

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

”Допущено до захисту”

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

_____ Олексій СМІРНОВ

“ ____ ” _____ 2023 р.

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

**“Дослідження та програмна реалізація системи компактного
представлення відеоданих з метою зменшення часу їх обробки і
передачі у мережі Інтернет”**

Виконав здобувач вищої освіти

II курсу, групи КІ-22М-2

ОПП «Комп’ютерна інженерія»

спеціальності 123 «Комп’ютерна інженерія»

_____ Шульга В.Р.

« ____ » _____ 2023 р.

Керівник проекту

кандидат технічних наук, доцент

_____ Смірнов С.А.

« ____ » _____ 2023 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Шульгі Владиславу Романовичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет

2. Керівник роботи

Смірнов Сергій Анатолійович, канд. техн. наук, доцент

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

затверджені наказом вищого навчального закладу № 35-13 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 компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

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

Шульга В.Р. Дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Метою розробки є дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Об'єктом дослідження є процес компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Предметом дослідження є методи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

Результат роботи – програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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 Delphi 10.4.

Ключові слова: комп'ютерна інженерія, відеодані, Інтернет

ABSTRACT

Shulha V.R. Research and software implementation of a system of compact presentation of video data in order to reduce the time of their processing and transmission on the Internet. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for a system of compact presentation of video data in order to reduce the time of their processing and transmission on the Internet.

The purpose of the development is the research and software implementation of a system of compact presentation of video data in order to reduce the time of their processing and transmission on the Internet.

The object of the study is the process of compact presentation of video data in order to reduce the time of their processing and transmission on the Internet.

The subject of research is methods of compact presentation of video data with the aim of reducing the time of their processing and transmission on the Internet.

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

The result of the work is the software implementation of a system of compact presentation of video data in order to reduce the time of their processing and transmission on the Internet.

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 Delphi 10.4 environment.

Keywords: computer engineering, video data, Internet

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	39
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	45
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	45
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	61
6 НАУКОВА НОВИЗНА	66

					ВКРМ-123.23.0051.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет	Літ.	Аркуш	Аркушів
Розроб.	Шульга В.Р.					М	1	105
Перев.	Смірнов С.А.					ЦНТУ КІ-22М-2		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	67
7.2 Розрахунок трудомісткості розробки програмної продукції.....	69
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	75
7.5 Визначення собівартості розробки та ціни програмної продукції.....	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	83
7.7 Визначення експлуатаційних витрат.....	84
7.8 Визначення економічної ефективності програмної продукції.....	85
7.9 Висновок.....	87
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1 Вступ.....	88
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	89
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	90
8.4 Розробка заходів з умов поліпшення охорони праці.....	93
8.5 Розрахунок штучного освітлення	94
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- АСУЗТ – автоматизована система управління залізничним транспортом
- СВІ – системи виведення інформації

КБПЗ-2023

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Більшість цифрових зображень містять високий ступінь надмірності, що означає, що ефективна техніка стиснення може значно зменшити кількість інформації, необхідної для їх зберігання або передачі. Цю надлишковість можна знайти між окремими пікселями, між лініями або між кадрами, коли сцена нерухома або трохи рухається. Приблизно з 1989 року рухомі цифрові відеозображення були інтегровані з програмами. Складність реалізації рухомого цифрового відео полягає у величезній смузі пропускання, необхідної для кодування відеоданих. Наприклад, зображення чверті екрана (320 x 240 пікселів), що відтворюється на відеоекрані RGB із повною швидкістю 30 кадрів/с, потребує зберігання та передачі 6,9 мільйонів байтів на секунду (МБ/с). Ця швидкість передачі даних просто непомірно висока, тому засоби стиснення цифрового відео, придатні для відтворення в реальному часі, є необхідним кроком для широкого впровадження додатків цифрового відео руху. Було розроблено та реалізовано багато алгоритмів стиснення цифрового відео. Коефіцієнти стиснення цих алгоритмів змінюються залежно від суб'єктивного прийняттого рівня помилки, визначення слова «стиснення» та того, хто робить заяву.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

- Огляд існуючих систем компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.
- Дослідження системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.
- Програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Об'єктом дослідження є процес компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Предметом дослідження є методи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

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

– Удосконалено метод компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

– Розроблено вітчизняний продукт компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Ідеальна техніка стиснення відео повинна мати такі характеристики:

- Забезпечуватиме рівні стиснення, які конкурують з MPEG, без небажаних артефактів.
- Можна відтворювати в реальному часі за допомогою недорогої апаратної підтримки.
- Може легко погіршуватися під час перевантаження мережі або на повільній платформі.
- Можна стискати в реальному часі за допомогою недорогої апаратної підтримки.

У цьому розділі ми зосередимося на двох усталених стандартних методах, MPEG для інтенсивних відеозастосунків і стандарті H.261/H.263 для відеотелекомунікацій. Ми також опишемо перспективну методику XYZ. Технології стиснення відео зробили можливим ряд застосувань.

Чотири різні застосування стисненого відео можна підсумувати як: (а) споживче телевізійне мовлення, (б) споживче відтворення, (в) настільне відео та (г) відеоконференції.

Споживче телевізійне мовлення, яке включає доставку цифрового відео в домівки, зазвичай вимагає невеликої кількості високоякісних компресорів і великої кількості недорогих декомпресорів. Очікуваний ступінь стиснення становить приблизно 50:1.

Споживацькі програми відтворення, такі як бібліотеки компакт-дисків та інтерактивні ігри, також потребують невеликої кількості компресорів і великої кількості недорогих декомпресорів. Необхідний ступінь стиснення становить приблизно 100:1.

						ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			6

Настільне відео, яке включає системи для створення та редагування відеопрезентацій, є симетричним додатком, що потребує однакової кількості кодерів і декодерів. Очікуваний ступінь стиснення знаходиться в діапазоні від 5:1 до 50:1.

Додатки для відеоконференцій також вимагають такої ж кількості кодерів і декодерів, а очікуваний коефіцієнт стиснення становить приблизно 100:1.

Тому актуальною є задача розроблення методу представлення зображень, який дозволяє кодувати зображення в реальному масштабі часу із урахуванням динаміки зміни зображень та без втрати інформації.

1.2 Область застосування

Стандарт H.261/263, який зазвичай називають рх64 Кбіт/с, оптимізований для досягнення дуже високого коефіцієнта стиснення для передачі повнокольорового відео руху в реальному часі. Алгоритм стиснення рх64 поєднує внутрішньокадрове та міжкадрове кодування, щоб забезпечити швидку обробку для стиснення та розпакування відео на льоту. Внутрішньокадрове кодування відноситься до кодування окремих кадрів, тоді як міжкадрове кодування – це кодування кадру відносно попередніх або майбутніх кадрів.

Стандарт рх64 оптимізований для таких програм, як телекомунікації на основі відео. Оскільки ці програми зазвичай не потребують інтенсивного руху, алгоритм використовує обмежені стратегії пошуку та оцінки руху для досягнення вищих коефіцієнтів стиснення. Для стандартних зображень відеозв'язку можна досягти коефіцієнта стиснення від 100:1 до понад 2000:1.

Стандарт стиснення рх64 призначений для покриття всієї пропускної здатності каналу ISDN ($p = 1, 2, \dots, 30$). Для $p = 1$ до 2 через обмежену доступну пропускну здатність за допомогою цього алгоритму стиснення можна реалізувати лише візуальний зв'язок на робочому столі обличчям до обличчя (відеотелефон).

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Однак для $p > 6$ передаються більш складні зображення, і алгоритм підходить для додатків відеоконференцій.

Для алгоритму H.261 визначено два формати: CIF і QCIF, а для алгоритму H.263 визначено три додаткові формати: SQCIF, 4CIF і 16CIF.

Загальний проміжний формат (CIF) – це формат без черезрядкової розгортки, заснований на 352x288 пікселів на кадр зі швидкістю 30 кадрів на секунду. Ці значення представляють половину активних рядків телевізійного сигналу 625/25 і швидкість зображення сигналу 525/30 NTSC. Таким чином, системам 625/25 потрібно лише перетворення швидкості зображення, тоді як системам NTSC потрібно лише перетворення номерів рядків.

Кольорові зображення кодується за допомогою однієї яскравості та двох кольорово-різницьких компонентів (формат YCbCr), визначених стандартом CCIR 601. Компоненти Cb і Cr субдискретизуються з коефіцієнтом два в горизонтальному та вертикальному напрямках і мають 176x144 пікселів на кадр. Співвідношення сторін зображення для всіх п'яти форматів на основі CIF становить 4:3.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Існує велика кількість різних форматів відео й аудіо. У той же час, більшість пристроїв (таких, як iPod) і програм (таких як Windows Media Player) найчастіше підтримують лише кілька конкретних форматів. DVD або WMV відео не буде програватися, наприклад, на iPod без конвертації в MP4 файл. У даній роботі мова йтиме про програми для конвертації відео, точніше, про їхні безкоштовні аналоги.

Всі програми для стиску відео умовно можна умовно розділити на кілька категорій.

1. Пристрій-орієнтовані програми підійдуть невимогливим користувачам, яким найчастіше потрібна лише кнопка "конвертація" для одержання жаданого формату. Такі програми дуже прості у використанні, містять набір попередніх передвстановок (пресетів), і лише деякі надають тонкі ручні налаштування. Їх ціль – швидко й без дратунок конвертувати в певний "апаратний" формат (3gp, iPod, PSP, Xbox і т.д.).

2. Так звані, користувальницькі програми, є протилежністю для програм попередньої категорії. Вони орієнтовані на більше спокушені й вимогливих користувачів. Програми мають багато тонких ручних налаштувань і, у теж час, пропонують передвстановки для різних форматів. Мають більше складний інтерфейс.

3. Гібридні програми – це якийсь компроміс між вищезгаданими категоріями й підходять для будь-яких типів завдань. Подібні програми, як правило, мають розширеного функціонала, приміром, містять убудований

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

відеоредактор, можливість вибору пресетів і доступу до ручних налаштувань і т.п.

Гібридні програми

Xvi4PSP

Xvi4PSP – безкоштовний відеоконвертер, шоста версія якого була визнана кращим російськомовним конвертером. Цей кодувальник є автономним, він містить у собі всі необхідні кодеки, що рятує вас від необхідності встановлювати горезвісні кодекпаки. Подібна реалізація забезпечує підтримку всіх мислимих форматів відео й аудіо. Крім того, програма пропонує величезну кількість шаблонів для конвертації відео, у їхнє число входять і пресети для конвертації під портативні пристрої. Конвертер постійно оновлюється розроблювачами, має більшу армію користувачів.



Рисунок 2.1 – Интерфейс користувача Xvi4PSP

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

iWisoft Video Converter

iWisoft Video Converter – при старті запускає браузер для перевірки відновлень. На щастя, це єдиний недолік цього безкоштовного конвертера. Інтерфейс програми продуманий і простий у використанні. Містить добре організований набір попередніх настроювань, підтримує їхнє створення й збереження. Також програма має убудований відеоредактор за допомогою якого можна наносити водяні знаки, обрізати й з'єднувати відеофрагменти, міняти контраст, яскравість, гучність звуку, знімати скріншоти з відео й т.п. Підтримує пакетну обробку відео й аудіо. Розмір: 8,7 Мб.



Рисунок 2.2 – Інтерфейс користувача iWisoft Video Converter

XMedia Recode

XMedia Recode – якісний відеоконвертер з функціями редактори (обрізка, зміна дозволу й т.п.). Програма володіє зручним і строгим, я б навіть сказав

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

класичним інтерфейсом і гарним, чи ледве не еталонним (є таку думку) якістю кодування. Підтримує безліч передвстановок кодування для різноманітних пристроїв, і, крім того, є присутнім можливість тонких ручних настроювань. Програма лише формально віднесена до категорії "гібридних". Насправді, вона прекрасно вписала б у кожную з категорій, тобто програма універсальна:) Розмір: 6.2 Мб.

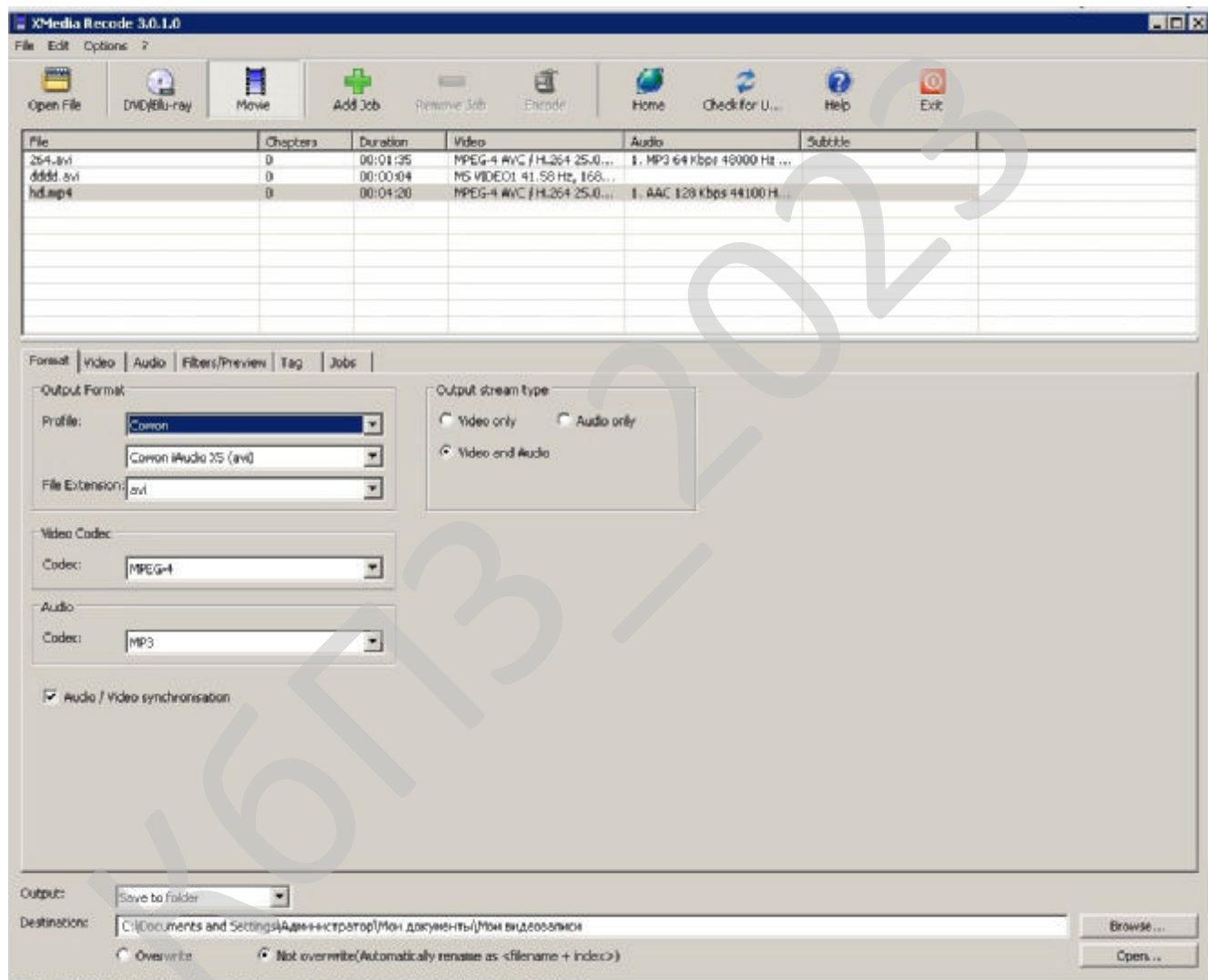


Рисунок 2.3 – Інтерфейс користувача XMedia Recode

Format Factory

Format Factory – є гарним вибором практично для будь-яких завдань. Містить пресети для більше 100 пристроїв, перетворить у десятків форматів, а також дозволяє робити ручні налаштування кодування. Крім того, як треба з назви, програма виявляє собою фактично фабрику по конвертації. І конвертує не тільки відео, але й аудіо, картинки, документи, словом, всі що можна стиснути або перевести в інший формат. Також уміє створювати образи DVD і CD, робити ріппінг DVD і Audio CD, відновлювати ушкоджені аудіо й відео файли. Мультимовна, є російська і українська мова. Розмір: 41,2 Мб.



Рисунок 2.4 – Интерфейс користувача Format Factory

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Користувальницькі програми

FFCoder

FFCoder – поєднує під однією оболонкою різні CLI-Утиліти, кожна їх яким краща у своєму класі. Як і більшість програм у цьому огляді, крім усього іншого, утиліта підтримує джентльменський набір функцій для стиску відео, володіючи при цьому простим і витонченим інтерфейсом. Підтримує досить просунуті конфігурації для кожного відеоформату й кодеку, аж до настроювань висвітлення й рендеринга. Незважаючи на деякі особливості (див. нижче), програма стане відмінним вибором для просунутого користувача, якому потрібний повний контроль над процесом. А особливість полягає в тому, що програма вимагає Microsoft .NET Framework 2.0 (для XP) і Nero AAC codec. Втім все це вона сама й завантажує при установці. Розмір: 32 Мб.

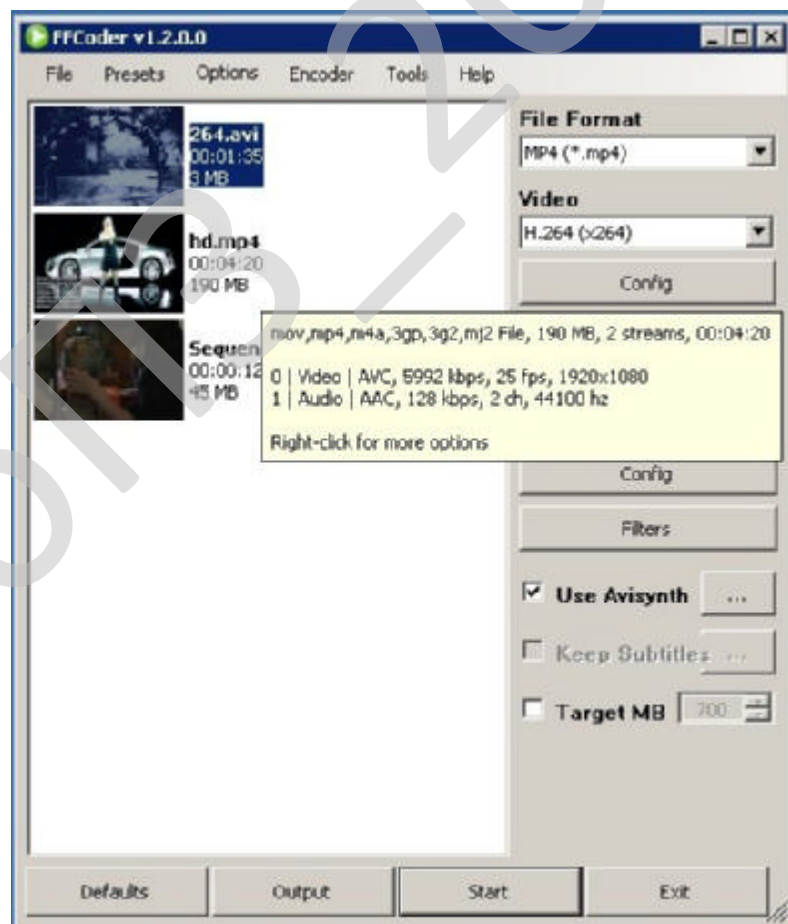


Рисунок 2.5 – Інтерфейс користувача FFCoder

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

SUPER

SUPER – має простий, але трохи дратівний інтерфейс (всі настройки й пресети в одному вікні). З її допомогою можна конвертувати файли різних форматів практично одним натисканням кнопки. По швидкості кодування SUPER обганяє багатьох своїх конкурентів. Однак розплатою за це стає трохи мудрована установка – програма нав'язливо пропонує встановити додатковий софт, що до конвертації відео має досить віддалене відношення. Розмір: 38 Мб.

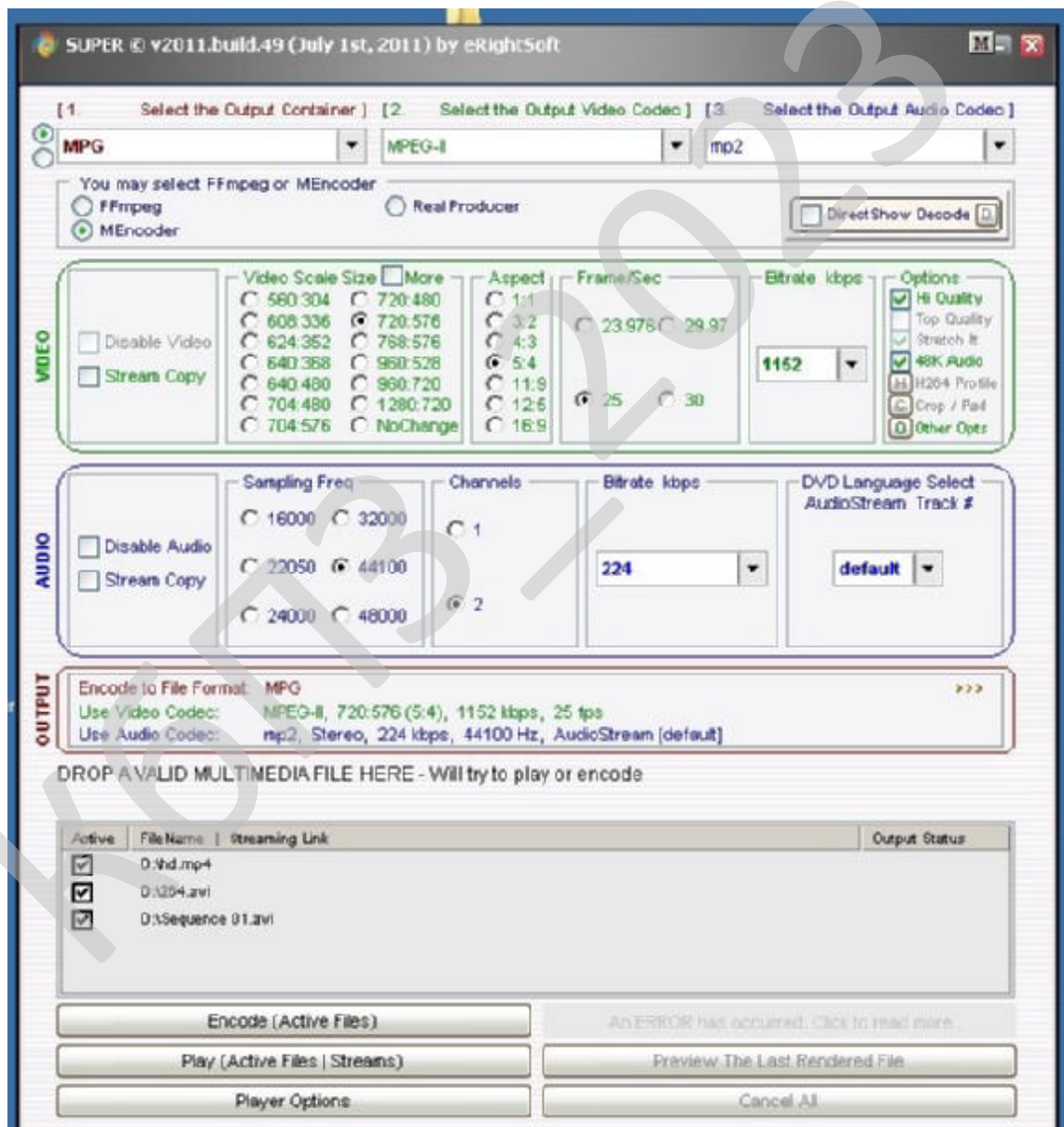


Рисунок 2.6 – Інтерфейс користувача SUPER

Пристрій-орієнтовані програми

Pazera Video Converters Suite

Pazera Video Converters Suite – це пакет з декількох окремих конверторів (AVI, MP4, MPEG, FLV, 3GP і т.п.) об'єднаних під загальної GUI (оболонкою). Кожний конвертер володіє декількома передвстановками для конкретних пристроїв, і при тім, дає можливість ручного коректування настроювань, які можна зберігати як пресети . Оболонкою для програми служить простий текстової інтерфейс. Також дозволяє робити "ріп" CD-audio дисків і витягати звукову доріжку з відеофайлів. Розмір: 33,4 Мб.

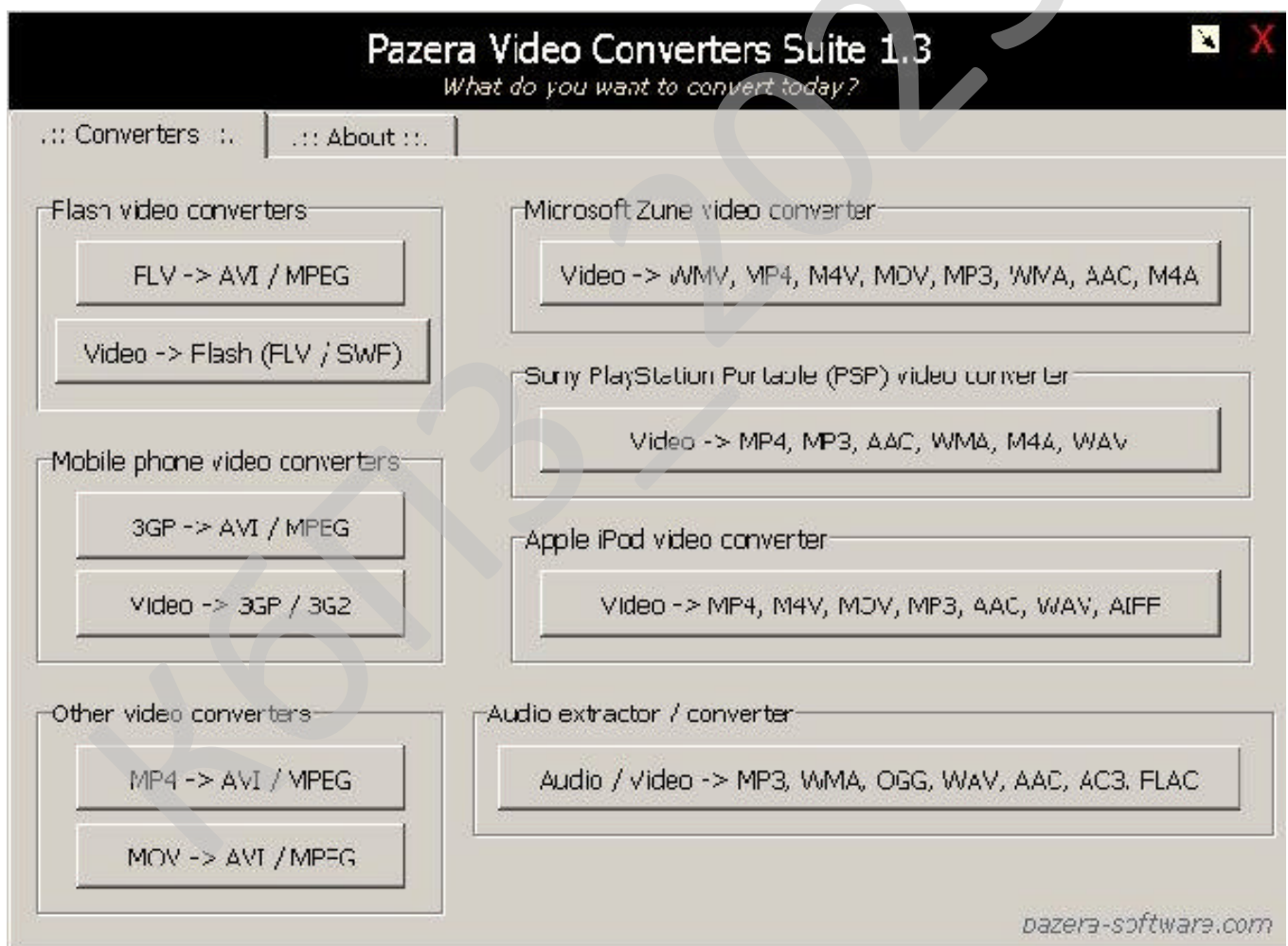


Рисунок 2.7 – Інтерфейс користувача Pazera Video Converters Suite

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Miro Video Converter

Miro Video Converter – дуже простий безкоштовний конвертор. Перегляд відео на мобільному телефоні, КПК або приставці стало щоденністю. Щоб одержати подібний формат відео, немає потреби вникати в налаштування, возитися з кодеками, дозволами й інтерлейсингом. Необхідна лише одна заповітна кнопка, що тиснеш і... Miro Video Converter – саме і є така кнопка. Пропонує набір пресетів для стиску відео, але підтримує конвертацію тільки одного файлу за раз. Розмір: 37,8 Мб.

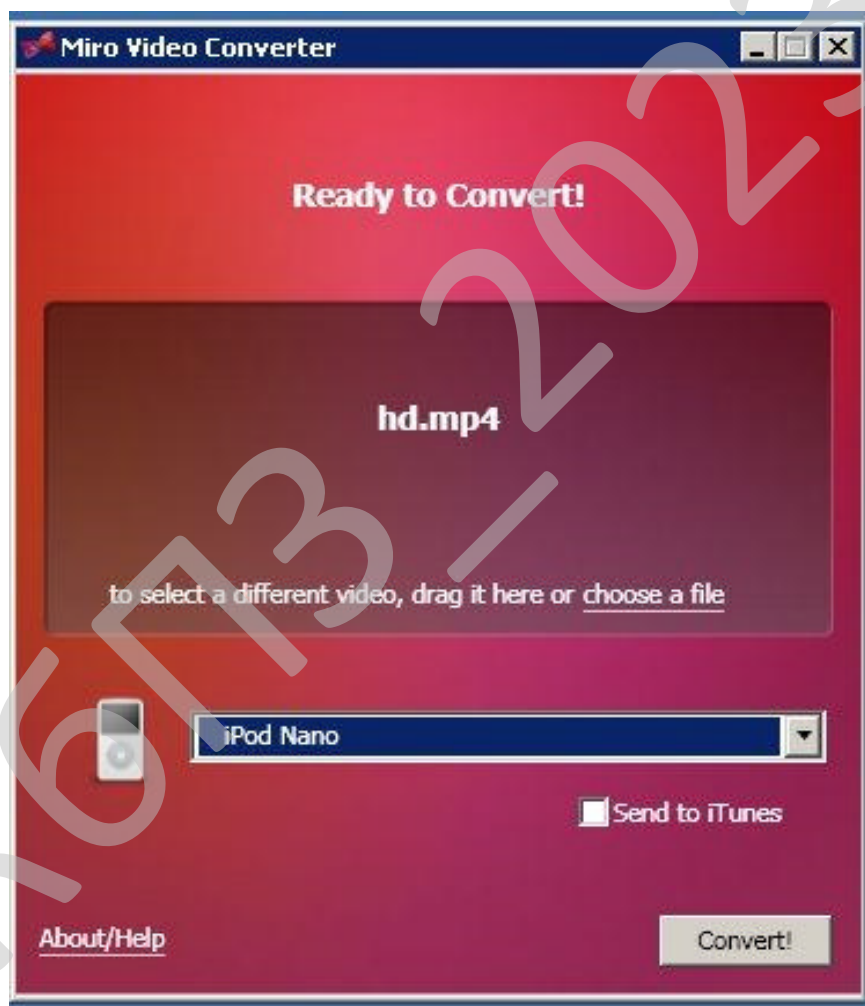


Рисунок 2.8 – Інтерфейс користувача Miro Video Converter

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

- Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.
- Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.
- Відладник Win 64 (на LLDB) і збирач для C++.
- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.
- Підтримка Metal Driver GPU для macOS і iOS.
- Вбудований Fmxlinux.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).
- Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.
- Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services
- У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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

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

Істотне поліпшення Delphi Code Insight

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

Delphi Custom Managed Records

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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 компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розмір відеофайлу може бути складним, і концепція зменшення розміру відеофайлу пов'язана з багатьма технічними змінними. Ви можете зменшити розмір файлу, щоб легше надсилати його колегам, або швидше завершити завантаження на YouTube. Може бути важко зрозуміти, з чого почати, намагаючись зменшити розмір файлу. Нижче ми зібрали корисну інформацію, яка допоможе вам розпочати подорож зі зменшення файлів!

Формат відео та тип файлу

Якщо ви вперше знайомі з цими поняттями, будь ласка, не припиняйте читати лише тому, що ми використовуємо багато жаргону, ми спробуємо пояснити кожному, і, перейшовши за відповідними посиланнями, ви також можете краще зрозуміти кожному з них.

Є багато різних відеоформатів, у які можна експортувати готове відео. Щоб допомогти вам зрозуміти відеокодеки та контейнери, ми написали докладну публікацію про відмінності типів відеофайлів . Кодеки кодують або стискають потоки даних для зберігання, відтворення та редагування відео на різних платформах. Дізнайтеся більше про кодеки тут. Можливо, ви чули про кодек h.264 (можливо, не чули, але тримайтеся) . Цей кодек створює MP4 , який зазвичай забезпечує найкращу якість із найменшим розміром файлу. Через це MP4 дуже популярний для веб-доставки, включаючи YouTube, Facebook, Twitter та Instagram. Щоб допомогти вам не надто турбуватися про розмір файлу, ми створили відеовиходи Camtasia та Snagit, які за замовчуванням використовують тип файлу MP4, і в більшості випадків рекомендуватимуть MP4 для обміну в Інтернеті.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Серед інших поширених типів відеофайлів:

- .MOV (Apple Quicktime Movie).
- .AVI (чергування аудіо і відео – Microsoft).
- .WMV (Windows Media Video – Microsoft).
- .FLV (формат Flash Video).

І багато іншого! Ви можете дізнатися все про різні типи відеофайлів, перераховані вище, прочитавши цю статтю .

Висновок: використовуйте тип файлу MP4, щоб зберегти якість і зменшити розмір файлу.

Розміри відео (у пікселях)

Роздільна здатність відео визначає розмір у висоту та ширину, у якому буде створено відео. Роздільна здатність визначається кількістю горизонтальних ліній відео зверху вниз. Відео більшої роздільної здатності значно збільшить розмір файлу. Як ви можете собі уявити, відео 1920 x 1080 (1080p) матиме більший розмір файлу, ніж відео 1280 x 720 (720p). Відео 1080p має більше горизонтальних ліній (більше інформації), що робить розмір відеофайлу більшим. Деякі записи екрану, зроблені на дисплеях з високою щільністю, можуть навіть створювати відео з роздільністю 3840 × 2160 пікселів або вище (4k). Дисплеї високої щільності мають *мільйони* пікселів і тисячі горизонтальних ліній; ще більше інформації! *Не записуйте весь екран* , якщо це дійсно необхідно, або подумайте про використання масштабування та панорамування під час редагування відео. Таким чином ви можете зменшити масштаб відео, але все одно зосередити увагу глядача на певній частині екрана. У Camtasia ви можете змінити налаштування проекту, щоб створювати відео з бажаною роздільною здатністю.

Висновок: екпортуйте з відповідною роздільною здатністю для вашого проекту. Експорт у 720p або 1080p буде працювати для більшості людей.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Скільки дійства у вашому відео?

Якщо у вас багато переходів (витирання, розмиття, анімація) разом із зображеннями, що швидко розвиваються, ваш файл буде більшим. Наприклад, запис досить статичної презентації слайдів (менше дій) призведе до меншого розміру файлу, ніж запис деякого живого відео на екрані (більше дій). Використання відеоформату MP4 також допоможе зменшити деякі проблеми з розміром файлу, пов'язані з переміщенням, оскільки MP4 дуже ефективний у цьому відношенні.

Висновок: додавання великої кількості рухів і переходів призведе до того, що ваш файл стане більшим.

Аудіо

Іноді аудіо може створювати файли більшого розміру, хоча в більшості випадків пристрої для запису та програмне забезпечення стискають аудіо, щоб не створювати файли великих розмірів. Крім того, дуже рідко коли нестиснене аудіо буде чути для ваших глядачів. Після завершення редагування відео обов'язково експортуйте його зі стислим звуком. Виходи за замовчуванням для форматів MP4 Camtasia та Snagit використовують стиснення аудіо AAC, яке забезпечує хороше поєднання якості звуку та розміру файлу.

Висновок: стискайте аудіо під час експорту. У більшості проектів рідко коли потрібен нестиснений звук. Ознайомтеся з нашими найкращими методами роботи з аудіо.

Частота кадрів (FPS)

Що таке частота кадрів? Радий, що запитав! Щоб отримати детальний опис, перегляньте посібник для початківців щодо частоти кадрів. Коротка відповідь полягає в тому, що частота кадрів вказує, скільки кадрів (думаю: зображень) показується глядачеві за секунду. Більшість фільмів знімають зі швидкістю 24 кадри в секунду. Домашнє відео та відеокамери зазвичай мають швидкість 30 кадрів/с. Що краще? Це суб'єктивне запитання, яке залежить від цільової аудиторії. Ви прагнете виглядати відео чи кінематографічно? Спеціальні

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

налаштування виробництва в Camtasia дозволяють від 1 до 30 кадрів на секунду (fps). Хоча за замовчуванням 30 кадрів в секунду є стандартними для відео, якщо ви хочете отримати більше кінематографічного, можете знизити її до 24 кадрів в секунду. Вища частота кадрів часто створює більший розмір файлу під час експорту.

Висновок: використовуйте частоту кадрів, яка підходить для вашого проекту. Пам'ятайте, що більша частота кадрів призведе до дещо більшого розміру відео.

3.2 Розробка структурної схеми

Ця робота представляє Deep4D – компактне генеративне представлення форми та зовнішнього вигляду із знятих 4D об'ємних відеорядів людей. Об'ємне 4D-відео забезпечує високореалістичне відтворення, відтворення та рендеринг вільної точки зору актора з систем збору відео з кількома видами. Глибока генеративна мережа тренується на 4D відеопослідовностях актора, який виконує кілька рухів, щоб вивчити генеративну модель динамічної форми та зовнішнього вигляду. Ми демонструємо, що запропонована генеративна модель може забезпечити компактне закодоване представлення, здатне до високоякісного синтезу 4D об'ємного відео зі стисненням на два порядки величини. Варіаційна мережа кодера-декодера використовується для вивчення закодованого латентного простору, який відображає 3D-позу скелета в 4D-форму та зовнішній вигляд. Це дозволяє синтезувати високоякісний об'ємний 4D-відео за допомогою скелетного руху, включаючи дані захоплення скелетного руху. Цей закодований латентний простір підтримує представлення кількох послідовностей із динамічною інтерполяцією для переходу між рухами. Тому ми представляємо графіки руху Deep4D, пряме застосування запропонованого генеративного представлення. Графіки руху Deep4D дозволяють створювати інтерактивну анімацію персонажів у реальному часі, зберігаючи правдоподібну реалістичність руху та зовнішнього

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

чітких 3D-моделей, які складаються з сіток і текстур, і анімуються за допомогою спеціальних моделей руху для імітації людської поведінки та діяльності.

Останні роботи показали, що можна навчитися та анімувати природну людську поведінку (наприклад, ходьбу, стрибки тощо) на основі даних захоплення рухів людського скелета (MoCap) акторської гри. З іншого боку, проектування реалістичної 3D-моделі людини все ще є трудомістким процесом. Враховуючи величезний успіх глибоких генеративних моделей, виникає запитання, чому б і ні також навчитися генерувати реалістичне зображення людини? Обумовлюючи процес створення зображення генеративної моделі додатковими вхідними даними, вивчаються відображення між різними областями даних, які, наприклад, дозволяють контролювати та маніпулювати формою об'єкта, перетворюючи ескізи на зображення, а зображення – на картини. Нещодавно генеративні методи покращили роздільну здатність і якість отриманих зображень. Проте генератори продовжують працювати як чорні скриньки, і, незважаючи на останні зусилля, розуміння різних аспектів процесу синтезу зображення невідоме. Властивості латентного простору також погано вивчені, а широко продемонстрована інтерполяція латентного простору не забезпечує кількісного способу порівняння різних генераторів один з одним. Керуючись останніми досягненнями в генеративних мережах, ми пропонуємо архітектуру для навчання генерації динамічної 4D-форми та зовнішнього вигляду з високою роздільною здатністю, яка розкриває способи керування зображенням синтез. Наш генератор зовнішнього вигляду починається з вивченого простору руху та регулює роздільну здатність зображення на кожному шарі згортки на основі коду прихованого руху, таким чином безпосередньо контролюючи силу характеристик зображення в різних масштабах.

Ця робота пропонує Deep4D, глибоке генеративне представлення динамічної форми та зовнішнього вигляду з 4D об'ємного відео людського характеру. Запропонований підхід вивчає ефективне представлення стисненого латентного простору та генеративну модель із 4D об'ємних відеопослідовностей

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

людини, яка виконує кілька рухів. Компактне представлення прихованого простору досягається за допомогою варіаційного кодера-декодера для вивчення відображення 3D-руху скелета до відповідної повної 4D-об'ємної форми, руху та зовнішнього вигляду. Закодований прихований простір підтримує інтерполяцію динамічної форми та зовнішнього вигляду для плавного переходу між знятими 4D об'ємними відеопослідовностями. У цій роботі представлені графіки руху Deer4D, які використовують генеративне представлення кількох 4D-об'ємних відеопослідовностей у вивченому латентному просторі для забезпечення інтерактивної анімації з оптимальним переходом між рухами. Основні нові внески цієї роботи:

– Deer4D, генеративне представлення форми та зовнішнього вигляду для 4D об'ємного відео, яке забезпечує компактне зберігання та інтерактивну анімацію в реальному часі.

– Відображення руху скелета у 4D-об'ємному відео для синтезу динамічної форми та зовнішнього вигляду.

– Графіки руху Deer4D, анімаційна основа, створена на основі представлення Deer4D, яка дозволяє використовувати високорівневі 4D-символи, уможливаючи синтез нових рухів і взаємодію з користувачем у реальному часі.

Пов'язана робота

4D об'ємне відео: було активною областю досліджень, яка виникла для вирішення проблеми зростання попиту на реалістичний зміст діяльності людини. Нещодавно представили повний конвеєр для захоплення, реконструкції та відтворення високоякісного об'ємного відео. Система використовує приблизно 100 синхронізованих камер, які одночасно знімають об'єм з кількох точок зору. Об'ємне відео фіксує динамічну геометрію поверхні та фотореалістичний вигляд об'єкта. Це розкриває величезний творчий потенціал для створення дуже реалістичного анімаційного контенту на основі відзнятих виступів. Нещодавні дослідження пропонують механізми для полегшення маніпулювання цим вмістом, що дозволяє художнику виконувати ручні коригування 4D-динамічної

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

геометрії та об'єднувати кілька послідовностей у моушн-графі. Однак використання об'ємного 4D-відео у виробництві контенту залишається обмеженим через складність маніпуляцій, анімації та рендерингу послідовностей форм, зберігаючи при цьому реалістичність зовнішнього вигляду та динаміку одягу.

Вивчені представлення послідовності сітки: Теджера та Хілтон (2013) запропонували техніку редагування послідовності просторово-часової сітки на основі частин, яка вивчає моделі деформації поверхні в лапласівських координатах. Цей підхід обмежує деформацію сітки правдоподібними формами поверхні, отриманими з набору прикладів. Часткове навчання деформації поверхні дозволяє локально маніпулювати сіткою та досягає більшої гнучкості анімації, дозволяючи генерувати нові створені сітки. Тан та ін. (2018) використовують варіаційний автокодер (VAE), щоб навчитися представленню параметризованих динамічних форм. Їхня мережа тренується на попередньо обробленому просторі ознак навчальних даних, демонструючи дуже низьку похибку реконструкції для наземних форм правдивості. Ломбарді та ін. (2018) запропонував навчену модель форми та зовнішнього вигляду, залежну від точки огляду, що дозволяє відновлювати деталі текстури, залежні від виду. Ця мережа демонструє здатність вивчати тривимірні динамічні форми з вершин, уникаючи необхідності попередньої обробки інформації. Це демонструє можливості VAE у реальному часі, здатні декодувати форму та зовнішній вигляд менш ніж за 5 мілісекунд. Нещодавно Regateiro et al. (2019) продемонстрували можливості вивчення 3D-динамічних форм для створення реалістичної анімації за допомогою VAE для вивчення геометричного простору людського персонажа та повторного використання декодера в режимі реального часу для синтезу 3D-геометрії.

Навчене представлення зовнішнього вигляду: Нещодавно Esser et al. (2019) представив підхід до цілісної навчальної основи для відтворення людської поведінки, навченої на основі даних захоплення рухів скелета для реалістичного контролю та відтворення. Вони вивчають відображення від абстрактного

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

представлення пози до цільових зображень, обумовлених латентним представленням VAE для зовнішнього вигляду. Karras та ін. (2017) пропонують нову методологію навчання для генеративних мереж, який поступово збільшує як генератор, так і дискримінатор, починаючи з низької роздільної здатності зображення і закінчуючи початковою роздільною здатністю зображення. Вони демонструють, що модель все більше вивчає дрібні деталі в ході навчання, отже покращуючи швидкість і стабільність навчання та створюючи високоякісні зображення. Хоча фотореалізм є проблемою, яку важко вирішити, цей підхід є кроком до відтворення високоякісних зображень, які неможливо відрізнити від реальних зображень. Зовсім недавно Karras et al. (2018) перевизначають архітектуру генеративних мереж для передачі на основі стилів. Використовуючи підхід, подібний до Karras et al., 2017, вони продемонстрували високоякісні результати зображень, наприклад, можливість дізнатися точне розташування волосся, щетини, веснянок або пори шкіри. Це демонструє потенціал для синтезу зображень людей із високою роздільною здатністю, зберігаючи при цьому природні деталі, необхідні для сприйняття реалізму.

4D об'ємна відеоанімація: графіки руху для анімації персонажів із послідовностей захоплення руху скелета використовують представлення структурованого графіка для забезпечення інтерактивного керування. Графіки скелетних рухів будуються за допомогою покадрової метрики подібності, яка визначає подібні пози та рух. Концепція графіків руху була застосована до об'ємного відео з використанням як неструктурованих сіток, так і тимчасово узгоджених структурованих сіток. Початкові підходи об'єднують неструктуровані динамічні послідовності сітки без часової узгодженості зв'язності сітки на основі подібності форми та руху. Прада та ін. (2016) замість цього виконує вирівнювання сітки та текстури у визначених точках переходу, щоб забезпечити плавне змішування. Це долає складну проблему глобального вирівнювання сітки та розглядає лише вирівнювання геометрії та текстури, де це необхідно. На відміну від цього, Boukhauma і Boyer (2017) і Casas et al. (2014)

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

використовували глобальне вирівнювання послідовності сітки для отримання узгодженого в часі з'єднання сітки з об'ємного відео. Це дозволяє створювати 4D графіки руху зі змішуванням сітки для високорівневого параметричного контролю руху та плавних переходів між рухами.

У цьому документі ми представляємо Deerp4D, навчене генеративне представлення об'ємних відеопослідовностей, представлене в Розділі 3. Deerp4D забезпечує компактне представлення, яке долає вимоги до пам'яті та обчислень попередніх підходів для явного представлення всіх захоплених послідовностей під час виконання через навчені параметри. мережі. У Розділі 4 ми представляємо графіки руху Deerp4D, пряме застосування запропонованої генеративної мережі для створення безперебійної анімації як динамічної форми, так і зовнішнього вигляду між вивченими захопленими послідовностями руху. Нарешті, у розділі 5 представлено кількісну та якісну оцінку запропонованого методу.

3 Представлення Deerp4D

Робота є кроком вперед, щоб дозволити контролювати та синтезувати 4D-об'ємне відео, зберігаючи при цьому реалістичність динамічної форми та зовнішнього вигляду. У цьому розділі представлено використання генеративної мережі для ефективного представлення 4D-об'ємного відеовмісту з даних захоплення продуктивності. Спочатку представлено попередню обробку знятого об'ємного відео у форму, придатну для нейронних мереж. Описано генеративну мережу для вивчення 4D-форми із захоплених об'ємних послідовностей разом із використанням варіаційного кодера-декодера для забезпечення відображення компактного латентного просторового представлення від 3D-скелетної пози до відповідної 4D-динамічної форми. Нарешті, ми представляємо генеративну мережу для вигляду 4D-відео, яка вчиться синтезувати вигляд динамічної текстури високої роздільної здатності з представлення компактного прихованого простору, рис. 1 . Застосування компактного латентного просторового представлення дозволяє інтерполювати між скелетними позами для створення правдоподібної проміжної форми та вигляду сітки. Ці розділи окремо описують

внесок генераторної мережі, проілюстрований на малюнку 1 . Генеративне представлення Deep4D дозволяє генерувати реалістичне відтворення людських персонажів із можливістю перенацілювання нової інформації про рух скелета.

Об'ємна попередня обробка відео

У контексті цієї роботи 4D-об'ємне відео представляє 4D-сітку на основі захоплення відео кількох переглядів..

Сучасна об'ємна зйомка людей із вільним одягом і волоссям забезпечує реконструйовану форму та текстуру з високою роздільною здатністю. Необроблене об'ємне відео зазвичай призводить до неструктурованої послідовності сітки, де як форма сітки, так і зв'язок змінюються від кадру до кадру. Було введено декілька підходів для тимчасового вирівнювання за короткими підпоследностями для стиснення вимог до зберігання або глобального вирівнювання за повними послідовностями.

У цій роботі ми використовуємо керовану скелетом структуру об'ємного вирівнювання поверхні для попередньої обробки знятих 4D-об'ємних відео людей, щоб отримати тимчасово когерентну сітчасту структуру в кількох послідовностях. Цей фреймворк отримує як вхідний сигнал синхронізоване відео з кількома переглядами від каліброваних камер і повертає 3D-скелетні суглоби та тимчасово узгоджені 3D-сітки з однаковим підключенням сітки в кожному кадрі. Зовнішній вигляд текстури отримується шляхом повторного відображення оригінальних зображень камери з кількома оглядами на узгоджену в часі 3D-сітку, що забезпечує динамічну карту текстури з узгодженими координатами для всіх захоплених кадрів. Вхідні дані для глибокої мережі, представлені в наступних розділах, складаються з центрованих 4D послідовностей сітки, узгоджених у часі, з відповідними 2D картами текстури та 3D розташуванням скелетних суглобів.

Цей розділ містить математичний опис і дослідження запропонованого способу кодування "гнучкою" розрядною сіткою і способів адаптації кодування як для серій із обмеженої, так й із необмеженою довжиною; теоретичну оцінку

ефективності і порівняльний аналіз способів адаптації та способу кодування: існуючої – фіксованої та розробленої – "гнучкою" розрядними сітками.

У залежності від виду серії застосовується відповідне кодування. Для серій із обмеженою довжиною використовується тільки кодування фіксованою розрядною сіткою. Для кодування серій із необмеженою довжиною застосовуються як фіксована, так і "гнучка" розрядні сітки.

Таким чином, характеристики відеоданих, представлених у цифровому вигляді, змінюються у залежності від обраного виду довжини серії і методики кодування.

Щоб вибрати найбільш ефективний варіант поєднання виду довжини серії і засобу кодування, проводиться теоретична оцінка змін характеристик відеоданих.

Оцінка і порівняння проводилися за такими характеристиками відеоданих як:

- об'єм;
- коефіцієнти стиснення;
- ефективності кодування.

Зазначені характеристики для кодування фіксованою розрядною сіткою розраховувалися за такими аналітичними виразами:

$$W_S^\infty = N \cdot p \cdot G_S;$$

$$K_S^\infty = \frac{G_\Pi}{p \cdot G_S};$$

$$\Theta_S^\infty = \frac{H_L^\infty(S)}{G_L^\infty};$$

$$W_S^C = \frac{N \cdot p}{1 - q^C} \cdot G_S;$$

$$K_S^C = \frac{(1 - q^C) \cdot G_\Pi}{p \cdot G_S};$$

$$\Theta_S^C = \frac{(1 - q^{C-1}) \cdot H_L^\infty(S)}{G_L^C},$$

де

W_S^∞ , K_S^∞ , Θ_S^∞ – об'єм, коефіцієнти стиснення й ефективності кодування для відеоданих поданих серіями із необмеженою довжиною;

W_S^C , K_S^C , Θ_S^C – об'єм, коефіцієнти стиснення й ефективності кодування для відеоданих поданих серіями із обмеженою довжиною;

G_S – розрядність кодового слова, необхідна для запису параметрів серії;

G_Π – розрядність кодового слова, яка відводиться на кодування параметру візуалізації;

G_L^C – розрядність кодового слова, яка відводиться на кодування серії із обмеженою довжиною;

$H_L^\infty(S)$ – ентропія довжини серії із необмеженою довжиною.

При кодуванні "гнучкою" розрядною сіткою W_S^D , K_S^D , Θ_S^D – об'єм, коефіцієнти стиснення й ефективності кодування дорівнюють:

$$W_S^D = N_S \cdot \left[(1 - q^{D_1}) \cdot G_{S_1}^D + \sum_{r=2}^{N_D} N \cdot p \cdot (q^{D_{r-1}} - q^{D_r}) \cdot G_{S_r}^D \right];$$

$$K_S^D = \frac{G_\Pi}{p \cdot \left[(1 - q^{D_1}) \cdot G_{S_1}^D + \sum_{r=2}^{N_D} (q^{D_{r-1}} - q^{D_r}) \cdot G_{S_r}^D \right]};$$

$$\Theta_S^D = \frac{H^\infty(S)}{(1 - q^{D_1}) \cdot G_{S_1}^D + \sum_{r=2}^{N_D} (q^{D_{r-1}} - q^{D_r}) \cdot G_{S_r}^D},$$

де

N_S – кількість серій;

D_1, D_r, D_{r-1} – інтервали кодових груп за довжиною серій;

$G_{S_1}^D, G_{S_r}^D$ – розрядність кодового слова, що необхідна для запису параметрів серії, які потрапляють у відповідну кодову групу;

$H^\infty(S)$ – ентропія серії із необмеженою довжиною.

Чисельні розрахунки характеристик відеоданих показали, що необхідно розробити способи адаптації кодування під тип зображення.

При кодуванні серій із обмеженою довжиною максимальні значення коефіцієнта ефективного кодування отримані для способу, що використовує ентропію довжини серії із необмеженою довжиною.

При кодуванні "гнучкою" розрядною сіткою найкращі показники коефіцієнта ефективного кодування мають два способи, які використовують ентропію серії із необмеженою довжиною та розрядність кодового слова, необхідну для кодування математичного чекання довжини необмеженої серії.

Для порівняння використовуються два варіанти сполучення довжин у послідовності серій: найкращий і найгірший.

Передбачається, що все сімейство залежностей характеристик відеоданих знаходиться в області, обмеженої залежностями, отриманими для найкращого і найгіршого варіантів.

Дослідження залежностей об'ємів і коефіцієнтів стиснення відеоданих показує, що мінімальний об'єм і найбільший коефіцієнт стиснення відеоданих буде для найкращого варіанту при кодуванні серій із необмеженими довжинами фіксованою розрядною сіткою.

Цей факт дозволив спростити дослідження характеристик відеоданих тільки для найгіршого варіанту сполучення довжин у послідовності серій.

Аналіз отриманих результатів показав, що для представлення зображень варто використовувати серії із необмеженою довжиною, які кодуються фіксованою або "гнучкою" розрядною сіткою в залежності від сполучення довжин у послідовності серій.

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру системи передачі відеоданих з урахуванням запропонованого у роботі відеокодека на основі компактного представлення відеоданих.

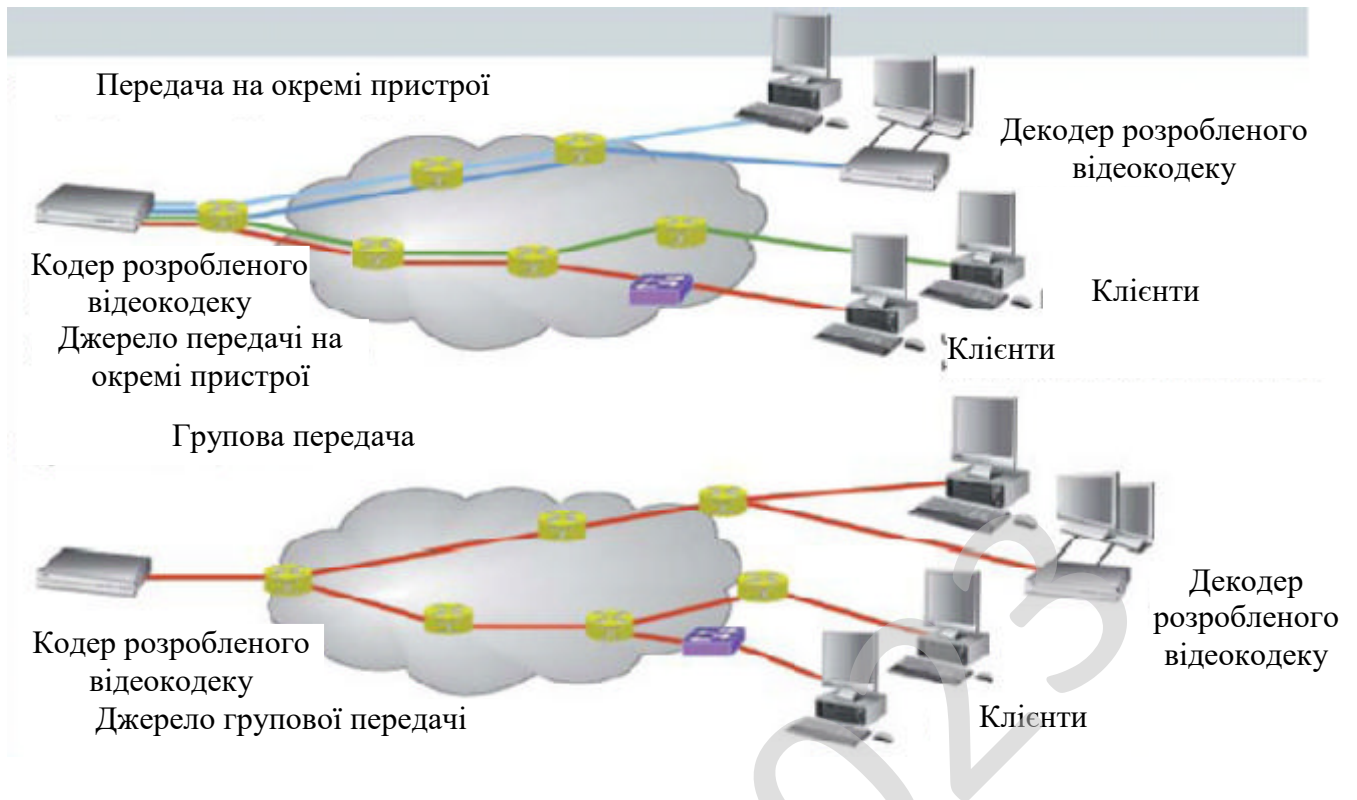


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Розробимо метод представлення відеоданих з виділенням серій максимальної довжини, який використовує серії із необмеженою довжиною, що кодуються "гнучкою" розрядною сіткою.

Алгоритм стиснення методом із виділенням серій максимальної довжини реалізується у три етапи:

– На першому етапі для статичних і динамічних зображень використовуються розроблені алгоритми формування серій із необмеженими довжинами, що дозволяють усунути інформаційну надмірність у послідовності параметрів візуалізації та забезпечити мінімальний час формування відеоданих;

– На другому етапі – алгоритм кодування серій. Під час кодування вихідна послідовність серій перетвориться в дві кодові групи: "сплесків" й "залишків". Для визначення умови, що дозволяє помістити серію в одну з груп, використовується розрядність подвійного слова, яка необхідна для кодування

значення середньоарифметичної довжини послідовності серій. У процесі кодування також формується послідовність параметрів повернення.

– На третьому етапі визначена кількість рівнів перетворення. Воно дорівнює двом, тому що в цьому випадку кількість кодових груп не перевищить 4.

Алгоритм відновлення реалізується у два етапи:

– На першому етапі працює розроблений алгоритм декодування довжин серій та зворотного перетворення послідовностей: "сплесків", "залишків" і параметрів повернення – у вихідну послідовність довжин серій.

– На другому етапі використовується запропонований алгоритм відновлення відеоданих представлених довжинами серій.

Для збереження та передачі стиснених відеоданих розроблено формат представлення даних, який складається з трьох частин:

- інформація про файл;
- інформація про відеодані та метод стиснення;
- послідовність відеоданих, яка складається з двох груп: послідовність параметрів візуалізації і послідовність довжин серій.

Все це відображено на функціональній схемі, зображеній на рисунку 3.2.

Зробимо оцінку ефективності методів представлення відеоінформації, аналізується завадостійкість відеоданих, записаних довжинами серій, розглянемо питання технічної реалізації методу стиснення із виділенням серій максимальної довжини.

Порівняльна оцінка ефективності представлення зображень ІКМ і розробленими способами проведена за такими характеристиками як:

- середній об'єм відеоданих;
- час формування зображення;
- потік або швидкість передачі відеоданих.

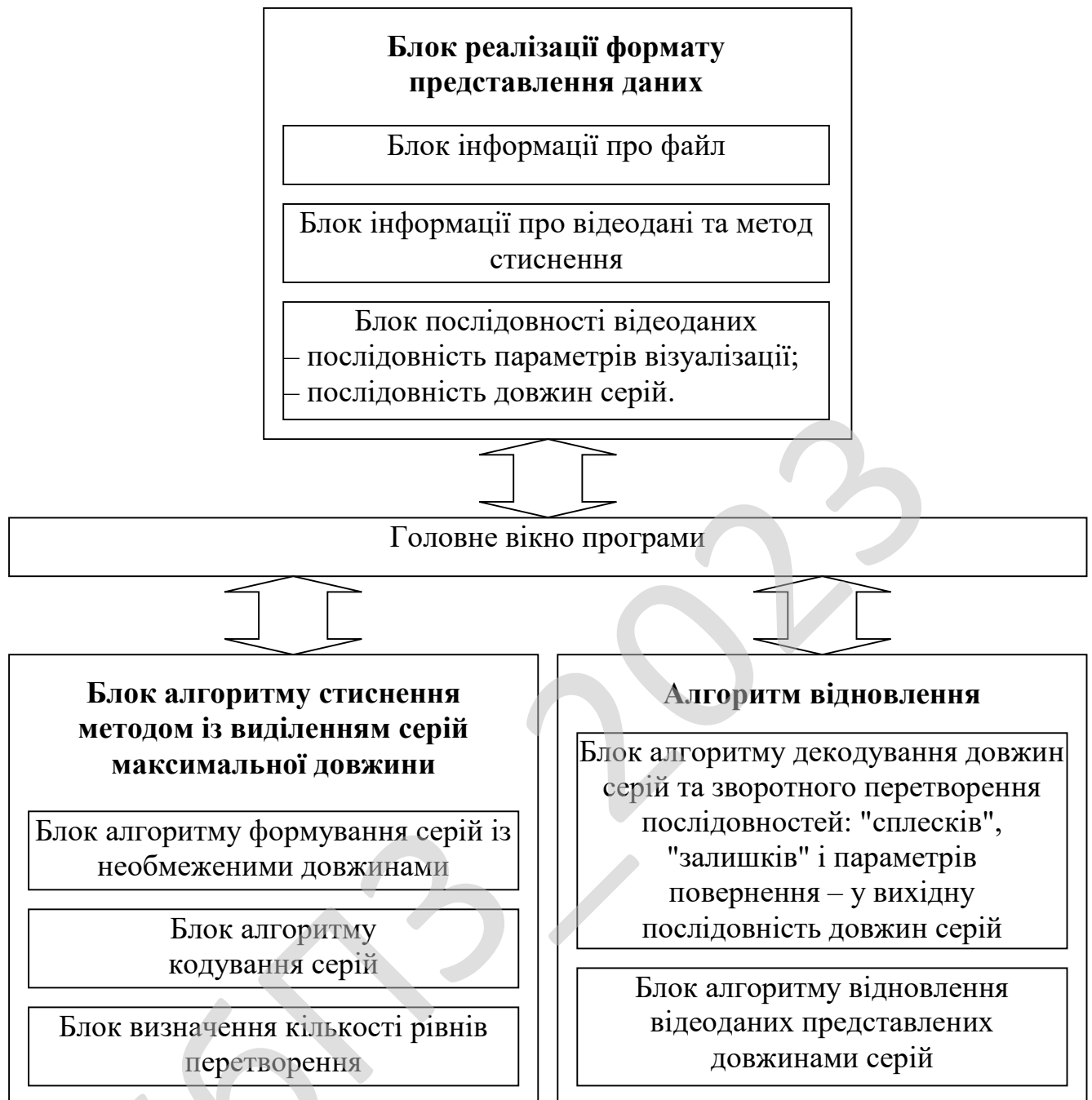


Рисунок 3.2 – Функціональна схема системи

Порівняльний аналіз показав, що застосування методу довжин серій, який використовує запропоновані способи адаптації, дозволяє зменшити середній об'єм відеоданих, що припадає на один елемент зображення, скоротити час формування зображень та зменшити потік відеоданих. Це дозволить без

додаткових витрат на підвищення швидкодії відеопідсистеми поліпшити якість зображення та частково звільнити ресурси для виконання інших задач.

Проведемо аналіз завадостійкості відеоданих, представлених довжинами серій. Він дозволив оцінити викривлення інформації при передачі зображень по дискретному каналу зв'язку з біноміальним розподілом імовірності помилок.

Для оцінки завадостійкості запропонований імовірнісний розрахунок, що використовує ефект "розмноження помилок", який дозволив перекласти аналіз завадостійкості кодограм з різними вагами символів у площину аналізу рівноважних кодограм. Це істотно спростило оцінку завадостійкості. Запропонований підхід можна застосувати і при розрахунку завадостійкості нерівноважних кодограм із надмірністю, тобто закодованих (n, k) -кодом, який має завадостійкість.

При оцінці завадостійкості було помічено, що виникнення однієї або більше помилок будь-яких комбінацій у нерівноважній кодограмі з приблизною межею довжини $n \geq 4$ (при використанні прийнятого ступеневого інформаційного критерію оцінки значення нерівноважних символів) еквівалентно ефекту поразки усіх біт кодограми, тобто абсолютній поразці кодограми у цілому; відомі коригувальні коди малоефективні для захисту нерівноважних блоків, тому для них варто використовувати спеціальні коди з пріоритетним нерівним захистом символів.

Розглянемо питання технічної реалізації представлення зображень. Для обробки відеоданих запропонована структурна схема із незалежною пам'яттю регенерації, у якій пам'ять регенерації є незалежним накопичувачем, а перетворення кодів графічних елементів та координат здійснюється спеціальною апаратною логікою, пов'язаною з мікро-ЕОМ.

За суттю, пристрій стиснення відеоданих уявляє собою спец-ЕОМ, яка призначена для виконання алгоритмів компресії і декомпресії зображення. Така структура відрізняється підвищеною складністю, але забезпечує найбільшу швидкість перетворення і редагування даних.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

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

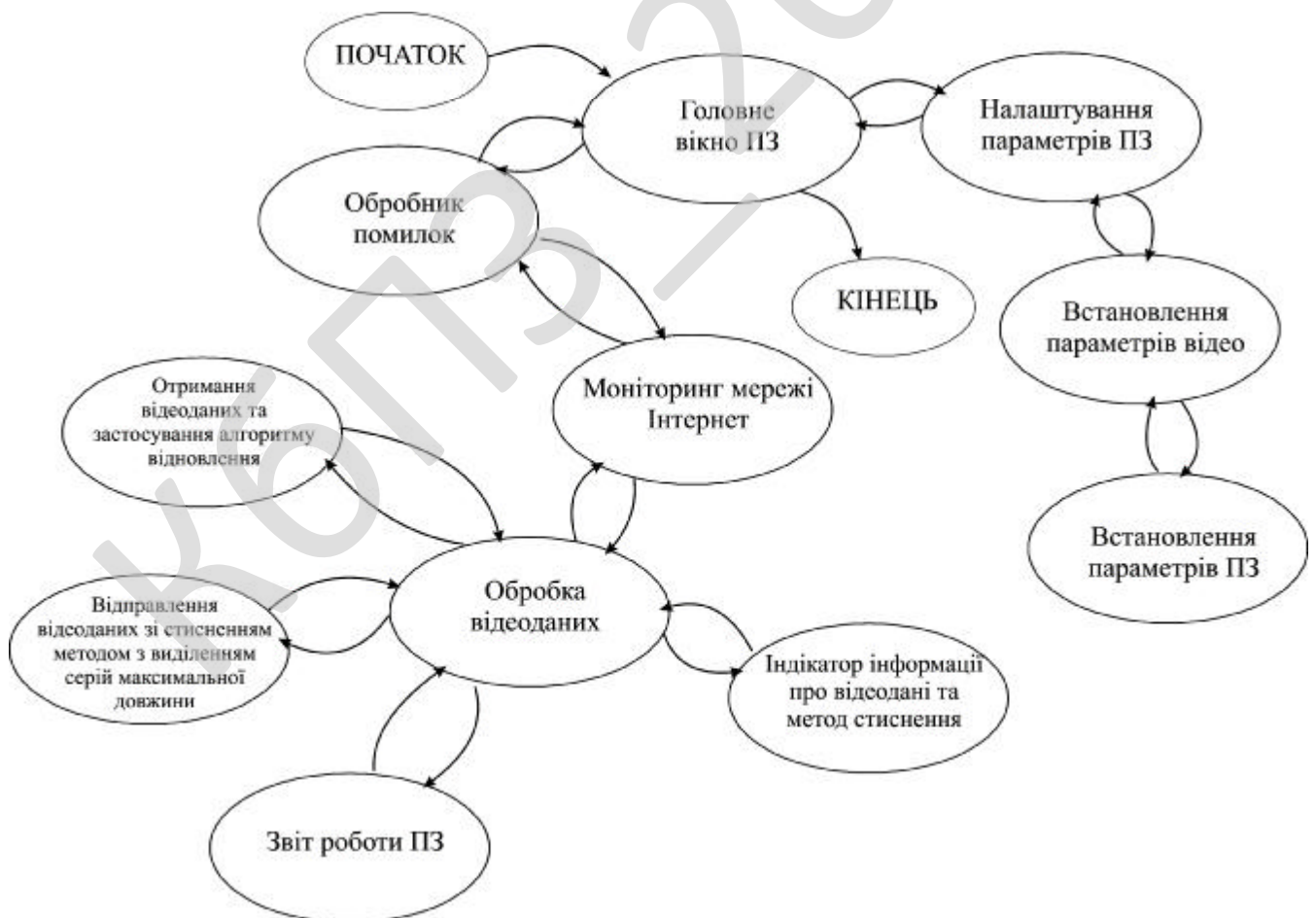


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБГПЗ - 2023

					VKPM-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

4 РЕАЛІЗАЦІЯ РОБОТИ. 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 та 4.3 зображено роботу підпрограм.

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

Опис алгоритмів функціонування системи

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 було враховано, 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 компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

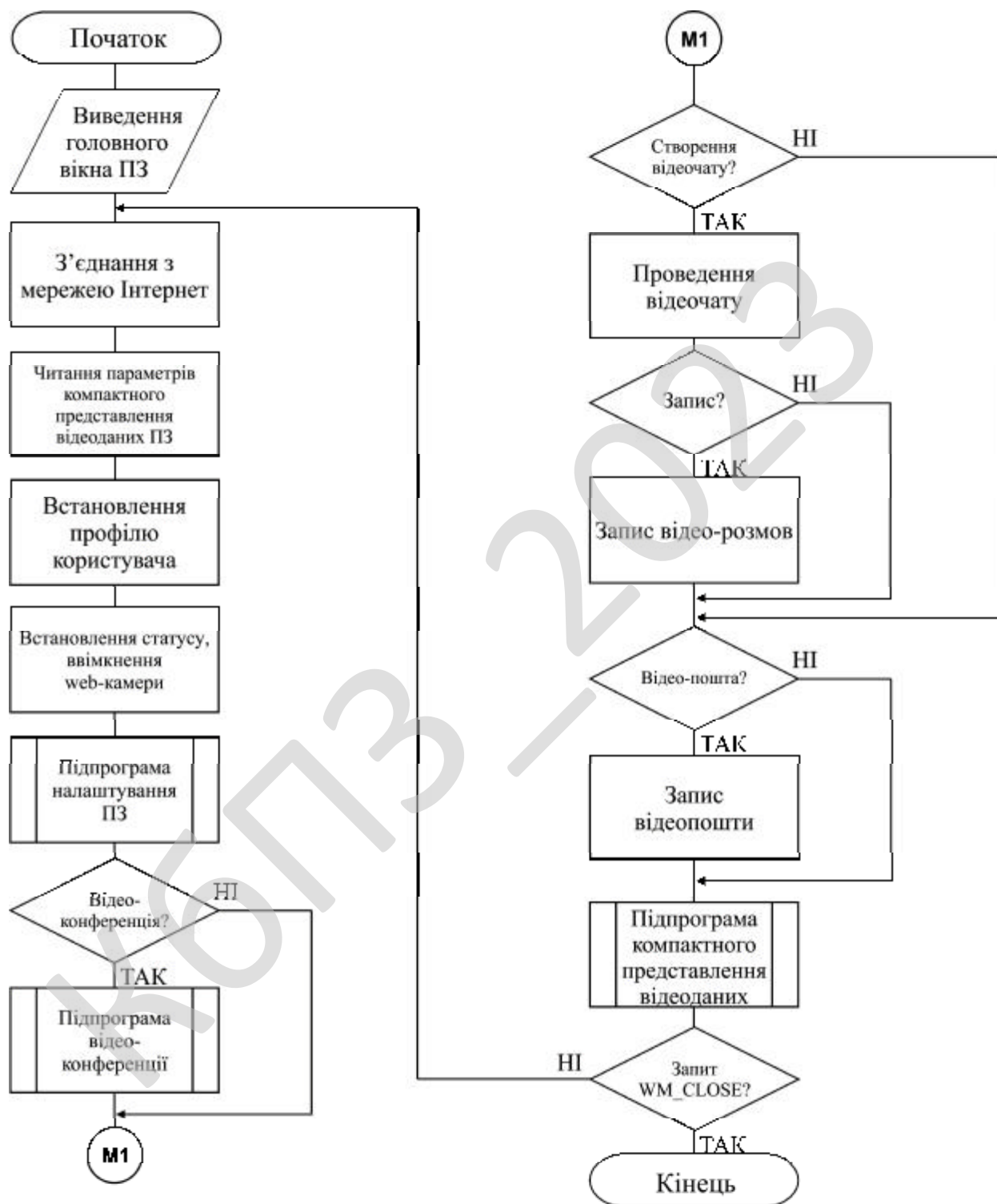


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

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.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Після цього встановити властивості, для SampleGrabber властивості: FilterGraph рівним CaptureGraph, у компонента CaptureGraph встановлюємо Mode рівним gmCapture і GraphEdit рівним True.

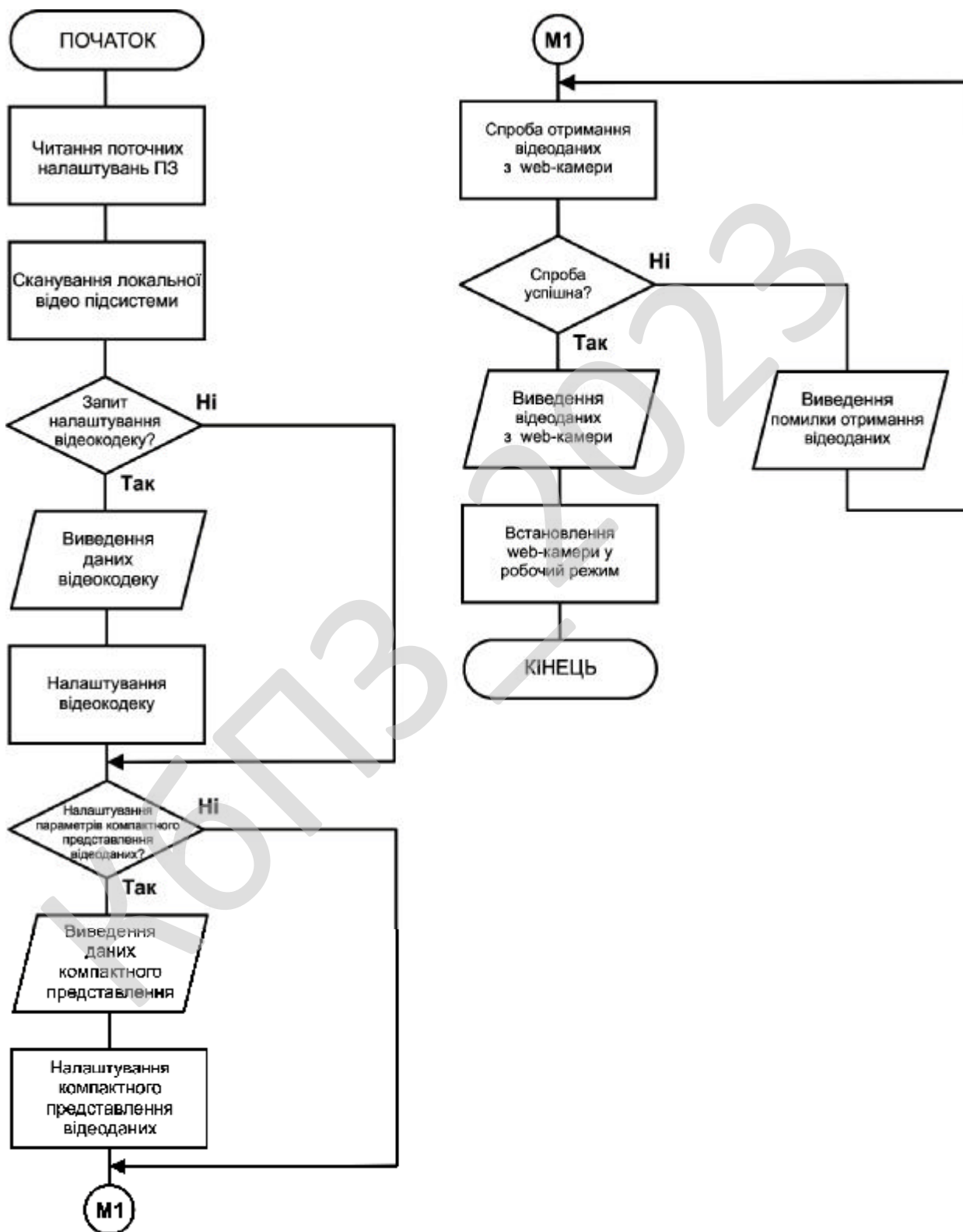


Рисунок 4.3 – Блок-схема роботи підпрограми

Далі на форму кидаємо компонент VideoWindow2 класи TVideoWindow, для нього встановлюємо значення FilterGraph рівним CaptureGraph, і задаємо властивість Align=alClient. Це властиво наше вікно відображення, у якому можна буде побачити події з web – камери.

Далі на оброблювач onCreate нашої форми ставимо код

```
CompFilter := TFilterList.Create;
CapFilters := TSysDevEnum.create(CLSID_VideoCompressorCategory);
CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
VideoMediaTypes := TEnumMediaType.Create;
```

Потім кидаємо на нашу форму кнопку або будь-який інший об'єкт керування й на оброблювач onClick пишемо наступний код:

```
VideoWindow2.FilterGraph:=CaptureGraph;
CapEnum:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
CaptureGraph.ClearGraph;
CaptureGraph.Active := false;
VideoSourceFilter.BaseFilter.Moniker := CapEnum.GetMoniker(0);
VideoSourceFilter.FilterGraph := CaptureGraph;
CaptureGraph.Active := true;
SetVideoParams(CaptureGraph as ICaptureGraphBuilder2,
PIN_CATEGORY_CAPTURE, VideoSourceFilter as IBaseFilter);
with CaptureGraph as ICaptureGraphBuilder2 do
RenderStream(@PIN_CATEGORY_PREVIEW, nil, VideoSourceFilter as IBaseFilter,
SampleGrabber as IBaseFilter, VideoWindow2 as IbaseFilter);
CaptureGraph.Play;
```

Функція, що служить для установки параметрів захвата відеопотоку, дозволяє регулювати такі параметри як частота відеозйомки, розрішення картинки, і кількість біт на піксел.

```
function SetVideoParams(CB_B2: ICaptureGraphBuilder2; Category: TGUID; fSource:
IBaseFilter): HRESULT;
var
    StreamConf: IAMStreamConfig;
    PAMT: PAMMediaType;
begin
    Result:= E_FAIL;
    StreamConf:= nil;
    PAMT:= nil;
    try
```

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```

Result:= CB_B2.FindInterface(@Category, @MEDIATYPE_Video, fSource,
IID_IAMStreamConfig, StreamConf);
if Assigned(StreamConf) then
begin
    StreamConf.GetFormat(PAMT);
    if Assigned(PAMT) then
    begin
        if PAMT.cbFormat= sizeof(TVideoInfoHeader) then
        begin
PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biWidth:=640;
PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biHeight:= 480;
PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biBitCount:=24;
PVIDEOINFOHEADER(PAMT^.pbFormat)^.AvgTimePerFrame:= 1000000 div 25;
with PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader do
PAMT^.lSampleSize :=((biWidth + 3) and (not (3))) * biHeight * biBitCount
shr 3;
PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biSizeImage:=PAMT^.lSampleSize;
            end;
            Result:= StreamConf.SetFormat(PAMT^);
        end;
    end;
    result:= S_OK;
except
on E: Exception do
    MessageBox(0, PChar(E.Message), '', MB_OK or MB_ICONERROR);
end;
StreamConf:= nil;
if Assigned(PAMT) then
    DeleteMediaType(PAMT);
end;

```

Отже тепер можете запустити програму й натиснути на кнопку. Тепер якщо у все правильно, то повинні з'явитися зображення з камери.

Далі розглядаємо приклад згравлювання картинки, він досить простий. На оброблювач події OnTimer компонента Timer пишемо от такий код

```

procedure T2016.TimerTimer(Sender: TObject);
begin
    SampleGrabber.GetBitmap(SendBt);
    SendBitmap(FindWindow(nil, PChar('Server/Client')), SendBt);
end;

```

Та дописуємо в глобальні змінні такий от запис:

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52


```

    if hTargetWnd <> 0 then
SendMessage(hTargetWnd, WM_COPYDATA, Longint(Handle), Longint(@ACopyDataStruct))
    else ShowMessage('Заданий Handle не знайдений');
end;

```

Ціль цих функцій перетворення бітової матриці у формат JPG, після чого цей потік ми перенаправляємо програмі з назвою Server/Client. Цілком логічним буде те, що й від цієї програми ми й будемо одержувати дані, так що функція прийому даних буде виглядати так:

```

procedure WMCopyData(var Msg: TWMCopyData); message WM_COPYDATA;
// ця функція оголошується у секції private.

procedure T2016.WMCopyData(var Msg: TWMCopyData);
var
    sText: array[0..99] of Char;
    ms: TMemoryStream;
    JpegIm: TJpegImage;
begin
    case Msg.CopyDataStruct.dwData of
    0: { текст }
        begin
StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
        end;
    1: { картинка }
        begin
            ms := TMemoryStream.Create;
            JpegIm := TJpegImage.Create;
            try
                with Msg.CopyDataStruct^ do
                    ms.Write(lpdata^, cbdata);
                    ms.Position := 0;
                    JpegIm.LoadFromStream(ms);
                    try
                        Height:=JpegIm.Height+50;
                        Width:=JpegIm.Width;
                        Image.Picture.Graphic.Assign(JpegIm);
                    end;
                except
                end;
            finally
                ms.Free;
            end;
        end;
    end;
end;

```

```

        JpegIm.Free;
        end;
    end;
end;
end;

```

Після цього потрібно дещо додати розділ глобальних змінних у мене виглядає так:

```

var
MainForm: TMainForm;
CapEnum: TSysDevEnum;
VideoMediaTypes: TEnumMediaType;
Writer: IFileSinkFilter;
PinList: TPinList;
CompFilter : TFilterList;
CapFilters : TSysDevEnum;
SendBt: TBitmap;
first: boolean =true ;

```

А на формі з'явився новий компонент Image класу TImage, він властиво й буде транслювати відео з віддаленого комп'ютера.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою MARS, який є блочно-симетричним шифром з відкритим ключем. Розмір блоку при шифруванні 128 біта, розмір ключа може варіюватися від 128 до 448 біт включно (кратні 32бітам). Творці прагнули поєднати в своєму алгоритмі швидкість кодування і стійкість шифру. В результаті вийшов один з самих криптостійкий алгоритм з алгоритмів, які брали участь в конкурсі AES.

Алгоритм унікальний тим, що використовував практично всі існуючі технології, застосовувані в криптоалгоритмах, а саме:

- Найпростіші операції (додавання, віднімання, виключаюче або).
- Підстановки з використанням таблиці замін.
- Фіксований циклічний зсув.
- Залежний від даних циклічний зсув.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

- Множення за модулем 2^{32} .
- Ключове забілювання.

Використання подвійного перемішування представляє складність для криптоаналізу, що деякі відносять до недоліків алгоритму. У той же час на даний момент не існує будь-яких ефективних атак на алгоритм, хоча деякі ключі можуть генерувати слабкі підключі.

Структура алгоритму

Автори шифру виходили з наступних припущень:

1. Вибір операцій. MARS був спроектований для використання на найсучасніших комп'ютерах того часу. Для досягнення найкращих захисних характеристик в нього були включені самі «сильні операції» підтримувані в них. Це дозволило добитися більшого відношення securityper-instruction для різних реалізацій шифру.

2. Структура шифру. Двадцятирічний досвід роботи в області криптографії підштовхнув творців алгоритму до думки, що кожен раунд шифрування грає свою роль в забезпеченні безпеки шифру. Зокрема, ми можемо бачити, що перший і останній раунди зазвичай сильно відрізняються від проміжних («центральных») раундів алгоритму в плані захисту від криптоаналітичних атак. Таким чином, при створенні MARSa використовувалася змішана структура, де перший і останній раунди шифрування істотно відрізняються від проміжних.

3. Аналіз. Швидше за все, алгоритм з гетерогенною структурою буде краще протистояти криптоаналітичним методам майбутнього, ніж алгоритм, всі раунди якого ідентичні. Розробники алгоритму MARS надали йому сильно гетерогенну структуру – раунди алгоритму дуже різняться між собою.

У шифрі MARS використовувалися такі методи шифрування:

1. Робота з 32-х бітними словами. Всі операції застосовуються до 32-бітовим словами. тобто вся початкова інформація розбивається на блоки по 32біта. (Якщо ж блок опинявся меншої довжини, то він доповнювався до 32біт)

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

2. Мережа Фейстеля. Творці шифру вважали, що це найкращий варіант поєднання швидкості шифрування і криптостійкості. В MARS використана мережа Фейстеля 3-го типу.

3. Симетричність алгоритму. Для стійкості шифру до різних атак всі його раунди були зроблені повністю симетричними, тобто друга частина раунду є дзеркальне повторення першої його частини.

Структуру алгоритму MARS можна описати таким чином:

1. Попереднє накладення ключа: на 32-бітові субблоки A, B, C, D накладаються 4 фрагмента розширеного ключа $k_0 \dots k_3$ операцією складання за модулем 2^{32} .

2. Виконуються 8 раундів прямого перемішування (без участі ключа шифрування).

3. Виконуються 8 раундів прямого криптоперетворення.

4. Виконуються 8 раундів зворотного криптоперетворення. [2]

5. Виконуються 8 раундів зворотного перемішування, також без участі ключа шифрування.

6. Фінальне накладення фрагментів розширеного ключа $k_{36} \dots k_{39}$ операцією віднімання за модулем 2^{32} .

Пряме перемішування

У першій фазі на кожне слово даних накладається слово ключа, а потім відбувається вісім раундів змішування згідно з мережею Фейстеля третього типу спільно з деякими додатковими змішування. У кожному раунді ми використовуємо одне слово даних (зване, вихідним словом) для модифікації трьох інших слів (звані, цільовими словами). Ми розглядаємо чотири байта вихідного слова як індексів на двох S-блоків, S_0 і S_1 , кожен, що складається з 256 32-розрядних слів, а далі проводимо операції XOR або додавання даних відповідного S-блоку в три інших слова.

Якщо чотири байти вихідного слова b_0, b_1, b_2, b_3 (де b_0 є першим байтом, а b_3 є старшим байтом), то ми використовуємо b_0, b_2 , як індекси в блоку S_0 і байти

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

b_1 , b_3 , як індекси в S-блоці S_1 . Спочатку зробимо XOR S_0 до першого цільовим речі, а потім додамо S_1 до того ж слова. Ми також додаємо S_0 до другого цільовим слову і XOR блоку- S_1 до третього цільовим слову. У висновку, ми обертаємо вихідне слово на 24 біта вправо.

У наступному раунді ми обертаємо наявні у нас чотири слова: таким чином, нинішні перші цільове слово стає наступним вихідним словом, поточним другий цільове слово стає новим першим цільовим словом, третє цільове слово стає наступний другий цільовим словом, і поточне вихідне слово стає третім цільовим словом.

Більш того, після кожного з чотирьох раундів ми додаємо одне з цільових слів назад у вихідне слово. Зокрема, після першого і п'ятого раундів ми додав третю цільове слово назад у вихідне слово, а після другого і шостого раунду ми додаємо першої цільової слово назад у вихідне слово. Причиною цих додаткових операцій змішування, є ліквідація декількох простих диференціальних криптоатаки в фазі перемішування, щоб порушити симетрію у фазі змішування та отримати швидкий потік.

Криптографічне ядро

Криптографічне ядро MARS – мережа Фейстеля 3-го типу, що містить в собі 16 раундів. У кожному раунді ми використовуємо ключову E-функцію, яка є комбінацією множень, обертань, а також звернень до S-блоків. Функція приймає на вхід слово даних, а повертає три слова, з якими згодом буде здійснена операція додавання або XOR до інших трьох слів даними. У доповненні вихідне слово обертається на 13 біт вліво.

Для забезпечення, серйозного опору до криптоатаки, три вихідних значення E-функції (O_1 , O_2 , O_3) використовуються в перших восьми раундах і в останніх восьми раундах в різних порядках. У перші вісім раундів ми додаємо O_1 і O_2 до першого і другого цільовим речі, відповідно, і XOR O_3 у третьому цільовим слову. За останні вісім раундів, ми додаємо O_1 і O_2 до третього і другого цільовим речі, відповідно, і XOR O_3 до першого цільовим слову.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

цільового слова. Нарешті, ми повертаємо початкове слово на 24 позицій вліво. Для наступного раунду ми обертаємо наявні слова так, щоб нинішнє перше цільове слово стало наступним вихідним словом, поточне друге цільове слово стало першим цільовим словом, поточне третє цільове слово стало другим цільовим словом, і поточне вихідне слово стало третім цільовим словом. Крім того, перед одним з чотирьох «особливих» раундів ми віднімаємо одне з цілових слів з вихідного слова: перед четвертим і восьмим раундами ми віднімаємо першої цільової слово, перед третьому і сьомим раундами ми віднімемо третя цільова слово з вихідного.

Дешифрування

Процес декодування обернений процесу кодування. Код дешифрування схожий (але не ідентичний) на код шифрування.

КБГІЗ-2023

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської роботи. Розроблене програмне забезпечення компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет складається з наступних функціональних блоків:

- Навігаційне меню: Дії; Фільтри; Налаштування; Довідка.
- Функції представлені у графічному вигляді (іконки).
- Вікно відображення відеопотоку.
- Навігаційного меню яке визивається натисканням правої клавіші маніпулятора миші.
- Блок функціональних кнопок ПЗ.

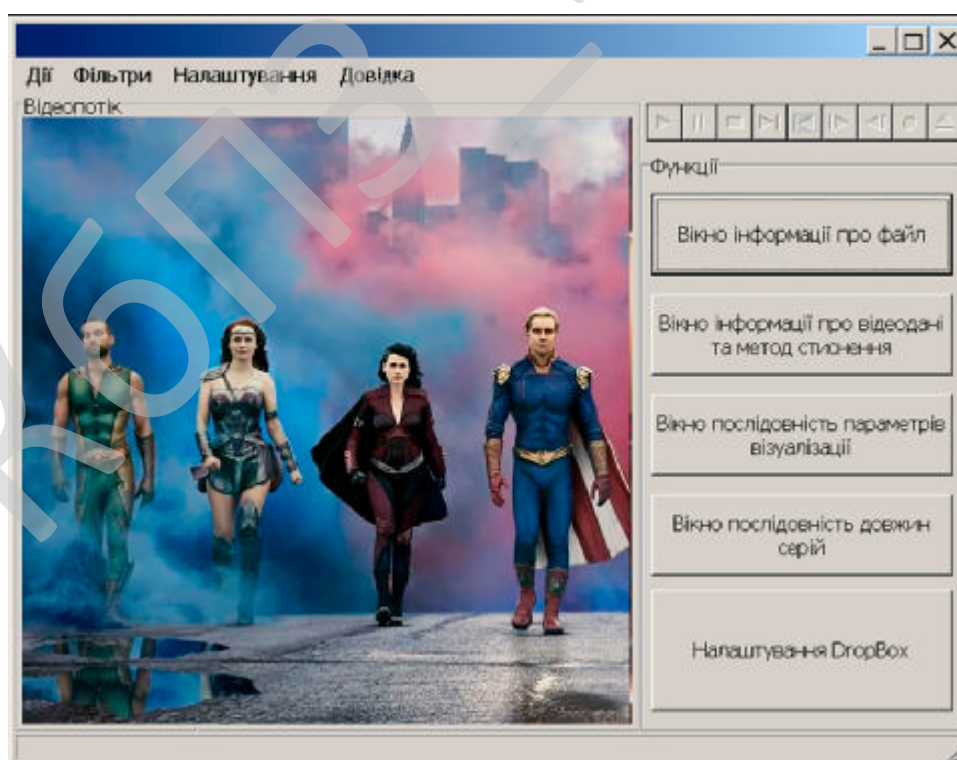


Рисунок 5.1 – Головне вікно розробленого ПЗ

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

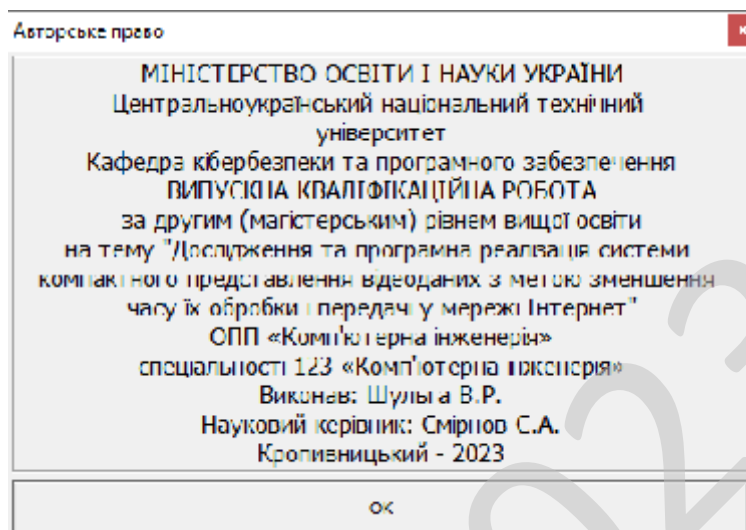


Рисунок 5.2 – Вікно розробника ПЗ

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

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

Проводилась оцінка:

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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

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

Ці тести демонструють:

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), zaplativshi avtorovi pevnu sumu.

V inshomu vipadku korystuvach povinen prypynity vykorystannya PZ ta vydaliti yogo zi svogo komp'yutera.

КБПЗ - 2023

					VKPM-123.23.0051.00.00.PZ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Метою розробки є дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Об'єктом дослідження є процес компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Предметом дослідження є методи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

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

- Удосконалено метод компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.
- Розроблено вітчизняний продукт компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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.	–	19000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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). Для вибраної мови програмування він складає 2,66;

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%. Приймаємо $S = 100$ %

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 100 = 168 \text{ люд/день.}$$

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

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

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

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

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

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

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	11	990	16,5
Монітор	60	11	660	11
Клавіатура	30	11	330	5,5
Маніпулятор «мишка»	30	11	330	5,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	2	120	2
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	5	150	2,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	400	1000	16,67
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	70,67

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{op}^c = \frac{71 \cdot 3}{1,2} = 177,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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	12525	37575
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	3,8	12000	136800
Інженер-електронщик	0,4	11500	13800
Інженер-системотехнік	0,25	11500	8625
Адміністратор мережі	0,5	11500	17250
Системний програміст	0,25	11500	8625
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	11700	8775
Бухгалтер-економіст	0,5	12500	18750
Всього за період розробки	$R_{cn} = 7,45$	-	$\Phi_{роб} = 268200$

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.

$$z_{cd} = \frac{268200}{7,45 \cdot 60} = 600 \text{ грн.}$$

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-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

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

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 ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 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:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 24.10.23 – джерело <http://computorg.ua/price.html>

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	AMD Ryzen 3 4100 (100-100000510BOX) AM4, 4 ядра, 8 потоків, 3.8 GHz, 4.0 GHz, (TDP) - 65 Вт, 7nm	-
Системна плата	Biostar B450MHP AM4, DDR4, LAN - 1 Гбіт/с, D-Sub (VGA), HDMI, - 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon RX 550 2Gb Biostar (VA5505RF21) PCI-Express 3.0, 2 ГБ, GDDR5, 128 Bit, Boost - 1183 MHz, 1 x HDMI, 1 x DVI, 1 x DisplayPort	-
Жорсткий диск	SSD 2.5" 240GB SA100 Acer, 3D TLC NAND, 2.5", SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 16GB 2666 MHz Essentials Mushkin	-
DVD-привод	DVDRW Pioneer DVR-TD10RS SATA Slim Black Bulk (DVR-TD10RS)	-
Корпус	ATX Middle Tower FOXCONN Pro, 3GTLA-489, PSU 550W(FSP Brand: ATX- 350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: A Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1 170/160, D-SUB, Wide)	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

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	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	28000	25	7000
Всього по групі	130690	-	27797,5
7. Нематеріальні активи	19000	10	1900
Разом	$K_p = 1672575$		$A_p = 157540$

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

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 600 \cdot 209 / 19 = 6600 \text{ грн.}$$

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, %.

$$Z_d = 6600 \cdot 10 \cdot 0,01 = 660 \text{ грн.}$$

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, %.

$$C_{oc} = 0,01 \cdot 22(6600 + 660) = 2686 \text{ грн.}$$

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, %.

$$G_{ocn} = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджів, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо 1/6 пачки паперу на три місяці розробки.

Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 210$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

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

$$Z_{M1} = 210 \cdot 1/6 \cdot 3 = 105 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 4 CD +1 DVD:

$$Z_{M2} = \sum Ц_d, \quad (7.17)$$

де: $Ц_d$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 20 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 27 грн./шт.

$$Z_{M2} = 20 \cdot 4 + 27 = 107 \text{ грн.}$$

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

$$Z_{M3} = \sum Ц_z, \quad (7.18)$$

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

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

$$Z_M = (105 + 107 + 1702)/19 = 101 \text{ грн.}$$

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

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

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

$$O_n = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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	6600
2. Додаткова зарплата виконавців	Z_o	660
3. Відрахування на соціальні потреби	C_{oc}	2686
4. Загальногосподарські витрати	G_{ocn}	990
5. Витрати на матеріали	Z_M	101
6. Освоєння нових операційних систем, мов програмування	O_n	990
7. Амортизація основних фондів	A_m	2073
8. Повна собівартість програмного забезпечення	C_n	14100
9. Плановий прибуток	P_p	7050
10. Ціна підприємства $C_n = C_n + P_p$	C_n	21150
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	4230
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	25380

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 19$ прим.):

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

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

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

$$Z_{ел\ баз} = 0,35 \cdot 4000 \cdot 3,8 = 5320 \text{ грн.}$$

$$Z_{ел\ нов} = 0,3 \cdot 4000 \cdot 3,8 = 4560 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	25380	–	6345
Всього відрахувань	-	–	25380	–	6345

7.8 Визначення економічної ефективності програмної продукції

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

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

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

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

$$E_e = (21150 - 14100) \cdot 19 - (0,05 \cdot 1408000 + 0,5 \cdot 114885 + 0,25 \cdot 33190 + 0,2 \cdot 97500 + 0,1 \cdot 19000) \cdot 3/12 = 94565 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_6 = \frac{264575}{(21150 - 14100) \cdot 19 \cdot 12 / 3} = 0,6 \text{ років.}$$

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.	19
2. Повна собівартість розробленої програми	Грн.	14100
3. Ціна розробленої програми	Грн.	21150
4. Плановий прибуток від реалізації розробленої програми	Грн.	7050
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1672575
7. Загальний прибуток від реалізації програмної продукції	Грн.	133950
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	94565
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	0,6
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	25380
11. Величина економічного ефекту у користувача програмної продукції	Грн.	1490
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	3,2

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

де: I_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_{δ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (37528 - 29693) - 0,25 \cdot 25380 = 1490 \text{ грн.}$$

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

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

$$T_{cn} = \frac{25380}{37528 - 29693} = 3,2 \text{ року.}$$

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-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Важливість небезпеки взагалі і охорони праці, як її складової частини, зокрема, не викликає сумнівів з давніх часів. Умови праці розглядав ще Аристотель (384-322 до н.е.).

Розвиток галузі інформаційних технологій (ІТ) привернув увагу спільноти на умови праці програмістів та інших спеціалістів ІТ сектора економіки.

Загальні положення державної політики в галузі охорони праці регламентуються Законом України “Про охорону праці” [2], а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Відомо, що програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м’язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення газу.

Вплив негативних чинників може привести до професійних захворювань.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

У даному розділі розглянемо шкідливі чинники роботи програмістів та інших спеціалістів ІТ, та проведемо аналіз умов праці. При цьому будемо керуватись наступними нормативно-правовими актами: «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10, «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

На умови праці програміста впливають наступні фактори:

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом [1]. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- монотонність праці;

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

У зазначеному приміщенні працюють 7 людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа приміщення у розрахунку на одного робочого місця програміста відповідає нормативним вимогам як 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», так і СанПіН 3.3.2 – 007 – 98.

А об'єм приміщення у розрахунку на одного робочого місця програміста відповідає 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», але дещо нижчий вимог СанПіН 3.3.2 – 007 – 98 (19,25 м³ замість 20 м³). Це цілком прийнятно, враховуючи хронологію та пріоритет опубліковання вищезгаданих нормативних актів.

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 врача України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

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

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

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

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

В літню пору застосовується кондиціонер.

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

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

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

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р 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. 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;
- бажано, щоб робочий стіл при необхідності можна було регулювати по висоті в межах 680-780 мм, а висота над рівнем підлоги робочої поверхні, на якій працює програміст, повинна складати 720 мм.

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

Додаткові рекомендовані заходи: забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення та занулення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

8.5 Розрахунок штучного освітлення

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 2,28 м, довжина – 2,61 м, висота – 3 м.

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

$$F = ESKZ/n,$$

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S = 2,6 \times 2,6 = 6,76$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$.

Обчислимо індекс приміщення за формулою:

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

$$i=S/(h(A+B)),$$

де: S – площа приміщення, $S = 6,76 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3 \text{ м}$ (збігає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 2,6 \text{ м}$;

B – довжина приміщення, $B = 2,6 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекса приміщення:
 $i=0,43$.

Знаючи індекс приміщення, за знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп). Підставимо всі значення у формулу, визначимо світловий потік:
 $F=14548 \text{ Лм}$.

Будемо використовувати лампи TL-F 12 36W LED 6000K IP65, світловий потік яких $F_{\text{л}} = 3000 \text{ Лм}$.

Число ламп визначається по формулі:

$$N=F/F_{\text{л}}$$

де: F – світловий потік,

$F_{\text{л}}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення: $N=$
 $14548/3000=4,8$ шт.

Приймаємо необхідну кількість ламп 5 шт.

Висновки до розділу

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

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

Список використаних джерел інформації

1. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
2. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. *Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>*
3. Центр післядипломної освіти та підвищення кваліфікації *Режим доступу до ресурсу: <https://cpo.stu.cn.ua/>*

КБГПЗ - 2023

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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 компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.
- Досліджена система компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.
- На основі отриманих результатів досліджень створена програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

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

Програма реалізована на мові високого рівня Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. 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 MARS.

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 1490 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 3,2 роки.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шульга В.Р. Дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
3. 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.
4. 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.
5. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph.. – 2011. – Vol. 30, no. 4. – P. 99:1–99:8.
6. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760–2768.
7. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
8. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,
9. 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.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

10. 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>

11. 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.

12. 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.

13. 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.

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

15. 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.

16. 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.

17. 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.

18. 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.

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

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

					БКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

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

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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).

40. 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

41. 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.

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

43. 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.

44. 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.

45. Puri A., Chen T. (eds.). Multimedia Systems, Standards and Networks. Marcel Dekker, 2000.

46. Sadka A. Compressed Video Communications. John Wiley & Sons, 2002.

47. Walsh A., Bourges-Sevenier M. (eds.). MPEG-4 Jump Start. Prentice-Hall, 2002.

48. Haskell B., Puri A., Netravali A. Digital Video: An Introduction to MPEG-2. Chapman & Hall, 1996.

49. Pereira F., Ebrahimi T. (eds.). The MPEG-4 Book. IMSC Press, 2002.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

50. Richardson I.E.G. Video Codec Design. John Wiley & Sons, 2002.
51. Parhi K.K., Nishitani T. (eds.). Digital Signal Processing for Multimedia Systems. Marcel Dekker, 1999.
52. Rao K.R., Hwang J.J. Techniques and Standards for Image, Video and Audio Coding. Prentice Hall, 1997.
53. Ghanbari M. Video Coding: An Introduction to Standard Codecs. IEE Press, 1999.
54. Riley M.J., Richardson I.E.G. Digital Video Communications. Artech House, 1997.
55. Bhaskaran V., Konstantinides K. Image and Video Compression Standards: Algorithms and Architectures. Kluwer, 1997.
56. Pennebaker W.B., Mitchell J.L., Fogg C., LeGall D. MPEG Digital Video Compression Standard, Chapman & Hall, 1997.
57. IEEE Transactions on Circuits and Systems for Video Technology, special issue on H.264/AVC, to appear in 2003.

					ВКРМ-123.23.0051.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.23.0051.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Шульга В.Р.				<i>Дослідження та програмна реалізація системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнов С.А.					М	1	6
Н. Контр.	Коваленко А.С.					ЦНТУ КІ-22М-2		
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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. № 35-13 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 компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет.

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;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-123.23.0051.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи компактного представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі Інтернет;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.23.0051.00.00.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 Delphi 10.4.

					ВКРМ-123.23.0051.00.00.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-123.23.0051.00.00.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 – 105 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 19.12.2023 р.

					ВКРМ-123.23.0051.00.00.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 С.А.

*Дослідження та програмна реалізація
системи компактного представлення відеоданих з метою зменшення часу
їх обробки і передачі у мережі Інтернет*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 16

Літера: РП

Кропивницький – 2023 року

VideoCap.dpr - головний файл проекту

```
program VideoCap;
//підключення бібліотек
uses
  Forms,
  main in 'main.pas' {MainForm},
  option in 'option.pas' {OptionForm},
  prog in 'prog.pas' {ProgForm},
  about in 'about.pas' {AboutForm};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Обробка відеоінформації з Відеокамери';
  Application.CreateForm(TMainForm, MainForm); // створення головного вікна
  програми
  Application.CreateForm(TOptionForm, OptionForm); // створення вікна «Параметри
  відео»
  Application.CreateForm(TProgForm, ProgForm); // створення вікна «Параметри
  програми»
  Application.CreateForm(TAboutForm, AboutForm); // створення вікна довідки
  Application.Run;
end.
```

main.pas - основна програма

```

unit main;

interface

uses
  Windows, Messages, SysUtils, jpeg ,Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, DSPack, DSUtil, DirectShow9, ComCtrls, ExtCtrls, IniFiles,
  XPMAN,
  ToolWin, ActnMan, ActnCtrls, ActnMenus, XPStyleActnCtrls, ActnList,
  ImgList, OleCtrls, Buttons, StrUtils,
  ScktComp, Sockets, AppEvnts;

type
  TMainForm = class(TForm)
    CaptureGraph: TFilterGraph;
    VideoSourceFilter: TFilter;
    StatusBar: TStatusBar;
    MenuBar: TActionMainMenuBar;
    ActionManager: TActionManager;
    VideoSend: TAction;
    Exit: TAction;
    OnTop: TAction;
    Help: TAction;
    About: TAction;
    OptionProg: TAction;
    SampleGrabber: TSampleGrabber;
    ImageList: TImageList;
    Timer: TTimer;
    Size: TAction;
    ApplicationEvents: TApplicationEvents;
    Image: TImage;
    bt2: TImage;
    bt1: TImage;
    VideoWindow2: TVideoWindow;
    procedure FormDestroy(Sender: TObject);
    function IniGetStringValue(
      TheIniFile: string;
      IniSection: string;
      StringName: string;
      DefaultString: string): string;
    function IniSetStringValue(
      TheIniFile: string;
      IniSection: string;
      StringName: string;
      StringValue: string): Boolean;
    procedure OnTopExecute(Sender: TObject);
    procedure VideoSendExecute(Sender: TObject);
    procedure OptionVideoExecute(Sender: TObject);
    procedure ExitExecute(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure OptionProgExecute(Sender: TObject);
    procedure AboutExecute(Sender: TObject);

    procedure SendBitmap(hTargetWnd : HWND; Btm: TBitmap);
    procedure SendText (hTargetWnd : HWND; Text: string);
    procedure TimerTimer(Sender: TObject);
    procedure CreateProgressDialog;
    procedure Wait(tim: integer);
    function SetVideoParams(CB_B2: ICaptureGraphBuilder2; Category: TGUID;
      fSource: IBaseFilter): HRESULT;
    procedure SizeExecute(Sender: TObject);
    procedure HelpExecute(Sender: TObject);
    procedure ApplicationEventsHint(Sender: TObject);
  end;

```

```

private
  { Private declarations }
  procedure SendCopyData(hTargetWnd: HWND; ACopyDataStruct:TCopyDataStruct);
  procedure WMCopyData(var Msg: TWMCopyData); message WM_COPYDATA;
public
  { Public declarations }
  Format, Mode, Bit, Kadr, arch : integer;
  gladco: string;
  Kadr2: REAL;
end;

var
  MainForm: TMainForm;
  CapEnum: TSysDevEnum;
  VideoMediaTypes: TEnumMediaType;
  Writer: IFileSinkFilter;
  PinList: TPinList;
  CompFilter : TFilterList;
  CapFilters : TSysDevEnum;
  SendBt: TBitmap;
  first: boolean =true ;

implementation

uses option, prog, about;

{$R *.dfm}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////// Функції й процедури для передачі даних по з'єднанню //////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// передача бітмапа
procedure TMainForm.SendBitmap(hTargetWnd : HWND; Btm: TBitmap);
var
  ms: TMemoryStream;
  JpegIm: TJpegImage;
  MyCopyDataStruct: TCopyDataStruct;
begin
  ms := TMemoryStream.Create;
  JpegIm := TJpegImage.Create;
  try
    JpegIm.Assign(Btm);
    JpegIm.CompressionQuality := arch;
    JpegIm.Compress;
    JpegIm.SaveToStream(ms);
    with MyCopyDataStruct do
      begin
        dwData := 1;
        cbData := ms.Size;
        lpData := ms.Memory;
      end;
    SendCopyData(hTargetWnd,MyCopyDataStruct);
  finally
    ms.Free;
    JpegIm.Free;
  end;
end;

// передача тексту
procedure TMainForm.SendText( hTargetWnd : HWND; Text: string);
var
  MyCopyDataStruct: TCopyDataStruct;
begin
  with MyCopyDataStruct do
    begin

```

```

    dwData := 0;
    cbData := StrLen(PChar(Text)) + 1;
    lpData := PChar(Text)
end;
SendCopyData(hTargetWnd, MyCopyDataStruct);
end;

// для передачі даних
procedure TMainForm.SendCopyData(hTargetWnd: HWND;
ACopyDataStruct:TCopyDataStruct);
begin
    if hTargetWnd <> 0 then
        SendMessage(hTargetWnd, WM_COPYDATA, Longint(Handle),
Longint(@ACopyDataStruct))
// else ShowMessage('Заданий Handle не знайдений');
end;

// для прийому бітмапа
procedure TMainForm.WMCopyData(var Msg: TWMCopyData);
var
    sText: array[0..99] of Char;
    ms: TMemoryStream;
    JpegIm: TJpegImage;
begin
    case Msg.CopyDataStruct.dwData of
        0: { текст }
            begin
                StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
            end;
        1: { картинка }
            begin
                ms := TMemoryStream.Create;
                JpegIm := TJpegImage.Create;
                try
                    with Msg.CopyDataStruct^ do
                        ms.Write(lpdata^, cbdata);
                        ms.Position := 0;
                        JpegIm.LoadFromStream(ms);
                        try
                            begin
                                Height:=JpegIm.Height+69;
                                Width:=JpegIm.Width;
                                Image.Picture.Graphic.Assign(JpegIm);
                            end;
                        except
                            end;
                        finally
                            ms.Free;
                            JpegIm.Free;
                        end;
                    end;
                end;
            end;
    end;
end;

////////////////////////////////////
//////////////////////////////////// Функції й процедури для роботи із графікою
////////////////////////////////////
////////////////////////////////////
function TMainForm.SetVideoParams(CB_B2: ICaptureGraphBuilder2; Category: TGUID;
fSource: IBaseFilter): HRESULT;
var
    StreamConf: IAMStreamConfig;
    PAMT: PAMMediaType;
begin
    Result:= E_FAIL;
    StreamConf:= nil;
    PAMT:= nil;
    try
        Result:= CB_B2.FindInterface(@Category, @MEDIATYPE_Video, fSource,

```

```

IID_IAMStreamConfig, StreamConf);
if Assigned(StreamConf) then
begin
  StreamConf.GetFormat(PAMT);
  if Assigned(PAMT) then
  begin
    if PAMT.cbFormat= sizeof(TVideoInfoHeader) then
    begin
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biWidth:= 640;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biHeight:= 480;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biBitCount:= 24;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.AvgTimePerFrame:= 10000000 div Kadr;
      //fps
      with PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader do
        PAMT^.lSampleSize := ((biWidth + 3) and (not (3))) * biHeight *
biBitCount
        shr 3;

PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biSizeImage:=PAMT^.lSampleSize;
      end;
      Result:= StreamConf.SetFormat(PAMT^);
      end;
      end;
      result:= S_OK;
    except
    on E: Exception do
      MessageBox(0, PChar(E.Message), '', MB_OK or MB_ICONERROR);
    end;
    StreamConf:= nil;
    if Assigned(PAMT) then
      DeleteMediaType(PAMT);
end;
end;

```

```

////////////////////////////////////
////////// Функції й процедури для роботи з файлами й директоріями //////////
////////////////////////////////////

```

```

// одержати значення з файлу
function TMainForm.IniGetStringValue(
  TheIniFile: string;
  IniSection: string;
  StringName: string;
  DefaultString: string): string;
var
  TheIni: TIniFile;
begin
  TheIni := TIniFile.Create(TheIniFile);
  try
    Result :=
      TheIni.ReadString(
        IniSection,
        StringName,
        DefaultString);
    if Result = '' then
      Result := DefaultString;
  finally
    TheIni.Free;
  end;
end;
end;

```

```

// записати значення у файл
function TMainForm.IniSetStringValue(
  TheIniFile: string;
  IniSection: string;
  StringName: string;
  StringValue: string): Boolean;
var

```

```

    TheIni: TIniFile;
begin
    TheIni := TIniFile.Create(TheIniFile);
    try
        try
            TheIni.WriteString(
                IniSection,
                StringName,
                StringValue);
            Result := True;
        except
            Result := False;
        end;
    finally
        TheIni.Free;
    end;
end;

/////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////// Основний текст програми ///////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

procedure TMainForm.Wait (tim: integer);
var
    whevent:THandle;
begin
    whevent:=createevent (nil,true,false,'et');
    while (true) do
        begin
            WaitForSingleobject (whevent,tim);
            Application.processmessages;
        end;
    end;
end;

procedure TMainForm.FormDestroy(Sender: TObject);
begin
    SendBt.Destroy;
    CaptureGraph.ClearGraph;
    CapEnum.Free;
    VideoMediaTypes.Free;
end;

procedure TMainForm.OnTopExecute(Sender: TObject);
begin
    // відображення програми поверх всіх вікон
    if OnTop.Checked then MainForm.FormStyle:=fsStayOnTop
    else MainForm.FormStyle:=fsNormal;
end;

procedure TMainForm.VideoSendExecute (Sender: TObject);
begin
    VideoWindow2.FilterGraph:=CaptureGraph;
    CapEnum:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
    CaptureGraph.ClearGraph;
    CaptureGraph.Active := false;
    VideoSourceFilter.BaseFilter.Moniker := CapEnum.GetMoniker (Format);
    VideoSourceFilter.FilterGraph := CaptureGraph;
    CaptureGraph.Active := true;

    SetVideoParams (CaptureGraph as ICaptureGraphBuilder2,
        PIN_CATEGORY_CAPTURE, VideoSourceFilter as IBaseFilter);

    with CaptureGraph as ICaptureGraphBuilder2 do
        RenderStream (@PIN_CATEGORY_PREVIEW, nil, VideoSourceFilter as IBaseFilter,
            SampleGrabber as IBaseFilter, VideoWindow2 as IbaseFilter);

    CaptureGraph.Play;

```

```

    Timer.Enabled:=true;
end;

procedure TMainForm.OptionVideoExecute(Sender: TObject);
begin
    // опції відео
    OptionForm.Show;
end;

procedure TMainForm.ExitExecute(Sender: TObject);
begin
    // завершення роботи
    Close;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
    // вивід прогрес діалогу
    CreateProgressDialog;
end;

procedure TMainForm.OptionProgExecute(Sender: TObject);
begin
    ProgForm.Show;
end;

procedure TMainForm.AboutExecute(Sender: TObject);
begin
    AboutForm.Show;
end;

procedure TMainForm.TimerTimer(Sender: TObject);
begin
    // перехоплення буфера
    SampleGrabber.GetBitmap(SendBt);
    SendBitmap(FindWindow(nil, PChar('Server/Client')), SendBt);
end;

// процедура виводу смуги завантаження
procedure TMainForm.CreateProgressDialog;
var
    AMsgDialog: TForm;
    AProgressBar: TProgressBar;
begin
    AMsgDialog := CreateMessageDialog('Процес підготовки програми до роботи',
mtCustom, []);
    AProgressBar := TProgressBar.Create(AMsgDialog);

    with AMsgDialog do
        try
            Caption := 'Проста відеоконференція';
            Height := 120;

            with AProgressBar do
                begin
                    Parent := AMsgDialog;
                    top := 50;
                    Left := 8;
                    Width := 220;
                    min:=0;
                    max:=100;
                end;

                Show;
            //////////////////////////////////////
            // створення об'єктів Graphic
            bt1.Picture.Graphic:=TBitmap.Create;

```

```

AProgressBar.Position:=10;
Application.ProcessMessages;

bt2.Picture.Graphic:=TBitmap.Create;

SendBt:= TBitMap.Create;
CompFilter := TFilterList.Create;
CapFilters := TSysDevEnum.create(CLSID_VideoCompressorCategory);
CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
AProgressBar.Position:=30;
Application.ProcessMessages;
VideoMediaTypes := TEnumMediaType.Create;
AProgressBar.Position:=50;
Application.ProcessMessages;
// завантаження налаштувань програми

Format:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Format',''));

Mode:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Mode',''));

Bit:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Bit',''));

Kadr:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Kadr',''));
AProgressBar.Position:=70;
Application.ProcessMessages;

arch:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Program', 'arch',''));

Kadr2:=StrToFloat (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_op
tion.ini','Program', 'Kadr2',''));
AProgressBar.Position:=90;
Application.ProcessMessages;
Timer.Interval:=round(Timer.Interval/kadr2);
AProgressBar.Position:=100;
Application.ProcessMessages;
////////////////////////////////////
finally
  AProgressBar.Free;
  Free;
end;
end;

procedure TMainForm.SizeExecute(Sender: TObject);
begin
  OptionForm.Show;
end;

procedure TMainForm.HelpExecute(Sender: TObject);
begin
  ShowMessage('А тут докладна інформація як з усім цим працювати :');
end;

procedure TMainForm.ApplicationEventsHint(Sender: TObject);
begin
  StatusBar.SimpleText:=Application.Hint;
end;

end.

```

prog.pas - параметри програми

```

unit prog;

interface

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

type

  TProgForm = class(TForm)
    BtnPanel: TPanel;
    Panel: TPanel;
    GroupBox: TGroupBox;
    cancel: TButton;
    save: TButton;
    KadrEdit: TLabelledEdit;
    CompressEdit: TLabelledEdit;
    procedure cancelClick(Sender: TObject);
    function OpenDir: string;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ProgForm: TProgForm;

implementation

uses main;

{$R *.dfm}

function TProgForm.OpenDir: string;
var
  TitleName : string;
  lpItemID : PItemIDList;
  BrowseInfo : TBrowseInfo;
  DisplayName : array[0..MAX_PATH] of char;
  TempPath : array[0..MAX_PATH] of char;
begin
  FillChar(BrowseInfo, sizeof(TBrowseInfo), #0);
  BrowseInfo.hwndOwner := ProgForm.Handle;
  BrowseInfo.pszDisplayName := @DisplayName;
  TitleName := 'Укажіть директорію';
  BrowseInfo.lpszTitle := PChar(TitleName);
  BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;
  lpItemID := SHBrowseForFolder(BrowseInfo);
  if lpItemID <> nil then
    begin
      SHGetPathFromIDList(lpItemID, TempPath);
      OpenDir:=TempPath;
      GlobalFreePtr(lpItemID);
    end;
end;

procedure TProgForm.cancelClick(Sender: TObject);
begin
  Close;
end;

procedure TProgForm.FormCreate(Sender: TObject);
begin

```

```
KadrEdit.Text :=FloatToStr(MainForm.Kadr2);  
CompressEdit.Text:=IntToStr(MainForm.arch);  
end;  
  
end.
```

К6П3 - 2023

option.pas - параметри відео

```

unit option;

interface

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

type
  TOptionForm = class(TForm)
    PageControll: TPageControl;
    VideoSheet: TTabSheet;
    PanelFormat: TPanel;
    VideoCapFilters: TRadioGroup;
    PanelBtn: TPanel;
    Ok: TButton;
    Cancel: TButton;
    ModePanel: TPanel;
    GroupBox: TGroupBox;
    VideoSourceFilter: TFilter;
    CaptureGraph: TFilterGraph;
    AudioSourceFilter: TFilter;
    GroupMode: TRadioGroup;
    GroupBit: TRadioGroup;
    EditKadr: TSpinEdit;
    LabelCadr: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure CancelClick(Sender: TObject);
    procedure OkClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  OptionForm: TOptionForm;
  CapEnum: TSysDevEnum;
  VideoMediaTypes: TEnumMediaType;
  CapFile: WideString = 'c:\capture.avi';

implementation
  uses main;

  {$R *.dfm}

  procedure TOptionForm.FormCreate(Sender: TObject);
  var i: integer;
  begin
    CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
      VideoCapFilters.Items.Add(CapEnum.Filters[i].FriendlyName);

    VideoMediaTypes := TEnumMediaType.Create;

    // установка налаштувань для візуального відображення
    VideoCapFilters.ItemIndex:=MainForm.Format;
    GroupMode.ItemIndex:=MainForm.Mode;
    GroupBit.ItemIndex:=MainForm.Bit;
    EditKadr.Value:=MainForm.Kadr;
  end;

  procedure TOptionForm.CancelClick(Sender: TObject);
  begin
    Close;
  end;

```

```
procedure TOptionForm.OkClick(Sender: TObject);
begin
    try

MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
, 'Video', 'Format', IntToStr(VideoCapFilters.ItemIndex));

MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
, 'Video', 'Mode', IntToStr(GroupMode.ItemIndex));

MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
, 'Video', 'Bit', IntToStr(GroupBit.ItemIndex));

MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
, 'Video', 'Kadr', IntToStr(EditKadr.value));
        ShowMessage('Збереження ВІДЕО налаштувань виконано.');
```

except

```
        ShowMessage('Збереження налаштувань не можливо.');
```

```
    end;
end;

end.
```

КБПЗ - 2023


```

procedure TMainForm.WMCopyData(var Msg: TWMCopyData);
var
  sText: array[0..99] of Char;
begin
  case Msg.CopyDataStruct.dwData of
    0: { текст }
      begin
        StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
      end;
    1: { картинка }
      begin
        try
          with Msg.CopyDataStruct^ do
            UDPClient.SendBuffer(lpdata^, cbdata);
          except
            end;
        end;
      end;
  end;
end;

/////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////// Робота клієнта й сервера ///////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

procedure TMainForm.Button1Click(Sender: TObject);
begin
  UDPClient.Active:=false;
  UDPClient.Host:=EditHost.Text;
  AdressLabel.Caption:=EditHost.Text;
  UDPClient.Active:=true;
end;

procedure TMainForm.UDPServerUDPRead(Sender: TObject; AData: TStream;
  ABinding: TIdSocketHandle);
var
  ms: TMemoryStream;
  s: String;
begin
  ms:= TMemoryStream.Create;
  ms.CopyFrom(AData, AData.Size);
  SendBitmap(FindWindow(nil, PChar('Проста відеоконференція')),ms);
  ms.Free;
end;

end.

```

about.pas - довідка

```
unit frmAbout;
interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls, Buttons,
ExtCtrls;

type
  TAboutBox = class(TForm)
    Panell: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    Memol: TMemo;
    OKButton: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var AboutBox: TAboutBox;

implementation
{$R *.dfm}
procedure TAboutForm.FormCreate(Sender: TObject);
begin
  Memol.Clear;
  Memol.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Memol.Lines.Add('');
  Memol.Lines.Add('на тему:');
  Memol.Lines.Add('');
  Memol.Lines.Add('Дослідження та програмна реалізація системи компактного
представлення відеоданих з метою зменшення часу їх обробки і передачі у мережі
Інтернет ');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
  Memol.Lines.Add('Керівник: Смірнов С.А. ');
  Memol.Lines.Add('');
  Memol.Lines.Add('Розробив: студент Шульга Владислав Романович');
  Memol.Lines.Add('                гр. КІ-22М-2');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
  Memol.Lines.Add('м. Кропивницький 2023');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
end;
procedure TAboutForm.Button1Click(Sender: TObject);
begin
  AboutForm.Close;
end;
end.
```