

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 оцінки**  
**реалізації відео-конференц-зв’язку”**

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

Керівник проекту  
доктор технічних наук, професор  
\_\_\_\_\_ Коваленко О.В.  
« \_\_\_\_ » \_\_\_\_\_ 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

*Москальову Антону Віталійовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

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

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку*

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

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

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

<i>Наукова новизна</i>	<i>1 аркуш</i>
<i>Структурна схема системи</i>	<i>1 аркуш</i>
<i>Функціональна схема системи</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>
<i>Показники економічної ефективності</i>	<i>1 аркуш</i>

## 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

**Москальов А.В. Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку. 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 оцінки реалізації відео-конференц-зв'язку.

Метою розробки є дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.

Об'єктом дослідження є процес оцінки реалізації відео-конференц-зв'язку.

Предметом дослідження є методи оцінки реалізації відео-конференц-зв'язку.

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

Результат роботи – програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.

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, відео-конференц-зв'язок

## ABSTRACT

**Moskalyov A.V. Research and software implementation of the system for evaluating the implementation of video conferencing. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the evaluation system of the implementation of video-conferencing.

The purpose of the development is the research and software implementation of the system for evaluating the implementation of video-conference communication.

The object of the study is the process of evaluating the implementation of video conferencing.

The subject of the study is methods of evaluating the implementation of video-conference communication.

Research methods are based on coding theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the system for evaluating the implementation of video-conference communication.

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

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

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

The program was developed in the Visual C# environment.

**Keywords:** computer engineering, video conferencing

## 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 .....	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	12
2.3 Розгорнута постановка завдання .....	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	17
3.1 Опис функціонування системи .....	17
3.2 Розробка структурної схеми.....	21
3.3 Розробка функціональної схеми .....	30
3.4 Розробка діаграми процесів.....	42
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	44
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	44
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	62
6 НАУКОВА НОВИЗНА .....	66

					ВКРМ-123.24.0025.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку	Літ.	Аркуш	Аркушів
Розроб.	Москальов А.В.					М	1	90
Перев.	Коваленко О.В.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	67
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	67
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	68
7.3	Вибір методу оцінки вартості ПЗ .....	69
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	70
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	71
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 санітарно-гігієнічних умов праці на робочому місці програміста ...	76
8.4	Розробка заходів з умов поліпшення охорони праці .....	79
8.5	Розрахункова частина .....	80
9	ОСНОВНІ ВИСНОВКИ.....	84
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	86

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

КБПЗ – 2024

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Ефективність відео-конференц-зв'язку (ВКЗ) звичайно оцінюють за економічними показниками – зниженню витрат на відрядження, дистанційному проведенню навчальних семінарів і т.д. Разом з тим відео-конференц-зв'язок допомагає більш оперативно обмінюватися інформацією, скоротити час прийняття колегіальних рішень і надає інші переваги, які досить важко оцінити в грошовому еквіваленті.

По даним IDC, в I кварталі 2024 року світовий обсяг продажів устаткування для відеоконференцій знизився на 16% у порівнянні з минулим роком, склавши 473,5 млн доларів. Кількість проданих систем зменшилося в цілому на 6,2%. Продажу мультимедійних систем телеприсутності впали в грошовому вираженні на 33,5%, а в кількісному на 26%, тоді як кімнатних відеосистем – на 10 і 1%.

Проте, по оцінках різних виробників, уже через два роки в Північній Америці і Європі системи ВКЗ стануть використовувати у своїй роботі 50% співробітників різних компаній, а відеозв'язок буде превалювати над поштою й мобільними дзвінками.

Згідно даним Wainhouse Research, за підсумками 2023 року обсяг українського ринку ВКЗ склав 130 млн доларів (приблизно 1/70 глобального ринку), при цьому темпи його розвитку істотно перевищували світові показники. Для порівняння, в 2021 році обсяг ринку ВКЗ в Україні не перевищував 40 млн доларів. У середньому аналітики прогнозували ріст ринку ВКЗ в Україні на 35–40%, в 2022–2023 роках він почав уповільнюватися й потім практично зупинився (у грошовому вираженні).

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.

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

- Огляд існуючих систем оцінки реалізації відео-конференц-зв'язку.

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Дослідження системи оцінки реалізації відео-конференц-зв'язку.
- Програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.

*Об'єктом дослідження* є процес оцінки реалізації відео-конференц-зв'язку.

*Предметом дослідження* є методи оцінки реалізації відео-конференц-зв'язку.

*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 оцінки реалізації відео-конференц-зв'язку.
- Розроблено вітчизняний продукт оцінки реалізації відео-конференц-зв'язку, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі оцінки реалізації відео-конференц-зв'язку.

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

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

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Відзначається помітне розширення використання систем ВКЗ на рівні клієнтських додатків і переговорних кімнат. Цьому факту можна знайти кілька пояснень. По-перше, вартість систем знижується. По-друге, такі рішення, як Skype і Facetime, допомогли користувачам освоїти відеоспілкування. Психологічний бар'єр переборений – люди поступово звикають дивитися один на одного через екран.

При створенні системи ВКЗ надто важливо правильно поставити завдання проєктувальникам, для чого необхідно врахувати основні критерії, що впливають на вартість системи ВКЗ:

- ціна устаткування;
- вартість каналів зв'язку (у тому числі розширення каналів зв'язку, необхідних для підтримки відео);
- витрати на навчання й зміст співробітників відділу експлуатації;
- плата за сервісну підтримку вендора.

Найчастіше замовник намагається оптимізувати витрати на устаткування, забуваючи, що пропускна здатність каналів зв'язку, необхідних для експлуатації ВКЗ на базі, наприклад, програмного забезпечення або устаткування маловідомих виробників, у два-три рази більше, ніж у каналу зв'язку, що забезпечує одержання зображення того ж або переважаючої якості на базі апаратного рішення одного з ведучих вендорів. При цьому витрати на устаткування – єдино-часові, а оплата трафіку – регулярна.

А часові витрати на обслуговування системи ВКЗ і підтримка її в працездатному стані в зазначених варіантах просто непорівнянні. У випадку системи ВКЗ на базі апаратного рішення досягається максимальна автоматизація

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

процесу – аж до створення багатоточкової відеоконференції в заздалегідь заданий період з резервуванням ключових вузлів системи й пошуком не абонента, що відповідає на виклик.

## 1.2 Область застосування

Відеоконференція застосовується як засіб оперативного ухвалення рішення в тих або інших випадках: при надзвичайних ситуаціях; для скорочення відрядних витрат у територіально розподілених організаціях; підвищення ефективності бізнесу; проведення судових процесів з дистанційною участю засуджених, а також як один з елементів технологій телемедицини й дистанційного навчання.

У багатьох державних і комерційних організаціях відеоконференція приносить істотну користь і забезпечує максимальну ефективність, а саме:

- зменшується число відряджень і знижуються пов'язані з ними витрати;
- прискорюються процеси прийняття рішень у надзвичайних ситуаціях;
- скорочується час розгляду справ у судах загальної юрисдикції;
- збільшується продуктивність праці;
- вирішуються кадрові проблеми й поліпшується соціально-економічна ситуація;
- надається можливість приймати більше обґрунтовані рішення за рахунок залучення при необхідності додаткових експертів;
- швидко й ефективно розподіляються ресурси;
- забезпечується можливість миттєвої організації наради «на вимогу» і швидкого ухвалення рішення з повноцінним обміном інформацією й візуальним контактом;
- створюються умови для проведення оперативних нарад із залученням при необхідності експертів;

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- підтримується спілкування не тільки усередині організації, але й із замовниками, партнерами й постачальниками по усьому світі;
- відкривається доступ до участі в нараді для тих співробітників, хто перебуває в поїзді поза офісом або у відрядженні.

Відповідно, можна зробити висновок, що ВКЗ – це підвищення оперативності бізнесу і якості прийняття рішень плюс об'єднання територіально розподіленої організації.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У цьому розділі ми розглянемо представлені на українському ринку моделі групових систем ВКЗ і виділимо найбільш перспективні. Для аналізу ми вибрали по двох групі системи початкового й бізнес рівня п'яти виробників: Polycom, Tandberg, Aethra, Sony і AddPac. Системи одного класу мають схожі технічні характеристики, але різною ціною, тому однієї із ключових завдань порівняння є виявлення моделей з найкращим співвідношенням «ціна/функціональні можливості».

#### **Aethra Vega X3**

Всі групові системи Aethra працюють абсолютно безшумно через відсутність вентиляторів. Система компактна й надійна, підтримує всі необхідні кодеки й стандарти, сумісна з устаткуванням інших виробників. Недавно з'явився убудований MCU підтримує до чотирьох абонентів (відеокодек H.263). Що стосується ціни, те це найдешевша модель серед розглянутих нами. З додаткових можливостей можна виділити інтеграцію із сервісами NetMeeting. Aethra Vega X3 має невелике число відеовходів і відеовиходів, однак у ряді систем початкового рівня лише вона підтримує підключення через DVI.

#### **Sony PCS-1P**

Дана модель сама функціональна, причому деякі можливості відрізняються від стандартних. Зокрема, є два входи XGA (опціонально), підтримується кодек MPEG-4 (на жаль, його можна використовувати тільки із системами ВКЗ від Sony), убудований MCU передбачає підключення до шести абонентів (відеокодек H.263), з декількох систем можна організувати каскад. При

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

досить конкурентоспроможній ціні PCS-1P має краще співвідношення «ціна/функціональні можливості» у своєї «вагарні» категорії. На жаль, незначні недоліки псують всю картину. І головні з них – невибагливий зовнішній вигляд і порівняно високий рівень шуму.

### **Polycom VSX 6000**

Груповою системою VSX 6000 дозволяє зберегти зроблені вкладення в довгостроковій перспективі. За допомогою конференц-телефону Polycom SoundStation VTX1000 можна побудувати систему аудіо-конференц-зв'язку, а потім розширити її за допомогою VSX 6000 до повноцінної системи ВКЗ. При цьому конференц-телефон інтегрується в структуру ВКЗ і служить як зовнішній мікрофон. З інших особливостей можна відзначити підтримку фірмового аудіокодеку Polycom StereoSurround (використовується тільки в системах Polycom). Головний мінус системи – надмірно висока ціна. При цьому модель не виділяється ні високими швидкостями з'єднання по IP, ні більшим числом відеовходів (немає навіть входу XGA для підключення ПК). Більше того, VSX 6000 не відрізняється стабільністю в роботі.

### **Tandberg 550 MXP**

Як і всі системи ВКЗ Tandberg, групова система 550 MXP досить надійна в експлуатації. Але це, мабуть, єдине достоїнство даної моделі, тому що при мінімальному функціоналі вона має досить високу вартість. Як наслідок, Tandberg 550 MXP має найгірше співвідношення «ціна/функціональні можливості». Наявність такої опції, як порт USB, нічого не міняє.

### **AddPac VC-1000**

Компанія AddPac – новий гравець на ринку групових систем відео-конференц-зв'язку. Система початкового рівня AddPac VC-1000 пропонує цікаві (у чомусь трохи дивні) можливості. Зокрема, вона забезпечує найвищу серед аналогів швидкість з'єднання по IP – до 4 Мбіт/с, при цьому підтримує не стандартний дозвіл 4CIF (704 x 576), а комп'ютерне VGA (640 x 480). Функція DualVideo (стандарт H.239) не реалізована, але уведено два нових, не сумісних з

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

іншими, відеокодеку. З аудіокодеків підтримуються тільки самі популярні різновиди. Порт USB придатний для підключення не тільки адаптера WiFi (як у конкурентів), але й різноманітної периферії: від клавіатури й миші до зовнішнього HDD. Все це наводить на думку, що при розробці VC-1000 виробник вирішив зробити акцент на сумісність у першу чергу із власними системами ВКЗ.

### **Aethra Vega X5**

Як і модель початкового рівня, Vega X5 відрізняє компактність, безшумність і надійність. На наш погляд, ця групова система ВКЗ є самої стильною (до її розробки залучалися італійські дизайнери). Вона має найбільшу швидкість з'єднання по IP (до 4 Мбіт/с) і убудованим сервером MCU на вісім абонентів. Крім того, в Aethra Vega X5 реалізована функція автоматичного наведення камери й унікальна опція вибору розкладки екрана. Остання передбачає 12 варіантів розміщення «вікон» на екрані, у яких відображаються вилучені абоненти. За своєю ціною ця система найбільш доступна у своєму класі й має краще співвідношення ціна/функціональні можливості.

### **Sony PCS-G70 VP**

У порівнянні з моделлю початкового рівня, Sony PCS-G70 VP має більше привабливий дизайн, надійністю в роботі, максимальної (4 Мбіт/с) швидкістю з'єднання по IP. Вона має дев'ять відеовиходів (на всі випадки життя й ще тричотири про запас). MCU підтримує ті ж шість абонентів, але зате додана функція автоматичного наведення камери. У цілому відмінності від PCS-1P невеликі – головним чином, вони складаються в усуненні недоліків молодшої моделі. При цьому вартість виросла майже вдвічі, що зробило PCS-G70 VP однією з найдорожчих систем бізнесу-класу.

### **Polycom VSX 7000s**

Компанія Polycom є лідером ринку ВКЗ, однак можливості цієї моделі не вражають: швидкість з'єднання по IP до 2 Мбіт/с, убудований MCU до чотирьох абонентів і власний аудіокодек Polycom StereoSurround. Середньому функціоналові відповідає середня ціна.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

## **Tandberg 990 MXP**

Як і модель початкового рівня, Tandberg 990 MX характеризується високою надійністю й посередніми можливостями, як в Polycom VSX 7000s (MCU до чотирьох абонентів, швидкість з'єднання по IP до 2 Мбіт/с). Зовнішній вигляд копіює молодшу модель 550 MXP. При цьому вона виявилася самою дорогою серед своїх аналогів, а тому має найнижче співвідношення «ціна/функціональні можливості».

## **AddPac VC-2000.**

Групова система VC-2000 незначно відрізняється від моделі початкового рівня. Крім додавання MCU на чотири абонентів було збільшене число відеовходів/відеовиходів, швидкість з'єднання по IP як і раніше гранична для систем такого рівня – до 4 Мбіт/с. Таким чином, AddPac VC-2000 потенційно може стати найпривабливішою (з погляду «ціна/функціональні можливості») груповою системою ВКЗ на українському ринку.

## **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, покликані спростити розробку додатків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю визнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожну із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи більше високий рівень безпеки й продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власне поводження ітерації, що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод `Main` – точку входу додатка – інкапсулюється у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримує спадкування.

На додаток до основних описаних об'єктно-орієнтованих принципів, мова Visual C# спрощує розробку компонентів програмного забезпечення завдяки декільком інноваційним конструкціям мови, у число яких входять наступні:

- Інкапсульовані підписи методів, називані делегатами, які підтримують строго-типізовані повідомлення про події.
- Властивості, що виступають у ролі методів доступу для закритих змінних-членів.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

забезпечення, яке призначено для системи оцінки реалізації відео-конференц-зв'язку.

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-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Ефективність – болюче питання для українського бізнесу. Дуже багато хто вимовляють це слово, але далеко не всі розуміють, що за ним коштує. Ефективність ВКЗ звичайно оцінюють за економічними показниками – зниженню витрат на відрядження, дистанційне проведення семінарів і т.д.

Безумовно, такі розрахунки є. Вони показують, що застосування ВКЗ приносить і очевидну вигоду – окупність проекту протягом 6-14 місяців зараз уже нікого не дивує.

Наявні інструменти економічної оцінки впровадження подібних рішень дозволяють порахувати, наприклад, коефіцієнт повернення інвестицій за трирічний період. Але в кожному разі це лише приблизні розрахунки, а реальні показники повинні бути засновані на певних метриках, які фіксуються протягом якогось часу до впровадження й потім, на постійній основі, після. Проблема в тім, що ніхто – навіть досить «просунуті» компанії – не займається збором цих метрик. Тому, якщо запитати співробітників підприємств, у яких була розгорнута система ВКЗ, що ж змінилося, то з великою ймовірністю ми почуємо масу епітетів і ні однієї, хоча б приблизної, чисельної оцінки.

Для оцінки ефективності будь-якого проекту з погляду віддачі інвестицій існують загальноприйняті методики, наприклад розрахунок ROI. Однак проекти впровадження ВКЗ, крім скорочення витрат на відрядні витрати, які можуть бути враховані в явному виді, націлені на підвищення ефективності бізнесу в цілому. Відео-конференц-зв'язок дозволяє більш оперативно обмінюватися інформацією, скоротити час прийняття колегіальних рішень і надає інші переваги, які досить важко оцінити в грошовому еквіваленті.

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Найчастіше розробка методології й розрахунок ефективності ВКЗ самі по собі є проектом, що вимагає окремих фінансових, часових і експертних ресурсів. Можливо, саме тому нам не доводилося зіштовхуватися з випадками, коли українська компанія намагалася б оцінити впровадження ВКЗ як інвестиційний проект.

Кілька слів хочеться приділити конкретним показникам, які приймаються в розрахунок при оцінці рішення ВКЗ. Наприклад, можна скористатися закордонним досвідом.

### **Фактор 1. Економія часу**

Потенціал зниження часових витрат – один з найбільш переконливих доводів на користь використання відеоконференцій. Найчастіше при організації зустрічей з їхньою допомогою цей фактор є вирішальним і переважає навіть вигоду від прямого скорочення фінансових витрат.

Відповідно до дослідження WorldCom, щоб взяти участь у типових зборах, що передбачають поїздку співробітника з іншого міста або країни, у середньому витрачається 21 год на переліт, трансфер, підготовку до виступу, саме збори й заходи, що проводяться по його закінченні. Як затверджується, при використанні відеозв'язку та ж людина на ті ж мети витрачає лише 4 год свої часи – тим самим заощаджується приблизно 17 год. Природно, ці дані не можна назвати безумовними, тому що під час подорожі можна виконувати деякі робочі завдання за допомогою ноутбуків, PDA-Пристроїв і інших інструментів, і ми обговоримо це нижче. У тому же дослідженні відзначається, що в середньому в зборах беруть участь п'ять співробітників і чотирьом з них доводиться відправлятися у відрядження. Отже, відеоконференція дозволить заощадити 68 год робочого часу. По даним WorldCom, у США кількість нарад, які щомісяця відвідує співробітник, що робить ділові поїздки, становить у середньому 4,6. Це означає, що якщо в 50% випадків він зможе замість поїздки скористатися відеоконференцією, то заощадить 39,1 год на місяць. Інакше кажучи, обсяг його корисного робочого часу збільшиться на 20%!

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

## **Фактор 2. Зниження витрат**

Відеоконференції забезпечують значне зниження витрат – як у явному виді (переліт, готель і т.д.), так і непряме (зкономлений час співробітників).

За даними опитування WorldCom, поїздка на ділову нараду в межах США спричиняє явні витрати в розмірі 1334 долара на один учасника. Кожний такий співробітник обходиться компанії на 30–50% (приблизно 50 доларів у годину) дорожче того, хто залишається в офісі, тобто непряма вигода при заміні поїздки на відеоконференцію становить 850 доларів для кожного учасника. Якщо об'єднати пряму й непрямую економію, те, згідно даним WorldCom, скасування одного відрядження знижує витрати в середньому на 2184 долара, а заміна наради на відеоконференцію – на більш ніж 8 тис. доларів.

## **Фактор 3. Якість життя**

Очевидно, що фінансові мотиви інвестування у відеоконференції привертають найбільшу увагу осіб, відповідальних за прийняття рішень. Однак багато співробітників вважають, що й інші переваги не менш важливі. На міжнародному симпозіумі по стресу в 2000 році був представлений докладний звіт про стреси, пов'язаних з діловими поїздками.

Як показали дослідження, більше 73% співробітників, що роблять такі поїздки, вважають їх причиною стресу. Більше того, понад половину із цих 73% указують на них як на надзвичайно сильне джерело стресу, що негативно позначається на якості життя, сні, здоров'ї й загальній працездатності, причому цей вплив відчувається як до, так і після поїздки. Мало того, стресу піддаються не тільки самі мандрівники, але й члени їхніх родин. Згідно деяким наведеним на симпозіумі даним, майже 100% подружжя або інших родичів затверджують, що ділові поїздки підривають стабільність їхнього сімейного життя й роблять їх менш щасливими.

Ще один цікавий факт: 76% співробітників, що регулярно відправляються у відрядження, повідомляють, що під час поїздок вони частіше випробовують проблеми зі здоров'ям. Крім цього, опитування Світового банку

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

свідчить, що такі співробітники в три рази частіше користуються корпоративними програмами страхування здоров'я для лікування психічних розладів. Якщо глянути на проблему із цього погляду, то можна затверджувати, що відеоконференції допомагають поліпшити якість життя співробітників.

#### **Фактор 4. Час простою співробітників**

Мова йде про період, коли співробітник не може працювати над досягненням корпоративних цілей, тобто змушений відволікатися від виконання своїх безпосередніх обов'язків. Очевидно, що відомість цього показника до мінімуму може підвищити продуктивність персоналу й ефективність організації в цілому. Якщо мова йде про ділові поїздки, то час простою містить у собі поїздки від офісу до аеропорту, очікування посадки в літак і трансфер до точки призначення.

Крім того, результати опитування, проведеного для компанії Delta Airlines, свідчать про те, що навіть час, проведений на борті літака, можна віднести до простою, оскільки лише 5% співробітників, що роблять бізнес-поїздки, у салоні лайнера «сфальцьовані на роботі». За даними дослідників, 81% з них читають неділову літературу, 64% просто розслаблюються або сплять, 55% дивляться кіно або грають у відеоігри. Відповідно, зводячи до мінімуму обсяг поїздок, відеоконференції допомагають організаціям максимально скоротити час простою своїх співробітників.

Крім того, шляхом нехитрих математичних обчислень можна виявити безліч цікавих статей економії.

**Оцінка очікуваних щорічних заощаджень.** Якщо прийняти, що  $X$  – це середня економія розраховуючи на співробітника при проведенні зустрічі по ВКЗ (руб/\$),  $Y$  – кількість поїздок у рік, що співробітник буде переводити у ВКЗ (шт.),  $Z$  – число тих, хто буде використовувати відео-конференц-зв'язок, то щорічна економія дорівнює  $X \times Y = Z$ .

Якщо взяти цифри з наведеного аналітичного звіту ( $X = \$2100$ ,  $Y = 22$ ), то середня економія на одного співробітника складе 46 200 доларів. Очевидно, що,

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

підставивши експертні дані по своїй компанії, будь-який користувач системи ВКЗ може одержати результат потенційної економії.

**Оцінка ROI.** На підставі отриманих даних розраховується ROI.

$$\text{ROI (\%)} = 100 \times \text{Щорічна економія} / \text{Інвестиції}$$

**Оцінка точки безбитковості.** Крім того, можна розрахувати рентабельність проекту.

$$\text{Точка безбитковості (мес)} = 12 \times \text{Інвестиції} / \text{Щорічна економія}$$

У кожній компанії значення запропонованих змінних будуть свідомо відрізнятися, тому варто використовувати індивідуальні значення для цільового розрахунку. Звичайно, повна рентабельність системи відео-конференц-зв'язку залежить від конкретних умов її застосування, та й не всі фактори піддаються кількісній оцінці. Наприклад, вигоди від підвищення ефективності керування за рахунок впровадження відеоконференцій підрахувати неможливо. Для цього прийде співвіднести між собою не тільки витрати на проведення наради з безпосереднім збором учасників, але й потенційні вигоди від своєчасного прийняття рішень і можливі втрати внаслідок запізнення із прийняттям цих рішень. Скажемо, якщо мова йде про банківський сектор, те фінансові втрати можуть бути просто величезними, а в медицині ціна зволікання – людське життя.

### 3.2 Розробка структурної схеми

Групові системи ВКЗ, або групові термінали й приставки ВКЗ, – це абонентські комплекти, призначені для забезпечення відеозв'язку. До складу системи входить кодек, відповідальний за кодування/декодування відео й звуку, відеокамера й мікрофон.

Такі системи звичайно встановлюють в окремих приміщеннях, наприклад, у переговорних кімнатах.

Їхній неодмінний атрибут – великий плазмений/РК телевізор або проектор для відображення учасників відеоконференції.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Основне призначення групових систем ВКЗ – забезпечити комфортне візуальне спілкування групи людей з вилученими співрозмовниками.

У загальному випадку такої системі доводиться вирішувати дві дуже прості, на перший погляд, завдання.

Перша полягає у відеозйомці одних учасників відеоконференції й передачі їхнього зображення (разом зі звуком від мікрофона) іншим.

Друга складається у відображенні картинки на одному або декількох ТБ (моніторах) і виводі звуку на динаміки телевізора або акустичні колонки (через звукопосилюючу апаратуру).

Однак у дійсності ці завдання не настільки прості. Наприклад, зображення й звук необхідно попередньо стиснути спеціальними кодами й тільки після цього їх можна відправити по мережі IP (рідше ISDN) відповідно до передбачених протоколів. На протилежній стороні інформацію приймає інша групова система ВКЗ, який доводиться виконувати ще більше операцій: крім декодування відеозображення й звуку потрібно відновити проходження, що втрапилися по шляху, або пакети, що прийшли із затримкою, IP і синхронізувати відео зі звуком. Крім того, у кімнаті, де проводяться сеанси відеозв'язку, може бути встановлена ще одна відеокамера, наприклад, для заднього плану, документ-камера або комп'ютер для демонстрації презентацій. Всі ці дані теж потрібно стискати, передавати, розпаковувати, відновлювати й синхронізувати.

Зі сказаного очевидно, що групові системи ВКЗ повинні мати велику обчислювальну потужність для обробки голосовий і відеоінформації в режимі реального часу, і тому вони являють собою складні, з інженерної точки зору, продукти.

### **Види групових систем ВКЗ**

Групові системи ВКЗ можна розділити на системи початкового рівня й системи бізнесу-класу.

Перші призначені для підключення до мінімально необхідного для роботи аудіо/відео устаткуванню й підтримують в основному з'єднання «точка-точка».

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Другі, як правило, більше продуктивні, підтримують високошвидкісні з'єднання й здатні автоматично наводити камеру на мовець. Крім того, за допомогою убудованого сервера багатоточкового конференц-зв'язку (Multipoint Control Unit, MCU) вони дозволяють організувати багатокористувальницькі відеоконференції й мають широкий набір мережних інтерфейсів і аудіо/відеопортів.

Приналежність устаткування до того або іншого класу зовсім не означає, що одне забезпечує спілкування більше високої якості, ніж інше.

Вибір конкретного класу визначається тільки завданнями, які покладають на систему ВКЗ.

Наприклад, якщо в компанії потрібно налагодити відеозв'язок між декількома відділеннями (офісами), але в багатобічних конференціях немає потреби, то використовувати системи бізнес-рівня не має змісту.

Можлива структура корпоративної мережі відео-конференц-зв'язку для малого й середнього бізнесу наведена на рисунку 3.1. Групова система ВКЗ із убудованим сервером для проведення багатокористувальницьких відеоконференцій звичайно розташовується в центральному офісі компанії. Вона з'єднана з іншими великими офісами, де встановлені групові системи початкового рівня, і/або з персональними терміналами ВКЗ, якими користуються деякі співробітники. Така структура корпоративної мережі відеозв'язку дозволяє, з одного боку, організувати всі види відеоконференцій, а з іншого боку – заощаджувати на групових терміналах.

У технічному описі будь-якої групової системи ВКЗ можна зустріти довгий перелік підтримуваних стандартів: відеокодеки, аудіокодеки, стандарти спільної роботи з даними, зв'язки й керування. Тим часом, більша частина характеристик, що приводяться в описах, не має практичного значення.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23



значно більше високу якість зображення, чим H.263, при тій же пропускній здатності.

– Інші відеокодеки, підтримка яких заявлена виробниками групових систем ВКЗ, потрібні, по-перше, для сумісності з більше раннім устаткуванням (кодек H.261), а по-друге, для забезпечення максимально ефективної роботи устаткування однієї марки. Скажемо, у ситуації, коли групові системи ВКЗ різних вендорів з якихось причин (наприклад, через велику кількість учасників) були б змушені перейти з відеокодеку H.264 на кодек H.263, системи одного виробника, можливо, зможуть працювати на іншій, більше зробленого різновиду цього стандарту: H.263+ або H.263++.

– Аудіокодеки. Для забезпечення аудіосумісності систем ВКЗ різних виробників у всіх групових терміналах повинна бути реалізована підтримка єдиного стандарту кодування звуку -G.711. Цей алгоритм кодування вузькополосного звуку (3,1 кГц) у каналі 48, 56 або 64 Кбит/із забезпечує краща якість звуку в порівнянні з іншими аудіокодеками – не гірше, ніж при звичайному телефонному зв'язку. Він обходиться мінімумом апаратних ресурсів, але має потребу в значній пропускній здатності.

– Всі інші аудіокодеки використовують алгоритми стиску інформації із втратами, що практично не впливає на якість звуку, але при цьому необхідна пропускна здатність каналу в 5-10 разів нижче, ніж в G.711. Так, стандарт кодування звуку G.729 цілком підходить для використання в Internet – йому потрібно всього 8 Кбіт/с. Однак кодеки із сильним стиском звуку нестійкі до втрат пакетів, а крім того, для них характерна висока складність обробки звуку й, як наслідок, підвищений навантаження на апаратні ресурси устаткування.

– Інші стандарти. Ще один важливий стандарт – H.239, також відомий як функція DualVideo, – дозволяє паралельно із зображенням учасників передавати статичне, рідко обновлюване відео (двопотокowa передача): знімки з документ-камери, слайди презентації, електронні таблиці або текстові файли з комп'ютера. Для виводу статичної «картинки» звичайно використовується окремий монітор з

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

високим дозволом (1024x768 точок і вище). Без DualVideo відеоконференції не мають змісту, тому з 2002 р. ця функція підтримується майже всіма виробниками устаткування для ВКЗ.

### **Основні характеристики групових систем ВКЗ**

Кожна групова система ВКЗ має широкий набір технічних характеристик. Якісь із них по-справжньому важливі, хоча необізнаному користувачеві можуть представлятися несуттєвими. Інші, навпаки, на перший погляд, вражають, але реального виграшу не дають.

#### **Дизайн**

Як не дивно, але зовнішній вигляд групової системи був і залишається одним із ключових критеріїв вибору: це устаткування завжди встановлюється на очах, підкреслює статус організації й повинне виглядати відповідним чином. Статистика світових продажів свідчить, що «непрезентабельні» групові системи ВКЗ не користуються популярністю, навіть незважаючи на найширшого функціонала або низьку ціну. Саме тому всі великі гравці ринку приділяють зовнішньому вигляду особливу увагу й навіть прибігають до послуг сторонніх дизайнерів.

#### **Швидкість з'єднання по IP**

Значення цього параметра в групових систем ВКЗ варіюється від 768 Кбіт/с до 4 Мбіт/с. Але чи не так уже потрібна підтримка високих швидкостей? Практика показує, що незалежно від використовуваного кодеку швидкість передачі відео ніколи не перевищує 768 Кбіт/с. Більше того, навіть якщо відеоконференція проводиться між двома учасниками, термінали яких підтримують швидкості передачі до 4 Мбіт/с, реальна швидкість передачі навряд чи перевищить 1 Мбіт/с, тому якість зображення залишиться таким же, як і при 768 Кбіт/с.

Зовсім інша ситуація, якщо групова система має убудований MCU. У такому випадку якість відео буде залежати від максимальної швидкості з'єднання по IP. Очевидно, що при загальній швидкості з'єднання, наприклад, 2 Мбіт/с,

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

груповий термінал ВКЗ фізично не зможе забезпечити проведення чотирибічної конференції зі швидкостями 768 Кбіт/с ( $3 \times 768 \text{ Кбіт/с} = 2,3 \text{ Мбіт/с}$ ). Швидше за все, швидкість передачі даних буде знижена до 384 кбіт/с, а виходить, постраждає якість.

Таким чином, для групових систем початкового рівня без MCU підтримка швидкостей з'єднання по IP більше 768 Кбіт/с не дає реальних переваг. З іншого боку, більша швидкість – це певний запас на майбутнє, наявність якого дозволить реалізувати нові, більше вимогливі до пропускної здатності стандарти шляхом простий «перепрошивання» устаткування.

Для групових систем бізнес-класу підтримка високої пропускної здатності по IP – необхідність і застава того, що багатобічна відео-конференц-зв'язок буде мати гарну якість.

### **Швидкість з'єднання по ISDN**

У групових систем ВКЗ швидкість з'єднання по ISDN становить від 384 Кбіт/с до 2 Мбіт/с. І це, мабуть, самий малозначимий параметр для сучасних систем ВКЗ. Сеанси відеозв'язку по телефонних каналах проводяться вкрай рідко, тому багато виробників взагалі відмовилися від підтримки ISDN у групових системах.

### **Дозвіл відео**

У відео-конференц-зв'язку є кілька стандартних дозволів відеосигналу: SQCIF (128x96), QCIF (176x144), CIF (352x288), 4CIF (704x576). Перші два (SQCIF і QCIF) забезпечують занадто низька якість зображення, щоб їх мало сенс використовувати, а от підтримка дозволів CIF і 4CIF обов'язкова.

### **Убудований сервер MCU**

Убудовані MCU не є прерогативою систем бізнес рівня й звичайно забезпечують проведення відеоконференцій від 4 до 9 учасників. Як визначити, MCU наскільки буде оптимальний? У першу чергу необхідно оцінити реальні потреби. Нерозумно вибирати систему, MCU якої розраховано на чотирьох користувачів, якщо в компанії є вісім представництв по всій країні. При рівному

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

числі підтримуваних учасників кращим вибором будуть моделі з найбільшою швидкістю з'єднання по IP, оскільки вони забезпечують більше високу якість відеозв'язку.

### **Відеовходи й відеовиходи**

У будь-якої групової системи ВКЗ повинні бути як відеовходи для підключення джерел відеосигналів, так і відеовиходи для виводу відеозображення.

Однак, коли число одних досягає п'яти, а інших дев'яти, виникає резонне питання: навіщо так багато?

Велика кількість відеовходів/виходів обумовлено наявністю різних типів переданих відеосигналів. Найбільше поширення одержали композитний (рознімання RCA) і S-Video. Незважаючи на сумісність практично з усіма видами апаратури (ТБ, відеокамери й т.п.), вони забезпечують найнижчий дозвіл і якість. Тому в багатьох групових системах вивід відеозображення можливий також у вигляді більше якісного компонентного сигналу (3xRCA рознімання на канал). Крім того, як уже говорилося, практично у всіх системах ВКЗ підтримується функція DualVideo, і в цьому випадку відеосигнал з комп'ютера передається у високому дозволі через рознімання DVI/XGA.

Для організації відеоконференції буде потрібно не менш двох відеовходів і двох відеовиходів:

- один вхід для підключення додаткової відеокамери;
- один вхід DVI/XGA для підключення ноутбука (проведення презентацій) або документа-камери. Як додаткові джерела відеосигналу до групових систем ВКЗ часто підключають DVD- або відеопрогравачі;
- один вихід для виводу зображення з основної відеокамери;
- один вихід DVI/XGA для виводу презентації й зображення з документ-камери або комп'ютера.

При проведенні великих відеоконференцій до додаткових відеовходів часто підключають монітори провідного інженера, що обслуговує сеанс ВКЗ, а

також відеомагнітофони або DVD-рекодері, призначені для запису (протоколювання) конференцій.

### **Висока чіткість (High Definition, HD)**

Останнім часом особливий інтерес викликають відеоконференції, що забезпечують високу якість зображення – з дозволом 1280x720 (стандарт 720p) або 1920x1080 (стандарт 1080i/p) точок.

Практично всі провідні виробники систем ВКЗ уже випустили устаткування з підтримкою HD. Однак попит на подібні системи поки невеликий через їхню дорожнечу й високу вимогливість до пропускну здатності, а реальний вигреш від застосування HD майже непомітний.

### **Додаткові функції**

Одна з найбільш корисних – автоматичне наведення камери на мовець у цей момент учасника відеоконференції, зображення якого виводиться на екран замість загального плану приміщення з усіма присутніми в ньому людьми. На жаль, ця функція часто надається за додаткову плату.

Дуже цікавої, але практично не використовуваної є підтримка каскадування. Вона дозволяє збільшити число учасників відеоконференції за рахунок застосування декількох групових систем ВКЗ із убудованим MCU. Таке рішення має цілий ряд обмежень, наприклад, на екрані відображаються не всі учасники, а тільки виступаючі. Тому при необхідності проведення більше багатолюдних відеоконференцій найчастіше встановлюють додаткові MCU.

Нерідко групові системи ВКЗ мають оригінальні убудовані функції.

Наприклад, конференц-телефони Polycom можна використовувати як зовнішні мікрофони.

У групових системах Tandberg є рознімання USB для підключення бездротового адаптера WiFi.

А системи Aethra сумісні з Microsoft NetMeeting, їх можна з'єднувати по локальній мережі з комп'ютерами й проводити конференції даних (Data Conference).

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Sony реалізувала можливість запису конференції на карти пам'яті MemoryStick.

А рознімання USB у групових систем ВКЗ виробництва AddPac забезпечує підключення практично будь-яких периферійних пристроїв.

### 3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.3.

Зі схеми видно, що розроблена система складається з наступних частин:

#### 1. Припустимі дозволи захвату відео.

– для локальних відеокамер 384x288, 480x360, 560x420, 640x480, 720x540, 768x576 (напівкадр і повний кадр).

– для мережних камер 176x144, 240x180, 320x240, 352x240, 352x288, 384x288, 480x360, 560x420, 640x480, 704x420, 704x576, 720x540, 720x576, 768x576, 1280x720, 1280x960, 1280x1024, 1600x1200.

#### 2. Програмний детектор руху.

Програмний детектор руху працює на основі порівняння кількості змінених пікселей у сусідніх відеокадрах із заданим у налаштуваннях граничним значенням для кожної відеокамери.

Додатково включає наступні технології:

- налаштування рівня шуму;
- виділення областей аналізу накладенням маски;
- групування в тимчасові сеанси руху.

#### 3. Режими запису відеокадрів.

- без запису на жорсткий диск,
- запис відеокадрів по спрацьовуванню детектора;
- запис по детектору + запис певної кількості відеокадрів до/після спрацьовування детектора (режими передзапису й післязапису).

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Відеокадри можуть зберігатися у файлах наступного формату:

– файли у форматі покадрового кодування та передачі відеоданих на основі ієрархічного кодування;

– окремі файли JPEG;

– фільми AVI, стиск кодеком MJPEG.

Відеоархів – циклічний (кільцевий), з видаленням частини старих даних при досягненні вільного місця жорсткого диска певної величини.

Забезпечено можливість накладення (малювання) тексту й часу на збережений відеокадр.

#### 4. Режими спостереження.

Перегляд у режимі реального часу можливий:

– за допомогою локальної програми, що виводить зображення форматах квадратор, мультіекран, поліекран і ін.;

– за допомогою веб-браузера з дозволених віддалених комп'ютерів.

#### 5. Керування й налаштування.

Реалізована для зареєстрованих користувачів по пароллю.

#### 6. Перегляд відеоархіву.

Реалізований для зареєстрованих користувачів по пароллю.

#### 7. Реакції на події.

– Програвання звукового файлу на початок і закінчення сеансу руху.

– Повідомлення через e-mail по початку сеансу руху із вкладеним JPEG файлом першого кадру руху (прим: по мережі на SMTP сервер).

– Виконання користувальницьких скриптів і завдань.

#### 8. Багатокористувальницький доступ.

У системі передбачені 4 групи користувачів:

– інсталятори (установники),

– адміністратори,

– оператори архіву,

– оператори спостереження.

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Кількість користувачів і віддалених робочих місць програмно не обмежено.

9. Допомога.

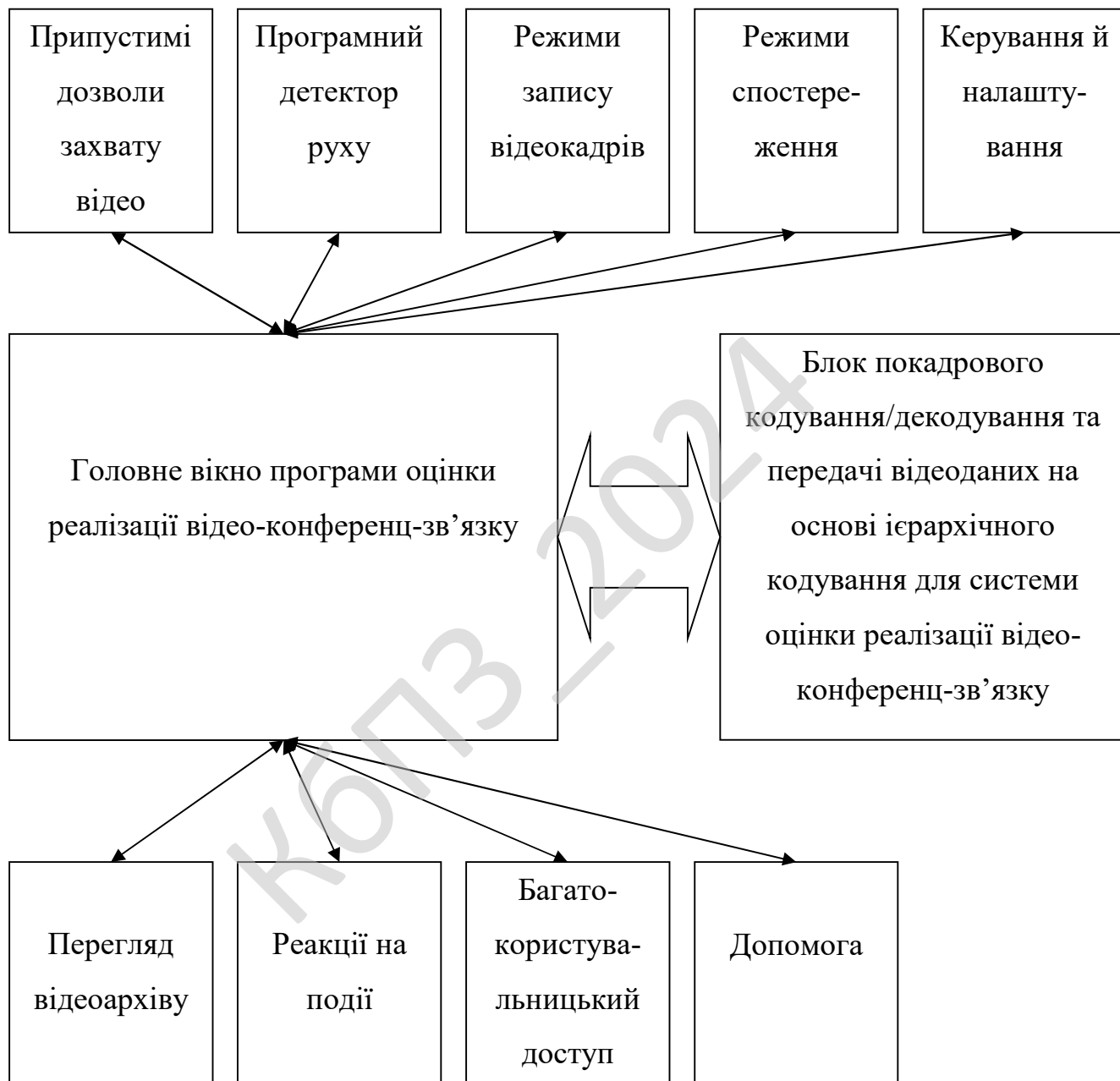


Рисунок 3.2 – Функціональна схема системи

На приймальній стороні інформація, яка поступає через Bluetooth-коннектор, або декодується й переглядається, або записується на носій інформації.

Стандарт 802.15.1 (Bluetooth) вибрано виходячи з наступних міркувань:

– Гранична швидкість, обумовлена стандартом ZigBee (250 Кбіт/с), істотно обмежує пропускну здатність і недостатня для передачі чіткого зображення.

– Швидкість Bluetooth (1 Мбіт/с) є достатньою й залишає можливість нарощування функціональності.

– У частині витрат на реалізацію стандарту Wi-Fi досить надлишковий, оскільки передбачає реалізацію мережної взаємодії по моделі ISO/OSI. Це надає зручність у роботі із пристроєм на основі стандарту Wi-Fi, але істотно збільшує його вартість.

– Стандарт Bluetooth використовує принцип стрибкоподібної перебудови частоти (FHSS), що підвищує рівень захисту від несанкціонованого доступу й забезпечує більшу завадостійкість у порівнянні з Wi-Fi.

Надамо опис стандарту 802.15.1 (Bluetooth).

**Bluetooth** або **блютус** – виробнича специфікація бездротових персональних мереж (Wireless personal area network, WPAN).

Bluetooth забезпечує обмін інформацією між такими пристроями як кишенькові й звичайні персональні комп'ютери, мобільні телефони, ноутбуки, принтери, цифрові фотоапарати, мишки, клавіатури, джойстики, навушники, гарнітури на надійній, недорогій, повсюдно доступній радіочастоті для ближнього зв'язку.

Bluetooth дозволяє цим пристроям повідомлятися, коли вони перебувають у радіусі до 10-100 метрів друг від друга (дальність дуже сильно залежить від перешкод), навіть у різних приміщеннях.

### **Принцип дії Bluetooth**

Радіозв'язок Bluetooth здійснюється в ISM-Діапазоні (Industry, Science and Medicine), що використовується в різних побутових приладах і бездротових

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

мережах (вільний від ліцензування діапазон 2,4-2,4835 ГГц). В Bluetooth застосовується метод розширення спектра зі стрибкоподібною перебудовою частоти (Frequency Hopping Spread Spectrum, FHSS). Метод FHSS простий у реалізації, забезпечує стійкість до широкополосних перешкод, а встаткування коштує недорого.

Відповідно до алгоритму FHSS, в Bluetooth несуча частота сигналу стрибкоподібно міняється 1600 разів у секунду (усього виділяється 79 робочих частот шириною в 1 МГц, а в Японії, Франції й Іспанії смуга більш вузька – 23 частотних канали). Послідовність перемикавання між частотами для кожного з'єднання є псевдовипадковою й відома тільки передавачу й приймачу, які кожні 625 мкс (один часовий слот) синхронно перебудовуються з однієї несучої частоти на іншу. Таким чином, якщо поруч працюють декілька пар приймач-передавач, то вони не заважають один одному. Цей алгоритм є також складовою частиною системи захисту конфіденційності переданої інформації: перехід відбувається по псевдовипадковому алгоритму й визначається окремо для кожного з'єднання. При передачі цифрових даних і аудіосигнала (64 кбіт/с в обох напрямках) використовуються різні схеми кодування: аудіосигнал не повторюється (як правило), а цифрові дані у випадку втрати пакета інформації будуть передані повторно. Без завадостійкого кодування це забезпечує передачу даних зі швидкостями 723,2 кбіт/с зі зворотним каналом 57,6 кбіт/с, або 433,9 кбіт/с в обох напрямках.

Протокол Bluetooth підтримує не тільки з'єднання «точка-точка», але й з'єднання «точка-багатоточка».

### **Специфікації**

Пристрої версій 1.0 (1998) і 1.0B мали погану сумісність між продуктами різних виробників. В 1.0 і 1.0B була обов'язковою передача адреси пристрою (BD\_ADDR) на етапі встановлення зв'язку, що унеможливило реалізацію анонімності з'єднання на протокольному рівні й було основним недоліком даної специфікації.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

В Bluetooth 1.1 була виправлена множина помилок, знайдених в 1.0B, додана підтримка для нешифрованих каналів, індикація рівня потужності прийнятого сигналу (RSSI).

У версії 1.2 була додана технологія адаптивної перебудови робочої частоти (AFH), що поліпшило опірність до електромагнітної інтерференції (перешкодам) шляхом використання рознесених частот у послідовності перебудови. Також збільшилася швидкість передачі й додалася технологія eSCO, що поліпшувала якість передачі голосу шляхом повторення ушкоджених пакетів. В HCI додалася підтримка трьох-провідного інтерфейсу UART.

Bluetooth версії 2.0 (2004) повністю сполучимий з версіями 1.x. Основним нововведенням стала підтримка EDR (Enhanced Data Rate), що дозволило підвищити швидкість передачі до 2,1 Мбіт/с.

Bluetooth 2.1. 2007 рік. Додано технологію розширеного запиту характеристик пристрою (для додаткової фільтрації списку при спарюванні), енергозберігаюча технологія Sniff Subrating, що дозволяє збільшити тривалість роботи пристрою від одного заряду акумулятора в 3-10 разів. Крім того оновлена специфікація істотно спрощує й прискорює встановлення зв'язку між двома пристроями, дозволяє робити відновлення ключа шифрування без розриву з'єднання, а також робить зазначені з'єднання більше захищеними, завдяки використанню технології Near Field Communication.

Bluetooth 2.1+EDR. Останнє відновлення Bluetooth було в серпні 2008 року, коли Bluetooth SIG представив версію 2.1+EDR. Того разу було знижене енергоспоживання й підвищене рівень захисту даних. Консорціум пообіцяв представити нову специфікацію в середині 2009 року, повідомивши, що максимальна швидкість передачі даних по Bluetooth 3.0 (тоді – Bluetooth 2.2) складе 300 Мбіт/с, стільки ж, скільки в Draft-N.

Bluetooth 4.0. У грудні 2009 року Bluetooth SIG анонсувала версію Bluetooth 4.0. Технологія, призначена для електронних датчиків (у спортивному взутті, тренажерах, мініатюрних сенсорах, розташовуваних на тілі пацієнтів і

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

т.д.), підтримує низьке енергоспоживання. Швидкість передачі даних – 1 Мбіт/с. До кінця 2009 року компанією ST-Ericsson був випущений чип з підтримкою Bluetooth 3.0 і Bluetooth 4.0.

### **Профілі Bluetooth**

Нижчевказані профілі визначені й схвалені групою розробки Bluetooth SIG:

– Advanced Audio Distribution Profile (A2DP). A2DP розроблений для передачі двоканального стерео аудіопотока, наприклад музики, до бездротової гарнітури або будь-якому іншому пристрою. Профіль повністю підтримує низькокомпресований кодек Sub\_Band\_Codec (SBC) і опціонально підтримує MPEG-1,2 аудіо, MPEG-2,4 AAC і ATRAC, здатний підтримувати кодеки визначені виробником.

– Audio/Video Remote Control Profile (AVRCP). Цей профіль розроблений для керування стандартними функціями телевізорів, Hi-Fi устаткування та іншм. Тобто дозволяє створювати пристрої з функціями дистанційного керування. Може використовуватися у зв'язуванні із профілями A2DP або VDPT.

– Basic Imaging Profile (BIP). Профіль розроблений для пересилання зображень між пристроями й включає можливість зміни розміру зображення й конвертування в підтримуваний формат приймаючого пристрою.

– Basic Printing Profile (BPP). Профіль дозволяє пересилати текст, e-mails, vCard і інші елементи на принтер. Профіль не жадає від принтера специфічних драйверів, що вигідно відрізняє його від HCRP.

– Common ISDN Access Profile (CIP). Профіль для доступу пристроїв до ISDN.

– Cordless Telephony Profile (CTP). Профіль бездротової телефонії.

– Device ID Profile (DID). Профіль дозволяє ідентифікувати клас пристрою, виробника, версію продукту.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– Dial-up Networking Profile (DUN). Протокол надає стандартний доступ до інтернету або іншому телефонному сервісу через Bluetooth. Базується на SPP, містить у собі команди PPP і AT, певні в специфікації ETSI 07.07.

– Fax Profile (FAX). Профіль надає інтерфейс між мобільним або стаціонарним телефоном і ПК на якому встановлене програмне забезпечення для факсів. Підтримує набір AT-команд у стилі ITU T.31 і/або ITU T.32. Голосовий дзвінок або передача даних профілем не підтримується.

– File Transfer Profile (FTP\_profile). Профіль забезпечує доступ до файлової системи пристрою. Включає стандартний набір команд FTP, що дозволяє одержувати список директорій, зміни директорій, одержувати, передавати й видаляти файли. Як транспорт використовується OBEX, базується на GOEP.

– General Audio / Video Distribution Profile (GAVDP). Профіль є базою для A2DP і VDP.

– Generic Access Profile (GAP). Профіль є базою для всіх інших профілів.

– Generic Object Exchange Profile (GOEP). Профіль є базою для інших профілів передачі даних, базується на OBEX.

– Hard Copy Cable Replacement Profile (HCRP). Профіль надає просту альтернативу кабельного з'єднання між пристроєм і принтером. Мінус профілю в тому, що для принтера необхідні специфічні драйвера, що робить профіль неуніверсальним.

– Hands-Free Profile (HFP). Профіль використовується для з'єднання бездротової гарнітури й телефону, передає монозвук в одному каналі.

– Human Interface Device Profile (HID). Забезпечує підтримку пристроїв з HID (Human Interface Device), таких як мишки, джойстики, клавіатури й інші. Використовує повільний канал, працює на зниженій потужності.

– Headset Profile (HSP). Профіль використовується для з'єднання бездротової гарнітури й телефону. Підтримує мінімальний набір AT-команд специфікації GSM 07.07 для забезпечення можливості робити дзвінки, відповідати на дзвінки, завершувати дзвінок, набудувати гучність. Підтримка

гарнітурою профілю Headset («навушники»). Профіль – набір функцій або можливостей, доступних для певного пристрою Bluetooth. Для спільної роботи Bluetooth-пристроїв необхідно, щоб всі вони підтримували загальний профіль. Через профіль Headset, при наявності Bluetooth 1.2 і вище, а також за його підтримкою головним пристроєм, можна виводити на гарнітуру весь звуковий супровід роботи телефону. Наприклад, прослуховувати на гарнітурі всі сигнали підтвердження операцій, mp3-музику із плеера, мелодії дзвінка, звуковий ряд відеороликів. Також з підтримкою даного профілю з'являється можливість міняти гучність, робити дзвінки, а також відповідати або відхиляти їх безпосередньо з гарнітури. Гарнітури, що підтримують такий профіль мають можливість передачі стереозвуку, на відміну від моделей, які підтримують тільки профіль Hands-Free.

– Intercom Profile (ICP). Забезпечує голосові дзвінки між Bluetooth-сумісними пристроями.

– LAN Access Profile (LAP). LAN Access profile забезпечує можливість доступу Bluetooth-пристроєм до обчислювальних мереж LAN, WAN або Internet за допомогою іншого Bluetooth-пристрою, що має фізичне підключення до цих мереж. Bluetooth-пристрій використовує PPP поверх RFCOMM для установки з'єднання. LAP також допускає створення ad-hoc Bluetooth-мереж.

– Object Push Profile (OPP). Базовий профіль для пересилання «об'єктів», таких як зображення, віртуальні візитні картки й ін. Передачу даних ініціює пристрій, що відправляє (клієнт), а не приймає (сервер).

– Personal Area Networking Profile (PAN). Профіль дозволяє використовувати протокол Bluetooth Network Encapsulation як транспорт через Bluetooth-з'єднання.

– Phone Book Access Profile (PBAF). Профіль дозволяє обмінюватися записами телефонних книг між пристроями.

– Serial Port Profile (SPP). Профіль базується на специфікації ETSI TS07.10 і використовує протокол RFCOMM. Профіль емулює послідовний порт, надаючи

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

можливість заміни стандартного RS-232 бездротовим з'єднанням. Є базовим для профілів DUN, FAX, HSP і AVRCP.

– Service Discovery Application Profile (SDAP). Профіль використовується для надання інформації про профілі, які використовує пристрій-сервер.

– SIM Access Profile (SAP, SIM). Профіль дозволяє одержати доступ до SIM-карти телефону, що дозволяє використовувати одну SIM-карту для декількох пристроїв.

– Synchronisation Profile (SYNCH). Профіль дозволяє синхронізувати особисті дані (PIM). Профіль запозичений зі специфікації інфрачервоного зв'язку й адаптований групою Bluetooth SIG.

– Video Distribution Profile (VDP). Профіль дозволяє передавати потокове відео. Підтримує H.263, стандарти MPEG-4 Visual Simple Profile, H.263 profiles 3, profile 8 підтримуються опціонально й не втримуються в специфікації.

– Wireless Application Protocol Bearer (WAPB). Протокол для організації P-to-P (Point-to-Point) з'єднання через Bluetooth.

### **Безпека**

У червні 2006 року Авиша Вул (Avishai Wool) і Янив Шакед (Yaniv Shaked) опублікували статтю, що містить докладний опис атаки на bluetooth-пристрої. Матеріал містив опис як активної, так і пасивної атаки, що дозволяє роздобути PIN коду пристрою й надалі здійснити з'єднання з даним пристроєм. Пасивна атака дозволяє відповідно екіпірованому зловмисникові «підслухати» (sniffing) процес ініціалізації з'єднання й надалі використовувати отримані в результаті прослуховування й аналізу дані для встановлення з'єднання (spoofing). Природно, для проведення даної атаки зловмисникові потрібно перебувати в безпосередній близькості й безпосередньо в момент встановлення зв'язку. Це не завжди можливо. Тому народилася ідея активної атаки. Була виявлена можливість відправлення особливого повідомлення в певний момент, що дозволяє почати процес ініціалізації із пристроєм зловмисника. Обидві процедури взлому досить складні й включають кілька етапів, основний з яких – збір пакетів

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>39</b>

даних і їхній аналіз. Самі атаки засновані на уразливостях у механізмі автентифікації й створення ключа-шифру між двома пристроями. І тому перед викладом механізму атак розглянемо механізм ініціалізації bluetooth-з'єднання.

### **Ініціалізація bluetooth-з'єднання**

Ініціалізацією, відносно bluetooth, прийнято називати процес установки зв'язку. Її можна розділити на три етапи:

- Генерація ключа Kinit.
- Генерація ключа зв'язку (він зветься link key і позначається, як Kab).
- Автентифікація.

Перші два пункти входять у так звану процедуру паринга.

Паринг (PAIRING) – або сполучення – процес зв'язку двох (або більше) пристроїв з метою створення єдиної секретної величини Kinit, що вони будуть надалі використовувати при спілкуванні. У деяких перекладах офіційних документів по bluetooth можна також зустріти термін «припасування пари».

Перед початком процедури сполучення на обох сторонах необхідно ввести PIN-код. Звичайна ситуація: два чоловіки хочуть зв'язати свої телефони й заздалегідь домовляються про PIN-код.

Для простоти будемо розглядати ситуацію із двома пристроями. Принципово це не вплине на механізми встановлення зв'язку й наступних атак. Далі пристрої, що з'єднуються, будуть позначатися А і В, більше того, один із пристроїв при сполученні стає головним (Master), а друге – веденим (Slave). Будемо вважати пристрій А головним, а В – веденим. Створення ключа Kinit починається відразу після того, як були уведені PIN-коди.

Kinit формується по алгоритму E22, що оперує наступними величинами:

- BD\_ADDR – унікальна адреса BT-пристрою. Довжина 48 біт (аналог MAC-адреси мережної карти PC).
- PIN-код і його довжина.
- IN\_RAND. Випадкова 128-бітна величина.

На виході E22 алгоритму одержуємо 128-бітне слово, іменоване Kinit. Число IN\_RAND відсилається пристроєм А у чистому виді. У випадку, якщо PIN незмінний для цього пристрою, то при формуванні Kinit використовується BD\_ADDR, отримане від іншого пристрою. У випадку якщо в обох пристроїв змінювані PIN-коди, буде використана BD\_ADDR(B) – адреса slave-пристрою. Перший крок сполучення пройдений. За ним треба створення Kab. Після його формування Kinit виключається з використання.

Для створення ключа зв'язку Kab пристрої обмінюються 128-бітними словами LK\_RAND(A) і LK\_RAND(B), генеруємими випадковим образом. Далі треба побітовий XOR із ключем ініціалізації Kinit. І знову обмін отриманим значенням. Потім треба обчислення ключа по алгоритму E21.

Для цього необхідні величини:

- BD\_ADDR.
- 128-бітний LK\_RAND (кожний пристрій зберігає своє й отримане від іншого пристрою значення).

На даному етапі pairing закінчується й починається останній етап ініціалізації bluetooth – Mutual authentication або взаємна автентифікація. Засновано вона на схемі «запит-відповідь». Один із пристроїв стає верифікатором, генерує випадкову величину AU\_RAND(A) і засилає його сусідньому пристрою (в plain text), названому пред'явником (claimant – в оригінальній документації). Як тільки пред'явник одержує це «слово», починається обчислення величини SRES по алгоритму E1, і вона відправляється верифікатору. Сусідній пристрій робить аналогічні обчислення й перевіряє відповідь пред'явника. Якщо SRES збіглися, то, виходить, все добре, і тепер пристрої міняються ролями, у такий спосіб процес повторюється заново.

E1-алгоритм оперує такими величинами:

- Випадково створене AU\_RAND.
- link key Kab.
- Свій власний BD\_ADDR.

### 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.

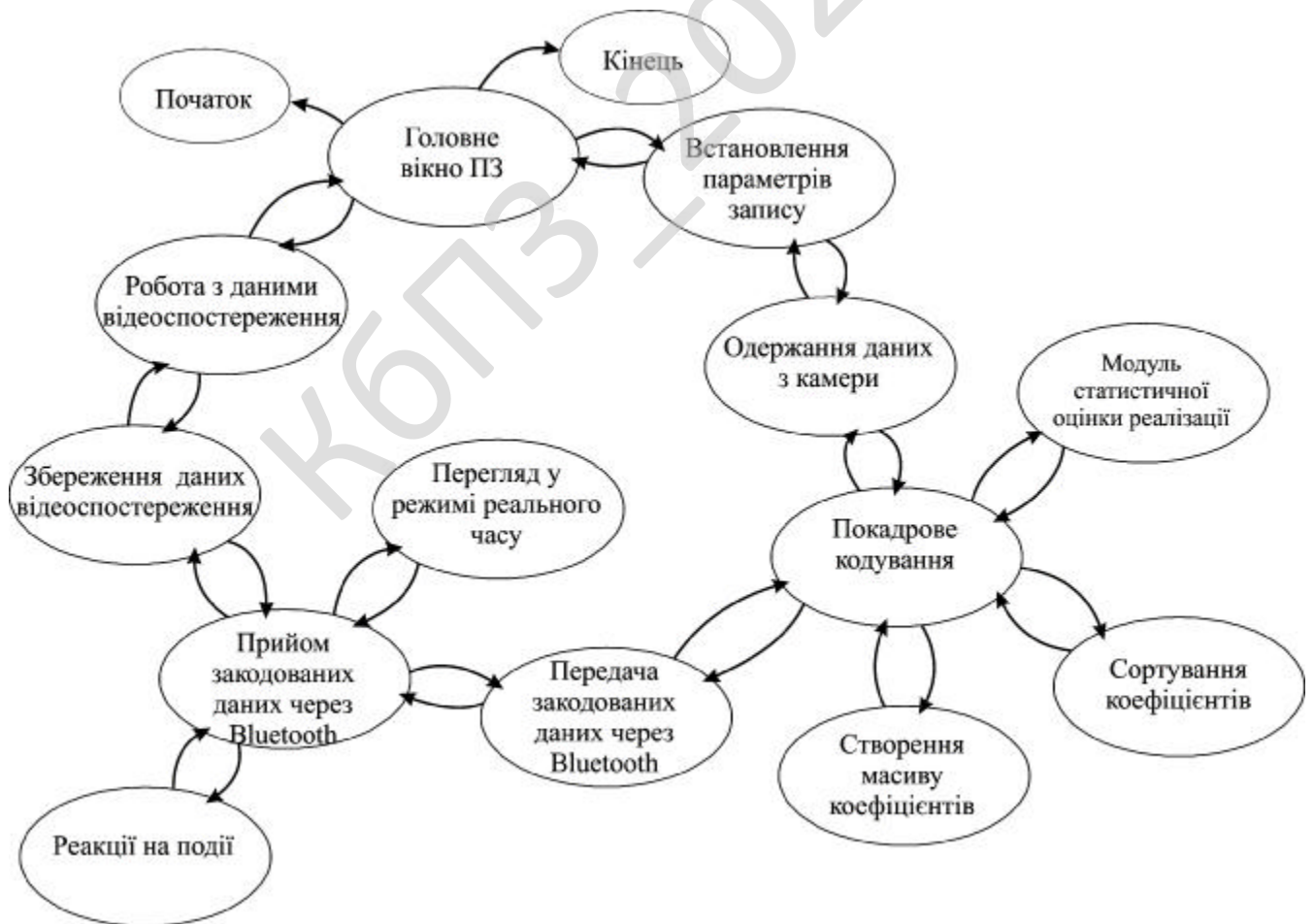


Рисунок 3.3 – Діаграма взаємодії процесів

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

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

## 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 оцінки реалізації відео-конференц-зв'язку.

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-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

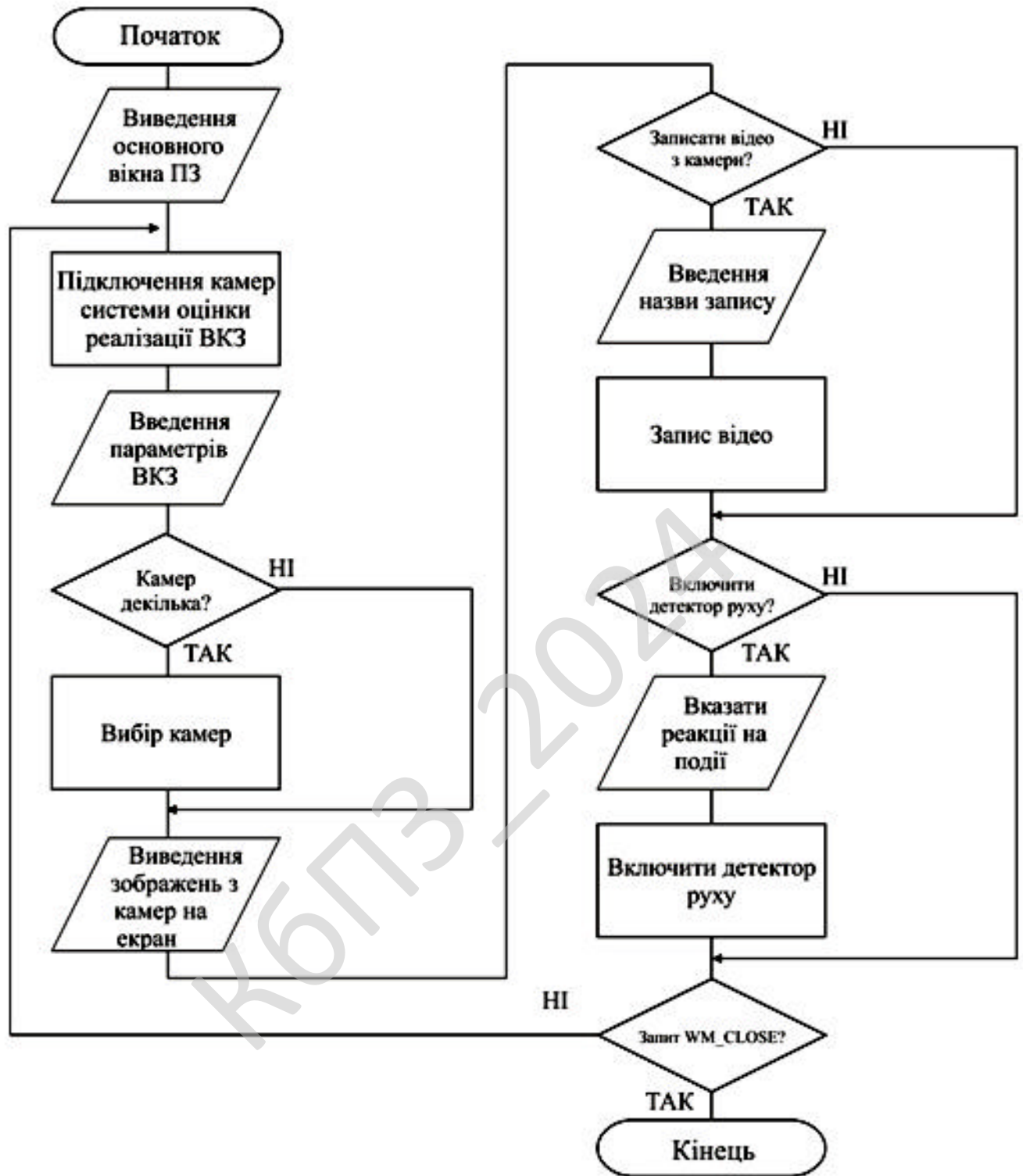


Рисунок 4.1 – Блок-схема основної програми

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

При розробці програми було використано архітектуру клієнт-сервер що є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними.

Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення.

Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

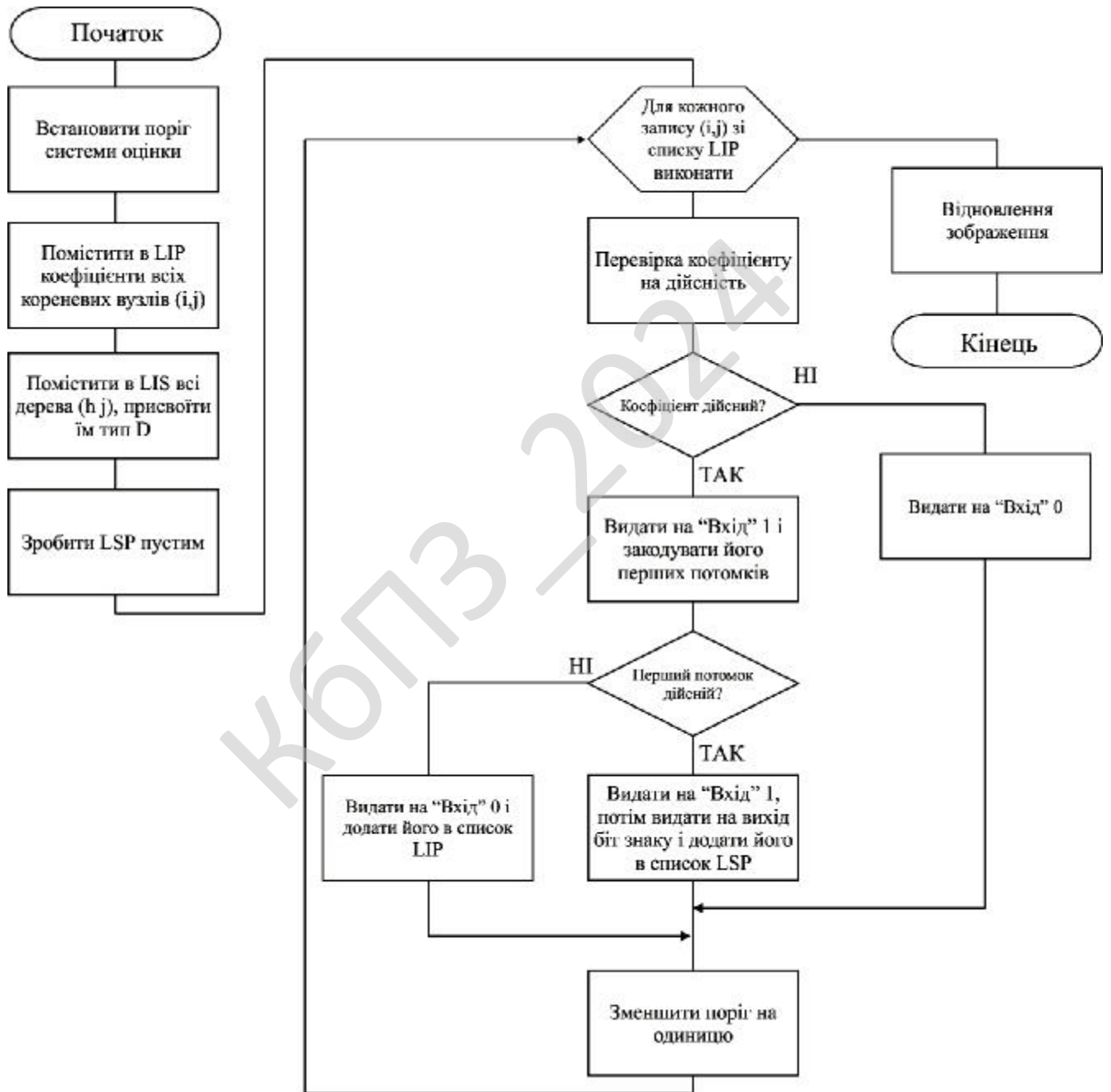


Рисунок 4.2 – Блок-схема роботи підпрограми

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<b>48</b>

розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерями.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ.

Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

та інстальовати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу.

Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL.

Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі.

Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft.

Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language). Нижче наведемо код алгоритму кодування.

```
namespace SPIHT
{
    using System;
    using System.Drawing;
    using System.IO;
    using System.Threading;
    using System.Net;
    using videource;
    /// <summary>
    /// SPIHTSource - Алгоритм SPIHT
    /// </summary>
    public class SPIHTSource : IVideoSource
    {
        private stringsource;
        private stringlogin = null;
        private stringpassword = null;
        private objectuserData = 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 Threadthread = null;
        private ManualResetEvent stopEvent = null;
        // нова подія у фреймі
        public event CameraEventHandler NewFrame;
        // SeparateConnectioGroup властивості
        // індикатор відкриття WebRequest у групі
```

```

public bool    SeparateConnectionGroup
{
    get { return useSeparateConnectionGroup; }
    set { useSeparateConnectionGroup = value; }
}
// PreventCaching властивості
// Якщо властивості є правильними, то керуємо параметрию 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 паролкування
public string Password
{
    get { return password; }
    set { password = value; }
}
// FramesReceived властивості
public int    FramesReceived
{

```

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

        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 SPIHTSource()
    {
    }
}

```

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

// 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();
    }
}

```

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

        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
    HttpWebRequest 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);

```

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

        // 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

```

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

        span = DateTime.Now.Subtract(start);
        // засипання у мілісекундах
        int msec = frameInterval - (int) span.TotalMilliseconds;
while ((msec > 0) && (stopEvent.WaitOne(0, true) == false))
    {
        // засипання ...
        Thread.Sleep((msec < 100) ? msec : 100);
        msec -= 100;
    }
}
catch (WebException ex)
{
System.Diagnostics.Debug.WriteLine("=====: " + ex.Message);
        // чекаємо у циклі наступну спробу
        Thread.Sleep(250);
    }
catch (Exception ex)
{
System.Diagnostics.Debug.WriteLine("=====: " + ex.Message);
    }
finally
{
        // помилка запиту
        if (req != null)
        {
            req.Abort();
            req = null;
        }
        // закриваємо потоку
        if (stream != null)
        {
            stream.Close();
            stream = null;
        }
        // close response
        if (resp != null)
        {
            resp.Close();
            resp = null;
        }
    }
}

```

						<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			57

```

// need to stop ?
if (stopEvent.WaitOne(0, true))
    break;
    }
}
}
}
}

```

## 4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7564:2014 («Купина»). В Україні на основі консервативного підходу із залученням відомих і добре досліджених конструкцій була розроблена геш-функція, що базується на новому блоковому шифрі „Калина” (ДСТУ 7624:2014).

Національний стандарт ДСТУ 7564:2014 визначає криптографічну функцію гешування „Купина”, додатковий режим її застосування для формування коду автентифікації повідомлення (імітовставки), а також значення для перевірки реалізацій.

Для скорочення обсягу тексту національного стандарту була застосована математична нотація, що дозволяє отримати точний і компактний запис.

Водночас, такий підхід може ускладнювати сприйняття сутності алгоритму для фахівців, що не мають фундаментальної криптологічної освіти.

У роботі наводиться розгорнутий альтернативний опис функції гешування „Купина” із позначеннями, традиційними для галузі комп’ютерних наук.

### Термінологія та позначення

Вектор ініціалізації – бітова послідовність фіксованої довжини (512 або 1024 біта), що використовується як початкове значення при обчисленні геш-значення.

Внутрішній стан – бітова послідовність фіксованої довжини (512 або 1024 біта), що є проміжним значенням на кожній ітерації перетворення функції

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

гешування, а також вхідним та вихідним значеннями перетворень  $P$  і  $Q$ ; для цих перетворень внутрішній стан подається як матриця розміром  $8 \times c$  байт.

Геш-значення (геш-вектор) – бітова послідовність фіксованої довжини ( $n = 8 \cdot s, s \in \{1, 2, \dots, 64\}$ ), що є результатом роботи функції гешування.

Доповнення – вставка додаткових біт у кінець повідомлення для отримання кратності довжини бітової послідовності довжині внутрішнього стану функції гешування.

Повідомлення – бітова послідовність довжини від 0 біт (порожній рядок) до  $2^{96} - 1$  біт.

Функція стиснення – ітеративне перетворення, що відображає  $l$ -бітний блок повідомлення та  $l$ -бітне значення, отримане функцією стиснення на попередньому кроці, у нове  $l$ -бітне значення.

Далі використовуються наступні позначення:

- $\oplus$  – додавання за модулем 2 (XOR);
- $0x$  – префікс числа, що записане у шістнадцятковій системі числення;
- $a \bmod b$  – ціле невід’ємне число, що дорівнює залишку від ділення цілого числа  $a$  на натуральне число  $b$ ;
- $B_i$  –  $i$ -й байт вхідної послідовності;
- $C^i$  – константа перетворення XORRoundKey або Add64RoundKey для  $i$ -го циклу;
- $c$  – кількість стовпців внутрішнього стану в матричному поданні;
- $\phi$  – функція стиснення;
- $H$  – визначена у стандарті функція гешування;
- $H(M)$  – результат обчислення функції гешування для повідомлення  $M$  (гешзначення);
- $IV$  – вектор ініціалізації;
- $l$  – розмір внутрішнього стану функції гешування (у бітах),  $l \in \{512, 1024\}$ ;
- $M$  – повідомлення;

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>59</b>

- $m_i$  –  $i$ -й блок повідомлення  $M$ ;
- $n$  – довжина обчисленого геш-значення;
- $N$  – довжина повідомлення  $M$  без доповнення;
- $P, Q$  – складові перетворення функції стиснення;
- $P_{512}$  – перетворення  $P$  для 512-бітного внутрішнього стану;
- $P_{1024}$  – перетворення  $P$  для 1024-бітного внутрішнього стану;
- $Q_{512}$  – перетворення  $Q$  для 512-бітного внутрішнього стану;
- $Q_{1024}$  – перетворення  $Q$  для 1024-бітного внутрішнього стану;
- $r$  – кількість ітерацій у перетвореннях  $P$  і  $Q$  ( $r = t$  в ДСТУ 7564:2014);
- $S$  – внутрішній стан геш-функції;
- $t$  – кількість блоків  $m$ , з яких складається повідомлення  $M$ , включаючи доповнення;
- $v_i$  –  $i$ -й біт вхідної послідовності;
- $\Omega$  – завершальне перетворення;
- Купина- $n$  – режим використання функції гешування з усіченням обчисленого гешзначення до розміру  $n$  біт.

### Загальні положення

Під функцією гешування  $H$  розуміється залежне від вектора ініціалізації IV відображення послідовності біт  $M$  у геш-значення  $H(M)$  фіксованої довжини  $n$ .

ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно.

Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт.

Режим роботи для формування геш-значення довжиною  $n$  біт позначається як «Купина- $n$ ».

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Основними режимами роботи функції гешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

### Структура перетворення

Функція гешування, визначена в ДСТУ 7564:2014, формує геш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до  $2^{96}-1$  біт.

При формуванні геш-значення повідомлення доповнюється, далі поділяється на  $l$ -бітні блоки  $m_0, \dots, m_t$ , після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення  $\varphi$ .

При цьому формуються значення  $h_i = \varphi(h_{i-1}, m_i)$  де  $i = 1, \dots, t$ , а початкове значення  $h_0 = IV$ .

Після обробки останнього блоку повідомлення результуюче геш-значення обчислюється як  $H(M) = \Omega(h_t)$ , де  $\Omega$  – завершальне перетворення, що повертає  $n$  - бітне значення, кратне 8 ( $0 < n \leq l/2$ ).

КБПЗ-2024

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи.

Розроблене програмне забезпечення системи оцінки реалізації відео-конференц-зв'язку складається з наступних функціональних блоків:

– Навігаційне меню: Файл; Камери; Відеозаписи; Детектор руху; Параметри; Довідка.

– Функції представлені у графічному вигляді (іконки).

– Блоку відображення інформації камер та архіву записів.

– Блоку відображення відео потоку з камер спостереження. Можуть бути обрані як поворотні камери так і роботизовані моделі камер (автоматичний похук).

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.



Рисунок 5.1 – Головне вікно розробленого ПЗ

					ВКРМ-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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

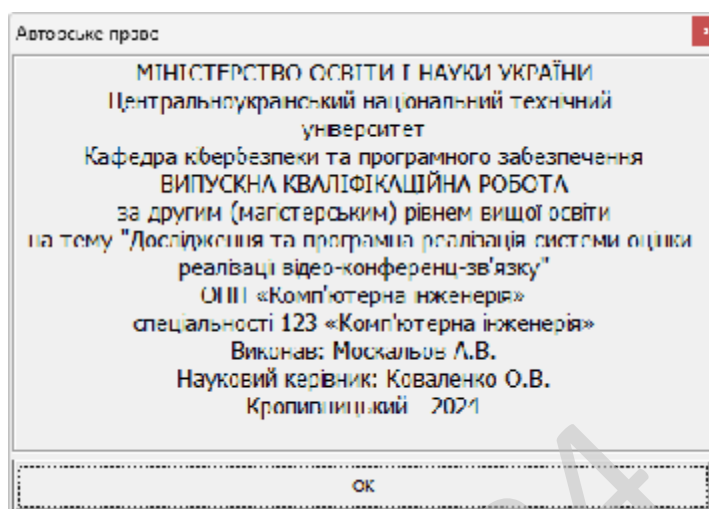


Рисунок 5.2 – Вікно авторського права

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

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

Проводилась оцінка:

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

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

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

Проводилось тестування форматом чорної скриньки.

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

Ці тести демонструють:

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;

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

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

Обрано умови розповсюдження – Shareware.

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

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи оцінки реалізації відео-конференц-зв'язку.

*Метою розробки є дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.*

*Об'єктом дослідження є процес оцінки реалізації відео-конференц-зв'язку.*

*Предметом дослідження є методи оцінки реалізації відео-конференц-зв'язку.*

*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 оцінки реалізації відео-конференц-зв'язку.
- Розроблено вітчизняний продукт оцінки реалізації відео-конференц-зв'язку, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи оцінки якості відео-конференц-зв'язку можуть зацікавити (рисунок 7.1).

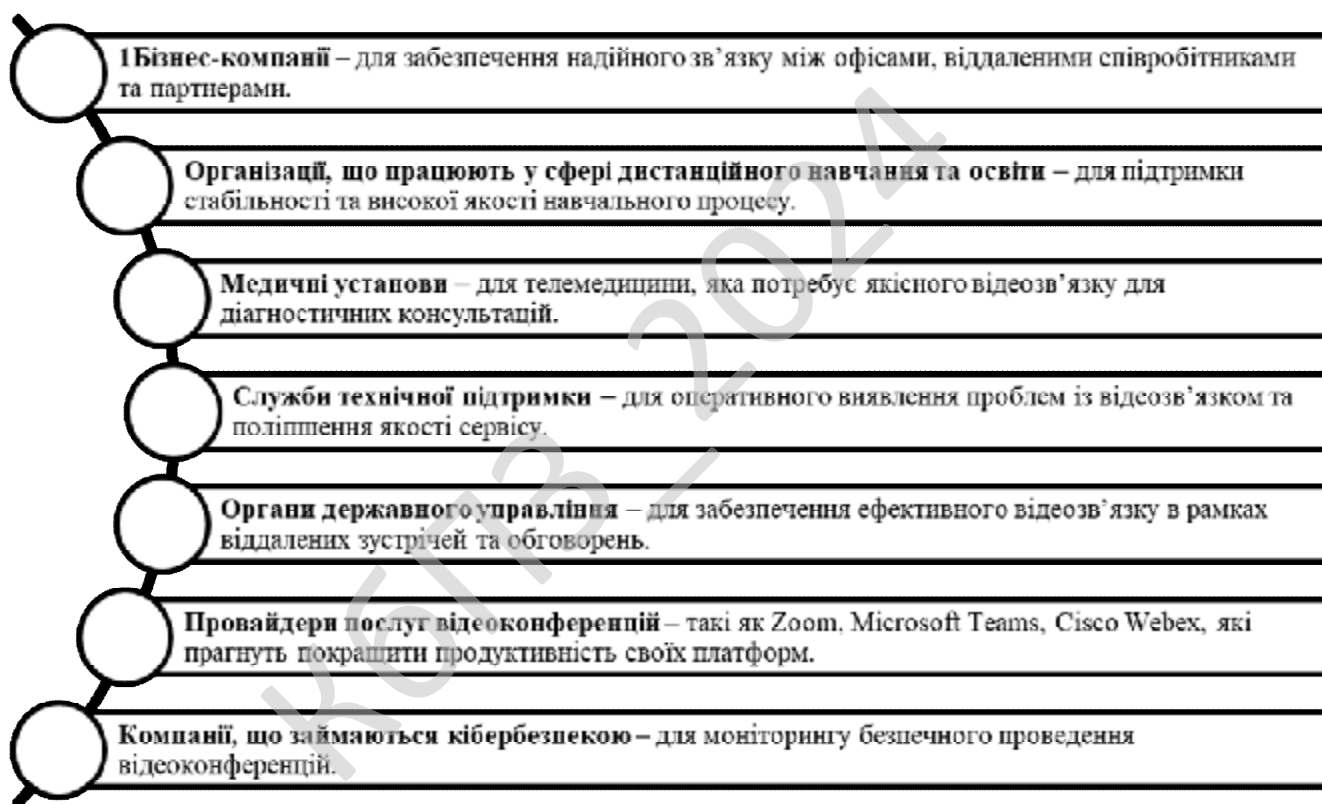


Рисунок 7.1 – Цільова аудиторія

Широка цільова аудиторіє може стати запорукою успішного просування проекту.

## 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи оцінки відео-конференц-зв'язку за допомогою методу експертних оцінок може включати такі кроки: визначення критеріїв, формування експертної групи, розробка шкали оцінювання, оцінку експертами, обробка результатів.

Спочатку потрібно визначити ключові критерії, за якими експерти оцінюватимуть систему (рисунок 7.2).

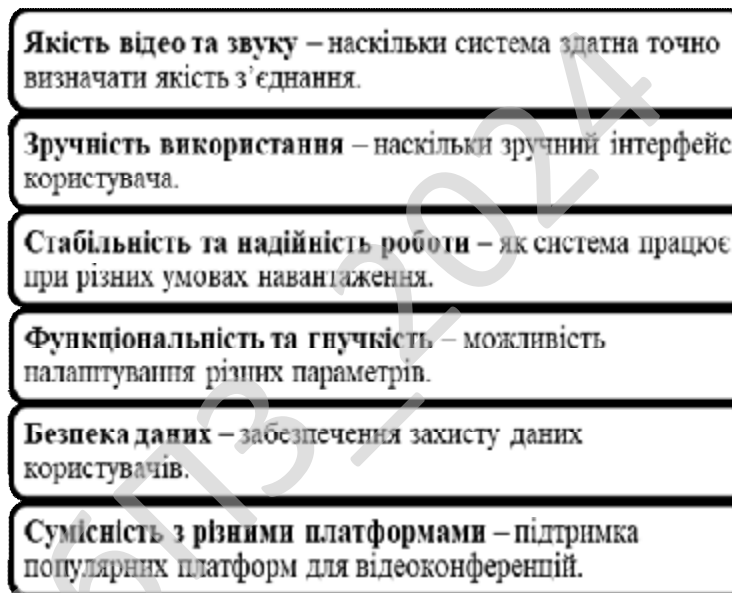


Рисунок 7.2 – Критерії експертної оцінки

Залучаються експерти у сфері відеоконференцій, інформаційних технологій, UI/UX дизайну, кібербезпеки та підтримки клієнтів. Обрано групу з 2 експертів.

Слід використовувати 5-бальну шкалу, де 1 – дуже низький рівень привабливості, а 5 – дуже високий.

Кожен експерт оцінює кожен критерій за визначеною шкалою:

– експерт 1: якість відео – 4, зручність використання – 5, стабільність – 4, функціональність – 3, безпека – 5, сумісність – 4.

– експерт 2: якість відео – 5, зручність використання – 4, стабільність – 5, функціональність – 4, безпека – 4, сумісність – 5.

Далі розраховується середній бал за кожним критерієм та загальний середній показник привабливості: якість відео:  $(4 + 5) / 2 = 4.5$ , зручність використання:  $(5 + 4) / 2 = 4.5$ , стабільність:  $(4 + 5) / 2 = 4.5$ , функціональність:  $(3 + 4) / 2 = 3.5$ , безпека:  $(5 + 4) / 2 = 4.5$ , сумісність:  $(4 + 5) / 2 = 4.5$ .

Загальний середній бал:  $(4.5 + 4.5 + 4.5 + 3.5 + 4.5 + 4.5) / 6 \approx 4.33$

На основі отриманих оцінок робиться висновок про загальну привабливість програмної реалізації системи. Наприклад, із середнім балом 4.33 можна вважати, що система є привабливою для ринку, але варто звернути увагу на функціональність, яку можна поліпшити.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи оцінки відео-конференц-зв'язку найкраще підходять наступні методи: метод оцінки за трудомісткістю та метод порівняння. Перший підходить, коли можна оцінити функціональні вимоги. Цей метод дозволяє визначити обсяг роботи на основі функціональних точок (кількість функцій системи) і потім перерахувати їх у вартість, враховуючи середню вартість години роботи розробників. Другий – якщо є доступ до даних щодо вартості схожих проектів, можна провести порівняння із подібними програмними рішеннями. Це дає реалістичні показники, базуючись на ринкових цінах для проектів схожого масштабу та функціоналу. Якщо існує достатньо інформації про функціональні вимоги та обсяги робіт, метод оцінки за трудомісткістю або СОСОМО можуть бути найточнішими. Для комплексного та гнучкого підходу можна комбінувати експертні оцінки з компонентною оцінкою, що дозволить отримати достовірну оцінку вартості, враховуючи всі етапи та ризики проекту.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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

Економічна ефективність від впровадження системи оцінки якості відео-конференц-зв'язку може проявлятися в таких аспектах (рисунок 7.3). З огляду на вищезазначені показники, період окупності такої системи буде не більше одного року.

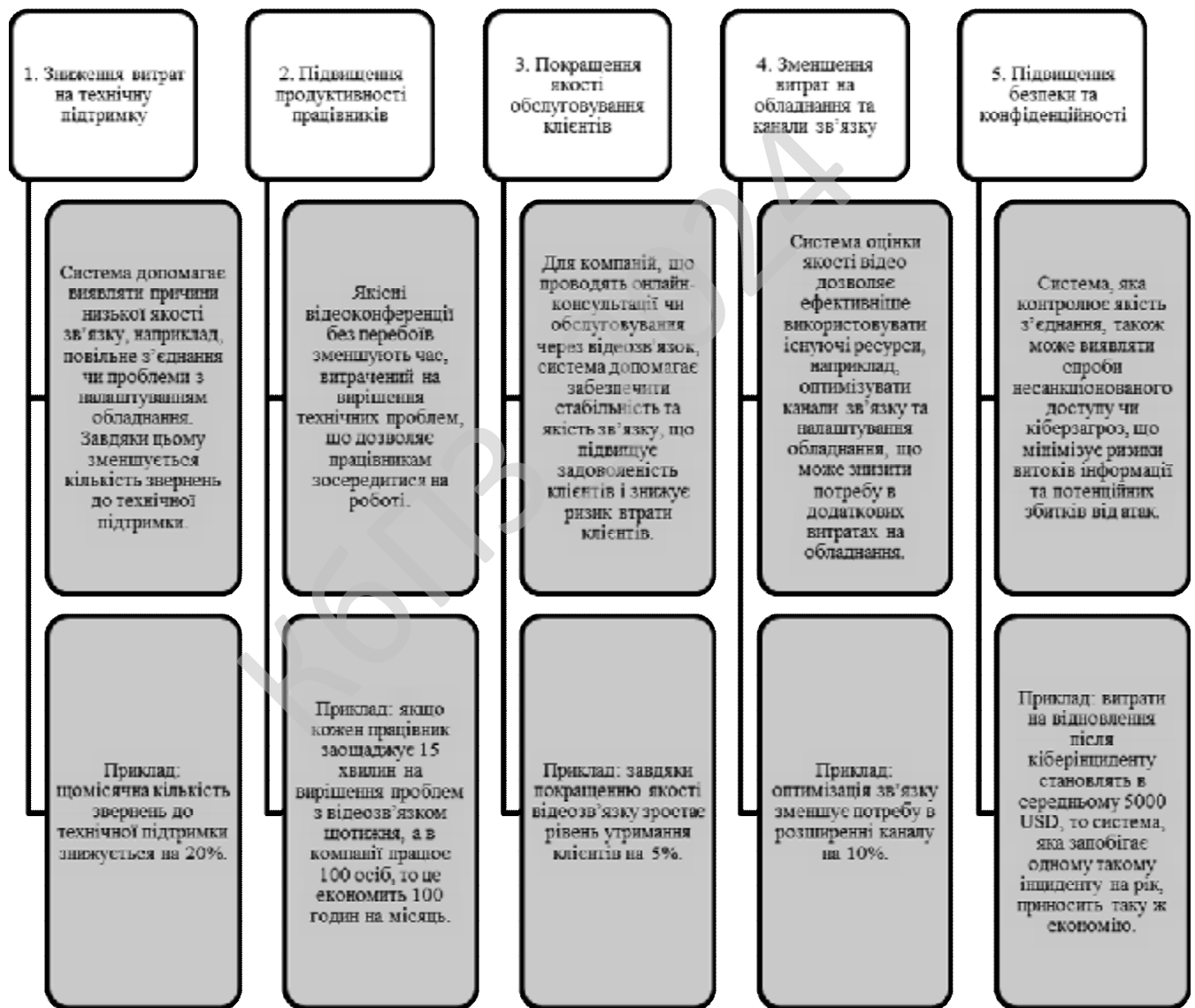


Рисунок 7.3 – Ключові аспекти ефективності впровадження проєкту для клієнта

## 7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проєкту програмної реалізації системи оцінки якості відео-конференц-зв'язку може включати етапи, наведені нижче (рисунок 7.4).

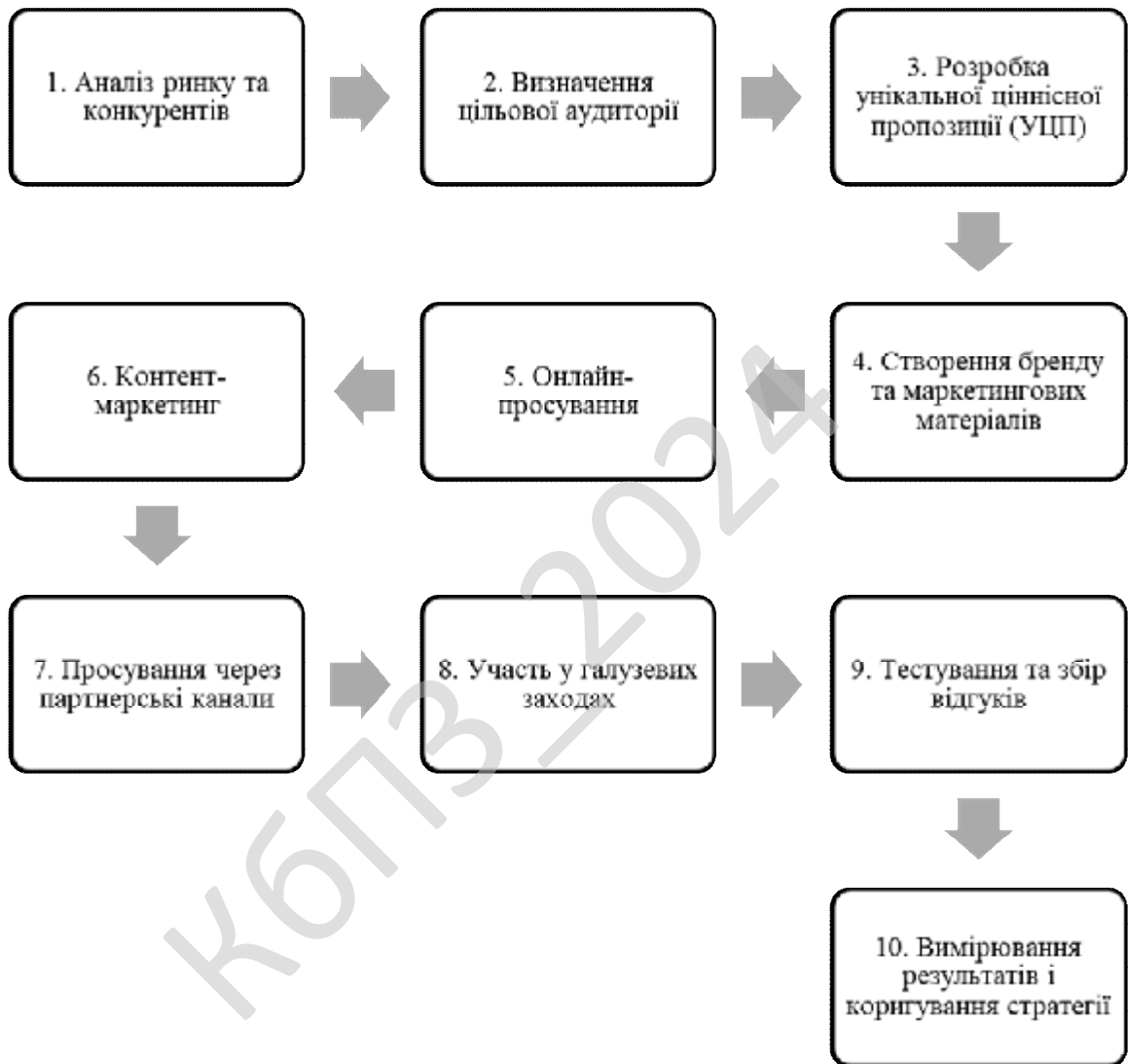


Рисунок 7.4 – Алгоритм просування проєкту

Такі кроки дозволяють комплексно підійти до просування, охоплюючи ключові канали та сегменти цільової аудиторії, що підвищує ймовірність успішного впровадження та розширення проєкту на ринку.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту для проєкту програмної реалізації оцінки якості відео-конференц-зв'язку може включати такі стратегічні кроки (рисунок 7.5).

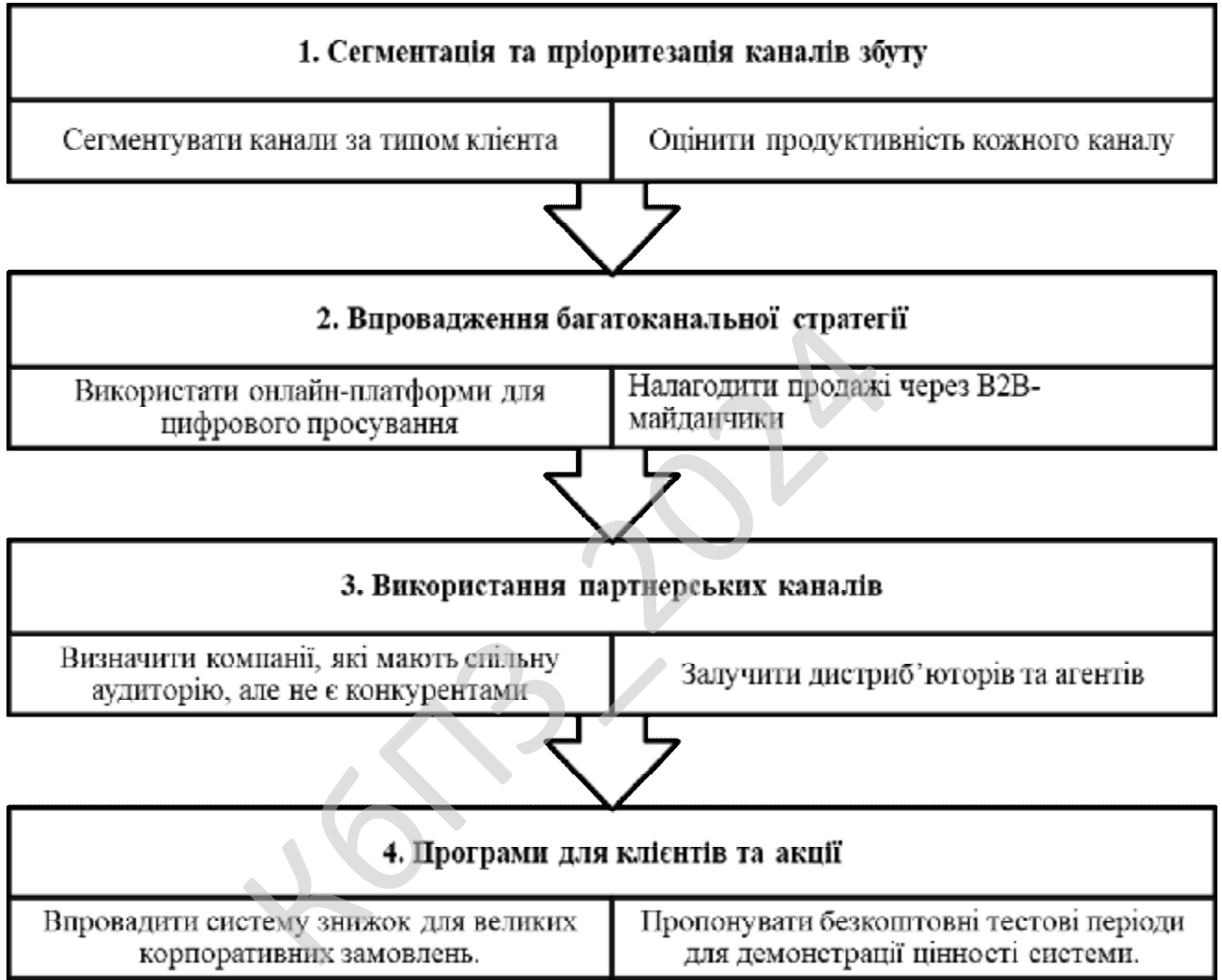


Рисунок 7.5 – Шляхи оптимізації каналів збуту

Ці кроки дозволять створити гнучку та продуктивну систему збуту, що підвищить обсяг продажів, розширить охоплення аудиторії та забезпечить стійкий ріст проєкту.

## 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху для проєкту програмної реалізації системи оцінки якості відео-конференц-зв'язку включають (рисунок 7.6). Ці фактори створюють основу для успішного виходу продукту на ринок і забезпечують його стабільне зростання та високу репутацію серед користувачів.



Рисунок 7.6 – Шляхи оптимізації каналів збуту

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Умови праці програміста включають наступні фактори:

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

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

## 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 програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,38
Довжина	5,95
Висота	2,8

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

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м <sup>2</sup>	не менше 6.0	8
Об'єм, V	м <sup>3</sup>	не менше 20.0	22,4

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

У зазначеному приміщенні працюють четверо людей. За даними, які наведено у табл. 8.1- 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Таним чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають нормативним вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація.

Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії важкості робіт 1а, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для 1а			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

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

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об’єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об’єкта розрізнення (символів на екрані дисплея), з темним тлом

(під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк [1]/ Крім того, все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

#### 8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

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

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

### 8.5 Розрахункова частина

Початкові дані для розрахунку захисного штучного заземлення: опір заземлювача, який нормується:  $R_{3Н} = 4 \text{ Ом}$ .

Для захисного штучного заземлення застосовуються вертикальні електроди з металевого прутка діаметром 30 мм ( $D = 30 \text{ мм} = 0,03 \text{ м}$ ) довжиною  $L=2,5 \text{ м}$  та горизонтальний електрод – металева полоса з перетином 40x4 мм. тип ґрунту – глина (питомий опір 40 Ом·м). Відстань між вертикальними заземлювачами (електродами)  $A=3 \text{ м}$ .

Глибина закладення горизонтального контура заземлення  $t = 0,8 \text{ м}$ .

Умовна товщина верхнього шару ґрунту:  $H=0,4 \text{ м}$ . Напруга – 220/380 В.

Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

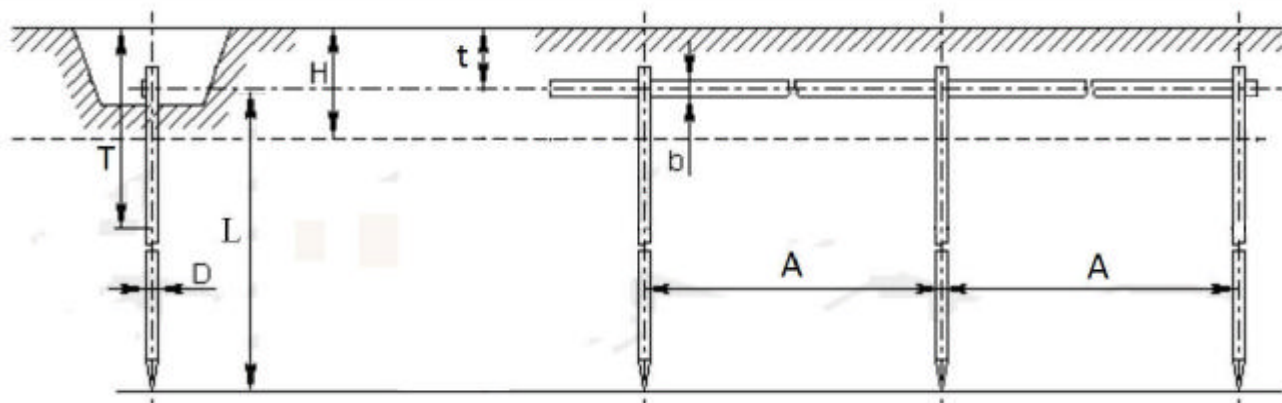


Рисунок 8.1 – Схема штучного заземлення

Виконаємо розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0,8 + 2,5/2 = 2,05 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho_2 = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м,}$$

де  $\psi = 1,36$  – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [8];

$\rho_1 = 50 \text{ Ом}\cdot\text{м}$ . – табличне значення питомого опору верхнього шару ґрунту [6];

$\rho_2 = 40 \text{ Ом}\cdot\text{м}$  – табличне значення питомого опору нижнього шару ґрунту [8].

Опір розтіканню електричного струму одного електрода вертикального заземлювача [8]:

$$R_{\text{о}} = 0,366 \frac{\rho}{L} \left( \lg \frac{2L}{D} + \frac{1}{2} \lg \frac{4T + L}{4T - L} \right) = 0,366 \frac{54,5}{2,5} \left( \lg \frac{2 \cdot 2,5}{0,03} + \frac{1}{2} \lg \frac{4 \cdot 2,05 + 2,5}{4 \cdot 2,05 - 2,5} \right) = 20,1 \text{ Ом.}$$

Відношення  $A/L = 3/2,5 = 1,2$ .

Визначаємо коефіцієнт екранування вертикальних електродів  $K_{ев}=0,8$  при попередній (орієнтовній) кількості вертикальних електродів, яке дорівнює 4 [8].

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

$$N=R_0 / ( K_{ев} \cdot R_{зН} ) = 20,1 / (0,8 \cdot 4) = 5,87 \approx 6 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси [8]:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 6 = 18,8 \approx 19 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси [8]:

$$\begin{aligned} R_{\Pi} &= 0,366 \cdot (\rho_2 \cdot K_{\Pi} / L_{\Pi} ) \cdot \lg(2(L_{\Pi} \cdot L_{\Pi} ) / (K \cdot t)) = \\ &= 0,366 \cdot (40 \cdot 5 / 16) \cdot [\lg(2 \cdot 16 \cdot 16) / (0,04 \cdot 0,8)] = 16,5 \text{ Ом.} \end{aligned}$$

де  $K_{\Pi} = 5$  – табличне значення коефіцієнта сезонності для відповідної кліматичної зони з'єднуючої полоси [8].

Загальний опір розтіканню електричного струму заземлювача [8]:

$$\begin{aligned} R &= (R_0 \cdot R_{\Pi}) / ( R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев} ) = \\ &= (20,1 \cdot 16,5) / (20,1 \cdot 0,75 + 6 \cdot 16,5 \cdot 0,8) = 3,32 \text{ Ом.} \end{aligned}$$

де  $\eta_{\Pi} = 0,75$  – табличне значення коефіцієнта екранування з'єднуючої полоси [8].

Умова  $R \leq R_{зН}$  виконується ( $3,32 \leq 4 \text{ Ом}$ ).

### Висновки до розділу

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

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

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	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 оцінки реалізації відео-конференц-зв'язку.

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 оцінки реалізації відео-конференц-зв'язку.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем оцінки реалізації відео-конференц-зв'язку.

– Досліджена система оцінки реалізації відео-конференц-зв'язку.

– На основі отриманих результатів досліджень створена програмна реалізація системи оцінки реалізації відео-конференц-зв'язку.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання оцінки реалізації відео-конференц-зв'язку.

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

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

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	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 7564:2014 («Купина»).

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

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

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>84</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Москальов А.В. Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press. 2022 1677 p.
3. Will Grant. 101 UX Principles. Packt Publishing. 2022. 432 p.
4. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
5. Henry Lloyd. Interactive Computer Graphics. States Academic Press. 2022. 247 p.
6. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
7. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
8. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
9. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
10. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
11. 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.
12. 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. ун-т. – Д.: НГУ, 2016. – 187 с.
13. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.

					<b>ВКРМ-123.24.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

14. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
15. 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.
16. 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
17. 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.
18. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
19. 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.
20. 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,
21. 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.

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

23. 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.

24. 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.

25. 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.

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

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

49. 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.

50. 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.

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-123.24.0025.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Москальов А.В.				Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи оцінки реалізації відео-конференц-зв'язку.

## 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 оцінки реалізації відео-конференц-зв'язку.

## 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-123.24.0025.00.00.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 оцінки реалізації відео-конференц-зв'язку;
- 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-123.24.0025.00.00.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-123.24.0025.00.00.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 санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-123.24.0025.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 90 аркушів.

## 10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

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

					<b>ВКРМ-123.24.0025.00.00.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

\_\_\_\_\_ Коваленко О.В.

*Дослідження та програмна реалізація  
системи оцінки реалізації відео-конференц-зв'язку*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 41

Літера: РП

Кропивницький – 2024 року

Файл `MultimodeVideoSource.cs` - робота з відеокамерами

```

namespace multisource
{
    using System;
    using System.Collections;
    using videosource;
    using SPIHT;
    using mSPIHT;

    /// <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.SPIHT:
                            videoSource = new SPIHTSource();
                            break;
                        case StreamType.MSPIHT:
                            videoSource = new MSPIHTSource();
                    }
                }
            }
        }
    }
}

```

```

        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 CameraInfo.cs – отримання інформації про камеру**

```

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;
        }
    }
}
}
```

К6П3\_2024

## Файл Multiplexer.cs – мультиплексування (для роботи з декількома камерами)

```

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;

```

```

80);
    this.cameraWindow10.Location = new System.Drawing.Point(328,
    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 виключає 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 SPIHTSource.cs - алгоритм кодування SPIHT

```

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

    using videosource;

    /// <summary>
    /// SPIHTSource - SPIHT скачувач
    /// </summary>
    public class SPIHTSource : 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 параметрию 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 SPIHTSource()
{
}

// 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 SPIHTSourcePage.cs - алгоритм кодування SPIHT (інтерфейс)

```

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

namespace SPIHT
{
    /// <summary>
    /// Основні дескриптори для SPIHTSourcePage.
    /// </summary>
    public class SPIHTSourcePage : 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 SPIHTSourcePage()
        {
            // 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;
//
// SPIHTSourcePage
//
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 = "SPIHTSourcePage";
        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()
    {
        SPIHTConfiguration config = new SPIHTConfiguration();

        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)
    {
        SPIHTConfiguration cfg = (SPIHTConfiguration) 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 = @"МАГІСТЕРСЬКА ДИПЛОМНА РОБОТА

```

на тему:

Дослідження та програмна реалізація системи оцінки реалізації відео-конференц-зв'язку

Керівник: Коваленко О.В.

Розробив: студент Москальов Антон Віталійович  
KI-23M

```

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
}
}

```