

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

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ____ ” _____ 2021 р.

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

**“Дослідження та програмна реалізація системи комплексного
відеонагляду банківської установи”**

Виконав здобувач вищої освіти
II курсу, групи КН-20М-1,4
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»

Заїкін І.І.

« ____ » _____ 2021 р.

Керівник проекту
доктор технічних наук, професор

Олексій СМІРНОВ

« ____ » _____ 2021 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

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

Олексій СМІРНОВ
« 6 » вересня 2021 року

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

Заїкіну Івану Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи комплексного відеонагляду банківської установи

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

затверджені наказом вищого навчального закладу № 39-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту 10.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи комплексного відеонагляду банківської установи

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

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

6. Консультанти розділів роботи

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

7. Дата видачі завдання « 6 » вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Заїкін І.І. Дослідження та програмна реалізація системи комплексного відеонагляду банківської установи. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

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

Метою розробки є дослідження та програмна реалізація системи комплексного відеонагляду банківської установи.

Об'єктом дослідження є процес комплексного відеонагляду банківської установи.

Предметом дослідження є методи комплексного відеонагляду банківської установи.

Методи дослідження базуються на методах технічного захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи комплексного відеонагляду банківської установи.

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

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

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

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: Комп'ютерні науки, захисту доступу, обробка графічних даних

ABSTRACT

Zaikin I.I. Research and software implementation of a comprehensive video surveillance system of a banking institution. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work on the second (master's) level of higher education the software which is intended for system of complex video surveillance of banking institution is developed.

The purpose of the development is research and software implementation of a comprehensive video surveillance system of a banking institution.

The object of the study is the process of comprehensive video surveillance of a banking institution.

The subject of the study are the methods of integrated video surveillance of a banking institution.

Research methods are based on methods of technical protection of information, methods of mathematical statistics, methods of software development.

The result is a software implementation of a comprehensive video surveillance system of a banking institution.

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

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

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

The program is developed in the environment of RAD Studio Delphi 10.4.

Keywords: Computer science, access protection, graphic data processing

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	20
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи.....	27
3.2 Розробка структурної схеми.....	40
3.3 Розробка функціональної схеми.....	45
3.4 Розробка діаграми процесів.....	46
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	48
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	48
4.2 Захист розробленого програмного забезпечення	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	58
6 НАУКОВА НОВИЗНА	61

ВКРМ-122.21.0002.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Заїкін І.І.			Дослідження та програмна реалізація системи комплексного відеонагляду банківської установи	Лім.	Аркуш	Аркушів
Перев.		Смірнов О.А.				М	1	101
Н.контр.		Гермак В.С.			ЦНТУ КН-20М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	62
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	62
7.2 Розрахунок трудомісткості розробки програмної продукції	64
7.3 Визначення чисельності виконавців і планового фонду зарплати	66
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	71
7.5 Визначення собівартості розробки та ціни програмної продукції.	75
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	78
7.7 Визначення експлуатаційних витрат.....	79
7.8 Визначення економічної ефективності програмної продукції.....	80
7.9 Висновок.	82
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	83
8.1 Вступ	83
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	84
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	85
8.4 Розробка заходів з умов поліпшення охорони праці.....	88
8.5 Розрахункова частина	90
8.6 Висновки до розділу.....	91
9 ОСНОВНІ ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- АРП – автоматичне регулювання посилення
- АТМ – автоматичний телевізійний моніторинг
- ЕПТ – електронно-променева трубка
- ІБ – інформаційна безпека
- ІС – інформаційна система
- НСД – несанкціонований доступ
- ПЗ – програмне забезпечення
- ПЗЗ – прилади із зарядовим зв'язком
- СКД – системи контролю даних
- СЦК – система централізованого керування
- ТВЛ – телевізійні лінії
- ІР – Internet Protocol. Протокол міжмережної взаємодії
- PIN-код – персональний ідентифікаційний код
- TCP – Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP. Встановлює зв'язок між двома ПК й організує обмін даними в дуплексному режимі
- TCP/IP – Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Фінансову систему неможливо представити без банківських установ. Коло задач, які вирішують банки постійно збільшується: починаючи від обігу фінансових потоків, закінчуючи різними видами кредитування. В усі часи проблемам забезпечення безпеки банківських установ надавали велике значення. При цьому при побудові сучасної комплексної системи безпеки банку приходиться враховувати велику кількість чинників, починаючи від законодавчих питань охорони банку, інформаційної безпеки конфіденційних даних, які передаються по каналам зв'язку банківських мереж, і закінчуючи фізичним захистом банківських установ [1-5]. В даному випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти увагу сконцентруємо на останньому чиннику. Розглянемо такий аспект фізичних засобів, як технічне забезпечення системи охорони. Система комплексного відеонагляду банківської установи повинна вирішувати наступні задачі:

- побудову комплексної системи безпеки банку;
- одержання повної картини ситуації на об'єкті;
- забезпечення безпеки банкоматів;
- реєстрацію й контроль доступу співробітників і відвідувачів;
- контроль транспортних засобів на території банку.

Усі ці задачі можуть вирішуватися за рахунок широкого застосування сучасних досягнень мікроконтролерної техніки, охоронного телебачення, захищеної передачі даних які отримані з вищеперерахованих пристроїв [1-7].

Таким чином сучасна технічна система комплексного відеонагляду банківської установи представляє собою програмно-апаратний комплекс, в якому застосовані усі останні напрацювання в області пристроїв забезпечення захисту банківської установи та інформаційно-програмного забезпечення даних які отримуються з цих пристроїв, їх передачі, зберігання та обробки.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи комплексного відеонагляду банківської установи.

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

- Огляд існуючих систем комплексного відеонагляду банківської установи.
- Дослідження системи комплексного відеонагляду банківської установи.
- Програмна реалізація системи комплексного відеонагляду банківської установи.

Об'єктом дослідження є процес комплексного відеонагляду банківської установи.

Предметом дослідження є методи комплексного відеонагляду банківської установи.

Методи дослідження базуються на методах технічного захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод комплексного відеонагляду банківської установи.
- Розроблено вітчизняний продукт комплексного відеонагляду банківської установи, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі комплексного відеонагляду банківської установи.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система комплексного відеонагляду банківської установи – комплекс програмно-технічних засобів для забезпечення безпеки й захисту:

- персоналу банку: керівників, співробітників, що мають безпосередній доступ до фінансів, валюти, цінностям, сховищам, до відомостей, що є банківською й комерційною таємницею, працівники зовнішньоекономічних служб і ін.;

- фінансових засобів, валюти, коштовностей;

- інформаційних ресурсів з обмеженим доступом, що становлять службову й комерційну таємницю, а також іншої конфіденційної інформації на паперових, магнітних, оптичних носіях;

- матеріальних засобів: окремих будинків, споруджень, приміщень банку, технічного устаткування, транспорту й ін.

Всі об'єкти, у відношенні яких можуть бути здійснені погрози безпеки або протиправні зазіхання, мають різну потенційну уразливість із погляду можливого матеріального або морального збитку.

Рішення – це система безпеки для запобігання потенційних погроз:

- для персоналу й співробітників банку;

- для приміщень:

– вибухи;

– підпали;

– нападу, вторгнення, захват, блокування;

– ушкодження входних дверей, ґрат, огорожень, вітрин, меблів, а також транспортних засобів: особистих і службових;

– технологічні аварії, пожежі, стихійні лиха й т.п.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- для фінансових ресурсів:
 - підроблені платіжні документи й пластикові карти;
 - розкрадання фінансових засобів з кас і інкасаторських машин.
- для інформаційних ресурсів:
 - несанкціонований доступ до охоронюваних відомостей;
 - знищення або псування носіїв стратегічно важливої інформації.

1.2 Область застосування

На основі аналізу перерахованих погроз можна виділити основні завдання, що є при впровадженні системи безпеки банку, і варіанти їхнього рішення:

Завдання 1: Побудова комплексної системи безпеки банку.

Рішення. Використання програмно-апаратного комплексу як платформи системи безпеки:

- інтеграція підсистем відеоспостереження, контролю доступу, охоронно-пожежної сигналізації, комплексу програмно-технічних засобів, включаючи устаткування банкоматів, різних датчиків;
- створення єдиної системи безпеки, у тому числі, територіально розподіленої, для всіх відділень і філій банку, мережі банкоматів;
- забезпечення віддаленого моніторингу, доступу до архівних даних, дистанційного керування системою, одержання звітів про стан всіх компонентів системи й зареєстрованих ними подій;
- налаштування логіки роботи системи і її єдиної реакції при реєстрації певних подій;
- створення єдиного графічного користувальницького інтерфейсу для зручної роботи.

Завдання 2: Одержання повної картини ситуації на об'єкті.

Рішення. Організація відеоспостереження:

- в операційних залах;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- у кімнатах переговорів;
- у приймальнях;
- у грошових сховищах;
- у зоні банкоматів;
- по зовнішньому периметру території.

Необхідність забезпечення високого рівня безпеки визначає наступні варіанти установки відеокамер:

- відкрито: по периметру будинку, у зоні центрального входу й ін.
- сховано: у кімнатах переговорів, окремих офісах і ін.

Організація цілодобового багатоканального відеоспостереження в різних приміщеннях банку забезпечить постійний моніторинг обстановки на охоронюваному об'єкті, необхідний для оперативного реагування у випадку виникнення потенційно небезпечних, позаштатних ситуацій.

Функціональні можливості й характеристики систем відеоспостереження дозволяють забезпечити високий рівень безпеки банку, з огляду на його особливості як об'єкта охорони, серед них:

- багатоканальний відеозапис;
- налаштування режиму відеозапису: безупинно, по команді оператора, по детектуванню рухів, гарячий запис, передтревожний й післятревожний запис;
- індивідуальні налаштування для кожної відеокамери: розрішення, кольоровість, контрастність зображення;
- керування поворотними камерами;
- детектор засвітки (перекриття) камери;
- цифрове збільшення зображення;
- віддалений доступ до даних;
- програмний детектор руху;
- налаштування зон забороненого/дозволеного руху;
- використання продуктивних алгоритмів компресії;
- миттєвий пошук відеозапису в архіві;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- експорт відеозображення у формати AVI, JPEG;
- синхронізація відео- і аудіоданих;
- формування відеоархіву великого обсягу й ін.

Для організації відеоспостереження використовуються:

- системи відеоспостереження (залежно від розміру охоронюваного об'єкта) для відеомоніторингу, у т.ч. віддаленого, і формування відеоархіву;
- програмно-апаратний комплекс для забезпечення відеоспостереження й інтеграції системи контролю доступу, охоронно-пожежної сигналізації й інших компонентів.

Завдання 3: Забезпечення безпеки банкоматів.

Рішення. Використання технологій передачі даних і інтеграції:

Сприяє захисту:

- банку від шахрайства при виконанні операцій;
- банкомату від дій зловмисників з метою оволодіння наявними коштами, що зберігаються в банкомату, і актів вандалізму;
- законних власників кредитних карт від можливих нападів зловмисників;
- інкасаторів при завантаженні/розвантаженні банкоматів наявними коштами.

Рішення ефективно для організації відеоконтролю великомасштабної мережі банкоматів, що складається з локальних систем і системи централізованого керування:

- забезпечується централізований моніторинг роботи мережі банкоматів у режимі реального часу;
- формується централізований відеоархів, єдиний протокол роботи всіх банкоматів: відеозапис, що надходить від відеокамер, повідомлення від банкомату, сигнали тривоги при виникненні позаштатних ситуацій передаються в центр відеоконтролю по лініях зв'язку, що підтримує протокол TCP/IP;
- здійснюється централізоване керування системою відеоспостереження мережі банкоматів з виділеного сервера.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Завдання 4: Реєстрація й контроль доступу співробітників і відвідувачів.

Рішення. Інтеграція в систему відеоспостереження модуля захвата й розпізнавання осіб дозволить здійснювати:

- автоматичну безконтактну реєстрацію всіх людей, що пройшли через рубіж контролю
- ідентифікацію людей – порівняння із зображеннями з бази даних, у тому числі, для дозволу/заборони по "чорному" або "білому" списку
- підтвердження повноважень особистості на прохід
- формування бази даних співробітників і відвідувачів банку зі збереженням інформації про дату, час, напрямок проходу, відеокадру із зображенням його особи й ін.
- забезпечення оповіщення операторів (телефон, e-mail, sms) за результатами розпізнавання й ідентифікації
- забезпечення реакції системи контролю доступу, іншого охоронного устаткування, виконавчих пристроїв (блокування дверей, включення сирени) за результатами розпізнавання й ідентифікації

Завдання 5: Контроль транспортних засобів на території банку.

Рішення. Інтеграція в систему відеоспостереження модуля розпізнавання автомобільних номерів забезпечить:

- реєстрацію транспорту що в'їжджає/виїжджає, із занесенням у базу даних установлених параметрів (номер, марка, дата, час в'їзду транспортного засобу, ФІО водія й т.п.), збереження відповідного відеозображення й пошук інформації із заданих параметрів;
- диференційований в'їзд для використання автостоянки співробітниками банку й заборони проїзду "чужих" транспортних засобів;
- необхідну реакцію виконавчих пристроїв (автоматичних воріт, шлагбаумів) за результатами розпізнавання номерів і їхнього зіставлення з інформацією з бази даних;
- безпека автомобілів на стоянці банку;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- відеоконтроль у режимі реального часу виїзду й в'їзду інкасаторських автомобілів, дотримання співробітниками служби інкасації встановленого порядку й ін.

- зіставлення даних про факти в'їзду й виїзду транспортних засобів, у першу чергу, інкасаторських автомобілів.

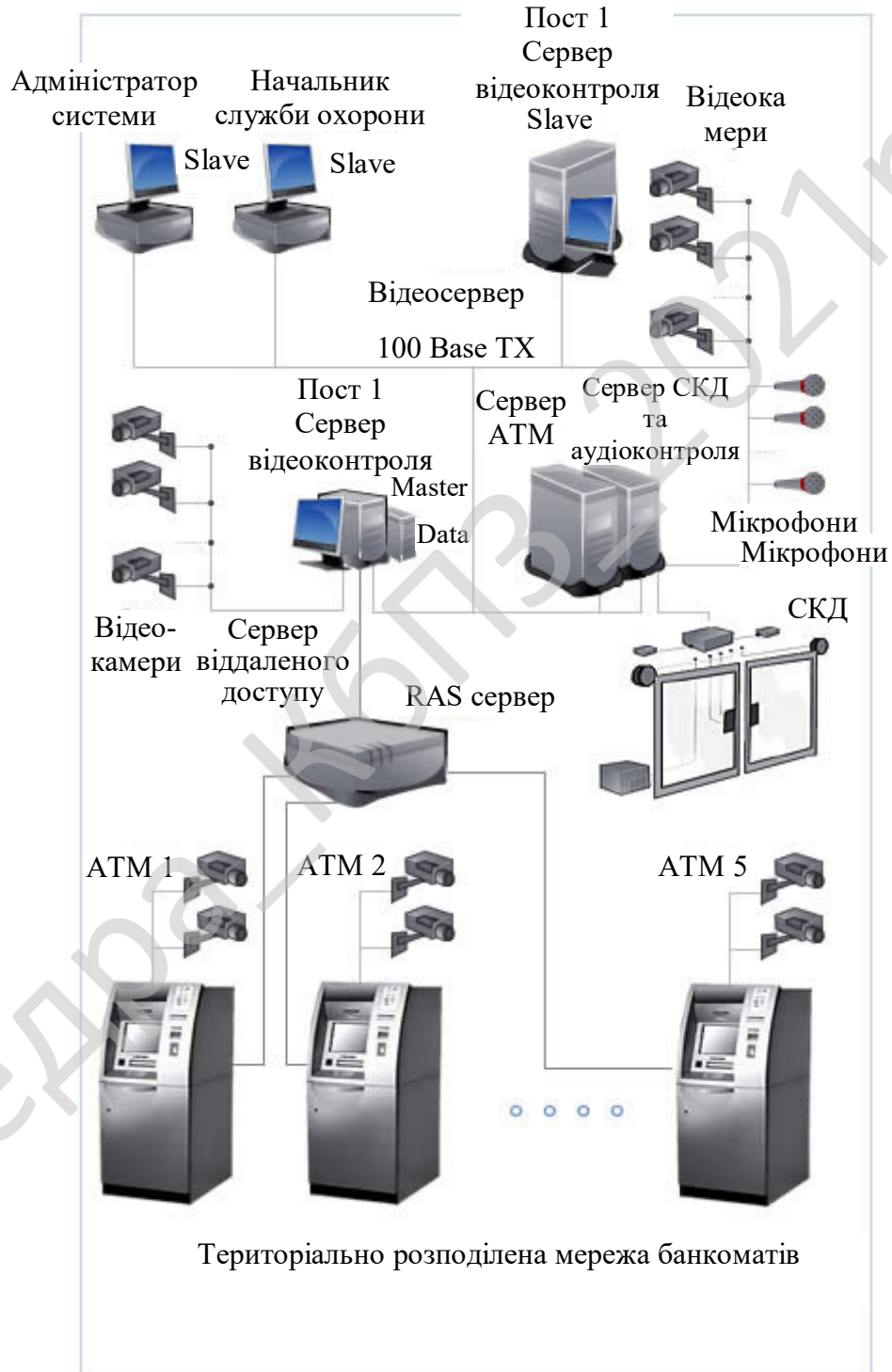


Рисунок 1.1 – Типова схема системи комплексного відеонагляду банківської установи

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

Кафедра _КБПЗ_ 2021 рік

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Цифрова система відеоспостереження відділень банку побудована на устаткуванні провідних світових виробників. Зокрема, застосовані цифрові відеореєстратори-мультиплексори, що дозволяють службі безпеки банку управляти роботою телекамер у приміщеннях банку й вуличних камер, а також записувати всю інформацію з телекамер у цифровому форматі.

При побудові системи пожежної безпеки банку застосований апаратно-програмний комплекс Simplex відомої американської компанії SimplexGrinnell у складі адресної системи автоматичної пожежної сигналізації, голосової системи оповіщення про пожежу й евакуацію людей і системи спринклерного пожежогасіння. Інтеграція пожежної сигналізації й системи оповіщення в єдину систему безпеки здійснена на апаратно-програмному рівні.

Для обмеження доступу в службові приміщення відділень банку застосовані традиційні для банку системи контролю доступу й охоронної сигналізації. Цілодобовий доступ клієнтів у приміщення з банкоматами забезпечується системою, що дозволяє використовувати для входу банківські карти VISA. Для обліку робочого часу персоналу використаний спеціальний модуль системи Simplex.

Функціональні можливості інтегрованої системи безпеки банку:

- Цілодобове відеоспостереження в службових приміщеннях банку, відеоконтроль за присутніми місцями банку й за прилягаючими територіями, із записом відео на пристрої системи безпеки.
- Архівування відеозаписів по приміщеннях банку з можливістю покадрового відтворення записів тривоги, у тому числі по даті, часу доби, по камері й т.п.
- Відображення відеоінформації з камер на моніторах системи безпеки в реальному часі.
- Архівування відеоінформації по тривожних подіях.
- Авторизація співробітників банку в системі контролю доступу при їхніх переміщеннях по службових приміщеннях.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Облік робочого часу співробітників відділень банку.
- Надання цілодобового доступу власників карток VISA у приміщення банку з банкоматами й забезпечення їхньої безпеки.
- Фіксація й оповіщення служби безпеки банку про виникнення задимлення або пожежі на території банку з адресою тривожного приміщення.
- Автоматичне включення системи оповіщення персоналу про пожежу й системи пожежогасіння в приміщеннях виникнення пожежі.
- Голосове оповіщення персоналу банку про пожежу й вказівка безпечних шляхів їхньої евакуації.
- Протоколювання й автоматичний запис на пристрої системи безпеки всіх тривожних подій, що відбулися на території відділення банку.

IP-відеоспостереження на базі TRASSIR

Компанія DSSL однією з перших російських компаній запропонувала мережні рішення на ринку СНД, поставивши мету зробити IP-відеоспостереження доступним за ціною й популярним серед широкого кола фахівців. Система цифрового відеоспостереження й аудіозаписи TRASSIR має у своєму складі лінійку пристроїв для мережного відеоспостереження.

Достоїнства IP-відеоспостереження

IP-відеоспостереження крім всіх достоїнств цифрового відеоспостереження має ряд власних додаткових переваг:

- швидке підключення й налаштування устаткування;
- простота й економічність монтажу устаткування;
- цифрова передача відеосигналу по проводам від відеокамер;
- впровадження на базі існуючої інфраструктури: відеокамери й мережне устаткування;
- використання будь-якого типу устаткування на базі протоколу TCP/IP;
- мобільне застосування: простої підключення до ноутбука.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Таблиця 2.1 – Порівняльна характеристика систем IP-відеоспостереження

	TRASSIR Lanser- 6M	TRASSIR Lanser- Real	TRASSIR Lanser-4M	TRASSIR Lanser- 4Real	TRASSIR Lanser- 4HDD
Каналів відео	6	1	4	4	4
Каналів аудіо	0	1	4	4	4
Розрішення/ швидкість запису	704x288 (2Fps на канал або 12 сумарно)	704x576 (25Fps)	704x576 (20Fps) – 2 канали 352x288 (25Fps) – 4 канали 704x576 (4Fps) – 4 канали	704x288 (25Fps) – 4 канали 704x576 (25Fps) – 2 канали	704x576 – (6Fps) 4 канала 704x288 (10Fps) – 4 канали 352x288 – (25Fps) 4 канала
Розмір кадру	20-50кб	5-15кб	3-10кб	3-10кб	3-10кб
Потік на канал	4, 5- 6Mbit на сервер	32- 8000kbit	32-2000kbit	32-2000kbit	32- 2000kbit
Формат стиску	Wavelet	H.264	H.264	H.264	H.264
Тривожні входи\виходи	6/2	1/1	4/2	4/2	4/2
Макс.пристр. на один ПК	10-12	64-128	10-12	10-12	Необмежен о (CMS режим)
АЦП	Philips 9bit	TI 9bit Ethernet 10/100Mbit	Techwell 10bit	Philips 9bit	Philips 9bit
Підключення	Ethernet 10/100Mbit	Ethernet 10/100Mbit	Ethernet 10/100Mbit	Ethernet 10/100Mbit	Ethernet 10/100Mbit
Керування PTZ	Немає	Так	Так	Так	Так
Дуплексне аудіо	Немає	Так	Так	Так	Немає
Інше		RS-232	RS-232	RS-232	EIDE 3,5, RS-232

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

18

Застосування: Об'єкти, що вимагають автономного відеозапису з функціями центрального моніторингу: банкомати, автозаправні станції, вузли зв'язку, ретранслятори, насосні станції, стільникові станції, інші територіально рознесені об'єкти.

Відеореєстратор CasinoGUARD

CasinoGUARD – відеореєстратор з високою продуктивністю й підвищеною відказостійкістю. Дозволяє записувати до 32 каналів відеосинхронно з аудіо зі швидкістю 25 кадрів у секунду й розрішенням 704x576. Відеореєстратор розроблений спеціально для об'єктів ігрового й банківського секторів і є єдиним подібним пристроєм у своєму класі на українському ринку відеоспостереження.

Відеореєстратор Nexus

Nexus – відеореєстратор з підвищеною відказостійкістю в економічному виконанні. Дозволяє записувати до 64 каналів відеосинхронно з аудіо зі швидкістю 6 кадрів у секунду й дозволом 704x576, а вивід відеона екран здійснюється зі швидкістю 25 кадрів у секунду на кожний канал. Дане рішення можна використовувати для відеоспостереження в службових приміщеннях і для огляду загальних планів приміщень і основних проходів.

Спеціалізовані відеосервера BankMonitorer

- максимально можливий розрішення відеозапису;
- до 32-х каналів відеов реальному часі синхронно зі звуком;
- інтеграція із системами доступу й охоронно-пожежною сигналізацією;
- архів 5 і більше Тб;
- підвищені вимоги до безпеки зберігання даних;
- шифрування даних;
- шифрування каналів передачі даних.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

2.2 Обґрунтування вибору засобів для побудови системи 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 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.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– 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 швидкодією. 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.

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 звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

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 TМето 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.

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 випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

забезпечення, яке призначено для системи комплексного відеонагляду банківської установи.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Головною метою системи безпеки є забезпечення стійкого функціонування банку й запобігання погроз його безпеки, захист законних інтересів кредитної організації від протиправних зазіхань, охорона життя й здоров'я персоналу, недопущення розкрадання фінансових і матеріально-технічних засобів, знищення майна й цінностей, розголошення, втрати, витоку, перекручування й знищення службової інформації, порушення роботи технічних засобів, забезпечення виробничої діяльності, включаючи й засобу інформатизації.

Завданнями системи безпеки є:

- прогнозування й своєчасне виявлення й усунення погроз безпеки персоналу й ресурсам банку;
- причин і умов, що сприяють нанесенню фінансового, матеріального й морального збитку, порушенню його нормального функціонування й розвитку;
- віднесення інформації до категорії обмеженого доступу (державній, службовій, банківській і комерційній таємницям, іншої конфіденційної інформації, що підлягає захисту від неправомірного використання), а інших ресурсів – до різних рівнів уразливості (небезпеки) і підметів збереженню;
- створення механізму й умов оперативного реагування на погрози безпеки й прояв негативних тенденцій у функціонуванні банку;
- ефективне припинення погроз персоналу й зазіхань на ресурси на основі правових, організаційних і інженерно-технічних мір і засобів забезпечення безпеки;
- створення умов для максимально можливого відшкодування й локалізації збитку, що наноситься неправомірними діями фізичних і юридичних

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

осіб, ослаблення негативного впливу наслідків порушення безпеки на досягнення стратегічних цілей банку.

Технічне забезпечення безпеки базується:

- на системі стандартизації й уніфікації;
- на системі ліцензування діяльності;
- на системах сертифікації засобів захисту;
- на системі сертифікації ТЗ і ПЗ об'єктів інформатизації;
- на системі атестації захищених об'єктів інформатизацією.

Основними складовими забезпечення безпеки ресурсів системи комплексного відеонагляду банківської установи є:

- система фізичного захисту (безпеки) матеріальних об'єктів і фінансових ресурсів;
- система безпеки інформаційних ресурсів.

Система фізичного захисту (безпеки) матеріальних об'єктів і фінансових ресурсів передбачає:

- систему інженерно-технічних і організаційних мір охорони;
- систему регулювання доступу;
- систему мір (режиму) схоронності й контроль імовірних каналів витоку інформації;
- систему мер повернення матеріальних цінностей (або компенсації).

Система охоронних мір передбачає:

- багаторубіжність побудови охорони (території, будинку, приміщення) по наростаючій до найціннішої конкретності, що оберігається;
- комплексне застосування сучасних технічних засобів охорони, виявлення, спостереження, збору й обробки інформації, що забезпечують достовірне відображення й об'єктивне документування подій;
- надійний інженерно-технічний захист імовірних шляхів несанкціонованого вторгнення в охоронювані межі;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- стійку (дубльовану) систему зв'язку й керування всіх взаємодіючих в охороні структур;
- високу підготовку й готовність основних і резервних сил охорони до оперативної протидії порушника;
- самоохорону персоналу.

Опис реалізації системи комплексного відеонагляду банківської установи

Для невеликого відділення банку досить не більше чотирьох – п'яти камер. Використовуючи монітор з вбудованим комутатором і вдало розташувавши камери, забезпечимо цілодобове спостереження за охоронюваною територією. Камери можуть розташовуватися усередині приміщення на поворотних пристроях. При цьому в денний час вони можуть використовуватися для контролю в банківському залі, а ввечері й уночі для контролю охоронюваної території. Кількість одночасно відображуваних камер повинне бути обмежено. При збільшенні кількості моніторів операторові важко стежити за всіма змінами. У багатокамерних системах використовуються додаткові пристрої. До додаткових пристроїв відносяться детектори руху, які аналізують зміни зображення, наприклад, переміщення будь-якого предмета в поле зору камери й сигналізують операторові про це.

Для дистанційного керування камерами використовуються поворотні пристрої. Вони дозволяють збільшити огляд камери за допомогою її повороту у двох площинах. Керування поворотними пристроями здійснюється джойстиком.

Для одночасного одержання декількох зображень (до 16) на екрані одного монітора використовуються квадратори («дільники екрана»). Квадратори перетворюють сигнали від декількох відеокамер у зображення, що відображається на одному моніторі. При цьому зображення від будь-якої камери можна оперативно розгорнути на весь екран.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Для послідовного виводу на екран зображення від декількох камер у системах спостереження використовуються мультиплексори (комутатори). У режимі перегляду вони послідовно підключають камери до монітора.

Для оперативної роботи оператор має можливість вивести на екран будь-яке зображення або виключити будь-яку камеру. Періодичність перемикання й час спостереження зображення задається для всіх камер одночасово. На великих об'єктах число камер може становити кілька десятків. Для підвищення ефективності роботи оператора використовують матричні комутатори. Вони дозволяють створити гнучку й нарощувану систему безпеки, у яку можуть входити не тільки компоненти телевізійних систем, але й системи сигналізації й контролю доступу.

Запис відеозображення може здійснюватися на спеціалізовані відеомагнітофони в традиційних системах або в цифровій формі за допомогою комп'ютера. Сучасні тенденції науково-технічного прогресу привели до того, що в основному використовуються цифрові носії, тобто дані по захищеному IP-каналю передаються в базу даних, де зберігаються, і звідки дістаються якщо буде потреба. Всі пристрої поєднуються в систему, що забезпечує можливість оперативного спостереження. Керування системами спостереження залежно від їхньої складності й обстановки на об'єкті може бути автоматичним або ручним.

Комп'ютерні системи спостереження володіють рядом особливостей, які в різних ситуаціях можуть грати як позитивну, так і негативну роль. Перерозподіл функцій між програмними й апаратними засобами приводить до того, що комп'ютерні системи не завжди можуть забезпечити швидке перемикання режимів. Крім того, підвищуються вимоги до оператора – уміння працювати з комп'ютером і графічним інтерфейсом.

Опис елементів системи безпеки банку

Телевізійна камера.

Якість зображення визначається, насамперед, телевізійною камерою. Вона являє собою закінчений пристрій, що будучи підключеним до відеовходу

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

монітора або телевізора дозволяє спостерігати зображення на екрані на значній відстані від об'єкта зйомки. У цей час випускаються відеокамери для систем телевізійного спостереження (включаючи модифікації), що відрізняються:

- характером зображення (чорно-біле або кольорове);
- чіткістю зображення;
- світлочутливістю (мінімальною робочою освітленістю об'єкта зйомки);
- можливістю цифрової обробки відеосигналу;
- припустимими кліматичними умовами роботи;
- напругою живлення.

З метою забезпечення якісної роботи в умовах змінної яскравості зображення й різних рівнів фонових засвіток сучасні телекамери, для систем телевізійного спостереження, оснащуються підсистемами компенсації цих впливів. З метою збільшення сектора огляду, телевізійні камери встановлюють на поворотні пристрої з горизонтальним або з горизонтальним і вертикальним скануванням. При повороті камери варто враховувати можливі реакції систем компенсації зовнішніх впливів (засвітка, вплив імпульсних джерел штучного висвітлення й т.д.). При установці на вулиці, телекамери містяться в спеціальні захисні корпуси.

Відеомонітор

Другим важливим елементом систем відеоспостереження є відеомонітор. Він повинен забезпечувати високу довгострокову стабільність і не вимагати регулярного калібрування. Надійність також залежить від того, на скільки оптимальні схемні рішення, міцна й зручна механічна конструкція. Технологія виробництва віщальних відеомоніторів за останні роки перетерпіла істотні зміни. У складі апаратури обробки відеоінформації звичайно використовуються два основних типи пристроїв: світчери й компресори. На додаток до основних пристроїв обробки широко застосовуються різні допоміжні пристрої:

- кабельні підсилювачі – для компенсації втрат у кабелі при передачі відеосигналу на відстань до 2 км;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– розгалужувачи, що дозволяють до одному тілі камері підключати кілька моніторів, відеомагнітофонів і т.п.;

– генератори допоміжної текстової інформації (дати, часу й т.п.).

Функціональні системи відеокамер

Основу будь-якої системи телевізійного спостереження становлять телекамери. У конструкції відеокамери можна виділити наступні основні функціональні системи:

- перетворювач світло-сигнал;
- синхронізації;
- автоматичного регулювання посилення;
- електронний затвор;
- автоматичної установки балансу чорного;
- гамма-корекції;
- зйомки при низьких рівнях освітленості;
- об'єktiv з автоматичною діафрагмою.

Функція зйомки при низьких рівнях освітленості (LOLUX) чудова тим, що дозволяє знімати майже без висвітлення. При цьому можна одержати прекрасне зображення з гарним колірним балансом без збільшення рівня шуму.

Перетворювачі «світло-сигнал»

Найважливішим елементом конструкції відеокамери є перетворювач «світло-сигнал», що забезпечує кодування зображення, що знімається, у формі електричних сигналів. Перетворювачі світло-сигнал являють собою або передавальної електронно-променевої ТБ трубки (ЕПТ), або твердотільні матриці – так звані «прилади із зарядовим зв'язком» (ПЗЗ). У сучасних відеокамерах, як правило, застосовуються матриці ПЗЗ, що забезпечують більшу надійність роботи при досить високих параметрах. Число рядків матриці приймає значення від 380 до 900.

Для камер на ПЗЗ, на відміну від трубкових аналогів, характерно також відсутність післязображень (інерційності мішені), продовжень, що тягнуться за

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

об'єктами, які рухаються, у зображенні, не говорячи вже про пропалювання фотопровідного шару мішені. Причому зазначені параметри не залежать від строку експлуатації матриць ПЗЗ. У відеокамерах застосовуються 2/3", 1/2", 1/3", 1/4" і 1/6" прилади із зарядовим зв'язком (ПЗЗ). Число пікселів (піксель – один елемент ПЗЗ) у ПЗЗ може бути від 300 до 1000. Кількість елементів матриці забезпечує горизонтальний дозвіл зображення залежно від моделі 300-600 телевізійних ліній (ТВЛ). Успіх мініатюрних відеокамер обумовлений високою надійністю і якістю перетворювачів на приладах із зарядовим зв'язком (ПЗЗ)

Пристрій синхронізації

Пристрій синхронізації забезпечує часове узгодження роботи всіх систем і блоків камери. Синхронізація відеокамер може здійснюватися від внутрішнього або зовнішнього генератора. Зовнішня синхронізація використовується в багатокамерних системах для одержання немиготливого перемикачання. При спільному використанні камер із внутрішньою синхронізацією, вони комутируються пристроями, що містять пам'ять на кадр.

Об'єктиви відеокамер

Об'єктиви до камер відрізняються величиною фокусної відстані, світлосилою, характером створюваного оптичного зображення. При зйомці з однієї й тої ж точки об'єктивами з різними фокусними відстанями масштаб зображення змінюється прямо пропорційно величині фокусної відстані. Короткофокусний об'єктив навіть при невеликому діафрагмуванні має велику глибину різкості. Довгофокусний об'єктив навіть при зйомці вилучених об'єктів має обмежену глибину різкості. Об'єктив камери вибирається відповідно до призначення камери. Для максимального огляду вибирають широкоугольні об'єктиви з фокусною відстанню порядку 3,5 мм. При цьому кут зору камери буде близько 90°. Довгофокусні об'єктиви з фокусною відстанню 12 мм і кутом зору 30° використовують при спостереженні периметра об'єкта. Для використання в умовах штучного висвітлення необхідна можливість відключення електронного затвора й автоматичного регулювання посилення камери.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Об'єктив зі змінною фокусною відстанню

Для забезпечення ефекту збільшення зображення використовуються об'єктиви із трансфокатором, спеціальні телекамери з електронним трансфокатором, або цифрову апаратуру збільшення/зменшення зображення (відеопроцесори). Об'єктиви відеокамер, що мають змінну фокусну відстань, називаються « вари-об'єктиви». Вони дозволяють здійснити плавну зміну масштабу зображення (робити «наїзд»). Масштаб змінюється вручну або за допомогою електропривода. При цьому зберігається фокусування зображення. Застосування трансфокаторів дозволяє «наблизити» зображення від 5 до 20 разів, що дозволяє розглянути навіть сильно віддалені об'єкти.

Об'єктив з автоматичною діафрагмою

Об'єктив з автоматичною діафрагмою встановлює розмір отвору діафрагми, що забезпечує оптимальну інтенсивність світлового потоку, що проходить через об'єктив і попадає на мішень перетворювача «світло-сигнал». Використання об'єктивів з автоматичною діафрагмою дозволяє одержувати якісне зображення як при яскравому сонці, так і при місячному світлі.

Додаткові можливості й сервісні пристрої відеокамер

Автоматичне регулювання посилення

Режим автоматичного регулювання посилення дозволяє робити безперервну зйомку при всіх рівнях освітленості без необхідності перемикати посилення або застосовувати відповідні фільтри й має також таку чудову властивість, як пріоритетність апертури. Вона полягає в тому, що після того, як вручну встановлена діафрагма, для одержання бажаної глибини різкості, система АРП автоматично встановлює необхідний рівень відеосигналу.

Електронний затвор

Структура матриці типу HAD дозволила застосувати електронний затвор з функцією змінного часу експозиції. Це дає можливість знімати передавальної ТВ камерою швидкоплинні динамічні процеси й об'єкти за час другої частини кожного поля, а це і є період відкривання електронного затвора.

						ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			34

Автоматична установка балансу білого

Ця функція корисна, коли оператор не має часу для установки камери в режим зйомки. Автоматична установка балансу білого полягає в підборі посилення в каналах червоного і синього кольору (у кольорових відеокамерах) стосовно посилення зеленого.

Гамма-корекція

Гамма-корекція – розтягування відеосигналу в області чорного. У деяких моделях відеокамер є схема, що дозволяє збільшити число градацій у передачі півтонів чорного й сірого кольору. Дія її фактично обратна дії схеми стиску контрастності, що підвищує й поглиблює контрастність півтонів у зображенні.

Монітори для систем телевізійного спостереження

Відеомонітор повинен забезпечувати строгу відповідність зображення подаваному відеосигналу. Параметри, що визначають якість зображення:

- чіткість;
- фокусування;
- відтворення кольору;
- відомість;
- геометричні перекручування.

Додаткові пристрої систем телевізійного спостереження

Спеціалізовані цифрові відеомагнітофони

Для запису зображення в системах телевізійного контролю служать спеціалізовані відеомагнітофони. Вони ведуть безперервний цифровий запис зображень у базу даних. Функціональні можливості відеомагнітофонів:

- запис і відтворення чорно-білого або кольорового зображення;
- програмування режимів запису;
- вивід на екран часу й дати;
- здійснення запису по таймері або по зовнішньому сигналі;
- програмування таймера з установкою щоденного початку й закінчення запису, а також установка режиму запису на тиждень;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

- спеціальні режими відтворення (покадрове відтворення, пауза, швидкісний пошук вперед та назад);
- стоп-кадр;
- видача сигналів синхронізації на зовнішні пристрої;
- програмування режимів роботи при спрацьовуванні сигналізації;
- реєстрація часу аварійного відключення живлення; – зберігання інформації в енергонезалежній пам'яті.

У багатокамерних системах відеоспостереження відеомагнітофони використовуються разом з відеокомпресорами й мультиплексорами.

Відеокомпресори

Відеокомпресор (квадратор) – пристрій, що дозволяє на екрані монітора одночасно спостерігати в режимі реального часу зображення від декількох відеокамер і записувати його на відеомагнітофон. Наявність входу тривоги (ALARM-вхід) дозволяє підключити до відеокомпресора систему сигналізації, щоб при її спрацьовуванні автоматично підключити необхідну камеру для спостереження за об'єктом тривоги.

Мультиплексори

Мультиплексор дозволяє послідовно виводити на монітор і записувати на один відеомагнітофон інформацію від декількох телевізійних камер. При цьому запис здійснюється без втрати якості зображення. До мультиплексорів можна підключити систему сигналізації до ALARM-Входу. У деяких моделях це дасть можливість автоматично включити ту камеру, де відбулося порушення. Більшість мультиплексорів мають режим «динамічного розподілу часу запису» для кожної камери, а моделі MV-209 і MV-216 – вбудований детектор руху.

Детектори руху

При кількості камер більше чотирьох увага оператора розсіюється й ефективність спостереження знижується. При охороні банку потрібна установка великої кількості камер. Вирішити цю проблему можна установкою детекторів руху. Детектори руху обробляють відеозображення від телекамер і при

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

необхідності можуть включати відеомагнітофон для запису зображення або подавати сигнал тривоги. Детектор реагує на зміну зображення об'єкта (контраст або рух) і подає сигнал тривоги.

Матричні комутатори

При великій кількості камер ефективність роботи оператора може бути підвищена шляхом застосування матричних комутаторів. Матричний комутатор дозволяє створити гнучку й нарощувану систему безпеки, у яку можуть входити не тільки системи телевізійного спостереження, але й системи охорони й контролю доступу. При наявності детектора руху, комутатор самостійно відслідковує ситуацію й, у випадку тривоги, виводить зображення від камер на монітори. Передумовки дозволяють задавати комутатору «маршрут» огляду об'єкта. При цьому на монітор будуть виводитися зображення обраних камер, збільшувати трансформатор й т.д. Він називається режимом «вартового».

Відеопринтери

Для реєстрації відеозображення, поряд зі спецвідеомагнітофонами, у системах охорони використовуються й відеопринтери. Відеопринтери дозволяють роздрукувати:

- фотографії клієнтів;
- фотографії небажаних відвідувачів;
- кадри надзвичайних ситуацій;
- кадри з будь-якої відеокасети.

Передача зображення через IP-канал

Система дозволяє передавати оцифроване зображення через IP-мережу. Можливо не тільки запитувати зображення, але й видавати сигнали керування на виконавчі пристрої, такі як поворотні пристрої, ворота, сирени й т.д. Здійснюється цифрова обробка й стиск відеоінформації. Цифровий метод передачі відеоінформації дозволяє використовувати одну лінію для передачі відео, графічних, інформаційних, тривожних, керуючих і програмних сигналів. Інформація передається блоками у відповідності зі спеціальним протоколом

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

обміну, що дозволяє уникнути втрату інформації. При передачі серії кадрів, видається інформація тільки про зміни в зображенні. Середня швидкість 4800 бод.

Складовою частиною системи є програмне забезпечення, розроблене як для передавального, так і для прийомного пристрою. Воно містить зручне меню, у якому можна вибрати роботу з однією картинкою або з послідовністю кадрів, програмування режиму роботи камери й доступ до керуючого й архівному меню. У керуючому меню встановлюються IP-адреса й паролі доступу. У тривожній ситуації можна витягти картинки, що зберігаються в архіві, і відтворити їх у будь-якій послідовності на екрані або на принтері.

Вибір системи телевізійного спостереження

Будь-яка система телевізійного спостереження включає три функціональні частини:

- телевізійні камери;
- апаратуру обробки відеоінформації;
- монітори.

По способу прийому й обробки відеоінформації розрізняють:

- традиційні системи тспостереження на базі спеціалізованої апаратури;
- комп'ютерні системи телевізійного спостереження.

Завдання системи телевізійного спостереження – наочно представити відеоінформацію про оперативну обстановку контрольованого об'єкта. Для рішення цього завдання, відповідно до характеристик контрольованих об'єктів, вибираються параметри системи. До основних факторів, що визначають вибір состава системи телевізійного спостереження відносяться:

- кількість контрольованих об'єктів;
- швидкість реакції системи;
- вартість;
- простота керування й можливість роботи у веденому режимі;
- надійність;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

– гнучкість.

Параметри елементів системи телевізійного спостереження вибираються відповідно до характеристик об'єктів:

- розміри об'єктів;
- середня відстань до об'єктів;
- швидкість переміщення об'єктів;
- умови висвітлення об'єктів.

Завдання системи телевізійного спостереження – наочно представити відеоінформацію про оперативну обстановку контрольованого об'єкта. У комп'ютерних системах на одному моніторі відображається не більше 16 камер. При більшому числі камер розміри окремих зображень сильно зменшуються, а відеоканали перемикаються в режимі перелистування блоками до 16 камер.

Наочність подання оперативної обстановки вище в системах з великою кількістю моніторів, тому що при цьому можливо відображення всіх камер одночасно із зображенням потрібного розміру. Швидкість обробки відеоінформації близька до обробки в масштабі реального часу й при оптимальному складі засобів обробки відеоінформації не залежить від кількості камер. У комп'ютерних системах швидкість обробки відеоінформації зменшується в міру росту кількості камер. Швидкість реакції апаратури на дії оператора вище в традиційних системах. Методи цифрової обробки дозволяють поліпшувати відеозображення, фільтрувати шуми, виділяти й досліджувати окремі деталі.

Комп'ютерні системи телевізійного спостереження

Комп'ютерні системи спостереження призначені для комплексного керування системою телеспостереження. Вони можуть забезпечувати охорону й контроль доступу в приміщення як невеликих, так і великих офісів банківської мережі й т.д.

Комп'ютерні системи забезпечують:

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- Перегляд кольорового й чорно-білого відеозображення від одного до шістнадцяти джерел відеосигналів одночасно або на вибір оператора.
- Автоматичне або напівавтоматичне покадрове збереження зображення в цифровому виді із заданою дискретністю.
- Накладення дати, часу, службових сигналів і іншої інформації на відеозображення.
- Стиск і передачу по каналах обчислювальної мережі (глобальній, локальній), а також по каналах телефонного зв'язку через модем.
- Покадровий перегляд збереженої відеоінформації з можливістю завдання вибірки й сортуванню по даті, часу, найменуванню об'єкта та ін.
- Обробку зображення цифровими методами в реальному масштабі часу:
 - трансфокація;
 - регулювання яскравості, колірної насиченості, контрастності;
 - монтаж відеозображень;
 - компенсація тла, засвіток, фільтрація шумів та ін.
- Дистанційне керування системою по телефонній лінії.
- Підключення службових сигналів (сигнал тривоги, виклику, і ін.) і можливість автоматичного керування системою по заданому алгоритмі (зменшення інтервалів часу запису кадрів при надходженні сигналів тривоги).
- Програмне й дистанційне керування системами охорони й багаторівневого доступу.

Основною перевагою комп'ютерних систем є їхня гнучкість. Для керування засобами телеспостереження, сигналізації й системою контролю доступу використовують програмні засоби. Графічний редактор дозволяє побудувати план будинку з автоматичним виводом зображення «тривожної зони». Програма керування пристроями телеспостереження дозволяє управляти комутаторами, відеомагнітофонами, мультиплексорами, трансфокаторами камер, моніторами й т.д.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

3.2 Розробка структурної схеми

В процесі практичної реалізації теоретичних принципів розробки системи, була розроблена структурна схема обробки відеосигналу, яка зображена на рисунку 3.1. Завдяки структурній схемі можна чітко побачити основні структурні блоки системи та взаємозв'язки між ними.

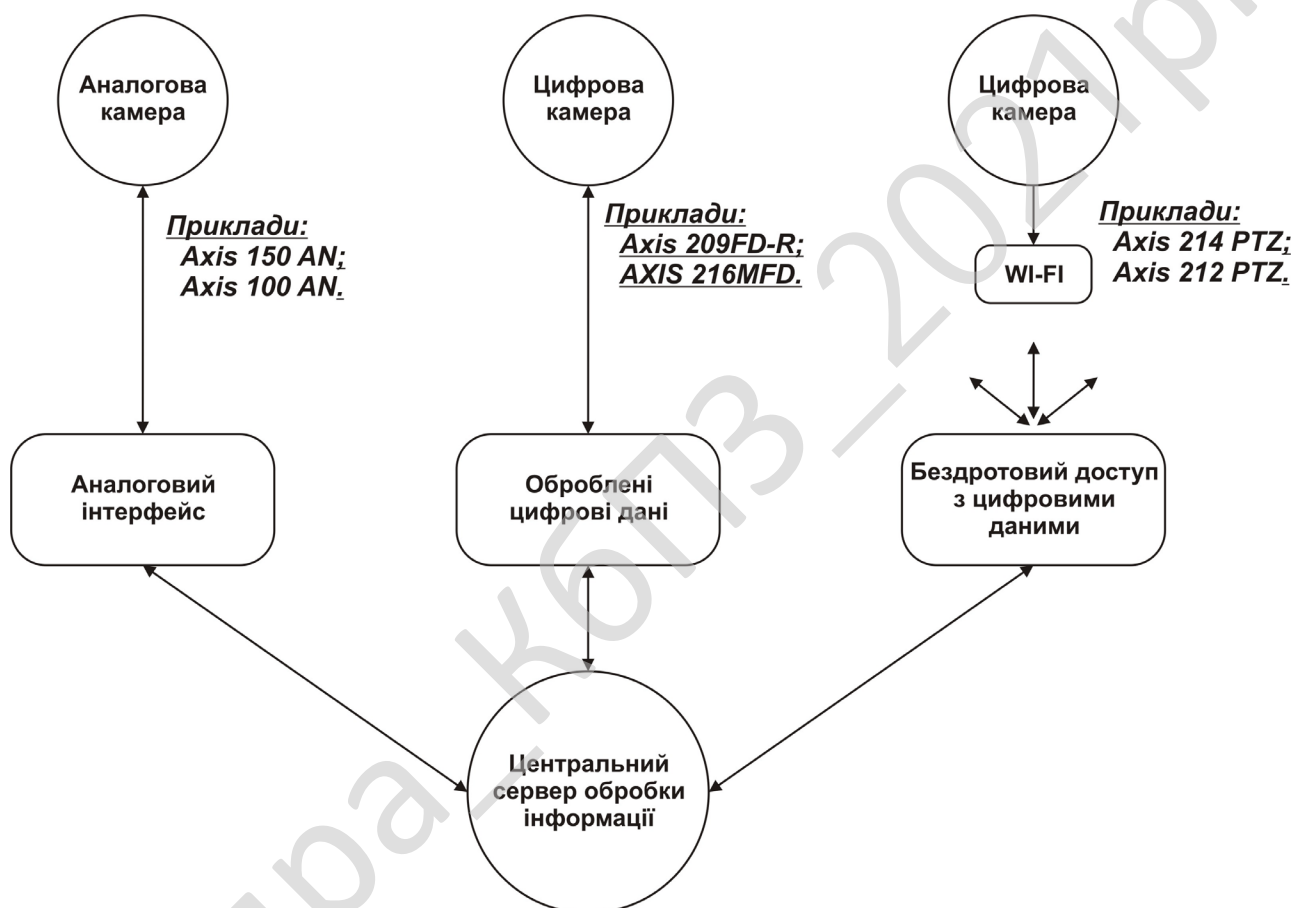


Рисунок 3.1 – Структурна схема обробки відеосигналу

При розробці структурної схеми основний упор робився на існуючі розробки ПЗ і їх модулі допомоги.

Аналіз рисунка 3.1 дозволяє чітко прослідити як працює програма. Розглянемо схему зверху вниз, в напрямку від пристрою до кінцевої програми.

Відеосигнал йде з блоку спостереження (цифрова відеокамера, аналогова відеокамера) після отримання відеоінформації з якого відбувається передача

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

41

поток у центральний сервер обробки інформації. Передача залежить від способу підключення пристрою відеоспостереження, що і відображено у середньому блоці структурної схеми обробки відеосигналу.

Для проведення спостереження за об'єктами застосовувалося встаткування фірми Axis Communications. Устаткування фірми дозволяє об'єднати всі пристрої цифрової й аналогової системи відеоспостереження в мережу з єдиним центром керування (приклади встаткування зазначені на рисунку 3.1.).

Системи відеоспостереження найбільш ефективні для банківської сфери, де є розгалужені приміщення, склади, офіси й інші об'єкти, де без вилученого відеоспостереження просто не обійтися.

За рахунок децентралізації всієї системи відеоспостереження служба охорони, може уникнути марного використання мережних ресурсів і незручностей, що виникають через обмежену пропускну здатність мережі.

Головним інтелектуальним вузлом цифрової системи відеоспостереження на об'єкті є мережний цифровий відеореєстратор AXIS 100 AN, 150 AN. До нього можна підключити 4 аналогові або 5 цифрових відеокамер (рисунок 3.2), установлених на віддалених об'єктах. Інформація з відеокамер стискується JPEG компресором відеореєстратора й записується на його жорсткі диски.



Рисунок 3.2 – Задня стінка відеокамери з доступними інтерфейсами взаємодії застосовувана при реалізації дипломного проекту

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Запис може здійснюватися в декількох режимах: безупинно, запис подій/тривожних ситуацій, ручна або за графіком. Відеореєстратор має всі необхідні компоненти для запису й перегляду відео по мережі: центральний процесор, Ethernet інтерфейс, контролери пам'яті, інтерфейси пристроїв, JPEG компресор, місце для установки до чотирьох жорстких дисків IDE.

Обраний метод побудови системи в дипломному проектуванні в побудові цифрової системи відеоспостереження, різко знижує обсяг переданої по мережі інформації й полегшує роботу операторів, що в остаточному підсумку дає великий економічний ефект. Висока надійність AXIS 100 AN, 150 AN і підтримка різних протоколів обміну інформації дозволяють використовувати його як у локальні (LAN), так і в глобальні (WAN) мережах, а застосування апаратури фірми Adder (розглянуто нижче) дозволяє забезпечити високий ступінь безпеки, що необхідно в банківській справі. Цифрову систему відеоспостереження досить легко інтегрувати й з уже наявними системами охоронної сигналізації й контролю доступу на об'єкт.

Для організації надійних каналів передачі відеопотоку, а також організації робочого місця оператора (відеотермінала). При розробці дипломного проекту була обрана апаратне забезпечення фірми Adder.

Устаткування Adder дозволяє збільшити число використовуваних моніторів і пристроїв введення (мишок/клавіатур), необхідних для керування системою відеоспостереження.

Абревіатура використовувана на структурній схемі (рисунок 3.3) побудови й роботи системи VKVM розшифровується як як: «клавіатура» («Keyboard»), «монітор» («Video monitor»), «миша» («Mouse»).

Розглянемо докладно як реалізована робота VKVM у випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти – до центрального ПК за допомогою інтерфейсного кабелю Adder VKVM-2M, підключається AdderView AVP4 – KVM перемикач. Він являє собою цифровий пристрій, комутуючий відеосигнал монітора й сигнали клавіатури й миші від

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

центрального комп'ютера на кілька комплектів Клавіатура/Монітор/Миша (Keyboard/Video/Mouse.) KVM перемикач AdderView AVP4, складається із двох основних пристроїв:

– відео-перемикач, що міняє напрямок аналогових відеоімпульсів між моніторами й комп'ютером спільного використання;

– мікропроцесорна система, що передає й приймає сигнали із клавіатури й миші й дає можливість управляти комп'ютером з кожного з робочих місць по черзі перемикаючись між ними).

При цьому число комплектів Клавіатура/Монітор/Миша визначається тільки можливостями й кількістю KVM-перемикачів. Немає необхідності мати спеціальне програмне забезпечення, і відсутні традиційні громіздкі процедури підключення все реалізується на апаратному рівні, що забезпечило можливість у випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти реалізувати практичну й зручну систему терміналів і зосередити зусилля на більше глибокому проробленні реалізації обробки відеопотоку інформації.

KVM перемикач AdderView AVP4 здатний віддаленно підключити до центрального комп'ютера до 4 комплектів Клавіатура/Монітор/Миша або інше встаткування. Для того, щоб розгалузити систему до AdderView AVP4 за допомогою інтерфейсних кабелів Adder VKVM – 0,5М використовувалося підключення 3 KVM перемикачі AdderView AVP2, кожний з яких дозволяє віддаленно підключити до 2 робочих місць, що підходить у поставленому завданні дипломного проектування.

До них, у свою чергу, підключаються подовжувачі AdderLink ALXT, які допомагають одержати високу якість зображення й звуку на відстані до 300 м.

Екстендери Adder, забезпечують передачу відеосигналу з високим розрішенням, і можливість гнучкого розподілу пристроїв, керування, контролю й взаємодії із пристроями відображення.

При використанні екстендерів Adder AdderLink немає необхідності використовувати оптоволоконні кабелі й розташовувати центральний процесор

поруч із кожним екраном. Екстендер ALXT, що є приймачем цифрових сигналів зв'язується з ALXR (передавач цифрових сигналів), за допомогою мережного кабелю CAT5 (кабель типу "кручена пара" категорії 5).

До екстендерів ALXR за допомогою кабелів Adder VKVM – 0,5М підключаються 6 комплектів Клавіатура/Монітор/Миша.

Вибір такого типу обладнання в дипломному проектуванні дозволило застосовувати тільки один кабель для кожного підключення до кожного робочого місця, що дуже зручно при прокладанні комунікаційних ліній.

3.3 Розробка функціональної схеми

На функціональній схемі, зображеній на рисунку 3.3, зображена схема внутрішньої будови системи відеомоніторингу банку.

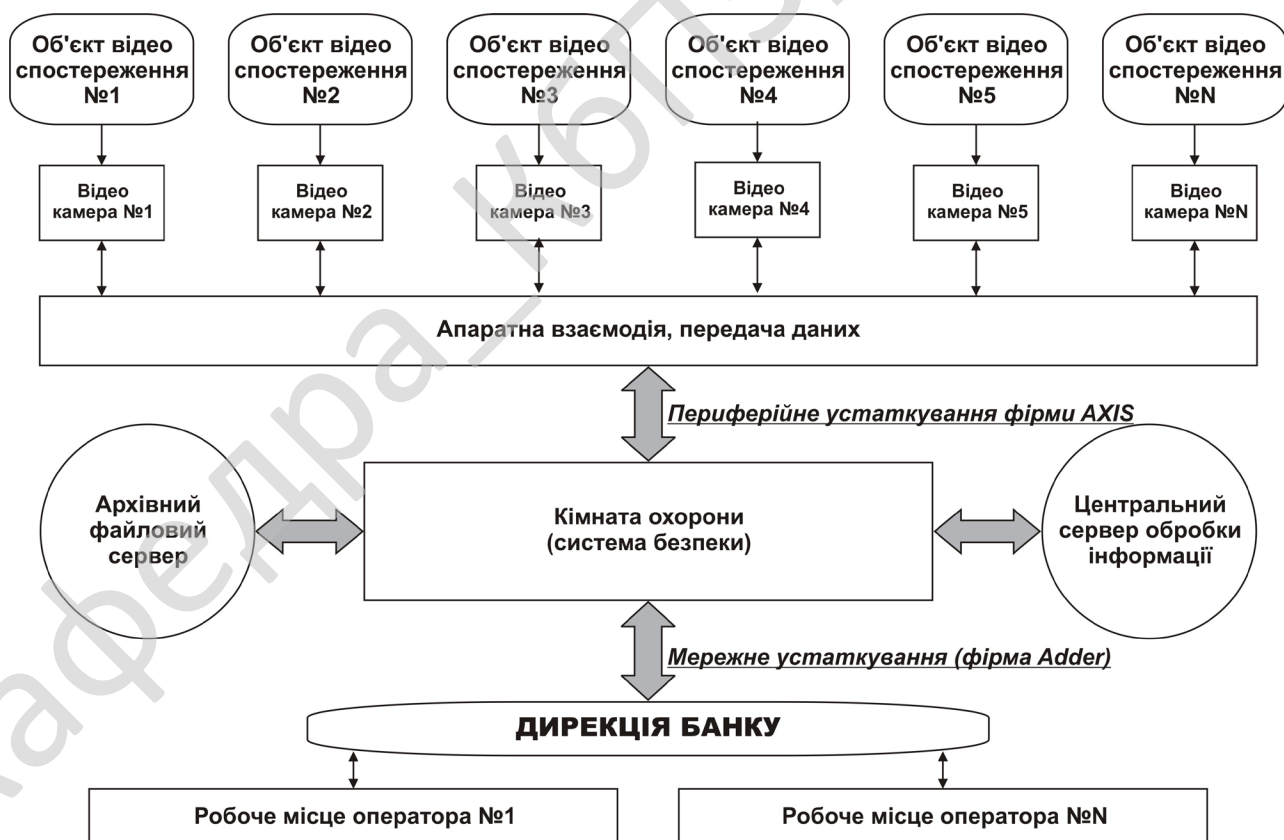


Рисунок 3.3 – Функціональна схема роботи системи

Найпершим блоком у схемі є кінцевий об'єкт за яким власне і ведеться спостереження у банку. Далі за допомогою відеокамер спостереження фірми AXIS за розробленою схемою (рисунок 3.1) і застосуванні апаратури фірми Adder (рисунок 3.2) відеосигнал надходить у кімнату охорони де обробляється операторами. Для організації масштабовості системи й контролю дій над співробітниками сигнал також надходить у дирекцію банку. Щодоби інформація з камер спостереження скидається в на сервер файлового архіву де зберігається в плинні місяця. Розробляючи апаратну й програмну частину дипломного проектування враховувався ціновий фактор, а також особливості застосування камер в умовах України, а саме – застосовувався ударостійкий корпус, були обрані камери з розширеними температурними діапазонами роботи (-40 С до +55 С) і великим кутом огляду камери.

3.4 Розробка діаграми процесів

Важливим критерієм при розробці програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів. Розглянемо діаграму процесів (рисунок 3.5). На діаграмі процесів можна точно зрозуміти як працює і взаємодіє ПЗ в цілому. Починається і закінчується програма в першому блоці є основною крапкою відрахунку діаграми. При переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їх входження один в одного.

З основного блоку програми через інтерфейс ПЗ, з можливістю налаштування ПЗ і перевірки програми на помилки ми попадаємо в блок обробки відеоінформації, через який відбувається аналіз якості й аналіз сигналу відеопотоку, а також доступ до файлового сервера й довідкової системи.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

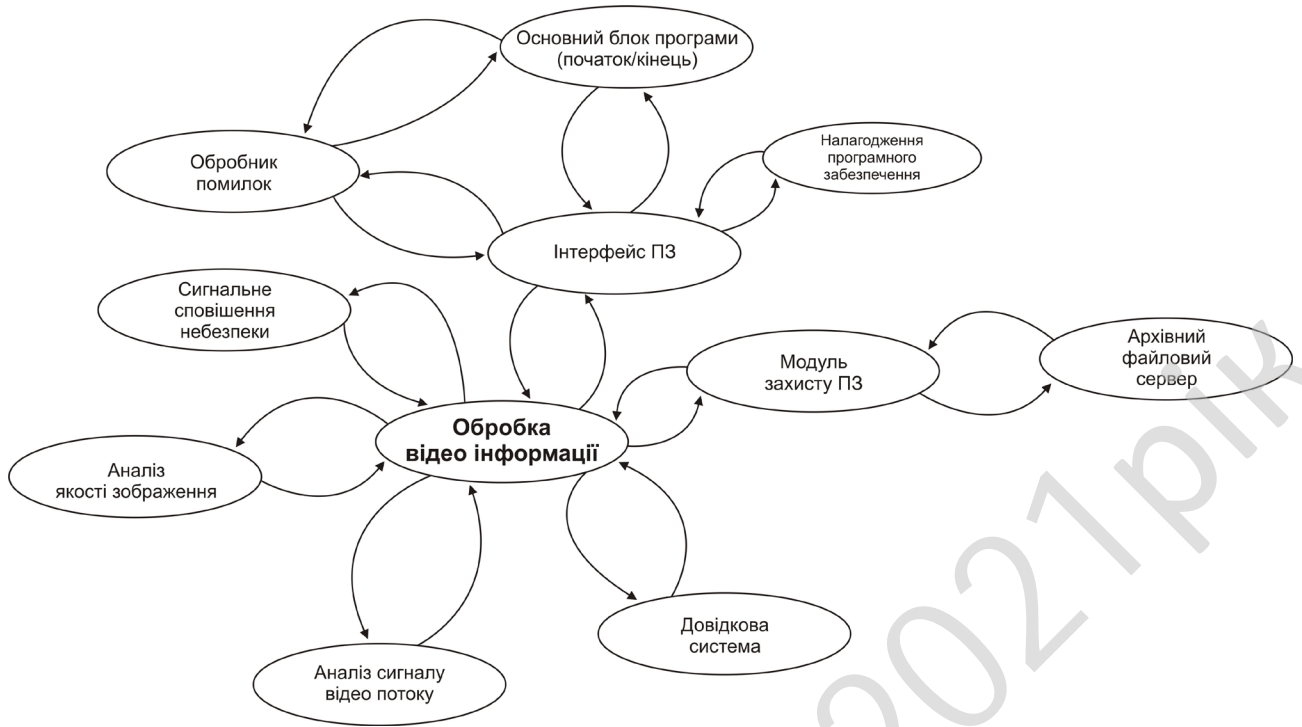


Рисунок 3.4 – Діаграма процесів

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.

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

47

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена основна блок-схема програми. Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єкно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю захвата відеопотоку зображення, модулю обробки помилок програми і основному модулю.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів взаємодії з відео даними я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем. Перед розглядом подробиць схеми роботи програми розглянемо виконані основні напрацювання.

Розроблені класи:

– Клас TAbstractAllocator

Оголошення класу:

```
Type AbstractAllocator = class()
```

Опис:

Абстрактне виділення покажчика для захоплення відео потоку.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

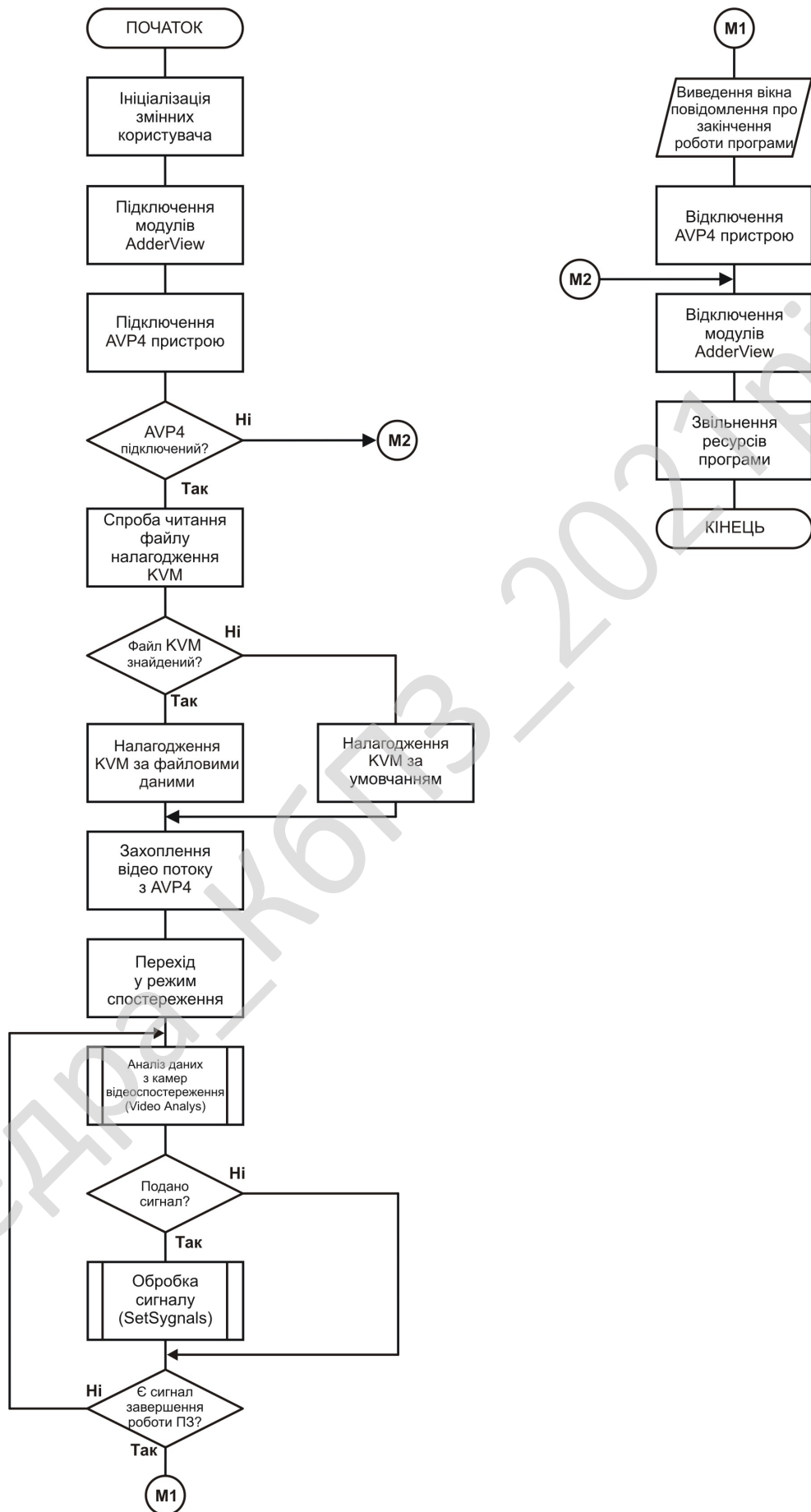


Рисунок 4.1 – Блок схема основної частини ПЗ

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

49

– Клас TASFWriter

Оголошення класу:

Type TASFWriter = class(A,B)

Опис:

Цей клас розроблений, щоб створити тимчасовий файл ASF (журнал роботи ПЗ).

– Клас TColorControl

Оголошення класу:

type TColorControl = class()

Опис:

Установка і отримання контролю кольору від Video вікна. Клас підтримує апаратні API функції операційної системи. Одночасно можна провести перевірку на здібність доступу програми до API функцій операційної системи.

– Клас TrackBar

Оголошення класу:

type TTrackBar = class(A,B)

Опис:

Панель коректування відео потоку інформації.

– Клас TVideoWindowEx2

Оголошення класу:

type TVideoWindowEx2 = class(A,B,C)

Опис:

Альтернатива до стандартного відеовікна, який пропонує більш швидкий і легкий шлях висновку відео на екран в додатку.

– Клас TVideoWindowEx2Caps

Оголошення класу:

type TVideoWindowEx2Caps = class()

Опис:

Перевірка на сумісність висновку потокової інформації.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

– Клас TFilter

Оголошення класу:

```
type TFilter = class(A,B)
```

Опис:

Клас пропонує можливість створення і підключення до програми додаткових фільтрів.

– Клас TFilterGraph

Оголошення класу:

```
type TFilterGraph = class()
```

Опис:

Центральний клас де відбувається синхронізація і протоколювання системи призначених для користувача повідомлень, а також повідомлень операційної системи.

– Клас TSampleGrabber

Оголошення класу:

```
type TSampleGrabber = class(A,B,C)
```

Опис:

Клас дозволяє створювати знімки екрану і знімки аудіо ділянок.

– Клас TVideoWindow

Оголошення класу:

```
type TVideoWindow = class(A,B,C)
```

Опис:

Управління висновком потокового відео на екран додатку.

– Клас TVMRBitmap

Оголошення класу:

```
type TVMRBitmap = class
```

Опис:

Коректування вида матриці формату BMP.

Два останні блоки є двома підпрограмами відображеними на рисунку 4.2. Підпрограма Video Analys описує алгоритм аналізу потокового відео операторами служби безпеки банку.

Саме в цьому блоці проводиться попереднє отримання апаратної інформації з AVP4, перетворення потокової інформації, аналіз кадру відео інформації та встановлення об'єктної чутливості, обробка кадру відео інформації та проведення запиту.

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

При застосуванні у випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти багатокрокових алгоритмів обробки кадру для створення відео датчиків сигналізацій відбувається виведення менше 24 кадрів в секунду, незалежно від потужності персонального комп'ютера, що приводить до величезних проблем при експлуатації програми (поява слайдшоу).

Після зменшення чутливості (залежно від настройок) програма дозволяє провести точний розрахунок який дає результат про зміни у відео кадрі.

При поверненні сигналу з Video Analys проходить обробка тривоги яка обробляється у другій підпрограмі обробки сигналу (SetSignals) яку можна побачити на рисунку 4.2.

Підпрограма SetSignals реалізує процес обробки прапора тривоги зміни зображення, що означає зміни на картинці відеокамери (відеосигнал змінений або спотворений). В першу чергу відбувається запис в журнал роботи програми про зміну з точною вказівкою часу і збереження кадрів результату. Далі залежно від дій операторів походить сповіщення незаконних чі сповіщення незаконних дій.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

**Підпрограма - аналіз даних
з камер відеоспостереження
(Video Analys)**



Підпрограма - обробка сигналу (SetSignals)

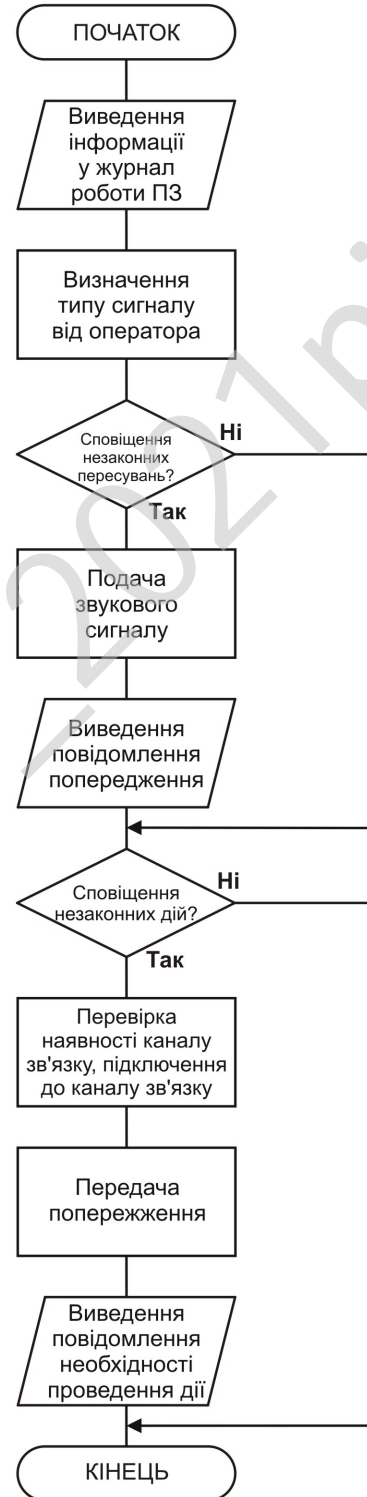


Рисунок 4.2 – Блок схема підпрограм

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192- або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири 8×8 -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавiш. Шар дифузiя використовує лінійне перетворення на основі матриці з номером галузей 5.

Аналіз безпеки

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Неможлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

Продуктивність

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

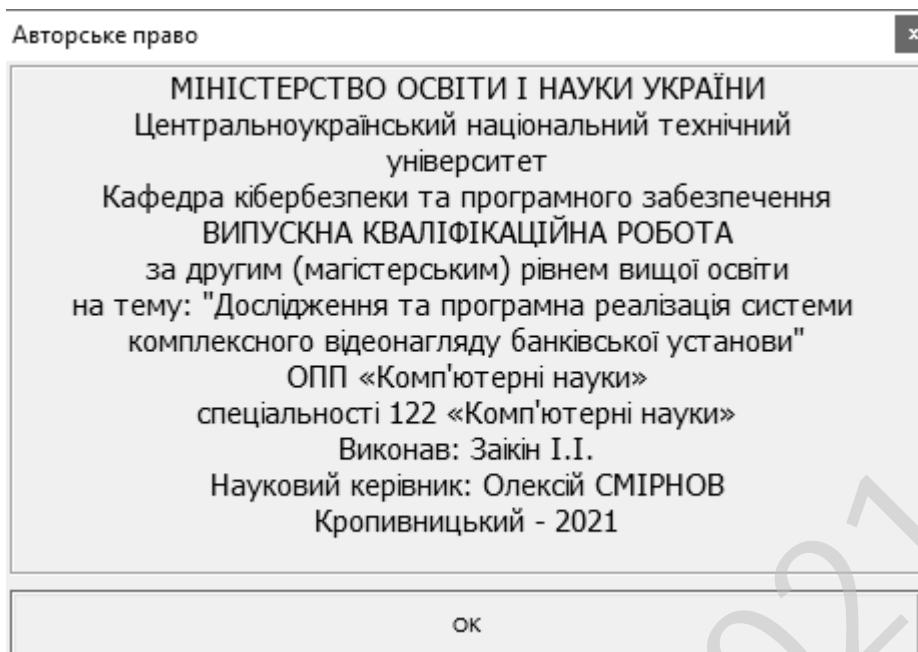


Рисунок 5.1 – Авторські дані ПЗ

Після запуску програми на екрані з'явиться головне вікно, яке зображене на рисунку 5.2.

Як видно з рисунка головного вікна програми умовно можливо розділити на чотири частини. У верхній частині програми виконане стандартне контекстне меню, яке дозволяє швидко отримати необхідну інформацію.

Нижче контекстного меню з правої сторони знаходиться інтерфейсний набір кнопок для реалізації основних переходів в програмі і запуску її методів.

З лівої сторони знаходиться область виводу обробленого потокового відео. Остання умовна частина програми це інформаційна стрічка, яка дублює вивід інформації на екран при наведенні мишки на описаний об'єкт.

В програму добавлено функцію переключання відео та аудіо стандарту. При не чіткому зображенні відео зображенні або як що в картинці забагато шуму, то можливо включити додаткову обробку відео потоку за допомогою стандартних відео фільтрів, або додаткових та ввімкнути обробку шумоподолання, що нормалізує відеопотік.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

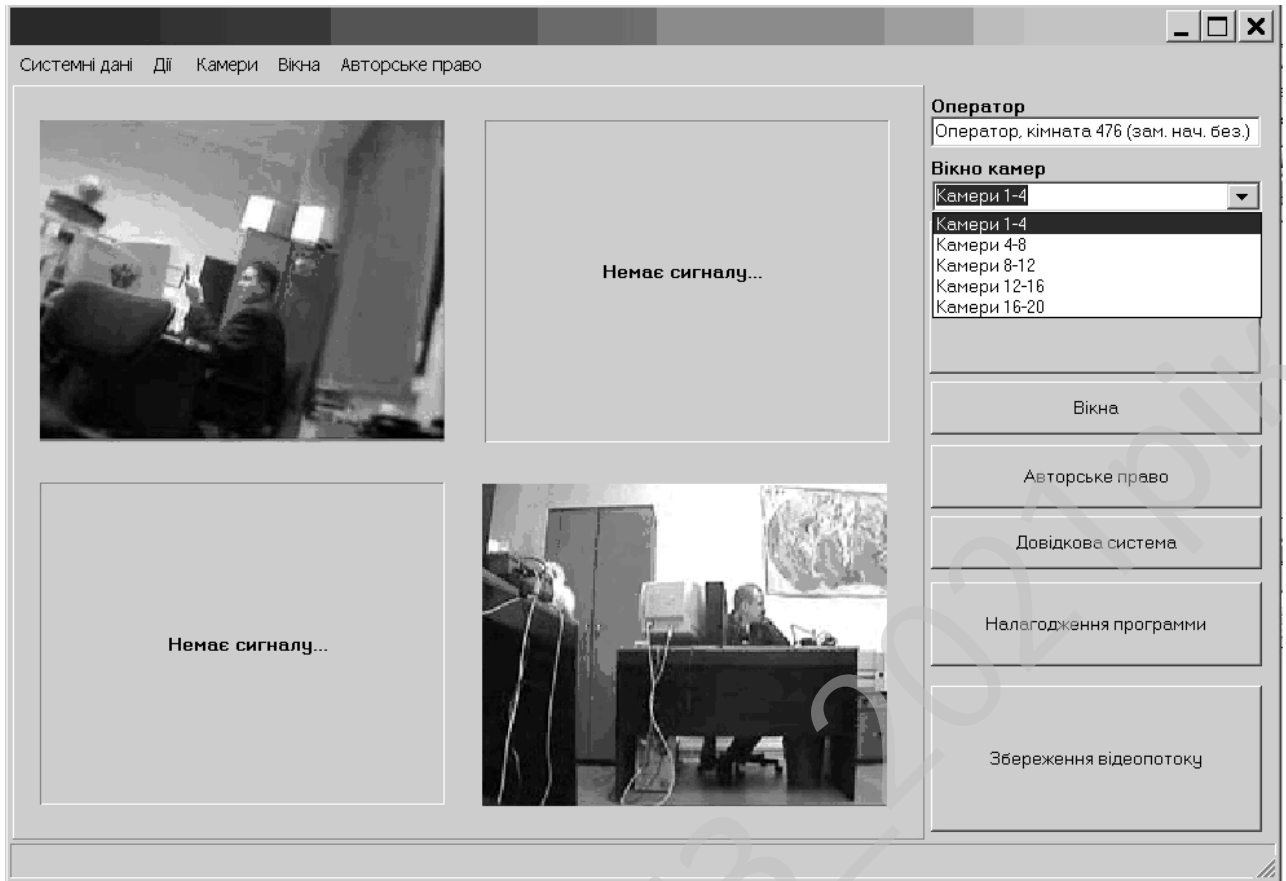


Рисунок 5.2 – Головне вікно програми з аналізатором потокового відео

Розроблене програмне забезпечення, рекомендується для впровадження в фірмах, банківських установах, організаціях, що займаються фінансовою діяльністю та потребують у відео спостереженні.

6 НАУКОВА НОВИЗНА

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

Метою розробки є дослідження та програмна реалізація системи комплексного відеонагляду банківської установи.

Об'єктом дослідження є процес комплексного відеонагляду банківської установи.

Предметом дослідження є методи комплексного відеонагляду банківської установи.

Методи дослідження базуються на методах технічного захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод комплексного відеонагляду банківської установи.
- Розроблено вітчизняний продукт комплексного відеонагляду банківської установи, який має більше широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

63

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	20000 (2 ост. цифр № зал*10 ⁴)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП як одна з найбільш тривалих і трудомістких робіт значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боєма, $A = 2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,84 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 100 = 168 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

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

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

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

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,1	0,3
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,1	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,1	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	2,1	
Всього		2,4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	0,2	0,1
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

69

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	18000	18000
Продакт-менеджер	0,25	16000	4000
Інженер-програміст	3,8	16000	60800
Інженер-електронщик	1,2	16000	19200
Інженер-системотехнік	0,25	16000	4000
Адміністратор мережі	0,3	16000	4800
Системний програміст	0,2	17000	3400
Дизайнер WEB	0,2	16000	3200
Інженер-верстальник	0,1	15000	1500
Бухгалтер-економіст	0,1	17000	1700
Всього за період розробки	$R_{cn} = 7,4$	-	$\Phi_{роб} = 120600$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{120600}{7,4 \cdot 26} = 627 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кропивницький) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 .

Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

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

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

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

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет-магазину Компбест за 19.10.21 – джерело <https://compbest.com.ua>.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	LogicPower SFF	7347
Процесор	Intel Core i5-7400 (4 ядра по 3.0 – 3.5 GHz) Cache Memory 6 MB	-
Системна плата	Gigabyte GA-B250M-D3H, 4x USB 3.1, 4x USB 2.0, 8x Audio Ports, 1x LAN (RJ-45), 1x PS/2, 1x HDMI, 1x DP, 1x DVI, 1x VGA, 1x M.2, 1x SATA Express, 6x SATA	-
Відеокарта	Integrated Intel HD Graphics 630, частота графічної системи 350 MHz – 1.00 GHz	-
Жорсткий диск	240 GB SSD Goodram CX200	-
Оперативна пам'ять	16 GB DDR3	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	ATX LogicPower SFF, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm) black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-Black int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0002.00.00.ПЗ

Арк.

72

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57443
Група 5, 6			
4. Вимірювальні пристрої	9031	25	2257,75
5. Транспортні засоби	143000	20	28600
6. Господарський інвентар	28000	25	7000
Всього по групі 5,6	180031	-	37857,75
Нематеріальні активи			
7. Нематеріальні активи	20000	10	2000
Разом	$K_p = 1722916$		$A_p = 167701$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 627 \cdot 209 / 20 = 6552 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 6552 \cdot 10 \cdot 0,01 = 655 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{ou} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{ou} = 0,01 \cdot 22(6552 + 655) = 1586 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 6552 \cdot 15 \cdot 0,01 = 983 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджей, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Згідно норм приймаємо одну пачку паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 105$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

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

$$Z_{M1} = 105 \cdot 1 = 105 \text{ грн.}$$

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

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

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 3 грн./шт.

$$Z_{M2} = 20 \cdot 2 + 3 = 43 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (105 + 43 + 1702) / 20 = 93 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 6552 \cdot 15 \cdot 0,01 = 983 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 20$ прим.):

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 167701 \cdot 1 / (20 \cdot 12) = 699 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

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

$$C_n = 6552 + 655 + 1586 + 983 + 93 + 983 + 699 = 11551 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 11551 = 5776 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	6552
2. Додаткова зарплата виконавців	Z_δ	655
3. Відрахування на соціальні потреби	C_{oc}	1586
4. Загальногосподарські витрати	Γ_{ocn}	983
5. Витрати на матеріали	Z_m	93
6. Освоєння нових операційних систем, мов програмування	O_n	983

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	699
8. Повна собівартість програмного забезпечення	C_n	11551
9. Плановий прибуток	P_p	5776
10. Ціна підприємства $C_n = C_n + P_p$	C_n	17327
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	3465,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	20792,4

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	20792
Всього капітальних витрат	–	20792

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

$$E_{cn} = (23586 - 14980) - 0,25 \cdot 20792 = 3408 \text{ грн.}$$

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

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{20792}{23586 - 14980} = 2,4 \text{ роки.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості управлінських рішень.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

«З давніх давен людство приділяє прискіпливу увагу безпеці життя і охорони праці як її складової частини. Умови праці розглядали Арістотель (384-322 до н.е.) та Гіппократ (460-377 до н.е.)» [4].

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

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

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,6
Довжина	6,6
Висота	2,75

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

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	7,4
Обсяг, V	м ³	не менше 20.0	20,3

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

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

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

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

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

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

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

8.5 Розрахункова частина

Тип заземлення: повторне заземлення нульового дроту на вводі в об'єкт. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина. Умовна товщина верхнього шару ґрунта: $H=0,4$ м. Для захисного заземлення: застосовуються вертикальні електроди – куток $D=12$ мм., довжиною $L=3$ м. Відстань між вертикальними заземлювачами (електродами) $A=4$ м. Тип горизонтального заземлювача: металева полоса з перетином 40×4 мм. ($b=0,004$ м.). Глибина закладення горизонтального контура заземлення $t=0,7$ м. Опір заземлювача, який нормується: $R_{3H} = 10$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр. 21-25 [6], або аналогічної.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+3/2=2,2 \text{ м.}$$

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

$$\rho = \psi \rho_2 = 1,2 * 40 = 54,5 \text{ Ом*м.}$$

де:

$\psi = 1,2$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [6];

$\rho_1 = 50 \text{ Ом*м.}$ – табличне значення питомого опору верхнього шару ґрунта [8];

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

Еквівалентний діаметр вертикального електрода (кутка):

$$D_{\text{в}} = 0,95 * b = 0,95 * 12 = 11,4 \text{ мм.} = 0,0114 \text{ м.}$$

Опір розтіканню електричного струму одного електрода вертикального заземлювача (з врахуванням заглиблення заземлювача):

$$R_0 = 0,366(\rho/L)[\lg(2L/D_{\text{в}}) + (1/2)\lg((4T+L)/4T-L)] = 0,366(54,5/3)[\lg(2*3/0,0475) + (1/2)\lg((4*2,2+3)/4*2,2+3))] = 5,62 \text{ Ом.}$$

Відношення $A/L = 4/3 = 1,3$.

Визначаємо коефіцієнт екранування вертикальних електродів $K_{\text{ев}} = 0,79$ при орієнтовній кількості вертикальних електродів, яке дорівнює 4 [3].

Визначаємо необхідну кількість вертикальних заземлювачів (без врахування горизонтального заземлювача), при $R_{3\text{Н}} = 10 \text{ Ом}$:

$$N = R_0 / (K_{\text{ев}} R_{3\text{Н}}) = 5,62 / (0,79 * 10) = 0,66 \approx 1 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\text{п}} = 1,05 * A * N = 1,05 * 4 * 1 = 4,05 \approx 4 \text{ м.}$$

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Опір розтіканню електричного струму з'єднуючої полоси:

$$R_{\Pi} = 0,366(\rho_2 * K_{\Pi} / L_{\Pi}) \lg(2(L_{\Pi} * L_{\Pi}) / (b * t)) = \\ = 0,366(40 * 5 / 15,75) * [\lg(2 * 15,75 * 15,75) / (0,04 * 0,7)] = 12,88 \text{ Ом.}$$

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

Загальний опір розтіканню електричного струму заземлювача:

$$R = (R_0 * R_{\Pi}) / (R_0 * \eta_{\Pi} + N * R_{\Pi} * K_{ев}) = \\ = (5,62 * 12,88) / (5,62 * 0,75 + 1 * 12,88 * 0,79) = 5,65 \text{ Ом.}$$

де $\eta_{\Pi} = 0,75$ – табличне значення коефіцієнта екранування з'єднуючої полоси [3].

Умова $R \leq R_{зН}$ виконується.

8.6 Висновки до розділу

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

9 ОСНОВНІ ВИСНОВКИ

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

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів комплексного відеонагляду банківської установи.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем комплексного відеонагляду банківської установи.
- Досліджена система комплексного відеонагляду банківської установи.
- На основі отриманих результатів досліджень створена програмна реалізація системи комплексного відеонагляду банківської установи.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комплексного відеонагляду банківської установи.

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

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

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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 RAD Studio Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Camellia.

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

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 3408 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 2,4 роки.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

8. Дреєв О.М. Оцінка якості стиснення зображень на основі дискретного перетворення Хартлі / О.В. Коваленко, О.П. Доренський, О.М. Дреєв // Системи озброєння і військова техніка. Науковий журнал 2(34)– X.: ХУПС – 2013. С. 99-102.

9. Дреєв О.М. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смирнов, О.М. Дреєв, О.П. Доренський // Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – X.: ХУПС – 2013. – С. 234-239.

10. Дреєв А.Н. Сравнение битовых плотностей при использовании различных методов кодирования информации / А.Н. Дреєв, А.А. Смирнов // Системи обробки інформації, 2014, випуск 2 (118), том 2– Харків: ХУПС – 2014. С 64-66.

11. Дреєв О.М. Моделювання впливу інтенсивності трафіку на оперативність доставляння інформації / О.М. Дреєв // Науково-виробничий журнал "Зв'язок". – Київ: ДУТ, 2014. – № 2 (108) С. 24-29.

12. Дреєв А.Н. Повышение вероятности доставки сообщений в телекоммуникационных системах и сетях для обеспечения информационной безопасности / А.Н. Дреєв, А.А. Смирнов // «Безпека інформації» Том 21, №1 2015 р. – Київ: НАУ – 2015. – С. 22-28.

13. Дреєв О.М. Узагальнення вейвлету Хаара / О.М. Дреєв, Г.М. Дреєва // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 15-16 жовтня 2010 р. – Кіровоград – С. 58

14. Дреєв О.М. Узагальнення вейвлету Хаара / О.М. Дреєв // Матеріали науково-практичної конференції, присвяченої 80-річчю фізико-математичного факультету КДПУ ім. В. Винниченка 26 листопада 2010 р. – Кіровоград – С. 12

15. Дреєв О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреєв, О.В. Коваленко // Тези доповідей Новітні технології – для захисту повітряного простору. Дев'ята наукова конференція. 18-19 квітня 2011 р. – X.: ХУПС. – 2012. – С. 206

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

22. Дреєв О.М. Визначення оптимального розміру блоку при бітовому арифметичному кодуванні / О.М. Дреєв, Г.М. Дреєва // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 11-12 квітня 2014 р. – Кіровоград – С. 44

23. Дреєв А.Н. Экстраполяция квазипериодических процессов с аддитивными помехами / А.Н. Дреєв, А.А. Смирнов // П'ята Міжнародна науково-практична конференція "Інформаційні технології та моделювання в економіці" 15-16 травня 2014 р. – Черкаси – С. 59

24. Дреєв А.Н. Статистическая модель передачи многопакетного сообщения в телекоммуникационной системе или сети / А.Н. Дреєв, А.А. Смирнов // «Компьютерное моделирование в наукоемких технологиях (КМНТ-2014)» Харьков, 28-31 мая 2014 года – С. 137-140

25. ДСТУ 2481 – 94 Системи оброблення інформації інтелектуальні інформаційні технології. Терміни та визначення. – Х.: ДЕРЖСТАНДАРТ УКРАЇНИ, 1994. – 33 с.

26. ДСТУ В 3265 – 95. Зв'язок військовий. Терміни та визначення. – К.:УкрНДІССІ, 1995. – 23 с.

27. Дымарский Я.С. Управление сетями связи: принципы, протоколы, прикладные задачи / Я.С. Дымарский., Н.П. Крутякова, Г.Г. Яновский – М.: ЭкоТрендз, 2003. – 384 с.

28. Ершов В.А. Мультисервисные телекоммуникационные сети / В.А. Ершов, Н.А. Кузнецов – М.: Изд. МГТУ им. Н.Э. Баумана, 2003. – 432 с.

29. Закон України «Про Концепцію Національної програми інформатизації» (Відомості Верховної Ради (ВВР), 1998, N 27-28, ст.181) (Із змінами, внесеними згідно із Законом N 5463-VI (5463 -17) від 16.10.2012, ВВР, 2014, N 4, ст.61). – <http://zakon1.rada.gov.ua/laws/show/74/98-%D0%B2%D1%80>.

30. Зайченко Ю.П. Компьютерные сети / Ю.П. Зайченко. – К.: Слово, 2003. – 256 с.

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

31. Зубко Р.А. Алгоритми стиснення зображень в системах цифрової обробки даних / Р.А. Зубко // Восточно-Европейский журнал передовых технологий. – 2013. – №1/2(61). – С. 40-44.

32. Иванов В.Г. Прогрессивные информационные технологии сжатия изображений / Иванов В. Г., Ломоносов Ю. В., Любарский М. Г. // Проблемы информатики та комп'ютерної техніки (ПІКТ – 2014) : пр. III-ї Міжнар. наук.-практ. конф., 27–30 трав. 2014 р. – Чернівці : Родовід, 2014. – С. 172–174.

33. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами / А.Г. Ивахненко – Киев: «Техніка». – 1975. – 312 с.

34. Игнатенко Е.Г., Бессараб В.И. Алгоритм адаптивного мониторинга загрузки кластерных web-серверов / Е.Г. Игнатенко, В.И. Бессараб // Збірник тез. Нові технології в телекомунікаціях. VI міжнародний науково-технічний симпозиум. – Карпати, Вишків: ДУІКТ. – 2011. – С.34.

35. Карпенко С.В. Метод компактного представления зображень в телекомунікаційних системах на основі тривимірного поліадичного кодування: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.12.02 – телекомунікаційні системи та мережі / С.В. Карпенко. – Харків: Харківський національний університет радіоелектроніки, 2009. – 22 с.

36. Касимов Р.Р. Вдосконалення алгоритмів QoS маршрутизації в мережах з технологією IP/MPLS на основі прогнозу трафіка: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.12.02 –телекомунікаційні системи та мережі / Р.Р. Касимов. – Київ: Київський державний університет інформаційно-комунікаційних технологій, 2011. – 24 с.

37. Кириченко Л.О. Влияние методов маршрутизации на QOS в мультисервисных сетях при самоподобной нагрузке / Л.О. Кириченко, Т.А. Радивилова, Э. Кайали // Восточно-Европейский журнал передовых технологий. Информационные технологии. 1/2 (49) 2011. – С. 15-18

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

56. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>

57. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

58. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

59. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

60. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

61. Охорона праці. Ч. 2. Занулення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM – сумісного типу / Центральноукраїн. нац. техн. ун -т. – 2–ге вид., перероб. та доп.; [укл. О. В. Оришака, Є. К. Солових, В. О. Та інші]. – Кропивницький: ЦНТУ, 2019. – 27 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/8769>

62. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

63. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

64. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://spo.stu.cn.ua>

					ВКРМ-122.21.0002.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.21.0002.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Заїкін І.І.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.						
					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КН-20М-1,4		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи комплексного відеонагляду банківської установи.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 39-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи комплексного відеонагляду банківської установи.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.21.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи комплексного відеонагляду банківської установи;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.21.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					ВКРМ-122.21.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута розробка заходів з умов поліпшення охорони праці.

					ВКРМ-122.21.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 101 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 24.12.2021 р.

					ВКРМ-122.21.0002.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Смірнов О.А.

*Дослідження та програмна реалізація
системи комплексного відеонагляду банківської установи*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 58

Літера: РП

Кропивницький – 2021 року

Файл основного файла программы Video_Surveillance_System_Project.dpr

```
program Video_Surveillance_System_Project;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

uses

    Forms,
    SysUtils,
    Unit1 in 'Unit1.pas' {Form1},
    frmLOG in 'G.pas' {Form3},
    frmLOG_GRAPH in 'G_GRAPH.pas' {Form4},
    frmSETTINGS in 'SS.pas' {Form2},
    frmSPLASH in 'SPLASH.pas' {Form5},
    frmAbout in 'About.pas' {AboutBox};

{$R *.res}

Begin

    Form5:=TForm5.Create(Application);
    Form5.Show;
    Form5.Update;
try

    Application.HintPause:=200;//ms
    Application.HintHidePause:=7000;//ms
    Application.HintShortPause:=25;//ms

    Application.Initialize;
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TForm3, Form3);
    Application.CreateForm(TForm4, Form4);
    Application.CreateForm(TForm2, Form2);
    Application.CreateForm(TAboutBox, AboutBox);

    finally
        Form5.Free;
    end;
Application.Run;

end.
```

Під час написання програми були використанні наступні стандартні бібліотеки та модулі для підтримки технології відеотехнології (DirectX):

comlited.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, amaudio.h, vptype.h,
vpconfig.h, vpnotify.h, mpegtype.h, dvdevcod.h, dvdmedia.h,
bdatypes.h, activecf.h, vfwmsgs.h, (edevdefs.h, XPrtDefs.h),
aviriff.h, evcode.h, uuids.h, ksuuids.h, DXVA.h, AMVA.h,
videoacc.h, regbag.h, tuner.h, DXTrans.h, QEdit.h, mpeguids.h,
dshowasf.h, amparse.h, audevcod.h, atsmmedia.h, MediaErr,
MedParam.h, mediaobj.h, dmodshow.h, dmoreg.h, DMORT.h,

dmoimpl.h, ks.h, ksproxy.h, ksmedia.h, dmksctrl.h, bdamedia.h,
BDATIF.idl, AMVPE.idl, Mixerocx.idl, Mpeg2Data.idl,
Mpeg2Structs.idl, Mpeg2Bits.h, Mpeg2Error.h, EDevCtrl.h,
sbe.idl, wmdxva.h, vmr9.idl

Файл основної форми програми Unit.pas

```
unit Unit1;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9,
  jpeg, Buttons;

type
  TForm1 = class(TForm)
    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;
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 declarations }
public
    procedure ShowCAPdevise;
    procedure SETKAS;
end;

var
    Form1: TForm1;
    SETKA: integer;
    CapEnum: TSysDevEnum;
    VideoMediaTypes, AudioMediaTypes: TEnumMediaType;
    CapFile: WideString = 'c:\zaxvat.avi';
    Ok_normal: boolean;
    AR: array of array of tcolor;
    GO: boolean;

implementation

uses frmAbout, frmLOG, frmSETTINGS;

{$R *.dfm}

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.AudioCapFilters.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_AudioInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
        Form2.AudioCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);
    VideoMediaTypes := TEnumMediaType.Create;
    AudioMediaTypes := TEnumMediaType.Create;
end;

```

```

procedure TForm1.Timer1Timer(Sender: TObject);
var
    position: int64;
    Hour, Min, Sec, MSec: Word;
const 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;

procedure TForm1.StartButtonClick(Sender: TObject);
var
    multiplexer: IBaseFilter;
    Writer: IFileSinkFilter;
    PinList: TPinList;
    i: integer;
begin
    // Activate the filter graph, at this stage the source filters are added to
    the graph
    CaptureGraph.Active := true;

    // configure output Audio media type + source
    if Form2.AudioSourceFilter.FilterGraph <> nil then
    begin
        PinList := TPinList.Create(Form2.AudioSourceFilter as IBaseFilter);
        i := 0;
        while i < PinList.Count do
            if PinList.PinInfo[i].dir = PINDIR_OUTPUT then
            begin
                if Form2.AudioFormats.ItemIndex <> -1 then
                    with (PinList.Items[i] as IAMStreamConfig) do
                        SetFormat(AudioMediaTypes.Items[Form2.AudioFormats.ItemIndex].AMMediaType^);
                PinList.Delete(i);
                end else inc(i);
            if Form2.InputLines.ItemIndex <> -1 then
                with (PinList.Items[Form2.InputLines.ItemIndex] as IMAudioInputMixer) do
                    put_Enable(true);
                PinList.Free;
            end;
    end;

```

```

end;

// configure output Video media type
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;

// now render streams
with CaptureGraph as IcaptureGraphBuilder2 do
begin
    // set the output filename
    SetOutputFileName (MEDIASUBTYPE_Avi, PWideChar(CapFile), multiplexer, Writer);
    // Connect Video preview (VideoWindow)
    if Form2.VideoSourceFilter.BaseFilter.DataLength > 0 then
        RenderStream(@PIN_CATEGORY_PREVIEW, nil, Form2.VideoSourceFilter as IBaseFilter,
            nil, VideoWindow as IBaseFilter);
        // Connect Video capture streams
        if Form2.VideoSourceFilter.FilterGraph <> nil then
            RenderStream(@PIN_CATEGORY_CAPTURE, nil, Form2.VideoSourceFilter as IBaseFilter,
                nil, multiplexer as IBaseFilter);

            // Connect Audio capture streams

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

        CaptureGraph.Play;
        StopButton.Enabled := true;
        StartButton.Enabled := false;
        Form2.AudioFormats.Enabled := false;
        Form2.AudioCapFilters.Enabled := false;
        Form2.VideoFormats.Enabled := false;
        Form2.VideoCapFilters.Enabled := false;

```

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

procedure TForm1.StopButtonClick(Sender: TObject);
begin
    Timer1.Enabled := false;
    StopButton.Enabled := false;
    StartButton.Enabled := true;
    CaptureGraph.Stop;
    CaptureGraph.Active := False;
    Form2.AudioFormats.Enabled := true;
    Form2.AudioCapFilters.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;
    Form2.show;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
    Form1.hide;
```

```
Form2.show;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin

if Button7.Caption='Включить детектор движения' then
begin
Button7.Caption:='Выключить детектор движения';
Go:=false;
Timer2.Enabled:=true;
end
else
begin
Button7.Caption:='Включить детектор движения';
Timer2.Enabled:=false;
end;

end;

procedure TForm1.Button8Click(Sender: TObject);
begin
Form1.hide;
Form2.show;

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;  
Form2.show;  
end;
```

```
procedure TForm1.N9Click(Sender: TObject);  
begin  
Application.Minimize;  
end;
```

```
procedure TForm1.N10Click(Sender: TObject);  
begin  
if Ok_normal then  
begin  
MessageDlg('Проверка прошла успешно', 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; BlyYa:=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;
    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;

procedure TForm1.Button11Click(Sender: TObject);
begin
  if SaveDialog1.Execute then
    begin
      if SaveDialog1.FileName<>' ' then
        begin
          SampleGrabber.GetBitmap(Image.Picture.Bitmap);
        end;
    end;
end;

```

```

        Image.Picture.SaveToFile(SaveDialog1.FileName);
    end;
end;
end;

procedure TForm1.SETKAS;
var
    x,y:integer;
    G:tcolor;
begin

//X VideoWindow.ClientWidth
//Y VideoWindow.ClientHeight
//VideoWindow.Canvas.Pixels[]
//  Label1.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);

Label1.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('изм г на г');
                    end;

////////*****

                end;

            if ((y mod SETKA)=0) and (x = 1) then
            begin
                //Image.Canvas.Rectangle(x,y,x+4,y+4);

```

```

////////*****
if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
begin
  Ar[X,Y]:=Image.Canvas.Pixels[x,y];
  Form3.Memo1.lines.add('изм г на 1');
end;
////////*****
end;

if (y = 1) and ((x mod СЕТКА)=0) then
begin
  //Image.Canvas.Rectangle(x,y,x+4,y+4);
  //////////*****
  if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
  begin
    Ar[X,Y]:=Image.Canvas.Pixels[x,y];
    Form3.Memo1.lines.add('изм 1 на г');
  end;
end;

////////*****

end;

if (y = 1) and (x = 1) then
begin
  //Image.Canvas.Rectangle(x,y,x+4,y+4);
  //////////*****
  if Ar[X,Y]<>Image.Canvas.Pixels[x,y] then
  begin
    Ar[X,Y]:=Image.Canvas.Pixels[x,y];
    Form3.Memo1.lines.add('изм 1 на 1');
  end;
end;

////////*****

end;
end;
end }
end;
//-----

if (not 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);

Label1.Caption:=ColorToString(VideoWindow.Canvas.Pixels[x,y]);
          Ar[X,Y]:=Image.Canvas.Pixels[x,y];
        end;

      if ((y mod SETKA)=0) and (x = 1) then
        begin
          Image.Canvas.Rectangle(x,y,x+4,y+4);
          Ar[X,Y]:=Image.Canvas.Pixels[x,y];
        end;

      if (y = 1) and ((x mod SETKA)=0) then
        begin
          //Image.Canvas.Rectangle(x,y,x+4,y+4);
          Ar[X,Y]:=Image.Canvas.Pixels[x,y];
        end;

      if (y = 1) and (x = 1) then
        begin
          //Image.Canvas.Rectangle(x,y,x+4,y+4);
          Ar[X,Y]:=Image.Canvas.Pixels[x,y];
        end;
      end;
    end
  end;
end;
//-----
go:=true;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Label1.Caption:=inttostr(Image.Picture.Width);
  Label2.Caption:=inttostr(Image.Picture.Height);
end;

end.

```

Файл модулю роботи ПЗ frmLOG.pas

```
unit frmLOG;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ExtCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Button1: TButton;
    Memo1: TMemo;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses Unit1;

{$R *.dfm}

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

```
procedure TForm3.FormCreate(Sender: TObject);
begin

    Form3.Memo1.Lines.Add('Программа успешно стартовала: '+DateToStr(now));

end;

procedure TForm3.Button2Click(Sender: TObject);
begin

    Form3.Memo1.Lines.Clear;

end;

end.
```

Кафедра _ КБПЗ _ 2021 рік

Файл модулю зберігання відеокадрів (переадресація) frmLOG_GRAPH.pas

```

unit frmLOG_GRAPH;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    Button1: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;
implementation
{$R *.dfm}

end.

```

Файл настройки захоплення відеопотока frmSETTINGS.pas

```

unit frmSETTINGS;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

interface

```

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, ExtCtrls, StdCtrls, DSPack, DSUtil, DirectShow9;

type

```
TForm2 = class(TForm)
  Panel1: TPanel;
  Panel2: TPanel;
  Panel3: TPanel;
  Button1: TButton;
  VideoSourceFilter: TFilter;
  AudioSourceFilter: TFilter;
  AudioFormats: TListBox;
  Label4: TLabel;
  VideoFormats: TListBox;
  Label3: TLabel;
  AudioCapFilters: TListBox;
  Label2: TLabel;
  VideoCapFilters: TListBox;
  Label1: TLabel;
  InputLines: TComboBox;
  procedure Button1Click(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure VideoCapFiltersClick(Sender: TObject);
  procedure AudioCapFiltersClick(Sender: TObject);
  procedure VideoFormatsClick(Sender: TObject);
  procedure AudioFormatsClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form2: TForm2;
```

implementation

```
uses Unit1, frmLOG;
```

```
{$R *.dfm}
```

```
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('Установка
            '+AudioCapFilters.Items.Strings[VideoCapFilters.ItemIndex]);
    end;
end;

procedure TForm2.AudioCapFiltersClick(Sender: TObject);
```

```

var
  PinList: TPinList;
  g,i, LineIndex: integer;
  ABool: LongBool;

begin

  CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
  if AudioCapFilters.ItemIndex <> -1 then

    begin
      AudioSourceFilter.BaseFilter.Moniker :=
        CapEnum.GetMoniker(AudioCapFilters.ItemIndex);

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

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

        for i := 0 to AudioMediaTypes.Count - 1 do
          begin
            AudioFormats.Items.Add(AudioMediaTypes.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 IAMAudioInputMixer) do get_Enable(ABool);
            if ABool then LineIndex := i;
          end;

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

```

```
Form3.Memo1.lines.add('Установка '+AudioCapFilters.Items.Strings[g]);

end;
end;

procedure TForm2.VideoFormatsClick(Sender: TObject);
begin
  if VideoCapFilters.ItemIndex <> -1 then
  begin

Form3.Memo1.lines.add('Видео'+VideoCapFilters.Items.Strings[VideoCapFilters.Item
Index]);

end;
end;

procedure TForm2.AudioFormatsClick(Sender: TObject);
begin
  if AudioFormats.ItemIndex <> -1 then
  begin

Form3.Memo1.lines.add('Видео

'+AudioFormats.Items.Strings[AudioFormats.ItemIndex]);

end;
end;

end.
```

Файл модулю підключення плагінів frmPLUG.pas

```
unit frmPLUG;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm_9 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    DoT_A:TpluginsAddStrings;
    DoT_B:TpluginsAddFiles;
    DoT_C:TpluginsAddData;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form9: TForm9;

implementation

{$R *.dfm}

end.
```

Файл модулю впливаючого вікна програми frmSPLASH.pas

```
unit frmSPLASH;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//
```

```
interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls;

type
    TForm5 = class(TForm)
        Image1: TImage;
        ScrollBox1: TScrollBox;
    private
        { Private declarations }

    public
        { Public declarations }

    end;

var
    Form5: TForm5;

implementation

{$R *.dfm}

end.
```

Файл модулю авторських прав frmAbout.pas

```
unit frmAbout;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//
interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
    Buttons, ExtCtrls;
type
    TAboutBox = class(TForm)
        Panell: TPanel;
        ProgramIcon: TImage;
        ProductName: TLabel;
        Version: TLabel;
        Copyright: TLabel;
```

```

    Comments: TLabel;
    OKButton: TButton;

    procedure OKButtonClick(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
end;

var
    AboutBox: TAboutBox;

implementation
uses Unit1;
{$R *.dfm}
procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
    Form1.show;
    AboutBox.hide;
end;
end.

```

Файл бібліотеки захвату потоку відео DSPack.pas

```

unit DSPack;
//
// Copyright (C) Zaikin I.I.
// Central Ukrainian National Technical University. Kropyvnytskyi. 2021
// Diploms programs
//

Interface

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

const
    WM_GRAPHNOTIFY = WM_APP + 1;

    WM_CAPTURE_BITMAP = WM_APP + 2;

type

    TVideoMode = (
        vmNormal,
        vmVMR
    );

```

```

TGraphMode = (
    gmNormal,
    gmCapture,
    gmDVD
);

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

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

    { Specifies the seeking capabilities of a media stream. }
    TSeekingCap = (
        CanSeekAbsolute, // The stream can seek to an absolute position.
        CanSeekForwards, // The stream can seek forward.
        CanSeekBackwards, // The stream can seek backward.
        CanGetCurrentPos, // The stream can report its current position.
        CanGetStopPos, // The stream can report its stop position.
        CanGetDuration, // The stream can report its duration.
        CanPlayBackwards, // The stream can play backward.
        CanDoSegments, // The stream can do seamless looping (see
IMediaSeeking.SetPositions).
        Source // Reserved.
    );

    { Specifies the seeking capabilities of a media stream. }
    TSeekingCaps = set of TSeekingCap;

```

```

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

{ Pointer to @link(TVMRPreferences).}
PVMRPreferences = ^TVMRPreferences;
{ Set of @link(TVMRPreference).}
TVMRPreferences = set of TVMRPreference;

TOnDSEvent=procedure(sender: TComponent; Event, Param1, Param2: Integer) of
object;
{@exclude}
TOnGraphBufferingData=procedure(sender: TObject; Buffering: boolean) of object ;
{@exclude}
TOnGraphComplete=procedure(sender: TObject; Result: HRESULT; Renderer:
IBaseFilter) of object;
    TOnGraphDeviceLost = procedure(sender: TObject; Device: IUnknown;
Removed: Boolean) of object ;           {@exclude}
    TOnGraphEndOfSegment = procedure(sender: TObject; StreamTime:
TReferenceTime; NumSegment: Cardinal) of object ;           {@exclude}

TOnDSResult = procedure(sender: TObject; Result: HRESULT) of
object ;           {@exclude}
    TOnGraphFullscreenLost = procedure(sender: TObject; Renderer:
IBaseFilter) of object ;           {@exclude}
    TOnGraphOleEvent = procedure(sender: TObject; String1, String2:
WideString) of object ;           {@exclude}
    TOnGraphOpeningFile = procedure(sender: TObject; opening: boolean) of
object ;           {@exclude}
    TOnGraphSNDDDevError = procedure(sender: TObject; OccurWhen:
TSndDevErr; ErrorCode: LongWord) of object ;           {@exclude}
    TOnGraphStreamControl = procedure(sender: TObject; PinSender: IPin;
Cookie: LongWord) of object ;           {@exclude}
    TOnGraphStreamError = procedure(sender: TObject; Operation: HRESULT;
Value: LongWord) of object ;           {@exclude}
    TOnGraphVideoSizeChanged = procedure(sender: TObject; Width, height: word)
of object ;           {@exclude}
    TOnGraphTimeCodeAvailable= procedure(sender: TObject; From: IBaseFilter;
DeviceID: LongWord) of object ;           {@exclude}

```

```

TOnGraphEXTDeviceModeChange = procedure(sender: TObject; NewMode, DeviceID:
LongWord) of object ;                               {@exclude}
TOnGraphVMRRenderDevice= procedure(sender: TObject; RenderDevice:
TVMRRenderDevice) of object;

{@exclude}
TOnDVDAudioStreamChange      = procedure(sender: TObject; stream, lcid:
Integer; Lang: string) of object;                    {@exclude}
TOnDVDCurrentTime            = procedure(sender: TObject; Hours,
minutes,seconds,frames,frate : Integer) of object;  {@exclude}
TOnDVDTitleChange = procedure(sender: TObject; title: Integer) of object;
{@exclude}
TOnDVDChapterStart=procedure(sender: TObject; chapter: Integer) of object;
{@exclude}
TOnDVDValidUOPSChange = procedure(sender: TObject; UOPS: Integer) of object;
{@exclude}
TOnDVDChange = procedure(sender: TObject; total,current: Integer) of object;
{@exclude}
TOnDVDStillOn = procedure(sender: TObject; NoButtonAvailable: boolean;
seconds: Integer) of object;                         {@exclude}
TOnDVDSubpictureStreamChange = procedure(sender: TObject; SubNum, lcid:
Integer; Lang: string) of object;                    {@exclude}
TOnDVDPlaybackRateChange = procedure(sender: TObject; rate: single) of
object;                                               {@exclude}
TOnDVDParentalLevelChange= procedure(sender: TObject; level: Integer) of
object; {@exclude}
TOnDVDAnglesAvailable=procedure(sender: TObject; available: boolean) of
object; {@exclude}
TOnDVDButtonAutoActivated=procedure(sender: TObject; Button: Cardinal) of
object; {@exclude}
TOnDVDCMD=procedure(sender: TObject; CmdID: Cardinal) of object;
{@exclude}
TOnDVDCurrentHMSFTime = procedure(sender: TObject; HMSFTimeCode:
TDVDHMSFTimeCode; TimeCode: TDVDTimeCode) of object;  {@exclude}
TOnDVDKaraokeMode=procedure(sender: TObject; Played: boolean) of object;
{@exclude}
TOnBuffer = procedure(sender: TObject; SampleTime: Double; pBuffer: Pointer;
BufferLen: longint) of object ;

//*****
// IFilter
//*****

{@exclude}
TFilterOperation = (
foAdding,      // Before the filter is added to graph.

```

```

foAdded,      // After the filter is added to graph.
foRemoving,   // Before the filter is removed from graph.
foRemoved,    // After the filter is removed from graph.
foRefresh     // Designer notification to Refresh the filter .
);

{@exclude}
IFilter = interface
['{887F94DA-29E9-44C6-B48E-1FBF0FB59878}']
  function GetFilter: IBaseFilter;

  function GetName: string;

  procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
end;

{@exclude}
TControlEvent = (
  cePlay,
  cePause,
  ceStop,
  ceFileRendering,
  ceFileRendered,
  ceDVDRendering,
  ceDVDRendered,
  ceActive
);

{@exclude}
IEvent = interface
['{6C0DCD7B-1A98-44EF-A6D5-E23CBC24E620}']
  { FilterGraph events. }
  procedure GraphEvent(Event, Param1, Param2: integer);
  { Control Events. }
  procedure ControlEvent(Event: TControlEvent; Param: integer = 0);
end;

// *****
// TFilterGraph
//*****

TFilterGraph = class(TComponent)
private

```

```

FActive      : boolean;
FAutoCreate : boolean;
FHandle      : THandle; // to capture events
FMode        : TGraphMode;

FFilters: TInterfaceList;
FGraphEvents: TInterfaceList;

// builders
FFilterGraph : IGraphBuilder;
FCaptureGraph : ICaptureGraphBuilder2;
FDVDGraph    : IDvdGraphBuilder;

// events interface
FMediaEventEx : IMediaEventEx;

// Graphedit
FGraphEdit    : boolean;
FGraphEditID  : Integer;

// Log File
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;
FOnGraphSNDDevInError     : TOnGraphSNDDevError;
FOnGraphSNDDevOutError    : TOnGraphSNDDevError;
FOnGraphStepComplete      : TNotifyEvent;

```

```

FOnGraphStreamControlStarted      : TOnGraphStreamControl;
FOnGraphStreamControlStopped      : TOnGraphStreamControl;
FOnGraphStreamErrorStillPlaying   : TOnGraphStreamError;
FOnGraphStreamErrorStopped        : TOnGraphStreamError;
FOnGraphUserAbort                  : TNotifyEvent;
FOnGraphVideoSizeChanged           : TOnGraphVideoSizeChanged;
FOnGraphTimeCodeAvailable         : TOnGraphTimeCodeAvailable;
FOnGraphEXTDeviceModeChange       : TOnGraphEXTDeviceModeChange;
FOnGraphClockUnset                 : TNotifyEvent;
FOnGraphVMRRenderDevice            : TOnGraphVMRRenderDevice;

FOnDVDAudioStreamChange           : TOnDVDAudioStreamChange;
FOnDVDCurrentTime                  : TOnDVDCurrentTime;
FOnDVDTitleChange                  : TOnDVDTitleChange;
FOnDVDChapterStart                 : TOnDVDChapterStart;
FOnDVDAngleChange                  : TOnDVDChange;
FOnDVDValidUOPSCheckChange        : TOnDVDValidUOPSCheckChange;
FOnDVDButtonChange                 : TOnDVDChange;
FOnDVDChapterAutoStop              : TNotifyEvent;
FOnDVDStillOn                       : TOnDVDStillOn;
FOnDVDStillOff                      : TNotifyEvent;
FOnDVDSubpictureStreamChange       : TOnDVDSubpictureStreamChange;
FOnDVDNoFP_PGC                     : TNotifyEvent;
FOnDVDPlaybackRateChange           : TOnDVDPlaybackRateChange;
FOnDVDParentalLevelChange          : TOnDVDParentalLevelChange;
FOnDVDPlaybackStopped              : TNotifyEvent;
FOnDVDAnglesAvailable              : TOnDVDAnglesAvailable;
FOnDVDPlayPeriodAutoStop           : TNotifyEvent;
FOnDVDButtonAutoActivated           : TOnDVDButtonAutoActivated;
FOnDVDCMDStart                     : TOnDVDCMD;
FOnDVDCMDEnd                       : TOnDVDCMD;
FOnDVDDiscEjected                  : TNotifyEvent;
FOnDVDDiscInserted                 : TNotifyEvent;
FOnDVDCurrentHMSFTime              : TOnDVDCurrentHMSFTime;
FOnDVDDKaraokeMode                 : TOnDVDDKaraokeMode;
// DVD Warning
FOnDVDWarningInvalidDVD1_0Disc     : TNotifyEvent;//=1,
FOnDVDWarningFormatNotSupported    : TNotifyEvent;//=2,
FOnDVDWarningIllegalNavCommand     : TNotifyEvent;//=3
FOnDVDWarningOpen                   : TNotifyEvent;//=4
FOnDVDWarningSeek                   : TNotifyEvent;//=5
FOnDVDWarningRead                   : TNotifyEvent;//=6
// DVDDomain
FOnDVDDomainFirstPlay               : TNotifyEvent;
FOnDVDDomainVideoManagerMenu       : TNotifyEvent;
FOnDVDDomainVideoTitleSetMenu      : TNotifyEvent;

```

```

FOnDVDDomainTitle          : TNotifyEvent;
FOnDVDDomainStop           : TNotifyEvent;
// DVDError
FOnDVDErrorUnexpected      : TNotifyEvent;
FOnDVDErrorCopyProtectFail : TNotifyEvent;
FOnDVDErrorInvalidDVD1_0Disc : TNotifyEvent;
FOnDVDErrorInvalidDiscRegion : TNotifyEvent;
FOnDVDErrorLowParentalLevel : TNotifyEvent;
FOnDVDErrorMacrovisionFail : TNotifyEvent;
FOnDVDErrorIncompatibleSystemAndDecoderRegions : TNotifyEvent;
FOnDVDErrorIncompatibleDiscAndDecoderRegions : 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
  {@exclude}
  procedure DoEvent (Event, Param1, Param2: Integer); virtual;
  {@exclude}
  procedure InsertFilter (AFilter: IFilter);
  {@exclude}
  procedure RemoveFilter (AFilter: IFilter);
  {@exclude}
  procedure InsertEventNotifier (AEvent: IEvent);
  {@exclude}
  procedure RemoveEventNotifier (AEvent: IEvent);
public
  { Retrieve the total duration of a stream. }
  property Duration: Integer read GetDuration;
  { Retrieve/Set the rate. }

```

```

property Rate: Double read GetRate write SetRate;
{ Retrieve the seeking capabilities. }
property SeekCapabilities: TSeekingCaps read GetSeekCaps;
{ The volume balance. }
property Balance: integer read GetBalance write SetBalance;
{ The volume. }
property Volume: integer read GetVolume write SetVolume;
{ Current state of the filter graph. }
property State: TGraphState read GetState;
{ TFilterGraph constructor. }
constructor Create(AOwner: TComponent); override;
{ TFilterGraph destructor. }
destructor Destroy; override;
{ @exclude}
procedure Loaded; override;

function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
{ The Run method runs all the filters in the filter graph. While the graph
  is running, data moves through the graph and is rendered. }
function Play: boolean;
{ The Pause method pauses all the filters in the filter graph. }
function Pause: boolean;
{ The Stop method stops all the filters in the graph. }
function Stop: boolean;
{ This method disconnect all pins.}
procedure DisconnectFilters;
{ Disconnect and remove all filters from the filter graph excepting the
custom components. }
procedure ClearGraph;
{ Render a single file. }
function RenderFile(FileName: WideString): HRESULT;
function RenderFileEx(FileName: WideString): HRESULT;
{ Render a DVD Video Volume or a File Name if specified. }
function RenderDVD(out status: TAMDVDRenderStatus;
  FileName: WideString = ''; Mode: Integer = AM_DVD_HWDEC_PREFER): HRESULT;
{ Save the current state and position of a DVD movie to a file.<br>
  See also: @link(DVDRestoreBookmark).}
procedure DVDSaveBookmark(BookMarkFile: WideString);
{ Restore the State and position of a DVD movie saved by
@link(DVDSaveBookmark).}
procedure DVDRestoreBookmark(BookMarkFile: WideString);
published

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;

// -----
// Events
// -----

property OnActivate: TNotifyEvent read FOnActivate write FOnActivate;

property OnDSEvent: TOnDSEvent read FOnDSEvent write FOnDSEvent;

property OnGraphBufferingData: TOnGraphBufferingData read
FOnGraphBufferingData write FOnGraphBufferingData;

{ The reference clock has changed. The filter graph manager sends this event
  when its IMediaFilter.SetSyncSource method is called.}
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 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 OnDVDAudioStreamChange: TOnDVDAudioStreamChange read  
FOnDVDAudioStreamChange write FOnDVDAudioStreamChange;
```

```
property OnDVDCurrentTime: TOnDVDCurrentTime read FOnDVDCurrentTime write  
FOnDVDCurrentTime;
```

```
property OnDVDTitleChange: TOnDVDTitleChange read FOnDVDTitleChange write  
FOnDVDTitleChange;
```

```
property OnDVDChapterStart: TOnDVDChapterStart read FOnDVDChapterStart write  
FOnDVDChapterStart;
```

```
property OnDVDAngleChange: TOnDVDChange read FOnDVDAngleChange write  
FOnDVDAngleChange;
```

```
property OnDVDValidUOPSChange: TOnDVDValidUOPSChange read  
FOnDVDValidUOPSChange write FOnDVDValidUOPSChange;
```

```
property OnDVDButtonChange: TOnDVDChange read FOnDVDButtonChange write  
FOnDVDButtonChange;
```

```
property OnDVDChapterAutoStop: TNotifyEvent read FOnDVDChapterAutoStop write  
FOnDVDChapterAutoStop;
```

```
property OnDVDStillOn: TOnDVDStillOn read FOnDVDStillOn write FOnDVDStillOn;
```

```
property OnDVDStilloff: TNotifyEvent read FOnDVDStilloff write
FOnDVDStilloff;
```

```
property OnDVDSubpictureStreamChange: TOnDVDSubpictureStreamChange read
FOnDVDSubpictureStreamChange write FOnDVDSubpictureStreamChange;
```

```
property OnDVDNoFP_PGC: TNotifyEvent read FOnDVDNoFP_PGC write
FOnDVDNoFP_PGC;
```

```
property OnDVDPlaybackRateChange: TOnDVDPlaybackRateChange read
FOnDVDPlaybackRateChange write FOnDVDPlaybackRateChange;
```

```
property OnDVDParentalLevelChange: TOnDVDParentalLevelChange read
FOnDVDParentalLevelChange write FOnDVDParentalLevelChange;
```

```
property OnDVDPlaybackStopped: TNotifyEvent read FOnDVDPlaybackStopped write
FOnDVDPlaybackStopped;
```

```
property OnDVDAnglesAvailable: TOnDVDAnglesAvailable read
FOnDVDAnglesAvailable write FOnDVDAnglesAvailable;
```

```
property OnDVDPlayPeriodAutoStop: TNotifyEvent read FOnDVDPlayPeriodAutoStop
write FOnDVDPlayPeriodAutoStop;
```

```
property OnDVDButtonAutoActivated: TOnDVDButtonAutoActivated read
FOnDVDButtonAutoActivated write FOnDVDButtonAutoActivated;
```

```
property OnDVDCMDStart: TOnDVDCMD read FOnDVDCMDStart Write FOnDVDCMDStart;
```

```
property OnDVDCMDEnd: TOnDVDCMD read FOnDVDCMDEnd Write FOnDVDCMDEnd;
```

```
property OnDVDDiscEjected: TNotifyEvent read FOnDVDDiscEjected Write
FOnDVDDiscEjected;
```

```
property OnDVDDiscInserted: TNotifyEvent read FOnDVDDiscInserted write
FOnDVDDiscInserted;
```

```
property OnDVDCurrentHMSFTime: TOnDVDCurrentHMSFTime read
FOnDVDCurrentHMSFTime write FOnDVDCurrentHMSFTime;
```

```
{ Indicates that the Navigator has either begun playing or finished playing
karaoke data.<br>
```

```
The DVD player signals this event whenever it changes domains.<br>
```

```

    <b>Played:</b> TRUE means that a karaoke track is being played and FALSE
means
    that no karaoke data is being played. }
    property OnDVDKaraokeMode: TOnDVDKaraokeMode read FOnDVDKaraokeMode write
FOnDVDKaraokeMode;

    { Performing default initialization of a DVD disc.}
    property OnDVDDomainFirstPlay: TNotifyEvent read FOnDVDDomainFirstPlay write
FOnDVDDomainFirstPlay;

    { Displaying menus for whole disc. }
    property OnDVDDomainVideoManagerMenu: TNotifyEvent read
FOnDVDDomainVideoManagerMenu write FOnDVDDomainVideoManagerMenu;

    { Displaying menus for current title set. }
    property OnDVDDomainVideoTitleSetMenu: TNotifyEvent read
FOnDVDDomainVideoTitleSetMenu write FOnDVDDomainVideoTitleSetMenu;

    { Displaying the current title. }
    property OnDVDDomainTitle: TNotifyEvent read FOnDVDDomainTitle write
FOnDVDDomainTitle;

    { The DVD Navigator is in the DVD Stop domain.}
    property OnDVDDomainStop: TNotifyEvent read FOnDVDDomainStop write
FOnDVDDomainStop;

    property OnDVDErrorUnexpected: TNotifyEvent read FOnDVDErrorUnexpected write
FOnDVDErrorUnexpected;
    property OnDVDErrorInvalidDVD1_0Disc: TNotifyEvent read
FOnDVDErrorInvalidDVD1_0Disc write FOnDVDErrorInvalidDVD1_0Disc;

    property OnDVDErrorInvalidDiscRegion: TNotifyEvent read
FOnDVDErrorInvalidDiscRegion write FOnDVDErrorInvalidDiscRegion;

    property OnDVDErrorLowParentalLevel: TNotifyEvent read
FOnDVDErrorLowParentalLevel write FOnDVDErrorLowParentalLevel;

    property OnDVDErrorMacrovisionFail: TNotifyEvent read
FOnDVDErrorMacrovisionFail write FOnDVDErrorMacrovisionFail;

    property OnDVDErrorIncompatibleSystemAndDecoderRegions: TNotifyEvent read
FOnDVDErrorIncompatibleSystemAndDecoderRegions write
FOnDVDErrorIncompatibleSystemAndDecoderRegions;

```

```

    property OnDVDErrorIncompatibleDiscAndDecoderRegions: TNotifyEvent read
FOnDVDErrorIncompatibleDiscAndDecoderRegions write
FOnDVDErrorIncompatibleDiscAndDecoderRegions;

property OnDVDWarningInvalidDVD1_0Disc: TNotifyEvent read
FOnDVDWarningInvalidDVD1_0Disc write FOnDVDWarningInvalidDVD1_0Disc;

    property OnDVDWarningFormatNotSupported : TNotifyEvent read
FOnDVDWarningFormatNotSupported write FOnDVDWarningFormatNotSupported;

    property OnDVDWarningIllegalNavCommand : TNotifyEvent read
FOnDVDWarningIllegalNavCommand write FOnDVDWarningIllegalNavCommand;

    { File Open failed. }
    property OnDVDWarningOpen: TNotifyEvent read FOnDVDWarningOpen write
FOnDVDWarningOpen;

    { File Seek failed. }
    property OnDVDWarningSeek: TNotifyEvent read FOnDVDWarningSeek write
FOnDVDWarningSeek;

    { File Read failed. }
    property OnDVDWarningRead: TNotifyEvent read FOnDVDWarningRead write
FOnDVDWarningRead;
end;

//*****
// TVMROptions
//*****

{@exclude}
TVideoWindow = class;
{ See VRMOptions.<br>}
TVMRVideoMode = (
    vmrWindowed,
    vmrWindowless,
    vmrRenderless
);

{ Video Mixer Renderer property editor. }
TVMROptions = class(TPersistent)
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
  { Constructor method. }
  constructor Create(AOwner: TVideoWindow);
published
  { Windowed or WindowLess}
  property Mode: TVMRVideoMode read FMode write SetMode;
  { Sets the number of streams to be mixed. }
  property Streams: Cardinal read FStreams write SetStreams default 4;
  { Sets various application preferences related to video rendering. }
  property Preferences: TVMRPreferences read FPreferences write SetPreferences
default [vpForceMixer];
  { Keep Aspect Ration on the video window. }
  property KeepAspectRatio: boolean read FKeepAspectRatio write
SetKeepAspectRatio default True;
end;

//*****
//  TVideoWindow
//*****

TAbstractAllocator = class(TInterfacedObject)
  constructor Create(out hr: HRESULT; wnd: THandle; d3d: IDirect3D9 = nil;
d3dd: IDirect3DDevice9 = nil); virtual; abstract;
end;
TAbstractAllocatorClass = class of TAbstractAllocator;

{ Manage a Video Renderer or a Video Mixer Renderer (VMR) Filter to display
a video in your application. }
TVideoWindow = class(TCustomControl, IFilter, IEvent)
private
  FMode          : TVideoMode;
  FVMROptions    : TVMROptions;
  FBaseFilter    : IBaseFilter;
  FVideoWindow   : IVideoWindow; // VMR Windowed & Normal
  FWindowLess    : IVMRWindowlessControl9; // VMR Windowsless

  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}
  procedure MouseMove (Shift: TShiftState; X, Y: Integer); override;
  {@exclude}
  procedure MouseUp (Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
  {@exclude}
  procedure Paint; override;
public
  {@exclude}

```

```

function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
{ Constructor. }
constructor Create(AOwner: TComponent); override;
{ Destructor. }
destructor Destroy; override;
{ Check if the Video Mixer Renderer is available (Windows XP). }
class function CheckVMR: boolean;
{ Retrieve the current bitmap, only in WindowLess VMR Mode. }
function VMRGetBitmap(Stream: TStream): boolean;
function CheckInputPinsConnected: boolean;
procedure SetAllocator(Allocator: TAbstractAllocatorClass; UserID:
Cardinal);
published
    property OnPaint: TNotifyEvent read FOnPaint write FOnPaint;
    { The video Window stay on Top in FullScreen Mode. }
    property FullScreenTopMost: boolean read FTopMost write SetTopMost default
false;
    {Video Mode, you can use Normal mode or VMR mode (VMR is only available on
WindowsXP). }
    property Mode: TVideoMode read FMode write SetVideoMode default vmNormal;
    { The @link(TFilterGraph) component }
    property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
    { Return the Handle where the video is displayed. }
    property VideoHandle: THandle read GetVideoHandle;
    { Video Mixer Renderer property editor. }
    property VMROptions: TVMROptions read FVMROptions write FVMROptions;

    property FullScreen: boolean read FFullScreen write SetFullScreen default
false;
    { Common properties & Events }
    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;           {@exclude}
property OnClick;               {@exclude}
property OnConstrainedResize;   {@exclude}
property OnDbClick;            {@exclude}
property OnMouseDown;          {@exclude}
property OnMouseMove;          {@exclude}
property OnMouseUp;            {@exclude}
property OnMouseWheel;         {@exclude}
property OnMouseWheelDown;     {@exclude}
property OnMouseWheelUp;       {@exclude}
property OnResize;

end;

//*****
// TFilterSampleGrabber declaration
//*****

{@exclude}
TSampleGrabber = class;

{ This class is designed make a snapshot of Video or Audio Datas.
WARNING: There is know problems with some DIVX movies, so use RGB32 Media
Type
instead of RBG24.}
TSampleGrabber = class(TComponent, IFilter, ISampleGrabberCB)
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
  {@exclude}
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public
  { ISampleGrabber Interface to control the SampleGrabber Filter.

```

```

    The FilterGraph must be active.}
SampleGrabber: ISampleGrabber;
{ The Input Pin.
    The FilterGraph must be active.}
InPutPin : IPin;
{ The Output Pin.
    The FilterGraph must be active.}
OutPutPin : IPin;
{ Constructor method. }
constructor Create(AOwner: TComponent); override;
{ Destructor method. }
destructor Destroy; override;
{ Configure the filter to capture the specified MediaType.
    This method disconnect the Input pin if connected.
    The FilterGraph must be active. }
procedure UpdateMediaType;
{@exclude}
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;
{ This method read the current buffer from the Sample Grabber Filter and
paint the bitmap.}
function GetBitmap(Bitmap: TBitmap): boolean; overload;
{ This method check if the Sample Grabber Filter is correctly registered on
the system. }

class function CheckFilter: boolean;
published

property OnBuffer: TOnBuffer read FOnBuffer write FOnBuffer;
{ The filter must connected to a TFilterGraph component.}
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;

property MediaType: TMediaType read FMediaType write FMediaType;
end;
//*****
// TFilter
//*****

TFilter = class(TComponent, IFilter)
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
  { Constructor method. }
  constructor Create(AOwner: TComponent); override;
  { Destructor method. }
  destructor Destroy; override;
  { Retrieve a filter interface. }
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published
  { This is the Filter Editor .}
  property BaseFilter: TBaseFilter read FBaseFilter write FBaseFilter;
  { The filter must be connected to a TFilterGraph component.}
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
end;

//*****
// TASFWriter
//*****

{ This component is designed to create a ASF file or to stream over a
network.}
TASFWriter = class(TComponent, IFilter)
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
  {@exclude}
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
public
  { Sink configuration. }
  WriterAdvanced2      : IWMWriterAdvanced2;
  { NetWork streaming configuration. }
  WriterNetworkSink    : IWMWriterNetworkSink;
  { The Audio Input Pin. }
  AudioInput           : IPin;
  { The Video Input Pin. }
  VideoInput           : IPin;
  { Audio Input configuration. }
  AudioStreamConfig    : IAMStreamConfig;
  { VideoInput configuration }
  VideoStreamConfig    : IAMStreamConfig;
  { Destructor method. }
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;
  {@exclude}
  function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
published
  { The filter must be connected to a TFilterGraph component.}
  property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
  { Windows media profile to use. }
  property Profile: TWMPofiles8 read GetProfile write SetProfile;
  { Destination file name to write the compressed file. }
  property FileName: String read GetFileName write SetFileName;
  { Port number to stream.}
  property Port: DWORD read FPort write FPort;
  { The max number of connections. }
  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;

```

```

//*****
//  TDSTackBar
//*****
    {@exclude}
    TTimerEvent = procedure(sender: TObject; CurrentPos, StopPos: Cardinal) of
object ;

{ This control implement a seek bar for a media-player application.
  The seek bar is implemented as a TTrackbar control. }
TDSTackBar = class(TTrackBar, IEvent)
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
    { constructor method. }
    constructor Create(AOwner: TComponent); override;
    { destructor method. }
    destructor Destroy; override;
published

```

```

    { Select the filtergraph to seek. }
    property FilterGraph: TFilterGraph read FFilterGraph Write SetFilterGraph;
    { Select the time interval in miliseconds. default = 1000 mls. }
    property TimerInterval: Cardinal read FInterval write SetInterval default
1000;

    property OnTimer: TTimerEvent read FOnTimer write SetOnTimer;
end;

{ @exclude }
TDSVideoWindowEx2 = class;

//*****
// TColorControl
//*****

TColorControl = class(TPersistent)
private
    FBrightness : Integer;
    FContrast   : Integer;
    FHue        : Integer;
    FSaturation : Integer;
    FSharpness  : Integer;
    FGamma      : Integer;
    FUtilColor  : Boolean;
    FDefault    : TDDColorControl;
protected
    { Protected declarations }
    { @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;
    { @exclude }
function GetContrast : Integer;
    { @exclude }
function GetHue : Integer;
    { @exclude }
function GetSaturation : Integer;
    { @exclude }
function GetSharpness : Integer;
    { @exclude }
function GetGamma : Integer;
    { @exclude }
function GetUtilColor : Boolean;
    { @exclude }
Procedure ReadDefault;
    { @exclude }
procedure UpdateColorControls;
    { @exclude }
procedure GetColorControls;
public
    { Public declarations }
    { @exclude }
    constructor Create(AOwner: TDSVideoWindowEx2); virtual;

procedure RestoreDefault;
    Published
        property Brightness : Integer read GetBrightness write SetBrightness;

        { The
        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;

//*****
// TDSVideoWindowEx2Caps
//*****

```

```

TDSVideoWindowEx2Caps = class(TPersistent)
protected
  { Protected declarations }
  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 declarations }
  { @exclude }
  constructor Create(AOwner: TDSVideoWindowEx2); virtual;
published
  { if CanOverlayGraphics return true, you draw on DSVideoWindowEx's canvas
and the
  graphic will bee ontop of the Video.}
  Property CanOverlayGraphic : Boolean read GetCanOverlay;

  { Repport if you can control Brightness on the video overlay }
  Property CanControlBrigtness : Boolean read GetCanControlBrigtness;
  { Repport if you can control Contrast on the video overlay }
  Property CanControlContrast : Boolean read GetCanControlContrast;
  { Repport if you can control Hue on the video overlay }
  Property CanControlHue : Boolean read GetCanControlHue;
  { Repport if you can control Saturation on the video overlay }
  Property CanControlSaturation : Boolean read GetCanControlSaturation;
  { Repport if you can control Sharpness on the video overlay }
  Property CanControlSharpness : Boolean read GetCanControlSharpness;
  { Repport if you can control Gamma on the video overlay }
  Property CanControlGamma : Boolean read GetCanControlGamma;
  { Repport if you can control ColorEnabled on the video overlay }
  Property CanControlColorEnabled : Boolean read GetCanControlUtilizedColor;
end;

// *****
// TOverlayCallback
//*****

  { @exclude }
TOverlayCallback = class(TInterfacedObject, IDDrawExclModeVideoCallBack)
  AOwner : TObject;

```

```

    constructor Create(Owner : TObject); virtual;
    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;

//*****
// TDSVideoWindowEx2
//*****

{ @exclude }
TRatioModes = (rmStretched, rmLetterBox, rmCrop);

{ @exclude }
TOverlayVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;

{ @exclude }
TCursorVisibleEvent = procedure (Sender: TObject; Visible : Boolean) of
object;

TDSVideoWindowEx2 = class(TCustomControl, IFilter, IEvent)
private
    FVideoWindow      : IVideoWindow;
    FFilterGraph      : TFilterGraph;
    FBaseFilter       : IBaseFilter;
    FOverlayMixer     : IBaseFilter;
    FVideoRenderer    : IBaseFilter;
    FDDXM             : IDDrawExclModeVideo;
    FFullScreen       : Boolean;
    FTopMost          : Boolean;
    FColorKey         : TColor;
    FWindowState     : LongWord;
    FWindowStateEx   : 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;
  {@exclude}
  procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
  {@exclude}
  procedure WndProc(var Message: TMessage); override;
  {@exclude}
  procedure Paint; override;
  {@exclude}

```

```

    procedure MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
    { @exclude }
    procedure MouseMove(Shift: TShiftState; X, Y: Integer); override;
    { @exclude }
    procedure MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
override;
    { @exclude }
    procedure MyIdleHandler(Sender: TObject; var Done: Boolean);
    { @exclude }
    procedure RefreshVideoWindow;
public
    { constructor method. }
    constructor Create(AOwner: TComponent); override;
    { destructor method. }
    destructor Destroy; override;

    { @exclude }
    function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;

    { Clear the graphic ontop of DSVideoWindowEx. }
    procedure ClearBack;

    { Use your Desktop as the Video renderer.
    The video will display as a "wallpaper" on your Desktop }
    procedure StartDesktopPlayback; overload;

    { Use your Desktop as the Video renderer.
    The video will display as a "wallpaper" on your Desktop on the
    specified monitor }
    procedure StartDesktopPlayBack(OnMonitor : TMonitor); overload;

    { Return to normal window playback from Fullscreen or Desktop mode. }
    procedure NormalPlayback;

    { Start playback in fullscreen }
    procedure StartFullScreen; overload;

    { Start playback in fullscreen on specified Monitor }
    procedure StartFullScreen(OnMonitor : TMonitor); overload;

    { reporting if you are currently playing in fullscreen. }
    property FullScreen: boolean read FFullScreen;

```

```

{ reporting if you are currently playing on the Desktop. }
property DesktopPlayback : Boolean read FDesktopPlay;
{ @inherited }
property Canvas;

property ColorKey : TColor read FColorKey;

{ @link(TDSVideoWindowEx2Caps) }
property Capabilities : TDSVideoWindowEx2Caps read FCaps;

property OverlayVisible : Boolean read FOverlayVisible;
published

property AspectRatio : TRatioModes read FAspectMode write SetAspectMode;

{ Set the amounts of milliseconds before the cursor is hidden, if it is not
moved.
Setting the value to 0 will disable this feature. }
property AutoHideCursor : Integer read FIdleCursor write FIdleCursor;

{ Specify a Zoom factor from 0 to 99 percent. }
property DigitalZoom : Integer read FZoom write SetZoom;

{ The @link(TFilterGraph) component }
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;

{ Select if the VideoWindow is topmost or not. }
property FullScreenTopMost: boolean read FTopMost write SetTopMost default
false;

{ Event to tell the main application that the Colorkey has changed.
Note: if you have controls placed on top of your VideoWindow that need to
act as
transparent, set their color to the same as the Colorkey.}
property OnColorKeyChanged: TNotifyEvent read FOnColorKey write FOnColorKey;

{ @link(TColorControl) }
property ColorControl : TColorControl read FColorControl write
FColorControl;

{ Setting this to true will prevent the screen from going into screensaver or
powerdown. }
property NoScreenSaver : Boolean read FNoScreenSaver write FNoScreenSaver;

```



```

);
TVMRBitmapOptions = set of TVMRBitmapOption;

TVMRBitmap = class
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

    constructor Create(VideoWindow: TVideoWindow);
    // Cleanup
    destructor Destroy; override;

    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);
    // update the video window with the current bitmap
    procedure Update;
    // Uses this property to draw on the internal bitmap.

```

```
property Canvas: TCanvas read FCanvas write FCanvas;
// Change Alpha Blending
property Alpha: Single read GetAlpha write SetAlpha;
// set the source rectangle
property Source: TRect read GetSource write SetSource;
// Destination Left
property DestLeft : Single read GetDestLeft write SetDestLeft;
// Destination Top
property DestTop : Single read GetDestTop write SetDestTop;
// Destination Right
property DestRight : Single read GetDestRight write SetDestRight;
// Destination Bottom
property DestBottom : Single read GetDestBottom write SetDestBottom;
// Destination
property Dest: TVMR9NormalizedRect read GetDest write SetDest;
// Set the color key for transparency.
property ColorKey: COLORREF read GetColorKey write SetColorKey;
// VMR Bitmap Options.
property Options: TVMRBitmapOptions read FOptions write SetOptions;
end;
implementation
uses ComObj;
End.
```