

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 відеонагляду
реалізованої на базі Axis P1364-E та Axis P1365-E Mk II”**

Виконав здобувач вищої освіти
II курсу, групи КІ-20М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»

Сергій В.В.

« ___ » _____ 2021 р.

Керівник проекту

кандидат фізико-математичних наук, доцент

Володимир ПЕТРЕНЮК

« ___ » _____ 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 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

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

Сергатову Владиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II

2. Керівник роботи Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 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 відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II

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

Сергати́й В.В. Дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Метою розробки є дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Об'єктом дослідження є процес відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Предметом дослідження є методи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Методи дослідження базуються на методах теорії обробки мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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 Delphi 10.3.2 Rio.

Ключові слова: комп'ютерна інженерія, відео нагляд, Axis P1364-E, Axis P1365-E Mk II

ABSTRACT

Serhatyi V.V. Research and software implementation of video surveillance system implemented on the basis of Axis P1364-E and Axis P1365-E Mk II. 123 Computer Engineering. 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 the video surveillance system realized on the basis of Axis P1364-E and Axis P1365-E Mk II is developed.

The purpose of development is research and software implementation of video surveillance system implemented on the basis of Axis P1364-E and Axis P1365-E Mk II.

The object of the study is the process of video surveillance implemented on the basis of Axis P1364-E and Axis P1365-E Mk II.

The subject of the research are video surveillance methods implemented on the basis of Axis P1364-E and Axis P1365-E Mk II.

Research methods are based on the methods of the theory of multimedia data processing, methods of mathematical statistics, methods of software development.

The result is a software implementation of a video surveillance system implemented on the basis of Axis P1364-E and Axis P1365-E Mk II.

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 an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment Delphi 10.3.2 Rio.

Keywords: computer engineering, video surveillance, Axis P1364-E, Axis P1365-E Mk II

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	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	23
2.3 Розгорнута постановка завдання	27
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	29
3.1 Опис функціонування системи.....	29
3.2 Розробка структурної схеми	41
3.3 Розробка функціональної схеми.....	48
3.4 Розробка діаграми процесів	50
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	52
4.2 Захист розробленого програмного забезпечення	70
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	72
6 НАУКОВА НОВИЗНА	77

ВКРМ-123.21.0018.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Сергачий В.В.			Дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II	Лім.	Аркуш	Аркушів
Перев.		Петренко В.І.				М	1	116
Н.контр.		Гермак В.С.			ЦНТУ КІ-20М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	78
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	78
7.2 Розрахунок трудомісткості розробки програмної продукції	80
7.3 Визначення чисельності виконавців і планового фонду зарплати	82
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	87
7.5 Визначення собівартості розробки та ціни програмної продукції.	91
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	94
7.7 Визначення експлуатаційних витрат	95
7.8 Визначення економічної ефективності програмної продукції.....	96
7.9 Висновок.	98
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	99
8.1 Вступ	99
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	100
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста	101
8.4 Розробка заходів з умов поліпшення охорони праці.....	104
8.5 Розрахункова частина	105
8.6 Висновки до розділу.....	107
9 ОСНОВНІ ВИСНОВКИ.....	108
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	110

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БПД	–	бездротова передача даних
ПЗ	–	програмне забезпечення
СПД	–	системи передачі даних
ACK	–	повідомлення підтвердження прийому
ARQ	–	протокол повторної передачі даних
BPSK	–	Binary phase-shift keying
FFD	–	повнофункціональний пристрій
GFSK	–	Gaussian frequency-shift keying
MAC	–	шар механізму доступу
NACK	–	повідомлення непідтвердження прийому
OSI	–	мережна модель
P2P	–	однорангові мережі
PAN	–	персональна мережа
PPS	–	Portable Protocol Stack
RFD	–	пристрій з полегшеними функціями
TDMA	–	часовий поділ
Wi-Fi	–	бездротова технологія

ВСТУП

Актуальність теми. Камери Axis лінійки P13 у вуличному виконанні розраховані на роботу в діапазоні температур від -40 до +50 0 С і широко застосовуються. Усього дана лінійка включає 9 моделей камер. Недавно компанія заявила про значне розширення асортиментів камер, що збираються в Україні – до 16 моделей, що випускаються раніше, додалося ще 10, причому дві з них з лінійки камер P13 (Axis P 1364-E й Axis P 1365-E Mk II).

Всі використовувані технічні засоби для її забезпечення, у тому числі камери відеоспостереження, підлягають обов'язковій сертифікації. Сертифікації підлягають як впроваджені засоби, так і вже встановлені.

До камер відеоспостереження пред'являється цілий ряд функціональних і технічних вимог у частині розв'язної здатності, оптичних характеристик, ступеня стиску зображення, частоти кадрів і т.д. Більшість сучасних камер для цього ринку легко їм задовольняють, зокрема, мінімальний припустима розв'язна здатність становить 1,2 Мпкселів. Дотримання даних характеристик повинне підвищити безпеку, про при цьому захищеність самих камер відеоспостереження ніяк не регламентується. Якщо відносно систем контролю доступу говориться, що вони повинні забезпечувати захист «технічних і програмних засобів від несанкціонованого доступу ... у вигляді системи паролів», то до відеокамер не пред'являється навіть цих елементарних вимог.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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

– Огляд існуючих систем відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Дослідження системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

– Програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Об'єктом дослідження є процес відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Предметом дослідження є методи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Методи дослідження базуються на методах теорії обробки мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

– Розроблено вітчизняний продукт відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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

Робота апробована на 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.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У силу географічного положення України на багатьох об'єктах інфраструктури тяжкі умови експлуатації. Це накладає високі вимоги до якості пристроїв, їх відказостійкості пристроїв і захищеності від перепадів температури, вологи, пилу. Axis приділяє велику увагу якості камер і проводить по усьому світі випробування свого встаткування в реальних умовах відеоспостереження. У режимі реального часу виробляється збір даних про навколишнє середовище й стан камер для наступного аналізу. Отримані дані дозволяють поліпшити експлуатаційні характеристики камер.

Визначаються вимоги в тому числі й до інтелектуального відеоспостереження в частині ідентифікації фізичних осіб і виявлення тривожних ситуацій. (однак наявність таких систем не є обов'язковим – у всякому разі вимоги до необхідності їхньої установки встановлюються). Камери Axis оснащуються власним чипом Axis ARTPEC (цього року з'явилася вже шоста версія цього процесора). Він використовується для завантаження додаткових застосунків, за допомогою яких камера перетворюється в пристрій по аналітичній обробки. Так, убудована аналітика Axis Perimeter Defender убудована в камери відеоспостереження Axis дозволяє ідентифікувати факт проникнення в охоронювану зону.

Камери використовуються як для розслідування інцидентів постфактум, так і для моніторингу поточної обстановки, у тому числі виявлення залишених без догляду речей. Висока світлочутливість камер і підтримка розв'язної здатності HDTV 1080p із частотою до 60 кадрів у секунду дозволяє реалізувати відеоаналітику в умовах щільного пасажирського потоку.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Застосування сучасних систем відеоспостереження не обмежується режимними об'єктами. Системи безпеки даного виду тепер широко застосовуються в самих різних сферах.

Як правило, такі системи підрозділяються на різні групи виходячи з області їхнього подальшого використання:

– Системи можуть бути як зовнішнього використання, так і внутрішнього. Природно, зовнішні камери, розташовані поза приміщеннями, повинні бути несприйнятливими до вологості, низьким і високим температурам, а також іншим агресивним факторам. Важливим в таких випадках є антивандальний захист.

– Для внутрішніх систем не висувається настільки твердих вимог. Як правило, вони використовуються в офісах, торгових центрах і місцях суспільного користування.

– У квартирах, будинках і дачах поряд із уже звичними відеодомофонами часто встановлюють зовнішні системи для відеоспостереження.

– Сховані системи призначені для неявного візуального контролю над об'єктом.

Основні завдання системи відеоспостереження

У сучасних умовах складно уявити собі систему безпеки якого-небудь об'єкта без використання відеоспостереження. Установка комплексної системи відеоспостереження на охоронюваному об'єкті дозволяє ефективно виконувати цілий ряд важливих функцій.

– Територія об'єкта й прилягаюча до неї область перебуває під постійним візуальним контролем.

– З'являється можливість оперативного контролю за всіма переміщеннями (санкціонованими й несанкціонованими) по території об'єкта, а також за робочим процесом на підприємствах.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

– Система відеоспостереження дозволяє фіксувати спроби несанкціонованого проникнення на об'єкт, а в комплексі з іншими системами й припиняти такі спроби.

– Такі системи дають можливість здійснювати не тільки оперативний візуальний контроль, але й проводити фіксацію подій для архіву.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Відеоспостереження вже давно використовується для рішення різних завдань. В 60-х роках це були аналогові закриті системи відеоспостереження, назва CCTV (closed circuit television – замкнута телевізійна система) розкриває зміст поняття.

Розвиток не стояв на місці й в 90 -х роках, а саме в 1996 була розроблена перша у світі цифрова IP відеокамера компанією Axis Communication. Технічні характеристики камери не здавалися настільки грандіозними, і, у порівнянні з аналоговими, камери виглядали блідо. Незважаючи на це, перших IP камер було продано порядку 10000 штук.

Вихід IP камер на ринок послужив відправною крапкою для розвитку систем відеоспостереження, поява камери вимагало появи програмного забезпечення для роботи с їй.

Згодом з'являлися нові IP камери відеоспостереження, разом ними й програмне забезпечення. Розвиток пішов двома шляхами: частина компаній зайнялася розробкою софта для роботи з будь-якими IP камерами, друга частина – виробництвом власних камер і софта для роботи винятково зі своїми камерами.

Такий підхід у розвитку розділив системи відеоспостереження на «відкриті» і «закриті». Відкриті, Open Platform системи розвиваються у бік багатовендорності – підтримці камер незалежно від виробника, і закриті – одновендорні системи, що підтримують роботу тільки з устаткуванням власного виробництва.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Однак зараз проглядається тенденція до того, що закриті системи починають дивитися у бік підтримки встаткування сторонніх виробників, але їм усе ще далеко до того, щоб називатися відкритими платформами.

Додатково варто згадати про відеоаналітику. На сьогоднішній день це невід'ємна частина систем відеонагляду (СВН), що стрімко розвивається, здатна полегшити й допомогти операторам не втратитися у величезному потоці відеоінформації. А також вирішувати бізнес-завдання й у цілому перетворювати відео в дані, якими можна з легкістю оперувати. Тому сучасна й комплексна система відеоспостереження повинна підтримувати роботу з аналітикою, підтримувати інтеграцію із системою контролю доступу, керування будинком.

Відкритих платформ відеоспостереження на ринку представлено досить. Щоб небагато розібратися в них, проведемо невеликий огляд деяких СВН, а більше детальну інформацію з характеристик можна побачити в таблиці порівняння VMS.

MileStone Systems. XProtect

(www.milestonesys.com)

Компанія Milestone займається розробкою ПЗ для IP систем відеоспостереження з кінця 90-х років. Через джерела своєї діяльності тісно співробітничав з компанією Axis Communication, лідером в області виробництва пристроїв відеоспостереження.

Протягом 10 років компанія Milestone є лідером на ринку СВН за результатами IHS Markit, а також засновником поняття відкритої платформи у відеоспостереженні.

Продуктова лінійка представлена групою програмного забезпечення XProtect, сім версій і лінійкою мережних реєстраторів Husky.

Продуктова лінійка покриває різні потреби до систем відеоспостереження: від простих і невеликих систем, до складних, географічно розподілених без обмеження по кількості камер на сервер або систему в цілому.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Для ознайомлення із програмним продуктом є можливість скачати ознайомлювальну версію будь-якої версії або ж використовувати зовсім безкоштовно молодшу версію продукту Essential+ на 8 пристроїв, без обмеження з розв'язної здатності камер, що підключаються користувачам, періоду зберігання й не тільки.

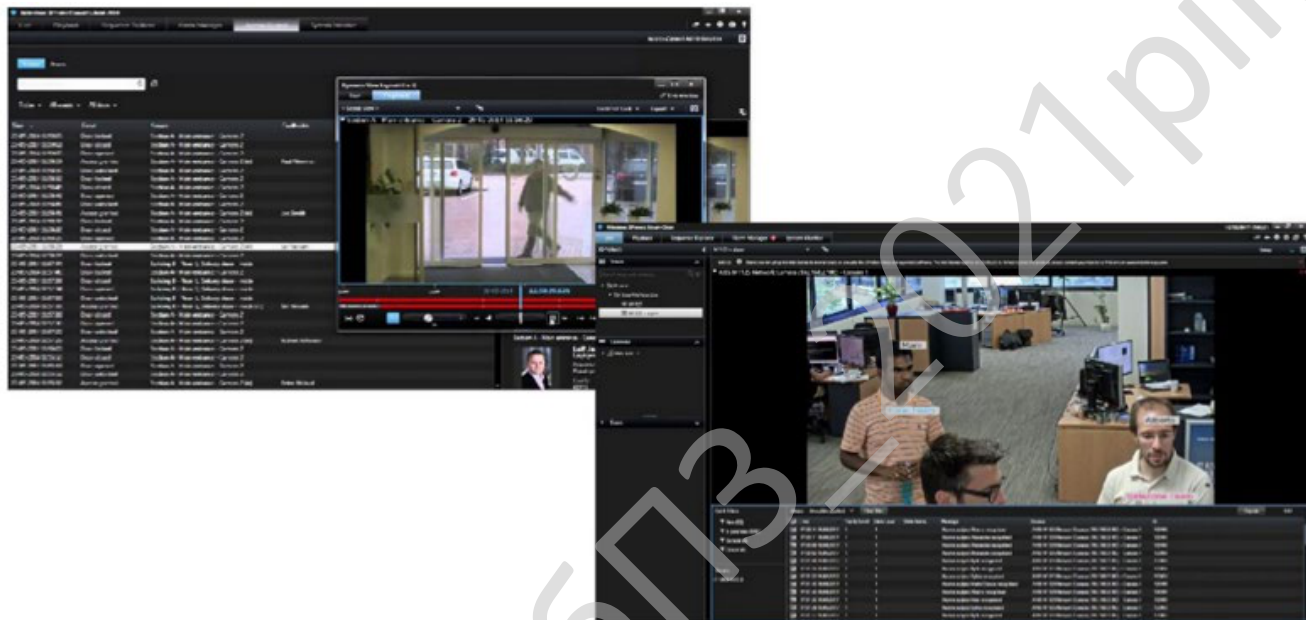


Рисунок 2.1 – Інтерфейс користувача Milestone XProtect

Milestone підтримує найбільша кількість IP пристроїв на рівні драйвера й зараз це більше 7000 перевірених моделей різних виробників і кількість готових інтеграцій – відеоаналітика (захист периметра, розпізнавання осіб, розпізнавання номерних знаків, рішень для паркінгу, бізнес аналітика, аналітика звуку), системи контролю доступу, системи керування будинками й т.д.

Для кастомізації функціонала є можливість використовувати безкоштовний SDK у всіх версіях і/або конфігураційний API у старшій версії.

Milestone має дуже просту систему ліцензування. Ліцензування складається з однієї базової ліцензії на ПЗ, без прив'язки до заліза сервера й ліцензії на пристрій. Ліцензування пристрою не обмежує його функціонал і

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

дозволяє використовувати, «вичавлювати» із пристрою максимум: кількість потоків (каналів), порти I/O, підтримка звуку, робота з картою пам'яті, убудована аналітика. Базова ліцензія на софт не обмежується однією інсталяцією й може бути встановлена на різні локації, сайти без обмеження. Повторне підключення пристрою в межах однієї базової ліцензії не вимагає додаткового ліцензування, що дає можливість здійснювати подвійний запис, наприклад. Підключення реєстраторів в Milestone XProtect у більшості випадків вимагає однієї ліцензії, у такий спосіб можна підключити NVR до 64-х каналів. Єдиним недоліком у роботі з реєстраторами є відсутність прямого доступу до архіву на реєстраторі.

В Milestone XProtect відсутнє обмеження на кількість користувачів і одночасних підключень (одна лише версія обмежується п'ятьма клієнтськими підключеннями). Підтримує три типи користувачів: пропрієтарний, Windows і Active Directory. Для зручності й гнучкості роботи зі СВН, у наявності три абсолютно безкоштовних клієнти: товстий клієнт, mobile і web клієнт, що дозволяють працювати з живим, архівним відео, реагувати на тривоги в системі. За допомогою мобільного клієнта можна перетворити пристрій з камерою в IP камеру й транслювати відео з можливістю запису в архів.

Для конфігурації системи використовується додаткове ПЗ, є клієнт-серверним і може використовуватися де-небудь у мережній доступності до сервера. Milestone не розробляє власної аналітики. Ідеологія компанії полягає в створенні зручного й відкритого інструмента для інтеграції, роботи з відео: перегляду, запису, забезпечення його схоронності й дійсності, а розробкою аналітики повинні займатися профільні компанії, у такий спосіб у портфелі Milestone величезна кількість технологічних партнерів і рішень по інтеграції.

Для розширення функціонала Milestone XProtect у своєму портфелі має кілька модулів, це Transact – для роботи із транзакціями (текстовою інформацією) і зіставлення їх з відео, LPR – система розпізнавання номерних знаків, Retail – модуль інтеграції з ERP системою, ACS – інтеграція із системами контролю доступу.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

СВН Milestone XProtect розроблений під використання в середовищі Windows, як серверна, так і клієнтська частина (товстий клієнт). Для зберігання конфігурації й різної служебної інформації використовує MS SQL, за замовчуванням це версія Express. Milestone підтримує роботу у віртуальному середовищі.

Genetec. Omnicast

(www.genetec.com)

Компанія Genetec веде свою діяльність із кінця 90-х років і є одним з лідерів ринку систем відеоспостереження.

Genetec – розроблювач уніфікованої платформи безпеки. Основними продуктами є Security Center і Omnicast. Останній продукт є безпосередньо системою відеоспостереження, а перший – верхньою крапкою, агрегатором додаткових компонентів системи, і деякі можливості системи Omnicast розкриваються при використанні Security Center.

І так, Omnicast представлений трьома версіями, що мають різні обмеження по кількості камер у системі, користувачів і інших функціональних можливостей.

Genetec здатний закрити різні потреби до системи відеоспостереження: від простих і невеликих систем, до складних, географічно розподілених, без обмеження по кількості камер на сервер або систему в цілому, але з обмеженням на потік сервера запису в 200 Мб/с, тому при інсталяціях з більшою кількістю камер, готуйтеся до великої кількості серверів запису.

Безкоштовної версії в Genetec не існує, тому єдиний варіант ознайомлення із програмним продуктом – запит демо версії. Genetec Omnicast підтримує порядку 6000 IP пристроїв і велика кількість інтегрованих рішень по відеоаналітиці (захист периметра, розпізнавання осіб, розпізнавання номерних знаків, бізнес аналітика) системи контролю доступу, системи керування будинками й т.д. Інформацію про підтримку пристроїв і пошук інтегрованих рішень можна знайти на сайті Genetec, через те, що підтримка, приміром, може бути можлива тільки в Security Center, а в Omnicast відсутня.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

можливість переглядати й працювати з відео з будь-якої крапки миру, але й використовувати камеру пристроєм мобільного як IP камеру.

Власні аналітичні модулі Genetec не розробляє, при цьому є присутнім широкий вибір партнерів по інтеграції різного роду відеоаналітики.

Функціонал Genetec може бути розширений власними модулями, це – система розпізнавання номерів AutoVu і системи контролю доступу Synergis. Використання даних компонентів має на увазі наявність Security Center.

Також Genetec випускає програмно-апаратні комплекси Streamvault.

Genetec Omnicast розроблений під Windows, як серверна, так і клієнтська частина. Для налаштування системи використовується окреме програмне забезпечення. Для зберігання різної службової інформації використовує MS SQL, підтримується робота у віртуальному середовищі.

Luxriot. Evo

(www.luxriot.com)

Luxriot – американська компанія по розробці програмного забезпечення для систем відеоспостереження, крім софта пропонує сервера із передвстановленим ПЗ, плати захвата для аналогових систем відеоспостереження.

Luxriot представлений на ринку трьома версіями програмного забезпечення й закриває потреби дрібних і більших систем відеоспостереження, у тому числі й географічно розподілених. У системи відсутні обмеження на кількість камер на сервер або систему в рамках придбаної ліцензії.

Ознайомитися із продуктами від Luxriot можна за допомогою запиту демо ліцензій або ж використовувати безкоштовну версію продукту. Безкоштовна версія має ряд обмежень, основні з яких: дев'ять підтримуваних каналів, максимальна розв'язна здатність камер до 2Мр, одне користувальницьке підключення на систему. Безкоштовна ліцензія може бути розширена до 16 каналів за потребою.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16



Рисунок 2.3 – Інтерфейс користувача Luxriot

Лінійка СВН Luxriot підтримує порядку 6000 IP пристроїв і підтримує невелику кількість інтеграцій сторонніх рішень по відео аналітиці й таким рішенням, як система контролю доступу. Що стосується бортової аналітики камер, Luxriot підтримує обмежений список виробників. Ліцензування в Luxriot поканальне й поширюється у вигляді бандлів на молодші версії й поканально в старшій версії. Тому що більшість IP камер на ринку зараз багатоканальні (профільні), поканальне ліцензування в деяких випадках накладає незручності й вимагає додаткового ліцензування при використанні багатоканальних пристроїв. Ліцензія в Luxriot прив'язується до апаратної частини сервера, тому при зміні конфігурації можливо буде потрібно повторна активація.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Незважаючи на невеликий вибір рішень по інтеграції рішень сторонніх виробників, функціонал СВН може бути розширений власними модулями аналітики. Luxriot розробляє наступні модулі відеоаналітики: захисту периметра, бізнес аналітики, розпізнавання номерних знаків і осіб.

Для рішення питань по розширенню функціонала й інтеграції надається API/SDK.

Користувальницькі підключення обмежені лише в безкоштовній версії. Підтримуються два типи облікових записів: пропрієтарні й Active Directory.

Для роботи операторів доступно три типи клієнтів: товстий клієнт, mobile і web клієнт. Клієнтські додатки абсолютно безкоштовні й додатково не ліцензуються. Використання камери пристрою мобільного клієнта дає можливість транслювати й записувати відео в систему.

Luxriot розроблений під Windows, його серверна частина, а основне ПЗ для оператора – під Windows і MacOS. Luxriot для зберігання конфігурації й системної інформації використовує пропрієтарну базу даних і підтримує роботу у віртуальному середовищі.

Network Optix. NxWitness / Hanwha. Wisenet Wave

(www.networkoptix.com / www.hanwhasecurity.com)

Дана VMS розглядається у двох назвах і від різних вендорів. Причина в OEM, компанія Network Optix брендує свою СВН NxWitness для Hanwha, що на ринку відома як Wisenet Wave. Network Optix не обмежується лише співробітництвом з Hanwha у цьому плані, як наслідок, можна знайти знайомий інтерфейс від інших вендорів.

Компанія Network Optix бере свій початок з 2010-2011 років і представлена на ринку основним програмним продуктом NxWitness.

NxWitness існує в одному варіанті, без розподілу на версії. Із всіх розглянутих, СВН має самий компактний інсталятор і позиціонується розроблювачами як дуже легковагий і невимогливий до апаратних ресурсів продукт, при цьому не на шкоду функціоналу.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

архіву на самому пристрої. Ліцензія прив'язується до апаратної частини, після зміни конфігурації сервера може знадобитися переактивація ліцензії. Ліцензія дає можливість використовувати в повному обсязі підтримуваний функціонал IP пристрою (порти I/O, звук, бортова аналітика), що стосується каналів – NxWitness підтримує тільки двоканальну схему роботи – основний і додатковий. В архітектурі NxWitness вторинний потік відіграє роль зниження навантаження на мережу або на АРМ оператора при перегляді відео. Другий потік за замовчуванням записується, але запис можна відключити.

Система має невеликий вибір уже готових рішень інтеграції відеоаналітики, але надає багатофункціональний API і SDK на безкоштовній основі. З використанням API повністю можна автоматизувати рутинну роботу із системою, аж до повного її налаштування або нестандартного моніторингу.

Типи облікових записів у системі дві: пропрієтарний і LDAP. Для роботи зі СВН існує три клієнтських ПЗ: товстий клієнт, mobile і web. Всі клієнти абсолютно безкоштовні. Програма налаштування системи сполучена із програмою роботи з відео, єдиний інтерфейс, поділ відбувається на рівні прав користувачів.

Серверна частина NxWitness підтримує роботу в середовищі Windows, Linux, клієнтська – Windows, Linux, MacOS. Відмінною рисою є підтримка ARM архітектури для північної частини програмного забезпечення. Вендором перевірена робота на одноплатних комп'ютерах Raspberry Pi і Banana Pi, хоча теоретично немає обмежень для використання на інших подібних пристроях.

NxWitness може бути розгорнутий у віртуальному середовищі, а також в Docker.

Сателіт Інновація. Macroscop (Eocortex)

(macroscop.com / eocortex.com)

Сателіт Інновація – компанія, що розробляє програмні рішення для професійних систем відеоспостереження за назвою Macroscop, для європейського ринку відома за назвою Eocortex. Macroscop бере свій початок з 2008 року й

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

розробляє програмне забезпечення, функції відеоаналітики й мережних реєстраторів.

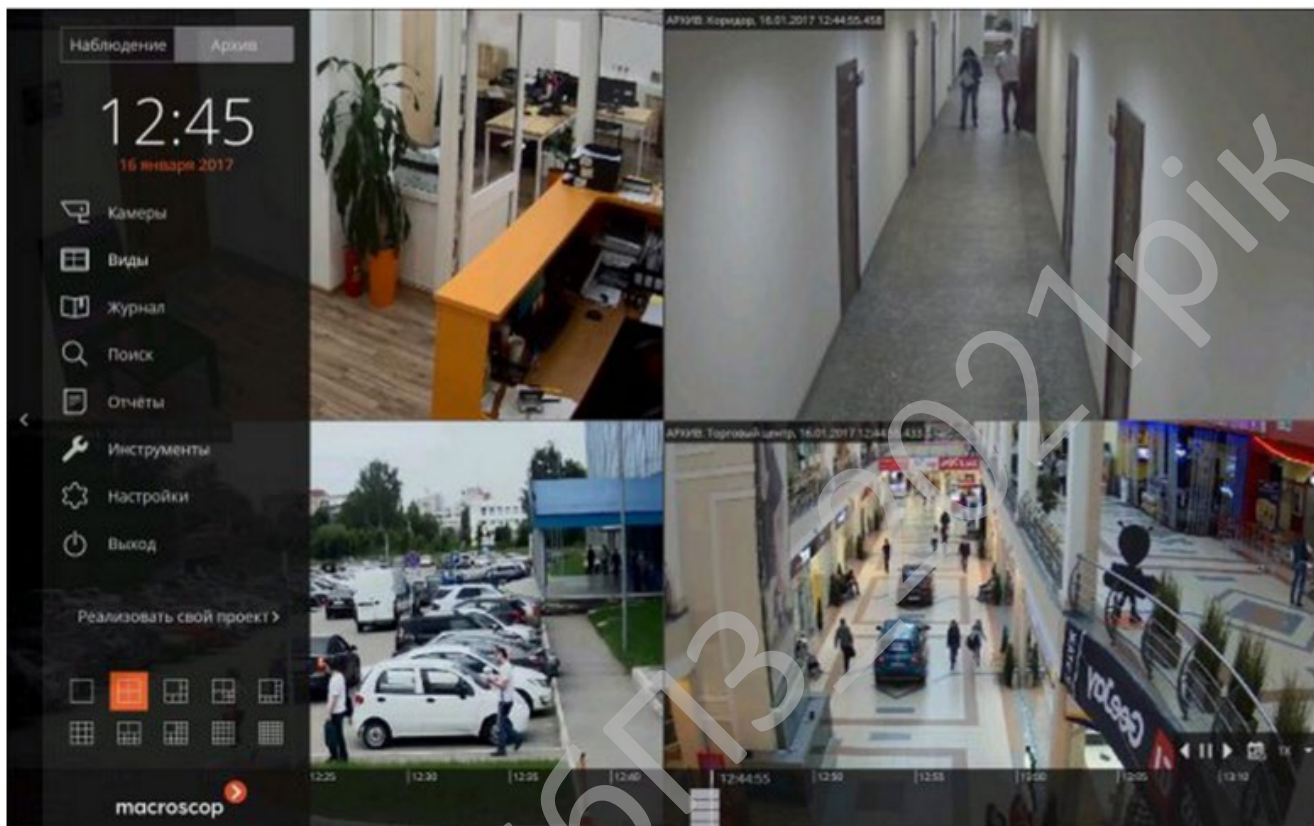


Рисунок 2.5 – Інтерфейс користувача Macroscop

Macroscop містить у собі три версії програмного забезпечення з різною кількістю підтримуваних камер, серверів, клієнтських підключень, підтримкою аналітики. Macroscop застосовна для дрібних об'єктів і великих об'єктів. Обмеження в старших версіях на кількість камер у системі відсутній, а от кількість камер на сервер обмежене до 479.

Для ознайомлення доступна демо версія продукту, безкоштовної версії в Macroscop немає.

Лінійка продуктів Macroscop підтримує близько 5000 IP камер і певні моделі реєстраторів. Ліцензування поканальне, більшість IP камер на ринку зараз багатоканальні (профільні), такий тип ліцензування в деяких випадках накладає

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

незручності й вимагає додаткового ліцензування при використанні багатоканальних пристроїв. Прив'язка ліцензії здійснюється до апаратної частини сервера, тому при зміні конфігурації можливо буде потрібно переактивація. Після активації ключа він не може бути перенесений на інший комп'ютер. Як варіант, є можливість використовувати USB ключа, що не має прив'язки.

Підтримка бортовий аналітики з камер в Macroscop не спостерігається. Для цих цілей Macroscop пропонує використовувати інтелектуальні модулі, убудовані в програмне забезпечення. Система нараховує порядку п'ятнадцяти модулів, вони ліцензуються окремо по кількості каналів, старша версія включає частину модулів на безкоштовній основі. Забавний факт, що деварпінг – розгорнення Fisheye-камери, ставиться до інтелектуального модуля й ліцензується додатково.

Macroscop розташовує власним SDK для потреб інтеграції й розширення функціонала системи. Готових сторонніх рішень по відеоаналітиці практично немає у вільному доступі, але існують інтеграції з деякими системами контролю доступу.

У системі підтримується один, пропрієтарний тип облікових записів. Підтримка користувачів Active Directory припинена. Для роботи зі СВН існує три клієнтських ПЗ: основне, mobile і web. Всі клієнти абсолютно безкоштовні. Існує окремий конфігуратор для налаштування системи.

Серверна й клієнтська частина підтримує роботу в середовищі Windows.

До вибору системи відеоспостереження потрібно підходити комплексно, ґрунтовно й з поглядом у майбутнє. Головний вибір може стояти між закритою й відкритою системою відеоспостереження, між системою з убудованою аналітикою або без, але з можливістю інтеграції; і завжди потрібно враховувати, є чи необхідність інтегрувати сторонні комплекси.

Що ж вибрати? Як по мені, то вибір очевидний – відкрита платформа, і бажано в повному обсязі цього слова.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Відкрита платформа надасть Вам гнучкість і право вибору – без прив'язки до IP устаткування, апаратним комплексам і найчастіше до аналітики. Відкритість платформи відеоспостереження дасть Вам поле для діяльності – тестуванню й вибору рішення, що дійсно буде вирішувати Ваші завдання.

Тому, вибираючи систему, важливо звертати увагу на її іноваційність і послужний список. Потрібно настроїти себе на одержання максимальної віддачі від продукту. І не варто дивитися тільки на вартість за канал, а скоріше більше розумним рішенням може бути оцінка співвідношення вартості каналу з його цінністю.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero RAD Studio Delphi 10.3.2 Rio Architect – це найшвидший спосіб створювати й обновляти інтенсивно працюючі з даними, сильно взаємодіючі застосунки з візуально насиченим користувальницьким інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою Object Pascal і C++. Широкий вибір функцій підтримки Windows 10, у тому числі нові компоненти VCL для Windows 10, стилі для VCL і FMX, а також служби UWP (універсальної платформи Windows), наприклад повідомлення, дозволяють легко й швидко перенести застосунки в Windows 10, зберігши користувачів. Нова платформа дозволяє підтримувати великі проекти на більшому числі платформ із подвоєним обсягом пам'яті в середовищі розробки й удвічі більшим розміром підтримуваних проектів. Крім того, підтримка декількох моніторів і десятки нових функцій середовища розробки, призначених для прискорення створення коду, зроблять роботу як ніколи ефективною. За допомогою RAD Studio 10 розроблювачі зможуть створювати застосунки в 5 разів швидше в порівнянні з іншими інструментами, а розробка застосунків для декількох настільних, мобільних, хмарних платформ і платформ баз даних, включаючи 32 і 64 -розрядні версії Windows 10, Mac OS X, iOS і Android, стане ще швидше.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

зміни подань, призначених для простої синхронізації даних застосунку по підписці без обмежень за розміром файлу бази даних.

– Поліпшена й оновлена підтримка для популярних баз даних, включаючи MySQL v8.0, MariaDB 10.3, SQL Server 2017, PostgreSQL v10, Firebird v3.0, MongoDB, InterBase, SQLite 3.23.1, SQL Anywhere і багатьох інших.

– Удосконалення DataSnap.

– Поліпшення REST. Підтримка додаткових родинних REST методів, типів і властивостей.

– Повністю оновлений модуль живлення версії Architect. Одержіть більше від версії Architect, включаючи ці ліцензії сімейства Idera.

– Ліцензія Sencha ExtJS Professional: Створіть свій ідеальний мережний вхідний інтерфейс за допомогою javascript і ExtJS.

– Ліцензія на розгортання InterBase ToGo. Додайте сховище даних у свої застосунки за допомогою цієї гнучкої, зашифрованої бази даних, що вбудовується.

– Ліцензія для розподіленого розгортання RAD Server. Ідеально підходить для серверного застосунку архітектури мікросервісів.

– Ліцензія AquaData Studio. Вражаючий аналіз бази даних.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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;

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

б) вибрати та обґрунтувати методика 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-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Опис AXIS P 1364-E

AXIS P 1364-E RU(0739-014) – вулична HDTV 720p камера IP-відеоспостереження, призначена для роботи при дуже слабкому освітленні, що підтримує фірмові технології Axis: Lightfinder, WDR – Forensic Capture, P-iris, Zipstream і оснащена системою віддаленого фокусування. Камера P 1364-E RU(0739-014) дозволяє одержати відеозображення з розв'язною здатністю HDTV 720p при частоті кадрів 50/60 кадр/с, забезпечуючи при цьому придушення шуму й зниження размиття об'єктів, що рухаються, при недостатнім освітленні. Технологія Axis Zipstream значно знижує обсяг переданого трафіку й вимоги до обсягу пам'яті для зберігання даних. Підтримка камерою широкого динамічного діапазону WDR – Forensic Capture дозволяє одержати відеозображення, що оптимально для розпізнавання об'єктів завдяки винятковому проробленню деталей – навіть у самих несприятливих умовах освітленості. Можливість віддаленого регулювання заднього фокуса підвищує ефективність установки камери й забезпечує чіткість зображення. Модель P 1364-E RU(0739-014) відповідає стандартам IP66, IP67, NEMA 4X і IK10. У комплект поставки камери входить кронштейн для кріплення на стіну й сонцезахистний козирок. Технологія Arctic Temperature Control гарантує запуск і роботу камери при температурах від -40 до 50°C.

Характеристики:

- Максимальна розв'язна здатність, пікс. – 1280x960.
- Виконання камери – Циліндрична.
- Об'єктив – Варіофокальний.
- Вулична – Так.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Поворотна – Немає.
- Функції – Підтримка аудіо/Живлення по PoE/Вулична.
- Живлення по PoE – Так.
- Підтримка звуку – Так.
- Кількість пікселів, Мпкс. – 1.2.
- Фокусна відстань, мм. – 2.8 – 8.5.
- Робоча температура, °C – – 40 – 50.
- Виробник – Axis.
- Посилання на виробника – <https://www.axis.com/ru-ru/products/axis-p1364-e>.

– Альтернативна структура – 106/126.

Особливості P 1364-E RU(0739-014):

- Технології Lightfinder і WDR – Forensic Capture.
- HDTV 720p із частотою 50/60 кадр/с.
- Технологія Axis Zipstream.
- Двостороння передача звуку.
- Технологія Arctic Temperature Control.

Технічні характеристики AXIS P 1364-E RU(0739-014):

- Зображення КМОП, 1/3”, прогресивне розгорнення, RGB.
- Об'єктив ІЧ-фільтр, об'єктив CS-mount, P-Iris, мегапіксельна розв'язна здатність Зі змінною фокусною відстанню 2,8–8,5 мм, F1,2 Горизонтальний кут огляду: 83°–33.3° Вертикальний кут огляду: 61°–24.7.

– Режим День/Ніч Автоматично керований інфрачервоний фільтр.

– Мінімальна освітленість Колір: 0,1 лк; ч/б: 0,01 лк, F1,2.

– HDTV 720p, 50/60 кадр/с: Колір: 0,4 лк; ч/б: 0,02 лк, F1,2.

– Швидкість спрацьовування затвора Від 1/28 000 з до 2 із при 50 Гц Від 1/33 500 з до 2 із при 60 Гц Відео.

– Стиск відео Профілі Baseline, Main і High кодека H.264 (MPEG-4, частина 10/AVC), Motion JPEG.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- Розв’язна здатність HDTV 720p, 25/30 кадр/с (з WDR): від 1280 x 960 до 160 x 90.
- HDTV 720p, 50/60 кадр/с (без WDR): від 1280 x 960 до 160 x 90.
- Частота кадрів До 50/60 кадр/с (50/60 Гц).
- Передача відеопотоку Передача декількох потоків, що налаштовуються окремо, у форматах H.264 і Motion JPEG.
 - Технологія Axis Zipstream при використанні H.264.
 - Контрольована частота кадрів і трафік VBR/MBR H.264.
 - Передача декількох відеопотоків.
 - До 8 окремих фрагментів загального зображення.
 - Налаштування зображення. Регулювання стиску, кольору, яскравості, чіткості, контрасту, балансу білого, експозиції й ділянок експонування; широкий динамічний діапазон WDR-Forensic Capture: до 120 дБ залежно від об’єкта зйомки; тонке налаштування дій при слабкому освітленні; поворот: 0°, 90°, 180° і 270°а , накладення тексту на зображення, зони маскуванню, дзеркальне відбиття зображень Аудіо.
 - Передача аудіо потоку. Двостороння повнодуплексна.
 - Стиск аудіо. AAC LC 8/16/32/48 кГц, G.711 PCM 48 кГц, G.726 ADPCM 48 кГц; Що налаштовується бітрейт.
 - Ввод/вивід аудіо. Вхід для зовнішнього мікрофона або лінійний вхід, лінійний вихід Мережа.
 - Безпека. Захист паролем, фільтрація IP-адрес, шифрування HTTPSb ,контроль мережного доступу IEEE 802.1Xb , дайджест-перевірка дійсності, журнал доступу користувачів.
 - Підтримувані протоколи: IPv4/v6, HTTP, HTTPSb , SSL/TLSb , Qo Layer 3 DiffServ, FTP, CIFS/SMB, SMTP, Bonjour, UPnP™, SNMPv1/v2c/v3 (MIB-II), DNS, DynDNS, NTP, RTSP, RTP, SFTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS, SSH Системна інтеграція.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- Програмний інтерфейс. Відкритий API для інтеграції ПЗ, включаючи VAPIX® і платформу застосунків камер AXIS; Система відеохостингу AXIS Video (AVHS) з підключенням камери одним клацанням; Профіль ONVIF S;
- Аналітика. Відеодетектор руху, активне оповіщення при несанкціонованих діях, детектор звуку.
- Підтримка платформи застосунків камер AXIS, що забезпечує установку таких застосунків як відеодетектор руху AXIS Video Motion Detection 3, детектор перетинання заданої лінії AXIS Cross Line Detection, автоматичне цифрове спостереження AXIS Digital Autotracking.
- Спрацьовування сигналу тривоги. Аналітика, події локального запису даних, зовнішній вхід, рівень звуку, розклад Дії по подіях.
- Завантаження файлів: по FTP, SFTP, HTTP, HTTPS і електронній пошті, через загальні мережні папки.
- Розсилання повідомлень: по електронній пошті, HTTP, HTTPS і TCP.
- Вихідний сигнал на зовнішнє встаткування.
- Відеозапис на зовнішній накопичувач, відтворення аудіокліпів.
- Буферизація відео до й після тривоги.
- Передвстановка PTZ, маршрут обходу охорони, накладення тексту.
- Перемикання режимів «день-ніч», активація індикаторів стану.
- Режим WDR.
- Відправлення SNMP-Пастки.
- Поточкова передача дані Дані про події.
- Убудовані засоби установки Помічник фокусування, лічильник пікселів, віддалена регулювання заднього фокуса Загальні характеристики.
- Матеріал корпусу Полімерний корпус, клас захисту IP66, IP67 і NEMA 4X, клас удароміцності IK10.
- Колір: білий NCS S 1002-B.
- Пам'ять ОЗП: 512 МБ, флеш-пам'ять: 256 МБ.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- Живлення. Технологія Power over Ethernet (PoE) IEEE 802.3af/802.3at, тип 1, клас 3; макс. 12,95 Вт, звичайно 5,0 Вт.
- Рознімання RJ45 10 BASE-T/100 BASE-TX PoE.
- Ввод-вивід: 4-контактна (2,5 мм) клемна колодка вводу/виводу для 2 входів, що налаштовуються-виходів.
- RS-485/422, 2 шт., 2 положення, повнодуплексний режим, клемна колодка.
- Мікрофонний/лінійний вхід 3,5 мм; лінійний вихід 3,5 мм.
- Рознімання керування діафрагмою P-iris (сполучимо з DC-iris).
- Локальне зберігання даних. Підтримка карт пам'яті microSD/microSDHC/microSDXC.
- Підтримка відеозапису по мережі на виділений мережний накопичувач (NAS).
- Умови експлуатації Від -40 до 50°C.
- Відносна вологість: 10-100% (з утворенням конденсату).
- Відповідність стандартам EN 55022, клас А; EN 61000-3-2; EN 61000-3-3; EN 55024; EN 61000-6-1; EN 61000-6-2; FCC, частина 15, розділ В, клас А; ICES-003, клас А; VCCI, клас А; С -tick AS/NZS CISPR 22, клас А; КСС KN32, клас А, KN35; IEC/EN/UL60950; IEC/EN/UL60950EN50121-4/IEC 62236; IEC/EN/UL 60950-22; IEC/EN 60529 IP66; NEMA 250, тип 4X; IEC 60068-2-6; IEC 60068-2-27; IEC/EN 62262 IK10.
- Розміри 382 x 155 x 120 мм.
- Маса 1,8 кг.
- Приналежності в комплекті поставки Комплект рознімань, посібник з установки, ліцензія на декодер для Windows на 1 користувача, настінний кронштейн, сонцезахистний козирок. Датчик, що подає сигнал тривоги при вторгненні.
- Додаткові аксесуари Кріплення Axis, об'єктиви Axis, інжектори Axis Кронштейн для коридорного режиму AXIS Bracket A.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- ПЗ для керування відео. Застосунки AXIS Camera Companion, AXIS Camera Station,.
- ПЗ для керування відео, що поставляються партнерами Axis по розробці застосунків.
- Мови. Українська, англійська, німецька, французька, іспанська, італійська, китайська (спрощена), японська, корейська, португальська.
- Гарантія. Відомості про 3 -літню гарантію.

Мережна камера AXIS P 1365-E Mk II

- Надійна конструкція із захистом від ударів.
- Працює при температурах від -40°C до 50°C.
- Технологія Lightfinder дозволяє одержувати кольорові зображення навіть при дуже слабкому освітленні..
- Розв'язна здатність HDTV 1080p при частоті 50/60 кадр/с.
- Розроблена компанією Axis технологія Zipstream.

Міцна мережна камера AXIS P 1365-E Mk II, призначена для роботи в суворих умовах на вулиці, забезпечує відмінну якість зображення з високою деталізацією навіть при слабкому освітленні. AXIS P 1365-E Mk II прекрасно підходить для установки в банках, урядових закладах, на паркуваннях і будь -яких інших об'єктах, де необхідні зображення з високою деталізацією або огляд більших площ на відкритому повітрі.

Особливості Axis P 1365-E Mk II

Міцна модель, готова до роботи

Камера AXIS P 1365-E Mk II призначена для зовнішньої установки й експлуатації в екстремальних умовах – вона може працювати навіть при температурах -40°C і 50°C. Функція Arctic Temperature Control гарантує безпечний запуск камери при температурах до -40°C навіть після аварійного відключення живлення. Ви можете бути впевнені, що кожух камери зможе витримати як суворі погодні умови, так і дії вандалів, оскільки він має захист, що відповідає стандартам IP66, IP67, NEMA 4X і IK10. Модель AXIS P 1365-E Mk II

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

поставляється в комплекті з настінним кронштейном і сонцезахисним козирком для захисту від пилу, дощу, снігу й сонячного світла.

AXIS P 1365-E Mk II дозволяє одержати якісне зображення, придатне для розпізнавання об'єктів – навіть при зйомці в темряві. Модель демонструє найвищу світлочутливість, що в сполученні з технологією Axis Lightfinder дозволяє одержувати високоякісні, причому кольорові, зображення навіть при слабкій освітленості й у темряві. А за допомогою технології WDR – Forensic Capture, що знижує рівень шуму й підсилює сигнал, ви будете мати зображення з високою деталізацією, незважаючи на нічний час зйомки. Відзначимо, що AXIS P 1365-E Mk II має розв'язна здатність HDTV 1080p і підтримує частоту до 50 або 60 кадр/с у форматах H.264 і Motion JPEG, що дозволяє вести зйомку швидко рухомих об'єктів і людей з високим розв'язною здатністю.

Висока якість. Низьке навантаження на мережу

У моделі AXIS P 1365-E Mk II застосована технологія Axis Zipstream, що дозволяє аналізувати відеопотік і визначати важливі ділянки зображення в режимі реального часу. Знайдені важливі ділянки стискаються в меншому ступені, чим інша частина зображення, щоб зберегти деталі, що представляють інтерес, з максимальним розв'язною здатністю. Такий підхід скорочує трафік і обсяг пам'яті для зберігання даних на величину до 50%, у порівнянні з вихідним відеозаписом. У результаті вдається одержати зображення високої якості при значній економії ресурсів.

Таблиця 3.1 – Технічні характеристики IP-камери Axis P 1365-E Mk II

Камера	
Матриця	КМОП, 1/2.8", прогресивне розгорнення, RGB
Об'єктив	ІЧ-фільтр, об'єктив CS-mount, діафрагма P-iris Зі змінною фокусною відстанню 2,8 – 8 мм, F1,3 Горизонтальний кут огляду: 84° – 39° Вертикальний кут огляду: 46° – 21°
День/Ніч	Автоматично керований інфрачервоний фільтр

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

ВКРМ-123.21.0018.00.00.ПЗ

Арк.

35

Продовження таблиці 3.1

Мінімальна освітленість	Колір – 0.11 лк; ч/б – 0.01 лк, F1.3 HDTV 1080р, 50/60 кадр/с: колір – 0.22 лк; ч/б – 0.02 лк; F1.3
Швидкість спрацьовування затвора	Від 1/66500 до 2 з
Панорамування, нахил і масштабування	Цифрове PTZ-керування, що завантажується драйвер PTZ (передвстановлений Pelco D)
Відео	
Відеокодек	Профілі Baseline, Main і High формату H.264 (MPEG-4, частина 10/AVC) Motion JPEG
Розв'язна здатність	HDTV 1080р, 25/30 кадр/с (з WDR): від 1920 x 1080 до 160 x 90 HDTV 1080р, 50/60 кадр/с (без WDR): від 1920 x 1080 до 160 x 90
Потокове відео	Передача декількох потоків, що налаштовуються окремо, у форматах H.264 і Motion JPEG технологія Axis Zipstream при використанні H.264 Контрольована частота кадрів і смуга пропускання, VBR/CBR H.264
Передача декількох відеопотоків	До 8 окремих фрагментів загального зображення

Продовження таблиці 3.1

Налаштування зображення	Стиск, колір, яскравість, чіткість, контраст, баланс білого, установка експозиції, області експозиції, компенсація зустрічної завітлення, широкий динамічний діапазон з функцією Forensic Capture до 120 дБ залежно від об'єкта зйомки, тонке налаштування дій при слабкому освітленні, поворот: 0°, 90°, 180°, 270°, включаючи включаючи коридорний режим, накладення тексту й зображень, маска закритих зон; дзеркальне відбиття зображень.
Аудіо	
Передача аудіопотоку	Двостороння повнодуплексна
Стиск аудіо	ААС-LC 8/16 кГц, G.711 PCM 8 кГц, G.726 ADPCM 8 кГц Що налаштовується бітрейт
Мережа	
Безпека	Захист паролем, фільтрація IP-адрес, шифрування HTTPS, контроль доступу по мережі IEEE 802.1Xc, дайджест-перевірка дійсності, журнал доступу користувачів
Підтримувані протоколи	IPv4/v6, HTTP, HTTPS, SSL/TLS, Qo Layer 3 DiffServ, FTP, SFTP, CIFS/SMB, SMTP, Bonjour, UPnP, SNMP v1/v2c/v3 (MIB-II), DNS, DynDNS, NTP, RTSP, RTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS, SSH

Продовження таблиці 3.1

Системна інтеграція	
Прикладний програмний інтерфейс	Відкритий інтерфейс API для інтеграції програмного забезпечення, включаючи VAPIX® і AXIS Camera Application Platform компанії Axis Communications; ONVIF, Підтримка AXIS Video Hosting System (AVHS) зі швидким підключенням камери
Засоби аналітики	Відеодетектор руху, активне оповіщення при несанкціонованих діях, виявлення звуків, підтримка відкритої прикладної платформи камер AXIS, що забезпечує установку застосунків AXIS Video Motion Detection 3, AXIS Cross Line Detection, AXIS Digital Autotracking, а також застосунків сторонніх розроблювачів, див. www.axis.com/acap
Спрацьовування сигналу тривоги	Засоби аналітики, зовнішній вхід, рівень звуку, розклад
Дії по подіях	Завантаження файлів: по FTP, SFTP, HTTP, HTTPS і електронній пошті, через загальні мережні папки Розсилання повідомлень: по електронній пошті, HTTP, HTTPS і TCP Вихідний сигнал на зовнішнє встаткування Відеозапис на зовнішній накопичувач, відтворення аудіокліпів Буферизація відео до й після тривоги Передвстановка PTZ, маршрут обходу охорони, накладення тексту Перемикання режимів «день – ніч», активація індикаторів стану Режим WDR Відправлення SNMP-пастки

Продовження таблиці 3.1

Потокова передача даних	Дані подій
Убудовані засоби установки	Помічник фокусування, лічильник пікселів, віддалене регулювання заднього фокуса
Загальні характеристики	
Корпус	Полімерний корпус, клас захисту IP66, IP67 і NEMA 4X, клас удароміцності IK10 Колір: білий NCS S 1002-B
Пам'ять	ОЗП: 512 МБ, флеш-пам'ять: 256 МБ
Живлення	технологія Power over Ethernet (PoE) IEEE 802.3af/802.3at, тип 1, клас 3, макс. 12,95 Вт, звичайно 4,3 Вт
Рознімання	RJ45 10 BASE-T/100 BASE-TX PoE Клемна колодка для 2 входів, що налаштовуються, /виходів для сигналів тривоги (на виході: 12 V пост. струму, макс. навантаження 50 mA) RS-485/422, 2 шт., 2 положення, повнодуплексний режим, клемна колодка Мікрофонний/лінійний вхід 3,5 мм; лінійний вихід 3,5 мм Рознімання керування діафрагмою P-Iris (сполучимо з DC-iris)
Локальне зберігання даних	Підтримка карт пам'яті microSD/microSDHC/microSDXC Підтримка відеозапису на виділений мережний накопичувач (NAS) Рекомендації з вибору карт SD і мережного накопичувача NAS можна знайти на сайті www.axis.com

Продовження таблиці 3.1

Умови роботи	Від -40°C до 50°C Відносна вологість 10-100% (з конденсатом)
Відповідність стандартам	EN 55022 клас B; EN 55024; EN 61000-6-1; EN 61000-6-2; FCC частина 15 розділ B клас B з екранованим кабелем (STP); FCC частина 15 розділ B клас A з неекранованим кабелем (UTP); ICES-003 клас B; VCCI клас B; RCM AS/NZS CISPR 22 клас B; КСС KN32 клас B; KN35; IEC/EN/UL 60950-1; IEC/EN/UL 60950-22 EN 50121-4/IEC 62236, IEC/EN 60529 IP66, IEC/EN 60529 IP67, NEMA 250, тип 4X, IEC 60068-2-6, IEC 60068-2-27, IEC/EN 62262 IK10
Вага	1,8 кг
Розміри	382 x 155 x 120 мм
Приналежності в комплекті	Комплект рознімань, посібник з установки, однокористувальницька ліцензія на декодер для Windows, компакт-диск із засобами для установки й керування, настінний кронштейн, сонцезахистний козирок
Додаткові аксесуари	Кабельна коробка AXIS T94R01P Кріплення для монтажу на стовпі AXIS T91A47 Кронштейн для коридорного режиму AXIS Bracket A Дверний датчик AXIS Door Switch A Монтажна шафа AXIS T98A 16-VE
ПЗ для керування відео	Застосунки AXIS Camera Companion, AXIS Camera Station, ПЗ для керування відео, що поставляються партнерами Axis по розробці застосунків, доступні на сторінці www.axis.com/techsup/software
Гарантія	Відомості про 3 -літньої гарантії Axis і варіанті розширеної гарантії

3.2 Розробка структурної схеми

Сучасне життя неможливо представити без систем відеоспостереження. Вони встановлюються всюди – в аеропортах і на вокзалах, у банках і супермаркетах, у заміських будинках і приватних квартирах. Легко можна переконатися, що установка відеоспостереження стала однією з головних прикмет сучасності.

Пристрої, що входять до складу систем відеоспостереження, постійно вдосконалюються. Завдяки цьому щорічно з'являються більше точні й більше надійні системи відеоспостереження.

У цей час найбільшим попитом користуються два типи систем відеоспостереження. У першому типі установка відеоспостереження використовує персональний комп'ютер у якому є необхідне програмне забезпечення й плата відеозахвата. Другий тип відеосистем використовує цифровий реєстратор, що скорочено називають DVR (Digital Video Recorder).

Сучасні системи відеоспостереження можуть не тільки спостерігати за подіями або записувати їх, але й транслювати відеозапис по Інтернету або передавати її на мобільний телефон.

По своєму центральному влаштуванню відеосистеми підрозділяються на кілька типів:

Перший тип – найпростіша й найдешевша система – аналогова система відеоспостереження, основним пристроєм якої є квадратор. Цей пристрій виводить на монітор зображення, що надходять від декількох різних відеокамер. Різні режими роботи квадратора дозволяють виводити на монітор зображення від однієї камери, або від всіх одночасно, або виводити зображення по черзі.

Однак система на базі квадратора може записувати події тільки тоді, коли вона оснащена додатковим пристроєм – відеомагнітофоном. Тому установка відеоспостереження, що використовує квадратор, доцільна тільки на тих об'єктах, які цілодобово охороняються. Це можуть бути, наприклад, паркування, прохідні або автостоянки.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Другий тип – система відеоспостереження на базі відеореєстратора. Вона має більше широке коло функцій, тому що реєстратор приймає, аналізує, обробляє й записує всі вступників на нього дані. До одного відеореєстратора модно підключити від 16-ти до 64-х відеокамер. Така відеосистема легко передає зображення по Інтернету, на смартфон, на персональний комп'ютер. Набір функцій відеореєстратора набагато більше, ніж квадратора.

Останнім часом все частіше встановлюється відеоспостереження на базі персонального комп'ютера. Такий тип відеосистем набагато більше зручний і функціональний по порівнянням з аналоговим типом. Для включення у відеосистему в комп'ютер вставляється додатковий пристрій – плата відеозахвату. Ця плата, за допомогою необхідного програмного забезпечення трансформує аналоговий сигнал у цифровий формат.

Перетворений сигнал може стискуватися, записуватися, архівуватися, виводитися на екран. До входу -виходу плати відеозахвата можна підключити відразу кілька камер відеоспостереження. Останні розробки відеоплат оснащені важливою функцією стиску відеоінформації. Примітно те, що відеоспостереження в нічних умовах поводитья також як і в денний час доби.

IP система відеоспостереження використовує, у якості складових, IP-камери й IP-відеореєстратори. Для передачі відеопотоків використовується стек протоколів TCP/IP. У даний момент це саме динамічно, що розвивається напрямом, ринку, чому сприяє неухильно, що знижується вартість, що становить системи.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

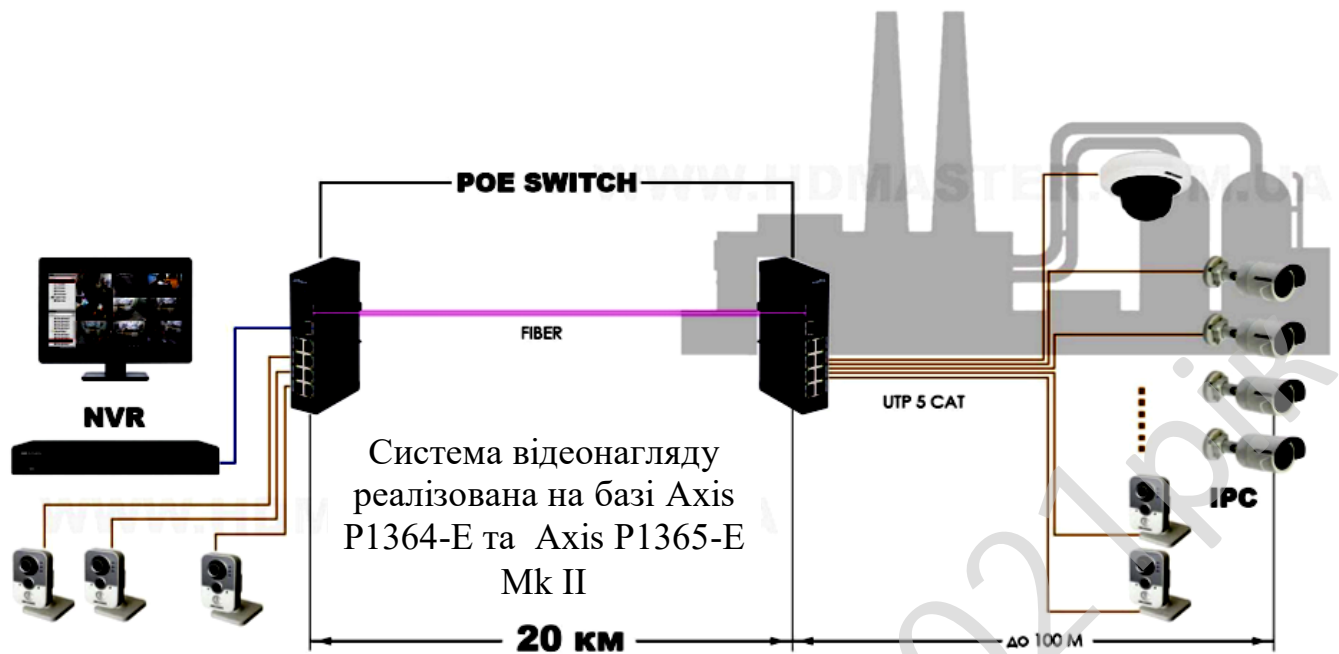


Рисунок 3.1 – Структурна схема системи

Переваги:

- Висока розв'язна здатність, обмежена лише пропускною здатністю мережі й параметрами використовуваного устаткування.
- Відстань передачі відеопотоку від IP-камери відеоспостереження не обмежено межами локальної мережі. У такий спосіб досягається масштабованість системи. IP-відеореєстратор здатний записувати відеопотоки від IP-камер що перебувають, наприклад, в іншому місті.
- Можливість використання для передачі відеопотоків існуючі комп'ютерні мережі.
- Можливість тонкого налаштування параметрів IP-відеокамер через web-інтерфейс. Налаштувати можна як параметри зображення, такі як: яскравість, контрастність, колірний баланс, так і вибрати кодек для роботи з відеозображенням, виставити зони зображення по яких камера видасть сигнал об виявлення руху, указати e-mail або параметри ftp сервера, на який будуть відправлені кадри у випадку виявлення руху в кадрі й багато чого іншого.

– TCP (Transmission Control Protocol) – протокол керуючий передачею даних, що перевіряє установку мережного з'єднання, надсилає новий запит у випадку втрати пакетів і не допускає дублювання пакетів. Таким чином, здійснюється надійна передача даних з повідомленням сторони, що відправляє, про якість передачі.

– UDP (User Datagram Protocol) – протокол передачі так званих «датаграм» – блоків даних, без перевірки успішності з'єднання, втрати пакетів і дублювання, що значно знижує якість передачі даних. Однак такий підхід буває досить корисний при коротких запитах від великої кількості клієнтів до сервера, як, наприклад, в онлайн-іграх, що звільняє сервер від очікування перевірки цілісності пакетів.

– DNS (Domain Name System) – система доменних імен, відповідальна за відповідність IP-адрес іменам хостов. Звичайно використовується для визначення IP-адреси по ім'ю хоста (по ім'ю сайту).

– DHCP (Dynamic Host Configuration Protocol) – протокол необхідний для автоматичного одержання комп'ютером IP-адреси й інших параметрів необхідних для нормальної роботи в мережі.

– SMTP (Simple Mail Transfer Protocol) – протокол для передачі пошти в Інтернеті, розроблений в 1982 році, застосовується, в основному, для відправлення вихідної пошти із клієнтської програми, наприклад "Outlook", на поштовий сервер.

– RTP (Real-time Transport Protocol) – протокол для передачі даних у реальному часі, з контролем послідовності пакетів і синхронізації даних. Даний протокол добре підходить для передачі відео й аудіоданих по мережі.

– DynDNS – сервіс який дозволяє користувачеві з динамічним IP-адресою, одержати піддомен (доменне ім'я третього рівня), зі статичною адресою, на який, сервіс DynDNS перенаправляє запит користувача.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- Bonjour – служба автоматичного виявлення мережних пристроїв у ближнім мережному оточенні, що використовує дані з DNS.
- UPnP (Universal Plug and Play) – технологія, що забезпечує автоматичне підключення й налаштування мережних пристроїв, відразу після приєднання до мережі. Дана технологія значно облягає використання мережних пристроїв звичайним користувачам.

3.3 Розробка функціональної схеми

Опишемо функції розробленої системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Функціональна схема системи складається з наступних блоків:

- Блок функцій адміністратора.
- Блок функцій оператора.
- Блок функцій інженера.

Блок функцій адміністратора

1. Додавання нових користувачів.

Система дозволяє не тільки контролювати обстановку по відеоряду, але й відслідковувати всі зміни на двомірному плані охоронюваного простору. На плані відображаються області видимості камер і виявлені об'єкти, що полегшує сприйняття загальної обстановки.

Так, наприклад, на плані легко помітити області контрольованого простору не видимі ні однією відеокамерою.

Крім того система надає користувальницький інтерфейс для керування камерами за допомогою кліків миші на компоненті головного вікна, із зображенням плану.

Система відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II має розвитий графічний інтерфейс і істотну обчислювальну частину.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Система має більш багату функціональність ніж існуючі комерційні розробки. Використання поворотних камер, розмаїтість можливостей по керуванню камерами й спостереженню за обстановкою дозволяє ефективніше вирішувати завдання відеонагляду, ніж при використанні традиційних систем відеонагляду з нерухливими камерами.

Таких можливостей удалося досягти завдяки використанню розроблених методів калібрування камер і виявлення областей інтересу.

Блок функцій оператора

1. Перегляд відео з камер.
2. Оцінка ситуації по плану:
 - Відображення областей спостереження камер.
 - Відображення доданих об'єктів.
 - Відображення знайдених об'єктів.
3. Керування камерами:
 - Клік по відео.
 - Обертання камери.
 - Клік по карті.
 - Режим супроводу.
4. Зміна налаштувань системи.

Блок функцій інженера

1. Завантаження плану контролюемого простору.
2. Додавання заборонених областей на план.
3. Додавання об'єктів на план.
4. Калібрування камер.
5. Додавання та видалення камер.
6. Калібрування плану.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

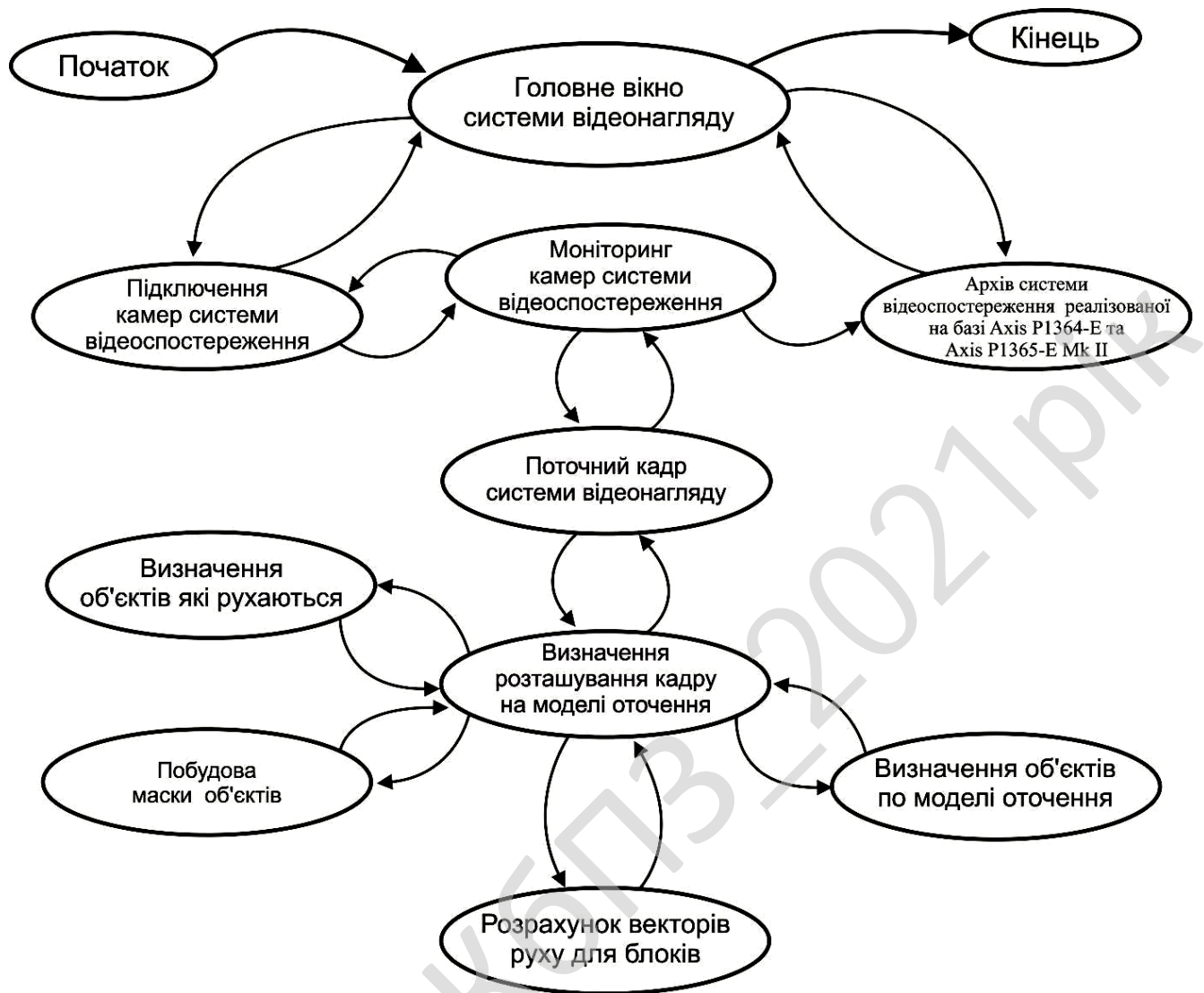


Рисунок 3.3 – Діаграма взаємодії процесів

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

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

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.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію магістерської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

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

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

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

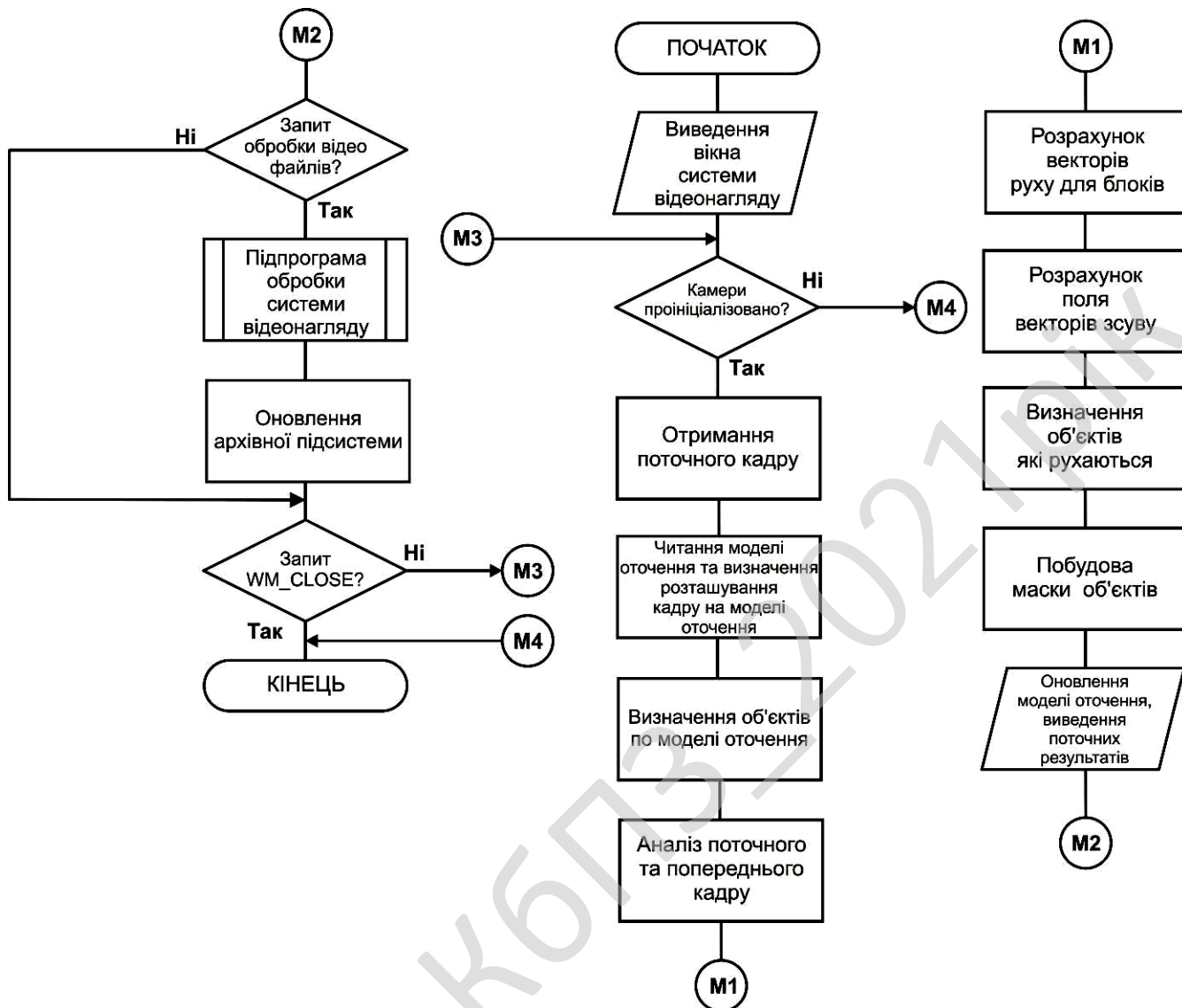


Рисунок 4.1 – Блок-схема основної програми

Гнучка розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні.

Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування.

Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий програмний проект готовий до випуску наприкінці кожної ітерації. Після закінчення кожної ітерації, команда виконує переоцінку пріоритетів розробки.

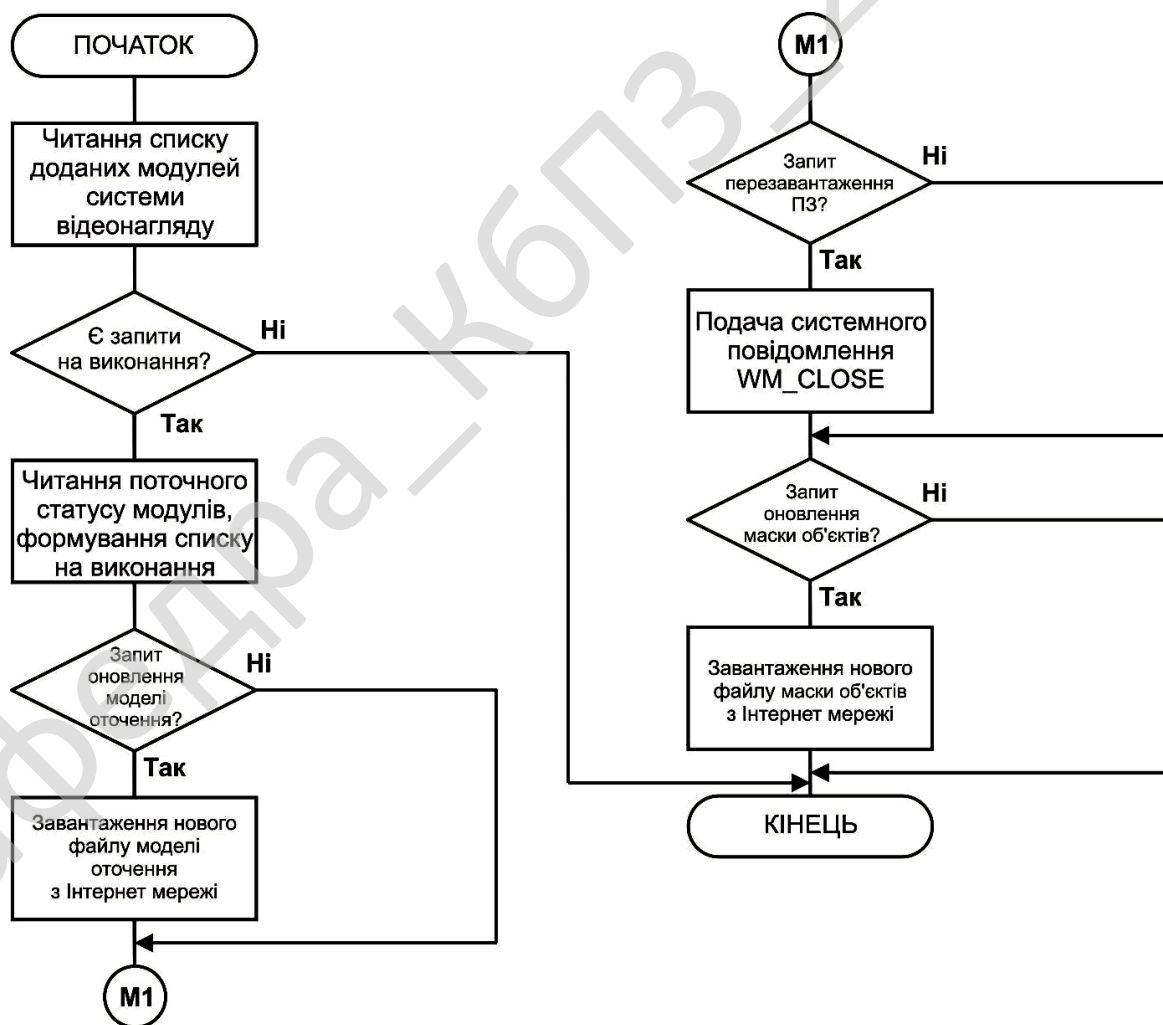


Рисунок 4.2 – Блок-схема роботи підпрограми

- часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);
- тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
- робоче програмне забезпечення – найкращий вимірювач прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійну увагу поліпшенню технічної майстерності та зручному дизайну;
- простота – мистецтво не робити зайвої роботи;
- найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;
- постійна адаптація до мінливих обставин.

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

Критика. Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре»), при цьому не враховується, що може перестати працювати при найменшій зміні або ж породити важкі до відтворення дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопиченню дефектів.

Методології. Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) Системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні». Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту. Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротака Такеучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос–функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеГрейс та Шталь у книжці Злі проблеми, справедливі рішення посилалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990–х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників.

Протягом кожного спринту, 15–30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

Product backlog – це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називається «історіями» (user story) або елементами backlog–у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum–процесу.

Обов'язкові поля:

1. ID – унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

2. Назва (Name) – стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти, про що йдеться і відрізнити одну історію від іншої.

3. Важливість (Importance) – ступінь важливості даної історії на погляд product owner 'а. Зазвичай являє собою натуральне число, іноді для цієї цілі використовуються числа Фібоначчі. Чим більше значення, тим більше пріоритет.

4. Попередня оцінка (initial estimate) – початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point'ах. Приблизно відповідає числу «ідеальних людино-днів».

5. Як продемонструвати (how to demo) – стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

Додаткові поля. Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner'у визначитися з його пріоритетами.

Категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет.

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Протягом наради кожен член команди відповідає на 3 запитання:

- Що зроблено з моменту попередньої щоденної наради?
- Що буде зроблено з моменту поточної наради до наступної?
- Які проблеми заважають досягненню цілей спринта? (Над рішенням цих

проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.)

Демонстрація (Sprint Review Meeting):

- Проходить у кінці ітерації (спринта).
- Команда демонструє внесок функціональності до продукту всім зацікавленим особам.

- Залучається максимальна кількість глядачів.

- Усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт).

- Обмежена 4-ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective):

- Члени команди висловлюють свою думку про минулий спринт.

- Відповідають на два основних запитання: Що було зроблено добре у минулому спринті? Що потрібно покращити в наступному?.

- Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).

- Обмежена 1–3ма годинами.

NetBIOS (Network Basic Input/Output System) – протокол для роботи в локальних мережах на персональних ЕОМ типу IBM/PC, розроблений у вигляді інтерфейсу, який не залежить від фірми-виробника. Був розроблений фірмою Sytek Corporation за замовленням IBM в 1983 році. Він включає в себе інтерфейс сеансового рівня (англ. NetBIOS interface), в якості транспортних протоколів використовує TCP і UDP.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

сесія починається з NETBIOS-запиту, завдання IP-адреси і визначення TCP-порту віддаленого об'єкта, далі йде обмін NETBIOS-повідомленнями, після чого сесія закривається.

Сесія здійснює обмін інформацією між двома NETBIOS – додатками. Довжина повідомлення лежить в межах від 0 до 131 071 байт. Припустимо одночасне встановлення декількох сесій між двома об'єктами.

При організації IP-транспорту через NETBIOS IP-дейтаграмма вкладається в NETBIOS-пакет. Інформаційний обмін відбувається в цьому випадку без встановлення зв'язку між об'єктами. Імена NETBIOS повинні містити в собі IP-адреси. Так, частина NETBIOS-адреси може мати вигляд IP.XX.XX. XX.XX, де IP вказує на тип операції (IP через Netbios), а XX. XX. XX.XX – IP- адреса.

Система NETBIOS має власну систему команд (call, listen, hang up, send, receive, session status, reset, cancel, adapter status, unlink, remote program load) і примітивів для роботи з дейтаграммами (send datagram, send broadcast datagram, receive datagram, receive broadcast datagram).

Всі кінцеві вузли NETBIOS діляться на три типи:

- широкомовні («b») вузли;
- вузли точка-точка («p»);
- вузли змішаного типу («m»).

IP-адреса може асоціюватися з одним із зазначених типів. В-вузли встановлюють зв'язок зі своїм партнером за допомогою широкомовних запитів. Р- і М-вузли для цієї мети використовують netbios сервер імен (NBNS) і сервер розподілу дейтаграм (NBDD).

NetBIOS забезпечує:

- реєстрацію і перевірку мережевих імен;
- встановлення і розрив з'єднань;
- зв'язок з підтвердженням доставки інформації;
- зв'язок без підтвердження доставки інформації;
- підтримку управління і моніторингу драйвера і мережевої карти.

SQLite – полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92.

Сирцевий код SQLite поширюється як суспільне надбання (public domain), тобто може використовуватися без обмежень та безоплатно з будь-якою метою. Фінансову підтримку розробників SQLite здійснює спеціально створений консорціум, до якого входять такі компанії, як Adobe, Oracle, Mozilla, Nokia, Bentley і Bloomberg.

З 2018р SQLite, як й JSON та CSV, рекомендований Бібліотекою Конгресу США формат зберігання структурованого набору даних. У 2005 році проект отримав нагороду Google-O'Reilly Open Source Awards.

Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми.

Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму.

SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок.

Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів або потоків можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі рушія можливо використовувати SQLite як на вбудовуваних (embedded) системах, так і на виділених машинах з гігабайтними масивами даних.

Особливості SQLite:

1. Транзакції атомарні, послідовні, ізольовані, і міцні (ACID) навіть після збоїв системи і збоїв живлення.
2. Встановлення без конфігурації – не потребує ані установки, ані адміністрування.
3. Реалізує значну частину стандарту SQL92.
4. База даних зберігається в одному крос-платформовому файлі на диску.
5. Підтримка терабайтних розмірів баз даних і гігабайтного розміру рядків і BLOBів.
6. Малий розмір коду: менше ніж 350KB повністю налаштований, і менш 200KB з опущеними додатковими функціями.
7. Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій.
8. Простий, легкий у використанні API.
9. Написана в ANSI C, включена прив'язка до TCL; доступні також прив'язки для десятків інших мов.
10. Добре прокоментований сирцевий код зі 100% тестовий покриттям гілок.
11. Доступний як єдиний файл сирцевого коду на ANSI C, який можна легко вставити в інший проект.
12. Автономність: немає зовнішніх залежностей.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

13. Крос-платформовість: з коробки підтримується Unix (Linux і Mac OS X), OS/2, Windows (Win32 і WinCE). Легко переноситься на інші системи.

14. Сирці перебувають в суспільному надбанні.

15. Поставляється з автономним клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних SQLite.

Інструменти створення та обслуговування БД

Створення та обслуговування БД можуть здійснюватись через текстову консоль SQL-командами або через спеціальні інструменти, у тому числі – з графічним інтерфейсом користувача.

Технології, що підтримують SQLite та мови програмування

Сама бібліотека SQLite написана мовою C. Проте є реалізація бібліотеки на JavaScript sql.js, яка дозволяє обробляти файли БД безпосередньо в браузері.

Для інших мов програмування розроблено механізм підключення й роботи з БД через цю бібліотеку: C++, Java, Python, Perl, PHP, Ruby, Haskell, Scheme, Smalltalk, Lua тощо. Засоби для роботи з Tcl включені в комплект постачання SQLite. Повний список наявних засобів можна знайти на сторінці проекту.

Web-інструментарії

У ряді інструментаріїв присутня можливість використання SQLite як бази даних, наприклад: Django; Java; PHP; Ruby on Rails; Trac; Symfony; Parser.

Багато програм підтримують SQLite як формат зберігання даних, зокрема:

1. AmaroK – може використовувати бази даних SQLite як сховище музичної колекції.

2. Gajim – SQLite використовується для зберігання історії контактів.

3. Songbird (як застосунок, заснований на XULRunner 1.9)

4. Banshee.

5. F-Spot.

6. Платформа XUL на рушії Gecko 1.9, XULRunner 1.9 і, потенційно, всі застосунки, засновані на цій платформі, у тому числі й Firefox починаючи з версії 3.0.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

XOR з кожним, крім другого, байтом блоку даних і відповідним байтом обраного ключа.

с. Продовжують ці дії з усім блоком даних (з 3 -10 байтами), поки не буде використаний кожний байт для вибору ключа з таблиці після виконання операції XOR з ним і відповідним значенням M1. Потім виконують операцію XOR з кожним, крім використаного для вибору ключа, байтом, і ключем.

d. Повторюють етапи а-с для M2.

Кафедра КБПЗ – 2021 рік

					VKPM-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у магістерської дипломної роботі система відеонагляду реалізованої на базі Axis P1364-E та Axis P1365 -E Mk II. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Функціональних кнопок ПЗ; Верхнього меню; Розділу обрання групи.



Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

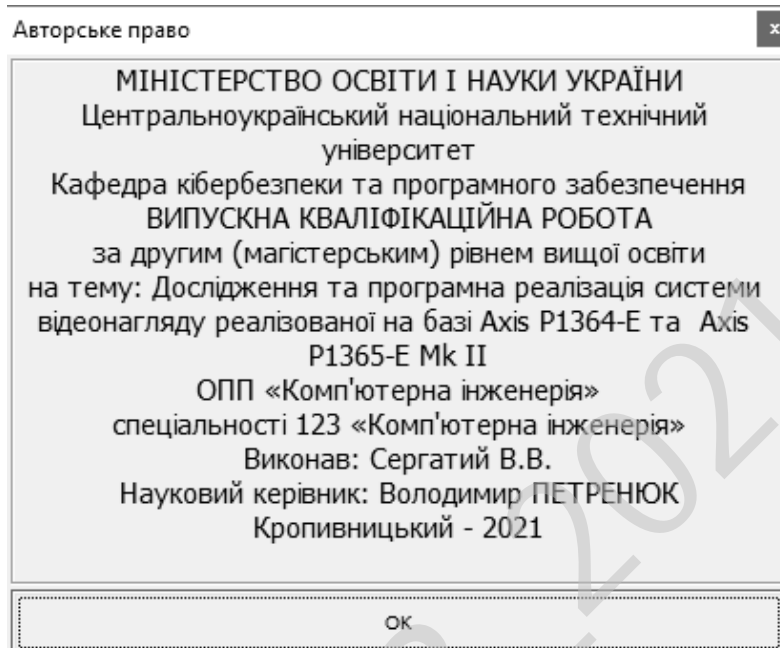


Рисунок 5.2 – Авторське право

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

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

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

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

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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

Ці тести демонструють:

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Метою розробки є дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Об'єктом дослідження є процес відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Предметом дослідження є методи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Методи дослідження базуються на методах теорії обробки мультимедійних даних, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

– Розроблено вітчизняний продукт відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
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)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0018.00.00.ПЗ

Арк.

79

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)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	18000
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	40
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 є найбільш тривалою і трудомісткою, що робить 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 Боема, $A = 2,45$;

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

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

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

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію 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)%.

$$T_{РП} = 0,3 \cdot 3,23 \cdot 12,9^{0,33+0,2(1,027-1,01)} \cdot 83 = 131 \text{ люд/день.}$$

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

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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	15	Д7
Робочий проект	131	Ф 7.1-7.4
Впровадження	15	Д13
Всього	180	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{180 \cdot 1}{24 - 3} = 8,6 \text{ ставки.}$$

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.

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	70	175	3
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	1	285	5
Усього за рік:			З _ч	221

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:

$$\Phi_{op}^c = \frac{Z_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{221 \cdot 1}{1,2} = 184,1 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 184,1 / (24 \cdot 8) = 1 \text{ ставки.}$$

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	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0018.00.00.ПЗ

Арк.

84

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	2	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13500	13500
Продакт-менеджер	0,5	12500	6250
Інженер-програміст	8,6	12000	103200
Інженер-електронщик	1	12000	12000
Інженер-системотехнік	0,25	12000	3000
Адміністратор мережі	0,5	12000	6000
Системний програміст	0,25	12000	3000
Дизайнер WEB	0,5	13000	6500
Інженер-верстальник	0,25	11700	2925
Бухгалтер-економіст	0,5	12500	6250
Всього за період розробки	$R_{cn} = 13,35$	-	$\Phi_{роб} = 162625$

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.

$$Z_{cd} = \frac{162625}{13,35 \cdot 24} = 508 \text{ грн.}$$

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:

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

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 200...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 27 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:

$$B_{y\partial} = 13 \cdot 8 \cdot 20000 = 2080000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 208000 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.

$$I_{нв} = 13 \cdot 3500 = 45500 \text{ грн.}$$

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 за прайсом фірми brain від 17.10.21 – джерело <https://brain.com.ua>.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11457
Системний блок		7509
Процесор	Intel Core i5 2500k 3.30GHz, socket1155, box	4200
Системна плата	Gigabyte GA-H61M-S2PV	1525
Відеокарта	PCIeX: ATI HD4670 SAPPHIRE 1024MB/128bit/DDR2/TV/DualDVI	430
Жорсткий диск	HDD: 320 Gb 7200 Serial ATA WD 16MB	490
Оперативна пам'ять	Kingston DDR3 2GB (KVR1333D3N9/2G Intel/AMD	117
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	216
Корпус	ATX Middle Tower GIGABYTE GZ-X Silver 500W (GZ-X4 Silver)	411
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	120
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1348

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	13	11457	14894,1	163835,1
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	185653,6

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	2080000	-	-
2. Передавальні пристрої	208000	-	-
Всього по групі	2288000	5	114400
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	999,75
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	45500	25	11375
Всього по групі	146999	-	31874,75
7. Нематеріальні активи	18000	10	1800
Разом	$K_p = 2666153$		$A_p = 250476,75$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 3611 USD, що враховуючи курс 27 складає 97500 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.

$$Z_o = 508 \cdot 180 / 180 = 508 \text{ грн.}$$

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, %.

$$Z_d = 508 \cdot 10 \cdot 0,01 = 51 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(508+51) = 123 \text{ грн.}$$

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, %.

$$G_{ocn} = 508 \cdot 15 \cdot 0,01 = 76 \text{ грн.}$$

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-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Згідно виданих викладачем норм приймаємо половину пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 80$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

$$З_{M1} = Ц_n \cdot N_m. \quad (7.16)$$

$$З_{M1} = 80 \cdot 0,5 \cdot 1 = 40 \text{ грн.}$$

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:

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 3 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 3 грн./шт.

$$З_{M2} = 180 \cdot 9 + 12 = 1632 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_{з.}, \quad (7.18)$$

де: $Ц_{з.}$ – 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.

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

$$З_M = (40 + 1632 + 1702) / 180 = 18,7 \text{ грн.}$$

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

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

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

$$O_n = 508 \cdot 15 \cdot 0,01 = 76 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 180$ прим.):

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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

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

$$A_m = 250477 \cdot 1 / (180 \cdot 12) = 116 \text{ грн.}$$

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

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

$$C_n = 508 + 51 + 123 + 76 + 18,7 + 76 + 116 = 968,7 \text{ грн.}$$

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 40%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 968,7 = 378 \text{ грн.}$$

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	508
2. Додаткова зарплата виконавців	Z_d	51
3. Відрахування на соціальні потреби	C_{oc}	123
4. Загальногосподарські витрати	Γ_{ocn}	76
5. Витрати на матеріали	Z_m	18,7
6. Освоєння нових операційних систем, мов програмування	O_n	76

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	116
8. Повна собівартість програмного забезпечення	C_n	968,7
9. Плановий прибуток	P_p	378
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1346,7
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	269,3
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1616

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 7.9.

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1616
Всього капітальних витрат	–	1616

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	33550	20130
2. Витрати на електроенергію	$Z_{ел}$	376	314
3. Витрати на амортизацію	$Z_{ам}$	0	404
Всього витрат за рік	I	33926	20848

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин на обслуговування системи зменшилася з 250 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 250 \cdot 100 \cdot 1,1 \cdot 1,22 = 33550 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 100 \cdot 1,1 \cdot 1,22 = 20130 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел}. \quad (7.24)$$

$$Z_{ел\ баз} = 0,15 \cdot 1140 \cdot 2,2 = 376 \text{ грн.}$$

$$Z_{ел\ нов} = 0,15 \cdot 950 \cdot 2,2 = 314 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	1616	–	404
Всього відрахувань	-	–	1616	–	404

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (Ц_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (1346,7 - 968,7) \cdot 180 - (0,05 \cdot 2288000 + 0,5 \cdot 185654 + 0,25 \cdot 49499 + 0,2 \cdot 97500 + 0,1 \cdot 18000) \cdot 1/12 = 47968 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(Ц_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p – балансова вартість основних фондів розробника.

$$T_e = \frac{2666153}{(1346,7 - 968,7) \cdot 180 \cdot 12 / 1} = 3,3 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	180
2. Повна собівартість розробленої програми	Грн.	968,7
3. Ціна розробленої програми	Грн.	1346,7
4. Плановий прибуток від реалізації розробленої програми	Грн.	378
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2666153
7. Загальний прибуток від реалізації програмної продукції	Грн.	68040
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	47968
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	3,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1616
11. Величина економічного ефекту у користувача програмної продукції	Грн.	12674
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,12

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.27)$$

де: $I_{\text{б}}$, $I_{\text{н}}$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\text{б}}$, $K_{\text{н}}$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{\text{сн}} = (33926 - 20848) - 0,25 \cdot 1616 = 12674 \text{ грн.}$$

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

$$T_{\text{сн}} = \frac{K_{\text{н}} - K_{\text{б}}}{I_{\text{б}} - I_{\text{н}}}, \quad (7.28)$$

$$T_{\text{сн}} = \frac{1616}{33926 - 20848} = 0,12 \text{ року.}$$

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 приймаючих управлінських рішень.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя. Охорона праці є складовою частиною безпеки життя [1].

Законом України “Про охорону праці” [2] 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», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

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 терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

«Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

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;
- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

- 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	7,8
Довжина	9,6
Висота	2,95

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	6,8
Обсяг, V	м ³	не менше 20.0	20

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

У зазначеному приміщенні працює 11 осіб. 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 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», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у 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 приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

висвітленні повинна становити 300 лк. 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. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору розтіканню електричного струму на землю).

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.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

$\rho_2 = 100 \text{ Ом}\cdot\text{м}$. – табличне значення питомого опору нижнього шару ґрунта (джерело: https://zandz.com/ru/udelnoe_soprotivlenie_grunta).

Опір розтіканню електричного струму одного електрода вертикального заземлювача:

$$R_0 = [\rho_1 / (2\pi \cdot L)] \cdot [(\ln) 2L/D + 0,5 \ln[(4T+L)/(4T-L)]] = 40,6 \text{ Ом.}$$

Опір розтіканню електричного струму заземлювача, який нормується, $R=1$ якщо $\rho_{екв} < 100$ (у нас $\rho_{екв} = 93,75 < 100$), та $R = R_{3Н} \cdot \rho_{екв} / 100$, якщо $\rho_{екв} > 100$.

Опір розтіканню електричного струму горизонтального заземлювача (полоси):

$$R_{\Pi} = 0,366 (\rho_{екв} \psi / L_{\Pi} \cdot \eta_{\Pi}) \lg(2 / L_{\Pi} \cdot L_{\Pi} / bt) = 147,98 \text{ Ом.}$$

де $\eta_{\Pi} = 0,3$ – табличне значення коефіцієнта використання горизонтального заземлювача, залежить від розташування (в ряд або по контуру) та співвідношення A/L ; довжина горизонтального заземлювача (полоси) при розташуванні заземлювачів по контуру:

$$L_{\Pi} = A \cdot n = 2 \cdot 2 = 4 \text{ м.};$$

де n – ітераційна кількість вертикальних заземлювачів.

Загальний опір штучного заземлюючого пристрою розтіканню електричного струму на землю:

$$R_B = R_{\Pi} R / (R_{\Pi} - R) = 37,6 \text{ Ом.}$$

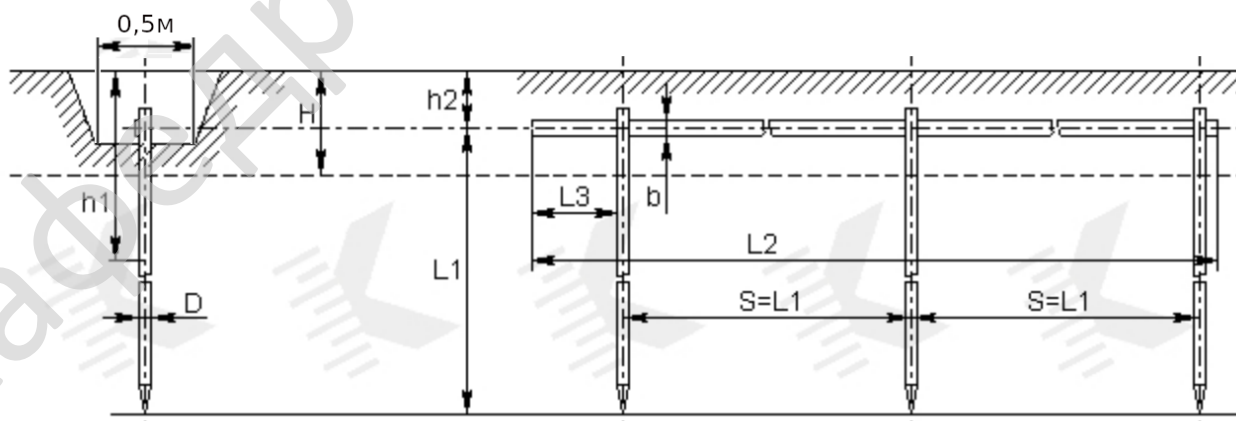


Рисунок 8.1 – Штучний заземлювач

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0018.00.00.ПЗ

Арк.

106

Кількість вертикальних заземлювачів [3]:

$$n = R_0 / R_B * \eta_B = 2 \text{ шт.}$$

де $\eta_B = 0,6$ – табличне значення коефіцієнта використання вертикального заземлювача, залежить від розташування (в ряд або по контуру) та співвідношення A/L.

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-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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 відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.
- Досліджена система відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.
- На основі отриманих результатів досліджень створена програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

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

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

Програма реалізована на мові високого рівня Delphi 10.3.2 Rio. 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 REDOC III.

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 12674 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,12 роки.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергатий В.В. Дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365 -E Mk II // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.

2. К.Н. Стрельников, Д.Л. Куликов. Система моделирования окружения управляемой камеры на основе анализа и обработки видеоданных // Международный журнал «Программные продукты и системы». – 2008.— № 3(83).- С. 66-69.

3. С.В. Гришин, Д.С. Ватолин, К.Н. Стрельников и др. Обзор блочных методов оценки движения в цифровых видео сигналах // Тематический сборник «Программные системы и инструменты».— 2008.— № 9.— С. 50-62.

4. Дреев А.Н. Использование неравномерного распределения единичных битов для дополнительного сжатия SPIHT кода/ А.Н. Дреев, А.А. Смирнов // Информационные системы в управлении, образовании, промышленности: монография. Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – С. 498.

5. Дреев О.М. Дослідження впливу шляху розгортки на ступінь ентропійного стиснення цифрового зображення / О.М. Дреев, О.В. Слюсар // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 21. – Кіровоград: КНТУ. – 2008 – С. 115-118.

6. Дреев О.М. Метод розвантаження телекомунікаційного сервера за рахунок кешування зображень / О.М. Дреев // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 25. Ч. I. – Кіровоград: КНТУ. – 2012 – С. 419-424.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

7. Дреєв О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреєв, О.А. Смірнов, Є.В. Мелешко, О.В. Коваленко // Системи обробки інформації. Випуск 3(101) Том 2. – Х.: ХУПС. – 2012. – С. 181-188.

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)– Х.: ХУПС – 2013. С. 99-102.

9. Дреєв О.М. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смірнов, О.М. Дреєв, О.П. Доренський // Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: 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

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

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. – Х.: ХУПС. – 2012. – С. 206

16. Дреєв О.М. Метод довгострокового прогнозування навантаження серверу телекомунікаційної мережі / О.М. Дреєв, Г.М. Дреєва // Комбінаторні конфігурації та їх застосування. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: “Ексклюзів-систем”. – 2012. – С. 50

17. Дреєв О.М. Вдосконалення стиснення зображень SPIHT методу шляхом додаткового кодування та відкладеної передачі уточнення вейвлет коефіцієнтів / О.М. Дреєв // Дискретна математика та її застосування у економіко-математичному моделюванні та інформаційних технологіях. 11-13 жовтня 2012 р. – Запоріжжя: ЗНУ – 2012. – С. 22-23.

18. Дреєв О.М. Методи підвищення якості обслуговування у телекомунікаційних системах та мережах / О.М. Дреєв, Г.М. Дреєва, О.А. Смірнов // Збірник тез доповідей. Академія внутрішніх військ МВС України “Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку” 20-21 березня 2013р. – Харків: АВВ. – 2013. С. – 18-19

19. Дреєв А.Н. SPIHT кодирование с отложенной передачей значимых битов / А.Н. Дреєв // Тези доповідей. Новітні технології – для захисту повітряного простору. Дев'ята наукова конференція 17 квітня 2013 р. – Х.: ХУПС. – 2013. – С. 206

20. Дреєв А.Н. Повышение оперативности доставки данных повышенной востребованности в телекоммуникационных системах и сетях / А.Н. Дреєв, А.А. Смирнов, Е.В. Мелешко // Проблеми і перспективи розвитку ІТ-індустрії 25-26 квітня 2013 р. Системи обробки інформації. – Випуск 3 (110). Том 2. – Харків: ХУПС. – 2013. С. – 199.

21. Дреєв О.М. Середньостатистичний та найімовірніший час доставки багатопакетного повідомлення в телекомунікаційній системі або

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

мережі / О.М. Дреєв, О.А. Смірнов // V Всеукраїнська науково-практична конференція "Інформатика та системні науки" ІСН–2014, 13-15 березня 2014 року, м. Полтава – С. 92

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>.

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

49. Кучук Г.А. Метод дослідження фрактального мережевого трафіка / Г.А. Кучук // Системи обробки інформації. – Х.: ХВУ, 2005. – Вип. 5 (45). – С. 74-84.

50. Кучук Г.А. Метод агрегування фрактального трафіка / Г.А. Кучук, А.А. Можаяев, О.В. Воробьов // Радіоелектронні та комп'ютерні системи. – 2006. – №6(18). – С. 181-188.

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

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

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

54. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>

					ВКРМ-123.21.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

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-123.21.0018.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Сергачий В.В.				Дослідження та програмна реалізація системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II	Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

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. № 42-13 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 відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.21.0018.00.00.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 відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk II;
- 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-123.21.0018.00.00.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 Delphi 10.3.2 Rio.

					ВКРМ-123.21.0018.00.00.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 повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.21.0018.00.00.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 – 116 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 20.12.2021 р.

					ВКРМ-123.21.0018.00.00.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

_____ Петренюк В.І.

*Дослідження та програмна реалізація
системи відеонагляду реалізованої на базі Axis P1364-E та Axis P1365-E Mk*

II

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 52

Літера: РП

Кропивницький – 2021 року

РОЗРОБЛЕНИЙ ГОЛОВНИЙ ФАЙЛ ПЗ

```
// Назва розробленої програми
program My_Video;
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Виконав: студент 6 курсу, Сергій Владислав Віталійович
Керівник: Петренко В.І.
}
{$R 'RES.res'}
// Підключення ресурсів
Uses
// Підключення бібліотек
  Forms, // форми
// Головне вікно ПЗ
  Video_RTSP in 'Video_RTSP.pas' {Form1},
// Вікно налаштувань
  Settings in 'Settings.pas' {Form2},
// Вікно роботи з мережою
  LAN in 'LAN.pas' {Form3},
/ Вікно авторського права
  About in 'About.pas' {AboutBox};
begin
// З цього місця починається робота коду
// Проходить ініціалізація ПЗ
  Application.Initialize;
// Встановлюється назва ПЗ, яка
// буде відображатись у рамці основного вікна
  Application.Title := 'VIDEO';
// Підключення форми 1
  Application.CreateForm(TForm1, Form1);
// Підключення форми 2
  Application.CreateForm(TForm2, Form2);
// Підключення форми 3
  Application.CreateForm(TForm3, Form3);
// Підключення форми авторського права
  Application.CreateForm(TAboutBox, AboutBox);
// Виведення на екран головної форми ПЗ
  Application.Run;
end.
```

Файл налаштувань

```
unit Settings;
// Назва модулю
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Виконав: студент 6 курсу, Сергій Владислав Віталійович
Керівник: Петренко В.І.
}

interface
{Інтерфейсна частина, тут проходить опис класів та типів}

Uses
// Підключення бібліотек
  Windows,
  WinSock,
  Mobile,
  SysUtils,
  Classes;

const
  mibLen = $0A;

type
  TMibId = array[1..mibLen] of integer;

  TIpConnInfo = class // Об'ява класу
  private // секція бачення
    FRemoteIp :TInAddr;
    FRemotePort :integer;
    FState :DWORD;
    FLocalIp :TInAddr;
    FLocalPort :integer;
    FProto :ShortString;
    function GetLocalPort :Integer;
    function GetLocalIpString :String;
    function Ip2Str( const Ip :TInAddr ):String;
    function GetRemotePort :Integer;
    function GetRemoteIpString :String;
    function GetStateString :String;
  public
    constructor Create( const aProto :String );
    property IpProtoName :ShortString
      read FProto;
    property LocalIp :TInAddr
      read FLocalIp;
    property LocalPort :Integer
      read GetLocalPort;
    property LocalIpString :String
      read GetLocalIpString;
```

```

property RemoteIpString :String
    read GetRemoteIpString;
property RemoteIp :TInAddr
    read FRemoteIp;
property RemotePort :Integer
    read GetRemotePort;
property State :DWORD
    read FState;
property StateString :String
    read GetStateString;
end;

EConnListError = class(Exception); // Об'ява класу
EConnListLockError = class(EConnListError); // Об'ява класу
EConnListUnlockError = class(EConnListError); // Об'ява класу
EConnListRefreshError = class(EConnListError); // Об'ява класу

TIpConnListStatus = ( connlist_ready, connlist_refreshing );

TIpProtocol = (udp_ip, tcp_ip);
TIpProtocols = set of TIpProtocol;

TFnugryIpConnectionList = class(TComponent) // Об'ява класу
private // секція бачення
    FConnections :TList;
    FWSInited :Bool;
    FMobileLib :THandle;
    FMobileQueryProc :Pointer;
    FMobileInitProc :Pointer;
    FProtocols :TIpProtocols;
    FStatus :TIpConnListStatus;
    FOnStatusChange :TNotifyEvent;
    FAccessMutex :THandle;
    FPollForTrapEvent :THandle;
    FSupportedViewRoot :TAsnObjectIdentifier;
    function GetConnCount :Integer;
    function GetConnections( Index :Integer ):TIpConnInfo;
protected
    procedure StatusChange; virtual;
    procedure SetStatus( Value :TIpConnListStatus ); virtual;
    procedure DoLock;
    procedure DoUnlock;
    procedure Clear;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override; // Деструктор
    procedure Refresh;
    function Lock( Timeout :DWORD ):Bool;
    function Unlock :Bool;
    property Status :TIpConnListStatus

```

```

        read FStatus;
    property Connections[ Index :Integer ] :TIpConnInfo
        read GetConnections; default;
    property ConnCount :Integer
        read GetConnCount;
published
    property Protocols :TIpProtocols
        read FProtocols write FProtocols;
    property OnStatusChange :TNotifyEvent
        read FOnStatusChange write FOnStatusChange;
end;

ENetstatError = class(Exception); // Об'ява класу

TVNstatCounterObject = class// Об'ява класу
private // секція бачення
    FName :String;
    FId    :TMibId;
    FDescription :String;
    FWsInited :Bool;
    FMobileLib :THandle;
    FMobileQueryProc :Pointer;
    FMobileInitProc :Pointer;
    function GetValue :DWORD;
public
    constructor Create( const aName, aDesc :String; const aId :TMibId );
    destructor Destroy; override; // Деструктор
    property Description :String
        read FDescription;
    property Name :String
        read FName;
    property Id :TMibId
        read FId;
    property Value :DWORD
        read GetValue;
end;

TCounterList = class(TComponent) // Об'ява класу
private // секція бачення
    FCounters :TStringList;
    function GetCount :Integer;
    function GetCounters( Index :Integer ):TVNstatCounterObject;
public
    constructor Create( aOwner :TComponent ); override;
    destructor Destroy; override; // Деструктор
    procedure Clear;
    procedure AddCounter( aCounter :TVNstatCounterObject );
    function IndexOf( const aName :String ):Integer;
    property Count :Integer
        read GetCount;
    property Counters[Index :Integer] :TVNstatCounterObject

```

```

        read GetCounters; default;
end;

TIpStats = class(TCounterList) // Об'ява класу
public
    constructor Create( aOwner :TComponent ); override; //Конструктор
end;

TRtspStats = class(TCounterList) // Об'ява класу
public
    constructor Create( aOwner :TComponent ); override; //Конструктор
end;

TTCPStats = class(TCounterList) // Об'ява класу
public
    constructor Create( aOwner :TComponent ); override; //Конструктор
end;

TUdpStats = class(TCounterList) // Об'ява класу
public
    constructor Create( aOwner :TComponent ); override; //Конструктор
end;

implementation
{Секція де проходить реалізація того що описано в інтерфейс ній частині}

const
    DEFAULT_LOCK_TIMEOUT = 100;
    MOBILE_LIB_NAME       = 'inetmib1.dll';
    MOBILE_INITPROC_NAME  = 'MobileExtensionInit';
    MOBILE_QUERYPROC_NAME = 'MobileExtensionQuery';

    mib_tcpConnTable :TMibId = (1, 3, 6, 1, 2, 1, 6, 13, 1, 1);
    mib_udpTable : TMibId = (1, 3, 6, 1, 2, 1, 7, 5, 1, 1);
    mib_ipDefaultTTL : TMibId = (1, 3, 6, 1, 2, 1, 4, 2, 1, 1);
    mib_ipInReceives : TMibId = (1, 3, 6, 1, 2, 1, 4, 3, 1, 1);
    mib_ipInHdrErrors : TMibId = (1, 3, 6, 1, 2, 1, 4, 4, 1, 1);
    mib_ipInAddrErrors : TMibId = (1, 3, 6, 1, 2, 1, 4, 5, 1, 1);
    mib_ipForwDatagrams : TMibId = (1, 3, 6, 1, 2, 1, 4, 6, 1, 1);
    mib_InUnknownProtos : TMibId = (1, 3, 6, 1, 2, 1, 4, 7, 1, 1);
    mib_ipInDiscards : TMibId = (1, 3, 6, 1, 2, 1, 4, 8, 1, 1);
    mib_ipInDelivers : TMibId = (1, 3, 6, 1, 2, 1, 4, 9, 1, 1);
    mib_ipOutRequests : TMibId = (1, 3, 6, 1, 2, 1, 4, 10, 1, 1);
    mib_ipOutDiscards : TMibId = (1, 3, 6, 1, 2, 1, 4, 11, 1, 1);
    mib_ipOutNoRoutes : TMibId = (1, 3, 6, 1, 2, 1, 4, 12, 1, 1);
    mib_ipReasmTimeout : TMibId = (1, 3, 6, 1, 2, 1, 4, 13, 1, 1);
    mib_ipReasmReqds : TMibId = (1, 3, 6, 1, 2, 1, 4, 14, 1, 1);
    mib_ipReasmOKs : TMibId = (1, 3, 6, 1, 2, 1, 4, 15, 1, 1);
    mib_ipReasmFails : TMibId = (1, 3, 6, 1, 2, 1, 4, 16, 1, 1);
    mib_ipFragOKs : TMibId = (1, 3, 6, 1, 2, 1, 4, 17, 1, 1);
    mib_ipFragFails : TMibId = (1, 3, 6, 1, 2, 1, 4, 18, 1, 1);

```

```

mib_ipFragCreates : TMibId = (1, 3, 6, 1, 2, 1, 4, 19, 1, 1);
mib_rtspInMsgs : TMibId = (1, 3, 6, 1, 2, 1, 5, 1, 1, 1);
mib_rtspInErrors : TMibId = (1, 3, 6, 1, 2, 1, 5, 2, 1, 1);
mib_rtspInDestUnreachs : TMibId = (1, 3, 6, 1, 2, 1, 5, 3, 1, 1);
mib_rtspInTimeExcds : TMibId = (1, 3, 6, 1, 2, 1, 5, 4, 1, 1);
mib_rtspInParmProbs : TMibId = (1, 3, 6, 1, 2, 1, 5, 5, 1, 1);
mib_rtspInSrcQuenchs : TMibId = (1, 3, 6, 1, 2, 1, 5, 6, 1, 1);
mib_rtspInRedirects : TMibId = (1, 3, 6, 1, 2, 1, 5, 7, 1, 1);
mib_rtspInEchos : TMibId = (1, 3, 6, 1, 2, 1, 5, 8, 1, 1);
mib_rtspInEchoReps : TMibId = (1, 3, 6, 1, 2, 1, 5, 9, 1, 1);
mib_rtspInTimestamps : TMibId = (1, 3, 6, 1, 2, 1, 5, 10, 1, 1);
mib_rtspInTimestampReps : TMibId = (1, 3, 6, 1, 2, 1, 5, 11, 1, 1);
mib_rtspInAddrMasks : TMibId = (1, 3, 6, 1, 2, 1, 5, 12, 1, 1);
mib_rtspInAddrMaskReps : TMibId = (1, 3, 6, 1, 2, 1, 5, 13, 1, 1);
mib_rtspOutMsgs : TMibId = (1, 3, 6, 1, 2, 1, 5, 14, 1, 1);
mib_rtspOutErrors : TMibId = (1, 3, 6, 1, 2, 1, 5, 15, 1, 1);
mib_rtspOutDestUnreachs : TMibId = (1, 3, 6, 1, 2, 1, 5, 16, 1, 1);
mib_rtspOutTimeExcds : TMibId = (1, 3, 6, 1, 2, 1, 5, 17, 1, 1);
mib_rtspOutParmProbs : TMibId = (1, 3, 6, 1, 2, 1, 5, 18, 1, 1);
mib_rtspOutSrcQuenchs : TMibId = (1, 3, 6, 1, 2, 1, 5, 19, 1, 1);

// Константи статусів сокетів (Socket)

CLOSED = 1;
LISTEN = 2;
SYN_SENT = 3;
SYN_RECEIVED = 4;
ESTABLISHED = 5;
CLOSE_WAIT = 6;
FIN_WAIT_1 = 7;
CLOSING = 8;
LAST_ACK = 9;
FIN_WAIT_2 = 10;
TIME_WAIT = 11;
TCB_DISCARD = 12;

type
  TMobileInitProc = function(
    dwTimeZeroReference : DWORD;
    Var hPollForTrapEvent : THandle;
    Var SupportedView : TAsnObjectIdentifier) : BOOL; stdcall;
  TMobileQueryProc = function(
    RequestType : Byte;
    Var VariableVindings : TRFC1157VarBindList;
    Var ErrorStatus : TAsnInteger;
    Var ErrorIndex : TAsnInteger) : BOOL; stdcall;

{ Реалізація TIpStats }

constructor TIpStats.Create( aOwner :TComponent );
//Конструктор
begin

```

```

inherited;
AddCounter( TVNstatCounterObject.Create( 'ipDefaultTTL', '',
mib_ipDefaultTTL));
AddCounter( TVNstatCounterObject.Create( 'ipInReceives', '',
mib_ipInReceives));
AddCounter( TVNstatCounterObject.Create( 'ipInHdrErrors', '',
mib_ipInHdrErrors));
AddCounter( TVNstatCounterObject.Create( 'ipInAddrErrors', '',
mib_ipInAddrErrors));
AddCounter( TVNstatCounterObject.Create( 'ipForwDatagrams', '',
mib_ipForwDatagrams));
AddCounter( TVNstatCounterObject.Create( 'ipInUnknownProtos', '',
mib_InUnknownProtos));
AddCounter( TVNstatCounterObject.Create( 'ipInDiscards', '',
mib_ipInDiscards));
AddCounter( TVNstatCounterObject.Create( 'ipInDelivers', '',
mib_ipInDelivers));
AddCounter( TVNstatCounterObject.Create( 'ipOutRequests', '',
mib_ipOutRequests));
AddCounter( TVNstatCounterObject.Create( 'ipOutDiscards', '',
mib_ipOutDiscards));
AddCounter( TVNstatCounterObject.Create( 'ipOutNoRoutes', '',
mib_ipOutNoRoutes));

AddCounter( TVNstatCounterObject.Create( 'ipReasmTimeout', '',
mib_ipReasmTimeout));
AddCounter( TVNstatCounterObject.Create( 'ipReasmReqds', '',
mib_ipReasmReqds));
AddCounter( TVNstatCounterObject.Create( 'ipReasmOKs', '',
mib_ipReasmOKs));
AddCounter( TVNstatCounterObject.Create( 'ipReasmFails', '',
mib_ipReasmFails));
AddCounter( TVNstatCounterObject.Create( 'ipFragOKs', '',
mib_ipFragOKs));
AddCounter( TVNstatCounterObject.Create( 'ipFragFails', '',
mib_ipFragFails));
AddCounter( TVNstatCounterObject.Create( 'ipFragCreates', '',
mib_ipFragCreates));

end;

{ Реализация TRtspStats }
constructor TRtspStats.Create( aOwner :TComponent ); //Конструктор
begin
    inherited;
AddCounter( TVNstatCounterObject.Create('rtspInMsgs','',mib_rtspInMsgs ));
AddCounter( TVNstatCounterObject.Create( 'rtspInErrors', '',
mib_rtspInErrors ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInDestUnreachs', '',
mib_rtspInDestUnreachs ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInTimeExcds', '',
mib_rtspInTimeExcds ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInParmProbs', '',
mib_rtspInParmProbs ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInSrcQuenchs', '',
mib_rtspInSrcQuenchs ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInRedirects', '',
mib_rtspInRedirects ));

```

```

    AddCounter( TVNstatCounterObject.Create( 'rtspInEchos', '', mib_rtspInEchos
));
    AddCounter( TVNstatCounterObject.Create( 'rtspInEchoReps', '',
mib_rtspInEchoReps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInTimestamps', '',
mib_rtspInTimestamps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInTimestampReps', '',
mib_rtspInTimestampReps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInAddrMasks', '',
mib_rtspInAddrMasks ));
    AddCounter( TVNstatCounterObject.Create( 'rtspInAddrMaskReps', '',
mib_rtspInAddrMaskReps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutMsgs', '', mib_rtspOutMsgs
));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutErrors', '',
mib_rtspOutErrors ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutDestUnreaches', '',
mib_rtspOutDestUnreaches ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutTimeExcds', '',
mib_rtspOutTimeExcds ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutParmProbs', '',
mib_rtspOutParmProbs ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutSrcQuenchs', '',
mib_rtspOutSrcQuenchs ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutRedirects', '',
mib_rtspOutRedirects ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutEchos', '', mib_rtspOutEchos
));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutEchoReps', '',
mib_rtspOutEchoReps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutTimestamps', '',
mib_rtspOutTimestamps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutTimestampReps', '',
mib_rtspOutTimestampReps ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutAddrMasks', '',
mib_rtspOutAddrMasks ));
    AddCounter( TVNstatCounterObject.Create( 'rtspOutAddrMaskReps', '',
mib_rtspOutAddrMaskReps ));
end;

{ Реализация TCP Stats }

constructor TTcpStats.Create( aOwner :TComponent ); //Конструктор
begin
    inherited;
    AddCounter( TVNstatCounterObject.Create( 'tcpRtoAlgorithm', '',
mib_tcpRtoAlgorithm));
    AddCounter( TVNstatCounterObject.Create( 'tcpRtoMin', '', mib_tcpRtoMin));
    AddCounter( TVNstatCounterObject.Create( 'tcpRtoMax', '', mib_tcpRtoMax));
    AddCounter( TVNstatCounterObject.Create( 'tcpMaxConn', '', mib_tcpMaxConn));
    AddCounter( TVNstatCounterObject.Create( 'tcpActiveOpens', '',
mib_tcpActiveOpens));
    AddCounter( TVNstatCounterObject.Create( 'tcpPassiveOpens', '',
mib_tcpPassiveOpens));
    AddCounter( TVNstatCounterObject.Create( 'tcpAttemptsFails', '',
mib_tcpAttemptsFails));

```

```

    AddCounter( TVNstatCounterObject.Create( 'tcpEstabResets', '',
mib_tcpEstabResets));
    AddCounter( TVNstatCounterObject.Create( 'tcpCurrEstab', '',
mib_tcpCurrEstab));
    AddCounter( TVNstatCounterObject.Create( 'tcpInSegs', '', mib_tcpInSegs));
    AddCounter( TVNstatCounterObject.Create( 'tcpOutSegs', '', mib_tcpOutSegs));
    AddCounter( TVNstatCounterObject.Create( 'tcpRetransSegs', '',
mib_tcpRetransSegs));
    AddCounter( TVNstatCounterObject.Create( 'tcpInErrs', '', mib_tcpInErrs));
    AddCounter( TVNstatCounterObject.Create( 'tcpOutRsts', '', mib_tcpOutRsts));
end;

constructor TUDPStats.Create( aOwner :TComponent ); //Конструктор
begin
    inherited;

    AddCounter( TVNstatCounterObject.Create( 'udpInDatagrams', '',
mib_udpInDatagrams ));
    AddCounter( TVNstatCounterObject.Create( 'udpNoPorts', '', mib_udpNoPorts ));
    AddCounter( TVNstatCounterObject.Create( 'udpInErrors', '', mib_udpInErrors
));
    AddCounter( TVNstatCounterObject.Create( 'udpOutDatagrams', '',
mib_udpOutDatagrams ));
end;

{ Реалізація TVNstatCounterObject }

function TVNstatCounterObject.GetValue :DWORD; // Реалізація функції
var // Об'ява змінних
    varBind      :TRFC1157VarBind;
    varBindList  :TRFC1157VarBindList;
    errorStatus  :TAsnInteger;
    errorIndex   :TAsnInteger;
begin
    varBindList.List := @varBind;
    varBindList.len := 1;
    fillchar(varBind, SizeOf(varBind), 0);
    varBind.Name.idLength := mibLen;
    varBind.Name.ids := @FId;
    if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
varBindList, errorStatus, errorIndex) then
        raise ENetstatError.Create(m_err_query);
    if not (varBindList.list.value.asnType in
[ASN_GAUGE32, ASN_INTEGER, ASN_INTEGER32, ASN_COUNTER32, ASN_UNSIGNED32])
then
        raise ENetstatError.Create(m_err_invtype);
    result := varBindList.list.value.Counter;
end;
//Конструктор
constructor TVNstatCounterObject.Create(
    const aName, aDesc :String; const aId :TMibId );
var
    WSDData :TWSADData;

```

```

PollForTrapEvent :THandle;
SupportedViewRoot :TAsnObjectIdentifier;
begin
    inherited Create;
    FWSInited := WsaStartup($0101, WSDData ) = 0;
    if not FWSInited then
        raise EConnListError.Create(m_err_wsstartup);
    FMobileLib := LoadLibrary(MOBILE_LIB_NAME);
    if FMobileLib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FMobileInitProc := GetProcAddress(FMobileLib, MOBILE_INITPROC_NAME);
    FMobileQueryProc := GetProcAddress(FMobileLib, MOBILE_QUERYPROC_NAME);
    if ( ( FMobileQueryProc = Nil ) or ( FMobileInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TMobileInitProc(FMobileInitProc)(GetTickCount,
        PollForTrapEvent, SupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FName := aName;
    FDescription := aDesc;
    FId := aid;
    dec(Fid[8]);
end;

destructor TVNstatCounterObject.Destroy; // Деструктор
begin
    if FMobileLib <> 0 then FreeLibrary(FMobileLib);
    if FWSInited then WSACleanup;
    inherited;
end;

{ Реалізація TCounterList }

procedure TCounterList.AddCounter( aCounter :TVNstatCounterObject );
// Реалізація процедури
begin
    assert( assigned( aCounter ) );
    FCounters.AddObject( aCounter.Name, aCounter );
end;

function TCounterList.GetCount :Integer; // Реалізація функції
begin
    assert( assigned( FCounters ) );
    result := FCounters.Count;
end;

function TCounterList.GetCounters( Index :Integer ):TVNstatCounterObject;
// Реалізація функції
begin
    assert( assigned( FCounters ) );
    result := TVNstatCounterObject(FCounters.Objects[Index]);
end;

```

```

function TCounterList.IndexOf( const aName :String ):Integer;
// Реалізація функції
begin
    result := FCounters.IndexOf( aName );
end;

constructor TCounterList.Create( aOwner :TComponent ); //Конструктор
begin
    inherited;
    FCounters := TStringList.Create;
end;

destructor TCounterList.Destroy; // Деструктор
begin
    Clear;
    if assigned(FCounters) then FCounters.Free;
    inherited;
end;

procedure TCounterList.Clear; // Реалізація процедури
var // Об'ява змінних
    i :integer;
    c :TObject;
begin
    if assigned(FCounters) then
        for i := FCounters.Count-1 downto 0 do
            begin
                c := FCounters.Objects[i];
                FCounters.Delete(i);
                if assigned(c) then c.free;
            end;
        end;
end;

{ Реалізація TIpConNInfo }

constructor TIpConNInfo.Create( const aProto :String ); //Конструктор
begin
    inherited Create;
    FState := LISTEN;
    FProto := aProto;
end;

function TIpConNInfo.Ip2Str( const Ip :TInAddr ):String;
// Реалізація функції
begin
    result := format('%d.%d.%d.%d',
        [Integer(ip.s_un_b.s_b1),
        Integer(ip.s_un_b.s_b2),
        Integer(ip.s_un_b.s_b3),
        Integer(ip.s_un_b.s_b4)]);
end;

```

```

end;

function TIpConNInfo.GetLocalPort :Integer; // Реалізація функції
begin
    result := FLocalPort;
end;

function TIpConNInfo.GetLocalIpString :String; // Реалізація функції
begin
    result := Ip2Str(FLocalIp);
end;

function TIpConNInfo.GetRemotePort :Integer; // Реалізація функції
begin
    if State = LISTEN then
        result := 0
    else
        result := FRemotePort;
    end;
end;

function TIpConNInfo.GetRemoteIpString :String; // Реалізація функції
begin
    result := Ip2Str(FRemoteIp);
end;

function TIpConNInfo.GetStateString :String; // Реалізація функції
const
    state_name :array[CLOSED..TCB_DISCARD] of string[16] =
        ('CLOSED',
         'LISTEN',
         'SYN_SENT',
         'SYN_RECEIVED',
         'ESTABLISHED',
         'CLOSE_WAIT',
         'FIN_WAIT_1',
         'CLOSING',
         'LAST_ACK',
         'FIN_WAIT_2',
         'TIME_WAIT',
         'TCB_DISCARD'
        );
begin
    if State in [CLOSED..TCB_DISCARD] then
        result := state_name[state]
    else
        result := 'UNKNOWN';
    end;
end;

{ Реалізація TFnugryIpConnectionList }

```

```

function TFnugryIpConnectionList.GetConnCount :Integer;
// Реалізація функції
begin
    assert(assigned(FConnections));
    result := FConnections.Count;
end;

function TFnugryIpConnectionList.GetConnections( Index :Integer ):TIpConnInfo;
// Реалізація функції
begin
    assert(assigned(FConnections));
    result := FConnections[Index];
end;

procedure TFnugryIpConnectionList.Clear; // Реалізація процедури
var // Об'ява змінних
    i :integer;
    p :TIpConnInfo;
begin
    if assigned(FConnections) then
        for i := FConnections.Count-1 downto 0 do
            begin
                p := FConnections[i];
                assert(assigned(p));
                FConnections.Delete(i);
                p.free;
            end;
        end;
end;

procedure TFnugryIpConnectionList.StatusChange; // Реалізація процедури
begin
    if assigned(FOnStatusChange) then FOnStatusChange(Self);
end;

procedure TFnugryIpConnectionList.SetStatus( Value :TIpConnListStatus );
// Реалізація процедури
begin
    if FStatus <> Value then
        begin
            FStatus := Value;
            StatusChange;
        end;
end;

procedure TFnugryIpConnectionList.DoLock; // Реалізація процедури
begin
    if not Lock(DEFAULT_LOCK_TIMEOUT) then
        raise EConnListLockError.Create(m_err_lock);
end;

procedure TFnugryIpConnectionList.DoUnlock;
// Реалізація процедури

```

```

begin
    if not UnLock then
        raise EConnListUnLockError.Create(m_err_unlock);
    end;

constructor TFnugryIpConnectionList.Create( aOwner :TComponent );
// Конструктор
var // Об'ява змінних
    WSDData :TWSADData;
begin
    inherited;
    FProtocols := [udp_ip, tcp_ip];
    FWsInited := WsaStartup($0101, WSDData ) = 0;
    if not FWsInited then
        raise EConnListError.Create(m_err_wsstartup);
    FMobileLib := LoadLibrary(MOBILE_LIB_NAME);
    if FMobileLib = 0 then
        raise EConnListError.Create(m_err_loadlib);
    FMobileInitProc := GetProcAddress(FMobileLib, MOBILE_INITPROC_NAME);
    FMobileQueryProc := GetProcAddress(FMobileLib, MOBILE_QUERYPROC_NAME);
    if ( ( FMobileQueryProc = Nil ) or ( FMobileInitProc = Nil ) ) then
        raise EConnListError.Create(m_err_loadlib);
    if not TMobileInitProc(FMobileInitProc)(GetTickCount,
        FPollForTrapEvent, FSupportedViewRoot) then
        raise EConnListError.Create(m_err_initlib);
    FAccessMutex := CreateMutex(nil, false, nil);
    if FAccessMutex = 0 then
        raise EConnListError.Create(m_err_alloc);
    FConnections := TList.Create;
    FStatus := connlist_ready;
end;

destructor TFnugryIpConnectionList.Destroy; // Деструктор
begin
    Clear;
    if assigned(FConnections) then FConnections.Free;
    if FAccessMutex <> 0 then CloseHandle(FAccessMutex);
    if FMobileLib <> 0 then FreeLibrary(FMobileLib);
    if FWsInited then WSACleanup;
    inherited;
end;

function TFnugryIpConnectionList.Lock( Timeout :DWORD ):Bool;
// Реалізація функції
begin
    result := WaitForSingleObject(FAccessMutex, Timeout ) = WAIT_OBJECT_0;
end;

function TFnugryIpConnectionList.Unlock :Bool; // Реалізація функції
begin

```

```

    result := ReleaseMutex(FAccessMutex);
end;

procedure TFnugryIpConnectionList.Refresh;
// Реалізація процедури
procedure ReadTcpTable;
var // Об'ява змінних
    varBind      :TRFC1157VarBind;
    varBindList  :TRFC1157VarBindList;
    errorStatus  :TAsnInteger;
    errorIndex   :TAsnInteger;
    ConnIndex    :Integer;
    Info         :TIpConnInfo;
    ListTail     :Integer;
begin
    fillchar( varBindList, SizeOf(varBindList), 0);
    varBindList.List := @varBind;
    varBindList.len := 1;
    fillchar(varBind, SizeOf(varBind), 0);
    varBind.Name.idLength := mibLen;
    varBind.name.ids := @mib_tcpConnTable;
    ListTail := FConnections.Count;
    if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    if varBindList.list.value.asnType = ASN_NULL then
        exit;
    {читання з'єднань }
    while (varBindList.list.value.asnType = ASN_INTEGER) do
    begin
        Info := TIpConnInfo.Create('TCP');
        Info.FState := varBindList.list.value.Counter;
        FConnections.Add(Info);
        if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
            varBindList, errorStatus, errorIndex) then
            raise EConnListRefreshError.Create(m_err_query);
        end;
        if varBindList.list.value.asnType = ASN_NULL then
            raise EConnListRefreshError.Create(m_err_queryend);
        { читання значень ips }
        ConnIndex := ListTail;
        while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
        begin
            move(varBindList.list.value.address.stream^,
                Connections[ConnIndex].FLocalIP, sizeof(TInAddr));
            if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
                varBindList, errorStatus, errorIndex) then
                raise EConnListRefreshError.Create(m_err_query);
            inc(ConnIndex);
        end;
        if varBindList.list.value.asnType = ASN_NULL then

```

```

    raise EConnListRefreshError.Create(m_err_queryend);
{ читання значень локальних відео портів TCP/IP }
ConnIndex := ListTail;

while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
    if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;

if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання значень віддалених відео портів TCP/IP }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do
begin
    move(varBindList.list.value.address.stream^,
        Connections[ConnIndex].FRemoteIP,
        sizeof(TInAddr));
    if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;

if varBindList.list.value.asnType = ASN_NULL then
    raise EConnListRefreshError.Create(m_err_queryend);
{ читання значень віддалених відео портів TCP/IP }
ConnIndex := ListTail;

while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
    Connections[ConnIndex].FRemotePort := varBindList.list.value.counter;
    if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
        varBindList, errorStatus, errorIndex) then
        raise EConnListRefreshError.Create(m_err_query);
    inc(ConnIndex);
end;
end;

procedure ReadUdpTable;
var // Об'ява змінних
    varBind      :TRFC1157VarBind;
    varBindList  :TRFC1157VarBindList;
    errorStatus  :TAsnInteger;
    errorIndex   :TAsnInteger;
    ConnIndex    :Integer;
    Info         :TIpConnInfo;

```

```

ListTail      :Integer;
begin
fillchar( varBindList, SizeOf(varBindList), 0);
varBindList.List := @varBind;
varBindList.len := 1;
fillchar(varBind, SizeOf(varBind), 0);
varBind.Name.idLength := mibLen;

varBind.name.ids := @mib_udpTable;
ListTail := FConnections.Count;
if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
varBindList, errorStatus, errorIndex) then
raise EConnListRefreshError.Create(m_err_query);
if varBindList.list.value.asnType = ASN_NULL then
exit;
{ читання значень локальних відео портів TCP/IP }
while (varBindList.list.value.asnType = ASN_RFC1155_IPADDRESS) do

begin
Info := TIpConnInfo.Create('UDP');
move(varBindList.list.value.address.stream^,
Info.FLocalIP,
sizeof(TInAddr));
FConnections.Add(Info);
if not TMobileQueryProc(FMobileQueryProc)(ASN_RFC1157_GETNEXTREQUEST,
varBindList, errorStatus, errorIndex) then
raise EConnListRefreshError.Create(m_err_query);
end;
if varBindList.list.value.asnType = ASN_NULL then
raise EConnListRefreshError.Create(m_err_queryend);
{ отримання даних }
ConnIndex := ListTail;
while (varBindList.list.value.asnType = ASN_INTEGER) do
begin
Connections[ConnIndex].FLocalPort := varBindList.list.value.counter;
if not TMobileQueryProc(FMobileQueryProc)
(ASN_RFC1157_GETNEXTREQUEST, varBindList, errorStatus,
errorIndex)
then raise EConnListRefreshError.Create(m_err_query);
inc(ConnIndex);

end;
end;

begin
DoLock;
try
if FStatus <> connlist_ready then
raise EConnListError.Create(m_err_inv_state);
SetStatus( connlist_refreshing );
Clear;

```

```
    if tcp_ip in FProtocols then
        ReadTcpTable;
    if udp_ip in FProtocols then

        ReadUdpTable;
finally
    DoUnlock;
    SetStatus(connlist_ready);

end;
end;

end.
```

Кафедра КБПЗ – 2021 рік

ФАЙЛ АВТОРСЬКОГО ПРАВА

```

unit About; // Назва модулю
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Виконав: студент 6 курсу, Сергій Владислав Віталійович
Керівник: Петренко В.І.
}

interface
{Інтерфейсна частина, тут проходить опис класів та типів}

Uses
// Підключення бібліотек
  WinTypes, WinProcs, Classes, Graphics, Forms, Controls,
  StdCtrls, ExtCtrls, Buttons, DsgnIntf, MgCommon, ShellAPI;

const
  mgcrHand = 5;

type
  TMgAboutBoxProperty = class( TPropertyEditor ) // Об'ява класу
  public
    procedure Edit; override;
    function GetAttributes : TPropertyAttributes; override;
    function GetValue : string; override;
  end;

  TMgAboutEditDlg = class( TForm )
  // Об'ява класу
    BtnOk: TButton;
    Panel1 : TPanel;
    LblCopyright: TLabel;
    BtnPaletteBmp: TSpeedButton;
    LblDescription: TLabel;
    LblCompany: TLabel;
    Bevel1: TBevel;
    LblComponentName: TLabel;
    lblURL: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure lblURLClick(Sender: TObject);
    procedure lblURLMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure lblURLMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
  private
  // секція бачення
    FAboutInfo : TMgAboutInfo;
    procedure SetAboutInfo( Value : TMgAboutInfo );
  public

```

```

    ComponentName : string;
    property AboutInfo : TMgAboutInfo read FAboutInfo write SetAboutInfo;
end;

implementation
{Секція де проходить реалізація того що описано в інтерфейсній частині}

{$R MgCursor.res}
{$R *.DFM}

Uses
// Підключення бібліотек
    SysUtils;

{ Реалізація класу TRzAboutBoxProperty      }

function TMgAboutBoxProperty.GetAttributes : TPropertyAttributes;
// Реалізація функції
begin
    Result := [ paDialog, paReadOnly ];
end;

function TMgAboutBoxProperty.GetValue : string; // Реалізація функції
begin
    Result := 'Повернення';
end;

procedure TMgAboutBoxProperty.Edit; // Реалізація процедури
var // Об'ява змінних
    ComponentRef : TComponent;
    Dialog : TMgAboutEditDlg;
begin
    Dialog := TMgAboutEditDlg.Create( Application );
    try
        ComponentRef := GetComponent( 0 );
    { Отримання назви класу з компоненту }
        Dialog.ComponentName := ComponentRef.ClassName;
        Dialog.AboutInfo := TMgAboutInfo( GetOrdValue );
        Dialog.ShowModal;
    finally
        Dialog.Free;
    end;
end;

{Реалізація класу TMgAboutEditDlg}

procedure TMgAboutEditDlg.SetAboutInfo( Value : TMgAboutInfo );
// Реалізація процедури
var // Об'ява змінних
    BmpName : array[ 0..128 ] of Char;
begin

```

```

if ComponentName[ 1 ] = 'T' then
  LblComponentName.Caption := Copy( ComponentName, 2, 255 )
else
  LblComponentName.Caption := ComponentName;

LblCopyright.Caption := 'Copyright © '+ Value.CopyrightDate;
LblCompany.Caption := Value.Company;
LblDescription.Caption := Value.Description;
LblURL.Caption := Value.URL;
{ Завантаження зображення }
StrPCopy( BmpName, UpperCase( ComponentName ) );
BtnPaletteBmp.Glyph.Handle := LoadBitmap( HInstance, BmpName );
end;

procedure TMgAboutEditDlg.FormCreate(Sender: TObject); // Реалізація процедури
begin
  Screen.Cursors[ mgcrHand ] := LoadCursor( HInstance, PChar( 'MGHAND' ) );
end;

procedure TMgAboutEditDlg.lblURLClick(Sender: TObject); // Реалізація процедури
var // Об'ява змінних
  tempStr : array[0..255] of Char;
begin
  ShellExecute( Application.Handle, 'open', StrPCopy(tempStr, lblURL.Caption),
    nil, nil, SW_NORMAL );
end;

procedure TMgAboutEditDlg.lblURLMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
// Реалізація процедури
begin
  if Button = mbLeft then begin
    Font.Color := clPurple;
  end;
end;

procedure TMgAboutEditDlg.lblURLMouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
// Реалізація процедури
begin
  if Button = mbLeft then Font.Color := clBlue;
end;

end.

```

ФАЙЛ БІБЛІОТЕКИ ВІДЕО VIDEO_PACK

```

unit Video_Pack; // Бібліотека відео
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Виконав: студент 6 курсу, Сергій Владислав Віталійович
Керівник: Петренко В.І.
}

interface // Інтерфейсна частина

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

Const // Константи
  WM_GRAPHNOTIFY = WM_APP + 1;

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

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

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

  {@exclude}
  TGraphState = ( // Поточне дія

```

```

    gsUninitialized,
    gsStopped,
    gsPaused,
    gsPlaying
);

{ Структура відео потоку }

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

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

PVMRPreferences = ^TVMRPreferences;
TVMRPreferences = set of TVMRPreference;

TOnDSEvent=procedure(sender: TComponent; Event, Param1, Param2: Integer) of
    object;
TOnGraphBufferingData=procedure(sender: TObject; Buffering: boolean) of object ;
TOnGraphComplete=procedure(sender: TObject; Result: HRESULT; Renderer:
IBaseFilter) of object;
    TOnGraphDeviceLost                = procedure(sender: TObject; Device: IUnknown;
        Removed: Boolean) of object ;
    TOnGraphEndOfSegment                = procedure(sender: TObject; StreamTime:
TReferenceTime; NumSegment: Cardinal) of object ;
    TOnDSResult                          = procedure(sender: TObject; Result: HRESULT) of
object ;
    TOnGraphFullscreenLost              = procedure(sender: TObject; Renderer:
IBaseFilter) of object ;
    TOnGraphOleEvent                    = procedure(sender: TObject; String1, String2:
WideString) of object ;
    TOnGraphOpeningFile                  = procedure(sender: TObject; opening: boolean) of
object ;
    TOnGraphSNDDDevError                 = procedure(sender: TObject; OccurWhen:
TSndDevErr; ErrorCode: LongWord) of object ;
    TOnGraphStreamControl                 = procedure(sender: TObject; PinSender: IPin;
        Cookie: LongWord) of object ;

```

```

TOnGraphStreamError          = procedure(sender: TObject; Operation: HRESULT;
                                         Value: LongWord) of object ;

TOnGraphVideoSizeChanged     = procedure(sender: TObject; Width, height: word)
                                         of object ;

TOnGraphTimeCodeAvailable= procedure(sender: TObject; From: IBaseFilter;
                                         DeviceID: LongWord) of object ;

TOnGraphEXTDeviceModeChange = procedure(sender: TObject; NewMode, DeviceID:
                                         LongWord) of object ;

TOnGraphVMRRenderDevice= procedure(sender: TObject; RenderDevice:
TVMRRenderDevice) of object;

TOnCAMDataStreamChange       = procedure(sender: TObject; stream, lcid: Integer;
                                         Lang: string) of object;

TOnCAMCurrentTime            = procedure(sender: TObject; Hours,
                                         minutes,seconds,frames,frate : Integer) of object;

TOnCAMTitleChange           = procedure(sender: TObject; title: Integer) of object;
TOnCAMChapterStart=procedure(sender: TObject; chapter: Integer) of object;
TOnCAMValidUOPChange        = procedure(sender: TObject; UOPS: Integer) of object;
TOnCAMChange                = procedure(sender: TObject; total,current: Integer) of object;
TOnCAMStillOn               = procedure(sender: TObject; NoButtonAvailable: boolean; seconds:
                                         Integer) of object;

TOnCAMSubpictureStreamChange = procedure(sender: TObject; SubNum, lcid: Integer;
Lang: string) of object;

TOnCAMPlaybackRateChange    = procedure(sender: TObject; rate: single) of object;
TOnCAMParentalLevelChange= procedure(sender: TObject; level: Integer) of object;
TOnCAMAnglesAvailable=procedure(sender: TObject; available: boolean) of object;
TOnCAMButtonAutoActivated=procedure(sender: TObject; Button: Cardinal) of
object;

TOnCAMCMD=procedure(sender: TObject; CmdID: Cardinal) of object;
TOnCAMCurrentHMSFTime       = procedure(sender: TObject; HMSFTimeCode:
                                         TCAMHMSFTimeCode; TimeCode: TCAMTimeCode) of object;
TOnCAMKaraokeMode=procedure(sender: TObject; Played: boolean) of object;
TOnBuffer                   = procedure(sender: TObject; SampleTime: Double; pBuffer: Pointer;
BufferLen: longint) of object ;

// Структура TFilterOperation

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

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

TControlEvent = (
    cePlay,
    cePause,
    ceStop,

```

```

ceFileRendering,
ceFileRendered,
ceCAMRendering,
ceCAMRendered,
ceActive
);

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

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

  FFilters: TInterfaceList;
  FGraphEvents: TInterfaceList;

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

  // Файл Log
  FLogFileName: String;
  FLogFile: TFileStream;
  FOnActivate: TNotifyEvent;
  // All Events Code
  FOnDSEvent : TOnDSEvent;
  // Generic Graph Events
  FOnGraphBufferingData : TOnGraphBufferingData;
  FOnGraphClockChanged : TNotifyEvent;
  FOnGraphComplete : TOnGraphComplete;
  FOnGraphDeviceLost : TOnGraphDeviceLost;
  FOnGraphEndOfSegment : TOnGraphEndOfSegment;
  FOnGraphErrorStillPlaying : TOnDSResult;
  FOnGraphErrorAbort : TOnDSResult;
  FOnGraphFullscreenLost : TOnGraphFullscreenLost;
  FOnGraphChanged : TNotifyEvent;
  FOnGraphOleEvent : TOnGraphOleEvent;
  FOnGraphOpeningFile : TOnGraphOpeningFile;
  FOnGraphPaletteChanged : TNotifyEvent;

```

```

FOnGraphPaused : TOnDSResult;
FOnGraphQualityChange : TNotifyEvent;
FOnGraphSNDDDevInError : TOnGraphSNDDDevError;
FOnGraphSNDDDevOutError : TOnGraphSNDDDevError;
FOnGraphStepComplete : TNotifyEvent;
FOnGraphStreamControlStarted : TOnGraphStreamControl;
FOnGraphStreamControlStopped : TOnGraphStreamControl;
FOnGraphStreamErrorStillPlaying : TOnGraphStreamError;
FOnGraphStreamErrorStopped : TOnGraphStreamError;
FOnGraphUserAbort : TNotifyEvent;
FOnGraphVideoSizeChanged : TOnGraphVideoSizeChanged;
FOnGraphTimeCodeAvailable : TOnGraphTimeCodeAvailable;
FOnGraphEXTDeviceModeChange : TOnGraphEXTDeviceModeChange;
FOnGraphClockUnset : TNotifyEvent;
FOnGraphVMRRenderDevice : TOnGraphVMRRenderDevice;

FOnCAMDataStreamChange : TOnCAMDataStreamChange;
FOnCAMCurrentTime : TOnCAMCurrentTime;
FOnCAMTitleChange : TOnCAMTitleChange;
FOnCAMChapterStart : TOnCAMChapterStart;
FOnCAMAngleChange : TOnCAMChange;
FOnCAMValidUOPSChange : TOnCAMValidUOPSChange;
FOnCAMButtonChange : TOnCAMChange;
FOnCAMChapterAutoStop : TNotifyEvent;
FOnCAMStillOn : TOnCAMStillOn;
FOnCAMStillOff : TNotifyEvent;
FOnCAMSubpictureStreamChange : TOnCAMSubpictureStreamChange;
FOnCAMNoFP_PGC : TNotifyEvent;
FOnCAMPlaybackRateChange : TOnCAMPlaybackRateChange;
FOnCAMParentalLevelChange : TOnCAMParentalLevelChange;
FOnCAMPlaybackStopped : TNotifyEvent;
FOnCAMAnglesAvailable : TOnCAMAnglesAvailable;
FOnCAMPlayPeriodAutoStop : TNotifyEvent;
FOnCAMButtonAutoActivated : TOnCAMButtonAutoActivated;
FOnCAMCMDStart : TOnCAMCMD;
FOnCAMCMDEnd : TOnCAMCMD;
FOnCAMDiscEjected : TNotifyEvent;
FOnCAMDiscInserted : TNotifyEvent;
FOnCAMCurrentHMSFTime : TOnCAMCurrentHMSFTime;
FOnCAMKaraokeMode : TOnCAMKaraokeMode;
FOnCAMWarningInvalidCAM1_0Disc : TNotifyEvent;//=1,
FOnCAMWarningFormatNotSupported : TNotifyEvent;//=2,
FOnCAMWarningIllegalNavCommand : TNotifyEvent;//=3
FOnCAMWarningOpen : TNotifyEvent;//=4
FOnCAMWarningSeek : TNotifyEvent;//=5
FOnCAMWarningRead : TNotifyEvent;//=6
FOnCAMDomainFirstPlay : TNotifyEvent;
FOnCAMDomainVideoManagerMenu : TNotifyEvent;
FOnCAMDomainVideoTitleSetMenu : TNotifyEvent;
FOnCAMDomainTitle : TNotifyEvent;
FOnCAMDomainStop : TNotifyEvent;

```

```

FOnCAMErrorUnexpected           : TNotifyEvent;
FOnCAMErrorCopyProtectFail     : TNotifyEvent;
FOnCAMErrorInvalidCAM1_Disc    : TNotifyEvent;
FOnCAMErrorInvalidDiscRegion  : TNotifyEvent;
FOnCAMErrorLowParentalLevel    : TNotifyEvent;
FOnCAMErrorMacrovisionFail     : TNotifyEvent;
FOnCAMErrorIncompatibleSystemAndDecoderRegions : TNotifyEvent;
FOnCAMErrorIncompatibleDiscAndDecoderRegions : TNotifyEvent;
procedure HandleEvents;
procedure WndProc(var Msg: TMessage);
procedure SetActive(Activate: boolean);
procedure SetGraphMode(Mode: TGraphMode);
procedure SetGraphEdit(enable: boolean);
procedure ClearOwnFilters;
procedure AddOwnFilters;
procedure GraphEvents(Event, Param1, Param2: integer);
procedure ControlEvents(Event: TControlEvent; Param: integer = 0);
procedure SetLogFile(FileName: String);
function GetState: TGraphState;
function GetVolume: integer;
procedure SetVolume(Volume: Integer);
function GetBalance: integer;
procedure SetBalance(Balance: integer);
function GetSeekCaps: TSeekingCaps;
procedure SetRate(Rate: double);
function GetRate: double;
function GetDuration: integer;
protected
procedure DoEvent(Event, Param1, Param2: Integer); virtual;
procedure InsertFilter(AFilter: IFilter);
procedure RemoveFilter(AFilter: IFilter);
procedure InsertEventNotifier(AEvent: IEvent);
procedure RemoveEventNotifier(AEvent: IEvent);
public //зона бачення змінних та типів даних
property Duration: Integer read GetDuration; // завдання властивості типу
property Rate: Double read GetRate write SetRate;
property SeekCapabilities: TSeekingCaps read GetSeekCaps;
property Balance: integer read GetBalance write SetBalance;
property Volume: integer read GetVolume write SetVolume;
property State: TGraphState read GetState;
constructor Create(AOwner: TComponent); override; //Конструктор
destructor Destroy; override; //Деструктор
procedure Loaded; override;
function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
function Play: boolean;
function Pause: boolean;
function Stop: boolean;
procedure DisconnectFilters;
procedure ClearGraph;
function RenderFile(FileName: WideString): HRESULT;

```

```

function RenderFileEx(FileName: WideString): HRESULT;
function RenderCAM(out status: TAMCAMRenderStatus;
FileName: WideString = ''; Mode: Integer = AM_CAM_HWDEC_PREFER): HRESULT;

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

// -----
// Події
// -----

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

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

property OnGraphDeviceLost: TOnGraphDeviceLost read FOnGraphDeviceLost write
FOnGraphDeviceLost;

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

property OnGraphErrorAbort: TOnDSResult read FOnGraphErrorAbort write
FOnGraphErrorAbort;

property OnGraphFullscreenLost: TOnGraphFullscreenLost read
FOnGraphFullscreenLost write FOnGraphFullscreenLost;

property OnGraphChanged: TNotifyEvent read FOnGraphChanged write
FOnGraphChanged;

property OnGraphOleEvent: TOnGraphOleEvent read FOnGraphOleEvent write
FOnGraphOleEvent;

property OnGraphOpeningFile: TOnGraphOpeningFile read FOnGraphOpeningFile
write FOnGraphOpeningFile;

property OnGraphPaletteChanged: TNotifyEvent read FOnGraphPaletteChanged
write FOnGraphPaletteChanged;

```

```

property OnGraphPaused: TOnDSResult read FOnGraphPaused write
FOnGraphPaused;

property OnGraphQualityChange: TNotifyEvent read FOnGraphQualityChange write
FOnGraphQualityChange;

property OnGraphSNDDDevInError: TOnGraphSNDDDevError read
FOnGraphSNDDDevInError write FOnGraphSNDDDevInError;

property OnGraphSNDDDevOutError: TOnGraphSNDDDevError read
FOnGraphSNDDDevOutError write FOnGraphSNDDDevOutError;

property OnGraphStepComplete: TNotifyEvent read FOnGraphStepComplete write
FOnGraphStepComplete;

property OnGraphStreamControlStarted: TOnGraphStreamControl read
FOnGraphStreamControlStarted write FOnGraphStreamControlStarted;

property OnGraphStreamControlStopped: TOnGraphStreamControl read
FOnGraphStreamControlStopped write FOnGraphStreamControlStopped;

property OnGraphStreamErrorStillPlaying : TOnGraphStreamError read
FOnGraphStreamErrorStillPlaying write FOnGraphStreamErrorStillPlaying;

property OnGraphStreamErrorStopped: TOnGraphStreamError read
FOnGraphStreamErrorStopped write FOnGraphStreamErrorStopped;

property OnGraphUserAbort: TNotifyEvent read FOnGraphUserAbort write
FOnGraphUserAbort;

property OnGraphVideoSizeChanged: TOnGraphVideoSizeChanged read
FOnGraphVideoSizeChanged write FOnGraphVideoSizeChanged;

property OnGraphTimeCodeAvailable: TOnGraphTimeCodeAvailable read
FOnGraphTimeCodeAvailable write FOnGraphTimeCodeAvailable;

property OnGraphEXTDeviceModeChange: TOnGraphEXTDeviceModeChange read
FOnGraphEXTDeviceModeChange write FOnGraphEXTDeviceModeChange;

property OnGraphClockUnset: TNotifyEvent read FOnGraphClockUnset write
FOnGraphClockUnset;

property OnGraphVMRRenderDevice: TOnGraphVMRRenderDevice read
FOnGraphVMRRenderDevice write FOnGraphVMRRenderDevice;

property OnCAMDataStreamChange: TOnCAMDataStreamChange read
FOnCAMDataStreamChange write FOnCAMDataStreamChange;

property OnCAMCurrentTime: TOnCAMCurrentTime read FOnCAMCurrentTime write
FOnCAMCurrentTime;

property OnCAMTitleChange: TOnCAMTitleChange read FOnCAMTitleChange write
FOnCAMTitleChange;

property OnCAMChapterStart: TOnCAMChapterStart read FOnCAMChapterStart write
FOnCAMChapterStart;

property OnCAMAngleChange: TOnCAMChange read FOnCAMAngleChange write
FOnCAMAngleChange;

property OnCAMValidUOPSChange: TOnCAMValidUOPSChange read
FOnCAMValidUOPSChange write FOnCAMValidUOPSChange;

property OnCAMButtonChange: TOnCAMChange read FOnCAMButtonChange write
FOnCAMButtonChange;

property OnCAMChapterAutoStop: TNotifyEvent read FOnCAMChapterAutoStop write
FOnCAMChapterAutoStop;

property OnCAMStillOn: TOnCAMStillOn read FOnCAMStillOn write FOnCAMStillOn;

property OnCAMStilloff: TNotifyEvent read FOnCAMStilloff write
FOnCAMStilloff;

property OnCAMSubpictureStreamChange: TOnCAMSubpictureStreamChange read
FOnCAMSubpictureStreamChange write FOnCAMSubpictureStreamChange;

property OnCAMNoFP_PGC: TNotifyEvent read FOnCAMNoFP_PGC write
FOnCAMNoFP_PGC;

property OnCAMPlaybackRateChange: TOnCAMPlaybackRateChange read
FOnCAMPlaybackRateChange write FOnCAMPlaybackRateChange;

```

```

property OnCAMParentalLevelChange: TOnCAMParentalLevelChange read
FOnCAMParentalLevelChange write FOnCAMParentalLevelChange;

property OnCAMPlaybackStopped: TNotifyEvent read FOnCAMPlaybackStopped write
FOnCAMPlaybackStopped;

property OnCAMAnglesAvailable: TOnCAMAnglesAvailable read
FOnCAMAnglesAvailable write FOnCAMAnglesAvailable;

property OnCAMPlayPeriodAutoStop: TNotifyEvent read FOnCAMPlayPeriodAutoStop
write FOnCAMPlayPeriodAutoStop;

property OnCAMButtonAutoActivated: TOnCAMButtonAutoActivated read
FOnCAMButtonAutoActivated write FOnCAMButtonAutoActivated;

property OnCAMCMDStart: TOnCAMCMD read FOnCAMCMDStart Write FOnCAMCMDStart;
property OnCAMCMDEnd: TOnCAMCMD read FOnCAMCMDEnd Write FOnCAMCMDEnd;
property OnCAMDiscEjected: TNotifyEvent read FOnCAMDiscEjected Write
FOnCAMDiscEjected;
property OnCAMDiscInserted: TNotifyEvent read FOnCAMDiscInserted write
FOnCAMDiscInserted;

property OnCAMCurrentHMSFTime: TOnCAMCurrentHMSFTime read
FOnCAMCurrentHMSFTime write FOnCAMCurrentHMSFTime;

property OnCAMKaraokeMode: TOnCAMKaraokeMode read FOnCAMKaraokeMode write
FOnCAMKaraokeMode;

property OnCAMDomainFirstPlay: TNotifyEvent read FOnCAMDomainFirstPlay write
FOnCAMDomainFirstPlay;

property OnCAMDomainVideoManagerMenu: TNotifyEvent read
FOnCAMDomainVideoManagerMenu write FOnCAMDomainVideoManagerMenu;
property OnCAMDomainVideoTitleSetMenu: TNotifyEvent read
FOnCAMDomainVideoTitleSetMenu write FOnCAMDomainVideoTitleSetMenu;
property OnCAMDomainTitle: TNotifyEvent read FOnCAMDomainTitle write
FOnCAMDomainTitle;

property OnCAMDomainStop: TNotifyEvent read FOnCAMDomainStop write
FOnCAMDomainStop;

property OnCAMErrorUnexpected: TNotifyEvent read FOnCAMErrorUnexpected write
FOnCAMErrorUnexpected;

property OnCAMErrorInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMErrorInvalidCAM1_0Disc write FOnCAMErrorInvalidCAM1_0Disc;

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

property OnCAMErrorMacrovisionFail: TNotifyEvent read
FOnCAMErrorMacrovisionFail write FOnCAMErrorMacrovisionFail;
property OnCAMErrorIncompatibleSystemAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleSystemAndDecoderRegions write
FOnCAMErrorIncompatibleSystemAndDecoderRegions;
property OnCAMErrorIncompatibleDiscAndDecoderRegions: TNotifyEvent read
FOnCAMErrorIncompatibleDiscAndDecoderRegions write
FOnCAMErrorIncompatibleDiscAndDecoderRegions;

property OnCAMWarningInvalidCAM1_0Disc: TNotifyEvent read
FOnCAMWarningInvalidCAM1_0Disc write FOnCAMWarningInvalidCAM1_0Disc;
property OnCAMWarningFormatNotSupported : TNotifyEvent read
FOnCAMWarningFormatNotSupported write FOnCAMWarningFormatNotSupported;
property OnCAMWarningIllegalNavCommand : TNotifyEvent read
FOnCAMWarningIllegalNavCommand write FOnCAMWarningIllegalNavCommand;
property OnCAMWarningOpen: TNotifyEvent read FOnCAMWarningOpen write
FOnCAMWarningOpen;
property OnCAMWarningSeek: TNotifyEvent read FOnCAMWarningSeek write
FOnCAMWarningSeek;

```

```

    property OnCAMWarningRead: TNotifyEvent read FOnCAMWarningRead write
FOnCAMWarningRead;
    end;

// Структура TVMROptions

    TVideoWindow = class;
// Оголошення класу

    TVMRVideoMode = (
        vmrWindowed,
        vmrWindowless,
        vmrRenderless
    );

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

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

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

    TAbstractAllocatorClass = class of TAbstractAllocator; // Оголошення класу
    TVideoWindow = class(TCustomControl, IFilter, IEvent) // Оголошення класу
    private
        FMode          : TVideoMode;
        FVMOptions     : TVMROptions;

```

```

FBaseFilter      : IBaseFilter;
FVideoWindow    : IVideoWindow;
FWindowLess     : IVMRWindowlessControl9;
FFullScreen     : boolean;
FFilterGraph    : TFilterGraph;
FWindowStyle    : LongWord;
FWindowStyleEx  : 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 //зона бачення змінних та типів даних
  {@exclude}

```

```

function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;

constructor Create(AOwner: TComponent);override; //Конструктор
destructor Destroy; override; //Деструктор
class function CheckVMR: boolean;
function VMRGetBitmap(Stream: TStream): boolean;
function CheckInputPinsConnected: boolean;
procedure SetAllocator(Allocator: TAbstractAllocatorClass; UserID:
Cardinal);
published //зона бачення змінних
property OnPaint: TNotifyEvent read FOnPaint write FOnPaint;
// завдання властивості типу
property FullScreenTopMost: boolean read FTopMost write SetTopMost
    default false;
property Mode: TVideoMode read FMode write SetVideoMode default vmNormal;
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
property VideoHandle: THandle read GetVideoHandle;
property VMROptions: TVMROptions read FVMROptions write FVMROptions;
property FullScreen: boolean read FFullScreen write SetFullScreen
    default false;

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

TSampleGrabber = class; // Оголошення класу

TSampleGrabber = class(TComponent, IFilter,
    ISampleGrabberCB) // Оголошення класу
private
    FOnBuffer: TOnBuffer;

```

```

FBaseFilter: IBaseFilter;
FFilterGraph : TFilterGraph;
FMediaType: TMediaType;
BMPInfo : PBitmapInfo;
FCriticalSection: TCriticalSection;
function GetFilter: IBaseFilter;
function GetName: string;
procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
procedure SetFilterGraph(AFilterGraph: TFilterGraph);
function SampleCB(SampleTime: Double; pSample: IMediaSample): HRESULT;
    stdcall;
function BufferCB(SampleTime: Double; pBuffer: PByte; BufferLen: longint):
    HRESULT; stdcall;

protected
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
    public //зона бачення змінних та типів даних
        SampleGrabber: ISampleGrabber;
        InPutPin : IPin;
        OutPutPin : IPin;
        constructor Create(AOwner: TComponent); override; //Конструктор
        destructor Destroy; override; //Деструктор
        procedure UpdateMediaType;
        function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;

        procedure SetBMPCompatible(Source: PAMMediaType; SetDefault: cardinal);
        function GetBitmap(Bitmap: TBitmap; Buffer: Pointer; BufferLen: Integer):
boolean; overload;
        function GetBitmap(Bitmap: TBitmap): boolean; overload;
        class function CheckFilter: boolean;
published //зона бачення змінних
    property OnBuffer: TOnBuffer read FOnBuffer write FOnBuffer;
    property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
        // завдання властивості типу
    property MediaType: TMediaType read FMediaType write FMediaType;
end;
// структура TFilter
TFilter = class(TComponent, IFilter) // Оголошення класу
private
    FFilterGraph : TFilterGraph;
    FBaseFilter: TBaseFilter;
    FFilter: IBaseFilter;
    function GetFilter: IBaseFilter;
    function GetName: string;
    procedure NotifyFilter(operation: TFilterOperation; Param: integer = 0);
    procedure SetFilterGraph(AFilterGraph: TFilterGraph);
protected
    {@exclude}
    procedure Notification(AComponent: TComponent; Operation: TOperation);
override;
    public
//зона бачення змінних та типів даних

```

```

    { Конструктор }
    constructor Create(AOwner: TComponent); override;
    { Деструктор method. }
    destructor Destroy; override; //Деструктор
    function QueryInterface(const IID: TGUID; out Obj): HRESULT; override;
stdcall;
    published //зона бачення змінних та типів даних
        property BaseFilter: TBaseFilter read FBaseFilter write FBaseFilter;
        property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
    end;

// структура TASFWriter

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

```

```

// завдання властивості типу
property Port: DWORD read FPort write FPort;
property MaxUsers: DWORD read FMaxUsers write FMaxUsers;
property AutoIndex : boolean read FAutoIndex write FAutoIndex default
True;
property MultiPass : boolean read FMultiPass write FMultiPass default
False;
property DontCompress: boolean read FDontCompress write FDontCompress
default False;

end;

// структура TDSTackBar

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

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

```

```

//Деструктор
published //зона бачення змінних
    property FilterGraph: TFilterGraph read FFilterGraph Write SetFilterGraph;
    property TimerInterval: Cardinal read FInterval write
        SetInterval default 1000;
    property OnTimer: TTimerEvent read FOnTimer write SetOnTimer;
end;

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

// структура TColorControl

TColorControl = class(TPersistent) // Оголошення класу
private
    FBrightness : Integer;
    FContrast   : Integer;
    FHue       : Integer;
    FSaturation : Integer;
    FSharpness  : Integer;
    FGamma     : Integer;
    FUtilColor  : Boolean;
    FDefault   : TDDColorControl;
protected
    { Protected секція }
    { @exclude }
    FOwner : TDSVideoWindowEx2;
    { @exclude }
    Procedure SetBrightness(Value : Integer);
    { @exclude }
    Procedure SetContrast(Value : Integer);
    { @exclude }
    procedure SetHue(Value : Integer);
    { @exclude }
    procedure SetSaturation(Value : Integer);
    { @exclude }
    procedure SetSharpness(Value : Integer);
    { @exclude }
    procedure SetGamma(Value : Integer);
    { @exclude }
    procedure SetUtilColor(Value : Boolean);
    { @exclude }
    function GetBrightness : Integer;
    function GetContrast : Integer;
    function GetHue : Integer;
    function GetSaturation : Integer;
    function GetSharpness : Integer;
    function GetGamma : Integer;
    { @exclude }
    function GetUtilColor : Boolean;
    { @exclude }

```

```

Procedure ReadDefault;
{ @exclude }
procedure UpdateColorControls;
{ @exclude }
procedure GetColorControls;
public //зона бачення змінних та типів даних
{ Public секція }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
procedure RestoreDefault;
Published //зона бачення змінних
property Brightness : Integer read GetBrightness write SetBrightness;
// завдання властивості типу
property Contrast : Integer read GetContrast write SetContrast;
property Hue : Integer read GetHue write SetHue;

property Saturation : Integer read GetSaturation write SetSaturation;

property Sharpness : Integer read GetSharpness write SetSharpness;

property Gamma : Integer read GetGamma write SetGamma;

property ColorEnable : Boolean read GetUtilColor write SetUtilColor;
end;

// структура TDSVideoWindowEx2Caps
TDSVideoWindowEx2Caps = class(TPersistent) // Оголошення класу
protected
{ Protected секція }
Owner : TDSVideoWindowEx2;
function GetCanOverlay : Boolean;
function GetCanControlBrigttness : Boolean;
function GetCanControlContrast : Boolean;
function GetCanControlHue : Boolean;
function GetCanControlSaturation : Boolean;
function GetCanControlSharpness : Boolean;
function GetCanControlGamma : Boolean;
function GetCanControlUtilizedColor : Boolean;
public
{ Public секція }
{ @exclude }
constructor Create(AOwner: TDSVideoWindowEx2); virtual; //Конструктор
published //зона бачення змінних
Property CanOverlayGraphic : Boolean read GetCanOverlay;
Property CanControlBrigttness : Boolean read GetCanControlBrigttness;
Property CanControlContrast : Boolean read GetCanControlContrast;
Property CanControlHue : Boolean read GetCanControlHue;

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

```

```

end;

// структура TOverlayCallback

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

// структура TDSVideoWindowEx2

TRatioModes = (rmStretched, rmLetterBox, rmCrop);

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

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

TDSVideoWindowEx2 = class(TCustomControl, IFilter, IEvent) // Оголошення класу
private
    FVideoWindow      : IVideoWindow;
    FFilterGraph      : TFilterGraph;
    FBaseFilter        : IBaseFilter;
    FOverlayMixer     : IBaseFilter;
    FVideoRenderer    : IBaseFilter;
    FDDXM              : IDDrawExclModeVideo;
    FFullScreen        : Boolean;
    FTopMost           : Boolean;
    FColorKey          : TColor;
    FWindowStyle       : LongWord;
    FWindowStyleEx     : LongWord;
    FVideoRect         : TRect;
    FOnPaint           : TNotifyEvent;
    FOnColorKey        : TNotifyEvent;
    FOnCursorVisible  : TCursorVisibleEvent;
    FOnOverlay         : TOverlayVisibleEvent;
    FColorControl      : TColorControl;
    FCaps              : TDSVideoWindowEx2Caps;
    FZoom              : Integer;
    FAspectMode        : TRatioModes;
    FNoScreenSaver     : Boolean;
    FIdleCursor        : Integer;
    FMonitor           : TMonitor;
    FFullscreenControl : TForm;

```

```

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

```

```

procedure NormalPlayback;
procedure StartFullScreen; overload;
procedure StartFullScreen(OnMonitor : TMonitor); overload;
property FullScreen: boolean read FFullScreen;
property DesktopPlayback : Boolean Read FDesktopPlay;
property Canvas;
property ColorKey : TColor read FColorKey;
property Capabilities : TDSVideoWindowEx2Caps read FCaps;
property OverlayVisible : Boolean read FOverlayVisible;
published //зона бачення змінних
property AspectRatio : TRatioModes read FAspectRatio write SetAspectRatio;
property AutoHideCursor : Integer read FIdleCursor write FIdleCursor;
// завдання властивості типу
property DigitalZoom : Integer read FZoom write SetZoom;
property FilterGraph: TFilterGraph read FFilterGraph write SetFilterGraph;
property FullScreenTopMost: boolean read FTopMost write SetTopMost default
false;
property OnColorKeyChanged: TNotifyEvent read FOnColorKey write FOnColorKey;
property ColorControl: TColorControl read FColorControl write FColorControl;

property NoScreenSaver: Boolean read FNoScreenSaver write FNoScreenSaver;

property OnOverlayVisible: TOverlayVisibleEvent read FOnOverlay write
FOnOverlay;

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

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

```

```

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

TVMRBitmapOption=(
    vmrbDisable,
    vmrbSrcColorKey,
    vmrbSrcRect);

TVMRBitmapOptions = set of TVMRBitmapOption;

TVMRBitmap = class // Класи
private
    FVideoWindow: TVideoWindow;
    FCanvas: TCanvas;
    FVMR9ALPHABITMAP: TVMR9ALPHABITMAP;
    FOptions: TVMRBitmapOptions;
    FBMPold: HBITMAP;
    procedure SetOptions(Options: TVMRBitmapOptions);
    procedure ResetBitmap;
    procedure SetAlpha(const Value: Single);
    procedure SetColorKey(const Value: COLORREF);
    procedure SetDest(const Value: TVMR9NormalizedRect);
    procedure SetDestBottom(const Value: Single);
    procedure SetDestLeft(const Value: Single);
    procedure SetDestRight(const Value: Single);
    procedure SetDestTop(const Value: Single);
    procedure SetSource(const Value: TRect);
    function GetAlpha: Single;
    function GetColorKey: COLORREF;
    function GetDest: TVMR9NormalizedRect;
    function GetDestBottom: Single;
    function GetDestLeft: Single;
    function GetDestRight: Single;
    function GetDestTop: Single;
    function GetSource: TRect;
public //зона бачення змінних та типів даних
    constructor Create(VideoWindow: TVideoWindow); //Конструктор
    destructor Destroy; override; //Деструктор
    procedure LoadBitmap(Bitmap: TBitmap);
    procedure LoadEmptyBitmap(Width, Height: Integer; PixelFormat: TPixelFormat;
        Color: TColor);
    procedure Draw;
    procedure DrawTo(Left, Top, Right, Bottom, Alpha: Single; doUpdate: boolean
        = false);
    procedure Update;
    property Canvas: TCanvas read FCanvas write FCanvas;
    property Alpha: Single read GetAlpha write SetAlpha;
    property Source: TRect read GetSource write SetSource;
    property DestLeft : Single read GetDestLeft write SetDestLeft;
    property DestTop : Single read GetDestTop write SetDestTop;
    property DestRight : Single read GetDestRight write SetDestRight;
    property DestBottom : Single read GetDestBottom write SetDestBottom;

```

```
property Dest: TVMR9NormalizedRect read GetDest write SetDest;  
// завдання властивості типу COLORREF  
property ColorKey: COLORREF read GetColorKey write SetColorKey;  
property Options: TVMRBitmapOptions read FOptions write SetOptions;  
end;  
End.
```

Кафедра КБПЗ – 2021 рік

ФАЙЛ ГОЛОВНОГО ВІКНА

```

unit Video_RTSP; // Назва модулю
{
Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет механіко-технологічний
Кафедра кібербезпеки та програмного забезпечення
Виконав: студент 6 курсу, Сергій Владислав Віталійович
Керівник: Петренко В.І.
}

Interface
{Інтерфейсна частина, тут проходить опис класів та типів}

uses Classes, SysUtils; // Підключення бібліотек

type

TVideo_RTSPType = class(TComponent) // Об'ява класу
private // секція бачення
    FValues: TStringList;
    FVideo_RTSPCap: string;
    FUserAgent: string;
    procedure ReadUserAgentInfo;
    procedure SetVideo_RTSPCap(const Value: string);
    procedure SetUserAgent(const Value: string);
    function GetCount: integer;
    function GetKey(Index: integer): string;
    function GetValue(const Index: variant): variant;
    function GetVideo_RTSP: string;
    function GetVersion: string;
    function GetMajorVer: integer;
    function GetMinorVer: integer;
    function GetFrames: boolean;
    function GetTables: boolean;
    function GetCookies: boolean;
    function GetBackgroundSounds: boolean;
    function GetVBScript: boolean;
    function GetJavaScript: boolean;
    function GetJavaApplets: boolean;
    function GetPlatform: string;
    function GetActiveXControls: boolean;
    function GetBeta: boolean;
public // секція бачення
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override; // Деструктор
    property Video_RTSPCap: string read FVideo_RTSPCap write
SetVideo_RTSPCap;
    property UserAgent: string read FUserAgent write SetUserAgent;
    property Video_RTSP: string read GetVideo_RTSP;
    property Version: string read GetVersion;
    property MajorVer: integer read GetMajorVer;
    property MinorVer: integer read GetMinorVer;

```

```

property      Frames: boolean read GetFrames;
property      Tables: boolean read GetTables;
property      Cookies: boolean read GetCookies;
property      BackgroundSounds: boolean read GetBackgroundSounds;
property      JavaScript: boolean read GetJavaScript;
property      JavaApplets: boolean read GetJavaApplets;
property      Platform: string read GetPlatform;
property      ActiveXControls: boolean read GetActiveXControls;
property      Beta: boolean read GetBeta;
property      Count: integer read GetCount;
property      Keys[Index: integer]: string read GetKey;
property      Values[const Index: variant]: variant read GetValue; default;
end;

procedure Register;

implementation
{Секція де проходить реалізація того що описано в інтерфейсній частині}

uses Windows, IniFiles; // Підключення бібліотек

constructor TVideo_RTSPType.Create(AOwner: TComponent); //Конструктор
begin
    inherited create(AOwner);
    FValues := TStringList.create;
    SetVideo_RTSPCap('BROWSCAP.INI');
end;

destructor TVideo_RTSPType.Destroy; // Деструктор
begin
    FValues.free;
    inherited destroy;
end;

function IsWindowsNT : Boolean; // Реалізація функції
begin
    result := (Win32Platform = VER_PLATFORM_WIN32_NT);
end;

procedure RaiseWin32Error; // Реалізація процедури
begin
    raise exception.create(SysErrorMessage(GetLastError));
end;

var IniLock: TRTLCriticalSection; // Об'ява змінних

procedure TVideo_RTSPType.ReadUserAgentInfo;
// Реалізація процедури
var // Об'ява змінних
    ini: TIniFile;

```

```

function matchWithWildcard(const s: string): boolean; // Реалізація функції
var // Об'ява змінних
    p: integer;
    rightPartLen: integer;
begin
    result := false;
    p := pos('*', s);
    // отримуємо строку
    if (p = 0) then
        exit;
    if (copy(s, 1, p - 1) <> copy(FUserAgent, 1, p - 1)) then
        exit;
    rightPartLen := length(s) - p;
    if (copy(s, p + 1, rightPartLen) <> copy(FUserAgent, length(FUserAgent)
- rightPartLen + 1, rightPartLen)) then
        exit;
    result := true;
end;

function determineSection: string; // Реалізація функції
const
    defaultVideo_RTSPSection = 'Default Video_RTSP Capability Settings';
var // Об'ява змінних
    sections: TStringList;
    i: integer;
begin
    sections := TStringList.create;
    try
        ini.readSections(sections);
        // спроба отримання даних
        if (sections.indexOf(FUserAgent) <> -1) then begin
            result := FUserAgent;
            exit;
        end;
        // пошук
        for i := 0 to (sections.count - 1) do begin
            if matchWithWildcard(sections[i]) then begin
                result := sections[i];
                exit;
            end;
        end;
        // нічого не знайдено, встановлення значень по замовчанню
        if (sections.indexOf(defaultVideo_RTSPSection) <> -1) then begin
            result := defaultVideo_RTSPSection;
            exit;
        end;
        // похибка
        result := '';
    finally
        sections.free;
    end;
end;

```

```

end;

procedure readSection(const section: string); // Реалізація процедури
var // Об'ява змінних
    parentSection: string;
    sectionValues: TStringList;
    i: integer;
    key: string;
begin
    parentSection := ini.readString(section, 'parent', '');
    if ((parentSection <> '') and (parentSection <> section)) then
        readSection(parentSection);
    sectionValues := TStringList.create;
    try
        ini.readSectionValues(section, sectionValues);
        for i := 0 to (sectionValues.count - 1) do begin
            key := sectionValues.names[i];
            FValues.values[key] := sectionValues.values[key];
        end;
    finally
        sectionValues.free;
    end;
end;

var // Об'ява змінних
    section: string;
begin
    FValues.clear;
    if (FUserAgent = '') then
        exit;

    ini := TIniFile.create(FVideo_RTSPCap);
    try
        EnterCriticalSection(IniLock);
        try
            section := determineSection;
            if (section <> '') then
                readSection(section);
        finally
            LeaveCriticalSection(IniLock);
        end;
    finally
        ini.free;
    end;
end;

procedure TVideo_RTSPType.SetVideo_RTSPCap(const Value: string);
// Реалізація процедури
begin
    if (value <> FVideo_RTSPCap) then begin
        FVideo_RTSPCap := value;
    end;
end;

```

```

        if (FUserAgent <> '') then
            ReadUserAgentInfo;
    end;
end;

procedure TVideo_RTSPType.SetUserAgent(const Value: string);
// Реалізація процедури
begin
    if (value <> FUserAgent) then begin
        FUserAgent := value;
        ReadUserAgentInfo;
    end;
end;

function TVideo_RTSPType.GetCount: integer; // Реалізація функції
begin
    result := FValues.count;
end;

type TVarKind = (vkStr, vkInt);

function getVarKind(const v: variant): TVarKind; // Реалізація функції
const
    varIntKindMask = varSmallint or varInteger or varByte;
    varStrKindMask = varOleStr or varString;
begin
    if ((varType(v) and varTypeMask and varIntKindMask) <> 0) then
        result := vkInt
    else if ((varType(v) and varTypeMask and varStrKindMask) <> 0) then
        result := vkStr
    else
        raise exception.create('Expecting an integer or a string');
end;

function browsCapStrToVar(const s: string): variant; // Реалізація функції
begin
    if (s = '') then
        result := ''
    else if (s[1] = '#') then begin
        try
            result := strToInt(copy(s, 2, maxInt));
        except
            on EConvertError do
                result := s;
        end;
    end else if (compareText(s, 'true') = 0) then
        result := true
    else if (compareText(s, 'false') = 0) then
        result := false
    else
        result := s;
end;

```

```

end;

function TVideo_RTSPType.GetValue(const Index: variant): variant;
// Реалізація функції
var // Об'ява змінних
    key: string;
begin
    case getVarKind(index) of
        vkInt: key := getKey(index);
        vkStr: key := string(index);
    end;
    if (FValues.IndexOfName(key) = -1) then
        result := null
    else
        result := browsCapStrToVar(FValues.values[key]);
    end;

function TVideo_RTSPType.GetKey(Index: integer): string; // Реалізація функції
begin
    result := FValues.Names[Index];
end;

function TVideo_RTSPType.GetVideo_RTSP: string; // Реалізація функції
begin
    result := VarToStr(values['Video_RTSP']);
end;

function TVideo_RTSPType.GetVersion: string; // Реалізація функції
begin
    result := VarToStr(values['Version']);
end;

function SafeVarToInt(const v: variant): integer; // Реалізація функції
begin
    if VarIsNull(v) then
        result := 0
    else
        result := integer(v);
    end;

function SafeVarToBoolean(const v: variant): boolean; // Реалізація функції
begin
    if VarIsNull(v) then
        result := false
    else
        result := boolean(v);
    end;

function TVideo_RTSPType.GetMajorVer: integer; // Реалізація функції
begin
    result := SafeVarToInt(values['MajorVer']);

```

```
end;

function    TVideo_RTSPType.GetMinorVer: integer; // Реалізація функції
begin
    result := SafeVarToInt(values['MinorVer']);
end;

function    TVideo_RTSPType.GetFrames: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['Frames']);
end;

function    TVideo_RTSPType.GetTables: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['Tables']);
end;

function    TVideo_RTSPType.GetCookies: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['Cookies']);
end;

function    TVideo_RTSPType.GetBackgroundSounds: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['BackgroundSounds']);
end;

function    TVideo_RTSPType.GetVBScript: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['VBScript']);
end;

function    TVideo_RTSPType.GetJavaScript: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['JavaScript']);
end;

function    TVideo_RTSPType.GetJavaApplets: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['JavaApplets']);
end;

function    TVideo_RTSPType.GetPlatform: string; // Реалізація функції
begin
    result := VarToStr(values['Platform']);
end;

function    TVideo_RTSPType.GetActiveXControls: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['ActiveXControls']);
end;
```

```
function    TVideo_RTSPType.GetBeta: boolean; // Реалізація функції
begin
    result := SafeVarToBoolean(values['Beta']);
end;

initialization
begin
    InitializeCriticalSection(IniLock);
end;

finalization
begin
    DeleteCriticalSection(IniLock);
end;

end.
```

Кафедра_КБПЗ_2021_рік