

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 першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“Програмне забезпечення системи кібербезпеки для**  
**відеоспостереження за багатоквартирними житловими**  
**будинками”**

Виконав здобувач вищої освіти  
IV курсу, групи КБ-20  
ОПП «Кібербезпека»  
спеціальності 125 «Кібербезпека»  
\_\_\_\_\_ Кривда О.С.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

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

**Центральноукраїнський національний технічний університет**

Факультет *Механіко-технологічний*

Кафедра *Кібербезпеки та програмного забезпечення*

Освітній ступінь *бакалавр*

Галузь знань . 12 *“Інформаційні технології”*

Спеціальність *125 “Кібербезпека”*

Освітньо-професійна (освітньо-наукова) програма *“Кібербезпека”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2024 року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

*Кривді Олексію Сергійовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи

*Програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками*

2. Керівник роботи

*Смірнов Олексій Анатолійович, докт. техн. наук, професор*

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

затверджені наказом вищого навчального закладу № 135-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту *23.05.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи кібербезпеки в промислову експлуатацію.*

*6. Висновки*

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

*Структурна схема системи кібербезпеки*

*1 аркуш*

*Функціональна схема системи кібербезпеки*

*1 аркуш*

*Діаграма процесів*

*1 аркуш*

*Блок-схема алгоритму роботи додатку*

*2 аркуша*

7. Дата видачі завдання « 17 » січня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання  
« 17 » січня 2024 р.

Підпис керівника

Смірнов О.А.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2024 р.

Підпис здобувача

Кривда О.С.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Кривда О.С. Програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

Метою розробки є програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

Результат роботи – програмна реалізація системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

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 Delphi 10.4.

**Ключові слова:** кібербезпека, відеоспостереження

## ABSTRACT

**Kryvda O.S. Cyber security system software for video surveillance of multi-apartment residential buildings. 125 Cyber security. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the first (bachelor) level of higher education, software is developed, which is intended for a cyber security system for video surveillance of multi-apartment residential buildings.

The purpose of the development is the software of the cyber security system for video surveillance of multi-apartment residential buildings.

The result of the work is the software implementation of a cyber security system for video surveillance of multi-apartment residential buildings.

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

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

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

The program was developed in the Delphi 10.4 environment.

**Keywords:** cyber security, video surveillance

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	19
2.3 Розгорнута постановка завдання .....	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	27
3.1 Опис функціонування системи .....	27
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми .....	34
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	40
4.2 Захист розробленого програмного забезпечення.....	51
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	53
6 ОСНОВНІ ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	57

					ВКРБ-125.24.0011.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Кривда О.С.				Програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками	Літ.	Аркуш	Аркушів
Перев.	Смірнов О.А.					Б	1	63
Н.контр.	Коваленко А.С.				ЦНТУ КБ-20			
Затв.	Смірнов О.А.							

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DS – DirectShow мультимедійне розширення Microsoft;

TrF1 – TranformFilter -Трансформ-фільтр;

RnF1 – RendererFilter – Рендер-фільтр;

SourceFilter – SourceFilter – Фільтр джерела;

IC – Intelligent Connect – інтелектуальне з'єднання;

ООП – об'єктно-орієнтоване програмування;

DSP – бібліотека компонентів для розробки мультимедійних програм;

VWINDOW – вертикальна синхронізація;

HWINDOW – горизонтальна синхронізація;

PLL – Цикл Блокування Стадії;

USB – Universal Serial Bus – універсальна послідовна шина;

URL – universal resource locator – локатор ресурсів інтернет;

HTML – мова розмітки гіпертекстових документів;

ОС – операційна система;

ПЕОМ – персональна електронно обчислювальна машина;

КС – комп'ютерна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ВСТУП

**Актуальність теми.** Для забезпечення безпеки мешканців, запобігання незаконних зазіхань на припарковані у дворі автомобілі, як прибудинкових територіях багатоквартирних житлових будинків, так і на території, що прилягає до двору приватного житлового будинку (або у дворі житлового будинку), власники часто вирішують установити камери відеоспостереження.

Переваги такого рішення очевидні – як правило, правопорушник, що виявив систему відеоспостереження, відмовиться від реалізації злочинного задуму, але навіть якщо НП все-таки відбулося, шанси на розшук порушника зростають у рази.

Відеоспостереження в багатоквартирному будинку необхідно для рішення безлічі завдань безпеки. Куди направилася Ваша дитина, вийшовши з під'їзду? Що відбувається на дитячому майданчику? У чи порядку автомобіль, залишений на ніч біля будинку. Хто заходив у під'їзд і підходив до квартири під час Вашої відсутності? Список питань, на які допоможе дати відповідь відеоспостереження в багатоквартирному будинку можна продовжувати.

Організувати установку подібних систем може власник однієї квартири, або жителі одного під'їзду й в ідеалі – всі жителі будинку.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

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

- Огляд існуючих систем для відеоспостереження за багатоквартирними житловими будинками.
- Дослідження системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– Програмна реалізація системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для відеоспостереження за багатоквартирними житловими будинками.

Таким чином, виходячи з вищеперахованого, програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ\_2024

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система, яка розробляється у даному бакалаврському проекті призначена для програмного забезпечення відеоспостереження за багатоквартирними житловими будинками. Якщо систему встановлює хазяїн однієї із квартир, то, швидше за все, він подбає в першу чергу про контроль своєї сходової площадки й простору у вхідних дверей у квартиру, а також місця паркування свого автомобіля біля будинку.

Якщо систему встановлюють жителі одного під'їзду, то контролюватися буде безпосередньо вхід у під'їзд і частину двору поблизу. А також, сходові площадки на всіх поверхах. Причому, для кожної квартири такий варіант обійдеться значно дешевше, ніж установка окремих систем відеоспостереження індивідуально.

Звичайно, найбільш ефективною буде установка системи відеоконтролю для всього будинку, або декількох будинків, що утворять один загальний двір. У цьому випадку обов'язково контролюються всі входи й в'їзди у двір, бажано великим планом. За допомогою декількох відеокамер спостерігається загальний план двору й звичайно всі входи в під'їзди. Так буде отримана найбільш повна картина що відбувається.

Розглянемо, що представляє із себе система відеоспостереження на прикладі одного під'їзду житлового будинку. Отже, на вулиці поставимо дві відеокамери по сторонах від входу, що дивляться хрест на хрест. Якщо перед під'їздом є дитячий майданчик, то одну камеру направляємо в її сторону. Всі вуличні відеокамери встановлюються вище в недоступних місцях.

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

У під'їзді біля входу розміщаємо ще одну відеокамеру, як правило, на стелі. Місце для цієї камери вибирається так, щоб великим планом знімати всіх тих людей, які входять.

Для запису зображень із відеокамер використовується спеціальний пристрій – відеореєстратор. Для перегляду записаної інформації, до нього підключається звичайний комп'ютерний монітор, або телевізор. Де буде встановлений записуючий пристрій, мешканці вирішують на загальних зборах.

Система відеоспостереження в багатоквартирному будинку дозволить не тільки розібратися в різних ситуаціях, але й запобігти небажаним інцидентам. Уже сама наявність відеокамер відлякує злочинця й дисциплінує хулігана.

## 1.2 Область застосування

Як установити камери відеоспостереження відповідно до вимог закону:

1. Система відеоспостереження повинна бути відкритою. Установка схованих камер, закамурфльованих по стіни, конструкції, побутові предмети, прямо заборонено законом і тягне настання відповідальності.

2. Відеоспостереження в багатоквартирному будинку. Якщо систему відеоспостереження необхідно встановити на прибудинковій території багатоквартирного будинку, необхідно:

– Провести загальні збори власників приміщень у багатоквартирному будинку, включивши до порядку денного питання установки системи відеоспостереження.

– Визначити місце розташування камер (у дворі житлового будинку із вказівкою точок монтажу, на горищах, у підвалах, сходових площадках).

– Визначити основні функціональні характеристики системи відеоспостереження – наявність цілодобового запису, наявність або відсутність екрана для перегляду записів у кожному окремому приміщенні та ін.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

– Оформити протокол загальних зборів, указавши в ньому згоду власників на встановлення системи відеоспостереження, а також наступні дані – функціональні особливості, вартість системи відеоспостереження, місце розташування камер, строк і умови зберігання відеозаписів і порядок надання записів власникам.

Звертаємо увагу, що збори вважаються правочинним, якщо в них беруть участь не менш 50% від загального числа голосів власників. Важливо: для ухвалення рішення необхідно дістати згоду не 2/3 присутніх на зборах, а 2/3 голосів всіх власників приміщень у багатоквартирному будинку. У випадку порушення цих вимог, власники, що вважають, що їхнього права порушені, вправі звернутися в суд і згодом система відеоспостереження буде демонтована. При установці систем відеоспостереження, необхідно розмістити таблички про те, що ведеться відеоспостереження.

3. Відеоспостереження на території, що прилягає до таунхаусу. Якщо камери відеоспостереження необхідно встановити на прилягаючій території таунхауса, бажано також дотримати вищевказаних рекомендацій дістати згоду власників всіх приміщень – правовий статус таунхауса в Україні не визначений на законодавчому рівні й можуть виникнути складності при віднесенні таунхауса до категорії багатоквартирних будинків.

4. Установлення відеокамери на сходовій площадці окремим власником. Заборони на встановлення камери відеоспостереження на сходовій площадці, для спостереження за окремо взятою квартирою, закон не містить. Однак, складності можуть виникнути при визначенні статусу сходової площадки. Так, якщо власник квартири встановлює на сходовій площадці камеру для спостереження за квартирою, без згоди власників інших житлових приміщень, розташованих на сходовій площадці, у випадку виникнення суперечки й судового розгляду, справа може закінчитися обов'язком власника зробити демонтаж системи відеоспостереження. Судова практика по даній категорії справ досить неоднозначна – суди приймають як рішення, що трактують установку камер на

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

сходовій площадці без згоди інших власників як порушення порядку володіння й користування сходовою площадкою як об'єктом спільної майно, так і рішення, що трактують установку камер як дії, спрямовані на захист власності й не порушують закон. У кожному разі, рішення як установити камеру на сходовій площадці – за власником квартири, однак одержання згоди інших власників, гарантовано позбавить від проблем у майбутньому. Рішенням, що виключає необхідність одержання згоди інших власників, є монтаж камери в конструкції вічка. У цьому випадку спільне майно власників будинку не використовується і єдина вимога, яку необхідно забезпечити при монтажі – не порушувати недоторканність приватного життя сусідів – камера повинна виключити огляд того, що відбувається в сусідній квартирі.

5. Установка камери відеоспостереження на території, що прилягає до приватного житлового будинку. Мабуть, найпростіше встановити систему відеоспостереження власникові частки будинку. Вимоги, які він повинен дотримати при установці – це відкритий характер відеоспостереження й виключення можливості огляду території сусіднього будинку – у випадку, якщо сусід порухає установку камери вторгненням у приватне життя й збором інформації про його приватне життя, суд може зобов'язати відповідача демонтувати систему відеоспостереження.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.24.0011.00.00.ПЗ	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 першим (бакалаврським) рівнем вищої освіти

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

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

Коли мова йде про завдання централізованого керування відеокамерами, існують два основних варіанти рішення – використання фірмового ПЗ, що поставляється виробником камер, і застосування продуктів незалежних розроблювачів. У першому випадку програмне забезпечення поставляється у вигляді безкоштовного додатка до встаткування, а в другому – майже завжди є додатковим джерелом витрат. Здавалося б, хто буде купувати стороннє ПЗ, якщо є фірмові розробки. Але на практиці в багатьох випадках «незалежне» програмне

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

забезпечення має більше широкі функціональні можливості, чим «рідні» продукти. До того ж фірмове ПЗ не дозволяє інтегрувати в мережу IP CCTV камери інших виробників, на відміну від програм спеціалізованих розроблювачів.

Справедливості заради варто відзначити, що прив'язка до власного устаткування відбувається не стільки зі шкідливості й небажання допустити камери інших виробників на проекти, скільки через технологічні особливості розвитку IP CCTV. Справа в тому, що сьогодні поки немає єдиних стандартів функціонування, яким би відповідали всі IP-камери. Виробіток загальних технологічних «правил гри» усе ще триває. Два найбільших альянси – ONVIF і PSIA – розвиваються паралельно.

Тому виробникові простіше й дешевше реалізувати повну підтримку власних стандартів устаткування, чим намагатися розробити універсальний програмний продукт. В умовах постійних цінових воєн, коли ринок активно завойовують недорогі IP-камери із КНР і Китайської Республіки (Тайваню), додаткові вкладення в розробку ПЗ природно здорожують і без того недешеві камери відеоспостереження європейських і американських виробників. При цьому розроблювачі спеціалізованого ПЗ всі сили направляють на те, щоб створювати як можна більше універсальні, гнучкі й функціональні продукти.

Що стосується безкоштовного ПЗ, то воно, як правило, являє собою «урізану» версію основної програми. Обмеження звичайно бувають по кількості камер, що підключаються, глибині зберігання відеоархіву й часу використання (до тридцяти днів). «Пробні» системи можуть також не підтримувати можливість подальшого нарощування й модернізації (у випадку покупки повної версії тестовий варіант не можна буде інтегрувати в загальну систему керування). Однак якщо для великої компанії тестової системи буде явно недостатньо, те невелика організація цілком може обійтися можливостями безкоштовного продукту.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Програмні системи для централізованого відеоспостереження існують для різних платформ – Windows, Linux, Mac OS, причому для кожної з них пропонується як мінімум один умовно безкоштовний продукт (табл. 2.1).

Таблиця 2.1 – Порівняння систем відеоспостереження

Компанія	Milestone	Ben Softwate	Zone Minder	Axxon Soft	ISS	"Скайрос"
Назва ПЗ	XProject Go	Security Spy	Zone Minder	Smart Start	SecurOS Lite	VideoNet Prime
Обмеження по кількості камер	8	-	-	16	4	16
Обмеження по кількості серверів	1	-	-	1	1	1
Обмеження відеоархіву	5 днів	-	-	1 ТБ	-	-
Підтримка H.264	+	-	-	+	+	+
Детектор руху	+	+	+	+	+	+
Аналітичні функції	-	+	+	+	-	+
Інтелектуальний пошук в архіві	-	+	-	+	-	+
Операційна система	Windows	Mac OS X	Linux	Windows	Windows	Windows
Обмеження за часом використання	-	30 днів	-	-	-	-

Відзначимо, що рекламні версії пропонують не тільки маловідомі компанії, але й світові лідери сегмента. Одна з них – датська компанія Milestone – також пропонує XProject Go (рисунок 2.1) – безкоштовну версію свого прапорманського продукту. Можливості цієї версії досить скромні. Система підтримує до восьми IP-камер, один сервер, усього лише п'ять днів зберігання відеоархіву, а також формат відеозапису avi. Перший платний варіант ПЗ Milestone здатний підтримувати 26 камер, п'ять користувачів, необмежений час зберігання даних, формати avi і jpg, а також віддалену роботу через веб-клієнт. Для порівняння: найбільш потужний варіант – XProject Corporate – здатний підтримувати необмежену кількість камер, користувачів і час зберігання відеоархіву, має розвинені аналітичні функції, можливість інтеграції ПЗ сторонніх виробників і т.д.

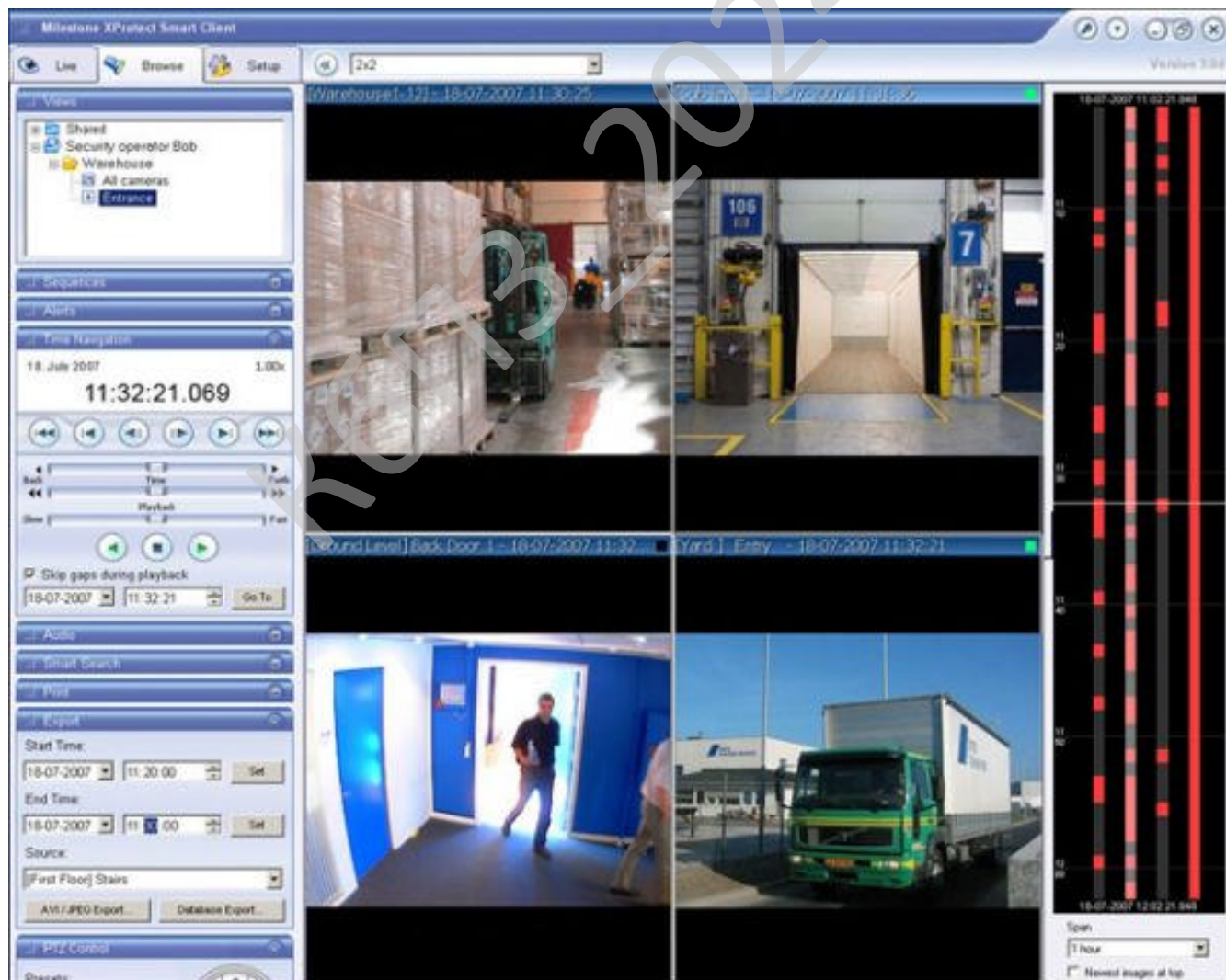


Рисунок 2.1 – Вікно керування ПЗ Milestone XProject Smart Client

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

У той же час XProject Go, як і всі розробки Milestone, підтримує біля дев'ятисот моделей камер більш ніж вісімдесятьох різних виробників (повний список можна побачити на сайті компанії) незалежно від того, у який альянс вони входять – ONVIF або PSIA. Також є можливість використання відеокамер, що підключаються по USB. Крім того, до системи можна приєднати також аналогові камери за допомогою спеціального IP-декодера; XProject Go буде працювати також і з ними. Незважаючи на те що пробна версія продукту є безкоштовною, у ній підтримуються сучасні формати стиску відео: MPEG4 ASP, MxPEG, H.264. До убудованого «інтелектуальним» функціям можна віднести програмний детектор руху, що працює незалежно від камери.

Як сервер можна використовувати звичайний ПК із процесором 2,4 ГГц і обсягом оперативної пам'яті 1-2 ГБ. Якщо планується використовувати продукт не більше місяця, те можна навіть не реєструватися на сайті. Процедура обов'язкова тільки у випадку застосування ПЗ понад тридцять днів. Інтерфейс програми підтримує двадцять мов, у тому числі росіянин. Однак XProject Go працює тільки з різними варіантами ОС Windows (XP, Vista, 7). Для роботи з іншими операційними системами існують інші продукти. Одним з них є умовно безкоштовна версія ПЗ для централізованого відеоспостереження, що пропонує компанія BenSoftware. Її розробка зветься SecuritySpy (рисунок 2.2) і розроблена вона для комп'ютерів Macintosh. Діапазон підтримуваних ОС – від Mac OS X 10.4.11 до 10.7.

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

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

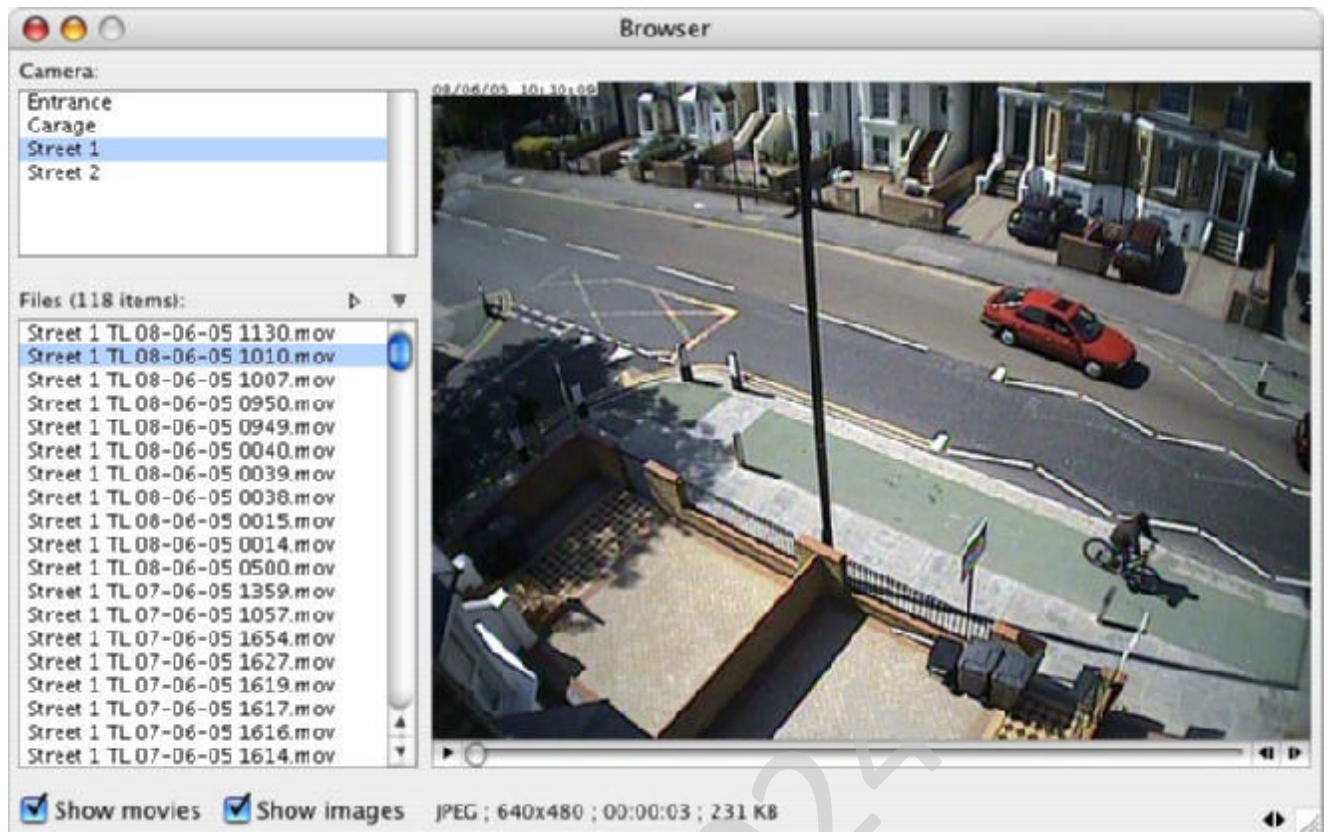


Рисунок 2.2 – Програма BenSoftware SecuritySpy призначена для комп'ютерів Macintosh

Система підтримує необмежене число відеокамер (аналогових, IP, USB), серверів і обсяг архіву. Використовуються кодеки MJPEG і MPEG4. Є убудований веб-сервер, функції детектування руху, оповіщення відповідальних осіб по e-mail, автоматичного завантаження даних на зазначений FTP-сервер. Цікава функція – буфер запису, у який автоматично зберігається відео, зняте за деяке, досить обмежене час, скажемо, за останні десять секунд. Це зроблено для того, щоб у випадку спрацьовування детектора руху можна було побачити не тільки тривожну подію, але й те, що йому передувало. В SecuritySpy є також можливості пошуку подій в архіві й автоматичній його оптимізації (яка зводиться головним чином до видалення застарілих даних). Підтримуються безліч камер, у тому числі роботизованих, таких виробників, як Axis, JVC, Panasonic, Pixord, D-Link і багатьох інших.

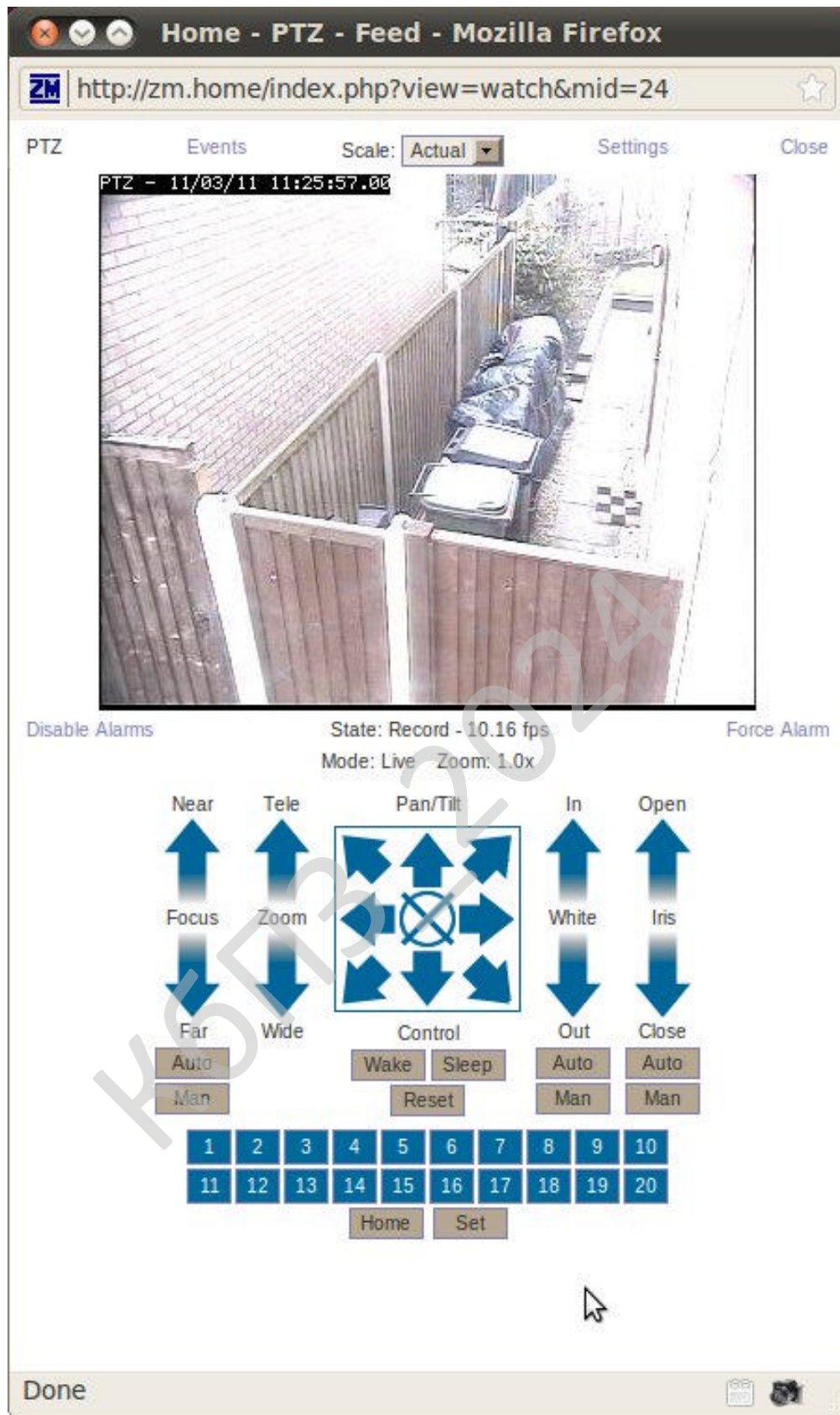


Рисунок 2.3 – Інтерфейс контролю над PTZ-камерою за допомогою ПЗ ZoneMinder

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

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

Axhon Soft дозволяє безкоштовно скачати зі свого сайту й надалі використовувати обмежену версію Smart IP (рисунок 2.4) – свого професійного програмного продукту для відеоспостереження. В урізаному варіанті ПЗ називається Smart Start.

При цьому в тестовій версії зберігаються всі функціональні можливості платного продукту. Обмеження стосуються кількості камер, що підключаються, (до шістнадцяти) і обсягу відеоархіву (не більше 1 ТБ). Також в Smart Start можна використовувати тільки один сервер, але разом з тим – необмежене число клієнтів. В іншому – це повнофункціональна версія.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

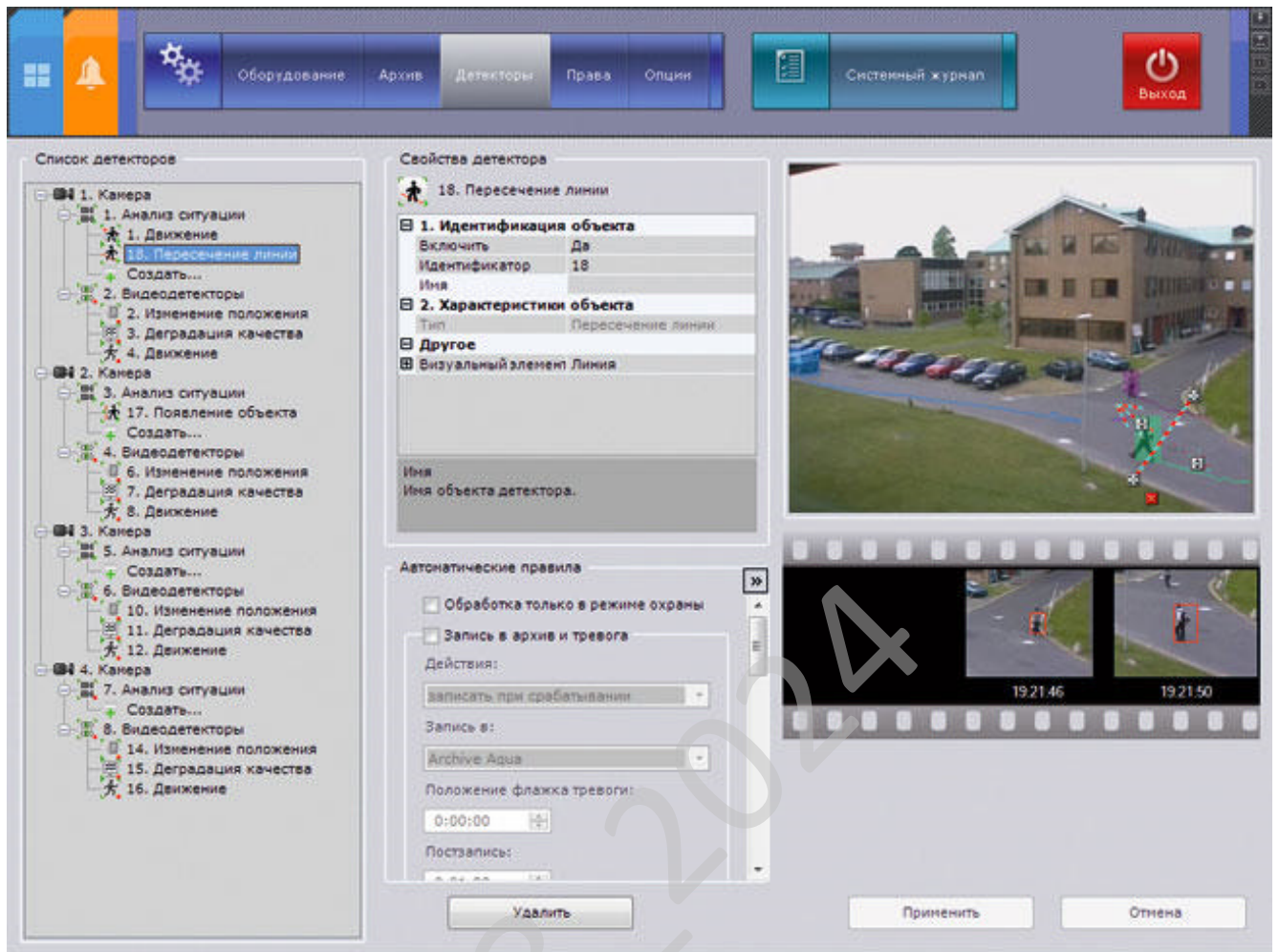


Рисунок 2.4 – Вікно оператора відеоспостереження Axxon Smart Start

Коротко розглянемо її можливості. Система працює на базі ОС Windows і здатна автоматично знаходити й розпізнавати підключене до мережі IP-устаткування. Є убудовані функції відеоаналітики, детектор руху й звуку, засобу оповіщення про тривожні події, автоматичний запис (у випадку інциденту) і т.д. Підтримуються основні формати стиску відео – MJPEG, MPEG-4, MxPEG, H.264. На сьогоднішній день система здатна працювати більш ніж із двома сотнями моделей камер різних виробників (яким на сайті компанії перераховане понад тридцять). Список постійно розширюється, і клієнти можуть доповнювати свої системи підтримкою нових моделей камер. Для цього досить установлювати оновлений набір драйверів (Drivers Pack) замість переустановки всієї системи.

Окремо варто сказати про систему пошуку в архіві. Незважаючи на те що Smart Start поширюється безкоштовно, завдяки фірмовим розробкам у відеоархіві можна здійснювати пошук за розміром, кольором, напрямком й швидкості руху об'єкта. Можна також шукати об'єкти, що рухаються, у заданій області, перетинання умовної лінії, появу або зникнення об'єктів і т.д. По суті, це єдиний програмний продукт в огляді, що надає такі можливості безкоштовно.

Обмежену версію свого ПЗ для відеоспостереження за назвою SecurOS Lite пропонує й інша російська компанія – Intelligent Security Systems (ISS). Безкоштовна версія підтримує до чотирьох відеокамер, детектор руху, формат відео avi. На відміну від інших безкоштовних продуктів, згаданих в огляді, SecurOS використовує інтерактивні карти будинків і багаторівневу систему прав доступу. Однак у той же час тут немає убудованих функцій відеоаналітики й розвиненого пошуку по архіву. Для одержання цих додаткових можливостей прийдеться купувати одну з комерційних версій продукту, які в порівнянні з розробками інших виробників виглядають не занадто презентабельно. Наприклад, прапорманська версія – SecurOS Enterprise – підтримує до 320 аналогових або відеокамер і до десяти серверів. Список виробників, із чийм устаткуванням SecurOS здатний працювати, містить не набагато більше двох десятків назв, що, у загальному, теж відносно небагато.

Компанія «Скайрос» пропонує безкоштовно скористатися ПЗ VideoNet Prime, що дозволяє підключати до одного сервера до шістнадцяти відеокамер. При цьому у внутрішній мережі може бути до чотирьох операторських робочих місць, а кількість що підключаються веб-клієнтів не обмежено. Відзначимо, що крім розповсюджених форматів стиску – MJPEG, MPEG4, H.264, система підтримує також спеціалізований алгоритм компресії DVPack2. Крім того, VideoNet Prime відрізняється широкими можливостями аналітики. Тут є різноманітні детектори – руху, напрямку, звуку, залишених/ зниклих предметів, перетинання умовної лінії, лічильник об'єктів і т.д. Згадане ПЗ орієнтовано на роботу з відеокамерами виробників, що складаються в альянсі ONVIF; при цьому

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

на своєму сайті в списку сумісного встаткування розроблювач указує біля тисячі моделей. По цьому показнику з VideoNet Prime може змагатися тільки система XProject компанії Milestone.

Як бачимо, універсальне й при цьому безкоштовне ПЗ для централізованого відеоспостереження існує, і одержати його не становить праці. При цьому можна підібрати систему, що володіє дуже широкими можливостями й до того ж під будь-яку популярну ОС. Можливості таких продуктів досить широкі, незважаючи на тестовий статус, і в ряді випадків невелика компанія може обійтися цим «безкоштовним сиром», не здобуваючи повну версію продукту. У той же час великий замовник цілком здатний оцінити достоїнства й недоліки основних функцій програмної розробки, перш ніж ухвалювати рішення щодо покупки комерційної версії ПЗ.

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

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

### **Delphi 10.4 Sydney**

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

#### Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватимуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

#### RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкістю. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

## **Істотне поліпшення Delphi Code Insight**

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

## **Delphi Custom Managed Records**

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

## **Єдине керування пам'яттю**

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

## **Розширена підтримка бібліотек C++**

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22



Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

### **Нові High DPI стилі й стилізація окремих VCL компонент**

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

### **Поліпшена кроссплатформеність**

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

### **Оновлений менеджер пакетів Getit**

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

## Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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 кібербезпеки 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;

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи кібербезпеки 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.

КБПЗ\_2024

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

У наш час установка системи відеоспостереження є обов'язковою й неодмінною умовою забезпечення високого рівня безпеки будь-якого об'єкта. У комплект устаткування при монтажі типової системи відеоспостереження входять наступні складові:

- пристрої для обробки відеосигналу;
- пристрої для запису й зберігання відеоінформації;
- що відображають пристрої.

За допомогою цифрових або аналогових систем відеоспостереження різної конфігурації забезпечується постійний і надійний відеоконтроль на об'єктах будь-якого призначення й масштабу.

Будь-яка система відеоспостереження містить:

- відеокамери, квадранти й мультиплекси – пристрою обробки відеосигналу;
- відеореєстратори – пристрої для запису й зберігання інформації;
- монітори – пристрої, що відображають.

#### Відеокамери і їхні види

Перед тим, як зробити установку системи відеоспостереження, проводиться попереднє визначення місця розташування відеокамер і майбутніх об'єктів спостереження. Залежно від призначення й умов експлуатація в системах відеоспостереження можуть використовуватися кольорові або чорно-білі відеокамери різних типів.

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

## **Модульні відеокамери**

Даний тип відеокамери являє собою друковану плату, на якій розміщена ПЗС-матриця. Модульні камери поставляються або з убудованим об'єктивом, або з роз'ємом під установку об'єктива.

Камери цього типу відрізняються мініатюрними розмірами (20x20 або 40x40 мм.), що уможлиблює їхню установку в компактний термокожух, корпус охоронного або пожежного датчика або інший підходящий предмет. Модульні відеокамери особливо затребувані при веденні схованого відеоспостереження.

## **Корпусні відеокамери**

Корпусний тип відеокамер являє собою камеру, розташовану в корпусі, у якому є місце для установки об'єктива. Таким чином, можливий підбор об'єктива з необхідними характеристиками, що підходять для експлуатації камери в конкретних умовах.

Зрозуміло, що можна підібрати об'єктиви, технічні характеристики яких значно краще, ніж в убудованих об'єктивів. Корпусні відеокамери можуть розміщатися в захисному термокожусі й установлюватися на вулиці для роботи в складних умовах експлуатації.

## **Вуличні відеокамери**

Якщо необхідно вести відеоспостереження в екстремальних умовах (пил, дощ, сніг, жара й т.п.) незамінні вуличні відеокамери. Можна придбати комплект для складання, що складає з відеокамери, об'єктива, термокожуху й кронштейна, а можна віддати перевагу готовій відеокамері.

Також у комплект із вуличними відеокамерами, як правило, входять обігрівачі, метою яких є виключення замерзання й утворення конденсату. Призначення термокожуху у вуличних відеокамерах полягає не тільки в запобіганні впливів навколишнього середовища, але й у захисті від механічних ушкоджень, у тому числі й від рук хуліганів.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

## **Відеокамери, обладнані ІЧ-підсвічуванням**

Такі камери доцільно застосовувати в приміщеннях при відсутності висвітлення, а також у системах вуличного спостереження. ІЧ-прожектор вбудовується в корпус відеокамери або є окремим пристроєм.

Можливо автоматичне включення й вимикання за допомогою реагуючого на освітленість фотодатчика, що дозволяє заощаджувати на витратах на електроенергію.

## **Купольні поворотні відеокамери**

Широке застосування в сфері відеоспостереження знаходять поворотні відеокамери, у яких одночасно функціонують відеокамера, трансфокатор і пристрій повороту. У порівнянні зі звичайною купольною відеокамерою, поворотна камера коштує дорожче, але при цьому відрізняється більшою функціональністю.

Поворотна відеокамера звичайно являє собою камеру високого розрішення для денного й нічного спостереження із двадцятиразовим трансфокатором. Висока вартість і широкий функціонал спричиняються застосування поворотних камер у професійних системах спостереження.

## **Відеореєстратори**

Відеореєстратор являє собою пристрій для відображення й запису, а також наступного зберігання вступника від камер відеоінформації. Сьогодні одержали поширення два типи відеореєстраторів:

- повністю закінчений апаратний пристрій;
- пристрій на базі персонального комп'ютера й плат.

Системам відеоспостереження, у яких використовуються реєстратори, що представляють закінчені пристрої, властиві висока надійність, ресурсоємність і стабільність роботи. До недоліків такої системи відеоспостереження можна віднести трохи обмежений набір функцій, а також невисоку швидкість запису.

У випадку монтажу системи відеоспостереження з використанням персонального комп'ютера гарантовані широкі можливості й висока

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

продуктивність, обумовлені технічними характеристиками, що постійно вдосконалюються, персональних комп'ютерів і розвитком програмного забезпечення для відеоспостереження. Комфорт при роботі оператора системи відеоспостереження забезпечується за рахунок звичної роботи за комп'ютером зі зручним інтерфейсом і безліччю потрібних налаштувань і функцій.

Якість запису. Запис інформації, що надходить від камер і мікрофонів, виробляється в цифровому виді із застосуванням форматів стиску на жорсткому диску. Сучасні формати стиску забезпечують максимальна якість записуваної відеоінформації.

Ємність архіву системи визначається й обмежується кількістю і ємністю встановлених у ній жорстких дисків. Можливе налаштування швидкості режиму запису для всієї системи й для окремих каналів. Існують наступні режими запису: безперервний режим, за розкладом, по команді оператора або по детектору руху й ін.

#### **Архів системи відеоспостереження**

Пошук відеоінформації в архіві здійснюється за часом і датою або по конкретній відеокамері. При цьому в момент перегляду архіву система продовжує поточний запис із камер. Можливий режим мультиекрану, при якому ведеться перегляд зображень із декількох камер, і експорт фрагментів запису на зовнішні носії.

#### **Охоронні датчики, детектори руху**

Режим, при якому запис починається за сигналом від датчика охоронної сигналізації або при спрацьовуванні детектора руху, забезпечує ощадливе використання пам'яті відеореєстратора, призначеної для запису, і надає можливість не зберігати зайву інформацію. Датчики охоронної сигналізації, спрацьовуюючи, сигналізують відеореєстраторові про подію, після чого здійснюється запис відео.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		30

## Проектування й установка відеоспостереження з роботою з мережі

Важливою функцією системи спостереження є робота з мережі, приміром, у локальній офісній мережі або інтернеті. Ця зручна функція дає можливість власникові об'єкта підключатися через інтернет до системи відеоспостереження, щоб переглядати ситуацію на об'єкті в режимі онлайн або архів прямо з будинку або будь-якого іншого місця. Техніко-експлуатаційні характеристики сучасного встаткування уможливають спроектувати й зробити монтаж систем відеоспостереження, що відрізняються високим рівнем надійності, якістю запису й відображення, а також широкою функціональністю.

### 3.2 Розробка структурної схеми

Розглянемо розроблену структурну схему роботи моєї системи – рисунок 3.1. Розглядати схему необхідно згори до низу. За допомогою розробленого бакалаврського ПЗ оператор з сервера системи відеоспостереження може спостерігати за під'їздом, входом до приміщень та прибудинковою територією багатоквартирного дому.

Система охоронного відеоспостереження житлового будинку з убудованими й прибудованими приміщеннями (далі – Об'єкт), призначена для контролю за станом охоронюваного об'єкта, для запису відеозображення на необхідний час, з можливістю її перегляду на будь-якому робочому місці. Ціль створення системи – захист приміщень Об'єкта від можливих вандалських дій, своєчасне реагування й припинення можливих терористичних і диверсійних дій, а також захист людей і майна від інших злочинних зазіхань.

Система відеоспостереження містить:

- Підсистему формування відеосигналів на Об'єкті, що складається зі стаціонарних вуличних і внутрішніх камер відеоспостереження.
- Підсистему відеоархівування, керування й контролю, що містить сервера, автоматизовані робочі місця – комп'ютери й відеомонітори.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– Підсистему локальної обчислювальної мережі безпеки, що поєднує АРМ операторів і сервера системи відеоспостереження.

– Підсистему електроживлення 220В/50Гц, що містить джерела безперебійного живлення (UPS) для гарантованого електроживлення встаткування системи телеспостереження протягом заданого часу.

– Програмно-апаратне забезпечення системи відеоспостереження.

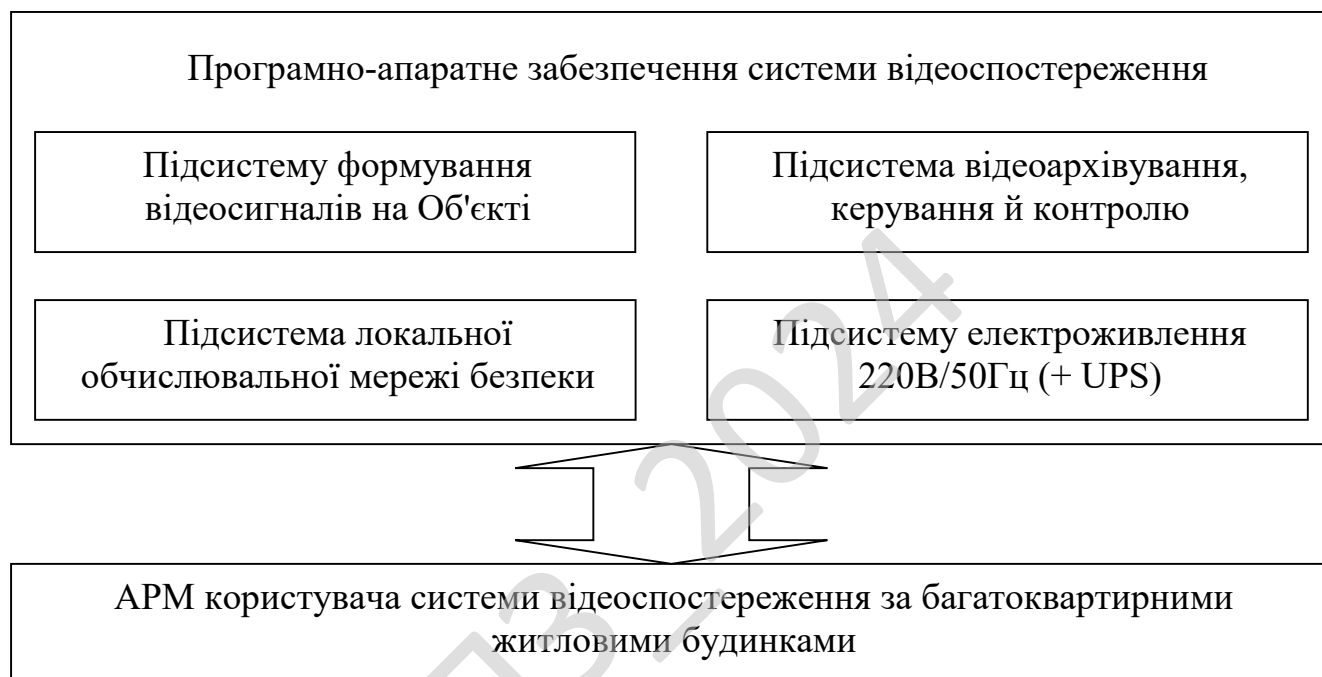


Рисунок 3.1 – Структурна схема роботи системи

Як правило всі сучасні житлові комплекси мають цілодобову охорону (консьержа). У допомогу йому, для забезпечення максимальної безпеки й спокою на території всього житлового комплексу ми пропонуємо свої сучасні технічні рішення.

Всі входи й в'їзди в житловий сектор, автопаркинг, під'їзди, ігрові площадки повинні обов'язково попадати під об'єктиви відеокамер. При недостатнім висвітленні для одержання якісного зображення встановлюються ІЧ підсвічування або ІЧ прожектор.

У даному технічному рішенні представлені відеокамери високого дозволу в термокожухах з кутами огляду до 90 градусів. Високий дозвіл дає можливість більш точно визначити особистість правопорушника. Широкий кут огляду дозволить охопити більшу територію житлового комплексу, зменшивши площу мертвих зон і кількість відеокамер, скоротивши тим самим витрати на систему в цілому. При необхідності на під'їзди, а також на сходові прольоти встановлюються вандалозахиснені відеокамери. Виключено доступ небажаних сторонніх осіб на територію двору. Упорядковано рух автотранспорту по території двору (підземному паркингу).

Відеореєстратор може бути встановлений у квартирі в старшого по під'їзду, а дублюючий монітор у приміщенні консьєржа. У ряді випадків все встаткування відеоспостереження під'їзду встановлюється в консьєржа. По будинковій локальній мережі або мережі Інтернет жителі будинку можуть підключатися до відеореєстратора й бачити дітей у дворі, машину на стоянці, що заходять у під'їзд людей. Це дозволяє контролювати ситуацію й почувати себе в безпеці, можна спокійно відпускати пограти у дворі дітей, не боятися, затримавшись на роботі вертатися додому.

У випадку виникнення події, завжди істи можливість підняти архів і переглянути його, виявивши винуватців. Відеоспостереження в під'їзді багатоквартирного будинку – рішення для людей, що хочуть бути спокійними за схоронність життя й здоров'я себе й своїх близьких. І звичайно ж свого майна й нервових кліток.

У цей час практично всі багатоквартирні будинки вже на етапі будівництва й уведення в експлуатацію комплектуються найпростішими системами контролю доступу – домофонами. Її можна модернізувати, установивши у квартирах відеодомофони, а на входах у під'їзди – домофони з камерами спостереження.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

### 3.3 Розробка функціональної схеми

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

Система відеоспостереження в житлових будинках вирішує наступні функціональні завдання:

- Візуальний контроль освітленості під'їздів і двірських територій у темний час доби.
- Візуальний контроль ліфтових кабін, з метою виявлення фактів вандалізму.
- Контроль якості й своєчасності збирання територій, вивозу сміття й т.п.
- Цілодобова візуальна оцінка обстановки в житловому секторі шляхом телевізійного спостереження за під'їздами й двірськими територіями, а також усередині будинку за ліфтами й сходовими площадками.
- Оперативне оповіщення служб охорони правопорядку й інших екстрених служб міста про виникнення або підозру на виникнення ситуацій, що загрожують життя й здоров'ю людей, схоронності їхнього майна, а також схоронності муніципального майна.
- Надання службам охорони правопорядку й інших зацікавлених служб міста архівної інформації телевізійного спостереження для відновлення ходу подій, підтримки проведення оперативно-слідчих заходів і т.п.
- Інтеграція систем відеоспостереження з локально-обчислювальними мережами, а також доступ до системи відеоспостереження з будь-якої точки миру, використовуючи підключення до Інтернет.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34



Рисунок 3.2 – Функціональна схема системи

Визначаємо ключові точки відеоспостереження житлового будинку з позиції людей які в ньому проживають:

- відеоспостереження стоянки автомобілів;
- відеоспостереження в під'їзді (вхід у під'їзди);
- відеоспостереження дитячого майданчика.

Тепер розглянемо докладніше кожну зону залежно від завдань системи відеоспостереження будинку.

### **Відеоспостереження стоянки автомобілів**

Загальний контроль за стоянкою й контроль що в'їжджає й виїжджає автотранспорту.

Необхідна камера вуличного виконання з повноцінним режимом день/ніч або ІЧ підсвічуванням. Бажана наявність у камері функції антивідблиск, для

запобігання засвітки камери автомобільними фарами. Як варіант на в'їзді на стоянку камера для фіксації автомобільних номерів, особливо зручно якщо в'їзд на стоянку обмежений за допомогою шлагбаума, або у випадку використання спеціального програмного забезпечення для відеоспостереження тоді спрощується облік що заїхали, що виїхали автомобілів, до речі стаття про відеоспостереження на транспорті, Тобто на стоянці розташовані як мінімум дві камери відеоспостереження одна для загального контролю відеоспостереження на стоянці, а друга для фіксації державних номерів що в'їжджають і виїжджають автомобілів.

### **Відеоспостереження дитячого майданчика**

Безпека якої просто неможливо обійти в побудові системи відеоспостереження житлового будинку.

Можливий варіант обійтися однією камерою, але високого дозволу розташувавши її так щоб проглядалася площадка цілком. Дозвіл 700 ТВЛ effio для аналогової камери а у випадку IP відеоспостереження мегапіксельна камера з дозволу й фокусна відстань об'єктива бажано дуже маленьке для широкого кута огляду напр. 2,5мм. 120GRAD.

Аналогова камера (NOVUS NVC-BDN3404H|IR). IP камера відеоспостереження (Mobotix Allround M24 L11) ну дуже широкий кут огляду 180 градусів і дозвіл 3МП.

### **Відеоспостереження в під'їзді (вхід у парадне)**

Особливо важлива насамперед фіксація осіб вхідних/вихідних і людей це камера внутрішня купольна, бажано достатня антивандальність такої камери для запобігання ушкоджень легкодоступного елемента системи відеоспостереження в будинку. Обов'язкова наявність wdr і компенсації задньої засвітки.

Крім камер відеоспостереження для організації системи відеоспостереження будинку знадобляться:

– Кабельна продукція кабель для відеоспостереження – краще використовувати якісний кабель "одесакабель" або європейських виробників.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Зверніть увагу на кількість кабелю для відеоспостереження, якщо купувати відразу бухту 305 м. то в продавця можна одержати знижку, у відмінності від покупки кабелю по метражу. Відносно кабелю для відеоспостереження використовувати UTP, коаксіальний або комбінований (коаксіальний + ШВВП) залежить у першу чергу від відстані наприклад: 10-70м. Коаксиал, без додаткового встаткування.

– По кручений парі UTP за допомогою приймально-передавачів пасивних або активних передавачів можливо передати відеосигнал до 1000м. Також ураховується ціна, UTP на порядок дешевше коаксіального кабелю для відеоспостереження. На більших відстанях навіть при обліку додаткового встаткування виходить істотна економія. Взагалі якщо відстані від камер відеоспостереження до відеореєстратора великі використовуйте спеціальний відеореєстратор TWIST уже оснащений спеціальним устаткуванням для передачі відеосигналу на більші відстані (до 1000м.), зручним меню, що вражають характеристиками, зроблений в Україні й з нормальним сервісом чого не можна сказати про багатьох неякісних реєстраторах невідомих виробників.

– Коннектори для відеоспостереження, також має сенс використовувати тільки якісні. Якісні коннектори повинні бути як мінімум з міді й мідного сплаву, місце з'єднання коннекторів убереже від впливів зовнішнього середовища монтажна коробка, бажано як мінімум місце контакту помістити в термоусадку.

– Без кабелю також можливо побудувати систему, ір відеоспостереження за просто, навіть у випадку, камер без wi-fi модуля, установкою додаткового мережного бездротового встаткування (wi-fi точка доступу для відеонагляду, wi-fi роутер)

– Блок живлення для системи відеоспостереження один або декілька, розраховується відповідно до споживаних потужностей камер, але з деяким запасом на втрату живлення в проводах або пікові перегони при включенні ІЧ підсвічування, наприклад якщо камери споживають 2,8 А, те вибираючи між блоком живлення 3А и 5 А варто зупинити вибір на 5А.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

– Реєстратор для системи відеоспостереження багатоквартирного будинку необхідний із частотою 25 кадрів в d1 25к/с як мінімум на каналах дитячого майданчика й вхід у парадне, там найбільша прохідність.

– А у випадку ІР відеоспостереження відповідально підійти до вибору ПЗ й модулів, що підключаються, наприклад модуль для фіксації номерів автотранспорту, або модуль визначення осіб – ідентифікація відвідувачів.

Також бажано вирішити питання з віддаленим доглядом для мешканців непогано настроїти доступ для віддаленого перегляду через хмарний сервіс для відеоспостереження.

Немаловажно при установці відеоспостереження будинку відразу вирішити питання про сервісне обслуговування цієї системи. Або це буде разовий візит фахівця у випадку виходу з ладу, або ж більше кращий варіант укласти договір сервісного обслуговування. У цьому випадку оплата буде щомісяця або щокварталу, але крім разового виїзду при поломці системи одержите профілактичні огляд, настроювання при необхідності і пріоритет в обслуговуванні в порівнянні з разовим виїздом на ремонт відеоспостереження.

Які переваги житлового будинку із системою безпеки й без такої, пояснювати по моєму не треба це насамперед безпека людей які там живуть.

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

### **3.4 Розробка діаграми процесів**

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3.

Після початку роботи розробленого ПЗ ми потрапляємо до головного вікна ПЗ, звідки можемо перейти до налаштування ПЗ, перегляду архіву відеоспостереження.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Також провести підключення камер системи відеоспостереження та через обробку відео трансляції перейти до виведення зображення з відео камер з можливістю застосування систем розпізнавання руху об'єктів, блоку оперативного оповіщення служб охорони правопорядку та провести санкціонування доступу. Далі провести запис руху об'єктів з конвертуванням відеопотоку до формату avi та записом файлу до архіву системи відеоспостереження.

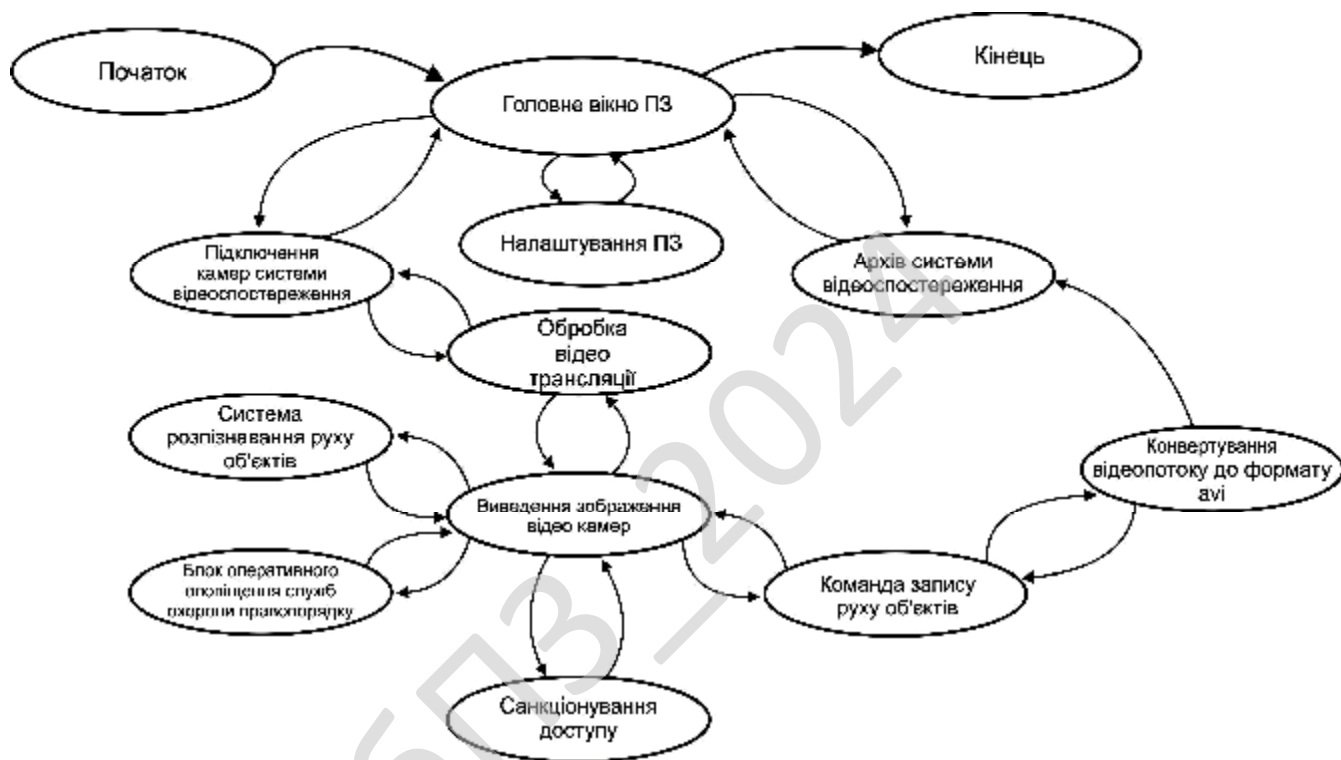


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

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

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків:

- Виведення головного вікна ПЗ.
- Ініціалізація системи відеоспостереження.
- Запит стоп-кадру.
- Запит створення відеороліку.
- Створення стоп-кадру з обраного потоку відеоспостереження.
- Обрання потоку відеоспостереження.
- Встановлення тривалості запису відео.
- Дія запису роліку з обраного відеопотоку.
- Є рух об'єктів у зоні бачення (запит).
- Виведення повідомлення.
- Створення та збереження стоп-кадру.
- Продовжити запис відео роліку (запит).
- Запит перегляду відеоархіву.
- Виведення списку та обрання файлу.
- Перегляд файлу.
- Запит змін відеоархіву.
- Підпрограма оновлення відеоархіву.
- Оновлення відеоархіву.
- Запит WM\_CLOSE.
- Підпрограма обробки запитів.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>

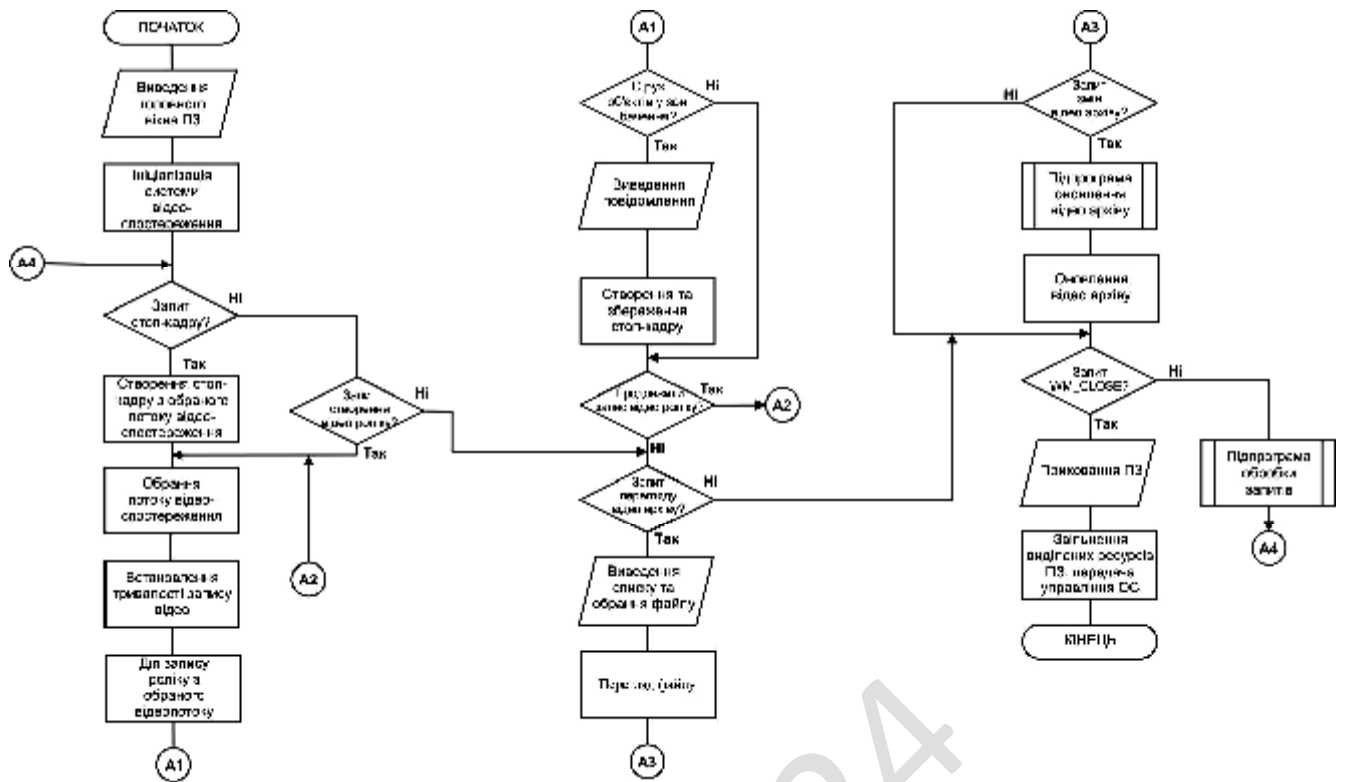


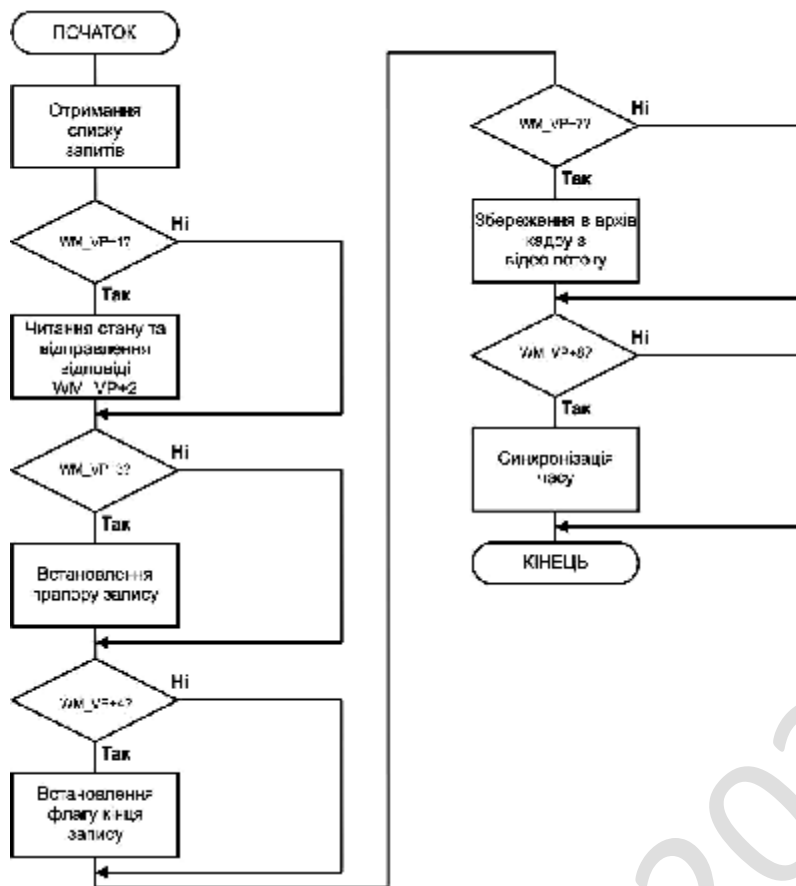
Рисунок 4.1 – Блок-схема основної програми

- Приховання ПЗ.
- Звільнення виділених ресурсів ПЗ, передача управління ОС.

На рисунку 4.2 наведено блок-схему підпрограми обробки запитів. Її робота складається з виконання наступних кроків:

- Отримання списку запитів.
- WM\_VP+1 (запит).
- Читання стану та відправлення відповіді WM\_VP+2.
- WM\_VP+3 (запит).
- Встановлення прапору запису.
- WM\_VP+4 (запит).
- Встановлення прапору кінця запису.
- WM\_VP+7 (запит).
- Збереження в архів кадру з відео потоку.
- WM\_VP+8 (запит).

Підпрограма обробки запитів



Підпрограма оновлення відео архіву

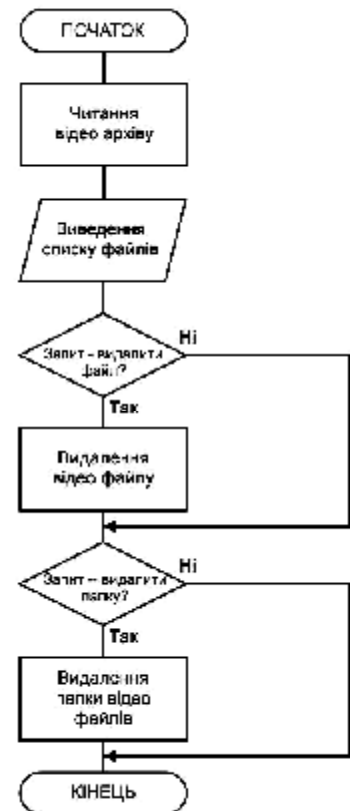


Рисунок 4.2 – Блок-схема підпрограми

– Синхронізація часу.

Також на рисунку 4.2 наведено блок-схему підпрограми оновлення відеоархіву. Її робота складається з виконання наступних кроків:

- Читання відеоархіву.
- Виведення списку файлів.
- Запит – видалити файл.
- Видалення відео файлу.
- Запит – видалити папку.
- Видалення папки відео файлів.

## Опис алгоритмів функціонування системи

При розробці ПЗ мені необхідно було точно знати розмір пікселя, область бачення, співвідношення сторін і кількість крапок на дюйм для різних діагоналей і роздрукочних дозволів моніторів.

Таблиця 4.1 – Розрахунки розміру пікселя

Діагональ	Роздрук. спроможн.	Позначення	Формат	Пікселів на дюйм, пкс/дюйм	Розмір пікселя, мм
15.0	1024x768	XGA	4:3	85.5	0.297
17.0	1280x1024	SXGA	5:4	96.2	0.264
17.0	1440x900	WXGA+	16:10	99.6	0.255
19.0	1280x1024	SXGA	5:4	86.3	0.294
19.0	1440x900	WXGA+	16:10	89.4	0.284
20.1	1400x1050	SXGA+	4:3	87.1	0.291
20.1	1680x1050	WSXGA+	16:10	98.4	0.258
20.1	1600x1200	UXGA	4:3	99.6	0.255
20.8	2048x1536	QXGA	4:3	122.7	0.207
21.0	1680x1050	WSXGA+	16:10	94.3	0.270
21.3	1600x1200	UXGA	4:3	94.0	0.270
22.0	1680x1050	WSXGA+	16:10	90.1	0.282
22.2	3840x2400	WQUXGA	16:10	204.0	0.1245
23.0	1920x1200	WUXGA	16:10	98.4	0.258
24.0	1920x1200	WUXGA	16:10	94.3	0.269
26.0	1920x1200	WUXGA	16:10	87.1	0.2865
27.0	1920x1200	WUXGA	16:10	83.9	0.303
30.0	2560x1600	WQXGA+	16:10	101.0	0.251

Реалізований стиск зображень у розробленому ПЗ. Було реалізовано стиск зображень за допомогою відомого та найпоширенішого алгоритму JPEG.

Розглянемо вихідний код.

```
unit JPEG_IO;
// Модуль стиску зображень
interface
// Інтерфейсна частина
{$WARN SYMBOL_PLATFORM OFF}
// компіляторні налаштування
{$WARNINGS OFF}
// Бібліотеки які використовуються
uses Windows, SysUtils, Classes, Graphics, IJL;

type
// створення типу виключення
EIJLException = class(Exception);

Type
// створення типу стиску
TIJLScale=(ijlFull, ijlHalf, ijlQuarter, ijlEighth);
...
// Процедура потокової конвертації до формату Jpeg
procedure SaveBmpToJpegStream(Bitmap:TBitmap;Stream: TMemoryStream; const
Quality: integer);
...
implementation
// Секція реалізації

procedure IJLCheck(Code:Integer);
// процедура роботи з помилками
begin
if Code<>IJL_OK then
raise EIJLException.Create(ijlErrorStr(Code));
end;
// Процедура потокової конвертації до формату Jpeg
procedure SaveBmpToJpegStream(Bitmap:TBitmap;Stream: TMemoryStream; const
Quality:integer);
var
// Об'ява змінних
jcprops:TJPEG_CORE_PROPERTIES;
DIB: TDIBSection;
```

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

```

Begin
// Початок процедури
    FillChar(jcprops, SizeOf (jcprops), 0);
// заповнення
    IJLCheck(ijlInit(@jcprops));
    Try
// блок спроби з можливим виключенням
        with jcprops do
            begin
                case bitmap.PixelFormat of
// перевірка формату
                    pf8bit:
                        begin
                            DIBChannels:=1;
                            DIBColor:=IJL_G;
                        end;
                    pf24bit:
                        begin
                            DIBChannels:=3;
                            DIBColor:=IJL_BGR;
                        end;
                    else raise EIJLException.Create('Cannot save bitmap as JPEG with
                        specified PixelFormat');
// виклик виключення
                end;
                FillChar(DIB, SizeOf(DIB), 0);
                Win32Check(GetObject(Bitmap.Handle, SizeOf(DIB),@DIB)<>0);
// перевірка
                DIBBytes:= PByte(DIB.dsBm.bmBits);
                DIBWidth := DIB.dsBm.bmWidth;
                DIBHeight :=-DIB.dsBm.bmHeight;
                DIBPadBytes:=((DIBWidth*DIBChannels+3) and -$04)-
                    (DIBWidth*DIBChannels);

                Stream.Size:= Abs(DIBHeight*DIBWidth*DIBChannels);
                JPGBytes := Stream.Memory;
                JPGSizeBytes := Stream.Size;
// Розмір
                JPGWidth := DIBWidth;
                JPGHeight:= -DIBHeight;
                JPGChannels:=3;
                JPGColor := IJL_YCBCR;

```

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

```

// палітра
        jquality := quality;
//Якість
        end;
        IJLCheck(ijlWrite(@jprops, IJL_JBUFF_WRITEWHOLEIMAGE));
        Stream.Size:=jprops.JPGSizeBytes;
// Розмір потоку
        finally
            IJLCheck(ijlFree(@jprops));
        end;
    end;
    ...
end.

```

Алгоритми машинної графіки можна розділити на два рівні: нижній і верхній. Група алгоритмів нижнього рівня призначена для реалізації графічних примітивів (математичних розрахунків, ліній, кіл, заповнень і т.п.).

Ці алгоритми або подібні їм відтворені в графічних бібліотеках мов високого рівня або реалізовані апаратний в графічних процесорах робочих станцій.

Серед алгоритмів нижнього рівня можна виділити наступні групи.

1. Найпростіші в значенні математичних методів і відмінні простотою реалізації, що використовуються. Як правило, такі алгоритми не є як найкращими за об'ємом виконуваних обчислень або необхідним ресурсам пам'яті.

2. Алгоритми з складнішими математичними передумовами, що відрізняють їх більшу ефективність.

3. До третьої групи слід віднести алгоритми, які можуть бути без великих ускладнень реалізовані апаратний (допускаючи розпаралелювання, рекурсивні, реалізовані в найпростіших командах). В цю групу можуть потрапити і алгоритми, представлені в перших двох групах.

4. Нарешті, до четвертої групи можна віднести алгоритми із спеціальним призначенням (наприклад, для усунення сходового ефекту, тремтіння кадру та інші).

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

До алгоритмів верхнього рівня відносяться в першу чергу алгоритми видалення невидимих ліній і поверхонь.

Задача видалення невидимих ліній і поверхонь продовжує залишатися центральною в машинній графіці..

До задачі видалення невидимих ліній і поверхонь примикає задача побудови (замальовування) півтонових (реалістичних) зображень, тобто обліку явищ, пов'язаних з кількістю і характером джерел світла, обліку властивостей поверхні тіла (прозорість, заломлення, віддзеркалення світла).

Проте, при цьому не слід забувати, що виведення об'єктів в алгоритмах верхнього рівня забезпечується примітивами, що реалізують алгоритми нижнього рівня, тому не можна ігнорувати проблему вибору і розробки ефективних алгоритмів нижнього рівня.

Для різних областей застосування машинної графіки на перший план можуть висуватися різні властивості алгоритмів.

Для наукової графіки велике значення має універсальність алгоритму, при цьому швидкодія може відходити на другий план. Для систем моделювання, відтворюючих об'єкти, що рухаються, швидкодія стає головним критерієм, оскільки потрібно генерувати зображення практично в реальному масштабі часу.

Особливості растрової графіки зв'язані з тим, що звичайні зображення, з якими стикається людина в своїй діяльності (креслення, графіки, карти, художні картини і т.п.), реалізовані на площині, що складається з нескінченного набору крапок.

Екран же растрового дисплея представляється матрицею дискретних елементів, що мають конкретні фізичні розміри. При цьому число їх істотно обмежено. Тому не можна провести точну лінію з однієї крапки в іншу, а можна виконати тільки апроксимацію цієї лінії з відображенням її на дискретній матриці (площини).

Таку площину також називають цілпа чисельними ґратами, растровою площиною або растром. Ці ґрати представляються квадратною сіткою з кроком 1.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Відображення будь-якого об'єкту на цільна чисельні грати називається розкладанням його в растр або просто растровим уявленням.

Розробка алгоритму взаємодії з відео камерами. При складанні блок-схем програмного забезпечення і напрацювання алгоритмів взаємодії з відеопотоком через камеру я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою розглянемо частину написаного вихідного коду для захвату декількох потоків відео інформації та їх обробки.

```
//Підключення бібліотек
#include <vcl.h>
#include <vfw.h>
#include <Clipbrd.hpp>
#pragma hdrstop
//-----
__fastcall TForm1::TForm1(TComponent* Owner): TForm(Owner)
{
    pCB = Clipboard ();
    Buffer = new Graphics::TBitmap;//Графічний буфер
}
//-----
//Підключаємо першу камеру
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    // Створення потоку
    hWndCLeft = capCreateCaptureWindow ("", WS_CHILD, this->Left, this->Top,
        this->Width, this->Height, this->Handle, 11011);

    // Підключення камери
    capDriverConnect (hWndCLeft, 0);
    //Встановлення параметрів таймера для оновлення кадрів
    Timer1->Interval = cseTime->Value;
    //Включення таймера
    Timer1->Enabled = true;
    Canvas->Pen->Width = 4;
    Canvas->Pen->Color = clRed;

    //Заголовок
    Caption = IntToStr (Buffer->Width) + ":" + IntToStr (Buffer->Height);}
//-----
//Підключаємо другу камеру
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    // Створення потоку
    hWndCRight = capCreateCaptureWindow ("", WS_CHILD, this->Left, this->Top,
```

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

        this->Width, this->Height, this->Handle, 11011);

// Підключення камери
capDriverConnect (hWndCRight, 0);
}
//Обробка виклику події таймера
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    capGrabFrame (hWndCLeft);
    if (capEditCopy (hWndCLeft)) {
Buffer->LoadFromClipboardFormat (CF_BITMAP,
pCB->GetAsHandle (CF_BITMAP), 0);
        int Hmin;
        int x = 10, y = 10;
        int z=0;
        for (int j = 0; j < Buffer->Height; j++) {
for (int i = 0; i < Buffer->Width; i++) {
if (i == 0 && j == 0) Hmin = (int) Buffer->Canvas->Pixels[j][i];
else if (Hmin < (int) Buffer->Canvas->Pixels[j][i]) {
                Hmin = (int) Buffer->Canvas->Pixels[j][i];
                x = j;
                y = i;
                z++;
            }
        }
    }
Caption = IntToStr (Buffer->Width) + " : " + IntToStr (Buffer->Height)+ " - " +
        IntToStr (x) + " : " + IntToStr (y);
    Buffer->Canvas->Pen->Color=0x000000ff;
    Buffer->Canvas->Pen->Style=psSolid;
    Buffer->Canvas->Ellipse(10 + x, 10 + y, x + 20, y + 20);
    Canvas->Draw (10, 10, Buffer);
}
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
//Ручне відключення від відео потоку 1
    capDriverDisconnect (hWndCLeft);
//Ручне відключення від відео потоку 2
    capDriverDisconnect (hWndCRight);}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)

```

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>49</b>

```

{
//Автоматичне відключення від відео потоку 1
    capDriverDisconnect (hWndCLeft);
//Автоматичне відключення від відео потоку 2
    capDriverDisconnect (hWndCRight);
}
//-----

```

Розрахунок синхронізації потоків відео інформації з web камер. Будь-яка вертикальна синхронізація, що виявлена в межах 64 ряду VWINDOW, скине лічильники, що керують сигналом VACTIVE і місцем розташування наступної очікуваної вертикальної синхронізації.

Повне блокування (налаштування якого  $\pm 1$  такт) досягнуті 12 рядків коли фактичний горизонтальний імпульс придбаний. Це відбувається незалежно від вертикального циклу.

Коли вертикальне блокування досягнуто, 64 ряд VWINDOW відкриє 32 ряду до наступного очікуваної вертикальної синхронізації. Якщо вертикальна синхронізація виявлена вперше, ніж VWINDOW відкривається, це буде ігноруватися. VRESET буде виводитися в очікуваній вертикальній синхронізації місці розташування, чи був вертикальна синхронізація виявлений чи ні.

Якщо ніяка вертикальна синхронізації не виявлена в межах VWINDOW, вертикальна синхронізації, виявлена до наступного VWINDOW буде прийнятий як законний, і вертикальний вибір годині буде скинутий, щоб синхронізувати це до цієї вертикальної синхронізації. VRESET буде виводитися і лічильники, що керують сигналом VACTIVE, місце розташування наступного очікуваного вертикальної синхронізації, і VWINDOW буде скинуто.

Тому, що лічильник VWINDOW скинутий, VWINDOW залишиться відкритим для 32 рядків після вертикальної синхронізації.

Будь-яка додаткова вертикальна виявлена синхронізації, у тієї годину як VWINDOW відкритий, також скине лічильники, що керують сигналом VACTIVE і місцем розташування наступного очікуваного вертикальної синхронізації.

Припущення, що рівень DC є тим самим, вертикальне блокування, буде досягнуте в 1-2 полях. Зсуви DC між двома сигналами через незафіксоване відео

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

можуть розширювати(продовжувати) вертикальне блокування (чим більший зсув DC, тим довше блокування година).

Для швидкого вертикального блокування рекомендується, щоб усі введення були DC, відновлена до загального(звичайному) рівню DC до переключення. Це може бути легко виконано, просто використовуючи діод, щоб фіксувати, щоб заснувати.

Горизонтальне блокування здійснене як цифровий Phase Lock Loop (PLL – Цикл Блокування Стадії), без скидання.

Якщо горизонтальна синхронізація виявлена, і HWINDOW відкритий, HWINDOW закритий, і сигнал помилки з генерований заснованим на розходженні між виявленим горизонтальної синхронізації місцем розташування і його очікуваним місцем розташування.

Цей сигнал помилки поданий потім, щоб коректувати наступне очікуване горизонтальної синхронізації місце розташування. HWINDOW відкриє 12 тактів до наступного очікуваної горизонтальної синхронізації, і не буде закривати поки горизонтальна синхронізація не буде виявлена. HRESET буде виводитися в кожнім очікуваному горизонтальній синхронізації місці розташування.

#### **4.2 Захист розробленого програмного забезпечення**

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На кожному етапі половина об'єднується за допомогою функції  $F[1]$  з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N-го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

КБПЗ – 2024

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню: Архівні дані; Налаштування; Довідка;
- Блоку відео потоків з функціональними кнопками: Додати відеосигнал; Видалити відеосигнал; Редагувати відеосигнал; Оповіщення служби охорони правопорядку; Надати доступ;
- Блоку функцій програми: Моніторинг; Налаштування; Функції; Налаштування запиту до служби охорони; Архів відео; Стан.

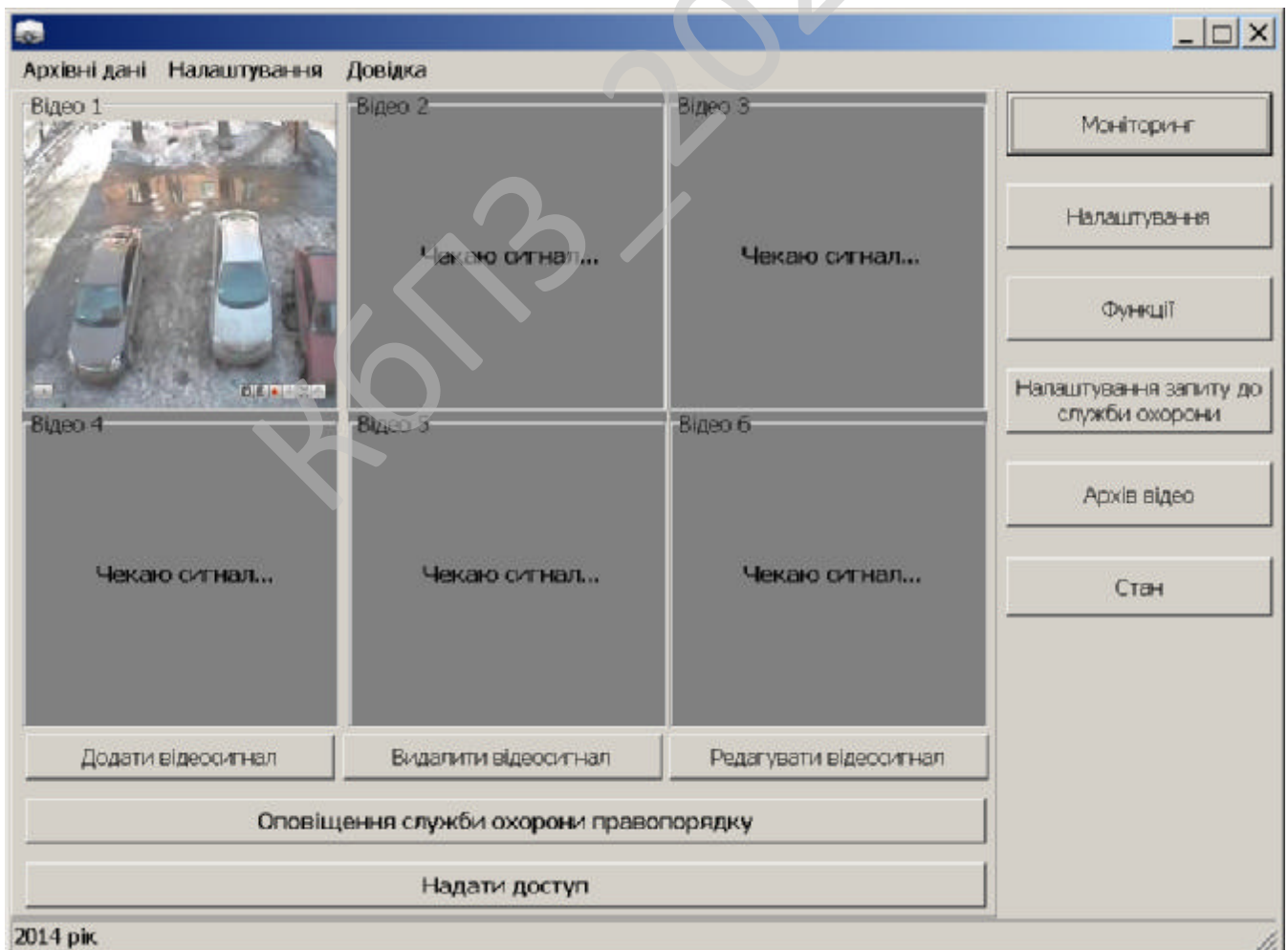


Рисунок 5.1 – Головне вікно програми

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>53</b>

На рисунку 5.2 зображено форму авторського права. Обрано ліцензію на програмне забезпечення – proprietary software тобто пропріетарне програмне забезпечення. Це ПЗ, на яке зберігаються як немайнові, так і майнові авторські права.

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

Методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття вихідного коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію (глава 4.2). Таке ПЗ називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

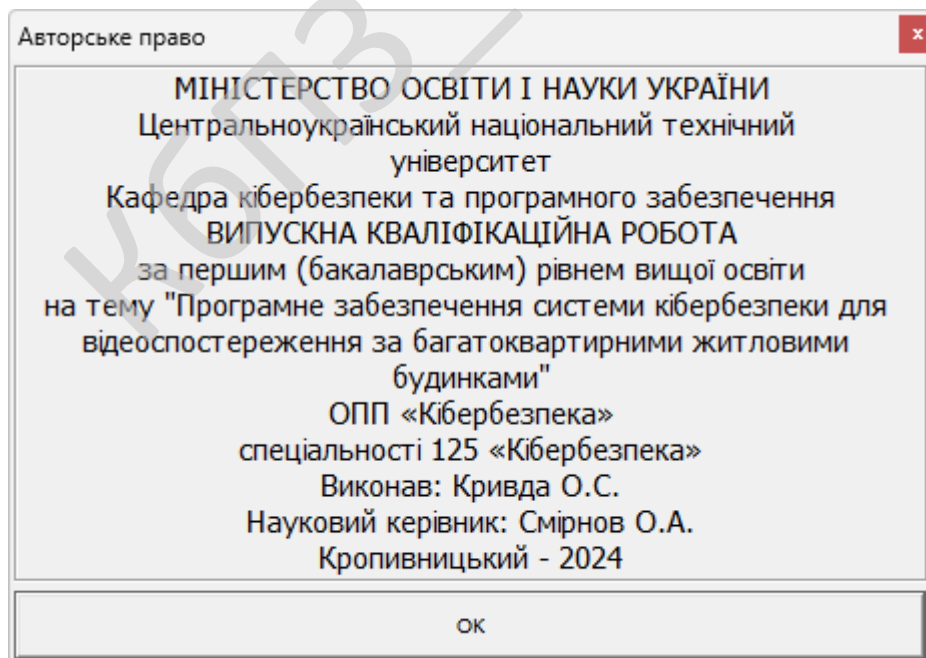


Рисунок 5.2 – Довідка розробника

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем для відеоспостереження за багатоквартирними житловими будинками.

– Досліджена система для відеоспостереження за багатоквартирними житловими будинками.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для відеоспостереження за багатоквартирними житловими будинками.

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

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

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для відеоспостереження за багатоквартирними житловими

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

будинками. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що постачається із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи кібербезпеки й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм FEAL.

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.

КБПЗ-2024

					ВКРБ-125.24.0011.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PeterShirley, SteveMarschner. Fundamentals of Computer Graphics. 2009
2. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
3. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
4. 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.
5. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. - Д.: НГУ, 2016. - 187 с.
6. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
7. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
8. 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.
9. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
10. 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.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

11. 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,
12. 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.
13. 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>
14. 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.
15. 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.
16. 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.
17. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

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

37. 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.

38. 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.

39. 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.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

40. 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.

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

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

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

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

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

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

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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

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

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

50. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

51. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції “Проблеми та перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

					<b>ВКРБ-125.24.0011.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>63</b>

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 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					<b>ВКРБ-125.24.0011.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Кривда О.С.				Програмне забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками	Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.					Б	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КБ-20			
Затв.	Смірнов О.А.							

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 135-02 від 01.04.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) 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;

					ВКРБ-125.24.0011.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи кібербезпеки з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи кібербезпеки для відеоспостереження за багатоквартирними житловими будинками;
- 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>ВКРБ-125.24.0011.00.00.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 Delphi 10.4.

					<b>ВКРБ-125.24.0011.00.00.ТЗ</b>	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 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 63 аркуші.

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					<b>ВКРБ-125.24.0011.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 7.06.2024 р.

					ВКРБ-125.24.0011.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

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

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
першим (бакалаврським) рівнем вищої освіти  
\_\_\_\_\_ Смірнов О.А.

*Програмне забезпечення системи кібербезпеки для відеоспостереження за  
багатоквартирними житловими будинками*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 37

Літера: РП

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - зберігання відео VIDEO\_GRAPH.pas**

```
// Copyrights @ Кривда Олексій Сергійович, КВ-20, ЦНТУ КБПЗ
unit VIDEO_GRAPH;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур
  TForm4 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Button1: TButton;
  private
    { Private секція }
  public
    { Public секція }
  end;

var // Об'ява змінних
  Form4: TForm4;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - налаштування SET\_DATA.pas**

```
// Copyrights © Кривда Олексій Сергійович, КБ-20, ЦНТУ КБПЗ
unit SET_DATA;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9;

type // Об'ява власних типів та структур

  TForm2 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Button1: TButton;
    VideoSourceFilter: TFilter;
    DataSourceFilter: TFilter;
    DataFormats: TListBox;
    Label4: TLabel;
    VideoFormats: TListBox;
    Label3: TLabel;
    DataCapFilters: TListBox;
    Label2: TLabel;
    VideoCapFilters: TListBox;
    Label1: TLabel;
    InputLines: TComboBox;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure VideoCapFiltersClick(Sender: TObject);
    procedure DataCapFiltersClick(Sender: TObject);
    procedure VideoFormatsClick(Sender: TObject);
    procedure DataFormatsClick(Sender: TObject);
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  Form2: TForm2;

Implementation // Реалізація

uses Unit1, LOG; // Які бібліотеки використовуємо у модулі

{$R *.dfm} // Підключення файлу ресурсів

procedure TForm2.Button1Click(Sender: TObject);
begin
  Form1.show;
  Form2.hide;
end;

procedure TForm2.FormShow(Sender: TObject);

begin
  Form1.ShowCAPdevise;

end;

procedure TForm2.VideoCapFiltersClick(Sender: TObject);
var
  PinList: TPinList;
  i: integer;
```

```

begin

    CapEnum.SelectGUIDCategory(CLSID_VideoInputDeviceCategory);
    if VideoCapFilters.ItemIndex <> -1 then
    begin

        VideoSourceFilter.BaseFilter.Moniker :=
            CapEnum.GetMoniker(VideoCapFilters.ItemIndex);

        VideoSourceFilter.FilterGraph := Form1.CaptureGraph;
        Form1.CaptureGraph.Active := true;

        PinList := TPinList.Create(VideoSourceFilter as IBaseFilter);
        VideoFormats.Clear;

        VideoMediaTypes.Assign(PinList.First);
        for i := 0 to VideoMediaTypes.Count - 1 do
            VideoFormats.Items.Add(VideoMediaTypes.MediaDescription[i]);

        Form1.CaptureGraph.Active := false;
        PinList.Free;

        Form1.StartButton.Enabled := true;
        Form3.Memo1.Lines.Add('Налаштовано'+DataCapFilters.Items.Strings[VideoCapFi
            lters.ItemIndex]);
    end;
end;

procedure TForm2.DataCapFiltersClick(Sender: TObject);
var
    PinList: TPinList;
    g,i, LineIndex: integer;
    ABool: LongBool;
begin

    CapEnum.SelectGUIDCategory(CLSID_DataInputDeviceCategory);
    if DataCapFilters.ItemIndex <> -1 then

    begin
        DataSourceFilter.BaseFilter.Moniker :=
            CapEnum.GetMoniker(DataCapFilters.ItemIndex);

        DataSourceFilter.FilterGraph := Form1.CaptureGraph;
        Form1.CaptureGraph.Active := true;

        PinList := TPinList.Create(DataSourceFilter as IBaseFilter);
        DataFormats.Clear;
        i := 0;
        while i < PinList.Count do
            if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
            begin
                DataMediaTypes.Assign(PinList.Items[i]);
                PinList.Delete(i);
            end else inc(i);

        for i := 0 to DataMediaTypes.Count - 1 do
            begin
                DataFormats.Items.Add(DataMediaTypes.MediaDescription[i]);
            end;

        Form1.CaptureGraph.Active := false;
        InputLines.Clear;
        LineIndex := -1;
        for i := 0 to PinList.Count - 1 do
            begin
                InputLines.Items.Add(PinList.PinInfo[i].achName);
                with (PinList.Items[i] as IAMDataInputMixer) do get_Enable(ABool);
            end;
        end;
    end;
end;

```

```
    if ABool then LineIndex := i;
end;

InputLines.ItemIndex := LineIndex;
PinList.Free;
Form1.StartButton.Enabled := true;
g:=DataCapFilters.ItemIndex;

Form3.Memo1.lines.add(' Налаштовано '+DataCapFilters.Items.Strings[g]);

end;
end;

procedure TForm2.VideoFormatsClick(Sender: TObject);
begin
    if VideoCapFilters.ItemIndex <> -1 then
        begin
            Form3.Memo1.lines.add('Відео'+VideoCapFilters.Items.Strings[VideoCapFilters.Item
                Index]);
        end;
    end;
end;

procedure TForm2.DataFormatsClick(Sender: TObject);
begin
    if DataFormats.ItemIndex <> -1 then
        begin
            Form3.Memo1.lines.add('Відео'+DataFormats.Items.Strings[DataFormats.ItemIndex]);
        end;
    end;
end;

end.
```

КБПЗ\_2024

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - головна форма програми Unit.pas**

```
// Copyrights © Кривда Олексій Сергійович, КВ-20, ЦНТУ КБПЗ unit Unit1;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9,
  jpeg, Buttons;

type // Об'ява власних типів та структур
  TForm1 = class(TForm) // Оголошення класу
    // Компоненти
    Panel1: TPanel;
    StatusBar1: TStatusBar;
    MainMenu1: TMainMenu;
    N1231: TMenuItem;
    Panel2: TPanel;
    Panel3: TPanel;
    VideoWindow: TVideoWindow;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Button7: TButton;
    Button8: TButton;
    Button9: TButton;
    Button10: TButton;
    CaptureGraph: TFilterGraph;
    SaveDialog: TSaveDialog;
    StartButton: TBitBtn;
    Image1: TImage;
    Timer1: TTimer;
    StopButton: TBitBtn;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    N15: TMenuItem;
    N16: TMenuItem;
    N17: TMenuItem;
    N18: TMenuItem;
    N19: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
    N22: TMenuItem;
    N23: TMenuItem;
    N24: TMenuItem;
    N25: TMenuItem;
    N26: TMenuItem;
    N27: TMenuItem;
    N28: TMenuItem;
    Timer2: TTimer;
    Button11: TButton;
```

```

Label1: TLabel;
SampleGrabber: TSampleGrabber;
Label2: TLabel;
SaveDialog1: TSaveDialog;
BitBtn1: TBitBtn;
Image: TImage;
// Реалізація подій
procedure Button9Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure StartButtonClick(Sender: TObject);
procedure StopButtonClick(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure N16Click(Sender: TObject);
procedure N18Click(Sender: TObject);
procedure N20Click(Sender: TObject);
procedure N22Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure N26Click(Sender: TObject);
procedure N28Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SampleGrabberBuffer(sender: TObject; SampleTime: Double;
  pBuffer: Pointer; BufferLen: Integer);
procedure Button11Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
  { Private секція }
public //зона бачення змінних та типів даних
  procedure ShowCAPdevisе;
  procedure SETKAS;
end;

var // Об'ява змінних
  Form1: TForm1;
  SETKA: integer;
  CapEnum: TSysDevEnum;
  VideoMediaTypes, DataMediaTypes: TEnumMediaType;
  Ok_normal: boolean;
  AR: array of array of tcolor;
  GO: boolean;

Implementation // Реалізація

uses About, LOG, SET_DATA; // Які бібліотеки використовуємо у модулі

{$R *.dfm} // Підключення файлу ресурсів

// Підпрограма виклику форми аворського права
procedure TForm1.Button9Click(Sender: TObject);
begin
  Form1.hide;
  Form3.Memo1.Lines.Add('Вызов просмотра о программе');
  AboutBox.show;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin

```

```

        MessageDlg('Beta version',mtInformation,[mbOK],0);
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    Form1.hide;
    Form3.show;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
    S:string;
begin
    InputQuery('Внимание','Введите путь для сохранения видео',S);
    CapFile:=S;
    Form1.hide;
    Form2.show;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    Form1.hide;
    Form2.show;
end;

procedure TForm1.Button10Click(Sender: TObject);
begin
    Form1.Close;
end;

procedure TForm1.ShowCAPdevise;
var
    i:integer;
begin
    Form2.VideoCapFilters.Items.Clear;
    Form2.DataCapFilters.Items.Clear;
    CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
        Form2.VideoCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);
    CapEnum.SelectGUIDCategory(CLSID_DataInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
        Form2.DataCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);
    VideoMediaTypes := TEnumMediaType.Create;
    DataMediaTypes := TEnumMediaType.Create;
end;
// Підпрограма виклику інтервального спрацювання
procedure TForm1.Timer1Timer(Sender: TObject);
var
    position: int64;
    Hour, Min, Sec, MSec: Word;
Const // Константи
    MiliSecInOneDay = 86400000;
begin
    if CaptureGraph.Active then
        begin
            with CaptureGraph as IMediaSeeking do
                GetCurrentPosition(position);
                DecodeTime(position div 10000 / MiliSecInOneDay, Hour, Min, Sec, MSec);
                StatusBar1.panels[1].Text := Format('%d:%d:%d:%d',[Hour, Min, Sec, MSec]);
            end;
        end;
end;

procedure TForm1.StartButtonClick(Sender: TObject);
var
    multiplexer: IBaseFilter;
    Writer: IFileSinkFilter;
    PinList: TPinList;
    i: integer;
begin

```

```

// Активувати фільтр
CaptureGraph.Active := true;

// налаштувати дані
if Form2.DataSourceFilter.FilterGraph <> nil then
begin
  PinList := TPinList.Create(Form2.DataSourceFilter as IBaseFilter);
  i := 0;
  while i < PinList.Count do
    if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
      begin
        if Form2.DataFormats.ItemIndex <> -1 then
          with (PinList.Items[i] as IAMStreamConfig) do
            SetFormat(DataMediaTypes.Items[Form2.DataFormats.ItemIndex].AMMediaType^);
            PinList.Delete(i);
            end else inc(i);
          if Form2.InputLines.ItemIndex <> -1 then
            with (PinList.Items[Form2.InputLines.ItemIndex] as IAMDataInputMixer) do
              put_Enable(true);
            PinList.Free;
          end;

// налаштування відео
if Form2.VideoSourceFilter.FilterGraph <> nil then
begin
  PinList := TPinList.Create(Form2.VideoSourceFilter as IBaseFilter);
  if Form2.VideoFormats.ItemIndex <> -1 then
    with (PinList.First as IAMStreamConfig) do
      SetFormat(VideoMediaTypes.Items[Form2.VideoFormats.ItemIndex].AMMediaType^);
      PinList.Free;
end;
// початок роботи
with CaptureGraph as IcaptureGraphBuilder2 do
begin
  // встановлення вихідного і'мя файлу
  SetOutputFileName(MEDIASUBTYPE_Avi, PWideChar(CapFile), multiplexer, Writer);
  // підключення
  if Form2.VideoSourceFilter.BaseFilter.DataLength > 0 then
    RenderStream(@PIN_CATEGORY_PREVIEW, nil, Form2.VideoSourceFilter as IBaseFilter,
      nil, VideoWindow as IBaseFilter);
    if Form2.VideoSourceFilter.FilterGraph <> nil then
      RenderStream(@PIN_CATEGORY_CAPTURE, nil, Form2.VideoSourceFilter as IBaseFilter,
        nil, multiplexer as IBaseFilter);

    if Form2.DataSourceFilter.FilterGraph <> nil then
      begin
        RenderStream(nil, nil, Form2.DataSourceFilter as IBaseFilter,
          nil, multiplexer as IBaseFilter);
      end;
end;

// вмикання захвату відеопотоку
CaptureGraph.Play;
StopButton.Enabled := true;
StartButton.Enabled := false;
Form2.DataFormats.Enabled := false;
Form2.DataCapFilters.Enabled := false;
Form2.VideoFormats.Enabled := false;
Form2.VideoCapFilters.Enabled := false;

Go:=false;
Timer1.Enabled := true;
VideoWindow.Visible:=false;
end;

// Підпрограма відключення захвату відеопотоку

```

```
procedure TForm1.StopButtonClick(Sender: TObject);
begin
    Timer1.Enabled := false;
    StopButton.Enabled := false;
    StartButton.Enabled := true;
    CaptureGraph.Stop;
    CaptureGraph.Active := False;
    Form2.DataFormats.Enabled := true;
    Form2.DataCapFilters.Enabled := true;
    Form2.VideoFormats.Enabled := true;
    Form2.VideoCapFilters.Enabled := true;
end;

procedure TForm1.N8Click(Sender: TObject);
begin
    Form1.Close;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
    Form1.hide;
    AboutBox.show;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
    MessageDlg('Beta version', mtInformation, [mbOK], 0);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    Form1.hide; // Приховати форму 1
    Form2.show; // Показати форму 2
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
    Form1.hide;
    Form2.show;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
    if Button7.Caption='B' then
    begin
        Go:=false;
        Timer2.Enabled:=true;
    end
    else
    begin
        Timer2.Enabled:=false;
    end;
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
    Form1.hide; // Приховати форму 1
    Form2.show; // Показати форму 2
end;

procedure TForm1.N16Click(Sender: TObject);
begin
    Form1.hide;
    Form2.show;
end;

procedure TForm1.N18Click(Sender: TObject);
```

```

begin
  Form1.hide;
  Form2.show;

end;

procedure TForm1.N20Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;

end;

procedure TForm1.N22Click(Sender: TObject);
begin
  Form1.hide;
  Form3.show;
end;

procedure TForm1.N24Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;

procedure TForm1.N26Click(Sender: TObject);
begin
  Form1.hide;
  Form2.show;
end;

procedure TForm1.N28Click(Sender: TObject);
begin
  Form1.hide; // Приховати форму 1
  Form2.show; // Показати форму 2
end;

procedure TForm1.N9Click(Sender: TObject);
begin
  Application.Minimize; // Приховання вікна ПЗ
end;

procedure TForm1.N10Click(Sender: TObject);
begin
  if Ok_normal then
  begin
    MessageDlg('Ok', mtInformation, [mbOK], 0);
  end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
  X, Y: integer;
begin
  Ok_normal:=true;
  SETKA:=50;
  VideoWindow.Canvas.Pen.Color:=clLime;
  VideoWindow.Canvas.Pen.Width:=4;
  BL:=1;

  for x :=1 to 320 do
  begin
    if (x mod SETKA)=0 then
    begin
      inc(BL);
    end;
  end;

  for y :=1 to 240 do

```

```

begin
    if (y mod SETKA)=0 then
    begin
        inc(BL);
    end;
    end;
SetLength(Ar, 320, 240);
end;

procedure TForm1.SampleGrabberBuffer(sender: TObject; SampleTime: Double;
pBuffer: Pointer; BufferLen: Integer);
begin
    Image.Canvas.Lock;
    try
        SampleGrabber.GetBitmap(Image.Picture.Bitmap, pBuffer, BufferLen);
        SETKAS;
    finally
        Image.Canvas.Unlock;
    end;
    SampleGrabber.GetBitmap(Image2.Picture.Bitmap, pBuffer, BufferLen);
    Image2.Canvas.Rectangle(1, 1, 400, 400);
end;

// Підпрограма захвату зображення з камери
procedure TForm1.Button11Click(Sender: TObject);
begin
    if SaveDialog1.Execute then
    begin
        if SaveDialog1.FileName<>' ' then
        begin
            SampleGrabber.GetBitmap(Image.Picture.Bitmap);
            Image.Picture.SaveToFile(SaveDialog1.FileName);
        end;
    end;
end;

// Підпрограма аналізу зображення накладенням сітки
procedure TForm1.SETKAS;
var
    x, y: integer;
    G: tcolor;
begin
    //Опис як обробляються координати
    //X VideoWindow.ClientWidth
    //Y VideoWindow.ClientHeight
    VideoWindow.Canvas.Pixels[]
    Labell1.Caption:=ColorToString(Image.Canvas.Pixels[20,20]);
    //-----
    if go and ((Image.Picture.Width<>0) and (Image.Picture.Height<>0))then
    begin
        for x :=1 to Image.Picture.Width do
        begin
            for y :=1 to Image.Picture.Height do
            begin
                if ((y mod SETKA)=0) and ((x mod SETKA)=0) then
                begin
                    Image.Canvas.Rectangle(x, y, x+10, y+10);
                    Labell1.Caption:=ColorToString(VideoWindow.Canvas.Pixels[x, y]);
                    //*****
                    if Ar[X, Y]<>Image.Canvas.Pixels[x, y] then
                    begin
                        Ar[X, Y]:=Image.Canvas.Pixels[x, y];
                        Form3.Memo1.lines.add('');
                    end;
                    //*****
                end;
            end;
        end;

        if ((y mod SETKA)=0) and (x = 1) then
        begin

```



```
    end;  
    //-----  
    go:=true;  
    end;  
  
    procedure TForm1.BitBtn1Click(Sender: TObject);  
    begin  
        Label1.Caption:=inttostr(Image.Picture.Width); // Діагональ зображення  
        Label2.Caption:=inttostr(Image.Picture.Height); // горизонталь зображення  
    end;  
  
end.
```

КБПЗ\_2024

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - підключення плагінів PLUG.pas**

```
// Copyrights © Кривда Олексій Сергійович, КБ-20, ЦНТУ КБПЗ
unit PLUG;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур

  TForm_9 = class(TForm) // Оголошення класу

    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;

    DoT_A:TpluginsAddStrings;

    DoT_B:TpluginsAddFiles;

    DoT_C:TpluginsAddData;

  private
    { Private секція }

  public//зона бачення змінних та типів даних
    { Public секція }

  end;

var
  Form9: TForm9;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - бібліотека відео потоку DSPack.pas**

```

// Copyrights © Кривда Олексій Сергійович, КБ-20, ЦНТУ КБПЗ
unit DSPack; // Бібліотека відео

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Classes, SysUtils, Messages, Graphics, Forms, Controls, ActiveX,
  DirectShow9, DirectDraw, DSUtil, ComCtrls, MMSystem, Math, Consts, ExtCtrls,
  MultiMon, Dialogs, Registry, SyncObjs, Direct3D9, WMF9;

Const // Константи
  WM_GRAPHNOTIFY = WM_APP + 1;

  WM_CAPTURE_BITMAP = WM_APP + 2;
type
// Об'ява типів
  TVideoMode = ( // Тип режиму
    vmNormal,
    vmVMR
  );

  TGraphMode = ( // Якості
    gmNormal,
    gmCapture,
    gmCAM
  );

{$IFDEF VER140}
  TVMRRenderDevice = (
    rdOverlay = 1,
    rdVidMem = 2,
    rdSysMem = 4
  );
{$ELSE}
  TVMRRenderDevice = Integer;
const
  rdOverlay = 1;
  rdVidMem = 2;
  rdSysMem = 4;
type
{$ENDIF}

  {@exclude}
  TGraphState = ( // Поточне дія
    gsUninitialized,
    gsStopped,
    gsPaused,
    gsPlaying
  );

  { Структура відео потоку }

  // Абсолютна позиція у потоці
  TSeekingCap = (
    CanSeekAbsolute,
    CanSeekForwards,
    CanSeekBackwards,
    CanGetCurrentPos,
    CanGetStopPos,
    CanGetDuration,
    CanPlayBackwards,
    CanDoSegments,
    IMediaSeeking.SetPositions).Source);
  TSeekingCaps = set of TSeekingCap;

```

```

TVMRPreference = (
    vpForceOffscreen,
    vpForceOverlays,
    vpForceMixer,
    vpDoNotRenderColorKeyAndBorder,
    vpRestrictToInitialMonitor,
    vpPreferAGPMemWhenMixing
);

PVMRPreferences = ^TVMRPreferences;
TVMRPreferences = set of TVMRPreference;

TOnDSEvent=procedure(sender: TComponent; Event, Param1, Param2: Integer) of
    object;
TOnGraphBufferingData=procedure(sender: TObject; Buffering: boolean) of object ;
TOnGraphComplete=procedure(sender: TObject; Result: HRESULT; Renderer:
IBaseFilter) of object;
    TOnGraphDeviceLost                = procedure(sender: TObject; Device: IUnknown;
        Removed: Boolean) of object ;
    TOnGraphEndOfSegment                = procedure(sender: TObject; StreamTime:
TReferenceTime; NumSegment: Cardinal) of object ;
    TOnDSResult                        = procedure(sender: TObject; Result: HRESULT) of
object ;
    TOnGraphFullscreenLost              = procedure(sender: TObject; Renderer:
IBaseFilter) of object ;
    TOnGraphOleEvent                    = procedure(sender: TObject; String1, String2:
WideString) of object ;
    TOnGraphOpeningFile                 = procedure(sender: TObject; opening: boolean) of
object ;
    TOnGraphSNDDevError                 = procedure(sender: TObject; OccurWhen:
TSndDevErr; ErrorCode: LongWord) of object ;
    TOnGraphStreamControl               = procedure(sender: TObject; PinSender: IPin;
        Cookie: LongWord) of object ;
    TOnGraphStreamError                 = procedure(sender: TObject; Operation: HRESULT;
        Value: LongWord) of object ;
    TOnGraphVideoSizeChanged            = procedure(sender: TObject; Width, height: word)
of object ;
    TOnGraphTimeCodeAvailable           = procedure(sender: TObject; From: IBaseFilter;
        DeviceID: LongWord) of object ;
    TOnGraphEXTDeviceModeChange         = procedure(sender: TObject; NewMode, DeviceID:
        LongWord) of object ;
    TOnGraphVMRRenderDevice             = procedure(sender: TObject; RenderDevice:
TVMRRenderDevice) of object;

TOnCAMDataStreamChange                 = procedure(sender: TObject; stream, lcid: Integer;
        Lang: string) of object;
    TOnCAMCurrentTime                    = procedure(sender: TObject; Hours,
        minutes,seconds,frames,frate : Integer) of object;
    TOnCAMTitleChange                    = procedure(sender: TObject; title: Integer) of object;
TOnCAMChapterStart=procedure(sender: TObject; chapter: Integer) of object;
TOnCAMValidUOPSChange = procedure(sender: TObject; UOPS: Integer) of object;
TOnCAMChange = procedure(sender: TObject; total,current: Integer) of object;
TOnCAMStillOn = procedure(sender: TObject; NoButtonAvailable: boolean; seconds:
        Integer) of object;
TOnCAMSubpictureStreamChange = procedure(sender: TObject; SubNum, lcid: Integer;
        Lang: string) of object;
TOnCAMPlaybackRateChange = procedure(sender: TObject; rate: single) of object;
TOnCAMParentalLevelChange= procedure(sender: TObject; level: Integer) of object;
TOnCAMAnglesAvailable=procedure(sender: TObject; available: boolean) of object;
TOnCAMButtonAutoActivated=procedure(sender: TObject; Button: Cardinal) of
object;
TOnCAMCMD=procedure(sender: TObject; CmdID: Cardinal) of object;
TOnCAMCurrentHMSFTime = procedure(sender: TObject; HMSFTimeCode:
        TCAMHMSFTimeCode; TimeCode: TCAMTimeCode) of object;
TOnCAMKaraokeMode=procedure(sender: TObject; Played: boolean) of object;
TOnBuffer = procedure(sender: TObject; SampleTime: Double; pBuffer: Pointer;
        BufferLen: longint) of object ;

```

```

// *****
// Структура TFilterOperation
// *****

TFilterOperation = (
    foAdding,
    foAdded,
    foRemoving,
    foRemoved,
    foRefresh
);

IFilter = interface // Об'ява інтерфейсу
['{887F94DA-29E9-44C6-B48E-1FBF0FB59878}']
    function GetFilter: IBaseFilter;
    function GetName: string;
    procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
end;

TControlEvent = (
    cePlay,
    cePause,
    ceStop,
    ceFileRendering,
    ceFileRendered,
    ceCAMRendering,
    ceCAMRendered,
    ceActive
);

IEvent = interface // Інтерфейс
['{6C0DCD7B-1A98-44EF-A6D5-E23CBC24E620}']
    procedure GraphEvent(Event, Param1, Param2: integer);
    procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
end;

// *****
// Структура TFilterGraph
// *****
TFilterGraph = class(TComponent) // Оголошення класу
private
    FActive      : boolean;
    FAutoCreate  : boolean;
    FHandle      : THandle; // ОСНОВНИЙ ВКАЗІВНИК
    FMode        : TGraphMode;

    FFilters: TInterfaceList;
    FGraphEvents: TInterfaceList;

    // налаштування
    FFilterGraph : IGraphBuilder;
    FCaptureGraph : ICaptureGraphBuilder2;
    FCAMGraph     : ICamGraphBuilder;
    FMediaEventEx : IMediaEventEx;
    FGraphEdit    : boolean;
    FGraphEditID  : Integer;

    // Файл Log
    FLogFileName: String;
    FLogFile: TFileStream;
    FOnActivate: TNotifyEvent;
    // All Events Code
    FOnDSEvent : TOnDSEvent;
    // Generic Graph Events
    FOnGraphBufferingData : TOnGraphBufferingData;
    FOnGraphClockChanged  : TNotifyEvent;
    FOnGraphComplete      : TOnGraphComplete;
    FOnGraphDeviceLost    : TOnGraphDeviceLost;

```

```

FOnGraphEndOfSegment           : TOnGraphEndOfSegment;
FOnGraphErrorStillPlaying      : TOnDSResult;
FOnGraphErrorAbort             : TOnDSResult;
FOnGraphFullscreenLost        : TOnGraphFullscreenLost;
FOnGraphChanged               : TNotifyEvent;
FOnGraphOleEvent              : TOnGraphOleEvent;
FOnGraphOpeningFile           : TOnGraphOpeningFile;
FOnGraphPaletteChanged        : TNotifyEvent;
FOnGraphPaused                : TOnDSResult;
FOnGraphQualityChange         : TNotifyEvent;
FOnGraphSNDDDevInError        : TOnGraphSNDDDevError;
FOnGraphSNDDDevOutError       : TOnGraphSNDDDevError;
FOnGraphStepComplete          : TNotifyEvent;
FOnGraphStreamControlStarted  : TOnGraphStreamControl;
FOnGraphStreamControlStopped  : TOnGraphStreamControl;
FOnGraphStreamErrorStillPlaying : TOnGraphStreamError;
FOnGraphStreamErrorStopped    : TOnGraphStreamError;
FOnGraphUserAbort             : TNotifyEvent;
FOnGraphVideoSizeChanged      : TOnGraphVideoSizeChanged;
FOnGraphTimeCodeAvailable     : TOnGraphTimeCodeAvailable;
FOnGraphEXTDeviceModeChange   : TOnGraphEXTDeviceModeChange;
FOnGraphClockUnset           : TNotifyEvent;
FOnGraphVMRRenderDevice       : TOnGraphVMRRenderDevice;

FOnCAMDataStreamChange        : TOnCAMDataStreamChange;
FOnCAMCurrentTime            : TOnCAMCurrentTime;
FOnCAMTitleChange            : TOnCAMTitleChange;
FOnCAMChapterStart           : TOnCAMChapterStart;
FOnCAMAngleChange            : TOnCAMChange;
FOnCAMValidUOPSCheck         : TOnCAMValidUOPSCheck;
FOnCAMButtonChange           : TOnCAMChange;
FOnCAMChapterAutoStop        : TNotifyEvent;
FOnCAMStillOn                : TOnCAMStillOn;
FOnCAMStillOff               : TNotifyEvent;
FOnCAMSubpictureStreamChange : TOnCAMSubpictureStreamChange;
FOnCAMNoFP_PGC               : TNotifyEvent;
FOnCAMPlaybackRateChange     : TOnCAMPlaybackRateChange;
FOnCAMParentalLevelChange    : TOnCAMParentalLevelChange;
FOnCAMPlaybackStopped        : TNotifyEvent;
FOnCAMAnglesAvailable        : TOnCAMAnglesAvailable;
FOnCAMPlayPeriodAutoStop     : TNotifyEvent;
FOnCAMButtonAutoActivated    : TOnCAMButtonAutoActivated;
FOnCAMCMDStart               : TOnCAMCMD;
FOnCAMCMDEnd                 : TOnCAMCMD;
FOnCAMDiscEjected            : TNotifyEvent;
FOnCAMDiscInserted           : TNotifyEvent;
FOnCAMCurrentHMSFTime        : TOnCAMCurrentHMSFTime;
FOnCAMKaraokeMode            : TOnCAMKaraokeMode;
FOnCAMWarningInvalidCAM1_0Disc : TNotifyEvent;//=1,
FOnCAMWarningFormatNotSupported : TNotifyEvent;//=2,
FOnCAMWarningIllegalNavCommand : TNotifyEvent;//=3
FOnCAMWarningOpen            : TNotifyEvent;//=4
FOnCAMWarningSeek            : TNotifyEvent;//=5
FOnCAMWarningRead            : TNotifyEvent;//=6
FOnCAMDomainFirstPlay        : TNotifyEvent;
FOnCAMDomainVideoManagerMenu : TNotifyEvent;
FOnCAMDomainVideoTitleSetMenu : TNotifyEvent;
FOnCAMDomainTitle            : TNotifyEvent;
FOnCAMDomainStop             : TNotifyEvent;
FOnCAMErrorUnexpected        : TNotifyEvent;
FOnCAMErrorCopyProtectFail   : TNotifyEvent;
FOnCAMErrorInvalidCAM1_0Disc : TNotifyEvent;
FOnCAMErrorInvalidDiscRegion : TNotifyEvent;
FOnCAMErrorLowParentalLevel   : TNotifyEvent;
FOnCAMErrorMacrovisionFail    : TNotifyEvent;
FOnCAMErrorIncompatibleSystemAndDecoderRegions : TNotifyEvent;
FOnCAMErrorIncompatibleDiscAndDecoderRegions : TNotifyEvent;
procedure HandleEvents;
procedure WndProc(var Msg: TMessage);

```

```

procedure SetActive(Activate: boolean);
procedure SetGraphMode(Mode: TGraphMode);
procedure SetGraphEdit(enable: boolean);
procedure ClearOwnFilters;
procedure AddOwnFilters;
procedure GraphEvents(Event, Param1, Param2: integer);
procedure ControlEvents(Event: TControlEvent; Param: integer = 0);
procedure SetLogFile(FileName: String);
function GetState: TGraphState;
function GetVolume: integer;
procedure SetVolume(Volume: Integer);
function GetBalance: integer;
procedure SetBalance(Balance: integer);
function GetSeekCaps: TSeekingCaps;
procedure SetRate(Rate: double);
function GetRate: double;
function GetDuration: integer;
protected
  procedure DoEvent(Event, Param1, Param2: Integer); virtual;
  procedure InsertFilter(AFilter: IFilter);
  procedure RemoveFilter(AFilter: IFilter);
  procedure InsertEventNotifier(AEvent: IEvent);
  procedure RemoveEventNotifier(AEvent: IEvent);
public //зона бачення змінних та типів даних
  property Duration: Integer read GetDuration; // завдання властивості типу
  property Rate: Double read GetRate write SetRate;
  property SeekCapabilities: TSeekingCaps read GetSeekCaps;
  property Balance: integer read GetBalance write SetBalance;
  property Volume: integer read GetVolume write SetVolume;
  property State: TGraphState read GetState;
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор
  procedure Loaded; override;
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  function Play: boolean;
  function Pause: boolean;
  function Stop: boolean;
  procedure DisconnectFilters;
  procedure ClearGraph;
  function RenderFile(FileName: WideString): HRESULT;
  function RenderFileEx(FileName: WideString): HRESULT;
  function RenderCAM(out status: TAMCAMRenderStatus;
  FileName: WideString = ''; Mode: Integer = AM_CAM_HWDEC_PREFER): HRESULT;

  procedure CAMSaveBookmark(BookMarkFile: WideString);
  procedure CAMRestoreBookmark(BookMarkFile: WideString);
published //зона бачення змінних
  property LogFile: String read FLogFileName write SetLogFile;
  property Active: boolean read FActive write SetActive default False;
  property AutoCreate: boolean read FAutoCreate write FAutoCreate default
False;
  property Mode: TGraphMode read FMode write SetGraphMode default gmNormal;
  property GraphEdit: boolean read FGraphEdit write SetGraphEdit;

  // -----
  // Події
  // -----

  property OnActivate: TNotifyEvent read FOnActivate write FOnActivate;
  property OnDSEvent: TOnDSEvent read FOnDSEvent write FOnDSEvent;

  property OnGraphBufferingData: TOnGraphBufferingData read
    FOnGraphBufferingData write FOnGraphBufferingData;
  property OnGraphClockChanged: TNotifyEvent read FOnGraphClockChanged write
FOnGraphClockChanged;
  property OnGraphComplete: TOnGraphComplete read FOnGraphComplete write
FOnGraphComplete;

```

```

property OnGraphDeviceLost: TOnGraphDeviceLost read FOnGraphDeviceLost write
FOnGraphDeviceLost;

property OnGraphEndOfSegment: TOnGraphEndOfSegment read FOnGraphEndOfSegment
write FOnGraphEndOfSegment;
property OnGraphErrorStillPlaying: TOnDSResult read
FOnGraphErrorStillPlaying write FOnGraphErrorStillPlaying;

property OnGraphErrorAbort: TOnDSResult read FOnGraphErrorAbort write
FOnGraphErrorAbort;

property OnGraphFullscreenLost: TOnGraphFullscreenLost read
FOnGraphFullscreenLost write FOnGraphFullscreenLost;

property OnGraphChanged: TNotifyEvent read FOnGraphChanged write
FOnGraphChanged;

property OnGraphOleEvent: TOnGraphOleEvent read FOnGraphOleEvent write
FOnGraphOleEvent;

property OnGraphOpeningFile: TOnGraphOpeningFile read FOnGraphOpeningFile
write FOnGraphOpeningFile;

property OnGraphPaletteChanged: TNotifyEvent read FOnGraphPaletteChanged
write FOnGraphPaletteChanged;

property OnGraphPaused: TOnDSResult read FOnGraphPaused write
FOnGraphPaused;

property OnGraphQualityChange: TNotifyEvent read FOnGraphQualityChange write
FOnGraphQualityChange;
property OnGraphSNDDevInError: TOnGraphSNDDevError read
FOnGraphSNDDevInError write FOnGraphSNDDevInError;
property OnGraphSNDDevOutError: TOnGraphSNDDevError read
FOnGraphSNDDevOutError write FOnGraphSNDDevOutError;
property OnGraphStepComplete: TNotifyEvent read FOnGraphStepComplete write
FOnGraphStepComplete;
property OnGraphStreamControlStarted: TOnGraphStreamControl read
FOnGraphStreamControlStarted write FOnGraphStreamControlStarted;
property OnGraphStreamControlStopped: TOnGraphStreamControl read
FOnGraphStreamControlStopped write FOnGraphStreamControlStopped;
property OnGraphStreamErrorStillPlaying : TOnGraphStreamError read
FOnGraphStreamErrorStillPlaying write FOnGraphStreamErrorStillPlaying;

property OnGraphStreamErrorStopped: TOnGraphStreamError read
FOnGraphStreamErrorStopped write FOnGraphStreamErrorStopped;
property OnGraphUserAbort: TNotifyEvent read FOnGraphUserAbort write
FOnGraphUserAbort;
property OnGraphVideoSizeChanged: TOnGraphVideoSizeChanged read
FOnGraphVideoSizeChanged write FOnGraphVideoSizeChanged;
property OnGraphTimeCodeAvailable: TOnGraphTimeCodeAvailable read
FOnGraphTimeCodeAvailable write FOnGraphTimeCodeAvailable;
property OnGraphEXTDeviceModeChange: TOnGraphEXTDeviceModeChange read
FOnGraphEXTDeviceModeChange write FOnGraphEXTDeviceModeChange;
property OnGraphClockUnset: TNotifyEvent read FOnGraphClockUnset write
FOnGraphClockUnset;
property OnGraphVMRRenderDevice: TOnGraphVMRRenderDevice read
FOnGraphVMRRenderDevice write FOnGraphVMRRenderDevice;
property OnCAMDataStreamChange: TOnCAMDataStreamChange read
FOnCAMDataStreamChange write FOnCAMDataStreamChange;
property OnCAMCurrentTime: TOnCAMCurrentTime read FOnCAMCurrentTime write
FOnCAMCurrentTime;
property OnCAMTitleChange: TOnCAMTitleChange read FOnCAMTitleChange write
FOnCAMTitleChange;
property OnCAMChapterStart: TOnCAMChapterStart read FOnCAMChapterStart write
FOnCAMChapterStart;
property OnCAMAngleChange: TOnCAMChange read FOnCAMAngleChange write
FOnCAMAngleChange;

```

```

    property OnCAMValidUOPSChange: TOnCAMValidUOPSChange read
FOnCAMValidUOPSChange write FOnCAMValidUOPSChange;
    property OnCAMButtonChange: TOnCAMChange read FOnCAMButtonChange write
FOnCAMButtonChange;
    property OnCAMChapterAutoStop: TNotifyEvent read FOnCAMChapterAutoStop write
FOnCAMChapterAutoStop;
    property OnCAMStillOn: TOnCAMStillOn read FOnCAMStillOn write FOnCAMStillOn;
    property OnCAMStilloff: TNotifyEvent read FOnCAMStilloff write
FOnCAMStilloff;
    property OnCAMSubpictureStreamChange: TOnCAMSubpictureStreamChange read
FOnCAMSubpictureStreamChange write FOnCAMSubpictureStreamChange;
    property OnCAMNoFP_PGC: TNotifyEvent read FOnCAMNoFP_PGC write
FOnCAMNoFP_PGC;
    property OnCAMPlaybackRateChange: TOnCAMPlaybackRateChange read
FOnCAMPlaybackRateChange write FOnCAMPlaybackRateChange;
    property OnCAMParentalLevelChange: TOnCAMParentalLevelChange read
FOnCAMParentalLevelChange write FOnCAMParentalLevelChange;
    property OnCAMPlaybackStopped: TNotifyEvent read FOnCAMPlaybackStopped write
FOnCAMPlaybackStopped;
    property OnCAMAnglesAvailable: TOnCAMAnglesAvailable read
FOnCAMAnglesAvailable write FOnCAMAnglesAvailable;
    property OnCAMPlayPeriodAutoStop: TNotifyEvent read FOnCAMPlayPeriodAutoStop
write FOnCAMPlayPeriodAutoStop;
    property OnCAMButtonAutoActivated: TOnCAMButtonAutoActivated read
FOnCAMButtonAutoActivated write FOnCAMButtonAutoActivated;
    property OnCAMCMDStart: TOnCAMCMD read FOnCAMCMDStart Write FOnCAMCMDStart;
    property OnCAMCMDEnd: TOnCAMCMD read FOnCAMCMDEnd Write FOnCAMCMDEnd;
    property OnCAMDiscEjected: TNotifyEvent read FOnCAMDiscEjected Write
FOnCAMDiscEjected;
    property OnCAMDiscInserted: TNotifyEvent read FOnCAMDiscInserted write
FOnCAMDiscInserted;
    property OnCAMCurrentHMSFTime: TOnCAMCurrentHMSFTime read
FOnCAMCurrentHMSFTime write FOnCAMCurrentHMSFTime;
    property OnCAMKaraokeMode: TOnCAMKaraokeMode read FOnCAMKaraokeMode write
FOnCAMKaraokeMode;
    property OnCAMDomainFirstPlay: TNotifyEvent read FOnCAMDomainFirstPlay write
FOnCAMDomainFirstPlay;
    property OnCAMDomainVideoManagerMenu: TNotifyEvent read
FOnCAMDomainVideoManagerMenu write FOnCAMDomainVideoManagerMenu;
    property OnCAMDomainVideoTitleSetMenu: TNotifyEvent read
FOnCAMDomainVideoTitleSetMenu write FOnCAMDomainVideoTitleSetMenu;
    property OnCAMDomainTitle: TNotifyEvent read FOnCAMDomainTitle write
FOnCAMDomainTitle;
    property OnCAMDomainStop: TNotifyEvent read FOnCAMDomainStop write
FOnCAMDomainStop;
    property OnCAMErrorUnexpected: TNotifyEvent read FOnCAMErrorUnexpected write
FOnCAMErrorUnexpected;
    property OnCAMErrorInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMErrorInvalidCAM1_0Disc write FOnCAMErrorInvalidCAM1_0Disc;

    property OnCAMErrorInvalidDiscRegion: TNotifyEvent read
FOnCAMErrorInvalidDiscRegion write FOnCAMErrorInvalidDiscRegion;
    property OnCAMErrorLowParentalLevel: TNotifyEvent read
FOnCAMErrorLowParentalLevel write FOnCAMErrorLowParentalLevel;

    property OnCAMErrorMacrovisionFail: TNotifyEvent read
FOnCAMErrorMacrovisionFail write FOnCAMErrorMacrovisionFail;
    property OnCAMErrorIncompatibleSystemAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleSystemAndDecoderRegions write
FOnCAMErrorIncompatibleSystemAndDecoderRegions;
    property OnCAMErrorIncompatibleDiscAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleDiscAndDecoderRegions write
FOnCAMErrorIncompatibleDiscAndDecoderRegions;
    property OnCAMWarningInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMWarningInvalidCAM1_0Disc write FOnCAMWarningInvalidCAM1_0Disc;
    property OnCAMWarningFormatNotSupported : TNotifyEvent read
FOnCAMWarningFormatNotSupported write FOnCAMWarningFormatNotSupported;
    property OnCAMWarningIllegalNavCommand : TNotifyEvent read
FOnCAMWarningIllegalNavCommand write FOnCAMWarningIllegalNavCommand;

```

```

    property OnCAMWarningOpen: TNotifyEvent read FOnCAMWarningOpen write
FOnCAMWarningOpen;
    property OnCAMWarningSeek: TNotifyEvent read FOnCAMWarningSeek write
FOnCAMWarningSeek;
    property OnCAMWarningRead: TNotifyEvent read FOnCAMWarningRead write
FOnCAMWarningRead;
    end;

// *****
//   Структура TVMROptions
// *****

TVideoWindow = class; // Оголошення класу

TVMRVideoMode = (
    vmrWindowed,
    vmrWindowless,
    vmrRenderless
);

TVMROptions = class(TPersistent) // Оголошення класу
private
    FOwner: TVideoWindow;
    FStreams: cardinal;
    FPreferences: TVMRPreferences;
    FMode: TVMRVideoMode;
    FKeepAspectRatio: boolean;
    procedure SetStreams(Streams: cardinal);
    procedure SetPreferences(Preferences: TVMRPreferences);
    procedure SetMode(AMode: TVMRVideoMode);
    procedure SetKeepAspectRatio(Keep: boolean);
public //зона бачення змінних та типів даних
    constructor Create(AOwner: TVideoWindow); //Конструктор
published //зона бачення змінних та типів даних
    property Mode: TVMRVideoMode read FMode write SetMode;
    // завдання властивості типу
    property Streams: Cardinal read FStreams write SetStreams default 4;
    property Preferences: TVMRPreferences read FPreferences write SetPreferences
default [vpForceMixer];
    property KeepAspectRatio: boolean read FKeepAspectRatio write
SetKeepAspectRatio default True;
    end;

// *****
//   структура TVideoWindow
// *****

TAbstractAllocator = class(TInterfacedObject) // Оголошення класу
    constructor Create(out hr: HRESULT; wnd: THandle; d3d: IDirect3D9 = nil;
d3dd: IDirect3DDevice9 = nil); virtual; abstract;
    end;
TAbstractAllocatorClass = class of TAbstractAllocator; // Оголошення класу
TVideoWindow = class(TCustomControl, IFilter, IEvent) // Оголошення класу
private
    FMode          : TVMRVideoMode;
    FVMOptions     : TVMROptions;
    FBaseFilter    : IBaseFilter;
    FVideoWindow  : IVideoWindow;
    FWindowLess   : IVMRWindowlessControl9;

    FFullScreen   : boolean;
    FFilterGraph  : TFilterGraph;
    FWindowState  : LongWord;
    FWindowStateEx : LongWord;
    FTopMost     : boolean;
    FIsFullScreen : boolean;
    FOnPaint     : TNotifyEvent;
    FKeepAspectRatio: boolean;

```

```

FAllocatorClass: TAbstractAllocatorClass;
FCurrentAllocator: TAbstractAllocator;
FRenderLessUserID: Cardinal;
procedure SetVideoMode(AMode: TVideoMode);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
procedure SetFullScreen(Value: boolean);
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure GraphEvent(Event, Param1, Param2: integer);
function GetName: string;
function GetVideoHandle: THandle;
procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
procedure SetTopMost(TopMost: boolean);
function GetVisible: boolean;
procedure SetVisible(Vis: boolean);
protected
  {@exclude}
  procedure Loaded; override;
  {@exclude}
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
  {@exclude}
  procedure Resize; override;
  {@exclude}
  procedure ConstrainedResize(var MinWidth, MinHeight, MaxWidth, MaxHeight:
Integer); override;
  {@exclude}
  function GetFilter: IBaseFilter;
  {@exclude}
  procedure WndProc(var Message: TMessage); override;
  {@exclude}
  procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y:
Integer); override;
  {@exclude}
  procedureMouseMove(Shift: TShiftState; X, Y: Integer); override;
  {@exclude}
  procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
  {@exclude}
  procedure Paint; override;
public //зона бачення змінних та типів даних
  {@exclude}
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  constructor Create(AOwner: TComponent);override; //Конструктор
  destructor Destroy; override; //Деструктор
  class function CheckVMR: boolean;
  function VMRGetBitmap(Stream: TStream): boolean;
  function CheckInputPinsConnected: boolean;
  procedure SetAllocator(Allocator: TAbstractAllocatorClass; UserID:
Cardinal);
  published //зона бачення змінних
  property OnPaint: TNotifyEvent read FOnPaint write FOnPaint;
  // завдання властивості типу
  property FullScreenTopMost: boolean read FTopMost write SetTopMost
    default false;
  property Mode: TVideoMode read FMode write SetVideoMode default vmNormal;
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
  property VideoHandle: THandle read GetVideoHandle;
  property VMROptions: TVMROptions read FVMROptions write FVMROptions;
  property FullScreen: boolean read FFullScreen write SetFullScreen
    default false;

  property Color;
  property Visible: boolean read GetVisible write SetVisible default True;
  property ShowHint;
  property Anchors;
  property Canvas;
  property PopupMenu;
  property Align;
  property TabStop default True;

```

```

property OnEnter;
property OnExit;
property OnKeyDown;
property OnKeyPress;
property OnKeyUp;
property OnCanResize;
property OnClick;
property OnConstrainedResize;
property OnDblClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
property OnMouseWheel;
property OnMouseWheelDown;
property OnMouseWheelUp;
property OnResize;
end;

TSampleGrabber = class; // Оголошення класу

TSampleGrabber = class(TComponent, IFilter,
                        ISampleGrabberCB) // Оголошення класу
private
    FOnBuffer: TOnBuffer;
    FBaseFilter: IBaseFilter;
    FFilterGraph : TFilterGraph;
    FMediaType: TMediaType;
    BMPInfo : PBitmapInfo;
    FCriticalSection: TCriticalSection;
    function GetFilter: IBaseFilter;
    function GetName: string;
    procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
    procedure SetFilterGraph(AFilterGraph: TFilterGraph);
    function SampleCB(SampleTime: Double; pSample: IMediaSample): HRESULT;
        stdcall;
    function BufferCB(SampleTime: Double; pBuffer: PByte; BufferLen: longint):
        HRESULT; stdcall;
protected
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
    SampleGrabber: ISampleGrabber;
    InPutPin : IPin;
    OutPutPin : IPin;
    constructor Create(AOwner: TComponent); override; //Конструктор
    destructor Destroy; override; //Деструктор
    procedure UpdateMediaType;
    function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;

    procedure SetBMPCompatible(Source: PAMMediaType; SetDefault: cardinal);
    function GetBitmap(Bitmap: TBitmap; Buffer: Pointer; BufferLen: Integer):
boolean; overload;
    function GetBitmap(Bitmap: TBitmap): boolean; overload;
    class function CheckFilter: boolean;
published //зона бачення змінних
    property OnBuffer: TOnBuffer read FOnBuffer write FOnBuffer;
    property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
    // завдання властивості типу
    property MediaType: TMediaType read FMediaType write FMediaType;
end;

// *****
// структура TFilter
// *****
TFilter = class(TComponent, IFilter) // Оголошення класу
private
    FFilterGraph : TFilterGraph;

```

```

FBaseFilter: TBaseFilter;
FFilter: IBaseFilter;
function GetFilter: IBaseFilter;
function GetName: string;
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
protected
  { @exclude }
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
  { Конструктор }
  constructor Create(AOwner: TComponent); override;
  { Деструктор method. }
  destructor Destroy; override; //Деструктор
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published //зона бачення змінних та типів даних
  property BaseFilter: TBaseFilter read FBaseFilter write FBaseFilter;
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
end;

// *****
// структура TASFWriter
// *****

TASFWriter = class(TComponent, IFilter) // Оголошення класу
private
  FFilterGraph : TFilterGraph;
  FFilter      : IBaseFilter;
  FPort        : Cardinal;
  FMaxUsers    : Cardinal;
  FProfile     : TWMPofiles8;
  FFileName    : WideString;
  FAutoIndex   : boolean;
  FMultiPass   : boolean;
  FDontCompress: boolean;
  function GetProfile: TWMPofiles8;
  procedure SetProfile(profile: TWMPofiles8);
  function GetFileName: String;
  procedure SetFileName(FileName: String);
  function GetFilter: IBaseFilter;
  function GetName: string;
  procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
  procedure SetFilterGraph(AFilterGraph: TFilterGraph);
protected
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public //зона бачення змінних та типів даних
  WriterAdvanced2 : IWMWriterAdvanced2;
  { NetWork streaming configuration. }
  WriterNetworkSink : IWMWriterNetworkSink;
  { Вхідні дані }
  DataInput          : IPin;
  VideoInput         : IPin;
  DataStreamConfig   : IAMStreamConfig;
  VideoStreamConfig  : IAMStreamConfig;
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published //зона бачення змінних
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
  property Profile: TWMPofiles8 read GetProfile write SetProfile;
  property FileName: String read GetFileName write SetFileName;
  // завдання властивості типу
  property Port: DWORD read FPort write FPort;
  property MaxUsers: DWORD read FMaxUsers write FMaxUsers;

```

```

    property AutoIndex      : boolean read FAutoIndex write FAutoIndex default
True;
    property MultiPass      : boolean read FMultiPass write FMultiPass default
False;
    property DontCompress: boolean read FDontCompress write FDontCompress
default False;

end;

// *****
// структура TDSTackBar
// *****

TTimerEvent = procedure(sender: TObject; CurrentPos, StopPos: Cardinal) of
                object ;

TDSTackBar = class(TTrackBar, IEvent) // Оголошення класу
private
    FFilterGraph: TFilterGraph;
    FMediaSeeking: IMediaSeeking;
    FWindowHandle: HWND;
    FInterval: Cardinal;
    FOnTimer: TTimerEvent;
    FEnabled: Boolean;
    FMouseDown: boolean;
    procedure UpdateTimer;
    procedure SetTimerEnabled(Value: Boolean);
    procedure SetInterval(Value: Cardinal);
    procedure SetOnTimer(Value: TTimerEvent);
    procedure SetFilterGraph(AFilterGraph: TFilterGraph);
    procedure GraphEvent(Event, Param1, Param2: integer);
    procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
    procedure TimerWndProc(var Msg: TMessage);
    property TimerEnabled: Boolean read FEnabled write SetTimerEnabled;
protected
    { @exclude }
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
    { @exclude }
    procedure MouseUp(Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer); override;
    { @exclude }
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
        X, Y: Integer); override;
    { @exclude }
    procedure Timer; dynamic;
public //зона бачення змінних та типів даних
    { Конструктор }
    constructor Create(AOwner: TComponent); override;
    { Деструктор }
    destructor Destroy; override; //Деструктор
published //зона бачення змінних
    property FilterGraph: TFilterGraph read FFilterGraph Write SetFilterGraph;
    property TimerInterval: Cardinal read FInterval write SetInterval default
1000;
    property OnTimer: TTimerEvent read FOnTimer write SetOnTimer;
end;

{ @exclude }
TDSVideoWindowEx2 = class; // Оголошення класу

// *****
// структура TColorControl
// *****

TColorControl = class(TPersistent) // Оголошення класу
private
    FBrightness : Integer;
    FContrast   : Integer;

```

```

    FHue          : Integer;
    FSaturation   : Integer;
    FSharpness    : Integer;
    FGamma        : Integer;
    FUtilColor    : Boolean;
    FDefault      : TDDColorControl;
protected
{ Protected секція }
{ @exclude }
FOwner : TDSVideoWindowEx2;
{ @exclude }
Procedure SetBrightness(Value : Integer);
{ @exclude }
Procedure SetContrast(Value : Integer);
{ @exclude }
procedure SetHue(Value : Integer);
{ @exclude }
procedure SetSaturation(Value : Integer);
{ @exclude }
procedure SetSharpness(Value : Integer);
{ @exclude }
procedure SetGamma(Value : Integer);
{ @exclude }
procedure SetUtilColor(Value : Boolean);
{ @exclude }
function GetBrightness : Integer;
function GetContrast : Integer;
function GetHue : Integer;
function GetSaturation : Integer;
function GetSharpness : Integer;
function GetGamma : Integer;
{ @exclude }
function GetUtilColor : Boolean;
{ @exclude }
Procedure ReadDefault;
{ @exclude }
procedure UpdateColorControls;
{ @exclude }
procedure GetColorControls;
public //зона бачення змінних та типів даних
{ Public секція }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
procedure RestoreDefault;
Published //зона бачення змінних
property Brightness : Integer read GetBrightness write SetBrightness;
// завдання властивості типу
property Contrast : Integer read GetContrast write SetContrast;
property Hue : Integer read GetHue write SetHue;

property Saturation : Integer read GetSaturation write SetSaturation;

property Sharpness : Integer read GetSharpness write SetSharpness;

property Gamma : Integer read GetGamma write SetGamma;

property ColorEnable : Boolean read GetUtilColor write SetUtilColor;
end;

// *****
// структура TDSVideoWindowEx2Caps
// *****
TDSVideoWindowEx2Caps = class(TPersistent) // Оголошення класу
protected
{ Protected секція }
Owner : TDSVideoWindowEx2;
function GetCanOverlay : Boolean;
function GetCanControlBrigtness : Boolean;
function GetCanControlContrast : Boolean;
function GetCanControlHue : Boolean;

```

```

function GetCanControlSaturation : Boolean;
function GetCanControlSharpness : Boolean;
function GetCanControlGamma : Boolean;
function GetCanControlUtilizedColor : Boolean;
public
{ Public секція }
{ @exclude }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
published //зона бачення змінних
Property CanOverlayGraphic : Boolean read GetCanOverlay;
Property CanControlBrightness : Boolean read GetCanControlBrightness;
Property CanControlContrast : Boolean read GetCanControlContrast;
Property CanControlHue : Boolean read GetCanControlHue;

Property CanControlSaturation : Boolean read GetCanControlSaturation;
Property CanControlSharpness : Boolean read GetCanControlSharpness;
Property CanControlGamma : Boolean read GetCanControlGamma;
Property CanControlColorEnabled : Boolean read GetCanControlUtilizedColor;
end;

// *****
//  структура TOverlayCallback
// *****

TOverlayCallback = class(TInterfacedObject, IDDrawExclModeVideoCallBack)
    AOwner : TObject; // Оголошення класу
    constructor Create(Owner : TObject); virtual; //Конструктор
    function OnUpdateOverlay(bBefore: BOOL; dwFlags: DWORD; bOldVisible: BOOL;
        var prcOldSrc, prcOldDest: TRECT; bNewVisible: BOOL; var prcNewSrc,
        prcNewDest: TRECT): HRESULT; stdcall;
    function OnUpdateColorKey(var pKey: TCOLORKEY; dwColor: DWORD): HRESULT;
        stdcall;
    function OnUpdateSize(dwWidth, dwHeight, dwARWidth, dwARHeight: DWORD):
        HRESULT; stdcall;
end;

// *****
//  структура TDSVideoWindowEx2
// *****

TRatioModes = (rmStretched, rmLetterBox, rmCrop);

TOverlayVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;

{ @exclude }
TCursorVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;
TDSVideoWindowEx2 = class(TCustomControl, IFilter, IEvent) // Оголошення класу
private
FVideoWindow          : IVideoWindow;
FFilterGraph          : TFilterGraph;
FBaseFilter           : IBaseFilter;
FOverlayMixer        : IBaseFilter;
FVideoRenderer       : IBaseFilter;
FDDXM                : IDDrawExclModeVideo;
FFullScreen          : Boolean;
FTopMost             : Boolean;
FColorKey            : TColor;
FWindowStyle         : LongWord;
FWindowStyleEx       : LongWord;
FVideoRect           : TRect;
FOnPaint             : TNotifyEvent;
FOnColorKey          : TNotifyEvent;
FOnCursorVisible     : TCursorVisibleEvent;
FOnOverlay           : TOverlayVisibleEvent;
FColorControl        : TColorControl;
FCaps                : TDSVideoWindowEx2Caps;
FZoom                : Integer;

```

```

FAspectMode      : TRatioModes;
FNoScreenSaver   : Boolean;
FIdleCursor      : Integer;
FMonitor         : TMonitor;
FFullscreenControl : TForm;
GraphWasUpdated  : Boolean;
FOldParent       : TWinControl;
OverlayCallback  : TOverlayCallback;
GraphBuildOK     : Boolean;
FVideoWindowHandle : HWND;
LMousePos        : TPoint;
LCursorMov       : DWord;
RememberCursor   : TCursor;
IsHidden         : Bool;
FOverlayVisible  : Boolean;
OldDesktopColor  : Longint;
OldDesktopPic    : String;
FDesktopPlay     : Boolean;
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure GraphEvent(Event, Param1, Param2: integer);
function GetName: string;
procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
procedure SetTopMost(TopMost: boolean);
procedure SetZoom(Value : Integer);
function UpdateGraph : HRESULT;
function GetVideoInfo : HRESULT;
procedure SetAspectMode(Value : TRatioModes);
procedure FullScreenCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure SetVideoZOrder;
protected
  {@exclude}
  function GetFilter: IBaseFilter;
  {@exclude}
  procedure resize; override;
  {@exclude}
  procedure Loaded; override;
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
  {@exclude}
  procedure WndProc(var Message: TMessage); override;
  procedure Paint; override;
  procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y:
Integer); override;
  procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
  procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
  procedure MyIdleHandler(Sender: TObject; var Done: Boolean);
  procedure RefreshVideoWindow;
public //зона бачення змінних та типів даних
  constructor Create(AOwner: TComponent); override; //Конструктор
  destructor Destroy; override; //Деструктор

  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
  procedure ClearBack;
  procedure StartDesktopPlayback; overload;
  procedure StartDesktopPlayBack(OnMonitor : TMonitor); overload;
  procedure NormalPlayback;
  procedure StartFullScreen; overload;
  procedure StartFullScreen(OnMonitor : TMonitor); overload;
  property FullScreen: boolean read FFullScreen;
  property DesktopPlayback : Boolean Read FDesktopPlay;
  property Canvas;
  property ColorKey : TColor read FColorKey;
  property Capabilities : TDSVideoWindowEx2Caps read FCaps;
  property OverlayVisible : Boolean read FOverlayVisible;
published //зона бачення змінних
  property AspectRatio : TRatioModes read FAspectMode write SetAspectMode;

```

```

property AutoHideCursor : Integer read FIdleCursor write FIdleCursor;
// завдання властивості типу
property DigitalZoom : Integer read FZoom write SetZoom;
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
property FullScreenTopMost: boolean read FTopMost write SetTopMost default
false;
property OnColorKeyChanged: TNotifyEvent read FOnColorKey write FOnColorKey;
property ColorControl: TColorControl read FColorControl write FColorControl;

property NoScreenSaver: Boolean read FNoScreenSaver write FNoScreenSaver;

property OnOverlayVisible: TOverlayVisibleEvent read FOnOverlay write
FOnOverlay;

property OnPaint : TNotifyevent read FOnPaint Write FOnPaint;
property OnCursorShowHide : TCursorVisibleEvent read FOnCursorVisible write
FOnCursorVisible;

property Color;
property Visible;
property ShowHint;
property Anchors;
property PopupMenu;
property Align;
property TabStop default True;
property OnEnter;
property OnExit;
property OnKeyDown;
property OnKeyPress;
property OnKeyUp;
property OnCanResize;
property OnClick;
property OnConstrainedResize;
property OnDblClick;
property OnMouseDown;
property OnMouseMove;
property OnMouseUp;
property OnMouseWheel;
property OnMouseWheelDown;
property OnMouseWheelUp;
property OnResize;
end;

////////////////////////////////////
// Оголошення класу TVMRBitmap
////////////////////////////////////
type // Об'ява власних типів та структур

TVMRBitmapOption=(
  vmrbDisable,
  vmrbSrcColorKey,
  vmrbSrcRect);

TVMRBitmapOptions = set of TVMRBitmapOption;

TVMRBitmap = class // Класи
private
  FVideoWindow: TVideoWindow;
  FCanvas: TCanvas;
  FVMR9ALPHABITMAP: TVMR9ALPHABITMAP;
  FOptions: TVMRBitmapOptions;
  FBMPold: HBITMAP;
  procedure SetOptions(Options: TVMRBitmapOptions);
  procedure ResetBitmap;
  procedure SetAlpha(const Value: Single);
  procedure SetColorKey(const Value: COLORREF);
  procedure SetDest(const Value: TVMR9NormalizedRect);
  procedure SetDestBottom(const Value: Single);
  procedure SetDestLeft(const Value: Single);
  procedure SetDestRight(const Value: Single);

```

```

procedure SetDestTop(const Value: Single);
procedure SetSource(const Value: TRect);
function GetAlpha: Single;
function GetColorKey: COLORREF;
function GetDest: TVMR9NormalizedRect;
function GetDestBottom: Single;
function GetDestLeft: Single;
function GetDestRight: Single;
function GetDestTop: Single;
function GetSource: TRect;
public //зона бачення змінних та типів даних
constructor Create(VideoWindow: TVideoWindow); //Конструктор
destructor Destroy; override; //Деструктор
procedure LoadBitmap(Bitmap: TBitmap);
procedure LoadEmptyBitmap(Width, Height: Integer; PixelFormat: TPixelFormat;
    Color: TColor);
procedure Draw;
procedure DrawTo(Left, Top, Right, Bottom, Alpha: Single; doUpdate: boolean
    = false);
procedure Update;
property Canvas: TCanvas read FCanvas write FCanvas;
property Alpha: Single read GetAlpha write SetAlpha;
property Source: TRect read GetSource write SetSource;
property DestLeft : Single read GetDestLeft write SetDestLeft;
property DestTop : Single read GetDestTop write SetDestTop;
property DestRight : Single read GetDestRight write SetDestRight;
property DestBottom : Single read GetDestBottom write SetDestBottom;
property Dest: TVMR9NormalizedRect read GetDest write SetDest;
// завдання властивості типу COLORREF
property ColorKey: COLORREF read GetColorKey write SetColorKey;
property Options: TVMRBitmapOptions read FOptions write SetOptions;
end;
End.

```

**Під час написання програмного продукту були використанні бібліотеки DirectShow:**

comlite.h, errors.h, dv.h, strmif.h, mmstream.h, amstream.h,  
ddstream.h, astream.h, mpconfig.h, control.h, qnetwork.h,  
playlist.h, il21dec.h, amvideo.h, amData.h, vptype.h  
sbe.idl, wmdxva.h, vmr9.idl

КБПЗ\_2024

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування Video\_Stream.dpr**

```
// Copyrights © Кривда Олексій Сергійович, КБ-20, ЦНТУ КБПЗ
program VideoStream; // Назва програми

uses // Які бібліотеки використовуємо у модулі

    Forms,
    SysUtils,
    Unit1 in 'Unit1.pas' {Form1},
    LOG in 'LOG.pas' {Form3},
    VIDEO_GRAPH in 'VIDEO_GRAPH.pas' {Form4},
    SET_DATA in 'SET_DATA.pas' {Form2},
    SPLASH_WINDOW in 'SPLASH_WINDOW.pas' {Form7},
    About in 'About.pas' {AboutBox};

{$R *.res} /// Підключення ресурсного файлу проекту

Begin
    Form7:=TForm7.Create(Application); // Створення форми заставки
    Form7.Show; // Виведення форми заставки
    Form7.Update; // Оновлення
try
    Application.Initialize; // Ініціалізація ПЗ
    Application.CreateForm(TForm1, Form1);
    // Створення форм
        Application.CreateForm(TForm3, Form3);
        Application.CreateForm(TForm4, Form4);
        Application.CreateForm(TForm2, Form2);
        Application.CreateForm(TAboutBox, AboutBox);
    finally
        Form7.Free; // Видалення форми заставки
    end;
Application.Run; // Початок роботи розробленого ПЗ

end.
```

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - модуль статистики роботи LOG.pas**

```

// Copyrights © Кривда Олексій Сергійович, КВ-20, ЦНТУ КБПЗ
unit LOG;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type // Об'ява власних типів та структур
  TForm3 = class(TForm) // Оголошення класу
    Panel1: TPanel;
    Panel2: TPanel;
    Button1: TButton;
    Memo1: TMemo;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private секція }
  public //зона бачення змінних та типів даних
    { Public секція }
  end;

var
  Form3: TForm3;

Implementation // Реалізація

uses Unit1; // Які бібліотеки використовуємо у модулі
{$R *.dfm} // Підключення файлу ресурсів

procedure TForm3.Button1Click(Sender: TObject);
begin
  Form1.show;
  Form3.hide;
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
  Form3.Memo1.lines.add('Ok'+DateToStr(now));
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
  Form3.Memo1.lines.Clear;
end;

end.

```

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування – вікно заставки SPLASH\_WINDOW.pas**

```
// Copyrights © Кривда Олексій Сергійович, КБ-20, ЦНТУ КБПЗ
unit SPLASH_WINDOW;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls;

type // Об'ява власних типів та структур

  TForm7 = class(TForm) // Оголошення класу
    Image1: TImage;
    ScrollBox1: TScrollBox;
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  Form7: TForm7;

Implementation // Реалізація

{$R *.dfm} // Підключення файлу ресурсів

end.
```

**Файл програмного забезпечення, розробленого у ході виконання бакалаврського проектування - авторське право About.pas**

```
// Copyrights © Кривда Олексій Сергійович, КВ-20, ЦНТУ КБПЗ
unit About;

interface // Інтерфейсна частина

uses // Які бібліотеки використовуємо у модулі
Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
Buttons, ExtCtrls;

type // Об'ява власних типів та структур
  TAboutBox = class(TForm) // Оголошення класу

    Panell: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    OKButton: TButton;
    procedure OKButtonClick(Sender: TObject);
  private
    { Private секція }
  public
    { Public секція }
  end;

var
  AboutBox: TAboutBox;

implementation // Реалізація

uses Unit1; // Які бібліотеки використовуємо у модулі
{$R *.dfm} // Підключення файлу ресурсів

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  Form1.show;
  AboutBox.hide;
end;

end.
```