з масивних даних, забезпечує високу точність і надійність виявлення.

– **Прогнозне обслуговування.** У поєднанні з системами AI та IoT системи відеомоніторингу можуть передбачати проблеми ще до їх виникнення. Агрегування великих обсягів даних у вичерпну аналітику, відеомоніторинг у реальному часі за допомогою датчиків штучного інтелекту та Інтернету речей полегшує стратегії превентивної безпеки. Таким чином, використовуються можливості швидкого пошуку, щоб ініціювати сповіщення про тривогу в режимі реального часу, щоб співробітники служби безпеки могли вчасно поінформувати про будь-яку тривожну ситуацію. І тут ми маємо на увазі не інтелектуальну обробку зображень, а інтеграцію аналізу відео з іншими системами, такими як контроль доступу, відстеження часу тощо. Шаблони на основі ШІ також відіграють важливу роль у аналізі поведінки.

– **Продажі та маркетингові можливості.** Виходячи за рамки традиційної парадигми систем відеонагляду, які генерують відзнятий матеріал, системи відеоспостереження в реальному часі дозволяють покращити досвід клієнтів. З огляду на це, ми маємо на увазі підприємства роздрібною торгівлі, які можуть отримати значну вигоду від передових відеотехнологій, покращуючи досвід своїх клієнтів. За допомогою глибокого аналізу відео ви можете отримати значущі теплові дані, наприклад, найбільш зайняті місця в магазині, де ваші клієнти найчастіше зупиняються та навіть куди вони дивляться. Існують різноманітні переваги, які допомагають приймати маркетингові рішення та підвищувати рентабельність інвестицій.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

## Варіанти використання віддаленого відеоспостереження

### Відеоспостереження в реальному часі для розумних міст

У розумних містах є багато територій, де відеоспостереження в реальному часі може допомогти підвищити ефективність і ефективне управління, а також посилити профілактичні заходи. Давайте розглянемо деякі приклади:

– Розумні системи відеоспостереження контролюють дорожню мережу та людей, які пересуваються містом. Завдяки відеоаналітиці такі системи можуть не лише ідентифікувати ДТП, а й негайно сповіщати швидку допомогу чи поліцію. Крім того, на основі інтелектуальної аналітики даних можна закривати дороги та коригувати маршрути, щоб уникнути проблем із заторами. Інтелектуальна система моніторингу дорожнього руху, підключена до датчиків камери, збирає дані. Така система моніторингу може автоматично виявляти інциденти на основі отриманих даних від датчиків. Одним із прикладів є програмне забезпечення для моніторингу дорожнього руху, яке було впроваджено компанією Euristici у Дублінському портовому тунелі. Система генерує на карті різні типи сповіщень і глибоку аналітику в режимі реального часу. За допомогою такої відеоаналітики ви можете бачити середню швидкість і час у дорозі транспортного засобу за певний день, середній час очікування в заторах, рух пішоходів тощо. На основі цих звітів системи IoT генерують відповідні сповіщення, щоб негайно реагувати на проблеми які сприяють загальному процвітанню міста.

– Відеомоніторинг у режимі реального часу, інтегрований з IoT, дає змогу розумним містам ефективно керувати паркуванням. Водії можуть легко знаходити вільні паркувальні місця за допомогою додатків на своїх смартфонах, а несанкціоновані транспортні засоби, припарковані в заздалегідь визначених зонах, можна легко виявити.

– Управління громадськими місцями також можна покращити завдяки відеоаналітиці, яка виявляє черги на вході. Дистанційне відеоспостереження відіграє важливу роль у вирішенні таких проблем і сприяє регулюванню людських потоків.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

– 3 точки зору сталого розвитку розумні системи моніторингу в реальному часі та відеоаналітика можуть допомогти міській владі зменшити забруднення, оптимізувати споживання енергії та запобігти кліматичним кризам. Наприклад, відеоаналітика та датчики на основі Інтернету речей можуть виявляти й позначати потенційне забруднення води або покращувати потік транспорту для захисту здоров'я громадян. Розумні рішення для міського освітлення можуть надзвичайно оптимізувати споживання енергії та зменшити витрати на обслуговування. Наша команда впровадила рішення для освітлення розумного міста для датської міжнародної компанії Novalume, що призвело до зниження витрат на обслуговування та викидів CO<sub>2</sub> до 75%.

### **Компоненти системи живого відеоспостереження дорожнього руху**

#### **Розпізнавання обличчя за допомогою дистанційного моніторингу**

Будучи біометричною технологією, розпізнавання обличчя може точно ідентифікувати людей шляхом аналізу їхніх обличчя на основі попередньо визначених шаблонів. Відскановане зображення з відео порівнюється із зображеннями в базі даних і на основі певних ознак зіставляється. Крім того, розпізнавання обличчя допомагає розпізнавати набагато більш конкретні речі про обличчя: стать, вік, напрямок очей тощо.

Окрім програм громадської безпеки та безпеки, де технологія розпізнавання обличчя переважно використовується для боротьби зі злочинністю, вона також застосовується в інших галузях для виконання конкретних завдань.

**В аеропортах**, наприклад, розпізнавання обличчя можна використовувати для перевірки особи мандрівників на митниці. Очікується, що через пандемію COVID-19 системи біометричного сканування, включаючи ідентифікацію обличчя, будуть широко використовуватися як безконтактні взаємодії. Такі рішення вже доступні в аеропортах США, де пасажери можуть використовувати сканування обличчя під час реєстрації, щоб сісти на літак. Крім того, технологію розпізнавання обличчя можна застосовувати в аеропортах для запобігання терористичним нападам.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

**Індустрія гостинності** – це сфера, де розпізнавання обличчя може значно вплинути на ефективне обслуговування клієнтів і надійні системи безпеки. Окреслимо основні випадки використання ідентифікації обличчя в готелях:

– З метою безпеки та доступу можна використовувати сканування обличчя, щоб надати гостям готелю доступ до їхніх номерів і ідентифікувати проблемних гостей, яких уже видалили з приміщення.

– Клієнти можуть авторизувати свої платежі за допомогою розпізнавання обличчя без необхідності взаємодіяти з одним співробітником готелю.

– Завдяки моніторингу в реальному часі за допомогою сканування обличчя власники готелів можуть досліджувати своїх гостей і збирати дані про їхні вирази обличчя, риси обличчя, щоб визначити вік і стать тощо.

– Навіть перед реєстрацією клієнта розпізнавання обличчя може допомогти співробітникам готелю надати вищий рівень підтримки клієнтів і, наприклад, більш персоналізовані привітання.

**Медичні установи та лікарні** також можуть отримати вигоду від розпізнавання обличчя. Це пов'язано з охороною психічного здоров'я, де аналітика відеовмісту аналізує риси обличчя, вирази, пози тіла та погляд, щоб визначити захворювання. Наприклад, Національний дослідницький інститут геному людини використовує програмне забезпечення для розпізнавання обличчя для діагностики рідкісного генетичного захворювання під назвою синдром ДіДжорджа, яке вражає дітей в Африці, Азії та Латинській Америці.

### **Відеомоніторинг у реальному часі в розумному фермерстві**

Завдяки системам моніторингу з цифрових камер фермери можуть отримувати аналітику в реальному часі та відстежувати кожну операцію ефективним і гнучким способом. Фермери вже почали використовувати технології на основі Інтернету речей, щоб зробити свої сільськогосподарські процеси більш складними, керованими даними та оптимізованими. Завдяки вдосконаленій аналітиці та системам віддаленого моніторингу підключене землеробство підвищує потенціал фермерів . Давайте визначимо переваги, які

										<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата							35

можуть отримати фермери:

– Збір даних у режимі реального часу за допомогою відеокамер допомагає відстежувати ефективність обладнання.

– Додатки для моніторингу в реальному часі дають фермерам більше контролю, оскільки вони можуть виявляти несправності та негайно їх усувати. Це, у свою чергу, дозволяє аграріям уникнути зупинок і великих витрат на ремонт активів.

– Фермери можуть значно заощадити час завдяки постійному доступу до своїх додатків IoT, а віддалене відеоспостереження сприяє безпечному управлінню землеробством. Немає необхідності брати участь у всіх сільськогосподарських процесах, оскільки віддалений моніторинг допомагає забезпечити безпеку та комфорт.

– Використовуючи бездротові технології та відеоаналітику, фермери мають змогу отримувати точні аналітичні дані для коригування харчування тварин і загального стану здоров'я, а також використовувати інтелектуальне обприскування в сільському господарстві для зменшення використання води та пестицидів. Крім того, завдяки системам відеоспостереження ви завжди будете в курсі місцезнаходження худоби.

### **Відеоспостереження в режимі реального часу в розумній роздрібній торгівлі**

За даними дослідження Oracle, 66% компаній використовують технології IoT для полегшення взаємодії з клієнтами. Алгоритми машинного навчання та передова відеоаналітика стали проривними технологічними трендами в секторі роздрібною торгівлі за останні роки.

Визначення ваших клієнтів і визначення їхніх уподобань і моделей поведінки може значно сприяти вашій загальній стратегії маркетингу та продажів. Програмне забезпечення для відеоаналітики допомагає генерувати практичну інформацію щодо характеристик клієнтів, статі, віку та їхньої поведінки. Ви навіть можете визначити, скільки часу ваш клієнт витратив на

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

перегляд певного продукту, проаналізувавши напрямок його погляду. Уся ця інформація може вплинути на ваші стратегії прийняття рішень, щоб ви могли максимізувати продажі та покращити взаємодію з клієнтами.

### Компоненти внутрішньобудинкової системи відеоспостереження

Віддалений відеоспостереження може стати вашим безпрограшним рішенням, якщо ви шукаєте відповіді на ці запитання:

- Хто ваші клієнти?
- Коли вони відвідують ваш магазин?
- Як вони себе поведуть?
- Де найкраще розмістити речі?
- тощо

Завдяки прогресу в глибокому навчанні та прогнозній аналітиці віддалене відеоспостереження стало вирішальним двигуном для прискорення продуктивності бізнесу. Завдяки можливості надавати деталізовані дані, які можуть стимулювати прийняття потужних рішень і забезпечувати новий рівень безпеки, системи моніторингу в реальному часі можуть виявляти аномалії, точно ідентифікувати людей та їхню поведінку, покращувати взаємодію з клієнтами та, нарешті, підвищувати рентабельність інвестицій.



Рисунок 3.1 – Структурна схема системи

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

### 3.3 Розробка функціональної схеми

Загальна схема взаємодії функціональних компонентів типового програмного рішення представлена на рисунку 3.2. Типове рішення програмного забезпечення складається з наступних компонентів:

- система зберігання;
- система балансування навантаження;
- моніторинг;
- статистика;
- серверна частина системи керування;
- система прийому потоків;
- система прямої трансляції;
- система кешування;
- серверна частина системи відновлення.

Система зберігання розділяє дані, отримані від системи прийому потоків, на фрагменти фіксованої тривалості й зберігає отримані файли в архів, забезпечуючи подальший довільний доступ до будь-якого тимчасового відрізка в архіві.

Система балансування навантаження забезпечує розподіл по серверах системи кешування запитів, що надходять від користувачів мережі Інтернет. Це дозволяє оптимізувати завантаження мережі й збільшити швидкість доступу до найбільш затребуваних відеоданих.

Система прийому потоків забезпечує безперервний прийом аудіо й відеопотоків від пристроїв відеоспостереження й передачу даних системі прямої трансляції й системі зберігання.

Система прямої трансляції забезпечує передачу відеоданих користувачам мережі Інтернет. При цьому формуються й передаються користувачам мережі Інтернет наступні переліки:

- доступні варіанти якості відеоданих;

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38



### 3.4 Розробка діаграми процесів

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. 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.

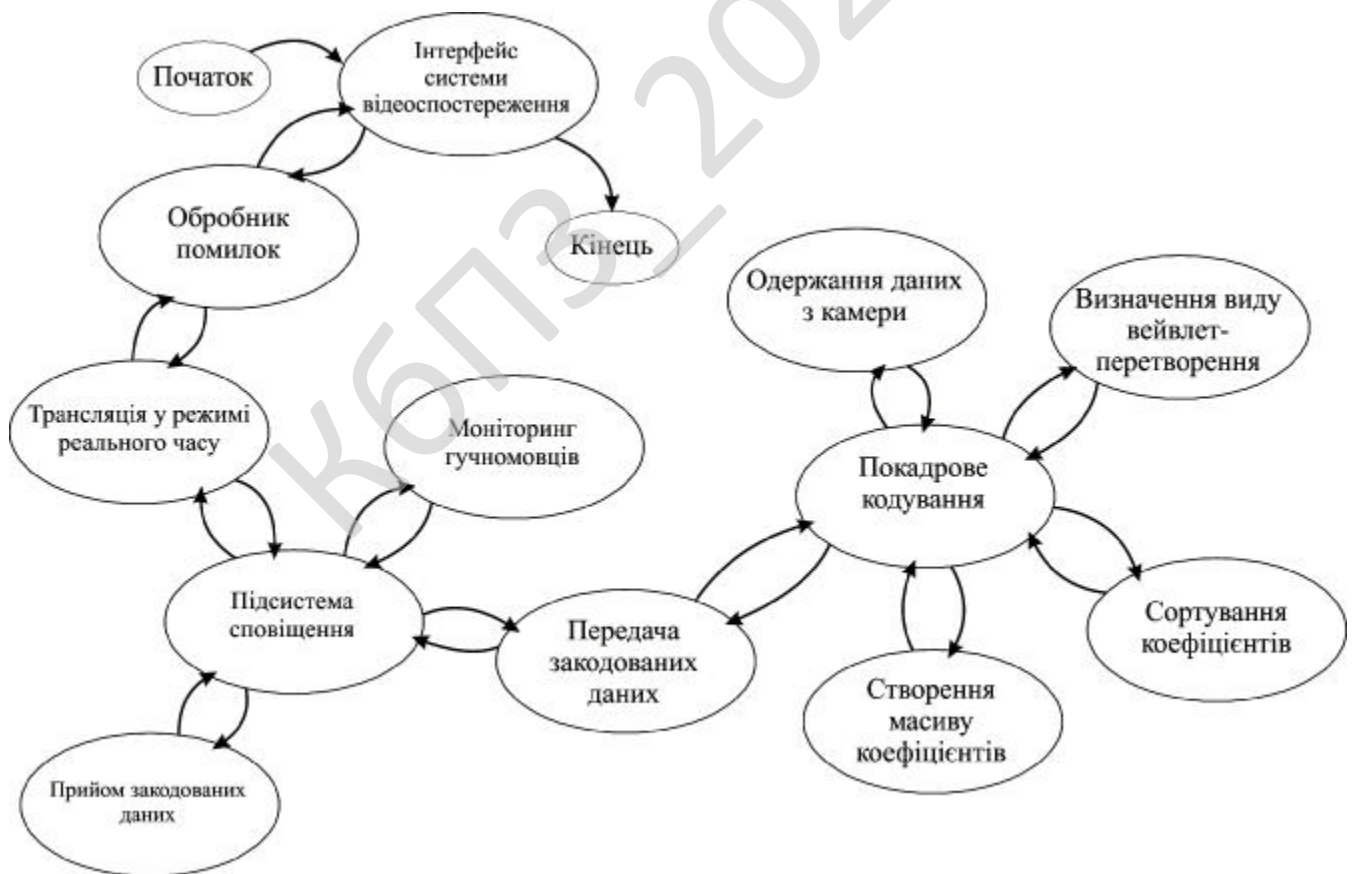


Рисунок 3.3 – Діаграма взаємодії процесів

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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.

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.

КБПЗ\_2024

					VKPM-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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

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

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю керування гучномовцями в системах відеоспостереження.

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

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 – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

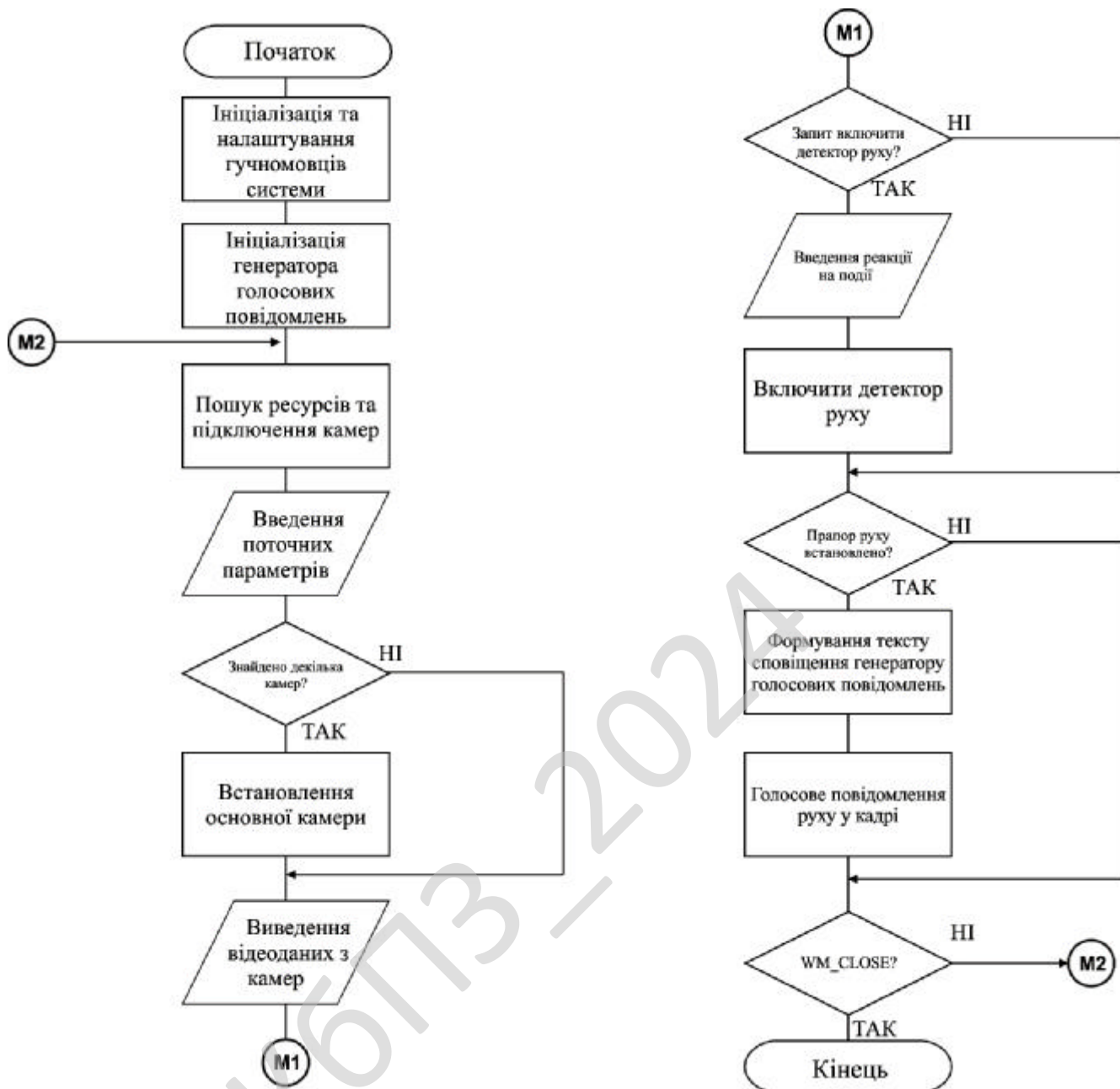


Рисунок 4.1 – Блок-схема основної програми

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, названої UML-моделлю.

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

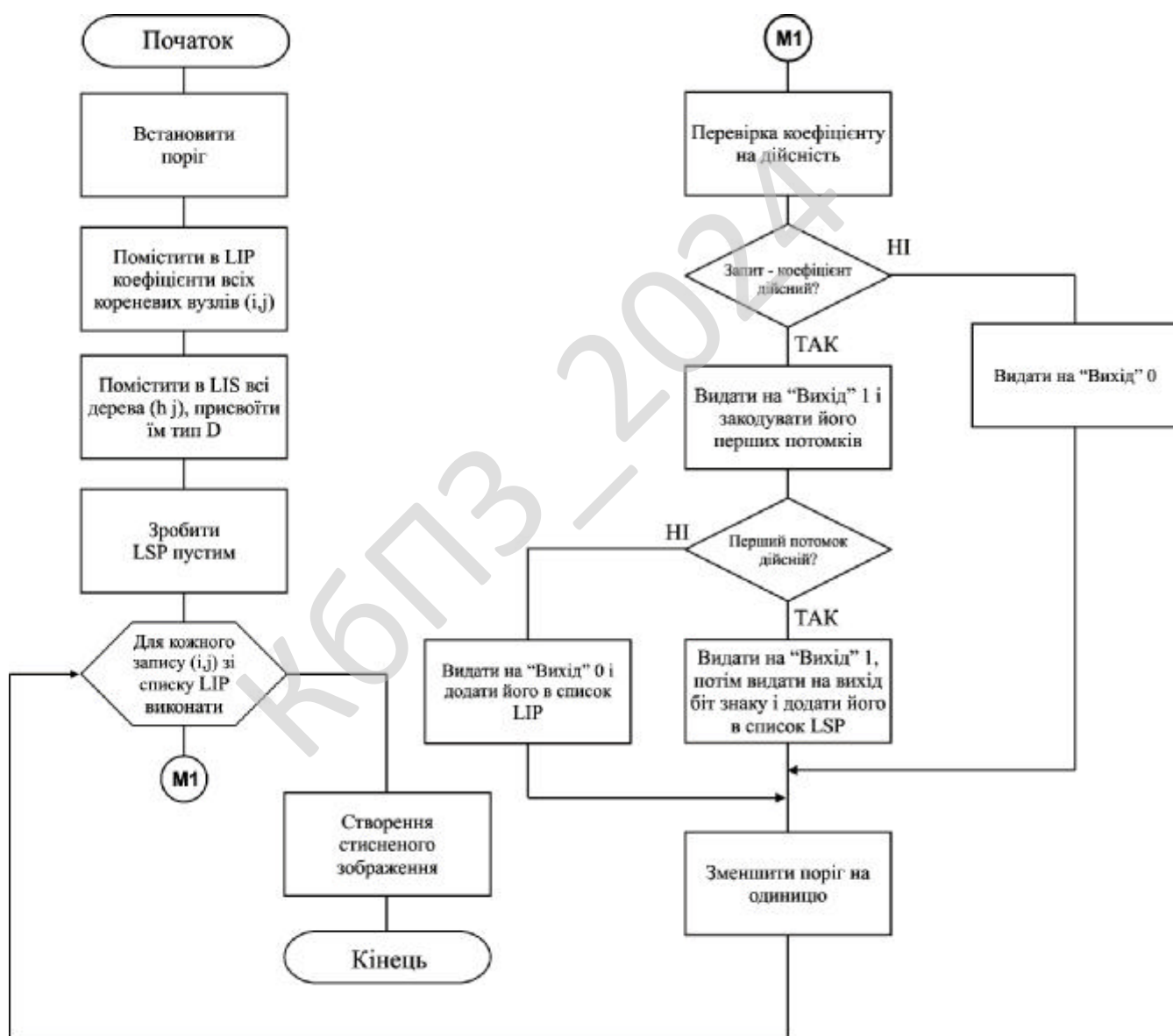


Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

**Подійно-орієнтована архітектура** (Event-driven architecture, надалі **EDA**) – шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з "на продаж" до "продано". Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим програмам даної архітектури.

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

Події не подорожують, вони просто відбуваються. Проте термін подія часто використовується метонімічно для позначення самого нотифікаційного повідомлення, що може призвести до певної плутанини.

Цей архітектурний шаблон може застосовуватися при проектуванні і реалізації ПЗ і систем, які передають події між слабкозв'язаними компонентами програмного забезпечення і сервісами (службами).

Подійно-орієнтована система як правило складається з емітерів подій (або агентів) і споживачів подій (або стоків).

Стоки несуть відповідальність за здійснення реагування на появу події. Реакція не завжди може бути повністю забезпечена самим стоком. Наприклад, стік, може бути відповідальним лише за фільтрацію, трансформацію і відправку події до іншого компонента або він може забезпечити повністю самостійну реакцію на таку подію. Перша категорія стоків може бути заснована на традиційних компонентах, таких як проміжне програмне забезпечення, орієнтоване на обробку повідомлень (message oriented middleware, MOM), в той

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

час, як друга категорія стоків (самостійна реакція в режимі он-лайн) може вимагати більш придатної платформи (фреймворку) для виконання транзакцій.

Розробка ПЗ і систем в подійно-орієнтованій архітектурі дозволяє їм бути сконструйованими способом, який більш відповідає вимогам до їх створення, оскільки такі системи в більшій мірі пристосовуються до непередбачуваних і асинхронних середовищ.

Подійно-орієнтована архітектура (EDA) може доповнювати сервісно-орієнтовану архітектуру (SOA), оскільки сервіси (служби) можуть бути активовані тригерами, які ініціюються при настанні подій.

Ця парадигма особливо корисна, коли стік не забезпечує власного виконання будь-яких дій.

Подійно-орієнтована SOA (SOA-2) розвиває архітектури SOA і EDA для забезпечення більш глибокого і надійного рівня сервісів за рахунок використання раніше невідомих причинно-наслідкових зв'язків, щоб сформувати новий шаблон подій. Цей новий шаблон бізнес-аналітики дає поштовх до небаченого раніше зростання рівня автоматизації підприємства за рахунок привнесення додаткової цінної інформації в описану раніше модель діяльності.

Обчислювальна техніка та сенсорні пристрої (сенсори, датчики, контролери) можуть виявляти зміни стану об'єктів або умов і створювати події, які потім можуть бути оброблені сервісом (службою) або системою.

Подія може складатися з двох частин: заголовка події та тіла події. Заголовок події може включати в себе інформацію таку як, наприклад, назва події, часова мітка події і тип події. Тіло події – це частина, яка описує факт, що стався в дійсності. Тіло події не слід плутати з шаблоном або логікою, яка може бути застосована як реакція на саму подію.

Архітектура, керована подіями, складається з чотирьох логічних рівнів (шарів). Вона починається з виявлення факту, його технічного подання у формі події і закінчується не пустою множиною реакцій на цю подію.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

## Генератор подій

Першим логічним шаром є генератор подій, який виявляє факт і представляє цей факт подією. Оскільки фактом може бути практично все, що може бути сприйнято, то ним може бути і генератор подій. Наприклад, генератором може бути клієнт електронної пошти, система електронної комерції або певний тип датчика.

Перетворення різних даних, отриманих від датчиків, в єдину стандартизовану форму даних, які можуть бути оцінені, є основною проблемою при розробці та реалізації цього шару. Однак, враховуючи, що подія є строго декларативною, можна легко застосовувати будь-які операції трансформації, тим самим усуваючи необхідність забезпечення високого рівня стандартизації.

## Канал подій

Канал подій – це механізм, через який інформація від генератора подій передається до обробника подій (event engine) або стоку.

Це може бути з'єднання TCP/IP або вхідний файл будь-якого типу (простий текст, формат XML, e-mail тощо). В один і той же час може бути відкрито кілька каналів подій. Як правило, оскільки обробник подій повинен працювати в режимі, наближеному до реального часу, канали подій зчитуються асинхронно. Події зберігаються в черзі, очікуючи наступної обробки механізмом обробки подій.

## Механізм обробки подій

Механізм обробки подій (event processing engine) є місцем, де подія ідентифікується і вибирається відповідна реакція на нього, яка потім виконується. Це також може призвести до породження ряду тверджень. Якщо подія, яка надійшла до механізму обробки подій, є наприклад такою «Запаси продукту ID досягли нижнього допустимого рівня», це може ініціювати, наприклад, такі реакції як «Замовити продукт ID» і «Сповістити персонал».

## Наступна подійно-орієнтована дія (післядія)

Щодо того, як можуть проявлятися наслідки події, слід відмітити, що вони можуть проявитись багатьма різними способами і у різноманітних формах

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

(наприклад, повідомлення електронної пошти, надіслане комусь, або ПЗ, що виводить деяке попередження на екран). Залежно від рівня автоматизації, який забезпечується стоком (механізмом обробки подій), ці дії можуть виявитись зайвими.

Є три основні стилі обробки подій: простий, потоковий і складний. Часто ці три стилі використовуються спільно у розвинутій подійно-орієнтованій архітектурі.

### **Проста обробка подій**

Проста обробка подій стосується подій, які безпосередньо належать до специфічних вимірних змін умов. У випадку простої обробки подій, мають справу з появою відомих подій, що ініціюють післядію (післядії). Проста обробка подій зазвичай використовується для управління потоком робіт в реальному часі, скорочуючи тим самим час затримки і вартість робіт.

Наприклад, прості події можуть створюватись (породжуватись) датчиком, що виявляє зміну тиску в шині або температуру навколишнього середовища.

### **Обробка потоку подій**

При обробці потоку подій (event stream processing, далі ESP) відбуваються як звичайні, так і відомі події. Звичайні події (заявки, передачі RFID) перевіряються на те, чи є вони відомими, і передаються інформаційним передплатникам. Обробка потоку подій зазвичай використовується для управління потоком інформації в реальному часі і на рівні підприємства, що дозволяє своєчасно приймати рішення.

### **Обробка складних подій**

Обробка складних подій (Complex event processing (CEP)) дозволяє за шаблонами простих і звичайних подій проводити аналіз того, чи наступила складна подія. Обробка складних подій полягає в оцінюванні взаємного впливу подій і в наступному виконанні дій. При цьому, типи подій (відомих або звичайних) можуть перетинатись, а події можуть виникати протягом тривалого періоду часу.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Кореляція подій може бути причинною, тимчасовою або просторовою. СЕР вимагає використання складних інтерпретаторів подій, визначення і підбору шаблонів подій, а також відповідних кореляційних методів. Обробка складних подій зазвичай використовується для виявлення і реагування на аномальну поведінку, загрози і можливості у бізнесі.

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

### **Етапи ризик-менеджменту**

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

- на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;
- на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;
- на третьому етапі вибираються методи і інструменти управління виявленим ризиком;
- на четвертому етапі проводиться безпосереднє управління ризиком;
- на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

### **Методи і інструментарій ризик-менеджменту**

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів.
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Була використана водоспадна (каскадна) модель життєвого циклу ПЗ (waterfall model) – послідовний метод розробки програмного забезпечення, названий так через діаграму схожу на водоспад.

Ця модель розробки запозичена з системної інженерії у виробництві та будівництві – областях, в яких зміни на пізніх етапах дуже дорогі, або неможливі. Наприклад, для створення складних інженерних конструкцій (споруд, літаків, мостів і т.п.). Зміни в проекті фундаменту будинку після того, як покладений дах коштують дуже дорого, тому перфекціонізм на початкових етапах проектування просто необхідний. Інженери, які починали займатись розробкою програмного забезпечення перейшовши з інших галузей, просто адаптували звичну модель, тому що на ранніх етапах розвитку комп'ютерної техніки не було методологій створених саме для програмування. Проте, схожі методології застосовуються для програмного забезпечення й далі, у випадках коли вимоги фіксовані, і вимагається висока якість та надійність, наприклад в системах для військових чи медичних потреб.

Перший формальний опис водоспадної моделі, після якої вона стала популярною був здійснений В. В. Ройсом у 1970. Попри те, що стаття містить переважно критику методу, на неї часто посилаються.

Переваги методу:

- Ніяких переробок.
- Гарна специфікація перетікає в гарну документацію.
- Зрозуміла модель.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

– Розробники можуть мати низьку кваліфікацію.

Недоліки:

– Необхідний перфекціонізм на кожному етапі.

– Важко вносити зміни (якщо взагалі можливо).

– Надлишкове проектування.

– Поділ розробників на "perfect" та "code monkeys".

Модифікації. Через те що цей метод погано підходить для розробки саме ПЗ, частіше використовують його модифікації.

Найвідоміша модифікація – Sashimi. Названа так через японську страву сашімі (суші нарізане і сервіроване так, що складені рядочком шматочки накладаються один на одного). В моделі розробки Сашімі фази життєвого циклу йдуть одна за одною, але при цьому перекриваються одна з одною в часі.

#### 4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). Одним із основних алгоритмів симетричного блокового шифрування, що використовуються в Україні, є ДСТУ 7624:2014 («Калина»), який визначає сучасний алгоритм симетричного блокового перетворення для забезпечення конфіденційності й цілісності інформації при її обробці та встановлює режими його роботи. В алгоритмі шифрування даних «Калина» використовуються криптографічні перетворення, які відповідають сучасним вимогам до рівня криптостійкості та швидкодії.

Даний стандарт розроблено з урахуванням існуючих та потенційних загроз, подальшого інтенсивного розвитку інформаційних технологій та необхідності активного використання протягом кількох наступних десятиліть.

Стандарт блокового симетричного шифрування ДСТУ 7624:2014 визначає десять різних режимів роботи, що широко поширені відповідно до міжнародних стандартів ISO/IEC 10116:2006.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Це спрямовано на забезпечення широкого застосування ДСТУ 7624:2014, у тому числі для захисту інформації, що передається комп'ютерними мережами, прозорого шифрування жорстких дисків і змінних носіїв, електронних документів, ключових даних.

Ефективність реалізації систем, засобів та протоколів криптографічного захисту інформації в інформаційно-телекомунікаційних системах різного призначення може бути забезпечена саме наявністю такої кількості режимів роботи алгоритму.

До блокового шифру «Калина» ставляться такі вимоги: високий рівень криптографічної стійкості з достатнім запасом у разі появи нових атак протягом тривалого часу; висока швидкодія програмної реалізації на сучасних та перспективних платформах; компактність програмної та програмно-апаратної реалізації; можливість ефективної інтеграції декількох алгоритмів в одному засобі криптографічного захисту; прозорість проектування, консервативний підхід до забезпечення стійкості; вища (або однакова) ефективність порівняно з найкращими світовими рішеннями.

Криптографічні алгоритми, які визначаються стандартами ДСТУ 7624:2014 і ДСТУ 7564:2014, є гнучкими, підтримують розмір блоку і довжину ключа від 128 до 512 біт.

Стандарт симетричного блокового шифрування «Калина» є результатом багаторічної плідної співпраці Державної служби спеціального зв'язку та захисту інформації України та провідних українських вчених.

Даний алгоритм шифрування враховує досвід і результати проведення міжнародних та відкритих національних конкурсів криптографічних алгоритмів.

Алгоритм ДСТУ 7624:2014 забезпечує досить високий рівень криптостійкості порівняно з міжнародним стандартом AES (ISO/IEC 18033-3:2010), оскільки дає можливість застосовувати блок даних і ключ шифрування розміром аж до 512 біт. Крім того, він має аналогічну або навіть більш високу швидкодію на сучасних і перспективних програмних та програмно-апаратних

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

платформах. На даний момент продовжуються роботи зі стандартизації вітчизняних криптографічних алгоритмів та протоколів. При цьому не обмежується застосування гармонізованих стандартів у сфері захисту конфіденційної інформації. Також зусилля зосереджено на використанні кращих практик застосування стандартів шифрування даних для захисту інформації в інформаційно-комунікаційних системах [10, 11].

Оскільки в стандартах симетричного блокового шифрування «Калина» та AES використовуються аналогічні криптографічні перетворення, на наш погляд, буде доцільним порівняти ці два алгоритми.

Основними відмінностями «Калина» від «Rijndael» (AES) є: збільшена кількість циклів шифрування (запас стійкості); використання додавання за модулем 264 і за модулем 2 для введення ключової інформації (захист від алгебричних атак, лінійного та диференціального криптоаналізів, інтерполяційної атаки тощо); використання чотирьох блоків нелінійного перетворення (S-блоків) замість одного (додатковий захист від алгебричних атак, поліпшення властивостей розсіювання алгоритму – покращені статистичні властивості, відповідно, більш високий рівень стійкості до диференціального та лінійного криптоаналізів тощо); використання випадково сформованих чотирьох блоків, відібраних критеріями стійкості до диференціального, лінійного криптоаналізів, ступені нелінійності булевих функцій (на відміну від S-блоку Rijndael/Camellia та інших шифрів, що використовують звернення в полі та, відповідно, квадратичні залежності між входом і виходом, – захист від алгебричних атак); принципово нова схема створення підключів (захист від усіх відомих атак на схеми створення підключів); досить висока продуктивність; можливість відновлення сеансового ключа за окремим підключем (додатковий захист від атак, що виконують відновлення підключів).

Усі поліпшення спрямовані на збільшення стійкості та запобігання потенційним вразливостям відносно Rijndael, виявленим в останні роки [12].

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ керування гучномовцями в системах відеоспостереження яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Дані; Гучномовці; Налаштування; Довідка.
- Розділу виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

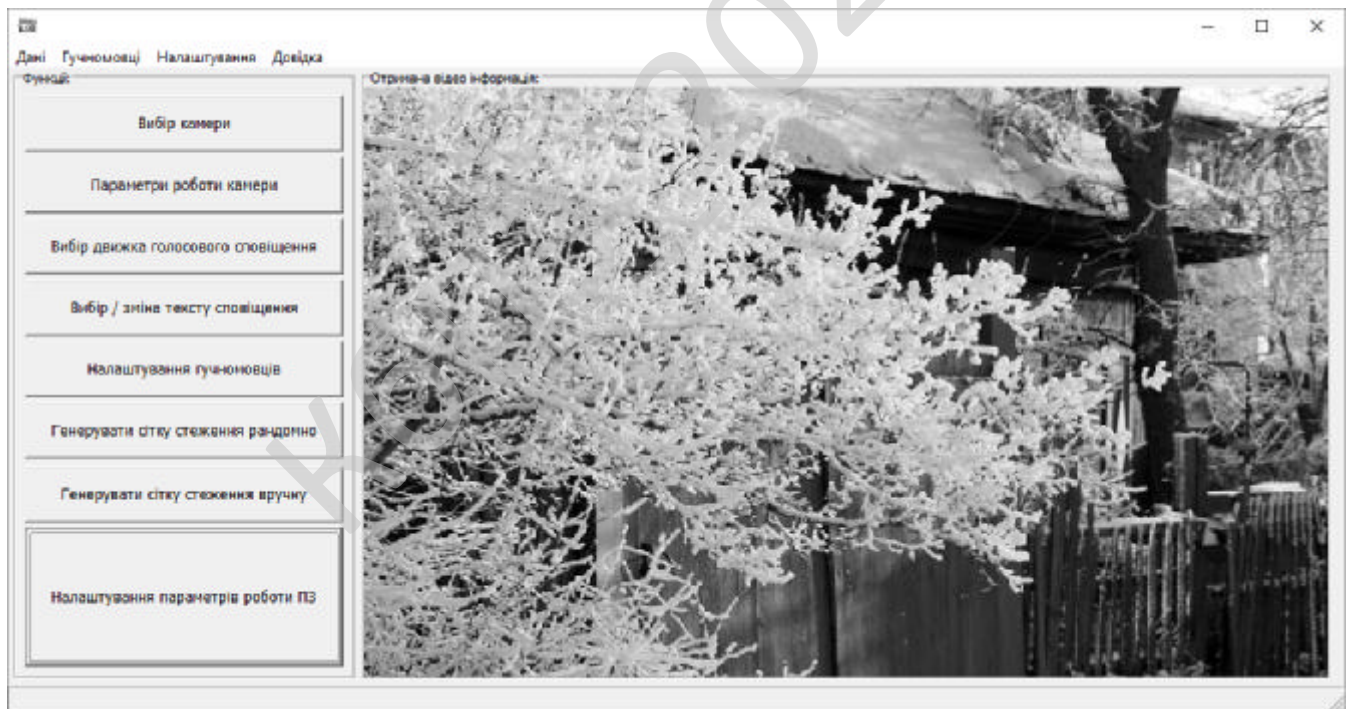


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

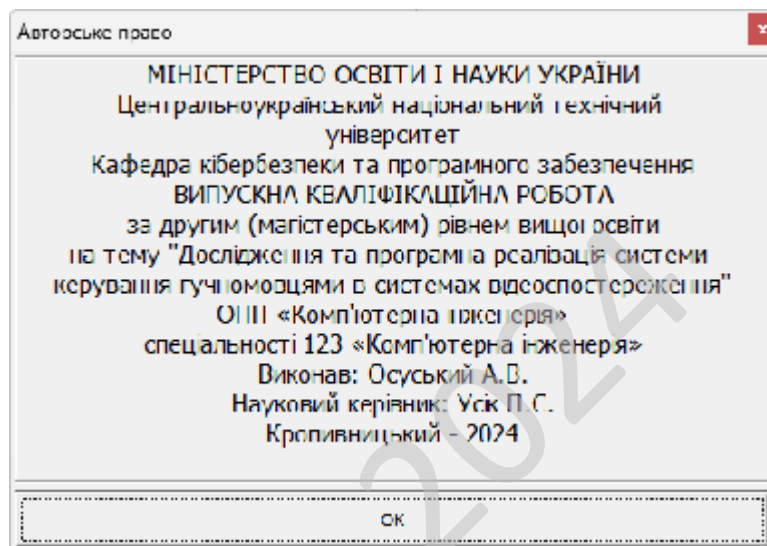


Рисунок 5.2 – Авторське право

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

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

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

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

- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

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

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 – proprietary software.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На протигагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування гучномовцями в системах відеоспостереження.

*Метою розробки є дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження.*

*Об'єктом дослідження є процес керування гучномовцями в системах відеоспостереження.*

*Предметом дослідження є методи керування гучномовцями в системах відеоспостереження.*

*Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.*

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

– Удосконалено метод керування гучномовцями в системах відеоспостереження.

– Розроблено вітчизняний продукт керування гучномовцями в системах відеоспостереження, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи керування гучномовцями в системах відеоспостереження можуть бути цікавими для різних груп користувачів, організацій і секторів(рисунок 7.1).

1. Телекомунікаційні та інтеграційні компанії
2. Охоронні компанії та служби безпеки
3. Підприємства та організації з високими вимогами до безпеки
4. Органи державного управління та місцеві влади
5. Постачальники програмного забезпечення та розробники
6. Вендори системи управління безпекою
7. Навчальні заклади та науково-дослідні установи
8. Клієнти в сфері розваг та громадських заходів

Рисунок 7.1 – Цільова аудиторія

Отже, результати дослідження та програмної реалізації таких систем можуть зацікавити різні організації, що займаються безпекою, комунікацією, а також розробниками технологій для відеоспостереження та інших подібних систем.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи керування гучномовцями в системах відеоспостереження за допомогою методів експертних оцінок можна застосувати кілька підходів, серед яких широко використовуються методи оцінки, засновані на експертних групах.:

Перше, що потрібно зробити – це визначити ключові критерії, за якими оцінюватиметься привабливість цієї системи. Вони можуть включати (рисунок 7.2):

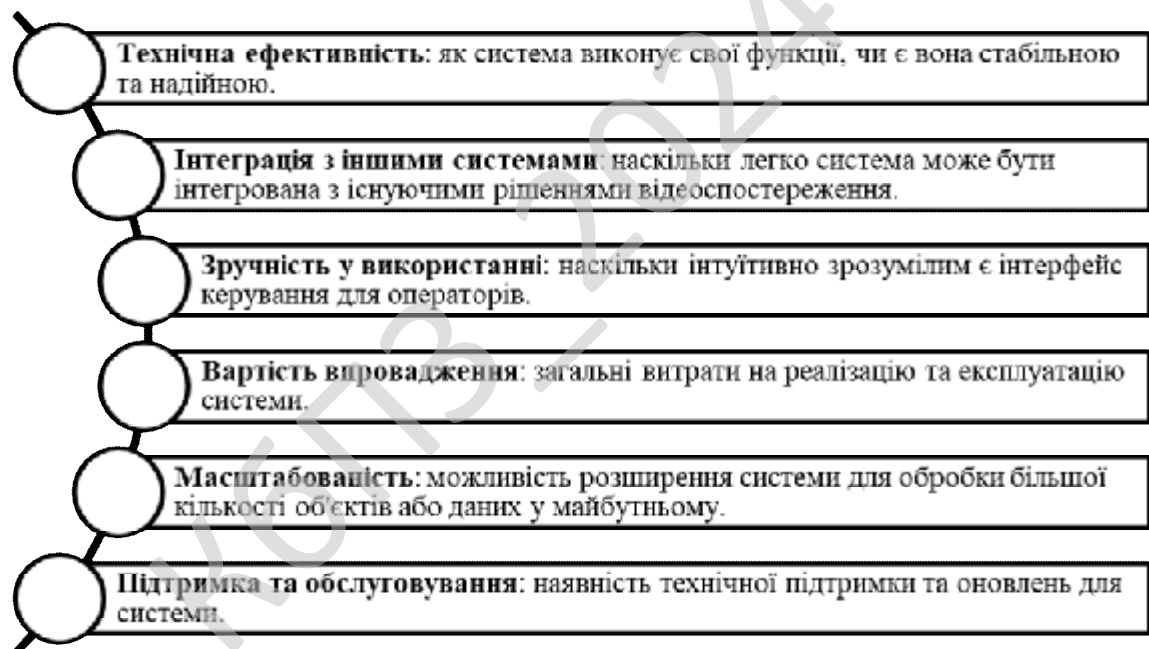


Рисунок 7.2 – Критерії експертної оцінки

Експертна оцінка може здійснюватися за шкалою, наприклад, від 1 до 5, де: 1 – дуже погано, 2 – погано, 3 – середньо, 4 – добре, 5 – дуже добре

Далі відбувається формування групи експертів, до складу якої входять фахівці в різних сферах, таких як: технічні фахівці (для оцінки технічної ефективності і сумісності), менеджери з впровадження та обслуговування (для

оцінки вартості та зручності використання), представники бізнесу та кінцеві користувачі (для оцінки комерційної привабливості).

Кожен експерт оцінює кожен з критеріїв за визначеною шкалою (наприклад, від 1 до 5) та надає пояснення щодо своєї оцінки. Після цього отримані оцінки можна узагальнити, щоб отримати загальний рейтинг системи (таблиця 7.1).

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середній бал
Технічна ефективність	5	4	5	4.67
Інтеграція з іншими системами	4	4	5	4.33
Зручність у використанні	3	4	4	3.67
Вартість впровадження	3	4	3	3.33
Масштабованість	5	5	5	5.00

На основі оцінок експертів, можна підсумувати результати для кожного критерію. Середній бал для кожного критерію дозволяє отримати загальну картину привабливості системи. За результатами можна вирішити, чи варто продовжувати впровадження цієї системи, чи потрібно вдосконалити окремі аспекти. В нашому випадку система набрала високі бали за технічну ефективність та масштабованість, що є великими перевагами. Однак, варто звернути увагу на вартість впровадження та зручність у використанні, які отримали середні оцінки, що може вказувати на необхідність удосконалення цих аспектів.

На основі отриманих результатів експертної оцінки можна сформулювати рекомендації для подальшого розвитку проєкту: покращення інтерфейсу користувача (для зручності використання), оцінка вартості (детальніше проаналізувати варіанти зменшення витрат на реалізацію системи), покращення

технічної підтримки (забезпечення надійної та швидкої підтримки для користувачів).

Такий метод оцінки дозволяє об'єктивно оцінити привабливість системи і прийняти обґрунтовані рішення щодо її впровадження та подальшого розвитку.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи керування гучномовцями в системах відеоспостереження можна використати кілька методів, залежно від вимог до точності оцінки та етапу розробки проекту.

Пропонуємо розглянути два з них, які можуть бути максимально корисними:

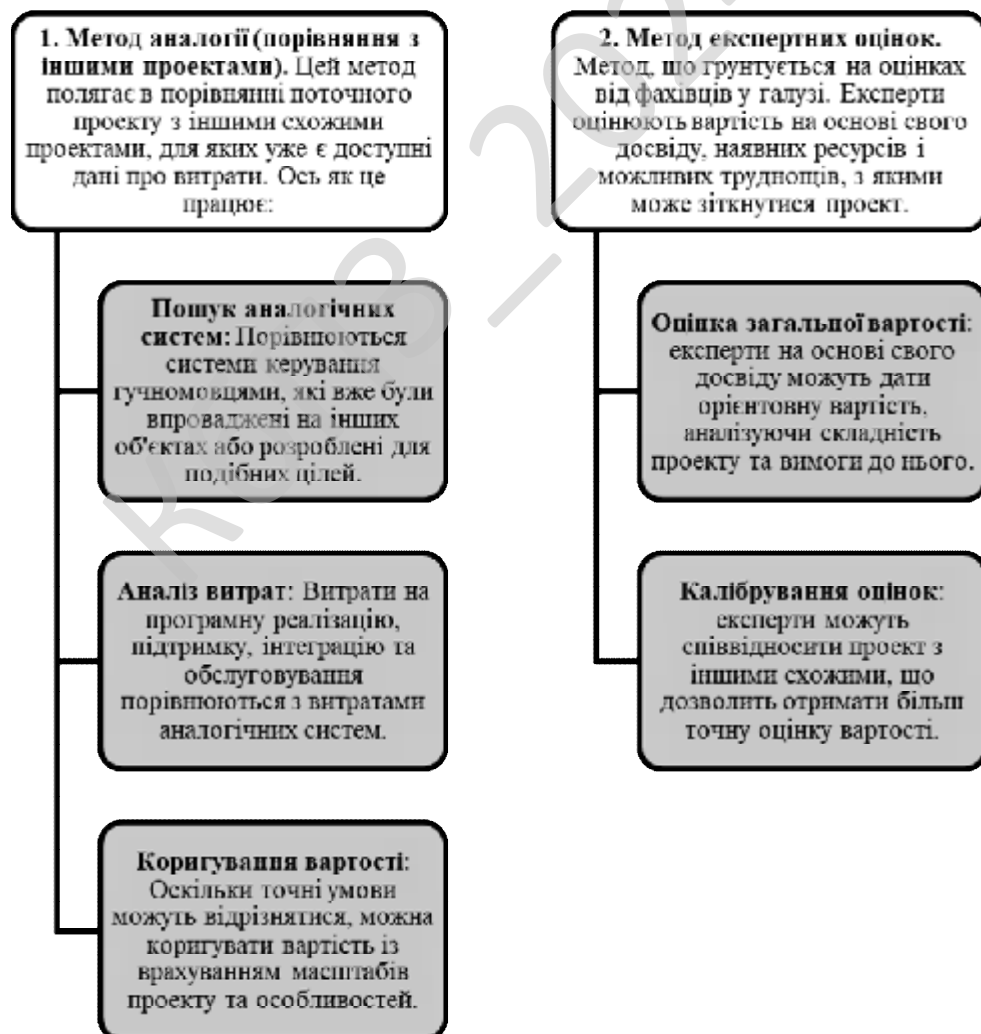


Рисунок 7.3 – Методи оцінки вартості

Застосування цих методів у поєднанні дозволить отримати точнішу оцінку вартості проекту, враховуючи всі аспекти його реалізації та експлуатації.

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

Оцінимо проєкт впровадження системи керування гучномовцями. Метою є впровадження системи автоматизованого керування гучномовцями в системах відеоспостереження для покращення ефективності комунікації в реальному часі та зниження витрат на охорону та управління безпекою.

Зведені вхідні дані заносимо в таблицю 7.2.

Таблиця 7.2 – Зведені витрати на реалізацію проєкту для клієнта

Інвестиційні витрати:	Операційні витрати:
Розробка програмного забезпечення для управління гучномовцями – 100 000 грн.	Обслуговування програмного забезпечення та апаратних засобів на рік – 50 000 грн.
Придбання та монтаж гучномовців (500 одиниць) – 200 000 грн.	Навчання персоналу для роботи з системою – 10 000 грн.
Інтеграція з існуючою системою відеоспостереження – 50 000 грн.	Загальні операційні витрати на рік: 60 000 грн.
Технічне обладнання для підключення (сервери, маршрутизатори тощо) – 80 000 грн.	
Загальні інвестиційні витрати: 430 000 грн.	

Очікувані вигоди від впровадження мають наступний вигляд.

Покращення швидкості реагування на інциденти: впровадження системи керування гучномовцями дозволить швидше передавати інструкції та

попередження співробітникам охорони та співробітникам в зоні відеоспостереження, що зменшує час реагування на надзвичайні ситуації.

Зниження часу реагування на інциденти на 30%. У разі, коли вартість збитків від одного інциденту становить 100 000 грн, економія від зниження часу реагування на 30% складе:  $100000 \text{ грн} \times 0.30 = 30000 \text{ грн/рік}$ .

Таблиця 7.3 – Зведені дані підсумкової економічної ефективності для клієнта

Загальні вигоди на рік:
Зниження часу реагування на інциденти: 30 000 грн.
Зниження витрат на персонал: 150 000 грн.
Зниження витрат через помилки: 18 000 грн.
Загальні вигоди = $30\,000 + 150\,000 + 18\,000 = 198\,000$ грн/рік.
Загальні витрати (первинні та операційні) на рік:
Операційні витрати: 60 000 грн.
Чисті вигоди:
Загальні вигоди за рік: 198 000 грн.
Операційні витрати: 60 000 грн.
Чисті вигоди = $198\,000 - 60\,000 = 138\,000$ грн/рік.
Початкові інвестиційні витрати: 430 000 грн.
Щорічні чисті вигоди: 138 000 грн.
Термін окупності: $\approx 3.11$ роки

Зменшення витрат на персонал: Завдяки автоматизації процесів управління гучномовцями, зменшується необхідність у великій кількості охоронців для вручну здійснюваного управління системою, що дозволяє скоротити витрати на оплату праці.

Економія на оплаті праці охоронців: скорочення потреби в 2-х співробітниках, які раніше вручну керували системою. Це економія в 150 000 грн на рік.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Зниження ризику помилок: Система автоматизує процеси, що зменшує ймовірність людських помилок під час управління гучномовцями, що може привести до значних економічних втрат через неправильні або запізнілі повідомлення.

Економія від зниження помилок: припускаємо, що помилки в управлінні вартували компанії 20 000 грн на рік. Завдяки автоматизації, ця сума може знизитись на 90%, що дає економію 18 000 грн.

Проект може окупитися за приблизно 3 роки, а далі приносити стабільний прибуток у вигляді економії на експлуатаційних витратах і зниженні збитків. Впровадження системи керування гучномовцями в системах відеоспостереження є економічно ефективним і забезпечує довгострокові вигоди для компанії.

### **7.5 Пропозиція алгоритму просування проєкту розробки ПЗ**

Для просування проєкту програмної реалізації системи керування гучномовцями в системах відеоспостереження можна розробити такий алгоритм (рисунок 7.4).

Алгоритм просування має включати багатоетапний підхід, що поєднує маркетингові стратегії, технічні рішення, партнерські угоди та персональні зустрічі для досягнення найкращих результатів у продажах і залученні клієнтів.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

1. Аналіз цільової аудиторії	<p>Хто є потенційними замовниками? Де можуть бути організації та підприємства, що мають великий обсяг об'єктів для відеоспостереження: торгові центри, аеропорти, промислові підприємства, навчальні заклади, органи правопорядку.</p> <p>Які їхні проблеми? Потреба в оперативному та ефективному управлінні функціями для передачі важливих повідомлень у реальному часі.</p> <p>Які критерії оцінки важливі для них? Надійність, економічна ефективність, простота інтеграції з існуючими системами, зручність у використанні, технічна підтримка.</p>
2. Розробка стратегії позиціонування продукту	<p>Позиціонування: Система керування функціями повинна бути представлена як інноваційне та ефективне рішення для автоматизації та покращення процесів управління безпекою та комунікацією.</p> <p>Переваги для клієнта: Зменшення затрат на персонал, підвищення швидкості реагування на надзвичайні ситуації, зниження помилок при передачі важливих повідомлень, зручність у управлінні.</p> <p>Унікальні пропозиції: Інтеграція з сучасними системами відеоспостереження, знижені витрати на операційне управління, високий рівень надійності та підтримки.</p>
3. Розробка маркетингових матеріалів	<p>Презентації та демонстрації: Підготовка детальних презентацій та відео, що демонструють переваги використання системи керування функціями в реальних умовах.</p> <p>Рекламні матеріали: Створення брошур, буклетів, інфографік, що наочно пояснюють функціонування системи, її переваги і можливості для бізнесу.</p> <p>Технічні документи: Опис можливостей інтеграції з різними типами систем відеоспостереження, вимоги до апаратного забезпечення, архітектура системи.</p>
4. Просування через онлайн-канали	<p>Створення сайту: Подання інформації про систему через сайт на веб-порталі презентації. Функціональні можливості, цінності, переваги та відгуки клієнтів.</p> <p>SEO-оптимізація: Оптимізація сайту для пошукових систем, щоб потенційні клієнти могли знайти вашу пропозицію за певними ключовими словами, такими як «спостереження та управління функціями».</p> <p>Реклама в соціальних мережах: Реклама через LinkedIn, Facebook, Twitter та інші платформи для таргетованого потенційних клієнтів і бізнес-партнерів.</p> <p>Інтернет-конференції та вебінари: Проведення онлайн-презентацій, вебінарів для потенційних клієнтів, що дозволять взаємодіяти з системою та отримувати відповіді на запитання.</p>
5. Прямі продажі та заходи	<p>Персональні зустрічі та демонстрації: Організація зустрічей з ключовими гравцями на ринку відеоспостереження та безпеки, демонстрація системи в реальних умовах.</p> <p>Участь у виставках і конференціях: Презентація продукту на спеціалізованих виставках і форумах з відеоспостереження, безпеки та технологій управління.</p> <p>Пробні версії та пілотні проекти: Запропонувати потенційним клієнтам можливість випробувати систему на обмеженій кількості об'єктів або у вигляді пілотного проекту.</p>
6. Партнерства та альянси	<p>Партнерства з постачальниками відеоспостереження: Співпраця з виробниками та постачальниками систем відеоспостереження для інтеграції програмного забезпечення для керування функціями в їхні системи.</p> <p>Партнерства з системними інтеграторами: Натягнення партнерських відносин з компаніями, які займаються інтеграцією різноманітних технологій для створення комплексних рішень.</p> <p>Міжнародні партнери: Якщо проект має міжнародний потенціал, шукайте партнерів у інших країнах для локалізації та адаптації продукту під місцеві ринки.</p>
7. Задучення відгуків і кейсів	<p>Відгуки клієнтів: Збір позитивних відгуків від перших клієнтів і використання їх у маркетингових матеріалах.</p> <p>Кейси впровадження: Створення кейсів впровадження системи на реальних об'єктах із детальним описом економічної ефективності та результатів.</p>
8. Підтримка та обслуговування	<p>Технічна підтримка: Запропонувати потенційним замовникам зручний доступ до технічної підтримки через онлайн-канали, гарячу лінію, чат-ботів.</p> <p>Оновлення та вдосконалення: Забезпечити постійну підтримку та оновлення програмного забезпечення, щоб підтримувати актуальність системи і її відповідність останнім вимогам.</p>

Рисунок 7.4 – Алгоритм просування проекту

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи керування гучномовцями в системах відеоспостереження можна застосувати кілька стратегій, що забезпечать ефективне охоплення цільової аудиторії та підвищення попиту на продукт.

Варто звернути увагу на наступні рекомендації.

1. Розширення каналів збуту: прямі продажі через власну команду, регулярні зустрічі та презентації продукту для ключових рішень у компаніях, партнерства з постачальниками відеоспостереження, інтеграція з системними інтеграторами;

2. Оптимізація маркетингових каналів: SEO та контент-маркетинг, реклама в Google Ads та соціальних мережах, просування через інфлюенсерів і технічних лідерів думок, реферальні програми;

3. Стратегія «пілотних проєктів»: запропонувати потенційним клієнтам провести пілотне впровадження програми в рамках конкретного об'єкта чи проєкту, що дозволить їм побачити переваги системи на практиці;

4. Створення спільнот та навчальних матеріалів: створення спеціалізованих форумів, вебінарів та онлайн-груп для обговорення питань безпеки, відеоспостереження і керування гучномовцями. Це дозволить залучити увагу до вашої системи та створити спільноту користувачів, які обмінюються досвідом.

5. Інтеграція з іншими технологіями: опрацювати можливість інтеграції вашої системи з іншими популярними платформами безпеки або системами управління інфраструктурою, такими як контроль доступу, охоронні системи або системи управління об'єктами, зробити систему модульною, щоб клієнти могли налаштовувати її відповідно до своїх потреб, що підвищить її гнучкість і привабливість для різних ринкових сегментів;

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

6. Післяпродажна підтримка та обслуговування: забезпечення технічної підтримки після продажу, включаючи оновлення системи, навчання персоналу та підтримку інсталяції.

Оптимізація каналів збуту та шляхів реалізації проекту включає в себе стратегічне партнерство, розширення маркетингових ініціатив, використання пілотних проектів для демонстрації реальної вартості продукту та забезпечення постійної підтримки користувачів. Такий підхід дозволить знизити ризики для клієнтів, підвищити лояльність і забезпечити сталий попит на систему.

### 7.7 Визначення ключових факторів успіху конкретного проекту

Ключові фактори успіху проекту програмної реалізації системи керування гучномовцями в системах відеоспостереження можна поділити на кілька основних аспектів (рис. 7.5):

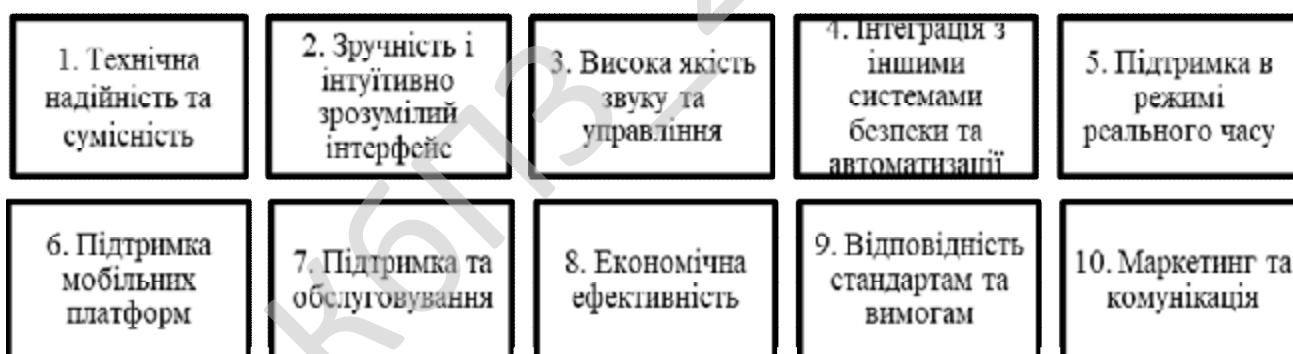


Рисунок 7.3 – Ключові фактори успіху проекту

Всі ці фактори сприяють успіху проекту програмної реалізації системи керування гучномовцями в системах відеоспостереження та допомагають забезпечити високу якість, ефективність та конкурентоспроможність продукту на ринку.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин (ЕОМ). Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливості. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, попереk тощо.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

## 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у тому числі програмісти), необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98 [2].

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

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

### 8.3 Аналіз умов праці на робочому місці програміста

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,8 м×7,2 м×2,8 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2,1 м×1,9 м, які виходять на північний схід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані у світло оранжевий колір до висоти 2,8 м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер, принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1245 мм×840 мм. Висота столів 750 мм. Висота стільців від рівня підлоги становить 425 мм.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м<sup>2</sup> на одне робоче місце. Кратність повітрообміну в приміщенні також регламентується ДСанПіН 3.3.2.007-98 [2], вона повинна становити 20 м<sup>3</sup>/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні оптимального значення вологості й складу повітря.

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

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

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення програмістів обчислювальних машин	71	61	54	49	45	42	40	38	50

Відповідно ДСанПіН 3.3.2.007-98 [2] рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

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

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

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів, відповідних вимогам нормативного акту

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

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

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

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

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

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

#### 8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною  $L=1,7$  м, та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір  $\rho_2 = 40$  Ом·м). Умовна товщина верхнього шару ґрунту  $H=0,5$  м. Відстань між вертикальними



Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [12]:

$$R_0 = 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_B) + (1/2) \cdot \lg((4T+L)/(4T-L))] = \\ = 0,366 \cdot (54,5/1,7) \cdot [(\lg(2 \cdot 1,7/0,0598) + (1/2) \cdot \lg((4 \cdot 1,45 + 1,7)/(4 \cdot 1,45 - \\ 1,7)))] = 22 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів  $K_{ев} = 0,62$  при орієнтовній кількості вертикальних електродів 5 [12].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при  $R_{3Н} = 4 \text{ Ом}$ :

$$N = R_0 / (K_{ев} \cdot R_{3Н}) = 22 / (0,62 \cdot 4) = 8,9 \approx 9 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{П} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 9 = 28,35 \approx 28 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту  $K_{П}$  [12]:

$$R_{П} = 0,366 \cdot (\rho \cdot K_{П}/L_{П}) \cdot \lg(2 \cdot L_{П}^2 / (B \cdot t)) = \\ = 0,366 \cdot (40 \cdot 5 / 28) \cdot \lg((2 \cdot 28^2) / (0,06 \cdot 0,6)) = 12,1 \text{ Ом.}$$

де  $K_{П} = 5$  – табличне значення кліматичного коефіцієнту питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [12]:

$B = 60 \text{ мм} = 0,06 \text{ м}$  – ширина з'єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [12]:

$$R = (R_0 \cdot R_{П}) / (R_0 \cdot \eta_{П} + N \cdot R_{П} \cdot K_{ев}) = \\ = (22 \cdot 12,1) / (22 \cdot 0,6 + 9 \cdot 12,1 \cdot 0,62) = 3,35 \text{ Ом.}$$

де  $\eta_{П} = 0,6$  – табличне значення коефіцієнту екранування з'єднуючої полоси [12].

Умова  $R \leq R_{3Н}$  виконується ( $3,35 \leq 4$ ).

Оскільки при 9 вертикальних електродах  $R$  суттєво менше  $R_{3Н}$ , зменшимо кількість вертикальних електродів  $N$  до 8 і виконаємо перерахунок. У результаті остаточно отримали:  $R = 3,7 \text{ Ом}$ . при кількості вертикальних електродів  $N = 8$ .

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>81</b>

## 8.5 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 умов праці на робочому місці програміста, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливу шкоду та небезпеку, що можуть підстерігати його на робочому місці, та дотримання вимог нормативних актів з 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.

КБПЗ - 2024

					VKPM-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування гучномовцями в системах відеоспостереження.

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 керування гучномовцями в системах відеоспостереження.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем керування гучномовцями в системах відеоспостереження.

– Досліджена система керування гучномовцями в системах відеоспостереження.

– На основі отриманих результатів досліджень створена програмна реалізація системи керування гучномовцями в системах відеоспостереження.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування гучномовцями в системах відеоспостереження.

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

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

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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 Visual C#. 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 ДСТУ 7624:2014.

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.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Осуський А.В. Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
4. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
5. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
6. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
7. 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.
8. 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.
9. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
10. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
11. 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.

					ВКРМ-123.24.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

12. 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

13. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

14. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchov, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

15. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

16. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

17. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

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

19. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing

Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

20. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

21. 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.

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

23. 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.

24. 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.

25. 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.

26. 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.

27. 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 Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

28. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

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

30. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

31. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

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

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

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

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

49. 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.

50. 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).

51. 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

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

					<b>ВКРМ-123.24.0029.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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-123.24.0029.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Осуський А.В.				<i>Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження</i>	Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.					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 керування гучномовцями в системах відеоспостереження.

## 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 керування гучномовцями в системах відеоспостереження.

## 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-123.24.0029.00.00.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 керування гучномовцями в системах відеоспостереження;
- 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-123.24.0029.00.00.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 Visual C#.

					ВКРМ-123.24.0029.00.00.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 розглянутий аналіз умов праці на робочому місці програміста.

					ВКРМ-123.24.0029.00.00.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 – 91 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 20.12.2024 р.

					<b>ВКРМ-123.24.0029.00.00.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

\_\_\_\_\_ Усік П.С.

*Дослідження та програмна реалізація  
системи керування гучномовцями в системах відеоспостереження*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 41

Літера: РП

Кропивницький – 2024 року

**Файл CameraInfo.cs – отримання інформації про камеру для керування гучномовцями  
в системах відеоспостереження**

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;

namespace CameraViewer
{
    /// <summary>
    /// інформація з CameraInfo.
    /// </summary>
    public class CameraInfo : System.Windows.Forms.Form
    {
        private Camera camera;

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label widthLabel;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label nameLabel;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label descriptionLabel;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label providerLabel;
        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.Label heightLabel;
        private System.Windows.Forms.Button closeButton;
        private System.Windows.Forms.PictureBox pictureBox2;
        /// <summary>
        /// Необхідні змінні розробника.
        /// </summary>
        private System.ComponentModel.Container components = null;

        // Camera властивості
        public Camera Camera
        {
            get { return camera; }
            set { camera = value; }
        }

        // Конструктор
        public CameraInfo()
        {
            //
            // Необхідно для підтримки Windows Form Designer
            //
            InitializeComponent();

            //
            // ПРИМІТКА Додаємо любий конструктор коду після виклику
InitializeComponent
            //
        }

        /// <summary>
        /// Очищуємо використані ресурси.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if( disposing )
            {
                if(components != null)
                {
                    components.Dispose();
                }
            }
        }
    }
}

```

```

        }
        base.Dispose( disposing );
    }

    #region Windows Form Designer generated code
    /// <summary>
    /// Необхідний метод розробника - не модифікується
    /// зміст цього методу з редактором коду.
    /// </summary>
    private void InitializeComponent()
    {
        this.label1 = new System.Windows.Forms.Label();
        this.widthLabel = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.nameLabel = new System.Windows.Forms.Label();
        this.label5 = new System.Windows.Forms.Label();
        this.descriptionLabel = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
        this.providerLabel = new System.Windows.Forms.Label();
        this.pictureBox1 = new System.Windows.Forms.PictureBox();
        this.heightLabel = new System.Windows.Forms.Label();
        this.closeButton = new System.Windows.Forms.Button();
        this.pictureBox2 = new System.Windows.Forms.PictureBox();
        this.SuspendLayout();
        //
        // label1
        //
        this.label1.Location = new System.Drawing.Point(8, 136);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(40, 14);
        this.label1.TabIndex = 0;
        this.label1.Text = "Width:";
        //
        // widthLabel
        //
        this.widthLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.widthLabel.Location = new System.Drawing.Point(60, 135);
        this.widthLabel.Name = "widthLabel";
        this.widthLabel.Size = new System.Drawing.Size(60, 16);
        this.widthLabel.TabIndex = 1;
        this.widthLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
        //
        // label2
        //
        this.label2.Location = new System.Drawing.Point(140, 136);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(40, 14);
        this.label2.TabIndex = 2;
        this.label2.Text = "Height:";
        //
        // label3
        //
        this.label3.Location = new System.Drawing.Point(8, 9);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(40, 14);
        this.label3.TabIndex = 3;
        this.label3.Text = "Name:";
        //
        // nameLabel
        //
        this.nameLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.nameLabel.Location = new System.Drawing.Point(60, 8);
        this.nameLabel.Name = "nameLabel";
        this.nameLabel.Size = new System.Drawing.Size(200, 16);
    }

```

```

        this.nameLabel.TabIndex = 4;
        //
        // label5
        //
        this.label5.Location = new System.Drawing.Point(8, 30);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(65, 14);
        this.label5.TabIndex = 5;
        this.label5.Text = "Description:";
        //
        // descriptionLabel
        //
        this.descriptionLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.descriptionLabel.Location = new System.Drawing.Point(8,
47);

        this.descriptionLabel.Name = "descriptionLabel";
        this.descriptionLabel.Size = new System.Drawing.Size(252, 40);
        this.descriptionLabel.TabIndex = 6;
        //
        // label4
        //
        this.label4.Location = new System.Drawing.Point(8, 96);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(50, 14);
        this.label4.TabIndex = 7;
        this.label4.Text = "Provider:";
        //
        // providerLabel
        //
        this.providerLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.providerLabel.Location = new System.Drawing.Point(60,
95);

        this.providerLabel.Name = "providerLabel";
        this.providerLabel.Size = new System.Drawing.Size(200, 16);
        this.providerLabel.TabIndex = 8;
        //
        // pictureBox1
        //
        this.pictureBox1.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
        this.pictureBox1.Location = new System.Drawing.Point(8, 120);
        this.pictureBox1.Name = "pictureBox1";
        this.pictureBox1.Size = new System.Drawing.Size(252, 2);
        this.pictureBox1.TabIndex = 9;
        this.pictureBox1.TabStop = false;
        //
        // heightLabel
        //
        this.heightLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
        this.heightLabel.Location = new System.Drawing.Point(200,
135);

        this.heightLabel.Name = "heightLabel";
        this.heightLabel.Size = new System.Drawing.Size(60, 16);
        this.heightLabel.TabIndex = 10;
        this.heightLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
        //
        // closeButton
        //
        this.closeButton.DialogResult =
System.Windows.Forms.DialogResult.OK;
        this.closeButton.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.closeButton.Location = new System.Drawing.Point(98, 175);
        this.closeButton.Name = "closeButton";
        this.closeButton.TabIndex = 11;

```

```

        this.closeButton.Text = "Close";
        //
        // pictureBox2
        //
        this.pictureBox2.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
        this.pictureBox2.Location = new System.Drawing.Point(9, 163);
        this.pictureBox2.Name = "pictureBox2";
        this.pictureBox2.Size = new System.Drawing.Size(252, 2);
        this.pictureBox2.TabIndex = 12;
        this.pictureBox2.TabStop = false;
        //
        // CameraInfo
        //
        this.AcceptButton = this.closeButton;
        this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
        this.CancelButton = this.closeButton;
        this.ClientSize = new System.Drawing.Size(270, 206);
        this.Controls.AddRange(new System.Windows.Forms.Control[] {

                this.pictureBox2,

                this.closeButton,

                this.heightLabel,

                this.pictureBox1,

                this.providerLabel,

                this.label4,

                this.descriptionLabel,

                this.label5,

                this.nameLabel,

                this.label3,

                this.label2,

                this.widthLabel,

                this.label1});
        this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedToolWindow;
        this.MaximizeBox = false;
        this.MinimizeBox = false;
        this.Name = "CameraInfo";
        this.Opacity = 0.85;
        this.ShowInTaskbar = false;
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterParent;
        this.Text = "Camera info";
        this.Load += new System.EventHandler(this.CameraInfo_Load);
        this.ResumeLayout(false);

    }
    #endregion

    // On load
    private void CameraInfo_Load(object sender, System.EventArgs e)
    {
        if (camera != null)
        {
            nameLabel.Text = camera.Name;
            descriptionLabel.Text = camera.Description;
            providerLabel.Text = camera.Provider.Name;
        }
    }
}

```

```
        if (camera.Width != -1)
        {
            widthLabel.Text = camera.Width.ToString();
            heightLabel.Text = camera.Height.ToString();
        }
        else
        {
            widthLabel.Text = string.Empty;
            heightLabel.Text = string.Empty;
        }
    }
}
}
```

K6ПЗ\_2024

**Файл MultimodeVideoSource.cs – робота з відеокамерами для керування гучномовцями в системах відеоспостереження**

```

namespace multisource
{
    using System;
    using System.Collections;
    using videosource;
    using CUDA;
    using mCUDA;

    /// <summary>
    /// MultimodeVideoSource - абстрактний клас для відеоресурсів, з
    підтримкою
    /// мульти робочого режиму
    /// </summary>
    public abstract class MultimodeVideoSource : IVideoSource
    {
        protected IVideoSource videoSource;
        protected StreamType streamType;
        private ArrayList delegates = new ArrayList();

        // Нова подія у фреймі
        public event CameraEventHandler NewFrame
        {
            add
            {
                videoSource.NewFrame += value;
                delegates.Add((object) value);
            }
            remove
            {
                videoSource.NewFrame -= value;
                delegates.Remove((object) value);
            }
        }

        // Конструктор
        public MultimodeVideoSource()
        {
        }

        // StreamType властивості
        public virtual StreamType StreamType
        {
            get { return streamType; }
            set
            {
                // покращує потоковий тип, якщо відео джерело не
                рухається
                if ((streamType != value) && (!videoSource.Running))
                {
                    streamType = value;

                    // зберігає дані
                    object userData = videoSource.UserData;
                    string login = videoSource.Login;
                    string password = videoSource.Password;

                    // створює нове базове відео джерело
                    switch (streamType)
                    {
                        case StreamType.CUDA:
                            videoSource = new CUDASource();
                            break;
                        case StreamType.MCUDA:
                            videoSource = new MCUDASource();
                    }
                }
            }
        }
    }
}

```

```

        break;
    }

    // бере дані та повертає у нове відеоджерело
    videoSource.Login      = login;
    videoSource.Password   = password;
    videoSource.UserData   = userData;

    // додає делегування до NewFrame події
    foreach (object handler in delegates)
        videoSource.NewFrame += (CameraEventHandler)

handler;

        UpdateVideoSource();
    }
}

// Властивості відеоджерела
public abstract string VideoSource
{
    get;
    set;
}

// Властивості підключення
public string Login
{
    get { return videoSource.Login; }
    set { videoSource.Login = value; }
}

// Властивості паролювання
public string Password
{
    get { return videoSource.Password; }
    set { videoSource.Password = value; }
}

// FramesReceived властивості
public int FramesReceived
{
    get { return videoSource.FramesReceived; }
}

// BytesReceived властивості
public int BytesReceived
{
    get { return videoSource.BytesReceived; }
}

// UserData властивості
public object UserData
{
    get { return videoSource.UserData; }
    set { videoSource.UserData = value; }
}

// Беремо стан відеоджерела
public bool Running
{
    get { return videoSource.Running; }
}

// Починаємо отримувати відеофрейми
public void Start()
{
    videoSource.Start();
}

```

```
// Закінчуємо отримувати відеофрейми
public void SignalToStop()
{
    videoSource.SignalToStop();
}

// Чекаємо закінчення
public void WaitForStop()
{
    videoSource.WaitForStop();
}

// Закінчення роботи
public void Stop()
{
    videoSource.Stop();
}

// Обновлюємо відеоджерело
protected abstract void UpdateVideoSource();
}
}
```

КБПЗ\_2024

**Файл Multiplexer.cs - мультиплексування (для роботи з декількома камерами для керування гучномовцями в системах відеоспостереження)**

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;

namespace CameraViewer
{
    /// <summary>
    /// Загальний опис для мультиплексора.
    /// </summary>
    public class Multiplexer : System.Windows.Forms.Panel
    {
        private const int      MaxRows = 5;
        private const int      MaxCols = 5;
        private CameraWindow[,] camWindows;

        private bool          fitToWindow = false;
        private bool          singleCameraMode = true;
        private bool          camerasVisible = false;

        private int           rows = 1;
        private int           cols = 1;
        private int           cellWidth = 320;
        private int           cellHeight = 240;

        private CameraWindow  lastClicked;

        private CameraViewer.CameraWindow cameraWindow1;
        private CameraViewer.CameraWindow cameraWindow2;
        private CameraViewer.CameraWindow cameraWindow3;
        private CameraViewer.CameraWindow cameraWindow4;
        private CameraViewer.CameraWindow cameraWindow5;
        private CameraViewer.CameraWindow cameraWindow6;
        private CameraViewer.CameraWindow cameraWindow7;
        private CameraViewer.CameraWindow cameraWindow8;
        private CameraViewer.CameraWindow cameraWindow9;
        private CameraViewer.CameraWindow cameraWindow10;
        private CameraViewer.CameraWindow cameraWindow11;
        private CameraViewer.CameraWindow cameraWindow12;
        private CameraViewer.CameraWindow cameraWindow13;
        private CameraViewer.CameraWindow cameraWindow14;
        private CameraViewer.CameraWindow cameraWindow15;
        private CameraViewer.CameraWindow cameraWindow16;
        private CameraViewer.CameraWindow cameraWindow17;
        private CameraViewer.CameraWindow cameraWindow18;
        private CameraViewer.CameraWindow cameraWindow19;
        private CameraViewer.CameraWindow cameraWindow20;
        private CameraViewer.CameraWindow cameraWindow21;
        private CameraViewer.CameraWindow cameraWindow22;
        private CameraViewer.CameraWindow cameraWindow23;
        private CameraViewer.CameraWindow cameraWindow24;
        private CameraViewer.CameraWindow cameraWindow25;
        /// <summary>
        /// Опис змінних розробника.
        /// </summary>
        private System.ComponentModel.Container components = null;

        // FitToWindow властивості
        [DefaultValue(false)]
        public bool FitToWindow
        {
            get { return fitToWindow; }
        }
    }
}

```

```

        set
        {
            fitToWindow = value;

            if ((camWindows[0, 0].AutoSize = (!fitToWindow &&
singleCameraMode)) == true)
            {
                camWindows[0, 0].UpdatePosition();
            }
            else
            {
                UpdateSize();
            }
        }
    }
}
// SingleCameraMode властивості
[DefaultValue(true)]
public bool SingleCameraMode
{
    get { return singleCameraMode; }
    set
    {
        singleCameraMode = value;
        if (!fitToWindow)
            camWindows[0, 0].AutoSize = value;
    }
}
// CamerasVisible властивості
[DefaultValue(false)]
public bool CamerasVisible
{
    get { return camerasVisible; }
    set
    {
        camerasVisible = value;

        // Показувати/приховувати усі камери
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                camWindows[i, j].Visible = value;
            }
        }
    }
}
// Rows властивості
[DefaultValue(1)]
public int Rows
{
    get { return rows; }
    set
    {
        {
            rows = Math.Max(1, Math.Min(MaxRows, value));
            UpdateVisiblity();
            UpdateSize();
        }
    }
}
// Cols властивості
[DefaultValue(1)]
public int Cols
{
    get { return cols; }
    set
    {
        {
            cols = Math.Max(1, Math.Min(MaxCols, value));
            UpdateVisiblity();
            UpdateSize();
        }
    }
}

```

```

}
// CellWidth
[DefaultValue(320)]
public int CellWidth
{
    get { return cellWidth; }
    set
    {
        cellWidth = Math.Max(50, Math.Min(800, value));
        UpdateSize();
    }
}
// CellHeight
[DefaultValue(240)]
public int CellHeight
{
    get { return cellHeight; }
    set
    {
        cellHeight = Math.Max(50, Math.Min(800, value));
        UpdateSize();
    }
}
// Контекстне меню у вікні камери
[DefaultValue(null)]
public ContextMenu CamerasContextMenu
{
    get { return camWindows[0, 0].ContextMenu; }
    set
    {
        for (int i = 0; i < MaxRows; i++)
        {
            for (int j = 0; j < MaxCols; j++)
            {
                camWindows[i, j].ContextMenu = value;
            }
        }
    }
}
// Камера при останньому нажатті
[Browsable(false)]
public Camera ContextCamera
{
    get { return (lastClicked == null) ? null :
lastClicked.Camera; }
}

// Конструктор
public Multiplexer()
{
    // Цей виклик використовується у Windows.Forms Form Designer.
    InitializeComponent();

    // ПРИМІТКА Додається ініціалізація після виклику InitForm
    camWindows = new CameraWindow[MaxRows, MaxCols];

    // row 1
    camWindows[0, 0] = cameraWindow1;
    camWindows[0, 1] = cameraWindow2;
    camWindows[0, 2] = cameraWindow3;
    camWindows[0, 3] = cameraWindow4;
    camWindows[0, 4] = cameraWindow5;
    // row 2
    camWindows[1, 0] = cameraWindow6;
    camWindows[1, 1] = cameraWindow7;
    camWindows[1, 2] = cameraWindow8;
    camWindows[1, 3] = cameraWindow9;
    camWindows[1, 4] = cameraWindow10;
    // row 3

```

```

        camWindows[2, 0] = cameraWindow11;
        camWindows[2, 1] = cameraWindow12;
        camWindows[2, 2] = cameraWindow13;
        camWindows[2, 3] = cameraWindow14;
        camWindows[2, 4] = cameraWindow15;
        // row 4
        camWindows[3, 0] = cameraWindow16;
        camWindows[3, 1] = cameraWindow17;
        camWindows[3, 2] = cameraWindow18;
        camWindows[3, 3] = cameraWindow19;
        camWindows[3, 4] = cameraWindow20;
        // row 5
        camWindows[4, 0] = cameraWindow21;
        camWindows[4, 1] = cameraWindow22;
        camWindows[4, 2] = cameraWindow23;
        camWindows[4, 3] = cameraWindow24;
        camWindows[4, 4] = cameraWindow25;
    }

    /// <summary>
    /// Очищуємо усі ресурси використовувані користувачем.
    /// </summary>
    protected override void Dispose( bool disposing )
    {
        if( disposing )
        {
            if(components != null)
            {
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }

    #region Component Designer generated code
    /// <summary>
    /// Необхідний метод для підтримки розробника - не модифікується
    /// контент цього методу з редактором коду.
    /// </summary>
    private void InitializeComponent()
    {
        this.cameraWindow1 = new CameraViewer.CameraWindow();
        this.cameraWindow2 = new CameraViewer.CameraWindow();
        this.cameraWindow3 = new CameraViewer.CameraWindow();
        this.cameraWindow4 = new CameraViewer.CameraWindow();
        this.cameraWindow5 = new CameraViewer.CameraWindow();
        this.cameraWindow6 = new CameraViewer.CameraWindow();
        this.cameraWindow7 = new CameraViewer.CameraWindow();
        this.cameraWindow8 = new CameraViewer.CameraWindow();
        this.cameraWindow9 = new CameraViewer.CameraWindow();
        this.cameraWindow10 = new CameraViewer.CameraWindow();
        this.cameraWindow11 = new CameraViewer.CameraWindow();
        this.cameraWindow12 = new CameraViewer.CameraWindow();
        this.cameraWindow13 = new CameraViewer.CameraWindow();
        this.cameraWindow14 = new CameraViewer.CameraWindow();
        this.cameraWindow15 = new CameraViewer.CameraWindow();
        this.cameraWindow16 = new CameraViewer.CameraWindow();
        this.cameraWindow17 = new CameraViewer.CameraWindow();
        this.cameraWindow18 = new CameraViewer.CameraWindow();
        this.cameraWindow19 = new CameraViewer.CameraWindow();
        this.cameraWindow20 = new CameraViewer.CameraWindow();
        this.cameraWindow21 = new CameraViewer.CameraWindow();
        this.cameraWindow22 = new CameraViewer.CameraWindow();
        this.cameraWindow23 = new CameraViewer.CameraWindow();
        this.cameraWindow24 = new CameraViewer.CameraWindow();
        this.cameraWindow25 = new CameraViewer.CameraWindow();
        this.SuspendLayout();
        //
        // cameraWindow1
    }

```

```

        //
        this.cameraWindow1.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow1.Camera = null;
        this.cameraWindow1.Location = new System.Drawing.Point(285,
17);

        this.cameraWindow1.Name = "cameraWindow1";
        this.cameraWindow1.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow1.TabIndex = 0;
        this.cameraWindow1.Text = "cameraWindow1";
        this.cameraWindow1.Visible = false;
        this.cameraWindow1.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow2
        //
        this.cameraWindow2.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow2.Camera = null;
        this.cameraWindow2.Location = new System.Drawing.Point(151,
17);

        this.cameraWindow2.Name = "cameraWindow2";
        this.cameraWindow2.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow2.TabIndex = 1;
        this.cameraWindow2.Text = "cameraWindow2";
        this.cameraWindow2.Visible = false;
        this.cameraWindow2.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow3
        //
        this.cameraWindow3.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow3.Camera = null;
        this.cameraWindow3.Location = new System.Drawing.Point(419,
17);

        this.cameraWindow3.Name = "cameraWindow3";
        this.cameraWindow3.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow3.TabIndex = 2;
        this.cameraWindow3.Text = "cameraWindow3";
        this.cameraWindow3.Visible = false;
        this.cameraWindow3.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow4
        //
        this.cameraWindow4.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow4.Camera = null;
        this.cameraWindow4.Location = new System.Drawing.Point(553,
17);

        this.cameraWindow4.Name = "cameraWindow4";
        this.cameraWindow4.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow4.TabIndex = 3;
        this.cameraWindow4.Text = "cameraWindow4";
        this.cameraWindow4.Visible = false;
        this.cameraWindow4.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow5
        //
        this.cameraWindow5.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow5.Camera = null;
        this.cameraWindow5.Location = new System.Drawing.Point(17,
54);

        this.cameraWindow5.Name = "cameraWindow5";
        this.cameraWindow5.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow5.TabIndex = 4;

```

```

        this.cameraWindow5.Text = "cameraWindow5";
        this.cameraWindow5.Visible = false;
        this.cameraWindow5.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow6
        //
        this.cameraWindow6.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow6.Camera = null;
        this.cameraWindow6.Location = new System.Drawing.Point(17,
91);

        this.cameraWindow6.Name = "cameraWindow6";
        this.cameraWindow6.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow6.TabIndex = 5;
        this.cameraWindow6.Text = "cameraWindow6";
        this.cameraWindow6.Visible = false;
        this.cameraWindow6.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow7
        //
        this.cameraWindow7.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow7.Camera = null;
        this.cameraWindow7.Location = new System.Drawing.Point(17,
91);

        this.cameraWindow7.Name = "cameraWindow7";
        this.cameraWindow7.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow7.TabIndex = 6;
        this.cameraWindow7.Text = "cameraWindow7";
        this.cameraWindow7.Visible = false;
        this.cameraWindow7.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow8
        //
        this.cameraWindow8.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow8.Camera = null;
        this.cameraWindow8.Location = new System.Drawing.Point(17,
91);

        this.cameraWindow8.Name = "cameraWindow8";
        this.cameraWindow8.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow8.TabIndex = 7;
        this.cameraWindow8.Text = "cameraWindow8";
        this.cameraWindow8.Visible = false;
        this.cameraWindow8.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow9
        //
        this.cameraWindow9.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow9.Camera = null;
        this.cameraWindow9.Location = new System.Drawing.Point(151,
91);

        this.cameraWindow9.Name = "cameraWindow9";
        this.cameraWindow9.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow9.TabIndex = 8;
        this.cameraWindow9.Text = "cameraWindow9";
        this.cameraWindow9.Visible = false;
        this.cameraWindow9.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow10
        //
        this.cameraWindow10.BackColor =
System.Drawing.SystemColors.ControlDarkDark;

```

```

        this.cameraWindow10.Camera = null;
        this.cameraWindow10.Location = new System.Drawing.Point(328,
80);
        this.cameraWindow10.Name = "cameraWindow10";
        this.cameraWindow10.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow10.TabIndex = 9;
        this.cameraWindow10.Text = "cameraWindow10";
        this.cameraWindow10.Visible = false;
        this.cameraWindow10.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow11
        //
        this.cameraWindow11.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow11.Camera = null;
        this.cameraWindow11.Location = new System.Drawing.Point(8,
152);
        this.cameraWindow11.Name = "cameraWindow11";
        this.cameraWindow11.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow11.TabIndex = 10;
        this.cameraWindow11.Text = "cameraWindow11";
        this.cameraWindow11.Visible = false;
        this.cameraWindow11.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow12
        //
        this.cameraWindow12.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow12.Camera = null;
        this.cameraWindow12.Location = new System.Drawing.Point(88,
152);
        this.cameraWindow12.Name = "cameraWindow12";
        this.cameraWindow12.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow12.TabIndex = 11;
        this.cameraWindow12.Text = "cameraWindow12";
        this.cameraWindow12.Visible = false;
        this.cameraWindow12.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow13
        //
        this.cameraWindow13.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow13.Camera = null;
        this.cameraWindow13.Location = new System.Drawing.Point(228,
152);
        this.cameraWindow13.Name = "cameraWindow13";
        this.cameraWindow13.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow13.TabIndex = 12;
        this.cameraWindow13.Text = "cameraWindow13";
        this.cameraWindow13.Visible = false;
        this.cameraWindow13.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow14
        //
        this.cameraWindow14.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow14.Camera = null;
        this.cameraWindow14.Location = new System.Drawing.Point(248,
152);
        this.cameraWindow14.Name = "cameraWindow14";
        this.cameraWindow14.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow14.TabIndex = 13;
        this.cameraWindow14.Text = "cameraWindow14";
        this.cameraWindow14.Visible = false;

```

```

        this.cameraWindow14.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow15
        //
        this.cameraWindow15.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow15.Camera = null;
        this.cameraWindow15.Location = new System.Drawing.Point(388,
152);

        this.cameraWindow15.Name = "cameraWindow15";
        this.cameraWindow15.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow15.TabIndex = 14;
        this.cameraWindow15.Text = "cameraWindow15";
        this.cameraWindow15.Visible = false;
        this.cameraWindow15.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow16
        //
        this.cameraWindow16.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow16.Camera = null;
        this.cameraWindow16.Location = new System.Drawing.Point(528,
152);

        this.cameraWindow16.Name = "cameraWindow16";
        this.cameraWindow16.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow16.TabIndex = 15;
        this.cameraWindow16.Text = "cameraWindow16";
        this.cameraWindow16.Visible = false;
        this.cameraWindow16.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow17
        //
        this.cameraWindow17.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow17.Camera = null;
        this.cameraWindow17.Location = new System.Drawing.Point(17,
189);

        this.cameraWindow17.Name = "cameraWindow17";
        this.cameraWindow17.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow17.TabIndex = 16;
        this.cameraWindow17.Text = "cameraWindow17";
        this.cameraWindow17.Visible = false;
        this.cameraWindow17.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow18
        //
        this.cameraWindow18.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow18.Camera = null;
        this.cameraWindow18.Location = new System.Drawing.Point(157,
189);

        this.cameraWindow18.Name = "cameraWindow18";
        this.cameraWindow18.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow18.TabIndex = 17;
        this.cameraWindow18.Text = "cameraWindow18";
        this.cameraWindow18.Visible = false;
        this.cameraWindow18.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow19
        //
        this.cameraWindow19.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow19.Camera = null;

```

```

    this.cameraWindow19.Location = new System.Drawing.Point(297,
189);
    this.cameraWindow19.Name = "cameraWindow19";
    this.cameraWindow19.Size = new System.Drawing.Size(75, 64);
    this.cameraWindow19.TabIndex = 18;
    this.cameraWindow19.Text = "cameraWindow19";
    this.cameraWindow19.Visible = false;
    this.cameraWindow19.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
    //
    // cameraWindow20
    //
    this.cameraWindow20.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
    this.cameraWindow20.Camera = null;
    this.cameraWindow20.Location = new System.Drawing.Point(17,
261);
    this.cameraWindow20.Name = "cameraWindow20";
    this.cameraWindow20.Size = new System.Drawing.Size(75, 64);
    this.cameraWindow20.TabIndex = 19;
    this.cameraWindow20.Text = "cameraWindow20";
    this.cameraWindow20.Visible = false;
    this.cameraWindow20.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
    //
    // cameraWindow21
    //
    this.cameraWindow21.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
    this.cameraWindow21.Camera = null;
    this.cameraWindow21.Location = new System.Drawing.Point(17,
298);
    this.cameraWindow21.Name = "cameraWindow21";
    this.cameraWindow21.Size = new System.Drawing.Size(75, 64);
    this.cameraWindow21.TabIndex = 20;
    this.cameraWindow21.Text = "cameraWindow21";
    this.cameraWindow21.Visible = false;
    this.cameraWindow21.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
    //
    // cameraWindow22
    //
    this.cameraWindow22.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
    this.cameraWindow22.Camera = null;
    this.cameraWindow22.Location = new System.Drawing.Point(17,
335);
    this.cameraWindow22.Name = "cameraWindow22";
    this.cameraWindow22.Size = new System.Drawing.Size(75, 64);
    this.cameraWindow22.TabIndex = 21;
    this.cameraWindow22.Text = "cameraWindow22";
    this.cameraWindow22.Visible = false;
    this.cameraWindow22.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
    //
    // cameraWindow23
    //
    this.cameraWindow23.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
    this.cameraWindow23.Camera = null;
    this.cameraWindow23.Location = new System.Drawing.Point(17,
335);
    this.cameraWindow23.Name = "cameraWindow23";
    this.cameraWindow23.Size = new System.Drawing.Size(75, 64);
    this.cameraWindow23.TabIndex = 22;
    this.cameraWindow23.Text = "cameraWindow23";
    this.cameraWindow23.Visible = false;
    this.cameraWindow23.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);

```

```

//
// cameraWindow24
//
this.cameraWindow24.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
this.cameraWindow24.Camera = null;
this.cameraWindow24.Location = new System.Drawing.Point(17,
335);

this.cameraWindow24.Name = "cameraWindow24";
this.cameraWindow24.Size = new System.Drawing.Size(75, 64);
this.cameraWindow24.TabIndex = 23;
this.cameraWindow24.Text = "cameraWindow24";
this.cameraWindow24.Visible = false;
this.cameraWindow24.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
//
// cameraWindow25
//
this.cameraWindow25.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
this.cameraWindow25.Camera = null;
this.cameraWindow25.Location = new System.Drawing.Point(17,
335);

this.cameraWindow25.Name = "cameraWindow25";
this.cameraWindow25.Size = new System.Drawing.Size(75, 64);
this.cameraWindow25.TabIndex = 24;
this.cameraWindow25.Text = "cameraWindow25";
this.cameraWindow25.Visible = false;
this.cameraWindow25.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
//
// Мультиплексер
//
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.cameraWindow25,
this.cameraWindow24,
this.cameraWindow23,
this.cameraWindow22,
this.cameraWindow21,
this.cameraWindow20,
this.cameraWindow19,
this.cameraWindow18,
this.cameraWindow17,
this.cameraWindow16,
this.cameraWindow15,
this.cameraWindow14,
this.cameraWindow13,
this.cameraWindow12,
this.cameraWindow11,
this.cameraWindow10,
this.cameraWindow9,

```

```

        this.cameraWindow8,
        this.cameraWindow7,
        this.cameraWindow6,
        this.cameraWindow5,
        this.cameraWindow4,
        this.cameraWindow3,
        this.cameraWindow2,
        this.cameraWindow1});
    this.Size = new System.Drawing.Size(424, 376);
    this.Resize += new
System.EventHandler(this.Multiplexer_Resize);
    this.ResumeLayout(false);

}
#endregion

// Закриваємо усі камери
public void CloseAll()
{
    for (int i = 0; i < MaxRows; i++)
    {
        for (int j = 0; j < MaxCols; j++)
        {
            camWindows[i, j].Camera = null;
        }
    }
}

// Беремо зображення з камери у спеціальну позицію для
мультиплексера
public void SetCamera(int row, int col, Camera camera)
{
    if ((row >= 0) && (col >= 0) && (row < MaxRows) && (col <
MaxCols))
    {
        camWindows[row, col].Camera = camera;
    }
}

// Встановлюємо розмір мультиплексера
public void SetSize(int rows, int cols, int cellWidth, int
cellHeight)
{
    this.rows = rows;
    this.cols = cols;
    this.cellWidth = cellWidth;
    this.cellHeight = cellHeight;
    UpdateSize();
}

// Оновлюємо зображення камери
private void UpdateVisiblity()
{
    if (camerasVisible)
    {
        for (int i = 0; i < MaxRows; i++)
        {
            for (int j = 0; j < MaxCols; j++)
            {

```

```

camWindows[i, j].Visible = ((i < rows) && (j
< cols));
    }
}

// Оновлюємо розмір та місце зображення камери
private void UpdateSize()
{
    int width, height;

    if (!fitToWindow)
    {
        // стандартні ширина та висота
        width = cellWidth;
        height = cellHeight;
    }
    else
    {
        // розраховуємо ширину та висоту камери для відображення
        width = (ClientRectangle.Width / cols) - 4;
        height = (ClientRectangle.Height / rows) - 4;
    }

    // встановлюємо позицію перегляду
    int startX = (ClientRectangle.Width - cols * (width + 4)) / 2;
    int startY = (ClientRectangle.Height - rows * (height + 4)) /
2;

    this.SuspendLayout();

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            camWindows[i, j].Location = new Point(startX +
(width + 4) * j + 1, startY + (height + 4) * i + 1);
            camWindows[i, j].Size = new Size(width + 2, height
+ 2);
        }
    }

    this.ResumeLayout(false);
}

// Змінюємо розмір
private void Multiplexer_Resize(object sender, System.EventArgs e)
{
    UpdateSize();
}

// Миша вимикає працю з камерою
private void cameraWindow_MouseDown(object sender,
System.Windows.Forms.MouseEventHandler e)
{
    lastClicked = (CameraWindow) sender;
}
}
}

```

## Файл VideoStream.cs - робота з відео

```

namespace stream
{
    using System;
    using System.Drawing;
    using System.Drawing.Imaging;
    using System.IO;
    using System.Threading;
    using System.Runtime.InteropServices;
    using System.Net;

    using videosource;
    using dshow;
    using dshow.Core;

    /// <summary>
    /// VideoStream - потік подачі відео
    /// </summary>
    public class VideoStream : IVideoSource
    {
        private string    source;
        private object    userData = null;
        private int       framesReceived;

        private Thread    thread = null;
        private ManualResetEvent stopEvent = null;

        // нова подія у фреймі
        public event CameraEventHandler NewFrame;

        // VideoSource властивості
        public virtual string VideoSource
        {
            get { return source; }
            set { source = value; }
        }
        // Властивості підключення
        public string Login
        {
            get { return null; }
            set { }
        }
        // Властивості паролювання
        public string Password
        {
            get { return null; }
            set { }
        }
        // FramesReceived властивості
        public int FramesReceived
        {
            get
            {
                int frames = framesReceived;
                framesReceived = 0;
                return frames;
            }
        }
        // BytesReceived властивості
        public int BytesReceived
        {
            get { return 0; }
        }
        // UserData властивості
        public object UserData
        {

```

```

        get { return userData; }
        set { userData = value; }
    }
    // Отримуємо стан вихідного відео
    public bool Running
    {
        get
        {
            if (thread != null)
            {
                if (thread.Join(0) == false)
                    return true;

                // Якщо стан не заданий, звільнюємо ресурси
                Free();
            }
            return false;
        }
    }

    // Конструктор
    public VideoStream()
    {
    }

    // Починаємо роботу
    public void Start()
    {
        if (thread == null)
        {
            framesReceived = 0;

            // Створюємо подію
            stopEvent = new ManualResetEvent(false);

            // Створюємо й стартуємо нову подію
            thread = new Thread(new ThreadStart(WorkerThread));
            thread.Name = source;
            thread.Start();
        }
    }

    // Сигнал події до остановки роботи
    public void SignalToStop()
    {
        // Остановлюємо подію
        if (thread != null)
        {
            // Сигнал остановки
            stopEvent.Set();
        }
    }

    // Чекаємо остановки події
    public void WaitForStop()
    {
        if (thread != null)
        {
            // Чекаємо остановки події
            thread.Join();

            Free();
        }
    }

    // Подія помилки
    public void Stop()
    {

```

```

        if (this.Running)
        {
            thread.Abort();
            // WaitForStop();
        }
    }

    // Визволяємо ресурси
    private void Free()
    {
        thread = null;

        // Випуск події
        stopEvent.Close();
        stopEvent = null;
    }

    // Точка входу події
    public void WorkerThread()
    {
        bool failed = false;

        // Граббер
        Grabber grabber = new Grabber(this);

        // Об'єкти
        object graphObj = null;
        object sourceObj = null;
        object grabberObj = null;

        // Інтерфейси
        IGraphBuilder graph = null;
        IBaseFilter sourceBase = null;
        IBaseFilter grabberBase = null;
        ISampleGrabber sg = null;
        IFileSourceFilter fileSource = null;
        IMediaControl mc = null;
        IMediaEventEx mediaEvent = null;

        int code, param1, param2;

        while ((!failed) && (!stopEvent.WaitOne(0, true)))
        {
            try
            {
                // Встановлюємо тип фільтру графіки
                Type srvType =
                    Type.GetTypeFromCLSID(Clsid.FilterGraph);
                if (srvType == null)
                    throw new ApplicationException("Failed
                    creating filter graph");

                // Створюємо фільтр графіки
                graphObj = Activator.CreateInstance(srvType);
                graph = (IGraphBuilder) graphObj;

                // Беремо тип фільтру вікна програвання джерела
                srvType =
                    Type.GetTypeFromCLSID(Clsid.WindowsMediaSource);
                if (srvType == null)
                    throw new ApplicationException("Failed
                    creating WM source");

                // Створюємо вікно програвання джерела
                sourceObj = Activator.CreateInstance(srvType);
                sourceBase = (IBaseFilter) sourceObj;

                // Беремо тип простого грабера

```

```

        srvType =
Type.GetTypeFromCLSID(Clsid.SampleGrabber);
        if (srvType == null)
            throw new ApplicationException("Помилка
створення простого грабера");

        // Створення простого грабера
grabberObj = Activator.CreateInstance(srvType);
sg = (ISampleGrabber) grabberObj;
grabberBase = (IBaseFilter) grabberObj;

        // Додаємо фільтр графічного джерела
graph.AddFilter(sourceBase, "source");
graph.AddFilter(grabberBase, "grabber");

        // Визначаємо тип медіа
AMMediaType mt = new AMMediaType();
mt.majorType = MediaType.Video;
mt.subType = MediaSubType.RGB24;
sg.SetMediaType(mt);

        // Редагуємо файл
fileSource = (IFileSourceFilter) sourceObj;
fileSource.Load(this.source, null);

        // Підключаємо піни
if (graph.Connect(DSTools.GetOutPin(sourceBase,
0), DSTools.GetInPin(grabberBase, 0)) < 0)
    throw new ApplicationException("Failed
connecting filters");

        // Беремо тип медіа
if (sg.GetConnectedMediaType(mt) == 0)
{
    VideoInfoHeader vih = (VideoInfoHeader)
Marshal.PtrToStructure(mt.formatPtr, typeof(VideoInfoHeader));

    grabber.Width = vih.BmiHeader.Width;
    grabber.Height = vih.BmiHeader.Height;
    mt.Dispose();
}

        // рендер
graph.Render(DSTools.GetOutPin(grabberBase, 0));

        //
sg.SetBufferSamples(false);
sg.SetOneShot(false);
sg.SetCallback(grabber, 1);

        // вікно
IVideoWindow win = (IVideoWindow) graphObj;
win.put_AutoShow(false);
win = null;

        // Беремо подію інтерфейсу
mediaEvent = (IMediaEventEx) graphObj;

        // Беремо управління медіа
mc = (IMediaControl) graphObj;

        // запускаємо
mc.Run();

while (!stopEvent.WaitOne(0, true))
{
    Thread.Sleep(100);

    // беремо подію

```

```

param1, out param2, 0) == 0)
    {
        // визначаємо параметри
        mediaEvent.FreeEventParams (code,
        param1, param2);

        //
        if (code == (int) EventCode.Complete)
        {
            break;
        }
    }

    mc.StopWhenReady();
}
// Визначаємо виключення
catch (Exception e)
{
    System.Diagnostics.Debug.WriteLine("----: " +
e.Message);
    failed = true;
}
// Фіналізуємо блок
finally
{
    // визначаємо усі об'єкти
    mediaEvent = null;
    mc = null;
    fileSource = null;
    graph = null;
    sourceBase = null;
    grabberBase = null;
    sg = null;

    if (graphObj != null)
    {
        Marshal.ReleaseComObject (graphObj);
        graphObj = null;
    }
    if (sourceObj != null)
    {
        Marshal.ReleaseComObject (sourceObj);
        sourceObj = null;
    }
    if (grabberObj != null)
    {
        Marshal.ReleaseComObject (grabberObj);
        grabberObj = null;
    }
}
}

// новий фрейм для обробки
protected void OnNewFrame (Bitmap image)
{
    framesReceived++;
    if (NewFrame != null)
        NewFrame (this, new CameraEventArgs (image));
}

// Граббер
private class Grabber : ISampleGrabberCB
{
    private VideoStream parent;
    private int width, height;

```

```

// Width властивості
public int Width
{
    get { return width; }
    set { width = value; }
}
// Height властивості
public int Height
{
    get { return height; }
    set { height = value; }
}

// Конструктор
public Grabber(VideoStream parent)
{
    this.parent = parent;
}

//
public int SampleCB(double SampleTime, IntPtr pSample)
{
    return 0;
}

// Повертаємо метод, який вказує на буфер взірця
public int BufferCB(double SampleTime, IntPtr pBuffer, int
BufferLen)
{
    // створюємо нову картинку
    System.Drawing.Bitmap img = new Bitmap(width, height,
PixelFormat.Format24bppRgb);

    // блокуємо дані бітової площини
    BitmapData bmData = img.LockBits(
        new Rectangle(0, 0, width, height),
        ImageLockMode.ReadWrite,
        PixelFormat.Format24bppRgb);

    // копіюємо дані зображення
    int srcStride = bmData.Stride;
    int dstStride = bmData.Stride;

    int dst = bmData.Scan0.ToInt32() + dstStride * (height -
1);
    int src = pBuffer.ToInt32();

    for (int y = 0; y < height; y++)
    {
        Win32.memcpy(dst, src, srcStride);
        dst -= dstStride;
        src += srcStride;
    }

    // розблокуємо дані бітової площини
    img.UnlockBits(bmData);

    // Увідомляємо батьків
    parent.OnNewFrame(img);

    // Будуємо картинку
    img.Dispose();

    return 0;
}
}
}
}

```

## Файл VideoStreamSetupPage.cs - робота з відео (інтерфейс)

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;
using videosource;

namespace stream
{
    /// <summary>
    /// Основні дескриптори для VideoStreamSetupPage.
    /// </summary>
    public class VideoStreamSetupPage : System.Windows.Forms.UserControl,
    IVideoSourcePage
    {
        private bool completed = false;
        private System.Windows.Forms.TextBox urlBox;
        private System.Windows.Forms.Label label1;
        /// <summary>
        /// Опис змінних розробника.
        /// </summary>
        private System.ComponentModel.Container components = null;

        // стан змінюваної події
        public event EventHandler StateChanged;

        // Конструктор
        public VideoStreamSetupPage()
        {
            // Цей виклик використовується у Windows.Forms Form Designer.
            InitializeComponent();
        }

        /// <summary>
        /// Очищуємо усі ресурси використовувани користувачем.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if( disposing )
            {
                if(components != null)
                {
                    components.Dispose();
                }
            }
            base.Dispose( disposing );
        }

        #region Component Designer generated code
        /// <summary>
        /// Необхідний метод для підтримки розробника - не модифікується
        /// контент цього методу з редактором коду.
        /// </summary>
        private void InitializeComponent()
        {
            this.urlBox = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // urlBox
            //
            this.urlBox.Anchor = ((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)

```

```

        | System.Windows.Forms.AnchorStyles.Right);
this.urlBox.Location = new System.Drawing.Point(50, 10);
this.urlBox.Name = "urlBox";
this.urlBox.Size = new System.Drawing.Size(240, 20);
this.urlBox.TabIndex = 1;
this.urlBox.Text = "";
this.urlBox.TextChanged += new
System.EventHandler(this.urlBox_TextChanged);
//
// label1
//
this.label1.Location = new System.Drawing.Point(10, 13);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(30, 14);
this.label1.TabIndex = 0;
this.label1.Text = "&URL:";
//
// VideoStreamSetupPage
//
this.Controls.AddRange(new System.Windows.Forms.Control[] {

        this.urlBox,

        this.label1});
this.Name = "VideoStreamSetupPage";
this.Size = new System.Drawing.Size(300, 150);
this.ResumeLayout(false);

}
#endregion

// Completed властивості
public bool Completed
{
    get { return completed; }
}

// Показуємо сторінку
public void Display()
{
    urlBox.Focus();
    urlBox.SelectionStart = urlBox.TextLength;
}

// Додаємо сторінку
public bool Apply()
{
    return true;
}

// Конфігуруємо об'єкт зображення
public object GetConfiguration()
{
    StreamConfiguration config = new StreamConfiguration();

    config.source = urlBox.Text;

    return (object) config;
}

// Встановлюємо конфігурацію
public void SetConfiguration(object config)
{
    StreamConfiguration cfg = (StreamConfiguration) config;

    if (cfg != null)
    {
        urlBox.Text = cfg.source;
    }
}

```

```
}  
  
// Змінюємо URL  
private void urlBox_TextChanged(object sender, System.EventArgs e)  
{  
    completed = (urlBox.TextLength != 0);  
  
    if (StateChanged != null)  
        StateChanged(this, new EventArgs());  
}  
}  
}
```

КБПЗ\_2024

Файл `CUDASource.cs` - алгоритм кодування CUDA

```

namespace CUDA
{
    using System;
    using System.Drawing;
    using System.IO;
    using System.Threading;
    using System.Net;

    using videosource;

    /// <summary>
    /// CUDASource - CUDA скачувач
    /// </summary>
    public class CUDASource : IVideoSource
    {
        private string    source;
        private string    login = null;
        private string    password = null;
        private object    userData = null;
        private int       framesReceived;
        private int       bytesReceived;
        private bool      useSeparateConnectionGroup = false;
        private bool      preventCaching = false;
        private int       frameInterval = 0;           // інтервал подання
        фреймів у мілісекундах

        private const int bufSize = 512 * 1024;      // розмір буферу
        private const int readSize = 1024;         // розмір блоку для читання

        private Thread    thread = null;
        private ManualResetEvent stopEvent = null;

        // нова подія у фреймі
        public event CameraEventHandler NewFrame;

        // SeparateConnectionGroup властивості
        // indicates to open WebRequest in separate connection group
        public bool SeparateConnectionGroup
        {
            get { return useSeparateConnectionGroup; }
            set { useSeparateConnectionGroup = value; }
        }

        // PreventCaching властивості
        // Якщо властивості є правильними, то керуємо параметри URL. Цей
        клієнт повинен бути встановлений на проксі-сервері.
        public bool PreventCaching
        {
            get { return preventCaching; }
            set { preventCaching = value; }
        }

        // FrameInterval властивості - інтервал між фреймами
        Якщо властивості встановлені в 100, тоді джерело формує 10 фреймів
        // в секунду
        public int FrameInterval
        {
            get { return frameInterval; }
            set { frameInterval = value; }
        }

        // VideoSource властивості
        public virtual string VideoSource
        {
            get { return source; }
            set { source = value; }
        }

        // Властивості підключення
    }
}

```

```

public string Login
{
    get { return login; }
    set { login = value; }
}
// Властивості паролювання
public string Password
{
    get { return password; }
    set { password = value; }
}
// FramesReceived властивості
public int FramesReceived
{
    get
    {
        int frames = framesReceived;
        framesReceived = 0;
        return frames;
    }
}
// BytesReceived властивості
public int BytesReceived
{
    get
    {
        int bytes = bytesReceived;
        bytesReceived = 0;
        return bytes;
    }
}
// UserData властивості
public object UserData
{
    get { return userData; }
    set { userData = value; }
}
// Отримуємо стан вихідного відео
public bool Running
{
    get
    {
        if (thread != null)
        {
            if (thread.Join(0) == false)
                return true;

            // Якщо стан не заданий, звільнюємо ресурси
            Free();
        }
        return false;
    }
}

// Конструктор
public CUDASource()
{
}

// Починаємо роботу
public void Start()
{
    if (thread == null)
    {
        framesReceived = 0;
        bytesReceived = 0;

        // Створюємо подію
        stopEvent = new ManualResetEvent(false);
    }
}

```

```

        // Створюємо й стартуємо нову подію
        thread = new Thread(new ThreadStart(WorkerThread));
        thread.Name = source;
        thread.Start();
    }
}

// Сигнал події до остановки роботи
public void SignalToStop()
{
    // Остановлюємо подію
    if (thread != null)
    {
        // Сигнал остановки
        stopEvent.Set();
    }
}

// Чекаємо остановки події
public void WaitForStop()
{
    if (thread != null)
    {
        // Чекаємо остановки події
        thread.Join();

        Free();
    }
}

// Подія помилки
public void Stop()
{
    if (this.Running)
    {
        thread.Abort();
        WaitForStop();
    }
}

// Визволяємо ресурси
private void Free()
{
    thread = null;

    // Випуск події
    stopEvent.Close();
    stopEvent = null;
}

// Точка входу події
public void WorkerThread()
{
    byte[] buffer = new byte[bufSize]; // буфер
читання потоку
    HttpRequest req = null;
    WebResponse resp = null;
    Stream stream = null;
    Random rnd = new Random((int)
DateTime.Now.Ticks);
    DateTime start;
    TimeSpan span;

    while (true)
    {
        int read, total = 0;

        try

```

```

{
    start = DateTime.Now;

    // створюємо запит
    if (!preventCaching)
    {
        req = (HttpWebRequest)
WebRequest.Create(source);
    }
    else
    {
        req = (HttpWebRequest)
WebRequest.Create(source + ((source.IndexOf('?') == -1) ? '?' : '&') + "fake=" +
rnd.Next().ToString());
    }
    // встановлюємо логін та пароль
    if ((login != null) && (password != null) &&
(login != ""))
        req.Credentials = new
NetworkCredential(login, password);
    // встановлюємо найменування групи підключення
    if (useSeparateConnectionGroup)
        req.ConnectionGroupName =
GetHashCode().ToString();

    // отримуємо відповідь
    resp = req.GetResponse();

    // отримуємо відповідь потоку
    stream = resp.GetResponseStream();

    // цикл
    while (!stopEvent.WaitOne(0, true))
    {
        // перевіряємо загальне читання
        if (total > bufSize - readSize)
        {
            total = 0;
        }

        // Читаємо наступний блок у потоці
        if ((read = stream.Read(buffer, total,
readSize)) == 0)
            break;
        total += read;
        // Додаємо лічильник зчитаних байт
        bytesReceived += read;
    }
    if (!stopEvent.WaitOne(0, true))
    {
        // додаємо лічильник фреймів
        framesReceived++;
        // остановка читання зображення
        if (NewFrame != null)
        {
            Bitmap bmp = (Bitmap)
Bitmap.FromStream(new MemoryStream(buffer, 0, total));
            // Клієнт увідомлення
            NewFrame(this, new
CameraEventArgs(bmp));

            // Будуємо картинку
            bmp.Dispose();
            bmp = null;
        }
    }
    // Чекаємо в циклі ?
    if (frameInterval > 0)
    {
        // діапазон часу
        span = DateTime.Now.Subtract(start);
    }
}

```



Файл `CUDASourcePage.cs` – алгоритм кодування CUDA (інтерфейс)

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;
using videosource;

namespace CUDA
{
    /// <summary>
    /// Основні дескриптори для CUDASourcePage.
    /// </summary>
    public class CUDASourcePage : System.Windows.Forms.UserControl,
IVideoSourcePage
    {
        private static int[] frameIntervals = new int[] {0, 100, 142, 200,
333, 1000,
                    5000, 10000, 15000, 20000, 30000, 60000};
        private bool completed = false;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox urlBox;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox loginBox;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.TextBox passwordBox;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.ComboBox rateCombo;
        /// <summary>
        /// Опис змінних розробника.
        /// </summary>
        private System.ComponentModel.Container components = null;

        // стан змінюваної події
        public event EventHandler StateChanged;

        // Конструктор
        public CUDASourcePage()
        {
            // Цей виклик використовується у Windows.Forms Form Designer.
            InitializeComponent();

            //
            rateCombo.SelectedIndex = 0;
        }

        /// <summary>
        /// Очищуємо усі ресурси використовувани користувачем.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if( disposing )
            {
                if(components != null)
                {
                    components.Dispose();
                }
            }
            base.Dispose( disposing );
        }

        #region Component Designer generated code
        /// <summary>
        /// Необхідний метод для підтримки розробника - не модифікується
        /// контент цього методу з редактором коду.

```

```

/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.urlBox = new System.Windows.Forms.TextBox();
    this.label2 = new System.Windows.Forms.Label();
    this.loginBox = new System.Windows.Forms.TextBox();
    this.label3 = new System.Windows.Forms.Label();
    this.passwordBox = new System.Windows.Forms.TextBox();
    this.label4 = new System.Windows.Forms.Label();
    this.rateCombo = new System.Windows.Forms.ComboBox();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.Location = new System.Drawing.Point(10, 13);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(41, 14);
    this.label1.TabIndex = 0;
    this.label1.Text = "&URL:";
    //
    // urlBox
    //
    this.urlBox.Anchor = ((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right);
    this.urlBox.Location = new System.Drawing.Point(70, 10);
    this.urlBox.Name = "urlBox";
    this.urlBox.Size = new System.Drawing.Size(220, 20);
    this.urlBox.TabIndex = 1;
    this.urlBox.Text = "";
    this.urlBox.TextChanged += new
System.EventHandler(this.urlBox_TextChanged);
    //
    // label2
    //
    this.label2.Location = new System.Drawing.Point(10, 43);
    this.label2.Name = "label2";
    this.label2.Size = new System.Drawing.Size(35, 14);
    this.label2.TabIndex = 2;
    this.label2.Text = "&Login:";
    //
    // loginBox
    //
    this.loginBox.Anchor = ((System.Windows.Forms.AnchorStyles.Top
| System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right);
    this.loginBox.Location = new System.Drawing.Point(70, 40);
    this.loginBox.Name = "loginBox";
    this.loginBox.Size = new System.Drawing.Size(220, 20);
    this.loginBox.TabIndex = 3;
    this.loginBox.Text = "";
    //
    // label3
    //
    this.label3.Location = new System.Drawing.Point(10, 73);
    this.label3.Name = "label3";
    this.label3.Size = new System.Drawing.Size(60, 14);
    this.label3.TabIndex = 4;
    this.label3.Text = "&Password:";
    //
    // passwordBox
    //
    this.passwordBox.Anchor =
((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right);
    this.passwordBox.Location = new System.Drawing.Point(70, 70);
    this.passwordBox.Name = "passwordBox";

```

```

this.passwordBox.Size = new System.Drawing.Size(220, 20);
this.passwordBox.TabIndex = 5;
this.passwordBox.Text = "";
//
// label4
//
this.label4.Location = new System.Drawing.Point(10, 103);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(63, 14);
this.label4.TabIndex = 6;
this.label4.Text = "&Frame rate:";
//
// rateCombo
//
this.rateCombo.Anchor =
((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right);
this.rateCombo.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
this.rateCombo.Items.AddRange(new object[] {

    "Uncontrolled",

    "10 фреймів у секунду",

    "7 фреймів у секунду",

    "5 фреймів у секунду",

    "3 фреймів у секунду",

    "1 фреймів у секунду",

    "12 фреймів у хвилину",

    "6 фреймів у хвилину",

    "4 фреймів у хвилину",

    "3 фреймів у хвилину",

    "2 фреймів у хвилину",

    "1 фреймів у хвилину"});
this.rateCombo.Location = new System.Drawing.Point(70, 100);
this.rateCombo.Name = "rateCombo";
this.rateCombo.Size = new System.Drawing.Size(220, 21);
this.rateCombo.TabIndex = 7;
//
// CUDASourcePage
//
this.Controls.AddRange(new System.Windows.Forms.Control[] {

    this.rateCombo,

    this.label4,

    this.passwordBox,

    this.label3,

    this.loginBox,

    this.label2,

    this.urlBox,

    this.label1});

```

```

        this.Name = "CUDASourcePage";
        this.Size = new System.Drawing.Size(300, 150);
        this.ResumeLayout(false);

    }
    #endregion

    // Completed властивості
    public bool Completed
    {
        get { return completed; }
    }

    // Показуємо сторінку
    public void Display()
    {
        urlBox.Focus();
        urlBox.SelectionStart = urlBox.TextLength;
    }

    // Додаємо сторінку
    public bool Apply()
    {
        return true;
    }

    // Конфігуруємо об'єкт зображення
    public object GetConfiguration()
    {
        CUDAConfiguration config = new CUDAConfiguration();

        config.source      = urlBox.Text;
        config.login       = loginBox.Text;
        config.password    = passwordBox.Text;
        config.frameInterval =
frameIntervals[rateCombo.SelectedIndex];

        return (object) config;
    }

    // Встановлюємо конфігурацію
    public void SetConfiguration(object config)
    {
        CUDAConfiguration cfg = (CUDAConfiguration) config;

        if (cfg != null)
        {
            urlBox.Text = cfg.source;
            loginBox.Text = cfg.login;
            passwordBox.Text = cfg.password;
            rateCombo.SelectedIndex = Array.IndexOf(frameIntervals,
cfg.frameInterval);
        }
    }

    // Змінюємо URL
    private void urlBox_TextChanged(object sender, System.EventArgs e)
    {
        completed = (urlBox.TextLength != 0);

        if (StateChanged != null)
            StateChanged(this, new EventArgs());
    }
}
}

```

## Файл About.cs - довідка

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;

namespace CameraAbout {
    public class About : System.Windows.Forms.Form {

        #region system stuff
        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.RichTextBox richTextBox1;
        private System.Windows.Forms.Button button1;
        private System.ComponentModel.IContainer components = null;

        public About() {
            InitializeComponent();
        }

        protected override void Dispose( bool disposing ) {
            if( disposing ) {
                if(components != null) {
                    components.Dispose();
                }
            }
            base.Dispose( disposing );
        }
        #endregion

        #region Windows Form Designer generated code
        /// <summary>
        /// Необхідний метод для підтримки розробника - не модифікується
        /// контент цього методу з редактором коду.
        /// </summary>
        private void InitializeComponent() {
            System.Resources.ResourceManager resources = new
System.Resources.ResourceManager( typeof( About ) );
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.richTextBox1 = new System.Windows.Forms.RichTextBox();
            this.button1 = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // pictureBox1
            //
            this.pictureBox1.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
            this.pictureBox1.Location = new System.Drawing.Point(8, 8);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(152, 192);
            this.pictureBox1.TabIndex = 0;
            this.pictureBox1.TabStop = false;
            //
            // richTextBox1
            //
            this.richTextBox1.BackColor = System.Drawing.Color.Black;
            this.richTextBox1.BorderStyle =
System.Windows.Forms.BorderStyle.None;
            this.richTextBox1.ForeColor = System.Drawing.Color.White;
            this.richTextBox1.Location = new System.Drawing.Point(176,
16);

            this.richTextBox1.Name = "richTextBox1";
            this.richTextBox1.Size = new System.Drawing.Size(304, 184);
            this.richTextBox1.TabIndex = 1;
            this.richTextBox1.Text = @"БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

```

на тему:

Дослідження та програмна реалізація системи керування гучномовцями в системах відеоспостереження

Керівник: Усік П.С.

Розробив: студент Осуський Артем Вікторович  
гр. КІ-23М

```

м. Кропивницький 2024";
    //
    // button1
    //
    this.button1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
    this.button1.ForeColor = System.Drawing.Color.White;
    this.button1.Location = new System.Drawing.Point(16, 168);
    this.button1.Name = "button1";
    this.button1.Size = new System.Drawing.Size(136, 24);
    this.button1.TabIndex = 2;
    this.button1.Text = "&Close";
    this.button1.Click += new
System.EventHandler(this.button1_Click);
    //
    // About
    //
    this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
    this.BackColor = System.Drawing.Color.Black;
    this.ClientSize = new System.Drawing.Size(480, 208);
    this.Controls.Add(this.button1);
    this.Controls.Add(this.richTextBox1);
    this.Controls.Add(this.pictureBox1);
    this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.None;
    this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
    this.Name = "About";
    this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
    this.Text = "About";
    this.ResumeLayout(false);

    }
    #endregion

    #region events

    private void button1_Click(object sender, System.EventArgs e) {
        this.Close();
    }

    #endregion
}
}

```