

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи передачі
відеоданих з використанням технології CUDA”

КБПЗ - 2023

Виконав здобувач вищої освіти
II курсу, групи КН-22МЗ
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Корнієнко О.С.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

Галузь знань 12 *“Інформаційні технології”*

Спеціальність 122 *“Комп’ютерні науки”*

Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерні науки”*

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Корнієнку Олександр Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA*

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

затверджені наказом вищого навчального закладу № 37-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA*

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

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

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

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 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	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Корнієнко О.С. Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі відеоданих з використанням технології CUDA.

Метою розробки є дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA.

Об'єктом дослідження є процес передачі відеоданих з використанням технології CUDA.

Предметом дослідження є методи передачі відеоданих з використанням технології CUDA.

Методи дослідження базуються на методах комп'ютерної графіки, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи передачі відеоданих з використанням технології CUDA.

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 10/11.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерні науки, CUDA

ABSTRACT

Korniienko O.S. Research and software implementation of video data transmission system using CUDA technology. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the video data transmission system using CUDA technology.

The goal of development is research and software implementation of a video data transmission system using CUDA technology.

The object of research is the process of video data transfer using CUDA technology.

The subject of the study is video data transfer methods using CUDA technology.

Research methods are based on computer graphics methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the video data transmission system using CUDA technology.

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

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

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Visual C# environment.

Keywords: computer science, CUDA

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.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	12
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	12
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	26
2.3 Розгорнута постановка завдання	29
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	31
3.1 Опис функціонування системи	31
3.2 Розробка структурної схеми.....	43
3.3 Розробка функціональної схеми	47
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.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	69
6 НАУКОВА НОВИЗНА	71

					ВКРМ-122.23.0068.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Корніснюк О.С.</i>					М	1	112
<i>Перев.</i>	<i>Смірнов О.А.</i>					<i>ЦНТУ КН-22МЗ</i>		
Н.контр.	<i>Коваленко А.С.</i>							
Затв.	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	73
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	73
7.2 Розрахунок трудомісткості розробки програмної продукції.....	75
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	77
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	82
7.5 Визначення собівартості розробки та ціни програмної продукції.....	85
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	88
7.7 Визначення експлуатаційних витрат.....	88
7.8 Визначення економічної ефективності програмної продукції.....	90
7.9 Висновок.....	92
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	93
8.1 Вступ.....	93
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	95
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	96
8.4 Розробка заходів з умов поліпшення охорони праці.....	99
8.5 Розрахункова частина	100
9 ОСНОВНІ ВИСНОВКИ.....	104
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	106

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

КБПЗ_2023

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Система відеотрансляції дозволяє одержувати й передавати множині користувачів дані від декількох відеокамер, розташованих на місці, де відбувається подія, наприклад, на стадіоні або в концертному залі.

У той же час застосування такої системи у випадку множини місць одночасного здійснення відеоспостереження, розташованих на значній відстані друг від друга, наприклад, від одного кілометра до декількох тисяч кілометрів, може бути неможливим або зв'язаним зі значними матеріальними або фінансовими витратами, оскільки якщо такі події відбуваються в різних містах або різних районах одного міста, то для передачі даних відеозображення в запам'ятовувальній пристрій буде потрібно виділений канал передачі даних, що володіє великою пропускнуою здатністю, у зв'язку з тим, що дані відеозображення як у незжатому, так і стислому виді мають великий обсяг. Підготовка каналу передачі даних з високою пропускнуою здатністю для передачі великих обсягів даних вимагає установку або оренду високопродуктивного встаткування, що спричинить значні фінансові витрати.

Іншим фактором, що обмежує застосування системи відеотрансляції, в умовах множини місць одночасного здійснення відеоспостереження є те, що на сервер лягає значне навантаження по прийому й обробці відеоданих із всіх відеокамер. Крім того, коли до перегляду відеозапису з однієї й тої ж відеокамери послідовно звертаються безліч користувачів, на сервер у системі відеотрансляції лягає збільшене навантаження, оскільки відеозапис, збережений у запам'ятовувальній пристрої для архівування відеозаписів, для передачі із сервера в мобільні пристрої, веб-браузери або контролери телевізійних пристроїв користувачів необхідно передавати кілька разів (по кількості запитів від користувачів).

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем передачі відеоданих з використанням технології CUDA.
- Дослідження системи передачі відеоданих з використанням технології CUDA.
- Програмна реалізація системи передачі відеоданих з використанням технології CUDA.

Об'єктом дослідження є процес передачі відеоданих з використанням технології CUDA.

Предметом дослідження є методи передачі відеоданих з використанням технології CUDA.

Методи дослідження базуються на методах комп'ютерної графіки, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод передачі відеоданих з використанням технології CUDA.
- Розроблено вітчизняний продукт передачі відеоданих з використанням технології CUDA, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі передачі відеоданих з використанням технології CUDA.

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.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Реалізується система передачі відеоданих з використанням технології CUDA, що містить:

- пристрій збору відеоданих;
- колекторний запам'ятовувальний пристрій;
- пристрій трансляції відеоданих;
- безліч пристроїв кешування;
- пристрій балансування навантаження пристроїв кешування.

Пристрій збору відеоданих може одержувати з підсистеми передачі дані відеодані з множини територіально розподілених відеокамер і далі зберігати їх у колекторному запам'ятовувальному пристрої. Пристрій трансляції відеоданих одержує із пристроєм збору відеоданих і/або колекторного запам'ятовувального пристрою відеодані й передає їх у пристрій кешування у вигляді фрагментів заданої тривалості, а пристроєм кешування з'єднані із системою передачі даних користувачеві й передають через неї відеодані користувачам. Пристрій балансування навантаження пристроїв кешування направляє в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу відеоданих у заданий пристрій кешування з метою вирівнюючи навантаження на пристрої кешування.

1.2 Область застосування

Система обробки даних і системи відеотрансляції для забезпечення можливості спостереження за множиною подій, що відбуваються одночасно, місця проведення яких мають широкий територіальний розподіл, тобто вони

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

можуть відбуватися на значних відстанях друг від друга, від декількох кілометрів до тисяч і десятків тисяч кілометрів і більше.

Іншим завданням системи передачі відеоданих з використанням технології CUDA, є зниження навантаження на підсистему передачі даних (зниження обсягів переданих даних) при одночасному зниженні ризику втрати відеоданих.

Ще одним завданням системи передачі відеоданих з використанням технології CUDA, є подання системи обробки даних, що забезпечує надійну трансляцію відеоданих множині користувачів.

Завдання системи передачі відеоданих з використанням технології CUDA, вирішуються за допомогою системи збору й трансляції відеоданих, що містить: щонайменше, один пристрій збору відеоданих; щонайменше, один колекторний запам'ятовувальний пристрій; щонайменше, один пристрій трансляції відеоданих; безліч пристроїв кешування; щонайменше, один пристрій балансування навантаження пристроїв кешування.

Відповідно до системи передачі відеоданих з використанням технології CUDA, пристрій збору відеоданих виконано з можливістю з'єднання з підсистемою передачі даних і одержання через неї відеоданих з множині територіально розподілених відеокамер, а також з можливістю передачі, щонайменше, частині одержуваних відеоданих у колекторний запам'ятовувальний пристрій. Колекторний запам'ятовувальний пристрій виконаний з можливістю збереження, щонайменше, частині відеоданих, одержуваних із пристрою збору відеоданих, а пристрій трансляції відеоданих виконано з можливістю одержання із пристрою збору відеоданих і/або колекторного запам'ятовувального пристрою відеоданих і передачі в пристрій кешування, щонайменше, частині відеоданих у вигляді фрагментів заданої тривалості.

Пристрої кешування виконані з можливістю з'єднання із системою передачі даних користувачеві й з можливістю передачі, щонайменше, частині відеоданих користувачам через систему передачі даних, а пристрій балансування

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

навантаження пристроїв кешування виконано з можливістю передачі в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу, щонайменше, частині відеоданих у заданий пристрій кешування.

У деяких варіантах здійснення пристрій збору відеоданих і/або пристрій трансляції відеоданих виконані з можливістю поділу, щонайменше, частині відеоданих на фрагменти заданої тривалості. Крім того, пристрій збору відеоданих і/або пристрій трансляції відеоданих переважно виконані з можливістю стиску й/або кодування й/або перетворення формату й/або виділення, щонайменше, частині відеоданих.

Завдання системи передачі відеоданих з використанням технології CUDA, також вирішуються за допомогою системи відеотрансляції, що містить: безліч відеокамер; безліч локальних запам'ятовувальних пристроїв; підсистему передачі даних; щонайменше, один пристрій збору відеоданих; щонайменше, один колекторний запам'ятовувальний пристрій; щонайменше, один пристрій трансляції відеоданих; безліч пристроїв кешування; щонайменше, один пристрій балансування навантаження пристроїв кешування; щонайменше, один інтерфейсний пристрій.

Відповідно, локальні запам'ятовувальні пристрої виконані з можливістю збереження, щонайменше, частині відеоданих з відеокамер, а підсистема передачі даних виконана з можливістю одержання з відеокамер і/або локальних запам'ятовувальних пристроїв і передачі в пристрій збору відеоданих, щонайменше, частині відеоданих.

Передбачається, що запам'ятовувальний пристрій розташований поблизу від пристрої відеоспостереження або в самому пристрої відеоспостереження й передає відеодані в сервер через підсистему передачі даних, наприклад, Інтернет. Якщо ж запам'ятовувальний пристрій було б розташоване по близькості від сервера або перебувало в складі сервера, то навантаження на підсистему передачі даних були б знижені, однак у цьому випадку зростала б погроза втрати відеоданих з відеокамер і пристрої відеоспостереження, оскільки в цьому випадку

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

до того, як відеодані будуть збережені, їх необхідно передавати через підсистему передачі даних, у якій або в інтерфейсних пристроях якої можливі збої або недостатні швидкості передачі даних.

Крім того, пристрій збору відеоданих виконано з можливістю одержання з підсистеми передачі даних і передачі в колекторний запам'ятовувальний пристрій, щонайменше, частині відеоданих з множині відеокамер. Колекторний запам'ятовувальний пристрій виконаний з можливістю збереження, щонайменше, частині відеоданих, одержуваних із пристроєм збору відеоданих, а інтерфейсний пристрій виконаний із забезпеченням можливості вказівки, щонайменше, одним користувачем відеокамери, відеодані з якої необхідні для перегляду, і/або відеоданих, необхідних для перегляду.

Далі, пристрій трансляції відеоданих виконано з можливістю одержання із пристроєм збору відеоданих і/або колекторного запам'ятовувального пристроєм відеоданих і передачі в пристрій кешування, щонайменше, частині відеоданих, відповідно до вказівки користувача, у вигляді фрагментів заданої тривалості, пристроєм кешування виконані з можливістю з'єднання із системою передачі даних користувачеві й з можливістю передачі, щонайменше, частині відеоданих користувачам через систему передачі даних, а пристрій балансування навантаження пристроїв кешування виконане з можливістю передачі в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу, щонайменше, частині відеоданих у заданий пристрій кешування. У переважному варіанті система відеотрансляції також містить пристрої відеоспостереження, виконані з можливістю одержання відеоданих з відеокамер і передачі, щонайменше, частині відеоданих у локальні запам'ятовувальні пристрої, а також одержання відеоданих з відеокамер і/або локальних запам'ятовувальних пристроїв і передачі, щонайменше, частині відеоданих у підсистему передачі даних. Пристрої відеоспостереження й/або пристрій збору відеоданих і/або пристрій трансляції відеоданих можуть бути виконані з можливістю поділу, щонайменше, частині відеоданих на фрагменти заданої тривалості. У кращому

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

варіанті здійснення пристрій відеоспостереження й/або пристрій збору відеоданих і/або пристрій трансляції відеоданих виконані з можливістю стиску й/або кодування й/або перетворення формату й/або виділення, щонайменше, частині відеоданих.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2023

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

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

Apache OpenMeetings

У цей час для організації відеоконференцій існує велика кількість рішень, що відрізняються по призначенню (внутрішнє, зовнішнє використання, віщання), типу (персональні, групові, кімнатні), виду (точка – точка й багатоточечні), реалізації (апаратним і програмні), використуванним протоколам, вартості й так далі. Ринкова ціна комерційних і особливо апаратних рішень досить висока, однак застосування вільного ПЗ допоможе знизити витрати.

Система веб-конференцій Apache OpenMeetings дозволяє організувати проведення аудіо- і відеонарад у багатоточечному режимі, коли до сервера підключені десятки людей. За кілька років проект перемінив кілька команд і ліцензій, у тому числі був в Google Code (під ліцензією Eclipse Public License). Остання його дислокація – інкубатор Apache, відповідно, помінялася й ліцензія, на Apache License 2.0. Остання офіційна версія 2.0 вийшла наприкінці липня 2012 року.

Головний плюс – для відеонаради не потрібна установка додаткового ПЗ, досить веб-браузера із плагіном для підтримки технології Flash. Передбачено можливість запису й наступного програвання нарад і експорту в AVI/FLV-файл, імпорт у конференцію документів більш ніж 20 форматів і зображень. Учасники можуть скачати завантажений файл і спільно редагувати, уводячи текст поверх оригіналу, малювати й позначати цікаві місця. Самі конференції можуть бути відкритими й частками.

Підтримується два режими:

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- нарада – до 16 учасників, кожен може передавати аудіо- і відеодані;
- лекції – до 200 учасників, передача аудіо й відео тільки в модератора/лектора.

Передбачений також обмін текстовими повідомленнями у вікні чату або приватними (використовується убудований Jabber-сервіс). Налаштування дозволяють створити опитування. Модератор, що організує конференцію, відправляє всім учасникам запрошення, що містить пряме посилання, воно ж управляє всіма доступними їм можливостями. У кожного зареєстрованого користувача є календар подій з нагадуванням про події (через електронну пошту або iCal). При підключенні вибирається варіант участі (відео + аудіо, тільки відео або аудіо, рисунки), дозвіл і пристрій. Налаштування в кімнаті прості й зрозумілі кожному, користувач, що вперше працює із сервісом, швидко освоїться. Можлива інтеграція OpenMeetings з іншими продуктами – сервером VoIP Asterisk, системою керування навчанням Moodle, Drupal, Joomla, SugarCRM і деякими іншими.

Реалізовано три рівні доступу – користувач, модератор і адміністратор сервера. Для автентифікації можливе використання внутрішньої бази або сервісу LDAP / Active Directory (в \$RED5_HOME/webapps/openmeetings/conf знайдеш готові шаблони для підключення). Можлива робота декількох серверів OpenMeetings у кластері, одна установка може обслуговувати кілька організацій.

Інтерфейс OpenMeetings переведений на кілька мов, серед яких є російська. Убудований редактор локалізованих повідомлень (LanguageEditor) дозволяє при необхідності скорегувати переклад. Зовнішній вигляд можна змінити за допомогою тим.

Побудований OpenMeetings з використанням технологій Java і XML. Для організації сервера задіюються: веб-сервер Apache Tomcat, Open Source Flash/RTMP Server Red5, OpenOffice.org/LibreOffice. Як база даних може бути використана MySQL, PostgreSQL, Oracle, DB2 або Apache Derby. Пропонується демосайт, на якому можна познайомитися з основними можливостями

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Вимоги до встаткування невисокі, мінімальні, які зазначені на сайті, – комп'ютер із процесором 1 ГГц CPU і 1 Гб ОЗУ. Але для конвертування документів, завантаження файлів і запису відео цієї потужності не вистачить. У якості рекомендованого зазначений комп'ютер 2x/4x 2 ГГц (32/64 біт) і 4 Гб ОЗУ. Для організації 100 з'єднань досить комп'ютера класу Pentium 4 з 2 Гб ОЗУ.

BIGBLUEBUTTON

Перша версія BigBlueButton була написана в 2007 році одним зі співробітників Карлтонського університету м. Оттава, Канада (Carleton University), за підтримкою програми розвитку інноваційних технологій і керування. Споконвічно продукт мав ім'я Blindside, але пізніше назва була змінена на BigBlueButton, щоб відбити простоту у використанні – для початку конференції потрібно всього лише натиснути символічну синю кнопку. Саме в простоті перевага BBB перед більш функціональним і оснащеним, а виходить, і ледве більше складним OpenMeetings. Проект якийсь час шукав своє місце й сьогодні орієнтований на організації, що пропонують послуги дистанційного утворення, дозволяючи проводити навчання через інтернет. Особлива роль у цьому процесі приділяється однієї з функцій – відеоконференції. Але BBB з таким же успіхом може бути використаний для простого спілкування, проведення брифінгів і вебінарів. В 2009 році була організована компанія Blindside Networks для надання платної підтримки користувачам продукту. Найбільш серйозним поштовхом до розробки продукту послужила участь в Google Summer of Code в 2010 році. Саме тоді був доданий API, що дозволяє підключати сторонні додатки, і сьогодні вмонтувати BBB можна в Sakai, WordPress, Moodle, Joomla, Redmine, Drupal, Matterhorn, LAMS і деякі інші. Ця можливість найбільше затребувана користувачами BBB, тому з налаштувань сервера був прибраний інтерфейс адміністратора: розроблювачі просто не бачать змісту його розвивати, тому що керування лягає на плечі того, хто вбудовує додаток. У випадку окремого сервера всі установки можна без проблем зробити за допомогою конфігураційних файлів BigBlueButton і можливостей веб-сервера. Проект перебуває в стадії активної

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

для практичного показу роботи з додатками й ОС, завантаження презентації у форматі PDF (і будь-якому іншому, підтримуваному OpenOffice.org/LibreOffice), функції малювання й віртуальну вказівку. Реалізовано автоматичний переклад при спілкуванні в чаті користувачів на різних мовах. Для підключення до сервера досить використовувати веб-браузер з підтримкою Adobe Flash, тобто це може бути будь-який комп'ютер, що працює під керуванням Windows, *nix або Mac OS X. Ведеться розробка клієнта для Android. Конференції можуть бути двох видів: відкриті (може підключитися будь-який зареєстрований користувач) і закриті. У випадку приватної конференції список допущених формує сам виступаючий, висилаючи їм дані для доступу. Користувачі в конференції можуть бути в ролі виступаючого, модератора (за замовчуванням одержує творець конференції) і слухача. Робота віртуального лектора мало відрізняється від реального: крім відео, він завантажує документи, використовуючи вказівку, акцентує увагу на важливих моментах, включає аудіо обраного слухача. Модератор може призначити будь-якого користувача виступаючої, тоді вся увага буде переключена на нього. Інтерфейс користувача дозволяє наблизити окремі фрагменти, щоб краще розглянути їх, привернути увагу, «піднявши руку», спілкуватися в груповому або приватному чаті. Модератор повністю контролює можливості присутніх, при необхідності відключає користувача або переводить у режим «тільки перегляд». Підтримується дозвіл 320 x 240, 640 x 480, 1280 x 720, на кількість підключень BBB яких-небудь обмежень не накладає.

У своїй роботі BBB використовує більше десятка Open Source додатків: FreeSWITCH, nginx, Flash-медіасервер Red5, MySQL, ActiveMQ, Tomcat, Redis, Grails, Xuggler, OpenOffice.org, Image Magick, SWFTools і інші. Раніше в цьому списку був IP-PBX Asterisk з модулем для керування конференціями app_konference, але в останніх версіях розроблювачі відмовилися від даного зв'язування на користь FreeSWITCH, тому що останній не вимагає додаткових зусиль при реалізації функції запису. Веб-інтерфейс BBB переведений на 40 мов, у цьому списку є й росіянин.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Для роботи BigBlueButton рекомендується сервер з CPU Dual Core 2,6 ГГц, 2 Гб ОЗУ й місце на жорсткому диску з урахуванням запису трансляцій. Кількість користувачів, які зможуть одночасно спілкуватися на сервері, залежить від потужності встаткування й пропускної здатності каналу. Окремий потік вимагає 30-50 Кб/с. Приблизні розрахунки можна знайти в FAQ, там же наводяться дані стрес-тесту. Для підключення клієнтів за замовчуванням використовується стандартний 80-й порт, що не повинен бути зайнятий іншим додатком. У правилах брандмауера повинні бути відкриті порти 80 (HTTP), 935 (RTMP) і 9123 (загальний робочий стіл).

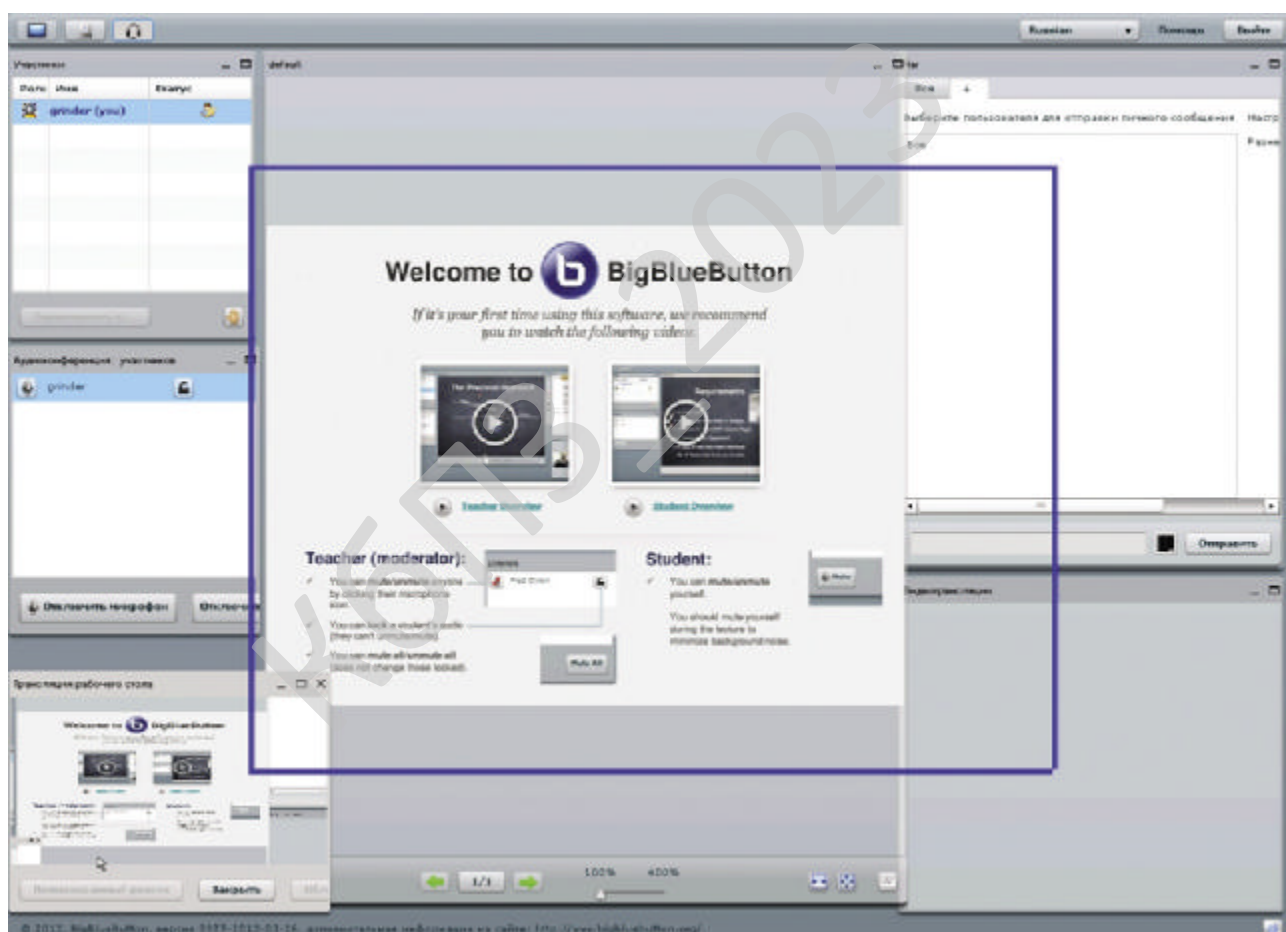


Рисунок 2.3 – Функція трансляції робочого стола в BigBlueButton

Доступний вихідний код, що дозволяє встановити BBV на будь-який комп'ютер, що працює під керуванням Linux, FreeBSD, Mac OS X або Windows.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Для Ubuntu і CentOS є готові пакети й репозиторії. Сервер ВВВ може працювати в хмарному середовищі начебто Amazon EC2.

Документація на сайті проекту досить докладна, у ній можна знайти відповіді практично на всі виникаючі питання – по установці (є готові конфігурації), конфігуруванню, API, локалізації, налаштуванню окремих компонентів (VoIP, nginx і подібних) і іншим моментам. Свої питання можна задати в списку розсилання, пропонується трохи відеокерівництв. Доступні образ VM і демосервер, що дозволяють познайомитися з основними можливостями ВВВ, не встановлюючи систему.

Практично половина всіх спроб впроваджень систем відеоконференцій провалюється через неготовність мережної інфраструктури. Тому ще на етапі вибору постачальника потрібно оцінити можливості своєї мережі й вимоги до пропускну здатності. Можливо, для підтримки Qo на рівні, достатньому для проведення відеоконференцій, виявиться необхідної модернізація. Кожний виробник звичайно дає приблизні розрахунки для одного каналу. Наприклад, для Apache OpenMeetings кожне підключення до сервера вимагає 256 Кбіт/с, хоча клієнт може вибрати підключення з меншою якістю, зменшуючи вимогу до 160 Кбіт/с. У підсумку для сервера потрібно забезпечити (N – кількість учасників):

- вхідний канал – $(256 \times N)$ Кбіт/с;
- вихідний канал – $((256 \times N \times (N - 1)))$ Кбіт/с.

Для клієнтської системи:

- вхідний канал – $(256 \times (N - 1))$ Кбіт/с;
- вихідний канал – 256 Кбіт/с.

Окремий потік в ВВВ вимагає 30–50 Кб/с. Приблизні розрахунки для ВВВ можна знайти в FAQ -goo.gl/Pii7Y. У тім же Skype для відеоконференцій рекомендується більше широкий канал – 4 Мбіт/с (прийом) / 512 Кбіт/с (передача).

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

CISCO WEBEX

Щоб спілкуватися із групою людей, не обов'язково встановлювати свій сервер або закуповувати встаткування, у деяких випадках досить і SaaS. Так, Google Talk і Skype дозволяють зібрати аудіоконференцію на трохи користувачів. Крім того, Skype для бізнес-користувачів (для home потрібно Premium-підписки) дозволяє організувати відеоконференції до десяти учасників. Вартість підписки відносно невисока, а можливостей цілком достатньо для невеликих організацій. Але одне із самих популярних SaaS-Рішень – сервіс веб-конференцій Cisco WebEx, що пропонує якісну відео- і аудіозв'язок, не вимагаючи установки спеціалізованого встаткування. По суті, це зручна платформа для спілкування, вилученого навчання й спільної роботи, де реалізовані всі необхідні функції: попереднє обговорення матеріалів, screen-sharing, whiteboard, вилучений контроль комп'ютера, презентації, VoIP, відео в HD-форматі, чат, запис наради на сервері або клієнті, календар та інше. Можливе надання учасникам різного роду анкет для заповнення. Запрошення можна розіслати за допомогою e-mail, по телефоні або через IM-клієнт. Причому аудіо-конференц-зв'язок можлива не тільки через VoIP, але й через мобільний/стаціонарний телефон. Також WebEx інтегрується з WebEx Social і додатками MS Office, дозволяючи спільно редагувати й публікувати документи. За допомогою спеціального плагіна користувач публікує матеріали й планує наради в MS Outlook, отримане запрошення відразу додається в календар. Для доступу до відеоконференції користувачеві досить підключитися до неї за допомогою браузера. Доступні клієнти для iPhone, iPad, Nokia і BlackBerry. З'єднання захищене за допомогою SSL. Інтерфейс дуже простий, хоча й не локалізований. Поширюється по підписці, у цей час реалізовано чотири рівні доступу – Free (до трьох учасників, відео стандартної якості, 250 Мб місця на диску для файлів), Premium 8, Premium 25 (відповідно до 8 і 25 учасників, 1 Гб місця й додаткові функції) і Enterprise (до 500 учасників). Щоб активувати Free-акаунт, досить просто зареєструватися (на 14 днів дають ще й тестовий Premium), указавши валідний e-mail.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

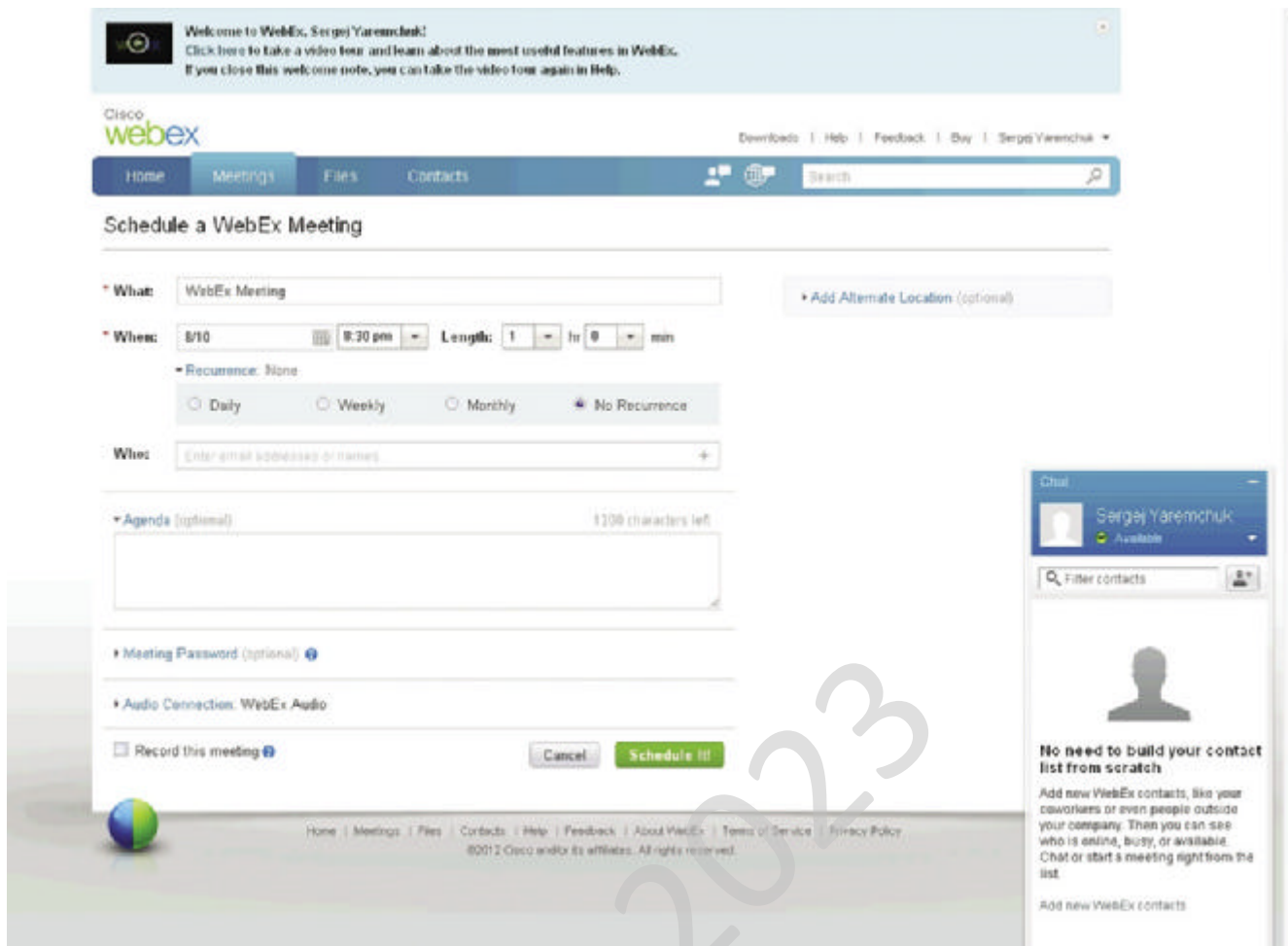


Рисунок 2.4 – Планування конференції в WebEx

TRUECONF SERVER

Сервер відеоконференцій – TrueConf Server це повноцінна система уніфікованих комунікацій, що підтримує кілька режимів віщання: відеодзвінок, групова відеоконференція (симетрична або багатоточечна відеоконференція), дозволяє зв'язати одночасно до 16 учасників так, що всі вони бачать і чувають один одного й можуть спільно працювати над документами.

Під час групової відеоконференції, можна управляти режимом розкладки/розгортання вікон. Для того щоб збільшити зображення певного учасника потрібно клікнути по ньому мишкою. Відеовіщання (асиметрична відеоконференція), дозволяє зв'язати одночасно до 16 учасників так, що всі учасники будуть бачити й чути тільки одного що віщає, а він у свою чергу буде

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

бачити й чути всіх учасників відеоконференції. Селекторна нарада на-246 або рольова відеоконференція під час, який можуть зв'язатися до 120 (TrueConf Online) або 250 (TrueConf Server) вилучених користувачів одночасно. У всіх випадках всі учасники конференції є повноцінними учасниками й можуть розмовляти один з одним за допомогою функції “Трибуна”, запросивши її в Провідного (призначається технічним адміністратором з'єднання й управляє передачею ролей у ході всієї конференції) або “Аудіореplikи” (для цього необхідно просто натиснути відповідну кнопку), а також їм доступні такі засоби для спільної роботи як чат, обмін файлами, електронна дошка для спільної роботи над документом, показ презентацій, показ робочого стола, федерація між серверами (дозволяє здійснювати відеодзвінки й групові відеоконференції між користувачами, зареєстрованими на різних серверах TrueConf Server).



Рисунок 2.5 – Інтерфейс користувача TrueConf Server

Якість відео може варіюватися від SD (320 * 180, один учасник, мінімальний канал) до Ultra HD (3840 * 1440, від 6 учасників). Реалізовано

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

можливість запису конференції, адресна книга й панель статусів. Користувач може показати свій робочий стіл іншим учасникам, які, у свою чергу, одержавши дозвіл, можуть їм управляти. Також, до можливостей інструментів для спільної роботи була додана можливість транслювати в конференцію зображення робочого стола або будь-якого відкритого на комп'ютері додатка у форматі живого відеопотоку. Таким чином, користувачі можуть передавати будь-який відеоконтент на будь-які ВКС термінали, включаючи SIP пристрої й WebRTC користувачів, підключених через браузер.

Групи користувачів дістають різні права, що розмежовують можливості по використанню системи. Для підключення до конференції потрібно Windows (після установки сервер генерує клієнт TrueConf Client), також доступні безкоштовні клієнти для Android, iOS, OS X і найбільш популярних дистрибутивів Linux (Debian, Ubuntu, openSuse, Fedora, CentOS, ALT Linux).

У сервер TrueConf убудований SIP/H.323 шлюз, що дозволяє запрошувати в конференції H.323/SIP відеотермінали. Система володіє й іншими унікальними можливостями, такими як: вихід на ТфОП (дозволяє користувачам програми TrueConf дзвонити іншим користувачам на звичайні телефони, а також запрошувати їх у групові конференції по телефоні), інтеграція з SIP-терміналами й АТС (дозволяє користувачам програми TrueConf здійснювати дзвінки на SIP-термінали та інше ВКС-устаткування), підтримка WebRTC (дозволяє приєднуватися до конференції, що проходить на сервері, прямо з веб-браузера. І ще безліч інших можливостей. Компанія TrueConf дотримується політики BYOD (Bring Your Own Device), що дозволяє перетворити будь-який мобільний пристрій у термінал ВКС.

Продукт з'явився в 2003 році й спочатку поширювався за назвою VideoPort VCS, нове ім'я TrueConf одержав в 2011 році. Розробляється російською компанією, а тому враховує місцеві реалії. Зокрема, невимогливий до якості з'єднання, може використовувати будь-які канали, у тому числі й супутникові, працює через NAT. У програмі закладене динамічне регулювання бітрейта й

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

виробників Lync підключається до SIP або телефонних мереж. Реалізовано можливості спільного доступу до робочого стола, функція віртуальної дошки для одночасної роботи з документом або навчання співробітників. Для тих, хто листується з іноземними партнерами, пропонується плагін Conversation Translator (Lync Guistic), що дозволяє автоматично переводити повідомлення за допомогою сервісу перекладача Bing. Користувач може підключитися до конференції за допомогою звичайного або мобільного телефону. Природно, Lync інтегрується з багатьма продуктами Microsoft – Exchange, SharePoint, Office і Office 365. Користувачам останнього немає необхідності розвертати свою інфраструктуру, досить підключитися до Lync Online. Запрошення для участі в конференції за замовчуванням відправляються за допомогою Outlook. Можна організувати кілька конференцій і перемикатися між ними. Є можливість вибору користувача, від імені якого конференція буде почату. Учасник, що має слово, буде виділений, і камера автоматично перемикається на нього. Приєднатися до зборів можна декількома способами: установивши клієнт Lync 2010 або Lync 2010 Attendee, через веб-браузер з підтримкою Silverlight (Microsoft Lync Web App) або клієнт Office Communicator. Користувачеві, що одержав запрошення, буде запропонований один із цих варіантів. Крім того, підтримується ряд IP-телефонів «Optimized for Lync» (деякі моделі можна знайти на сайті MS – goo.gl/C3fGb). Для проведення відеоконференцій також не потрібне спеціальне встаткування, хоча тут пропонуються сумісні комплекти MS RoundTable, що роблять спілкування більше комфортним. Доступний клієнт Lync Mobile для WP7, Android, iOS, Symbian і BlackBerry, що спрощує підключення до конференції, має функції пошуку адресата й відомостей про присутність співробітника.

Завдяки своїм можливостям, Lync Server буде цікавий тим організаціям, які використовують Windows і будують або реорганізують свою ІТ-інфраструктуру.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написано мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних додатків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські додатки Windows, веб-служби XML, розподілені компоненти, додатки типу “ сервер-клієнт”, додатки баз даних і багато яких інших. В Visual C# 2012 є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку додатків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

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

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод `Main` – точку входу додатка – інкапсулюється у визначення класів. Клас може

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Архітектура платформи .NET Framework

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

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, названому складанням, з розширенням EXE або DLL у більшості випадків. Складання містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

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

Взаємодія між мовами є ключовою особливістю .NET Framework. Оскільки код IL, створюваний компілятором Visual C# відповідає специфікації CTS, код IL на основі Visual C# може взаємодіяти з кодом, створюваним версіями мов Visual Basic, Visual C++, Visual J# платформи .NET Framework і ще більш ніж 20 CTS-сумісних мов. В одному складанні може бути кілька модулів, написаних на

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

різних мовах платформи .NET Framework, і типи можуть посилатися один на одного, як якби вони були написані на одній мові.

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи передачі відеоданих з використанням технології CUDA.

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

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

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

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

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

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.

КБПЗ_2023

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Технічним результатом системи передачі відеоданих з використанням технології CUDA, є подання системи обробки даних і системи відеотрансляції для забезпечення можливості спостереження за множиною одночасно, що відбуваються подій, місця проведення яких мають широкий територіальний розподіл. Система передачі відеоданих з використанням технології CUDA, також забезпечує зниження навантаження на підсистему передачі даних (зниження обсягів переданих даних) при одночасному зниженні ризику втрати відеоданих. Ще одним результатом системи передачі відеоданих з використанням технології CUDA, є подання системи обробки даних, що забезпечує надійну трансляцію відеоданих множині користувачів.

Відеокамери виконані з можливістю передачі відеоданих у пристрої відеоспостереження й можуть бути цифровими або аналоговими. Сигнал від аналогових камер перед подачею в пристрої відеоспостереження піддаються оцифровці за допомогою АЦП. У системі переважно втримуються модулі захвата, виконані з можливістю захвата відеосигналу, наприклад, від ПЗС- або КМОП-матриць відеокамер і формування потоку відеоданих у якому-небудь загальноприйнятому або спеціально для цієї системи розробленому форматі. Модулі захвата (до складу яких можуть входити АЦП) можуть перебувати у відеокамері або в пристрої відеоспостереження.

У тому випадку, коли в систему відеотрансляції входить пристрій відеоспостереження, до кожного із пристроїв відеоспостереження може бути приєднана одна або кілька відеокамер. З'єднання може бути провідним або, щонайменше, частково бездротовим. У випадку провідного з'єднання відеокамера може прямо приєднуватися до пристрої відеоспостереження або ж таке з'єднання

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

може виконуватися через розгалужувач. Також з'єднання може здійснюватися за допомогою загальної шини даних. У переважному варіанті здійснення в системі передачі відеоданих з використанням технології CUDA, використовуються веб-камери, які широко поширені в цей час. Веб-камери звичайно являють собою відносно малогабаритні вироби, що містять оптичний елемент і матрицю для перетворення зображення в електричний сигнал з наступної оцифруванням і передачею сигналу відеозображення в комп'ютер, наприклад, по шині USB.

Наявні в доступі веб-камери також дозволяють разом із зображеннями відеоданих одержувати аудіодані, що представляють собою перетворені в електричний цифровий сигнал звуку, які сприймаються мікрофоном, убудованим у веб-камеру. Хоча в справжньому описі описується обробка, передача й збереження відеоданих, необхідно розуміти, що спільно, паралельно або на додаток до зазначених відеоданих можуть передаватися аудіодані. Через те, що обсяг переданих протягом заданого проміжку часу аудіоданих являє собою лише малу частину в порівнянні з переданими відеоданих, далі можливо вважати, що при передачі відеоданих одночасно відбувається передача аудіоданих, хоча можуть передаватися й відеодані без аудіоданих. Аудіодані можуть зніматися не мікрофоном, убудованим у веб-камеру, а окремим мікрофоном, до таким аудіоданим також застосовне все вищесказане. У заявці описана передача відеоданих і аудіоданих, які можуть передаватися уривками (наприклад, файлами або шматками) або ж потоком відеоданих і аудіоданих, відповідно. Передача відеоданих може відбуватися з вилученням частини відеоданих і/або аудіоданих, наприклад, можуть вилучатися окремі кадри або послідовності кадрів (вилучені дані переважно передаються пізніше).

При використанні в системі відеотрансляції пристроїв відеоспостереження, кожне з них повинне мати можливість з'єднання з відеокамерою, якщо вона зовнішня, або мати убудовану відеокамеру, і підсистемою передачі даних для забезпечення можливості одержання й передачі даних. Зазначені з'єднання можуть бути провідними або бездротовими.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Відповідно, пристрої відеоспостереження виконані з можливістю прийому відеоданих з відеокамер, для чого вони забезпечені можливістю підключення до них зазначених відеокамер і мають відповідне апаратне й програмне забезпечення. Наприклад, у випадку застосування зазначених веб-камер, пристрій відеоспостереження повинен мати один або більше USB-портів і програмне забезпечення, що дозволяє приймати зазначені відеодані. Пристрої відеоспостереження встановлюються на територіально розподілених ділянках по одному або більше залежно від необхідних можливостей по охопту відеоспостереженням ділянок.

На кожній ділянці, де здійснюється відеоспостереження, переважно встановлюється, щонайменше, один пристрій відеоспостереження, до якого приєднуються одна або більше відеокамер і одне або кілька запам'ятовувальних пристроїв. Пристрої відеоспостереження відповідно до системи передачі відеоданих з використанням технології CUDA, виконані з можливістю прийому відеоданих з відеокамер, а також з можливістю передачі, щонайменше, частині відеоданих, отриманих з відеокамер, у локальні запам'ятовувальні пристрої. Пристрої відеоспостереження можуть містити в собі наступні модулі, які можуть бути виконані апаратно, програмно або апаратно: модулі обробки даних, виконані з можливістю обробки відеоданих, одержуваних з відеокамер і/або локальних запам'ятовувальних пристроїв і модулі передачі, щонайменше, частині відеоданих, отриманих з відеокамер, і модулі передачі, виконані з можливістю передачі, щонайменше, частині відеоданих, наприклад, у пристрій збору й трансляції відеоданих через підсистему передачі даних.

Модулі обробки даних можуть бути виконані в різних формах, відомих з рівня техніки, однак вони переважно виконані на основі процесорів, що виконують операції відповідно до програми. У той же час модулі обробки даних можуть бути виконані як відеокарти, убудовані або що підключаються, спеціально призначені для обробки відеоданих. Обробка відеоданих може являти собою стиск і/або кодування й/або перетворення формату відеоданих. До обробки

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

відеоданих також ставиться визначення тих даних, які необхідно передавати, якщо, наприклад, передається частина потоку відеоданих, присвоєння відеоданим ідентифікаційної інформації, напрямок відеоданих у запам'ятовувальні пристрої й модулі передачі даних.

Модулі передачі даних переважно являють собою мережні карти, що забезпечують інтерфейс між пристроями відеоспостереження й підсистемою передачі даних. Мережні карти можуть бути убудованими або що підключаються, наприклад, по шині USB.

Окремі модулі пристрої відеоспостереження або пристрій відеоспостереження в цілому можуть бути розміщені у відеокамері й/або локальному запам'ятовувальному пристрої, наприклад, у їхніх корпусах. У свою чергу окремі елементи відеокамер і/або локальних запам'ятовувальних пристроїв, а також самі відеокамери й/або локальні запам'ятовувальні пристрої можуть бути розміщені в пристроях відеоспостереження, наприклад, у їхніх корпусах. Пристрої відеоспостереження можуть бути встановлені в місцях з обмеженим доступом. У той же час пристрої відеоспостереження можуть бути відсутніми, у такому випадку відеокамери можуть прямо підключатися до локальних запам'ятовувальних пристроїв і/або підсистемі передачі даних.

Важливу роль у системі передачі відеоданих з використанням технології CUDA, грають локальні запам'ятовувальні пристрої, виконані з можливістю одержання з модулів обробки даних і/або відеокамер і збереження, щонайменше, частині відеоданих. Локальні запам'ятовувальні пристрої переважно входять до складу пристроїв відеоспостереження або можуть бути зовнішніми запам'ятовувальними пристроями, що мають з'єднання із пристроями відеоспостереження й розташованими, наприклад, на тій же ділянці, що й пристрої відеоспостереження, або у віддаленому розташуванні. Локальні запам'ятовувальні пристрої можуть бути розташовані в місцях з обмеженим доступом. У деяких варіантах можливо безпосереднє з'єднання локальних запам'ятовувальних пристроїв з відеокамерами й/або підсистемою передачі даних

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

для прямого одержання/передачі даних. Локальні запам'ятовувальні пристрої можуть бути виконані в різній формі – вони можуть являти собою жорсткі диски, енергонезалежну пам'ять, оперативні запам'ятовувальні пристрої, оптичні записуючі пристрої, карти пам'яті й інші запам'ятовувальні пристрої з рівня техніки, широко представлені на ринку. Важливими параметрами, які повинні забезпечувати використовувані запам'ятовувальні пристрої, це обсяг пам'яті й швидкість збереження даних. Обсяг пам'яті переважно повинен бути достатній для збереження всіх відеоданих (а також аудіоданих), одержуваних протягом усього періоду спостережуваної події, або частини відеоданих, які очікують своєї черги передачі. Швидкість збереження даних повинна дозволяти зберігати всі дані, що надходять, без втрат. У випадку недостатності обсягу пам'яті або швидкості збереження даних одного або декількох запам'ятовувальних пристроїв можливе використання додаткових запам'ятовувальних пристроїв так, щоб забезпечити необхідні параметри. У тому випадку, коли одночасно зі збереженням на локальному запам'ятовувальному пристрої здійснюється передача відеоданих через підсистему передачі даних у пристрій збору й трансляції відеоданих, колекторний запам'ятовувальний пристрій і/або інтерфейсний пристрій, повинні бути забезпечені відповідні швидкості збереження відеоданих одночасно з достатніми швидкостями передачі відеоданих.

У локальних запам'ятовувальних пристроях зберігаються переважно всі відеодані, що надходять із відеокамер і, переважно, що пройшли обробку в пристроях відеоспостереження, для того, щоб знизити ризик їхньої втрати, наприклад, при їхній передачі в пристрій збору й трансляції відеоданих, і забезпечити наявність найбільш повних записів відеоданих у пристрої збору й трансляції відеоданих і/або колекторному запам'ятовувальному пристрої. Це необхідно з тієї причини, що підсистема передачі даних може не забезпечувати необхідну швидкість передачі даних або не забезпечувати необхідної якості передавального середовища каналу для безпосереднього збереження в пристрої

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

збору й трансляції відеоданих (колекторних запам'ятовувальних пристроях, з'єднаних із пристроєм збору й трансляції відеоданих) або ж сам пристрій збору й трансляції відеоданих може не мати можливість прийняти всі дані з множини пристроїв відеоспостереження – у такому випадку передається частина даних, що може бути передана підсистемою передачі даних і може бути прийнята пристроєм збору й трансляції відеоданих, а непередані дані, збережені в локальному запам'ятовувальному пристрої, передаються пізніше. Часткова передача відеоданих може здійснюватися на основі поділу записаних на запам'ятовувальному пристрої або одержуваних безпосередньо з відеокамер відеоданих на часові інтервали або ж зі зниженням частоти кадрів у секунду. У випадку обриву каналу зв'язку, тобто коли відсутня можливість передачі даних через підсистему передачі даних, локальні запам'ятовувальні пристрої також забезпечують збереження інформації для наступної передачі. При цьому локальні запам'ятовувальні пристрої можуть відслідковувати стан підсистеми передачі даних і доступність колектора й у випадку відновлення каналу зв'язку продовжити передачу даних на колекторний пристрій. Локальні запам'ятовувальні пристрої можуть бути виконані з можливістю передачі збережених даних пристрої відеоспостереження, у пристрій збору й трансляції відеоданих або в який-небудь інший пристрій зберігання даних, що може входити до складу, наприклад, підсистеми передачі даних або бути окремим запам'ятовувальним пристроєм.

Підсистема передачі даних відповідно до системи передачі відеоданих з використанням технології CUDA, виконана з можливістю одержання від множини пристроїв відеоспостереження й/або відеокамер і/або локальних запам'ятовувальних пристроїв і передачі в пристрій збору відеоданих, щонайменше, частині відеоданих. Підсистема передачі даних може являти собою провідні, оптичні, супутникові й/або радіолінії зв'язку, а також може бути виконана у вигляді мережі передачі даних, такий як локальна мережа зв'язку або Інтернет. Зазначені види засобів зв'язку можуть комбінуватися, канали передачі даних можуть мати з'єднання або бути паралельними й переважно використання

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

вже наявних каналів зв'язку. Елементи підсистеми передачі даних можуть використовуватися для передачі відеоданих від усіх або частини пристроїв відеоспостереження. Переважно підсистема передачі даних виконана із забезпеченням достатніх швидкостей передачі даних від відеокамер, пристроїв відеоспостереження й/або локальних запам'ятовувальних пристроїв у пристрій збору відеоданих, колекторний запам'ятовувальний пристрій, пристрій трансляції відеоданих і/або інтерфейсний пристрій. Пристрій збору відеоданих або безліч таких пристроїв з'єднані з підсистемою передачі даних і виконані з можливістю одержання через підсистему передачі даних відеоданих з відеокамер. Незалежно від того, чи передаються відеодані в підсистему передачі даних для передачі в пристрій збору відеоданих з відеокамер, пристрій відеоспостереження або локальних запам'ятовувальних пристроїв, ці відеодані в кожному разі «захоплені» у відеокамерах і отримані з них. Таким чином, вираження «одержувати відеодані з відеокамер» означає не стільки одержання відеоданих безпосередньо з відеокамер (наприклад, при провідному підключенні відеокамери), а скоріше джерело відеоданих, які можуть бути передані в підсистему передачі даних не тільки з відеокамер, але й із пристроїв відеоспостереження або локальних запам'ятовувальних пристроїв.

Для збереження отриманих відеоданих пристрій збору відеоданих має у своєму складі або з'єднано з колекторним запам'ятовувальним пристроєм, де можуть безпосередньо зберігатися зазначені дані. Відповідно, пристрій збору відеоданих виконано з можливістю збереження в колекторному запам'ятовувальні пристрої, щонайменше, частині отриманих відеоданих. Далі відеодані, збережені в колекторному запам'ятовувальні пристрої, можуть бути витягнуті й переглянуті. У якості колекторного запам'ятовувального пристрою можуть бути використані ті ж типи запам'ятовувальних пристроїв, що й зазначені стосовно локальних запам'ятовувальних пристроїв, однак їхній обсяг і/або швидкості збереження даних повинні бути більше відповідно до збільшившихся обсягів даних, що зберігаються одночасно.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Інтерфейсний пристрій виконаний із забезпеченням можливості вказівки, щонайменше, одним користувачем відеокамери й/або пристрої відеоспостереження й/або локального запам'ятовувального пристрою, відеодані з яких можна вибрати для перегляду, або безпосередньо відеоданих, які необхідні для перегляду. Крім того, інтерфейсний пристрій переважно забезпечує передачу в пристрій трансляції відеоданих і інформації про зазначений користувачем відеокамери й/або пристрої відеоспостереження й/або локальному запам'ятовувальному пристрої й/або відеоданих, які необхідні для перегляду. Для цієї мети інтерфейсний пристрій може надавати користувачеві можливість вибору серед множині відеокамер і/або пристроїв відеоспостереження й/або локальних запам'ятовувальних пристроїв тої відеокамери й/або пристрої відеоспостереження й/або локального запам'ятовувального пристрою, відеодані від яких користувач бажає побачити. В іншому варіанті користувач може вказувати за допомогою інтерфейсного пристрою ідентифікаційну інформацію відеокамери й/або пристрої відеоспостереження й/або локального запам'ятовувального пристрою для тих же цілей.

Відеодані з відеокамери можуть бути цілком однозначно ідентифіковані за допомогою пристрої відеоспостереження, до якого підключена відеокамера, або за допомогою локального запам'ятовувального пристрою, у якому відеодані з відеокамери зберігаються. Це можливо як у тому випадку, коли до пристрої відеоспостереження й/або локальному запам'ятовувальному пристрою підключена одна відеокамера, так і у випадку декількох відеокамер, оскільки користувачеві можуть бути передані відеодані з декількох камер, з яких він потім може вибирати необхідні. Крім того, користувач може запросити відразу необхідні відеодані, якщо в нього є ідентифікаційна інформація цих відеоданих або інформація про те, як їх знайти (наприклад, у файловій системі).

В одному з варіантів пристрій трансляції відеоданих може бути виконане з можливістю одержання із пристрою збору відеоданих (пряма трансляція) і/або колекторного запам'ятовувального пристрою (трансляція збережених даних)

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

відеоданих, отриманих із зазначених користувачем відеокамери й/або пристрої відеоспостереження й/або локального запам'ятовувального пристрою, причому пристрій трансляції відеоданих може бути виконане з можливістю передачі, щонайменше, частині отриманих відеоданих у пристрої кешування.

Пристрою кешування переважно виконані з можливістю з'єднання із системою передачі даних користувачеві, що може відрізнятися від підсистеми передачі даних або, щонайменше, частково збігатися з нею. Крім того, пристрою кешування також переважно виконані з можливістю передачі, щонайменше, частині відеоданих користувачам через зазначену систему передачі даних. Система передачі даних переважно являє собою мережа Інтернет, однак можуть бути використані й інші види мереж і систем зв'язку. Використання пристроїв кешування дозволяє знизити навантаження на пристрій трансляції відеоданих, оскільки в цьому випадку після завантаження пристроєм трансляції відеоданих у пристрій кешування якого-небудь фрагмента відеоданих у випадку повторного запиту цей фрагмент може бути переданий із пристрою кешування без повторного завантаження з колекторного запам'ятовувального пристрою або пристрою збору відеоданих за допомогою пристрою трансляції відеоданих. Крім того, у системі відеотрансляції може бути передбачений пристрій балансування навантаження пристроїв кешування, що може передавати в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу, щонайменше, частині відеоданих у заданий пристрій кешування. Відповідно до цим перевантажений пристрій кешування (тобто змушене передавати безліч відеоданих користувачам з високим або граничним навантаженням) може переслати по команді із пристрою балансування навантаження пристроїв кешування деякий фрагмент відеоданих в інший пристрій кешування, що має менше навантаження. З іншого боку, у випадку високого навантаження на один із пристроїв кешування пристрій балансування навантаження пристроїв кешування може передати в пристрій трансляції відеоданих команду про передачу наступних фрагментів відеоданих в інший пристрій кешування з меншим навантаженням.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Інформацію про завантаження пристрою кешування можуть передавати в пристрій балансування навантаження пристроїв кешування самостійно або по запит із цього пристрою. Крім того, можуть бути передбачені інші способи оцінки завантаження пристроїв кешування, наприклад, по навантаженню передавальних пристроїв, пов'язаних з відповідними пристроями кешування. Завданням балансування навантаження пристроїв кешування є забезпечення навантаження пристроїв кешування приблизно на одному рівні.

У тому випадку, якщо в системі відеоспостереження втримується безліч пристроїв трансляції відеоданих і пов'язаних з ними груп пристроїв кешування, одне або кілька пристроїв балансування навантаження пристроїв кешування можуть регулювати навантаження на систему передачі даних. Наприклад, якщо користувач запросив відеодані в першому пристрої трансляції відеоданих (або відеодані, які перебувають тільки в ньому), а територіально до нього ближче другий пристрій трансляції відеоданих і пов'язані з ним пристрою кешування, одне або кілька пристроїв балансування навантаження пристроїв кешування можуть передати в перший пристрій трансляції відеоданих або в пов'язані з ним пристрою кешування команду передати запитаний фрагмент відеоданих у другий пристрій трансляції відеоданих або пов'язане з ним пристрій кешування. Тим самим може бути знижена довжина ліній зв'язку, задіяних для трансляції відеоданих користувачеві із пристрою кешування. Крім того, перерозподіл переданих відеоданих між пристроями трансляції відеоданих і/або пов'язаними з ними пристроями кешування може здійснюватися з метою балансування навантаження в зазначених групах пристроїв кешування в цілому.

Істотною перевагою системи передачі відеоданих з використанням технології CUDA, є те, що пристрій трансляції відеоданих виконано з можливістю одержання із пристрою збору відеоданих і/або колекторного запам'ятовувального пристрою відеоданих і передачі в пристрій кешування, щонайменше, частині відеоданих у вигляді фрагментів заданої тривалості. Відповідно, поділ відеоданих на фрагменти заданої тривалості може

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

виконуватися різними пристроями, наприклад, відеокамерами, пристроями відеоспостереження, пристроями збору відеоданих і/або пристроями трансляції відеоданих. Крім того, поділ на фрагменти може виконуватися цими пристроями паралельно, причому тривалості фрагментів можуть бути в різних пристроїв різними. Наприклад, у кращому варіанті здійснення пристрій трансляції відеоданих передає в пристрій кешування відеодані у вигляді фрагментів тривалістю 15-30 секунд, а пристроєм збору відеоданих зберігає отримані відеодані в колекторному запам'ятовувальному пристрої відеодані у фрагментах тривалістю близько 1 хвилини. Крім того, кожне з вищевказаних пристроїв у деяких варіантах здійснення системи передачі відеоданих з використанням технології CUDA, може здійснювати обробку відеоданих, що полягає, наприклад, у стиску й/або кодуванні й/або перетворенні формату й/або виділенні, щонайменше, частині відеоданих. Виділення частини відеоданих може здійснюватися безперервними фрагментами, а можуть вирізатися окремі кадри, наприклад, через один – у такий спосіб може бути виконане зниження частоти кадрів.

Пристрій збору відеоданих, колекторний запам'ятовувальний пристрій, пристрій трансляції відеоданих, пристроєм кешування й пристрій балансування навантаження пристроїв кешування можуть бути виділені у вигляді системи збору й трансляції відеоданих, оскільки вони, у деякому змісті, є базовими для організації системи відеотрансляції відповідно до системи передачі відеоданих з використанням технології CUDA, і при наявності такої системи збору й трансляції відеоданих для формування повноцінної системи відеотрансляції до неї необхідно приєднати підсистему передачі даних, до якої потім підключаються відеокамери й/або пристрої відеоспостереження й/або локальні запам'ятовувальні пристрої, а також інтерфейсний пристрій і систему передачі даних для з'єднання з користувачами. Підсистема передачі відеоданих з використанням технології CUDA, може бути виконана у вигляді єдиного апаратного пристрою, у якому складені пристрої являють собою програмні модулі, або, у кращому варіанті,

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

складатися з декількох блоків або модулів, кожний з яких спеціалізується на виконанні своєї функції, що дозволить підвищити продуктивність системи в цілому. У деяких варіантах система передачі відеоданих з використанням технології CUDA, і інтерфейсний пристрій можуть бути об'єднані або бути виконані в складі одного пристрою.

З метою забезпечення наявності в колекторному запам'ятовувальному пристрої повних відеоданих, отриманих відеокамерами й збережених локальних запам'ятовувальних пристроях, але не отриманих за якимись причинами в пристрої збору відеоданих і/або не збережених у колекторному запам'ятовувальному пристрої, у пристрій відеоспостереження й/або локальний запам'ятовувальний пристрій може бути передана інформація про одержання/неотриманні й/або наявності/відсутності в колекторному запам'ятовувальному пристрої, щонайменше, частині відеоданих. У такому випадку пристрої відеоспостереження й/або локальні запам'ятовувальні пристрої можуть бути додатково виконані з можливістю передачі, щонайменше, частині відсутніх у колекторному запам'ятовувальному пристрої й відеоданих у пристрій збору відеоданих через підсистему передачі даних. Передача інформації про одержання/неотриманні й/або наявності/відсутності, щонайменше, частині відеоданих може здійснюватися через підсистему передачі даних, або через інший канал передачі даних замість її або на додаток до неї. Для забезпечення наявності в колекторному запам'ятовувальному пристрої повних відеоданих необхідні відеодані можуть бути передані з локального запам'ятовувального пристрою в колекторний пристрій цілком (за весь часовий проміжок відеоспостереження) або за необхідний інтервал часу через підсистему передачі даних або іншим образом. Така передача також може здійснюватися вибірково або на регулярній основі, наприклад, для всіх або частини локальних запам'ятовувальних пристроїв.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

3.2 Розробка структурної схеми

Пропонується система передачі відеоданих з використанням технології CUDA, що містить:

- щонайменше, один пристрій збору відеоданих;
- щонайменше, один колекторний запам'ятовувальний пристрій;
- щонайменше, один пристрій трансляції відеоданих;
- безліч пристроїв кешування;
- щонайменше, один пристрій балансування навантаження пристроїв кешування.

кешування.

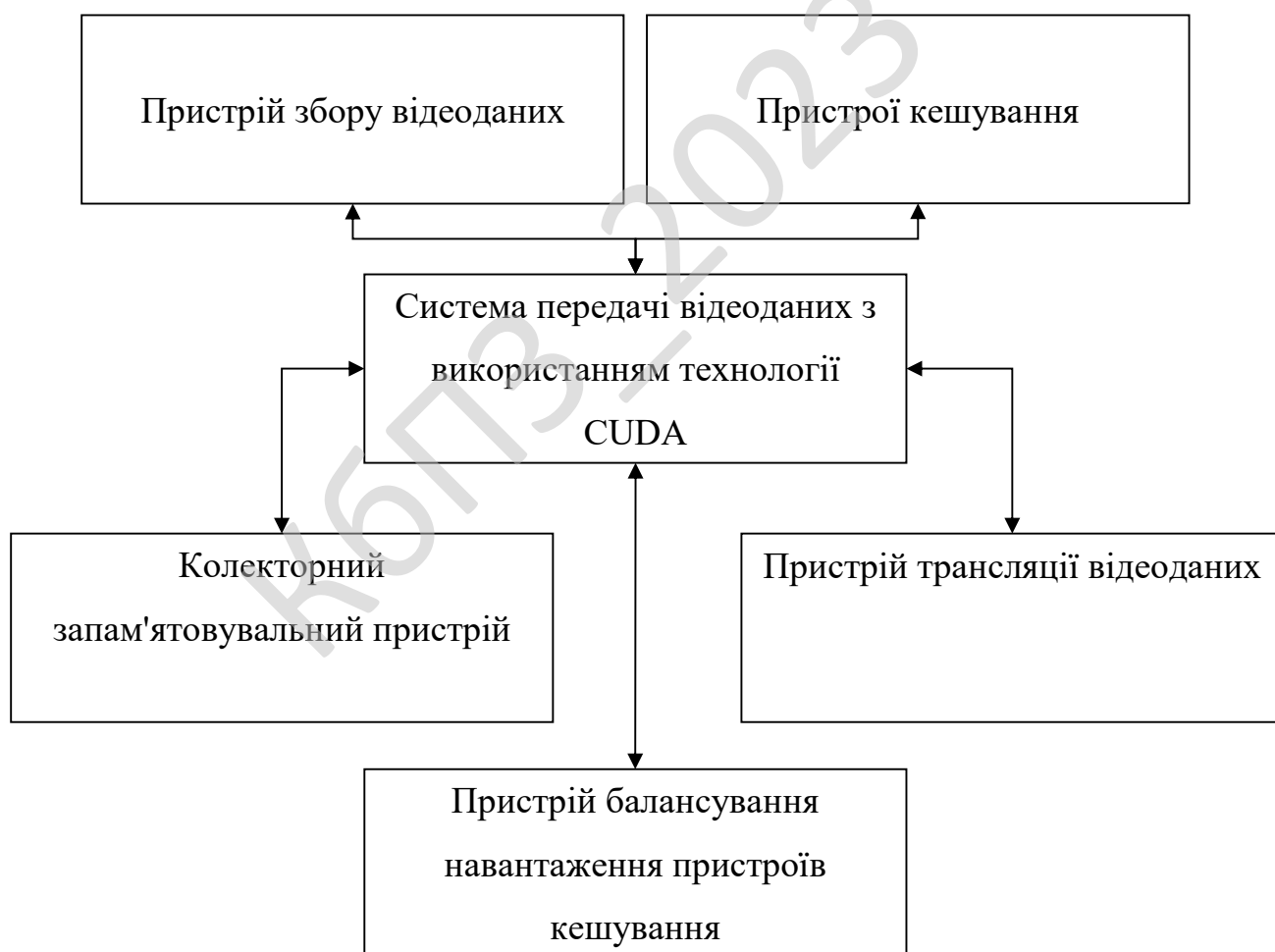


Рисунок 3.1 – Структурна схема системи

Пристрій збору відеоданих виконано з можливістю з'єднання з підсистемою передачі даних і одержання через неї відеоданих з множині територіально розподілених відеокамер.

Пристрій збору відеоданих виконано з можливістю передачі, щонайменше, частині одержуваних відеоданих у колекторний запам'ятовувальний пристрій.

Колекторний запам'ятовувальний пристрій виконаний з можливістю збереження, щонайменше, частині відеоданих, одержуваних із пристроєм збору відеоданих.

Пристрій трансляції відеоданих виконано з можливістю одержання із пристроєм збору відеоданих і/або колекторного запам'ятовувального пристроєм відеоданих і передачі в пристрій кешування, щонайменше, частині відеоданих у вигляді фрагментів заданої тривалості.

Пристрої кешування виконані з можливістю з'єднання із системою передачі даних користувачеві й з можливістю передачі, щонайменше, частині відеоданих користувачам через систему передачі даних.

Пристрій балансування навантаження пристроїв кешування виконано з можливістю передачі в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу, щонайменше, частині відеоданих у заданий пристрій кешування.

Підсистема відеотрансляції, містить:

- безліч відеокамер;
- безліч локальних запам'ятовувальних пристроїв;
- підсистему передачі даних;
- щонайменше, один пристрій збору відеоданих;
- щонайменше, один колекторний запам'ятовувальний пристрій;
- щонайменше, один пристрій трансляції відеоданих;
- безліч пристроїв кешування;
- щонайменше, один пристрій балансування навантаження пристроїв

кешування;

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

– щонайменше, один інтерфейсний пристрій.

Локальні запам'ятовувальні пристрої виконані з можливістю збереження, щонайменше, частині відеоданих з відеокамер.

Підсистема передачі даних виконана з можливістю одержання з відеокамер і/або локальних запам'ятовувальних пристроїв і передачі в пристрій збору відеоданих, щонайменше, частині відеоданих.

Пристрій збору відеоданих виконано з можливістю одержання з підсистеми передачі даних і передачі в колекторний запам'ятовувальний пристрій, щонайменше, частині відеоданих з множині відеокамер.

Колекторний запам'ятовувальний пристрій виконаний з можливістю збереження, щонайменше, частині відеоданих, одержуваних із пристроєм збору відеоданих.

Інтерфейсний пристрій виконаний із забезпеченням можливості вказівки, щонайменше, одним користувачем відеокамери, відеодані з якої необхідні для перегляду, і/або відеоданих, необхідних для перегляду.

Пристрій трансляції відеоданих виконано з можливістю одержання із пристроєм збору відеоданих і/або колекторного запам'ятовувального пристроєм відеоданих і передачі в пристрій кешування, щонайменше, частині відеоданих, відповідно до вказівки користувача, у вигляді фрагментів заданої тривалості.

Пристрої кешування виконані з можливістю з'єднання із системою передачі даних користувачеві й з можливістю передачі, щонайменше, частині відеоданих користувачам через систему передачі даних, причому пристрій балансування навантаження пристроїв кешування виконано з можливістю передачі в пристрій трансляції відеоданих і/або пристрій кешування команди про передачу, щонайменше, частині відеоданих у заданий пристрій кешування.

Пропонується система, здатна відобразити по запиті відеозображення спільно зі звуковим супроводом і утримуючою множиною камер, здатних захоплювати відеозображення, пристрій відеоспостереження, що містить модуль стиску відеосигналу, що надходить від камер, модуль програмування архівування

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

відеозаписів, здатне обробляти відеосигнали, стислі модулем стиску, запам'ятовувальний пристрій для архівування відеозаписів, здатне зберігати стислі відеосигнали, вторинний модуль стиску відеосигналу, підключений до запам'ятовувального пристрою й здатний стискати відеозображення, отримане від запам'ятовувального пристрою, у формат, обраний з формату мобільного пристрою, формату веб-відео й формату ТБ-відео, сервер, адаптований до згаданого формату й здатний опитувати запам'ятовувальний пристрій, і здатне здійснювати зв'язок з вилученим мобільним пристроєм, веб-браузером або контролером телевізійного пристрою зображення й зі шлюзом платежу, що відповідає згаданому формату, і в якому архівуванні відеозображення стискаються й ретранслюються на сервер і потім на згадані мобільний пристрій, веб-браузер або контролер телевізійного відображення по запиті від сервера при ініціюванні запитом від будь-якого з вилученого мобільного пристрою, веб-браузера або контролера телевізійного відображення для конкретного відеозображення, один раз авторизованого згаданим засобом платежу.

CUDA

CUDA (англ. Compute Unified Device Architecture) – технологія GPGPU (англ. General-purpose computing on Graphics Processing Units), що дозволяє програмістам реалізовувати мовою програмування C алгоритми, що виконуватимуться на графічних процесорах Geforce восьмого покоління і вище (Geforce 8 Series, Geforce 9 Series, Geforce 200 Series), Nvidia Quadro і Tesla компанії Nvidia. Технологія CUDA розроблена компанією Nvidia.

Технологія CUDA – це середовище розробки на C, яка дозволяє програмістам і розробникам писати програмне забезпечення для вирішення складних обчислювальних завдань за менший час завдяки багатоядерній обчислювальній потужності графічних процесорів. Простіше кажучи, графічна підсистема комп'ютера з підтримкою CUDA може бути використана, як обчислювальна.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

CUDA дає розробникові можливість на свій розсуд організувати доступ до набору інструкцій графічного прискорювача і управляти його пам'яттю, організувати на ньому складні паралельні обчислення. Графічний процесор з підтримкою CUDA стає потужною програмованою відкритою архітектурою подібно до сьгоднішніх центральних процесорів.

Все це надає в розпорядження розробника низькорівневий, розподілюваний і високошвидкісний доступ до устаткування, роблячи CUDA необхідною основою при побудові серйозних високорівневих інструментів, таких як компілятори, зневаджувачі, математичні бібліотеки, програмні платформи.

Використовує grid-модель пам'яті, кластерне моделювання потоків і SIMD інструкції. Застосовується в основному для високозатратних графічних обчислень і розробок nvidia-сумісного графічного API. Включена можливість підключення до застосунків, що використовують OpenGL 9 і Microsoft Direct3D . Створений у версіях для Linux і Windows.

Первинна версія CUDA SDK була представлена 15 лютого 2007 року. У основі CUDA API лежить розширена мова C. Для успішної трансляції коду цією мовою, до складу CUDA SDK входить власний C-компілятор командного рядка nvcc компанії Nvidia. Компілятор nvcc створений на основі відкритого компілятора

3.3 Розробка функціональної схеми

Загальна схема взаємодії функціональних компонентів типового програмного рішення представлена на рисунку 3.2.

Типове рішення програмного забезпечення складається з наступних компонентів:

- система прийому потоків;
- система прямої трансляції;
- система кешування;

- система зберігання;
- система балансування навантаження;
- моніторинг;
- статистика;
- серверна частина системи керування;
- серверна частина системи відновлення.

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

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

- доступні варіанти якості відеоданих;
- перелік відеоданих, де вказуються адреси доступу до відеофрагментів для кешування й відтворення користувальницьким ПО (браузером).

Система кешування служить для одночасної передачі відеоданих по великій кількості різнорідних з'єднань ТСП і дозволяє скоротити час на доступ до переданих даних файлів. Використання системи дозволяє оптимізувати й прискорити роботу системи прямої трансляції.

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

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

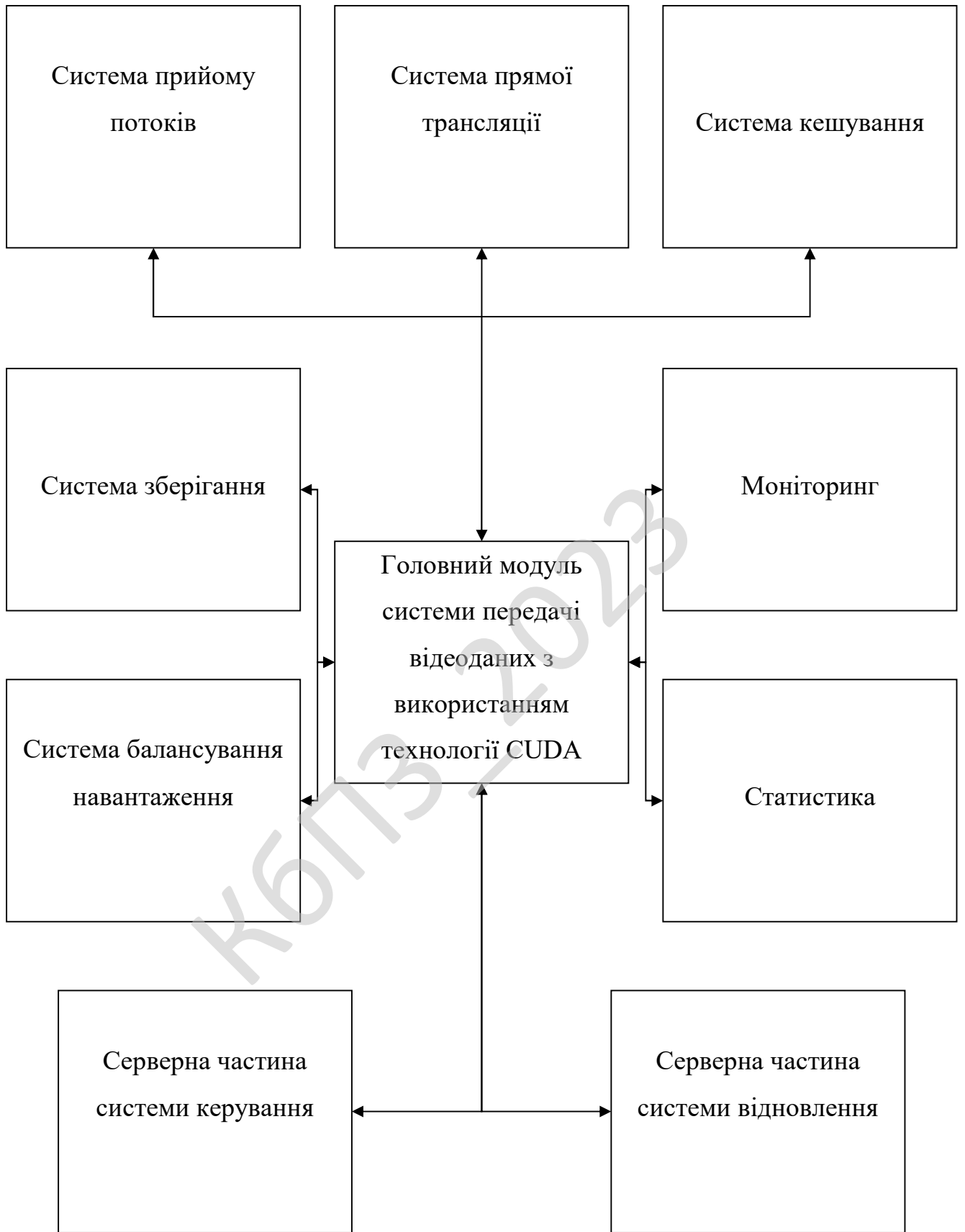


Рисунок 3.2 – Функціональна схема системи

Система керування забезпечує централізоване керування налаштуваннями пристрої відеоспостереження, включаючи налаштування параметрів трансляції: дозволу відеокамер, якості аудіо й відеопотоків, годинного пояса, часу початку й закінчення трансляції. Система також забезпечує авторизацію пристроїв відеоспостереження й вказівка сервера для передачі аудіо й відеопотоків.

Система відновлення забезпечує централізоване відновлення програмного забезпечення пристроїв відеоспостереження.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Обробник помилок.
- Збереження відеоданих з використанням технології CUDA.
- Колекторний запам'ятовувальний пристрій.
- Трансляція відеоданих у режимі реального часу.
- Прийом закодованих даних.
- Передача закодованих даних.
- Покадрове кодування.
- Одержання даних з камери.
- Визначення виду вейвлет-перетворення.
- Сортуння коефіцієнтів.
- Створення масиву коефіцієнтів.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

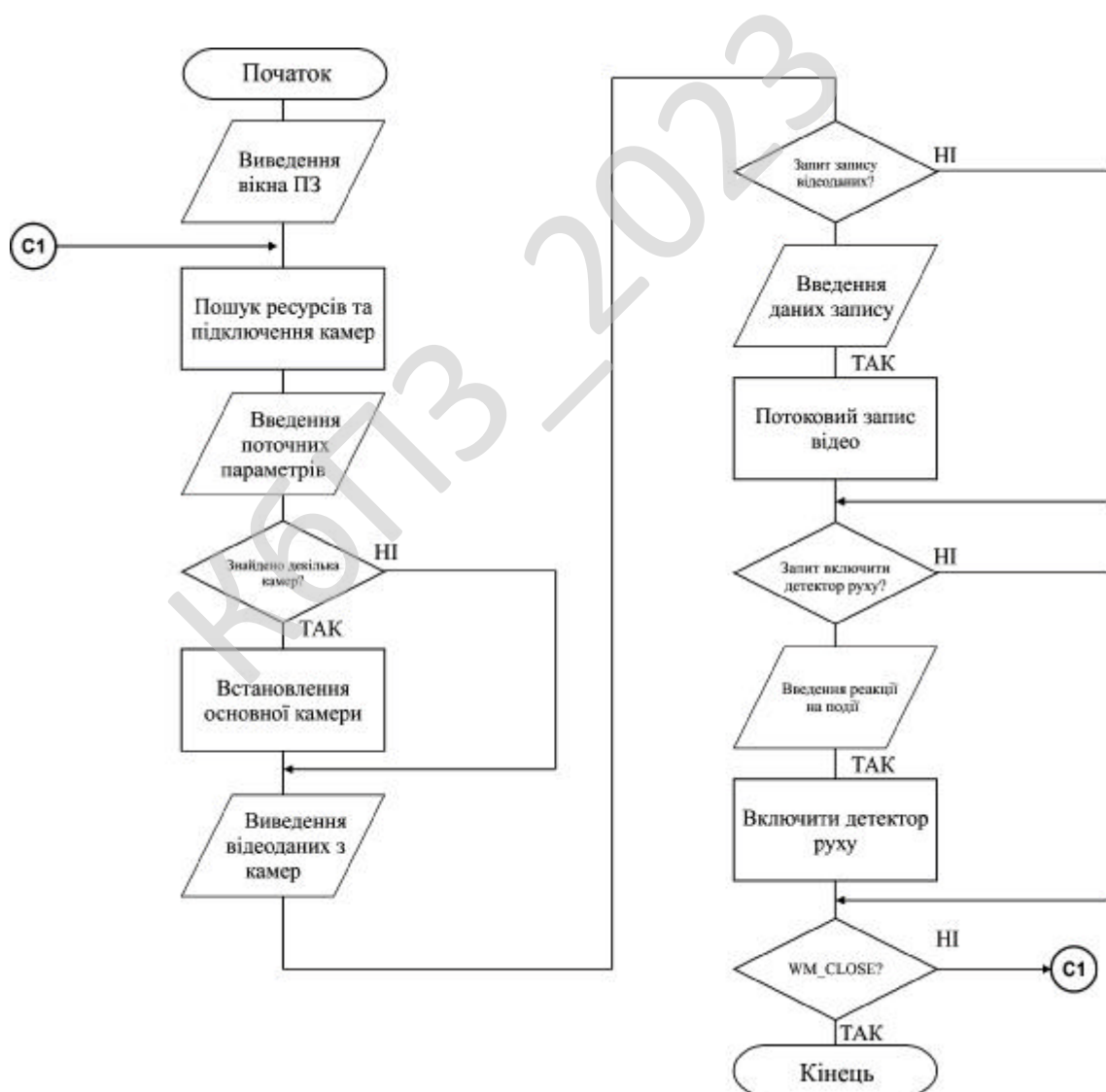


Рисунок 4.1 – Блок схема основної програми

З якої видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – 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.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи передачі відеоданих з використанням технології CUDA.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

Опис алгоритмів функціонування системи

Розглянемо вихідний код алгоритму кодування SPIHT який використовується в системах передачі відеоданих з використанням технології CUDA.

```
namespace DATA
{
    using System;
    using System.Drawing;
    using System.IO;
    using System.Threading;
    using System.Net;
    using videosource;
    /// <summary>
```

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

/// DATASource - Алгоритм SPIHT
/// </summary>
public class DATASource : IVideoSource
{
    private stringsource;
    private stringlogin = null;
    private stringpassword = null;
    private objectuserData = null;
    private int framesReceived;
    private int bytesReceived;
    private bool useSeparateConnectionGroup = false;
    private bool preventCaching = false;
    private int frameInterval = 0;
// інтервал подання фреймів у мілісекундах

```

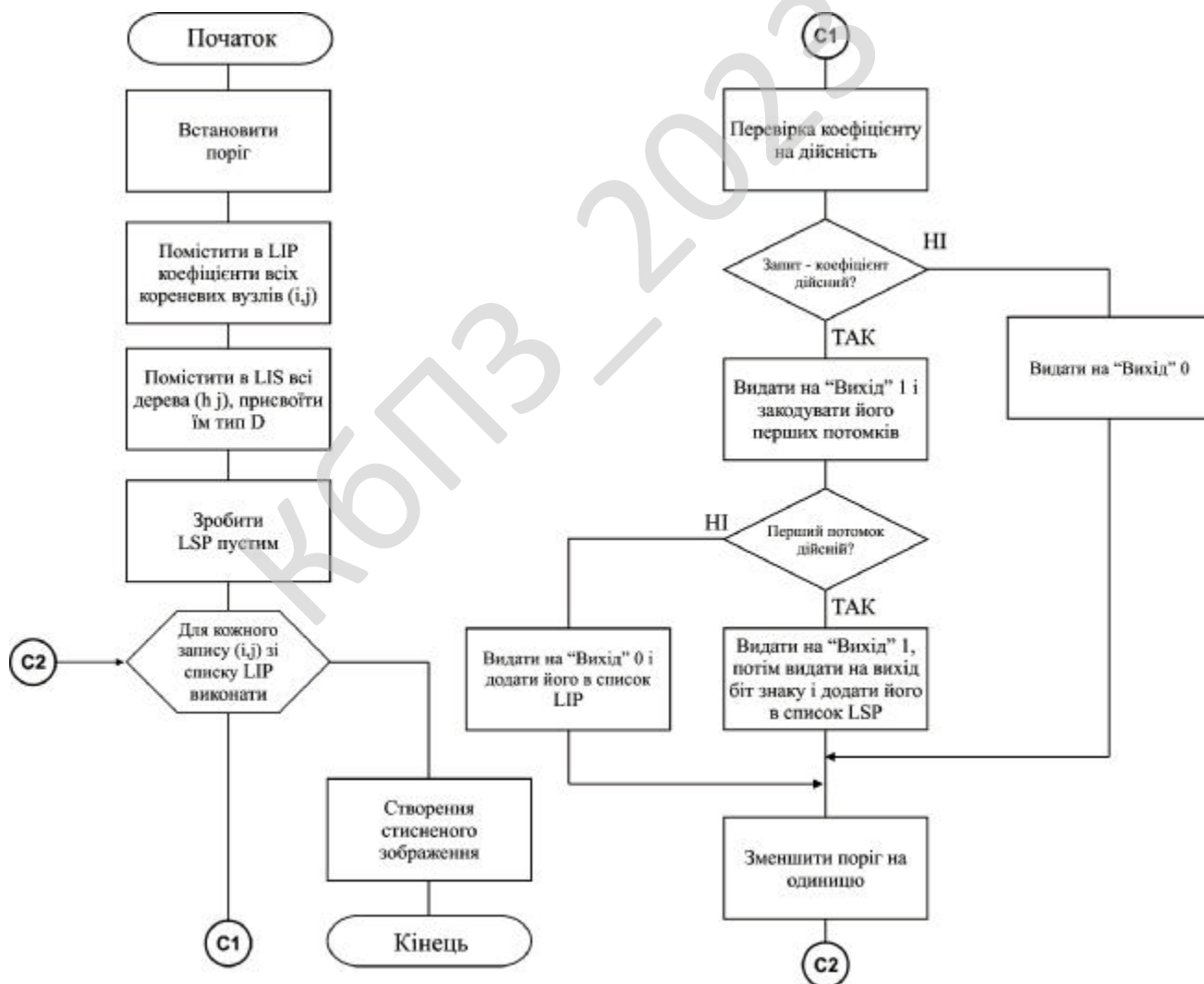


Рисунок 4.2 – Блок схема підпрограми

```

        private const int  bufSize = 512 * 1024;
// розмір буферу
        private const int  readSize = 1024;
// розмір блоку для читання
        private Threadthread = null;
        private ManualResetEvent stopEvent = null;
// нова подія у фреймі
        public event CameraEventHandler NewFrame;
// SeparateConnectioGroup властивості
// індикатор відкриття WebRequest у групі
        public bool    SeparateConnectionGroup
        {
            get { return useSeparateConnectionGroup; }
            set { useSeparateConnectionGroup = value; }
        }
// PreventCaching властивості
// Якщо властивості є правильними, то керуємо параметрию URL.
// Цей клієнт повинен бути встановлений на проксі-сервері.
        public bool    PreventCaching
        {
            get { return preventCaching; }
            set { preventCaching = value; }
        }
// FrameInterval властивості - інтервал між фреймами
//Якщо властивості встановлені в 100, тоді джерело формує 10 фреймів
// в секунду
        public int FrameInterval
        {
            get { return frameInterval; }
            set { frameInterval = value; }
        }
// VideoSource властивості
        public virtual string VideoSource
        {
            get { return source; }
            set { source = value; }
        }
// Властивості підключення
        public string Login
        {
            get { return login; }
            set { login = value; }
        }

```

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

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

```

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

```

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

// Конструктор
public DATASource()
{
}

// Починаємо роботу
public void Start()
{
    if (thread == null)
    {
        framesReceived = 0;
        bytesReceived = 0;
// Створюємо подію
        stopEvent = new ManualResetEvent(false);

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

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

// Чекаємо зупинки події
public void WaitForStop()
{
    if (thread != null)
    {
// Чекаємо зупинки події

```

						ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			57

```

        thread.Join();
        Free();
    }
}
// Подія помилки
public void Stop()
{
    if (this.Running)
    {
        thread.Abort();
        WaitForStop();
    }
}
// Визволяємо ресурси
private void Free()
{
    thread = null;
// Випуск події
    stopEvent.Close();
    stopEvent = null;
}
// Точка входу події
public void WorkerThread()
{
    byte[]buffer = new byte[bufSize];
// буфер читання потоку
    HttpRequest req = null;
    WebResponse resp = null;
    Stream stream = null;
    Random rnd = new Random((int) DateTime.Now.Ticks);
    DateTime start;
    TimeSpan span;
    while (true)
    {
        int read, total = 0;
        try
        {
            start = DateTime.Now;
// створюємо запит
            if (!preventCaching)
            {
                req = (HttpRequest) WebRequest.Create(source);

```

						ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			58

```

        }
        else
        {
            req = (HttpWebRequest) WebRequest.Create(source +
((source.IndexOf('?') == -1) ? '?' : '&') + "fake=" + rnd.Next().ToString());
        }
// встановлюємо логін та пароль
        if ((login != null) && (password != null) && (login != ""))
            req.Credentials = new NetworkCredential(login, password);
// встановлюємо найменування групи підключення
            if (useSeparateConnectionGroup)
                req.ConnectionGroupName = GetHashCode().ToString();
// отримуємо відповідь
            resp = req.GetResponse();
// отримуємо відповідь потоку
            stream = resp.GetResponseStream();
// цикл
            while (!stopEvent.WaitOne(0, true))
            {
// перевіряємо загальне читання
                if (total > bufSize - readSize)
                {
                    total = 0;
                }
// Читаємо наступний блок у потоці
                if ((read = stream.Read(buffer, total, readSize)) == 0)
                    break;
                total += read;
// Додаємо лічильник зчитаних байт
                bytesReceived += read;
            }
            if (!stopEvent.WaitOne(0, true))
            {
// додаємо лічильник фреймів
                framesReceived++;
// зупинка читання зображення
                if (NewFrame != null)
                {
Bitmap bmp = (Bitmap) Bitmap.FromStream(new MemoryStream(buffer, 0, total));
// Клієнт повідомлення
                NewFrame(this, new CameraEventArgs(bmp));
// Будуємо картинку

```

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

```

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

```

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів. Таким чином розглянувши блок-схеми та опис алгоритмів функціонування системи, перейдемо до опису алгоритму та механізмів захисту розробленого програмного забезпечення.

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму CAST-128 (або CAST5) у криптографії, це блоковий алгоритм симетричного шифрування на основі мережі Фейстеля, який використовується в цілому ряді продуктів криптографічного захисту, зокрема деяких версіях PGP і GPG і крім того схвалений для використання Канадським урядом.

Основні відомості

Алгоритм був створений в 1996 році Карлайлом Адамсом (Carlisle Adams) і Стаффордом Таваресом (Stafford Tavares) використовуючи метод побудови шифрів CAST, який використовується також і іншим їхнім алгоритмом CAST-256 (алгоритм-кандидат AES).

CAST-128 складається з 12 або 16 раундів мережі Фейстеля з розміром блоку 64 біта й довжиною ключа від 40 до 128 біт (але тільки з інкрементацією по 8 біт). 16 раундів використовуються коли розміри ключа перевищують 80 біт. В алгоритмі використовуються 8x16 S- блоки, засновані на бент-функції, операції XOR і модулярної арифметиці (модулярне додавання й вирахування). Є три різні типи функцій раундів, але вони схожі за структурою й різняться тільки у виборі виконуваної операції (додавання, вирахування або XOR) у різних місцях.

Хоча CAST-128 захищений патентом Entrust, його можна використовувати в усьому світі для комерційних або некомерційних цілей безкоштовно.

Опис

CAST – це популярний 64-бітовий шифр, що допускає розміри ключа аж до 128 біт

Алгоритм CAST використовує 64-бітовий блок і 64-бітовий ключ. CAST стійкий до диференціального й лінійного криптоаналізу. Сила алгоритму CAST укладена в його S-блоках. В CAST немає фіксованих S-блоків і для кожного додатка вони конструюються заново. Створений для конкретної реалізації CAST S-блок уже більше ніколи не міняється. Інакше кажучи, S-блоки залежать від

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

$zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$
 $K1 = S5[z8] \wedge S6[z9] \wedge S7[z7] \wedge S8[z6] \wedge S5[z2]$
 $K2 = S5[zA] \wedge S6[zB] \wedge S7[z5] \wedge S8[z4] \wedge S6[z6]$
 $K3 = S5[zC] \wedge S6[zD] \wedge S7[z3] \wedge S8[z2] \wedge S7[z9]$
 $K4 = S5[zE] \wedge S6[zF] \wedge S7[z1] \wedge S8[z0] \wedge S8[zC]$
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$
 $xCxDxExF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$
 $K5 = S5[x3] \wedge S6[x2] \wedge S7[xC] \wedge S8[xD] \wedge S5[x8]$
 $K6 = S5[x1] \wedge S6[x0] \wedge S7[xE] \wedge S8[xF] \wedge S6[xD]$
 $K7 = S5[x7] \wedge S6[x6] \wedge S7[x8] \wedge S8[x9] \wedge S7[x3]$
 $K8 = S5[x5] \wedge S6[x4] \wedge S7[xA] \wedge S8[xB] \wedge S8[x7]$
 $z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$
 $z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$
 $z8z9zAzB = xCxDxExF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$
 $zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$
 $K9 = S5[z3] \wedge S6[z2] \wedge S7[zC] \wedge S8[zD] \wedge S5[z9]$
 $K10 = S5[z1] \wedge S6[z0] \wedge S7[zE] \wedge S8[zF] \wedge S6[zC]$
 $K11 = S5[z7] \wedge S6[z6] \wedge S7[z8] \wedge S8[z9] \wedge S7[z2]$
 $K12 = S5[z5] \wedge S6[z4] \wedge S7[zA] \wedge S8[zB] \wedge S8[z6]$
 $x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$
 $x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$
 $x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$
 $xCxDxExF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$
 $K13 = S5[x8] \wedge S6[x9] \wedge S7[x7] \wedge S8[x6] \wedge S5[x3]$
 $K14 = S5[xA] \wedge S6[xB] \wedge S7[x5] \wedge S8[x4] \wedge S6[x7]$
 $K15 = S5[xC] \wedge S6[xD] \wedge S7[x3] \wedge S8[x2] \wedge S7[x8]$
 $K16 = S5[xE] \wedge S6[xF] \wedge S7[x1] \wedge S8[x0] \wedge S8[xD]$

половина, що залишається, ідентична тому, що дане вище, продовження від останнього створило $x0..xf$, щоб генерувати ключі K17 – K32.

$$z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$$

$$z8z9zAzB = xCxDxExF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$$

$$K17 = S5[z8] \wedge S6[z9] \wedge S7[z7] \wedge S8[z6] \wedge S5[z2]$$

$$K18 = S5[zA] \wedge S6[zB] \wedge S7[z5] \wedge S8[z4] \wedge S6[z6]$$

$$K19 = S5[zC] \wedge S6[zD] \wedge S7[z3] \wedge S8[z2] \wedge S7[z9]$$

$$K20 = S5[zE] \wedge S6[zF] \wedge S7[z1] \wedge S8[z0] \wedge S8[zC]$$

$$x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$$

$$xCxDxExF = zCzDzEzF \wedge S5[xA] \wedge S6[x9] \wedge S7[xB] \wedge S8[x8] \wedge S6[z3]$$

$$K21 = S5[x3] \wedge S6[x2] \wedge S7[xC] \wedge S8[xD] \wedge S5[x8]$$

$$K22 = S5[x1] \wedge S6[x0] \wedge S7[xE] \wedge S8[xF] \wedge S6[xD]$$

$$K23 = S5[x7] \wedge S6[x6] \wedge S7[x8] \wedge S8[x9] \wedge S7[x3]$$

$$K24 = S5[x5] \wedge S6[x4] \wedge S7[xA] \wedge S8[xB] \wedge S8[x7]$$

$$z0z1z2z3 = x0x1x2x3 \wedge S5[xD] \wedge S6[xF] \wedge S7[xC] \wedge S8[xE] \wedge S7[x8]$$

$$z4z5z6z7 = x8x9xAxB \wedge S5[z0] \wedge S6[z2] \wedge S7[z1] \wedge S8[z3] \wedge S8[xA]$$

$$z8z9zAzB = xCxDxExF \wedge S5[z7] \wedge S6[z6] \wedge S7[z5] \wedge S8[z4] \wedge S5[x9]$$

$$zCzDzEzF = x4x5x6x7 \wedge S5[zA] \wedge S6[z9] \wedge S7[zB] \wedge S8[z8] \wedge S6[xB]$$

$$K25 = S5[z3] \wedge S6[z2] \wedge S7[zC] \wedge S8[zD] \wedge S5[z9]$$

$$K26 = S5[z1] \wedge S6[z0] \wedge S7[zE] \wedge S8[zF] \wedge S6[zC]$$

$$K27 = S5[z7] \wedge S6[z6] \wedge S7[z8] \wedge S8[z9] \wedge S7[z2]$$

$$K28 = S5[z5] \wedge S6[z4] \wedge S7[zA] \wedge S8[zB] \wedge S8[z6]$$

$$x0x1x2x3 = z8z9zAzB \wedge S5[z5] \wedge S6[z7] \wedge S7[z4] \wedge S8[z6] \wedge S7[z0]$$

$$x4x5x6x7 = z0z1z2z3 \wedge S5[x0] \wedge S6[x2] \wedge S7[x1] \wedge S8[x3] \wedge S8[z2]$$

$$x8x9xAxB = z4z5z6z7 \wedge S5[x7] \wedge S6[x6] \wedge S7[x5] \wedge S8[x4] \wedge S5[z1]$$

					БКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні розділи:

- Меню.
- Функціональні кнопки.
- Блок опису наявних камер.
- Вікно виведення поточних відеоданих.



Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

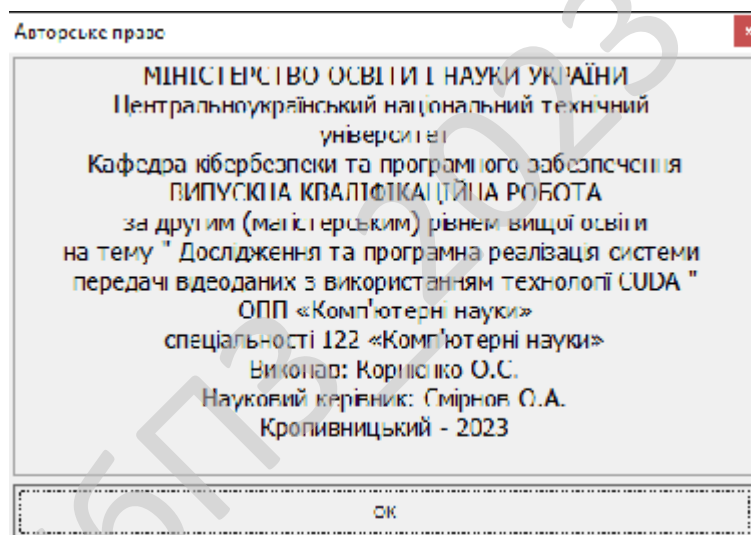


Рисунок 5.2 – Авторське право

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи передачі відеоданих з використанням технології CUDA.

Метою розробки є дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA.

Об'єктом дослідження є процес передачі відеоданих з використанням технології CUDA.

Предметом дослідження є методи передачі відеоданих з використанням технології CUDA.

Методи дослідження базуються на методах комп'ютерної графіки, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод передачі відеоданих з використанням технології CUDA.

– Розроблено вітчизняний продукт передачі відеоданих з використанням технології CUDA, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи передачі відеоданих з використанням технології CUDA. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	99
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000
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	55
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-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

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

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

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 39 = 66 \text{ люд/день.}$$

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	66	Ф 7.1-7.4
Впровадження	13	Д13
Всього	107	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{107 \cdot 1}{60 - 5} = 2 \text{ ставки.}$$

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	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	1	30	0,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	160	400	6,67
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	37,33

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:

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

$$\Phi_{\text{ор}}^c = \frac{37 \cdot 3}{1,2} = 92,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

Продовження таблиці 7.4

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

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	11824	35472
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	2	13000	78000
Інженер - електронщик	0,2	11000	6600
Інженер-системотехнік	0,25	10000	7500
Адміністратор мережі	0,5	10000	15000
Системний програміст	0,1	10000	3000
Дизайнер WEB	0,25	10000	7500
Інженер-верстальник	0,25	10000	7500
Бухгалтер-економіст	0,1	11000	3300
Всього за період розробки	$R_{cn} = 4,9$	-	$\Phi_{роб} = 172872$

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.

$$z_{cd} = \frac{172872}{4,9 \cdot 60} = 588 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

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

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

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 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:

$$B_{y\delta} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми supercomp за 02.11.23 – джерело <https://supercomp.kiev.ua/>.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" LG 22MP58VQ-P 5 мс IPS 1920x1080 250/1000M:1 178/178 D-Sub+HDMI+DVI	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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 випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	199177

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

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	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

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

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 588 \cdot 107 / 99 = 635 \text{ грн.}$$

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, %.

$$Z_d = 635 \cdot 10 \cdot 0,01 = 64 \text{ грн.}$$

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, %.

$$C_{oc} = 0,01 \cdot 22(635+64) = 154 \text{ грн.}$$

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, %.

$$G_{ocn} = 635 \cdot 15 \cdot 0,01 = 95 \text{ грн.}$$

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

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

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Згідно прийнятих норм на підприємстві $n_{\text{вип}}$ приймаємо 1,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=206$ грн., визначаємо вартість паперу за період розробки:

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

$$З_{M1} = 206 \cdot 1,5 = 309 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 51):

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

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 23,6 грн./шт., DVD-R box – 35 грн./шт.

$$З_{M2} = 23,6 \cdot 50 + 35 = 1215 \text{ грн.}$$

Згідно 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.

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

$$З_M = (309 + 1215 + 1702) / 99 = 33 \text{ грн.}$$

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, %.

$$O_n = 635 \cdot 15 \cdot 0,01 = 95 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 99$ прим.):

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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

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

$$A_m = 249636 \cdot 3 / (99 \cdot 12) = 631 \text{ грн.}$$

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	635
2. Додаткова зарплата виконавців	Z_o	64
3. Відрахування на соціальні потреби	C_{oc}	154
4. Загальногосподарські витрати	G_{ocn}	95
5. Витрати на матеріали	Z_M	33
6. Освоєння нових операційних систем, мов програмування	O_n	95
7. Амортизація основних фондів	A_m	631
8. Повна собівартість програмного забезпечення	C_n	1707
9. Плановий прибуток	P_p	940
10. Ціна підприємства $C_n = C_n + P_p$	C_n	2647
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	529,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3176,4

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

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

$$C_n = 635 + 64 + 154 + 95 + 33 + 95 + 631 = 1707 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 55 \cdot 1707 = 940 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3176
Всього капітальних витрат	–	3176

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 технічне обслуговування	Z_p	26840	6710
2. Витрати на електроенергію	$Z_{ел}$	288	72
3. Витрати на амортизацію	$Z_{ам}$	0	794
Всього витрат за рік	I	27128	7576

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 системи на рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 200 годин на рік до 50 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 200 \cdot 100 \cdot 1,1 \cdot 1,22 = 26840 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 50 \cdot 100 \cdot 1,1 \cdot 1,22 = 6710 \text{ грн}.$$

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 ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,45 \cdot 200 \cdot 3,2 = 288 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 0,45 \cdot 50 \cdot 3,2 = 72 \text{ грн}.$$

$$T_e = \frac{479208}{(2647-1707) \cdot 99 \cdot 12 / 3} = 1,3 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	99
2. Повна собівартість розробленої програми	Грн.	1707
3. Ціна розробленої програми	Грн.	2647
4. Плановий прибуток від реалізації розробленої програми	Грн.	940
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	93060
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	30651
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3176
11. Величина економічного ефекту у користувача програмної продукції	Грн.	18758
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,6

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (27128 - 7576) - 0,25 \cdot 3176 = 18758 \text{ грн.}$$

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

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

$$T_{cn} = \frac{3176}{27128 - 7576} = 0,16 \text{ року.}$$

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-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Техніка безпеки – це система правил і заходів, які допомагають запобігти травмам, хворобам і аваріям на робочому місці або в повсякденному житті. Знання техніки безпеки дуже важливе, бо воно рятує життя і здоров'я людей. Наприклад, якщо людина працює на шахті, то вона повинна знати правила безпеки у вугільних шахтах, щоб не потрапити під обвал або не підірватися на міні. Або якщо людина хоче навчитися програмувати, то вона повинна дотримуватися гігієнічних вимог і не сидіти за комп'ютером занадто довго, щоб не зашкодити своїм очам і спині.

Охорона праці та здоров'я у сфері ІТ – це комплекс заходів, які спрямовані на забезпечення безпечних і здорових умов праці для працівників, які використовують інформаційні технології, а також на запобігання травматизму, професійним захворюванням і стресу.

Охорона праці та здоров'я у сфері ІТ включає такі напрями, як:

– Ергономіка – це наука про адаптацію робочого середовища до фізичних і психологічних особливостей людини². Ергономіка вимагає врахування таких факторів, як розміри, форми, кольори, освітлення, шум, температура, вологість, вентиляція, постава, рухи, пози, втома, навантаження на очі та ін. Ергономіка допомагає покращити комфорт, продуктивність і задоволення працівників.

– Комп'ютерна безпека – це захист комп'ютерних систем і даних від несанкціонованого доступу, зміни, знищення або блокування. Комп'ютерна безпека вимагає використання антивірусних програм, фаєрволів, паролей, шифрування, резервного копіювання та інших технологій. Комп'ютерна безпека допомагає запобігти крадіжці, шпигунству, шантажу, саботажу та іншим загрозам.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

– Забезпечувати безпечні і нешкідливі умови праці для працівників, використовуючи сучасні засоби техніки безпеки, санітарно-гігієнічні умови, засоби клектинго та індивідуального захисту, оптимальні режими праці та відпочинку.

– Проводити атестацію робочих місць на відповідність нормативно-правовим актам з охорони праці та аудит з охорони праці.

– Проводити навчання та інструктаж з питань охорони праці, з надання першої медичної допомоги потерпілим від нещасних випадків і правил поведінки у разі виникнення аварії⁵.

– Забезпечувати лікувально-профілактичне обслуговування працюючих, санітарно-побутове обслуговування, пільги і компенсації для працівників, які працюють у важких і шкідливих умовах.

– Нести відповідальність за порушення законодавства про охорону праці та зподіяння шкоди життю і здоров'ю працівників.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Це дуже важливі питання, адже робота з комп'ютером може впливати на здоров'я людини. За ДСанПіН 3.3.2.007-98, шкідливі і небезпечні фактори при роботі з комп'ютером можуть бути такі:

– Електромагнітне випромінювання – це випромінювання, яке створюється комп'ютером і його периферійними пристроями, такими як монітор, пртер, сканер тощо. Це випромінювання може викликати головний біль, запаморчення, розлади сну, зниження імунітету та інші негативні ефекти.

– Електростатичне поле – це поле, яке утворюється внаслідок накопичення електричних зарядів на поверхні комп'ютера і його частин. Це поле може спричинити сухість шкіри, свербіж, подразнення очей, алергічні реакції та інші проблеми.

– Нервово-емоційне напруження – це напруження, яке виникає внаслідок трвлої концентрації уваги, високої вимогливості до результату роботи,

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

нестабільності програмного забезпечення, конфліктних ситуацій тощо. Це напруження може призводити до стресу, депресії, роздратування, погіршення пам'яті та інших порушень.

– Навантаження на органи зору – це навантаження, яке виникає внаслідок догго перебування перед монітором, низької якості зображення, недостатнього освітлення приміщення, поганої ергономіки робочого місця тощо. Це навантаження може спричиняти зниження гостроти зору, появу блимавок і кругів перед очима, синдром сухого ока та інші захворювання.

– Дрібні стереостатичні рухи кінцівок – це рухи, які пов'язані з керуванням клавіатурою і мишею. Ці рухи можуть призводити до перевантаження і запалення суглобів і сухожиль рук і пальців, а також до тендовагінітів і синдрому зап'ясткового каналу.

Таким чином, робота з комп'ютером не така безпечна, як може здатися на перший погляд. Тому дуже важливо дотримуватися правил охорони праці і гігієни при роботі з комп'ютером.

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	3
Довжина	4,6
Висота	3

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	6,9
Об'єм, V	м ³	не менше 20.0	20,7

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

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

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

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

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. 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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

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 ланцюга).

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

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

8.5 Розрахункова частина

Проводемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 3 м, довжина – 4,6 м, висота – 3 м.

У зазначеному приміщенні працює 1 особа.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F=ESKZ/n,$$

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

Для розрахунку дудемо використовувати *світлодіодні панелі* LED панель 42Вт 6000K SUNLED 000000127, світловий потік яких $F_n = 3990$ Лм.

Число ламп визначається по формулі:

$$N = F / F_n$$

де: F – світловий потік,

F_n – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначемо індекса приміщення: $N = 29700 / 3990 = 7,4$ шт.

Приймаємо необхідну кількість *світлодіодних світильників* 8 шт.

Висновки до розділу

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

З цих міркувань було здійснено аналіз умов праці, 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.

Список використаних джерел інформації

1. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи передачі відеоданих з використанням технології CUDA.

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 передачі відеоданих з використанням технології CUDA.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем передачі відеоданих з використанням технології CUDA.

– Досліджена система передачі відеоданих з використанням технології CUDA.

– На основі отриманих результатів досліджень створена програмна реалізація системи передачі відеоданих з використанням технології CUDA.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання передачі відеоданих з використанням технології CUDA.

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

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм CAST-128.

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 18758 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,6 роки.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корнієнко О.С. Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Henry Lloyd. Interactive Computer Graphics. States Academic Press. 2022. 247 p.
3. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
4. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
5. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
6. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
7. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
8. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
9. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. – Д.: НГУ, 2016. – 187 с.
10. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph.. — 2011. — Vol. 30, no. 4. — P. 99:1--99:8.
11. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. — 2011. — Vol. 20, no. 10. — P. 2760—2768.
12. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises».

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

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

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

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

16. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

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

18. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

20. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

21. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

22. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

23. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

24. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

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

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

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

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

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

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

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

32. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

33. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

34. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

35. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

36. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

37. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

38. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

39. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

40. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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

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

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

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

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

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

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

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

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

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

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

52. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. Коваленка О.В., Гриф "Навчальний посібник" надано у відповідності з листом Міністерства освіти і науки України від 26.02.2013 року № 1/11-4368. – Кіровоград: КНТУ 2013. – 257с.

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

54. Смірнов О.А., Доренський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.

					ВКРМ-122.23.0068.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0068.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Корнієнко О.С.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.						
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22МЗ		
Затв.	Смірнов О.А.						
Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи передачі відеоданих з використанням технології CUDA.

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. № 37-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.23.0068.00.00.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 передачі відеоданих з використанням технології CUDA;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0068.00.00.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 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Visual C#.

					ВКРМ-122.23.0068.00.00.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 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-122.23.0068.00.00.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	– 112 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 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 14.12.2023 р.

					ВКРМ-122.23.0068.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Смірнов О.А.

*Дослідження та програмна реалізація
системи передачі відеоданих з використанням технології CUDA*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 41

Літера: РП

Кропивницький – 2023 року

Файл CameraInfo.cs – отримання інформації про камеру

```

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

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

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

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

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

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

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

```

```

    }
    base.Dispose( disposing );
}

#region Windows Form Designer generated code
/// <summary>
/// 	Необхідний метод розробника - не модифікується
/// 	зміст цього методу з редактором коду.
/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.widthLabel = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.nameLabel = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.descriptionLabel = new System.Windows.Forms.Label();
    this.label4 = new System.Windows.Forms.Label();
    this.providerLabel = new System.Windows.Forms.Label();
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
    this.heightLabel = new System.Windows.Forms.Label();
    this.closeButton = new System.Windows.Forms.Button();
    this.pictureBox2 = new System.Windows.Forms.PictureBox();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.Location = new System.Drawing.Point(8, 136);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(40, 14);
    this.label1.TabIndex = 0;
    this.label1.Text = "Width:";
    //
    // widthLabel
    //
    this.widthLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
    this.widthLabel.Location = new System.Drawing.Point(60, 135);
    this.widthLabel.Name = "widthLabel";
    this.widthLabel.Size = new System.Drawing.Size(60, 16);
    this.widthLabel.TabIndex = 1;
    this.widthLabel.TextAlign =
System.Drawing.ContentAlignment.TopCenter;
    //
    // label2
    //
    this.label2.Location = new System.Drawing.Point(140, 136);
    this.label2.Name = "label2";
    this.label2.Size = new System.Drawing.Size(40, 14);
    this.label2.TabIndex = 2;
    this.label2.Text = "Height:";
    //
    // label3
    //
    this.label3.Location = new System.Drawing.Point(8, 9);
    this.label3.Name = "label3";
    this.label3.Size = new System.Drawing.Size(40, 14);
    this.label3.TabIndex = 3;
    this.label3.Text = "Name:";
    //
    // nameLabel
    //
    this.nameLabel.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
    this.nameLabel.Location = new System.Drawing.Point(60, 8);
    this.nameLabel.Name = "nameLabel";
    this.nameLabel.Size = new System.Drawing.Size(200, 16);
    this.nameLabel.TabIndex = 4;

```

```

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

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

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

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

```

```

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

        this.pictureBox2,

        this.closeButton,

        this.heightLabel,

        this.pictureBox1,

        this.providerLabel,

        this.label4,

        this.descriptionLabel,

        this.label5,

        this.nameLabel,

        this.label3,

        this.label2,

        this.widthLabel,

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

}
#endregion

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

```

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

К6П3_2023

Файл MultimodeVideoSource.cs - робота з відеокамерами

```

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

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

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

        // Конструктор
        public MultimodeVideoSource()
        {
        }

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

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

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

```

```

    }

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

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

handler;

        UpdateVideoSource();
    }
}

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

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

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

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

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

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

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

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

```

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

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

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

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

КБПЗ_2023

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

```

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

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

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

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

        private CameraWindow  lastClicked;

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

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

```

```

        {
            fitToWindow = value;

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

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

```

```

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

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

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

    // row 1
    camWindows[0, 0] = cameraWindow1;
    camWindows[0, 1] = cameraWindow2;
    camWindows[0, 2] = cameraWindow3;
    camWindows[0, 3] = cameraWindow4;
    camWindows[0, 4] = cameraWindow5;
    // row 2
    camWindows[1, 0] = cameraWindow6;
    camWindows[1, 1] = cameraWindow7;
    camWindows[1, 2] = cameraWindow8;
    camWindows[1, 3] = cameraWindow9;
    camWindows[1, 4] = cameraWindow10;
    // row 3
    camWindows[2, 0] = cameraWindow11;

```

```

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

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

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

```

```

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

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

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

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

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

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

```

```

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

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

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

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

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

```

```

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

```

```

//
// cameraWindow15
//
this.cameraWindow15.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
this.cameraWindow15.Camera = null;
this.cameraWindow15.Location = new System.Drawing.Point(388,
152);

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

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

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

this.cameraWindow18.Name = "cameraWindow18";
this.cameraWindow18.Size = new System.Drawing.Size(75, 64);
this.cameraWindow18.TabIndex = 17;
this.cameraWindow18.Text = "cameraWindow18";
this.cameraWindow18.Visible = false;
this.cameraWindow18.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
//
// cameraWindow19
//
this.cameraWindow19.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
this.cameraWindow19.Camera = null;
this.cameraWindow19.Location = new System.Drawing.Point(297,
189);

this.cameraWindow19.Name = "cameraWindow19";

```

```

        this.cameraWindow19.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow19.TabIndex = 18;
        this.cameraWindow19.Text = "cameraWindow19";
        this.cameraWindow19.Visible = false;
        this.cameraWindow19.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow20
        //
        this.cameraWindow20.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow20.Camera = null;
        this.cameraWindow20.Location = new System.Drawing.Point(17,
261);
        this.cameraWindow20.Name = "cameraWindow20";
        this.cameraWindow20.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow20.TabIndex = 19;
        this.cameraWindow20.Text = "cameraWindow20";
        this.cameraWindow20.Visible = false;
        this.cameraWindow20.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow21
        //
        this.cameraWindow21.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow21.Camera = null;
        this.cameraWindow21.Location = new System.Drawing.Point(17,
298);
        this.cameraWindow21.Name = "cameraWindow21";
        this.cameraWindow21.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow21.TabIndex = 20;
        this.cameraWindow21.Text = "cameraWindow21";
        this.cameraWindow21.Visible = false;
        this.cameraWindow21.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow22
        //
        this.cameraWindow22.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow22.Camera = null;
        this.cameraWindow22.Location = new System.Drawing.Point(17,
335);
        this.cameraWindow22.Name = "cameraWindow22";
        this.cameraWindow22.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow22.TabIndex = 21;
        this.cameraWindow22.Text = "cameraWindow22";
        this.cameraWindow22.Visible = false;
        this.cameraWindow22.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow23
        //
        this.cameraWindow23.BackColor =
System.Drawing.SystemColors.ControlDarkDark;
        this.cameraWindow23.Camera = null;
        this.cameraWindow23.Location = new System.Drawing.Point(17,
335);
        this.cameraWindow23.Name = "cameraWindow23";
        this.cameraWindow23.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow23.TabIndex = 22;
        this.cameraWindow23.Text = "cameraWindow23";
        this.cameraWindow23.Visible = false;
        this.cameraWindow23.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // cameraWindow24
        //

```

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

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

        this.cameraWindow25.Name = "cameraWindow25";
        this.cameraWindow25.Size = new System.Drawing.Size(75, 64);
        this.cameraWindow25.TabIndex = 24;
        this.cameraWindow25.Text = "cameraWindow25";
        this.cameraWindow25.Visible = false;
        this.cameraWindow25.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.cameraWindow_MouseDown);
        //
        // Мультиплексер
        //
        this.Controls.AddRange(new System.Windows.Forms.Control[] {

                this.cameraWindow25,

                this.cameraWindow24,

                this.cameraWindow23,

                this.cameraWindow22,

                this.cameraWindow21,

                this.cameraWindow20,

                this.cameraWindow19,

                this.cameraWindow18,

                this.cameraWindow17,

                this.cameraWindow16,

                this.cameraWindow15,

                this.cameraWindow14,

                this.cameraWindow13,

                this.cameraWindow12,

                this.cameraWindow11,

                this.cameraWindow10,

                this.cameraWindow9,

                this.cameraWindow8,

                this.cameraWindow7,
```

```

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

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

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

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

// Оновлюємо зображення камери
private void UpdateVisiblity()
{
    if (camerasVisible)
    {
        for (int i = 0; i < MaxRows; i++)
        {
            for (int j = 0; j < MaxCols; j++)
            {
                camWindows[i, j].Visible = ((i < rows) && (j
< cols));
            }
        }
    }
}

```

```

}

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

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

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

    this.SuspendLayout();

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

    this.ResumeLayout(false);
}

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

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

```

Файл VideoStream.cs - робота з відео

```

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

    using videosource;
    using dshow;
    using dshow.Core;

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

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

        // нова подія у фреймі
        public event CameraEventHandler NewFrame;

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

```

```

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

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

    // Конструктор
    public VideoStream()
    {
    }

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

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

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

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

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

            Free();
        }
    }

    // Подія помилки
    public void Stop()
    {

```

```

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

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

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

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

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

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

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

        int code, param1, param2;

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

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

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

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

                // Беремо тип простого грабера

```

```

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

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

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

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

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

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

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

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

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

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

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

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

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

        // запускаємо
mc.Run();

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

    // беремо подію

```

```

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

            //
            if (code == (int) EventCode.Complete)
            {
                break;
            }
        }
    }
    mc.StopWhenReady();
}
// Визначаємо виключення
catch (Exception e)
{
    System.Diagnostics.Debug.WriteLine("----: " +
e.Message);
    failed = true;
}
// Фіналізуємо блок
finally
{
    // визначаємо усі об'єкти
    mediaEvent = null;
    mc = null;
    fileSource = null;
    graph = null;
    sourceBase = null;
    grabberBase = null;
    sg = null;
    if (graphObj != null)
    {
        Marshal.ReleaseComObject (graphObj);
        graphObj = null;
    }
    if (sourceObj != null)
    {
        Marshal.ReleaseComObject (sourceObj);
        sourceObj = null;
    }
    if (grabberObj != null)
    {
        Marshal.ReleaseComObject (grabberObj);
        grabberObj = null;
    }
}
}

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

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

```

```

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

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

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

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

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

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

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

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

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

    // Увідомляємо батьків
    parent.OnNewFrame(img);

    // Будуємо картинку
    img.Dispose();

    return 0;
}
}
}
}

```

Файл VideoStreamSetupPage.cs - робота з відео (інтерфейс)

```

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

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

        // стан змінюваної події
        public event EventHandler StateChanged;

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

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

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

```

```

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

        this.urlBox,

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

}
#endregion

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

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

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

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

    config.source = urlBox.Text;

    return (object) config;
}

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

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

```

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

КБПЗ_2023

Файл CUDASource.cs - алгоритм кодування CUDA

```

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

    using videosource;

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

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

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

        // нова подія у фреймі
        public event CameraEventHandler NewFrame;

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

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

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

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

        // Властивості підключення
    }
}

```

```

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

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

// Конструктор
public CUDASource()
{
}

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

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

```

```

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

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

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

        Free();
    }
}

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

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

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

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

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

        try

```

```

{
    start = DateTime.Now;

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

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

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

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

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

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

```


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

```

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

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

        // стан змінюваної події
        public event EventHandler StateChanged;

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

            //
            rateCombo.SelectedIndex = 0;
        }

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

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

```

```

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

```

```

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

    "Uncontrolled",

    "10 фреймів у секунду",

    "7 фреймів у секунду",

    "5 фреймів у секунду",

    "3 фреймів у секунду",

    "1 фреймів у секунду",

    "12 фреймів у хвилину",

    "6 фреймів у хвилину",

    "4 фреймів у хвилину",

    "3 фреймів у хвилину",

    "2 фреймів у хвилину",

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

    this.rateCombo,

    this.label4,

    this.passwordBox,

    this.label3,

    this.loginBox,

    this.label2,

    this.urlBox,

    this.label1});

```

```

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

    }
    #endregion

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

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

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

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

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

        return (object) config;
    }

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

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

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

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

```

Файл About.cs - довідка

```

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

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

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

        public About() {
            InitializeComponent();
        }

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

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

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

```

на тему:

Дослідження та програмна реалізація системи передачі відеоданих з використанням технології CUDA

Керівник: Смірнов О.А.

Розробив: студент Корнієнко Олександр Сергійович
гр. КН-22МЗ

```

м. Кропивницький 2023";
    //
    // button1
    //
    this.button1.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
    this.button1.ForeColor = System.Drawing.Color.White;
    this.button1.Location = new System.Drawing.Point(16, 168);
    this.button1.Name = "button1";
    this.button1.Size = new System.Drawing.Size(136, 24);
    this.button1.TabIndex = 2;
    this.button1.Text = "&Close";
    this.button1.Click += new
System.EventHandler(this.button1_Click);
    //
    // About
    //
    this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
    this.BackColor = System.Drawing.Color.Black;
    this.ClientSize = new System.Drawing.Size(480, 208);
    this.Controls.Add(this.button1);
    this.Controls.Add(this.richTextBox1);
    this.Controls.Add(this.pictureBox1);
    this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.None;
    this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
    this.Name = "About";
    this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
    this.Text = "About";
    this.ResumeLayout(false);

    }
    #endregion

    #region events

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

    #endregion
}
}

```