

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 інтелектуальної
системи обробки та конвертації зображень”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Лисогор В.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Минайленко Р.М.
« ____ » _____ 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

Лисогору Владиславу Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень*

2. Керівник роботи *Минайленко Роман Миколайович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 34-13 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 інтелектуальної системи обробки та конвертації зображень*

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

Лисогор В.В. Дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень. 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 інтелектуальної системи обробки та конвертації зображень.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

Об'єктом дослідження є процес інтелектуальної системи обробки та конвертації зображень.

Предметом дослідження є методи інтелектуальної системи обробки та конвертації зображень.

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

Результат роботи – програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

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 RAD Studio Delphi.

Ключові слова: комп'ютерна інженерія, обробка та конвертація зображень

ABSTRACT

Lysohor V.V. Research and software implementation of an intelligent image processing and conversion system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of intelligent image processing and conversion.

The goal of the development is the research and software implementation of an intelligent image processing and conversion system.

The object of research is the process of an intelligent image processing and conversion system.

The subject of research is the methods of an intelligent image processing and conversion system.

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

The result of the work is the software implementation of the intelligent image processing and conversion system.

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 RAD Studio Delphi environment.

Keywords: computer engineering, image processing and conversion

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи	26
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	50
3.4 Розробка діаграми процесів.....	54
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	56
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	56
4.2 Захист розробленого програмного забезпечення.....	67
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	70
6 НАУКОВА НОВИЗНА	74

						ВКРМ-123.23.0014.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата		Літ.	Аркуш	Аркушів
Розроб.	Лисогор В.В.				Дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень	М	1	112
Перев.	Минайленко Р.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22М-1		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	75
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	75
7.2 Розрахунок трудомісткості розробки програмної продукції.....	77
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	79
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	84
7.5 Визначення собівартості розробки та ціни програмної продукції.....	88
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	91
7.7 Визначення експлуатаційних витрат.....	91
7.8 Визначення економічної ефективності програмної продукції.....	93
7.9 Висновок.....	95
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	96
8.1 Вступ.....	96
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	97
8.3 Аналіз умов праці програміста	98
8.4 Розробка заходів з умов поліпшення охорони праці.....	101
8.5 Розрахункова частина	102
9 ОСНОВНІ ВИСНОВКИ.....	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ДКП	–	дискретне косинусоїдальне перетворення
МК	–	матриця квантування
BMP	–	Bitmap Picture – графічний формат
DDB	–	апаратно-залежні растри
DIB	–	апаратно-незалежні растри
GIF	–	Graphics Interchange Format – графічний формат
ISO	–	міжнародна організація по стандартизації
JPEG	–	Joint Photographic Experts Group – графічний формат
LZW	–	алгоритм Лемпеля – Зіва – Велча стискання інформації
RLE	–	Run Length Encoding – групове кодування

КБГЗ-2023

ВСТУП

Актуальність теми. Конвертер зображень – це простий інструмент, який забезпечує швидкий спосіб виконання основних завдань перетворення зображень:

Він може конвертувати будь-який формат зображення, який розуміє конвертер зображень, у формати BMP, GIF, PNG і JPG.

Він може обертати зображення з кроком 90 градусів, а також автоматично обертати, щоб компенсувати орієнтацію EXIF.

Він може змінювати розмір або збільшувати зображення у відсотках або до абсолютного розміру, за бажанням зберігаючи оригінальне співвідношення сторін.

Щоб отримати доступ до конвертера зображень, виберіть зображення або зображення, які потрібно конвертувати, і, як правило, виберіть команду «Перетворити» в меню «Інструменти / Перетворити зображення». Група типів файлів «Зображення» також додає команду до контекстного меню поширених форматів зображень, тому ви також можете отримати доступ до конвертера, клацнувши правою кнопкою миші файл зображення та вибравши команду «Перетворити зображення».

Конвертер зображень відображає мініатюру наступного зображення, яке буде оброблено. Ви можете ввімкнути деякі або всі наведені нижче параметри одночасно; наприклад, ви можете обертати та змінювати розмір за один крок.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем інтелектуальної системи обробки та конвертації зображень.
- Дослідження системи інтелектуальної системи обробки та конвертації зображень.
- Програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

Об'єктом дослідження є процес інтелектуальної системи обробки та конвертації зображень.

Предметом дослідження є методи інтелектуальної системи обробки та конвертації зображень.

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

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

- Удосконалено метод інтелектуальної системи обробки та конвертації зображень.
- Розроблено вітчизняний продукт інтелектуальної системи обробки та конвертації зображень, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної системи обробки та конвертації зображень.

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГПЗ - 2023

					VKPM-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Конвертер зображень, як було вже визначено вище, це простий інструмент, який забезпечує швидкий спосіб виконання основних завдань перетворення зображень, призначенням якого може бути реалізація також наступних функцій.

Перетворити: увімкніть цей параметр, якщо ви хочете перетворити зображення в інший формат. конвертер зображень може зберігати зображення у форматах Bitmap, GIF, PNG і JPEG. Якщо вихідне зображення ще не в одному з цих форматів, вам потрібно вибрати вихідний формат для збереження, але якщо вихідне зображення вже є одним із цих форматів, ви можете залишити параметр «Перетворити» вимкненим.

Під час конвертації у формат JPEG необхідно вибрати параметр якості JPEG від 0 до 100. JPEG – це стиснення з втратами метод, а параметр якості визначає, наскільки точно перетворене зображення представлятиме вихідні дані зображення. Чим вище число, тим кращою буде якість, але також тим більшим буде кінцевий розмір файлу.

Під час конвертації з формату, який підтримує альфа-канал (прозорість), як-от PNG, ви можете вказати колір тла альфа-конверсії. Цей колір використовуватиметься як колір фону, якщо збережено у форматі, який не підтримує альфа-канал.

Повернути: увімкніть цю опцію, якщо ви хочете повернути зображення. Ви можете вибрати величину повороту на 90, 180 або 270 градусів – попередній перегляд мініатюр оновлюватиметься в режимі реального часу відповідно до ваших налаштувань. Якщо у вихідному зображенні збережена інформація про орієнтацію EXIF (наприклад, фотографія з цифрової камери, зроблена як

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

портретна, а не альбомна), ви також можете вибрати Використовувати інформацію EXIF зі спадного меню. Якщо вибрано, конвертер зображень використовуватиме інформацію EXIF для автоматичного повороту зображення до правильної орієнтації. Якщо зображення не містить інформації EXIF або воно вже має правильну орієнтацію, вихідне зображення не буде повернуто.

Вибір «Скинути» зі спадного меню має ефект, протилежний використанню інформації EXIF – замість використання орієнтації EXIF для обертання зображення конвертер зображень залишить зображення в спокої та скине поле орієнтації EXIF на 0. Знову ж таки, якщо зображення не має EXIF ця опція не змінюватиме зображення.

Конвертер зображень підтримує обертання JPEG без втрат, якщо це можливо. Зазвичай, коли ви завантажуєте та повторно зберігаєте зображення JPEG, якість нового зображення погіршується. Спочатку це важко помітити, але помилки накопичуються, і файл JPEG, який неодноразово розпаковувався та повторно стискався, часто виглядатиме набагато гірше, ніж оригінал. Формат JPEG дозволяє обертати зображення без втрат (тобто без зниження якості), якщо виконуються певні умови; а саме, ширина та висота зображення мають бути точно кратними «розміру блоку», у якому було збережено зображення (зазвичай 8x8 або 16x16). На щастя, більшість цифрових камер створюють зображення таких розмірів. Вам не потрібно робити нічого особливого в конвертер зображень, щоб увімкнути обертання без втрат – якщо зображення JPEG можна обертати без втрат, це буде так.

Змінити розмір: увімкніть цю опцію, якщо ви хочете змінити розмір зображення. Ви можете вибрати деякі попередньо визначені вихідні розміри або коефіцієнти зменшення чи збільшення зі спадного меню або вибрати Нестандартний розмір і ввести власні значення ширини та висоти (указані в пікселях).

Параметр «Додати суфікс імені файлу» змушує конвертер зображень автоматично додавати відповідний суфікс до імені файлу, коли він зберігає

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

вихідний файл. Наприклад, якщо ви змінили розмір IMG_2483.jpg до 150% від вихідного розміру, вихідний файл називатиметься IMG_2483-150%.jpg.

Увімкніть опцію «Зберігати співвідношення сторін», щоб уникнути зміни співвідношення сторін зображення під час зміни його розміру. Конвертер зображень налаштує вихідні розміри, щоб зберегти те саме співвідношення між шириною та висотою, що й вихідне зображення.

Записати перетворені зображення в цільову папку: Зазвичай конвертер зображень зберігає перетворені зображення в тій же папці, що й вихідні зображення. Якщо ви увімкнете цю опцію, конвертер зображень замість цього збереже перетворені зображення до поточного місця призначення/відображення файлу. Якщо ви залишите цей параметр вимкненим, а ім'я вихідного файлу не змінено через параметр «Додати суфікс імені файлу» (вище), вихідний файл зображення буде перезаписано. У цьому випадку вам буде запропоновано підтвердження (і можливість змінити ім'я вихідного файлу вручну).

Коли ви встановите потрібні параметри перетворення, натисніть кнопку ОК, щоб перетворити поточне зображення. Якщо ви спочатку вибрали кілька зображень, ви можете застосувати те саме перетворення до всіх із них (як пакетну операцію), натиснувши замість цього кнопку «Усі». Якщо ви натиснете «ОК» і виберете кілька зображень, перше зображення буде оброблено, а потім знову відкриється діалогове вікно перетворення зображення для другого зображення.

1.2 Область застосування

Областю застосування є обробка й конвертування зображень.

Конвертер, зручний для редагування зображень і швидкої зміни форматів зображень.

Конвертує з наступних форматів:

- BMP, JPG, GIF, PNG, TIFF, ICO, PSD.
- Додаткові формати, у тому числі які зустрічаються рідко й застарілі.
- Метафайли Windows®: EMF, WMF.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– Формати PostScript: PDF, EPS, PS, AI, EPI.

– Неопрацьовані RAW фотографії із цифрових камер: CRW, CR2, DNG, DCR, NEF, RAW, RAF, X3F, ORF, SRF, MRW, BAY, PEF.

Зберігає в наступні формати:

– JPG, GIF, PNG, TIFF, PDF, BMP, ICO, EMF.

– 20 додаткових графічних форматів.

Пакетна обробка зображень:

– Редагування: поворот, зміна розміру й пропорцій зображення, розмиття, різкість, заміна кольору.

– Фотокорекція: RGB, HLS, HSV, гистограмма, криві, яскравість, контраст, усунення ефекту червоних очей.

– Ефекти: рамки, підпису, декорації, фільтри, підтримка плагінів photoshop®.

Додаткові функції:

– Пакетна обробка зображень.

– Підтримка багатосторінкових файлів.

– Створення зменшених копій зображень.

– Розширені налаштування для перейменування файлів.

– Налаштування прозорості кольору.

– Підтримка CCITT стисків.

– Передача EXIF, IPTC інформації й альфа-каналу.

– Робота в багатозадачному режимі.

– Конвертація з контекстного меню.

– Убудований планувальник.

– Робота з командного рядка.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

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

Розглянемо існуючі системи інтелектуальної системи обробки та конвертації зображень

Dx Optics Pro

У цей момент на стадії тестування перебуває п'ята версія даної утиліти, що стане доступна для придбання найближчим часом. Перевагою цієї версії є повністю перероблена технологія колірної інтерполяції, що, за словами розроблювача, забезпечує кращий результат у плані деталізації й передачі кольору. Чи не так це насправді, можна перевірити на сайті програми, де представлені приклади для порівняння. Інша ключова можливість Dx Optics Pro v5 – принципово новий шумоподавлювач, що працює на ранньому етапі конвертації, зберігаючи більшість деталей.

Інтерфейс простий. Всі інструменти зібрані в лівій частині вікна програм і об'єднані в дев'ять панелей, що згортаються, з яких чотири відповідають за налаштування параметрів: Light, Color, Geometry, Details. Налаштування зберігаються усередині власної бази дані програми. Нова технологія видалення слідів пилу й артефактів працює приблизно так само, як засіб автоматичної корекції червоних очей. Досить клацнути мишкою по ділянці з артефактом, і він відразу ж «замазується» програмою. Вартість версії Pro – \$170, версії Elite – \$300 (розходження між версіями складається в класі сумісних камер – для зеркалок рівня Canon Mark III буде потрібно версія Elite).

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

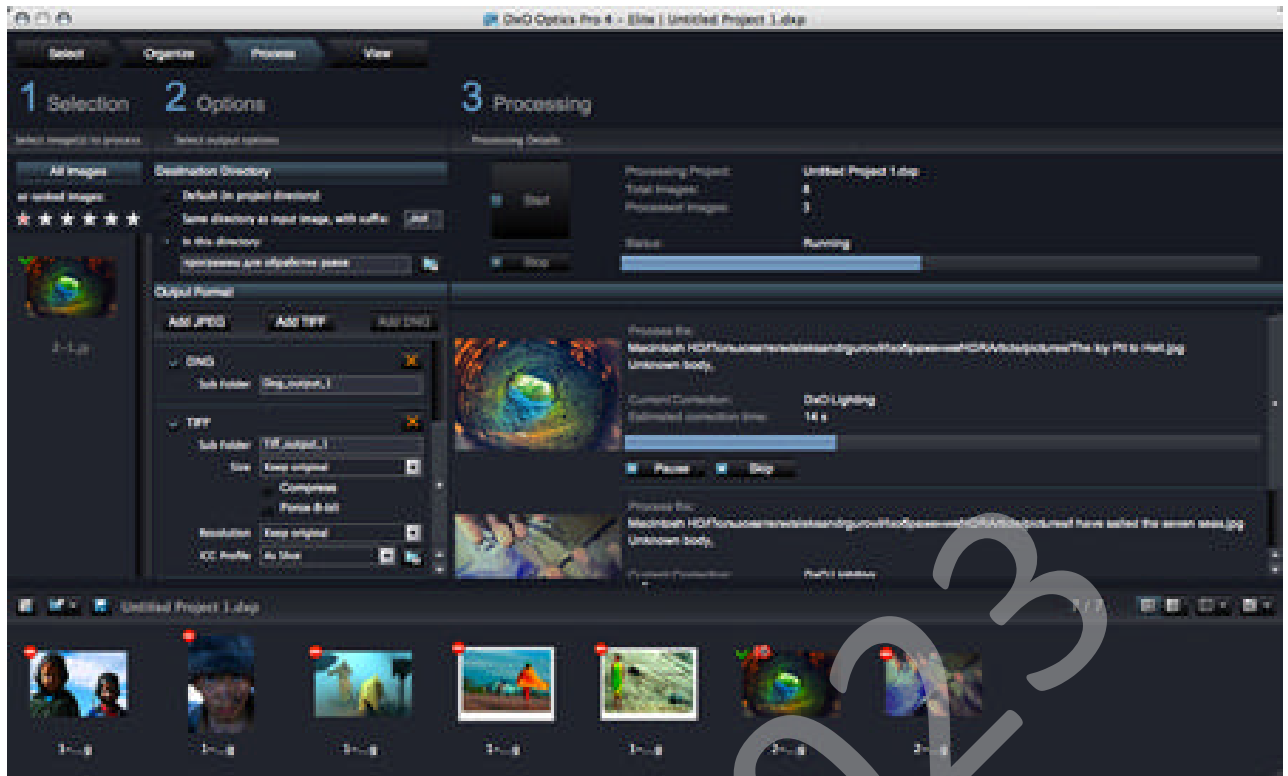


Рисунок 2.1 – Інтерфейс користувача Dx Optics Pro

Після придбання однієї з версій ви можете завантажити модулі для конкретних об'єктивів. Ця особливість програми підвищує ефективність при автоматичній обробці.

Перевага Dx Optics – у результатах пакетної обробки. Творці програми приділили особливу увагу автоматичним функціям, що дозволяють домогтися якісного результату без участі користувача (усунення оптичних перекручувань, параметри кольору, експокорекція).

Lightcrafts LightZone

По революційності LightZone може посперечатися з Aperture. LightZone – програма «повного циклу», що забезпечує розвинені можливості обробки файлів RAW, TIFF, JPEG і DNG. Вартість повної версії для Windows і Mac – \$250, а під Linux програма поширюється безкоштовно.

						ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			12

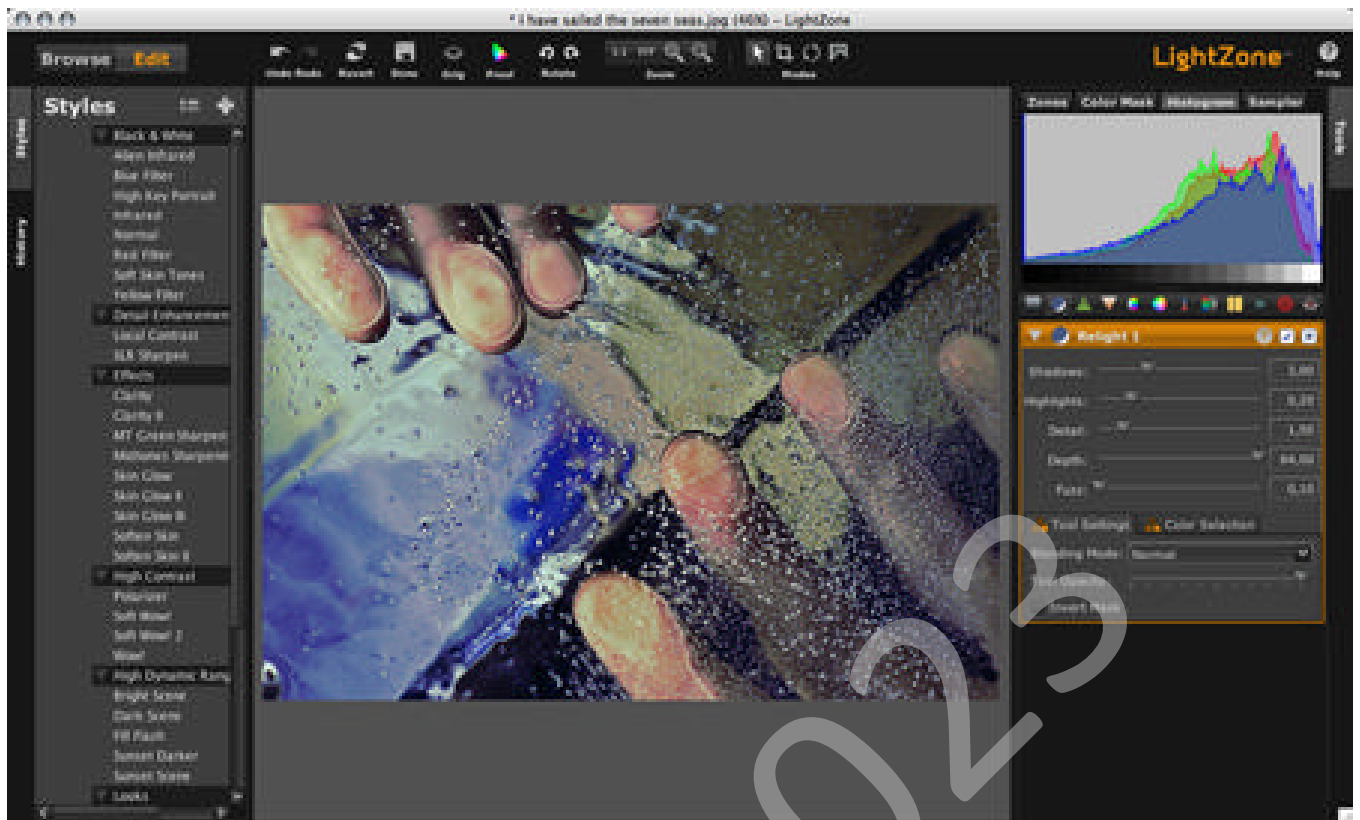


Рисунок 2.2 – Інтерфейс користувача Lightcrafts LightZone

«Візуальний» підхід до процесу роботи із зображеннями дозволяє зосередитися безпосередньо на роботі з ними. Операції здійснюються в реальному часі. Кожну з налаштувань можна скасувати окремо від інших, незалежно від послідовності операцій. Всі налаштування здійснюються в 16-бітному поданні, щоб усунути які-небудь втрати на цьому етапі. LightZone має інструменти регулювання експозиції, насиченості, передачі кольору, балансу білого, різкості (вибірково до різних ділянок по яскравості), видалення артефактів і придушення шуму. Інструмент Relight дозволяє змінити розподіл висвітлення на знімках, вибірково впливаючи на контраст і яскравість окремих ділянок гістограми.

Програма має у своєму розпорядженні потужний інструмент редагування деяких полів метаданих, але з погляду організації роботи з фотоархівами значно поступає таким гігантам, як Aperture і Lightroom. Особливістю LightZone є набір

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

«стилів» обробки: High Dynamic Range (HDR-Обробка), Contrast and Pop (у стилі поп-арт), Lomo Look (ломографія) і інші. Стили – не просто набір конкретних параметрів. Застосовуючи одну із передвстановок, ви запускаєте алгоритм автоматичної обробки, що враховує властивості вихідного файлу, навіть коли мова йде про пакетну обробку даних. Стили можна поєднувати в групи (папки) і навіть обмінюватися ними, як Actions в Photoshop.

Nikon Capture NX

Сама зроблена програма конвертування файлів NEF (також працює з TIFF і JPEG). Входить у комплект професійних дзеркальних камер D300 і D3, але підтримує всі випущені за останні кілька років ЦФК Nikon, здатні знімати у форматі RAW. Ціна – біля \$200.

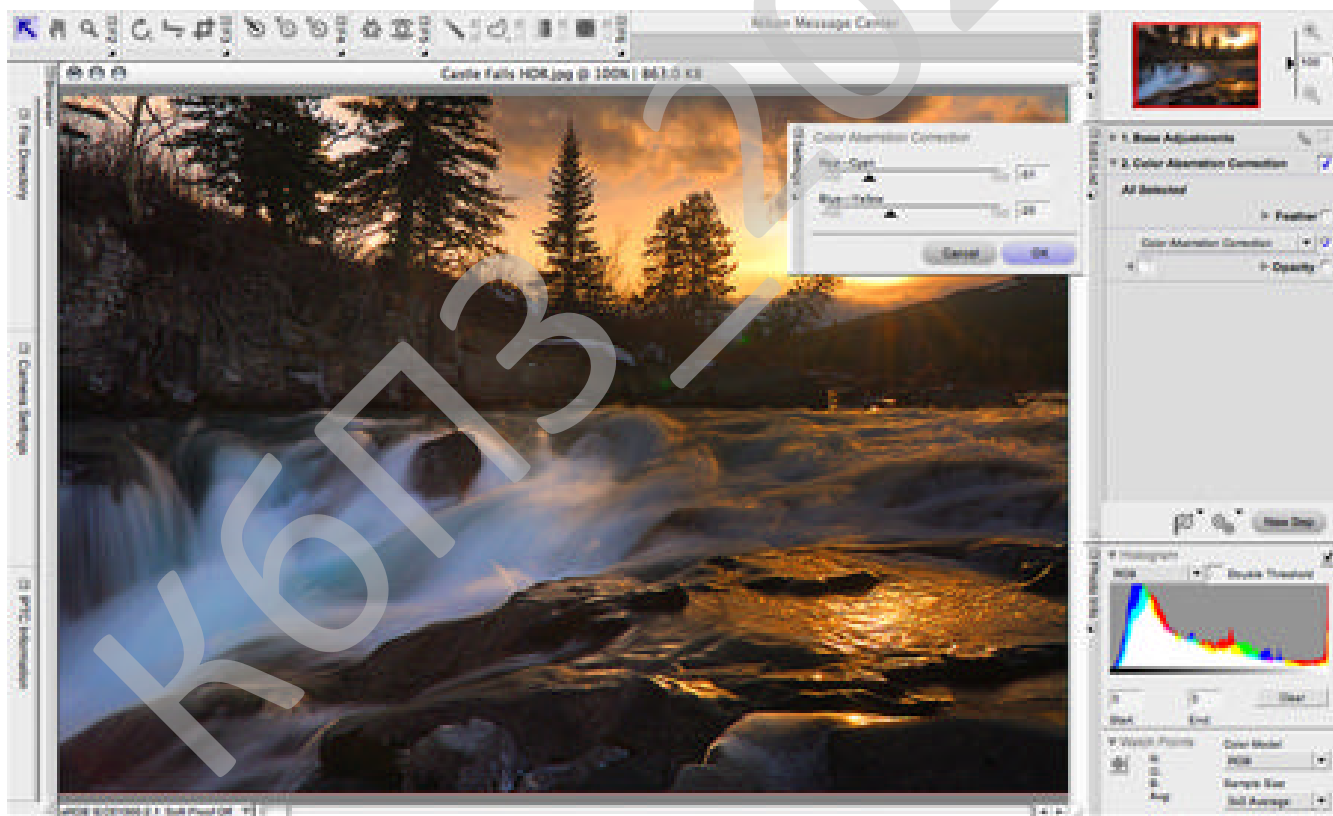


Рисунок 2.3 – Інтерфейс користувача Nikon Capture NX

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Переваги Capture NX – не тільки в чудових результатах обробки. Програма має передовий інтерфейс, заснований на концепції «контрольних крапок» (U-point). Інтерфейс дозволяє відзначити окремі ділянки зображення й виконати складне маскування або обробку шарів. У кожній «крапці» зберігаються обрані для даної області налаштування, пов'язані з корекцією кольори. Робота з «крапками» дозволяє повною мірою реалізувати потенційні переваги, закладені в концепції недеструктивної обробки. Ви можете багаторазово змінювати ті або інші параметри, не зачіпаючи вихідні дані, а також вибірково скасовувати зроблені зміни. Щоб відслідковувати результати корекції, є панель watch points, що відображає параметри кольору в «спостережливих крапках», довільно розставлених на зображенні.

Програма має стандартні засоби корекції оптичних перекручувань (аберації, вин'єтування, ефект бочки). Засіб боротьби із шумом Noise Reduction має сенс застосовувати не тільки по прямому призначенню, але й для «зм'якшення» портретів. Елементи інтерфейсу, включаючи місця, що займають чимало, вікна кривих і рівнів, прибрані в бічні панелі, що згортаються.

Функціональність програми може бути розширена за допомогою модуля фільтрів, що підключається, Color eFex Pro від компанії Nik Multimedia.

Phase One Capture One Pro

Самий улюблений професіоналами RAW-конвертер, але й найдорожчий (\$500 за ліцензію). Не відрізняється настільки широкою підтримкою різних фотокамер, як Adobe Camera RAW або LightZone, але підтримує всі професійні моделі. Споконвічно програма створювалася для дистанційного керування й обробки знімків, отриманих за допомогою цифрових задників для студійних камер компанії Phase One.

Capture One призначений тільки для роботи з RAW і не покликаний замінити графічний редактор і програму для створення фотоархівів. Працює за принципом рендерінгу. Для вибору зображень є файл-браузер, що дозволяє переміщатися по фізичних каталогах на ПК. Набір інструментів стандартний –

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

У вкладці Focus налаштування розподілені по двох групах – Sharpening і Noise. Передбачено два режими «різкості» – Soft Look (для портретів) і Standard. Фільтр шумів простий, уступає спеціалізованим рішенням, начебто Noise Ninja, але зате є засіб боротьби з ефектом бандінга.

При налаштуваннях за замовчуванням результат уступає деяким конкурентам (наприклад, Bibble). Схоже, що Capture One більше націлено на ґрунтовну індивідуальну роботу зі знімками, чим на Batch Process. Це підтверджують не найвищі показники по швидкості пакетної обробки.

Silkipix Developer Studio

Добротна програма від японських розроблювачів. Призначена для фотолюбителів-ентузіастів і підтримує близько 120 моделей ЦФК, включаючи багато просунутих незеркалок. Ціна ліцензії – \$150.

Інструменти для роботи з «сирим» форматом: баланс білого, контраст, різкість, насиченість, корекція оптичних і перспективних перекручувань, зменшення шуму, підрізування, вивід зображень на друк. Особливістю програми є не цей стандартний список інструментів, а їхнє конкретне наповнення. Silkipix несе безліч передвстановок балансу білого (плюс налаштування за допомогою колірного кола, як в Capture One) і підвищення різкості. Фільтр Noise Reduction (NR) припускає тонкі налаштування параметрів (роздільно для кольорних каналів і яскравості), які використовуються для того, щоб урятувати дрібні деталі.

Програма не працює із зображеннями в реальному часі. Принцип недеструктивної обробки реалізований у повному обсязі. Параметри конвертування можна зберігати як індивідуальні переваги користувача (Tastes), а потім застосовувати до інших зображень.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17



Рисунок 2.5 – Інтерфейс користувача Silkpix Developer Studio

У новій версії 3.0 розроблювачі вдосконалили движок конвертера й розширили можливості інструментів Gamma, Tone Curve і Highlight, щоб більш тонко працювати з тінями й світлами. Все це не могло не відбитися на результатах. Silkpix версії 3.0 – один із кращих конвертерів з погляду деталізації зображення. Іншим достоїнством програми є стабільно вірна передача кольору при використанні установок за замовчуванням.

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.

						ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			18

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.

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

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCl, 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.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– 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.

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

Єдине керування пам'яттю

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

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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

Розширена підтримка бібліотек C++

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

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

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.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– 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.

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 стилізуємі FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуальної системи обробки та конвертації зображень.

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.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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.

КБГІЗ-2023

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Обробка зображень – зміна оригіналу зображення класичними або цифровими методами. Також може позначатися терміном ретушування, ретуш. Метою обробки є корекція дефектів, підготовка до публікації, рішення творчих завдань.

Крім статичних двомірних зображень, обробляти потрібно також послідовності зображень.

Джерела зображень:

– Зображення із цифрового фотоапарата може бути скопійоване прямо в комп'ютер для обробки. Переваги – швидкість і оперативність. Недоліки – цифровий шум, висока вартість професійних рішень.

– Негативні фотоплівки й слайди після оцифровки за допомогою сканера можна обробляти на комп'ютері. Перевагою такого зображення є широкий динамічний діапазон, відсутність цифрового шуму. Недолік – зернистість плівки, звичайно низька якість сканування (одержати зображення із плівки, порівнянне по якості із зображенням із професійної цифрової камери, можна тільки на дорогому професійному сканері). Із широкоформатних негативів і слайдів можна одержати зображення дуже великого розміру й високої якості.

– Друковані оригінали, поліграфічні відбитки, надруковані фотографії після перекладу в цифровий вид за допомогою сканера, можна обробляти на комп'ютері. Недоліки – малий динамічний діапазон, у поліграфічних відбитків – растр, що може провокувати утворення муару.

– Фотобанки – більші сховища цифрових і аналогових зображень.

– Сервери файлообміну й пошукові системи. На цих ресурсах нерідко можна зустріти зображення без обмежень на використання.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Види й мети обробки зображень:

1. Усунення дефектів зображення:

- шум (випадкові погрішності кольору в кожній крапці зображення);
- недостатня або надлишкова яскравість;
- недостатня або надлишкова контрастність (вуаль або надлишковий динамічний діапазон зображення);
- неправильний колірний тон;
- нерізкість (розфокусування);
- пил, подряпини, «биті пікселі»;
- усунення дисторсії й виньєтування об'єктива.

2. Структурна обробка зображень

- кадрування;
- створення панорам;
- усунення непотрібних деталей зображення, зміна композиції;
- фотомонтаж – створення із частин декількох зображень нового зображення;
- домальовування, включення в зображення технічних креслень, написів, символів, покажчиків та ін.;
- застосування спецефектів, фільтрів, тіней, фонів, текстур, підсвічування.

3. Підготовка фотографій до публікації в пресі, на телебаченні, в Інтернеті.

У кожного пристрою виводу (монітор, принтер, офсетна друкована машина й т.п.) є свої можливості по колірному охопту (не будь-який колір можна відтворити). Наприклад, на папері співвідношення по світу між білим і чорним досягає 40, у той час як у слайда воно більше 200. Основним завданням є передати задум автора з найменшими втратами. Виконується конвертації кольору, наприклад, у випадку друку на паперовому носії, визначення кількості фарби для передачі кожного кольору.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Фахівець, підготовляючи фотографії до публікації, діючи творчо, як художник, або використовуючи стандартні методи, приводить зображення до виду, що відповідає технічним можливостям процесу, що репродукує, при максимальному збереженні ідеї зображення.

Обробка зображень цифровими методами

Сьогодні обробка зображень проводиться в основному на комп'ютері растровими редакторами в цифровому виді. Для цього зображення, навіть отримане із традиційного носія (плівки), переводиться в цифровий вид – наприклад, за допомогою сканера.

Досвід показує, що оригінал оброблюваного зображення повинен бути по можливості збережений. Копії можна редагувати як завгодно – це будуть копія 1, копія 2, копія 3 і т.д.

Інструменти технічного обробки цифрових зображень

Для обробки зображень застосовуються різні програми, однак існують основні можливості й алгоритми роботи програм і оператора. За допомогою більшості графічних редакторів можна:

– Виділяти фрагмент зображення для обробки. У більшості програм використовується метод обробки зображення вроздріб. Спочатку частина зображення виділяється, після чого робота ведеться тільки з нею, не торкаючись залишку зображення. Виділення певних ділянок зображення можна реалізувати як вказівка контуру (наприклад, інструмент ласо), так і з використанням масок, що редагуються. Останній варіант надає більше можливостей. Виділену частину зображення звичайно можна також рухати, обертати, масштабувати, деформувати, домальовувати й т.п.

– Виділення може бути як тимчасове, так і постійне – виділена частина зображення в різних графічних редакторах може бути оформлена як постійний «шар» або «об'єкт». Це дозволяє розбивати зображення на фрагменти, які накладаються один на одного, і модифікувати кожний з них окремо.

– Вибирати алгоритм, що програма застосує до всього зображення, групі зображень, виділеному фрагменту або об'єкту.

Інструменти структурного обробки цифрових зображень:

– Зміна розміру зображення, кадрування. Розмір зображення може бути змінений до необхідного за допомогою математичних алгоритмів, які вираховують колір пікселів виходячи з кольору пікселів оригіналу. Варто враховувати, що при збільшенні зображень губиться різкість, при зменшенні – деталізація. Часто композицію зображення можна також поліпшити, видаливши зайві області по краях. Це називається кадруванням.

– Коллажування (фотомонтаж). Створення із частин декількох зображень нового зображення. Має майже таку ж історію розвитку, як і фотографія. У цей час часте застосовується не тільки професійними, але й простими людьми.

– Обтравка. Обтравкою називається процес виділення якогось об'єкта на зображенні з метою його відділення від тла.

– Придушення шуму. У програмах обробки зображення присутні різні алгоритми для видалення або зменшення шуму. Це в першу чергу цифровий шум матриці цифрової фотокамери. У подібній корекції також можуть бідувати зерно плівки, артефакти стиску, пил і подряпини на оригіналі.

– Ретуш. У вузькому змісті ретуш – усунення непотрібних деталей зображення, шумів, зміна композиції. Часто до ретуші доводиться прибїгати для того, щоб забрати дефекти шкіри, пил на одязі моделі.

– Орієнтація зображення. Програми обробки можуть повертати зображення під будь-яким кутом або відбивати дзеркально.

– Фільтри й спецефекти. Фільтри й спецефекти використовуються для додання зображенню незвичайного виду. За допомогою фільтрів картинка може бути перекручена незвичайним образом, стилізована, може бути додана видимість об'ємного рельєфу, змінені кольори.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Розширення динамічного діапазону зображення

Розширення динамічного діапазону зображення шляхом комбінування фотографій одного сюжету, отриманих з різною експозицією. Для цього робиться кілька кадрів з одним положенням камери (бажано зі штатива). При обробці, зображення з різною експозицією поєднуються в одне. Це дозволяє зафіксувати висококонтрастний сюжет без вибілювання яскраво освітлених об'єктів і без неприпустимого зашумлення темних деталей.

Кольорокорекція

Кольорокорекція – внесення змін у колір оригіналу. Багато хто відносять до кольорокорекції ті процедури, які не пов'язані зі зміною сюжету зображення. У більш вузькому змісті кольорокорекція – це така конвертація зображення, об'єкта або фрагмента, коли новий колір оброблюваного пікселя залежить від старого значення цього пікселя й не залежить від сусідніх пікселів. Основна причина, по якій доводиться виконувати корекцію кольору, що впливає: людське око має здатність адаптуватися до сили й спектральних характеристик висвітлення таким чином, що зберігається сприйняття кольору предметів у більшості випадків незалежно від спектрального состава висвітлення, камера ж фіксує світлове випромінювання без адаптації й, при перегляді в інших умовах, фотографії іноді сильно відрізняються від того, що ми бачили, коли фотографували. Для усунення цієї проблеми у фотографії використовуються алгоритми вибору й налаштування білого кольору. Ці алгоритми вже можна назвати кольорокорекцією. Інші причини застосування кольорокорекції: недостатній або надлишковий контраст зображення, вуаль, вицвітання зображення. Також необхідно відзначити кольорокорекцію з метою внесення гармонії між фотозображеннями й дизайном публікації або змістом матеріалів, подібну до процесу фотодруку у фотографічному мистецтві.

Як і будь-який інший метод, кольорокорекцію можна застосувати як інструмент реалізації творчого задуму або побажань замовника.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Цифрове зображення завжди представлене в якійсь колірній моделі (Red Green Blue, Lab, і ін.), що припускає трохи (три, як правило) характеристик для кожної крапки зображення (пікселя). Характеристики всіх крапок зображення називають каналами. Наприклад, у моделі RGB кожний піксель характеризується значенням яскравості червоної, зеленої й синьої складових його кольору. Відповідно, у зображенні можна виділяти канали червоної, зеленої й синьої квіток. Функції можуть бути задані як незалежні для кожного каналу, так і більше складні – наприклад, «Змішання каналів» (Channel Mixer).

Методи конвертації кольори можуть бути самими різними, однак найбільше часто використовуваними методами кольорокорекції є наступні:

1. Вказівка аргументів для функцій конвертації вхідних значень у вихідні. У програмах ці інструменти називаються «рівні», «гама» і т.п. Іноді набори значень аргументів вибираються з переліку заздалегідь заданих варіантів. До цього виду перетворень можна віднести так само:

– Установка балансу білого (облік висвітлення) при перетворенні електронного сигналу матриці у файл зображення або скануванні плівки.

– Конвертації, що безпосередньо задають зміни контрастності, яскравості, гама, тону, світу, насиченості зображення або його частин.

2. Безпосереднє завдання графіків конвертації значень по каналах. Цей інструмент звичайно називається «Криві» (Curves). Він дозволяє виконати будь-які конвертації усередині кожного каналу шляхом ручного формування графіка, аналогічного тим, які обчислюються функціональними алгоритмами по заданих аргументах. При тім, що функції типових перетворень – рівнів, контрастності, яскравості, гама й т.п. – прості й досить зрозумілі, інструмент «Криві» здатний виявитися гнучкіше й наочніше окремих функціональних перетворень.

Зміна колірних просторів (кольороподіл)

Для різних цілей (наприклад, відображення на екрані комп'ютера й друк на папері) використовуються різні способи відтворення зображень і різних математичних моделей, що описують колір (колірні простори) залежно від

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

способу відтворення. Програми обробки зображень здатні конвертувати зображення з одного колірного простору в інше.

Основне завдання підготовки до публікації – привести зображення до вимог технічного процесу, максимально зберігши при цьому саме зображення. Наприклад, при підготовці до офсетного друку необхідно провести колірне конвертації в колірний простір друку (найчастіше – СМҮК), забезпечити відсутність перевищення сумарної щільності фарби й «білих плям», тобто ділянкою, де мінімальний зміст фарби менше мінімально відображуваного даним друкованим процесом, скорегувати зображення для того, щоб нейтральні кольори були передані певними для даного друкованого процесу сполученням фарб, попередити зниження різкості в процесі зміни растра під новий техпроцес (наприклад, с використання нерізкого маскування).

У широкому змісті, обробка зображень – це будь-яка форма обробки інформації, для якої входом є зображення, наприклад, фотографії або відеокадри. Тому термін «Обробка зображень» є окремою подією терміна «обробка зображень». Обробка зображень – зміна деталей оригінального зображення (у цей час, в основному, цифровими методами).

Обробка фотографічних зображень широко застосовується в космічній фоторозвідці для розпізнавання військових об'єктів і зброї ймовірного супротивника. При розпізнаванні військових об'єктів і зброї ймовірного супротивника обробка зображень украй небажана.

Більшість методів обробки зображень представляють зображення як двовимірні сигнали, застосовуючи до них стандартні методи обробки сигналів.

Методи цифрової обробки зображень є більше точними, надійні, гнучкі й простими в реалізації, ніж звичайні методи. У цифровій обробці й редагуванні зображень широко застосовується спеціалізоване встаткування, таке як процесори з конвеєрною обробкою інструкцій і багатопроцесорні системи. В особливій мері це стосується систем обробки відео. Проте, стандартні завдання обробки зображень найчастіше можуть бути вирішені й на персональному комп'ютері.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3.2 Розробка структурної схеми

Структурна схема системи складається з наступних блоків.

1. Блок алгоритмів перетворень різних форматів:

– Формат JPEG – один з популярних графічних форматів, застосовуваний для зберігання фотозображень і подібних їм зображень. Файли, що містять дані JPEG, звичайно мають розширення .jpeg, .jfif, .jpg, .JPG, або .JPE. Однак з них .jpg саме популярне розширення на всіх платформах. MIME-типом є image/jpeg.

– Формат JPEG 2000 – графічний формат, що замість дискретного косинусного конвертації, застосовуваного у форматі JPEG, використовує технологію вейвлет-конвертації, що ґрунтується на поданні сигналу у вигляді суперпозиції базових функцій – хвильових пакетів. У результаті такої компресії зображення виходить більше гладкий і чітким, а розмір файлу в порівнянні з JPEG при однаковій якості виявляється меншим. JPEG 2000 повністю вільний від головного недоліку свого попередника: завдяки використанню вейвлетів, зображення, збережені в цьому форматі, при високих ступенях стиску не містять артефактів у вигляді «ґрат» із блоків розміром 8x8 пікселів. Формат JPEG 2000 також, як і JPEG, підтримує так званий «прогресивний стиск», що дозволяє в міру завантаження бачити спочатку розмите, але потім усе більше чітке зображення. Поки цей формат мало розповсюджений і підтримується не всіма сучасними браузерами.

– Формат BMP – формат зберігання растрових зображень. Споконвічно формат міг зберігати тільки апаратно-залежні растри (DDB), але з розвитком технологій відображення графічних даних формат BMP став переважно зберігати апаратно-незалежні растри (DIB). З форматом BMP працює величезна кількість програм, тому що його підтримка інтегрована в операційні системи Windows і OS/2. Файли формату BMP можуть мати розширення .bmp, .dib і .rle. Крім того, дані цього формату включаються у двійкові файли ресурсів RES і в PE-файли.

– Формат РСХ – стандарт подання графічної інформації. Використовувався графічною програмою ZSoft PC Paintbrush для MS-DOS компанії Microsoft, текстових процесорів і настільних видавничих систем типу Microsoft Word і Ventura Publisher, розроблений компанією ZSoft Corporation. Не настільки популярний аналог BMP, хоча підтримується специфічними графічними редакторами, такими, як Adobe Photoshop, Corel Draw і ін. У цей час витиснутий форматами, які підтримують кращий стиск: GIF, JPEG і PNG. Тип формату – растровий. Більшість файлів такого типу використовує стандартну палітру кольорів, але формат був розширений з розрахунку на зберігання 24-бітних зображень. РСХ – апаратно-залежний формат. Призначається для зберігання інформації у файлі в такому ж виді, як і у відеоплаті. Для сумісності зі старими програмами необхідна підтримка EGA-режиму відеоконтролером. Алгоритм такого стиску дуже швидкий і займає невеликий обсяг пам'яті, однак не дуже ефективний, непрактичний для стиску фотографій і більше детальної комп'ютерної графіки.

– Формат GIF – популярний формат графічних зображень. Здатний зберігати стислі дані без втрати якості у форматі не більше 256 кольорів. Незалежний від апаратного забезпечення формат GIF був розроблений в 1987 році (GIF87a) фірмою CompuServe для передачі растрових зображень по мережах. В 1989-му формат був модифікований (GIF89a), були додані підтримка прозорості й анімації. GIF використовує LZW-компресію, що дозволяє непогано стискати файли, у яких багато однорідних заливань (логотипи, написи, схеми).

– Формат ICO – формат зберігання значків файлів в Microsoft Windows. Формат ICO аналогічний формату CUR (Windows cursors), призначеному для зберігання курсорів. Відмінність складається в чисельному значенні одного поля в заголовній структурі, і інтерпретації значень двох інших полів цієї ж структури. Один ICO-файл містить один або декілька значків, розмір і кольоровість кожного з яких задається окремо. Розмір значка може бути будь-яким, але найбільше вживані квадратні значки зі стороною 16, 32 і 48 пікселів. Також

						ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			34

використовуються значки з розміром 24, 40, 60, 72, 92, 108, 128 пікселів. Починаючи з Windows 98/2000, формат підтримує впровадження зображень у форматі JPEG і PNG, але звичайно дані значків зберігаються в незжатому виді. Значки бувають у природному кольорі (True Color, глибина кольору 24 біт), High Color (глибина кольору 16 біт), або з фіксованою палітрою (із двохсот п'ятдесятьох шести, шістнадцяти, або всього із двох кольорів). У цьому випадку число, що відповідає кожному пікселю вказує не на колір, а на номер кольору в палітрі. По своїй структурі зображення у файлі ICO найбільш близькі до BMP, але принципово відрізняються від них наявністю додаткового зображення – маски, що накладається на задній план за допомогою операції побітового «І», що дозволяє реалізувати (повну) прозорість рисунка. Наступне накладення основного зображення за допомогою «АБО що виключає» може навіть дати «інверсні» піксели в тих місцях, де задній план не був замаскований. Крім того, починаючи з Windows XP підтримуються 32-бітні значки – кожному пікселю відповідає 24-біта кольору й 8-бітний альфа-канал, що дозволяє реалізувати 256 рівнів часткової прозорості. За допомогою альфа-каналу можна відображати значки зі згладженими (розмитими) краями й тінню, що сполучаються з будь-яким тлом; маска в цьому випадку ігнорується.

– Формат PNG – був розроблений як альтернатива GIF формату. У відмінності від останнього він дозволяє зберігати всю колірну інформацію й інформацію про альфа-канали масок і шарів. Це надзвичайно стислий RGB-формат, у якому використовується схема стиску без втрат. На сьогоднішній день це найбільш перспективний Web-формат, що володіє потужними можливостями, які найбільше яскраво проявляються при його зіставленні з форматом GIF. В PNG форматі реалізований відкритий, не запатентований алгоритм стиску даних, що забезпечує більше високі результати. У відмінності від обмеженої 256 кольорами палітри GIF, формат PNG дозволяє зберігати повнокольорові зображення з колірною глибиною 24 біт/пікселя і навіть 48 біт/пікселя.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

спеціалізованій програмі – графічній програмі або текстовому редактору, САПР і т.д., а потім експортується у формат PDF для поширення в електронному виді, передачі в друкарню й т.п. PDF з 1 липня 2008 року є відкритим стандартом ISO 32000. Формат PDF дозволяє впроваджувати необхідні шрифти (порядковий текст), векторні й растрові зображення, форми й мультимедіа-вставки. Підтримує RGB, CMYK, Grayscale, Lab, Duotone, Bitmap, кілька типів стиску растрової інформації. Має власні технічні формати для поліграфії: PDF/ X-1, PDF/ X-3. Включає механізм електронних підписів для захисту й перевірки дійсності документів. У цьому форматі поширюється велика кількість супутньої документації.

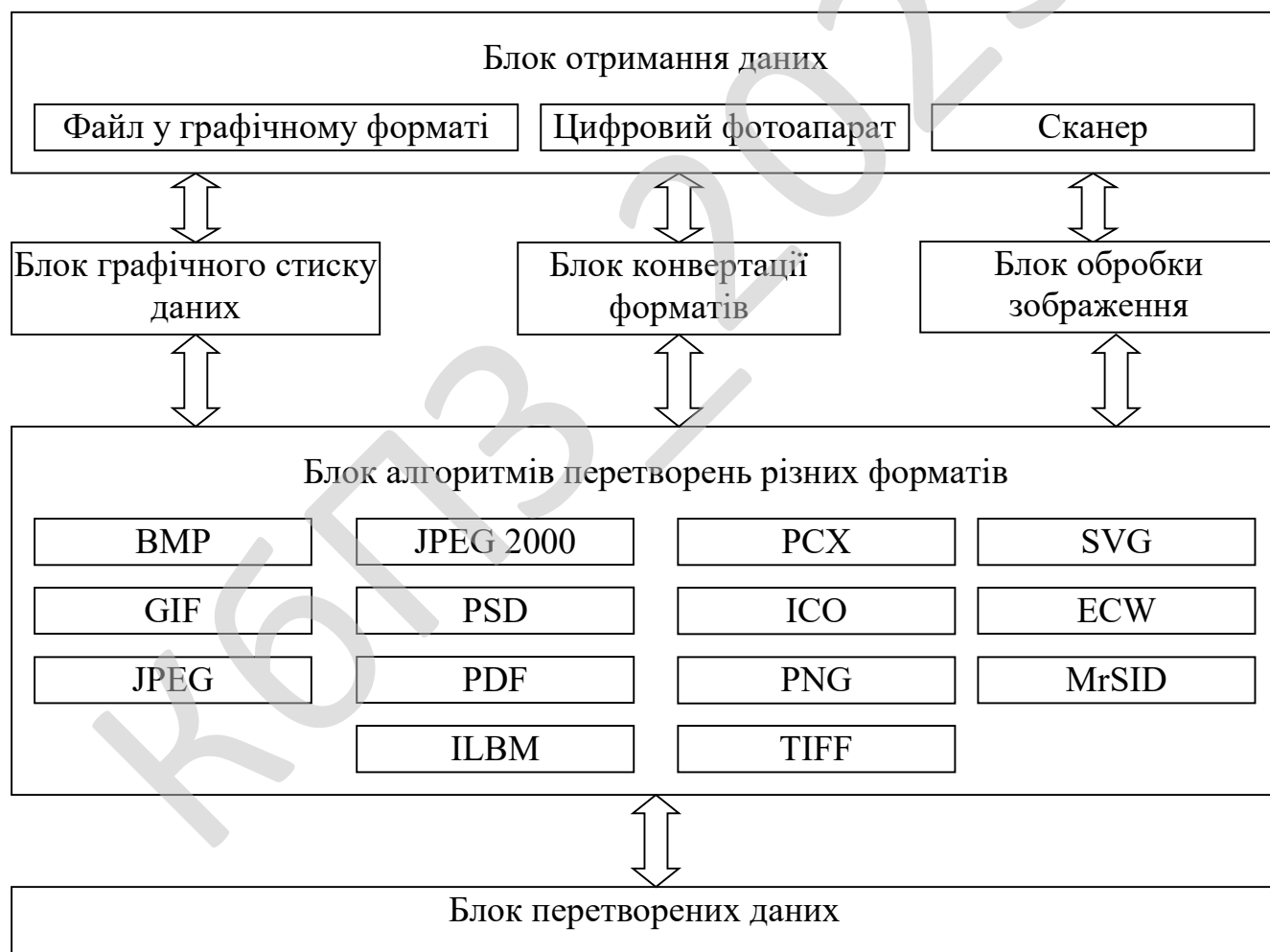


Рисунок 3.1 – Структурна схема системи

нової матриці коефіцієнтів. Для одержання вихідного зображення застосовується зворотне конвертації.

ДКП розкладає зображення по амплітудах деяких частот, таким чином, при перетворенні одержуємо матрицю, у якій багато коефіцієнтів або близькі, або дорівнюють нулю. Крім того, людська система колірною сприйняття слабо розпізнає певні частоти. Тому можна апроксимувати деякі коефіцієнти більш грубо без помітної втрати якості зображення.

Для цього використовується квантування коефіцієнтів (quantization). У найпростішому випадку – це арифметичне побітове зрушення вправо. При цьому перетворенні губиться частина інформації, але можуть досягатися більші коефіцієнти стиску.

Отже, розглянемо алгоритм докладніше. Нехай ми стискаємо 24-бітне зображення.

Крок 1.

Переводимо зображення з колірною простору RGB, з компонентами, відповідальними за червону (Red), зелену (Green) і синю (Blue) складового кольору крапки, у колірний простір YCrCb (іноді називають YUV).

У ньому Y – ярісна складова, а Cr, Cb – компоненти, відповідальні за колір (хроматичний червоний і хроматичний синій). За рахунок того, що людське око менш чутливе до кольору, чим до яскравості, з'являється можливість архівувати масиви для Cr і Cb компонент із більшими втратами й, відповідно, більшими коефіцієнтами стиску. Подібне конвертації вже давно використовується в телебаченні. На сигнали, відповідальні за колір, там виділяється більше вузька смуга частот.

Спрощено переклад з колірною простору RGB у колірний простір YCrCb можна представити так:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}. \quad (3.1)$$

Зворотне конвертації здійснюється множенням вектора YUV на зворотну матрицю.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{pmatrix} * \left(\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} - \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \right). \quad (3.2)$$

Крок 2.

Розбиваємо вихідне зображення на матриці 8x8. Формуємо з кожної три робочі матриці ДКП – по 8 біт окремо для кожного компонента. При великих коефіцієнтах стиску, цей крок може виконуватися ледве складніше. Зображення ділиться по компоненті Y – як і в першому випадку, а для компонентів Cr і Cb матриці набираються через рядок і через стовпець. Тобто з вихідної матриці розміром 16x16 виходить тільки одна робоча матриця ДКП. При цьому, як неважко помітити, ми втрачаємо 3/4 корисної інформації про колірні складові зображення й одержуємо відразу стиск у два рази. Ми можемо поступати так завдяки роботі в просторі YCrCb. На результуючому RGB зображенні, як показала практика, це позначається не сильно.

Крок 3.

Застосовуємо ДКП до кожної робочої матриці. При цьому ми одержуємо матрицю, у якій коефіцієнти в лівому верхньому куті відповідають низькочастотній складовій зображення, а в правому нижньому – високочастотній.

У спрощеному виді це конвертації можна представити так:

$$Y[u, v] = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C(i, u) \times C(j, v) \times y[i, j], \quad (3.3)$$

де:

$$C(i, u) = A(u) \times \cos\left(\frac{(2 \times i + 1) \times u \times \pi}{2 \cdot n}\right), \quad (3.4)$$

$$A(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } u \equiv 0 \\ 1, & \text{for } u \neq 0 \end{cases}, \quad (3.5)$$

Крок 4.

Робимо квантування. У принципі це просто розподіл робочої матриці на матрицю квантування поелементно. Для кожного компонента (Y , U і V), у загальному випадку, задається своя матриця квантування $q[u,v]$ (далі МК).

$$Yq[u,v] = \text{IntegerRound} \left(\frac{Y[u,v]}{q[u,v]} \right), \quad (3.6)$$

На цьому кроці здійснюється керування ступенем стиску, і відбуваються самі більші втрати. Зрозуміло, що, задаючи МК із більшими коефіцієнтами, ми одержимо більше нулів і, отже, більший ступінь стиску.

У стандарт JPEG включені рекомендовані МК, побудовані дослідним шляхом. Матриці для більшого або меншого коефіцієнтів стиску одержують шляхом множення вихідної матриці на деяке число γ .

Із квантуванням зв'язані й специфічні ефекти алгоритму. При великих значеннях коефіцієнта γ втрати в низьких частотах можуть бути настільки великі, що зображення розпадеться на квадрати 8×8 . Втрати у високих частотах можуть виявитися в так званому “ефекті Гіббса”, коли навколо контурів з різким переходом кольору утвориться своєрідний “німб”.

Крок 5.

Переводимо матрицю 8×8 в 64-елементний вектор за допомогою “зигзаг”-сканування, тобто беремо елементи з індексами $(0,0)$, $(0,1)$, $(1,0)$, $(2,0)$...

Таким чином, на початку вектора ми одержуємо коефіцієнти матриці, що відповідають низьким частотам, а наприкінці – високим.

Крок 6.

Згортаємо вектор за допомогою алгоритму групового кодування. При цьому одержуємо пари типу (пропустити, число), де “пропустити” є лічильником нулів, що пропускаються, а “число” – значення, яких необхідно поставити в наступний осередок. Так, вектор $42\ 3\ 0\ 0\ 0\ -2\ 0\ 0\ 0\ 0\ 1...$ буде згорнутий у пари $(0,42)\ (0,3)\ (3,-2)\ (4,1)...$

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

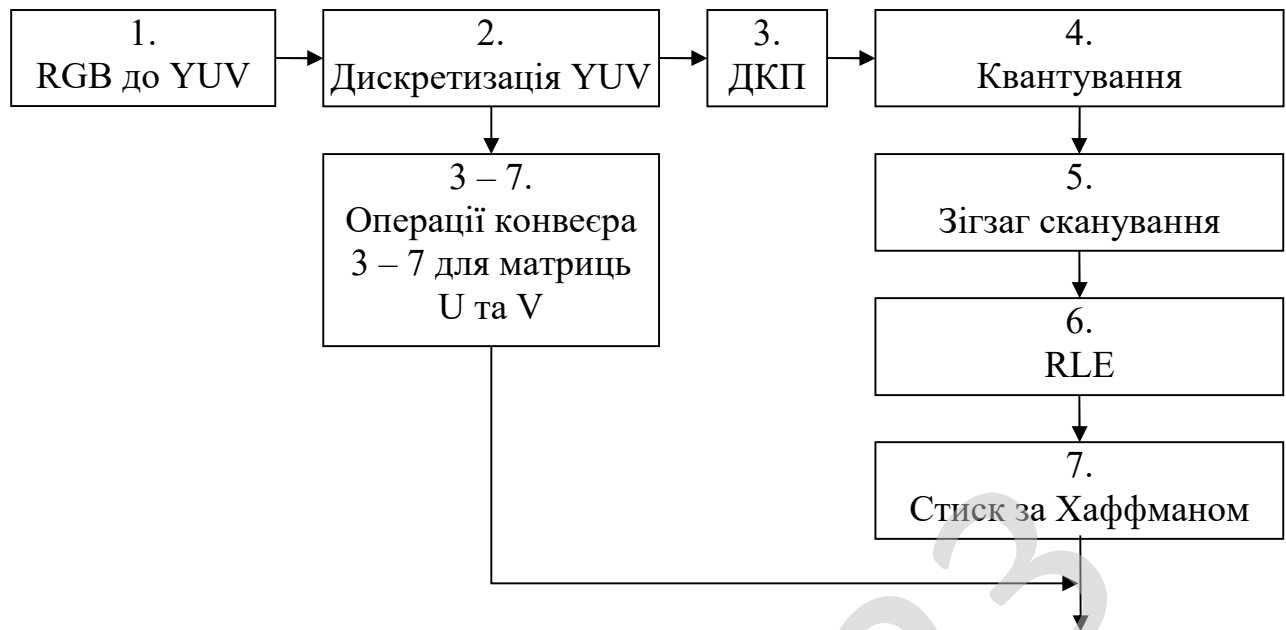


Рисунок 3.3 – Конвеєр операцій, використовуваний в алгоритмі JPEG

Істотними позитивними сторонами алгоритму є те, що:

- Задається ступінь стиску.
- Вихідне кольорове зображення може мати 24 біта на крапку.

Негативними сторонами алгоритму є те, що:

– При підвищенні ступеня стиску зображення розпадається на окремі квадрати (8x8). Це пов'язане з тим, що відбуваються більші втрати в низьких частотах при квантуванні, і відновити вихідні дані стає неможливо.

– Проявляється ефект Гіббса – ореоли по границях різких переходів кольорів.

BMP

BMP – формат зберігання растрових зображень. Споконвічно формат міг зберігати тільки апаратно-залежні растри (DDB), але з розвитком технологій відображення графічних даних формат BMP став переважно зберігати апаратно-незалежні растри (DIB).

З форматом BMP працює величезна кількість програм, тому що його підтримка інтегрована в операційні системи Windows і OS/2. Файли формату

BMP можуть мати розширення.bmp,.dib і.rle. Крім того, дані цього формату включаються у двійкові файли ресурсів RES і в PE-файли.

Глибина кольору в даному форматі може бути 1, 2, 4, 8, 16, 24, 32, 48 біт на піксел, максимальні розміри зображення 65535x65535 пікселів. Однак, глибина 2 біт офіційно не підтримується.

У форматі BMP є підтримка стиску по алгоритму RLE, однак тепер існують формати з більше сильним стиском, і через великий обсяг BMP рідко використовується в Інтернеті, де для стиску без втрат використовуються PNG і більше старий GIF

Даний алгоритм надзвичайно простий у реалізації. Групове кодування (RLE) – один із самих старих і найпростіших алгоритмів архівації графіки. Зображення в ньому (як і в декількох алгоритмах, описаних нижче) витягається в ланцюжок байт по рядках растра. Сам стиск в RLE відбувається за рахунок того, що у вихідному зображенні зустрічаються ланцюжки однакових байт. Заміна їх на пари <лічильник повторень, значення> зменшує надмірність даних.

Алгоритм декомпресії, реалізований на псевдокоді, при цьому виглядає так:

```
Function Initialization;
do
  byte:= ImageFile.ReadNextByte;
  if (є лічильником (byte)) then
    begin
      counter:= Low6bits (byte) +1;
      value:= ImageFile.ReadNextByte;
      for i:=1 to counter do
        DecompressedFile.WriteByte (value)
      end
    else
      DecompressedFile.WriteByte (byte)
  while (ImageFile.EOF);
```

У даному алгоритмі ознакою лічильника (counter) служать одиниці у двох верхніх бітах зчитаного файлу. Відповідно 6 біт, що залишилися витрачаються на

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45


```

CurStr:=CurStr+C; //Приклеїти символ до рядка
else begin
    code:=CodeForString(CurStr); // code-не байт!
    CompressedFile.WriteCode (code);
    AddStringToTable (CurStr+C);
    CurStr:=C; // Рядок з одного символу
end
end
code:=CodeForString(CurStr);
CompressedFile.WriteCode (code);
CompressedFile.WriteCode (CodeEndOfInformation);

```

Як говорилося вище, функція InitTable ініціалізує таблицю рядків так, щоб вона містила всі можливі рядки, що складаються з одного символу. Наприклад, якщо ми стискаємо байтові дані, то таких рядків у таблиці буде 256 (“0”, “1”,..., “255”). Для коду очищення (ClearCode) і коду кінця інформації (CodeEndOfInformation) зарезервовано значення 256 і 257. У розглянутому варіанті алгоритму використовується 12-бітний код і, відповідно, під коди для рядків нам залишаються значення від 258 до 4095. Рядки, що додаються, записуються в таблицю послідовно, при цьому індекс рядка в таблиці стає її кодом.

Функція ReadNextByte читає символ з файлу. Функція WriteCode записує код (не рівний по розмірі байту) у вихідний файл. Функція AddStringToTable додає новий рядок у таблицю, приписуючи їй код. Крім того, у даній функції відбувається обробка ситуації переповнення таблиці. У цьому випадку в потік записується код попереднього знайденого рядка й код очищення, після чого таблиця очищається функцією InitTable. Функція CodeForString знаходить рядок у таблиці й видає код цього рядка.

Особливість LZW полягає в тому, що для декомпресії нам не треба зберігати таблицю рядків у файл для розпакування. Алгоритм побудований таким чином, що ми в стані відновити таблицю рядків, користуючись тільки потоком кодів. Ми знаємо, що для кожного коду треба додавати в таблицю рядок, що складається із уже присутнього там рядка й символу, у який починається наступний рядок у потоці.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Алгоритм декомпресії, що здійснює цю операцію, реалізований на псевдокодi, виглядає в такий спосiб:

```
code:=File.ReadCode;
while (code <> CodeEndOfInformation) begin
  if (code:= Clearcode) then begin
    InitTable;
    code:=File.ReadCode;
    if (code:= CodeEndOfInformation) then
      begin
        закінчити роботу
      end;
    ImageFile.WriteString(StrFromTable(code));
    old_code:=code;
  else begin
    if (InTable(code)) then begin
      ImageFile.WriteString(FromTable(code));
      AddStringToTable(StrFromTable(old_code)+
        FirstChar(StrFromTable(code)));
      old_code:=code;
    end
    else begin
      OutString:= StrFromTable(old_code)+
        FirstChar(StrFromTable(old_code));
      ImageFile.WriteString(OutString);
      AddStringToTable(OutString);
      old_code:=code;
    end
  end
end
end
```

Тут функція ReadCode читає черговий код з декомпресуємого файлу. Функція InitTable виконує ті ж дії, що й при компресії, тобто очищає таблицю й заносить до неї всі рядки з одного символу. Функція FirstChar видає нам перший символ рядка. Функція StrFromTable видає рядок з таблиці по коду. Функція AddStringToTable додає новий рядок у таблицю (привласнюючи їй перший вільний код). Функція WriteString записує рядок у файл.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

3.3 Розробка функціональної схеми

На рисунку 3.4 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функціональна схема складається з наступних блоків:

1. Головне вікно інтерфейсу користувача.
2. Панель швидкого доступу до елементів програми:
 - Відкрити зображення.
 - Перетворити зображення.
 - Зберегти зображення.
 - Налаштування зображення.
 - Збільшити зображення.
 - Зменшити зображення.
 - Роздрукувати зображення.
 - Повернути зображення.
 - Розтягти зображення.
 - Інформація про розроблювача.
3. Панель меню користувача:
 - Файл.
 - Виправлення.
 - Зображення.
 - Ефекти.
 - Вид.
 - Масштаб.
 - Утиліти.
 - Вибране.
 - Довідка.



Рисунок 3.4 – Функціональна схема системи

4. Блок функцій по налаштуванню зображень:
 - зміна яскравості зображення;
 - зміна колірного дозволу;
 - подання зображення кольоровим або чорно-білим;
 - зміна контуру;
 - налаштування контрастності;
 - налаштування розширення;
 - видалити непотрібні пікселі;
 - змінити кількість пікселів на дюйм.
5. Блок функцій по конвертуванню графічних форматів з одного в інший.
6. Конвертація прямо з буфера обміну.
7. Зняття скріншотів.
8. Створення власного слайд-шоу з ефектами.
9. Блок пакетної обробки.
10. Підтримка декількох мов.
11. Блок функцій по редагуванню зображень:
 - авто-обрізка;
 - кадрування;
 - видалення нахилу;
 - фліп;
 - дзеркальне відображення;
 - зміна розміру зображення;
 - поворот зображення;
 - зрушення зображення;
 - додавання шуму;
 - вигин зображення;
 - видалення границь.
12. Блок ефектів:
 - тиснення;
 - розмиття;

- гравірування;
- корекція гами;
- розмиття по гауссу;
- світіння;
- градієнт;
- відтінки сірого;
- півтону;
- контрастність;
- зміна кольору;
- застосування ефекту "імпресіонізм";
- інвертування зображення й тексту;
- видалення ліній;
- видалення сегментів ліній;
- максимальна фільтрація;
- медіани;
- мінімальна фільтрація;
- мозаїка;
- розмиття в русі;
- множення;
- ефект старих фотографій;
- пікселізація;
- полярність;
- постеризовування;
- панч, радіальний хвильовий;
- хвильовість;
- насиченість;
- різкість;
- гладкість.

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.5. З нього видно, що процеси взаємодіють у системі наступним чином.

Спершу запускається процес виведення головного вікна програми. Він взаємодіє з наступними процесами:

- Процес відкриття файлу із зображенням.
- Процес збереження файлу із зображенням.

Процес відкриття файлу із зображенням взаємодіє з процесом виведення списку відкритих зображень.

Процес виведення списку відкритих зображень взаємодіє з процесом вибору зображення зі списку, який у свою чергу взаємодіє з наступними процесами:

- Процес виведення властивостей зображення.
- Процес виведення зображення на екран.

Останній процес взаємодіє з наступними процесами:

- Процес зміни масштабу зображення.
- Процес конвертації зображення.

Процес конвертації зображення, у свою чергу, взаємодіє з наступними процесами:

- Процес вибору формату.
- Процес перетворення даних.
- Процес стиску даних.
- Процес збереження файлу із зображенням.

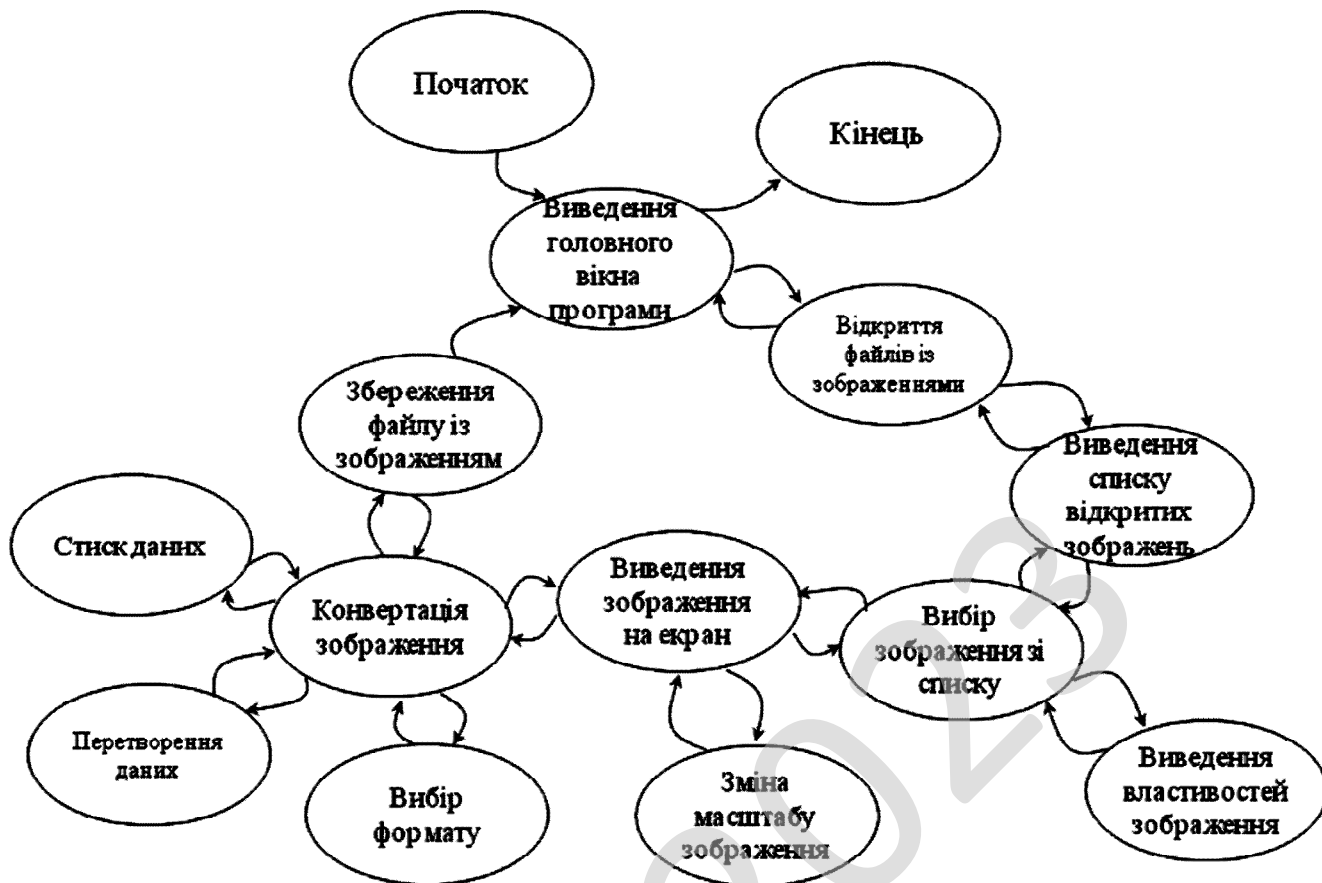


Рисунок 3.5 – Діаграма взаємодії процесів

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

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

4.1 Блок–схеми та опис алгоритмів функціонування системи

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

Спершу відбувається виведення основного вікна програми. Після цього користувач обирає одну з наступних дій:

- Відкриття файлу.
- Вибрати файл зі списку.
- Конвертувати зображення.
- Вибір розміру зображення.
- Зберегти конвертоване зображення.

Якщо користувач обирає відкриття файлу, тоді виконуються наступні дії:

- Відкриття вибраних файлів із зображенням.
- Виведення списку відкритих зображень.

Якщо користувач обирає вибір файл зі списку, тоді виконуються наступні дії:

- Вибір зображення зі списку.
- Виведення зображення та його властивостей на екран.

Якщо користувач обирає конвертувати зображення, тоді виконуються наступні дії:

- Вибір формату зображення.
- Вибір коефіцієнту стиснення.
- Запуск підпрограми стиснення та перетворення формату зображення.
- Виведення зображення після конвертації.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

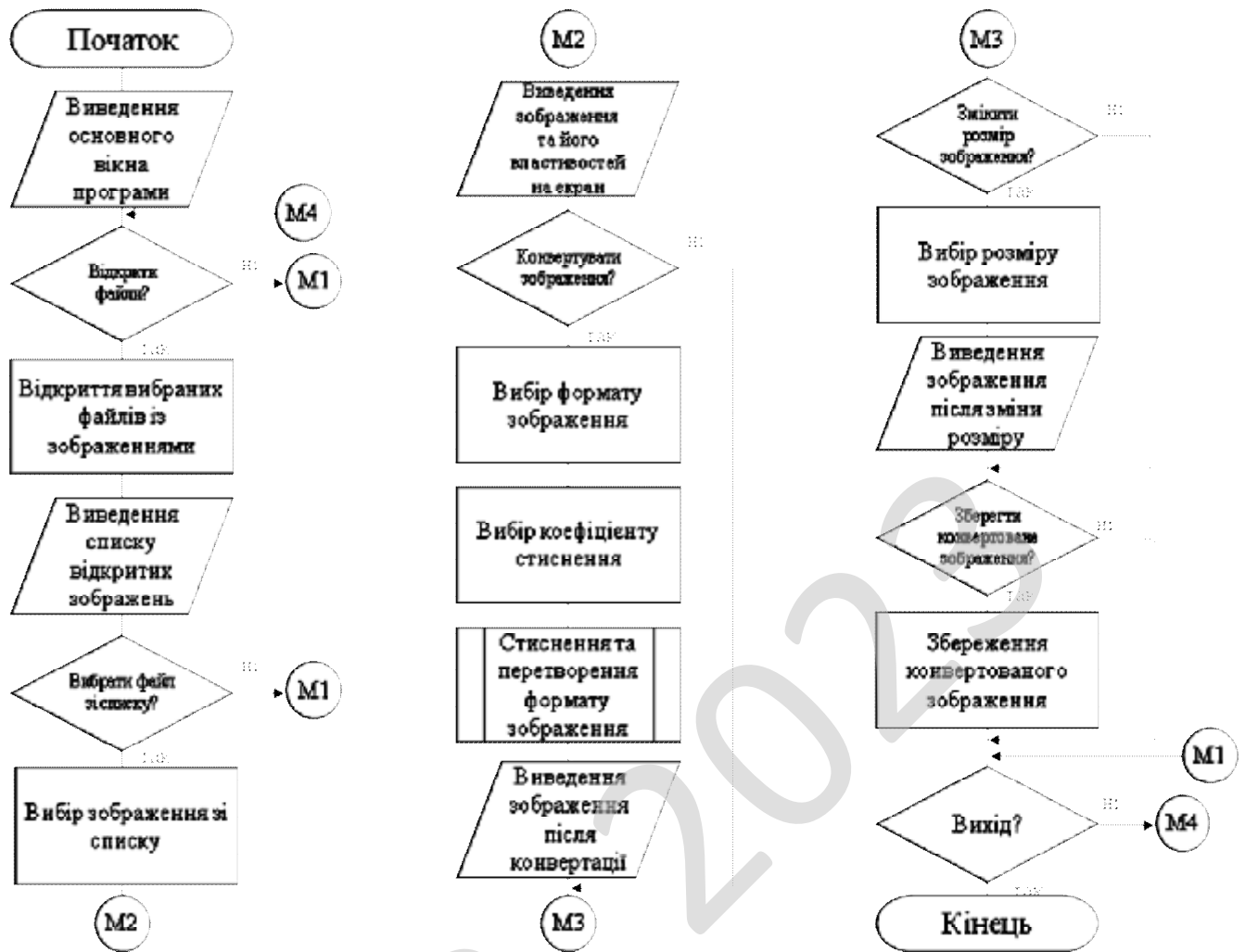


Рисунок 4.1 – Блок-схема роботи основної програми

Якщо користувач обирає зміну розміру зображення, тоді виконуються наступні дії:

- Вибір розміру зображення.
- Виведення зображення після зміни розміру.

Якщо користувач обирає збереження конвертованого зображення, тоді відбувається збереження конвертованого зображення у обраному форматі даних.

Після виконання усіх вищеперерахованих дій користувач обирає працювати йому далі з програмою, або ні.

Далі, доки не закінчуються блоки, виконуються наступні дії:

- Читання блоку на який вказує лічильник.
- Квантування блоку.
- Кодування по Хаффману.

Після цього відбувається повернення квантованого зображення й підпрограма закінчує свою роботу.

Приведемо елементи програмного коду, які реалізують деякі функції кодування файлу у заданий графічний формат.

Конвертування BMP в EMF

```
function bmp2emf(const SourceFileName: TFileName): Boolean;
var
  Metafile: TMetafile;
  MetaCanvas: TMetafileCanvas;
  Bitmap: TBitmap;
begin
  Metafile := TMetafile.Create;
  try
    Bitmap := TBitmap.Create;
    try
      Bitmap.LoadFromFile(SourceFileName);
      Metafile.Height := Bitmap.Height;
      Metafile.Width := Bitmap.Width;
      MetaCanvas := TMetafileCanvas.Create(Metafile, 0);
      try
        MetaCanvas.Draw(0, 0, Bitmap);
      finally
        MetaCanvas.Free;
      end;
    finally
      Bitmap.Free;
    end;
    Metafile.SaveToFile(ChangeFileExt(SourceFileName, '.emf'));
  finally
    Metafile.Free;
  end;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
```



```

XOrMask.Canvas.Ellipse(4, 4, IconSize - 4, IconSize - 4);
{Малюємо як тест}
Form1.Canvas.Draw(IconSize * 4, IconSize, XOrMask);
{Створюємо іконку}
Icon := TIcon.Create;
IconInfo.fIcon := true;
IconInfo.xHotspot := 0;
IconInfo.yHotspot := 0;
IconInfo.hbmMask := AndMask.Handle;
IconInfo.hbmColor := XOrMask.Handle;
Icon.Handle := CreateIconIndirect(IconInfo);
{Знищуємо тимчасові бітмапи}
AndMask.Free;
XOrMask.Free;
{Малюємо як тест}
Form1.Canvas.Draw(IconSize * 6, IconSize, Icon);
{Повідомляємо іконку як іконка додатка}
Application.Icon := Icon;
{генеруємо перемальовування}
InvalidateRect(Application.Handle, nil, true);
{Звільняємо іконку}
Icon.Free;
end;

```

Конвертування BMP в JPG

```

uses
  Jpeg, ClipBrd;
procedure TfrmMain.ConvertBMP2JPEG;
var
  jpgImg: TJPEGImage;
begin
  // копіюємо бітмап у буфер обміну
  chrtOutputSingle.CopyToClipboardBitmap;
  // беремо буфер обміну й завантажуюмо в Image1
  Image1.Picture.Bitmap.LoadFromClipboardFormat(cf_BitMap,
    Clipboard.GetAsHandle(cf_BitMap), 0);
  // створення jpeg-графіки
  jpgImg := TJPEGImage.Create;
  // призначте крапковий малюнок jpeg, це перетворює крапковий малюнок
  jpgImg.Assign(Image1.Picture.Bitmap);
  // і зберігаємо це у файл
  jpgImg.SaveToFile('TChartExample.jpg');

```

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

```

end;
Конвертування ICO в BMP
Var
Icon : TIcon;
Bitmap : TBitmap;
begin
Icon := TIcon.Create;
Bitmap := TBitmap.Create;
Icon.LoadFromFile('c:\picture.ico');
Bitmap.Width := Icon.Width;
Bitmap.Height := Icon.Height;
Bitmap.Canvas.Draw(0, 0, Icon);
Bitmap.SaveToFile('c:\picture.bmp');
Icon.Free;
Bitmap.Free;
end;

```

Конвертування JPG в BMP

```

uses JPEG;
procedure JPEGtoBMP(const FileName: TFileName);
var
jpeg: TJPEGImage;
bmp: TBitmap;
begin
jpeg := TJPEGImage.Create;
try
jpeg.CompressionQuality := 100; {Default Value}
jpeg.LoadFromFile(FileName);
bmp := TBitmap.Create;
try
bmp.Assign(jpeg);
bmp.SaveToFile(ChangeFileExt(FileName, '.bmp'));
finally
bmp.Free
end;
finally
jpeg.Free
end;
end;

```

Завантаження BMP файлу з DLL

```

procedure TForm1.Button1Click(Sender: TObject);
var

```

						ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			62

```

    AModule: THandle;
begin
    AModule := LoadLibrary('Images.dll');
    image1.Picture.Bitmap.LoadFromResourceName(AModule, 'StartMine');
    FreeLibrary(AModule);
end;

```

Визначення розміру картинки для JPG, GIF і PNG файлів

```

unit ImgSize;
interface
uses Classes;
procedure GetJPGSize(const sFile: string; var wWidth, wHeight: Word);
procedure GetPNGSize(const sFile: string; var wWidth, wHeight: Word);
procedure GetGIFSize(const sGIFFile: string; var wWidth, wHeight: Word);
implementation
uses SysUtils;
function ReadMWord(f: TFileStream): Word;
type
    TMotorolaWord = record
        case Byte of
            0: (Value: Word);
            1: (Byte1, Byte2: Byte);
        end;
var
    MW: TMotorolaWord;
begin
    f.read(MW.Byte2, SizeOf(Byte));
    f.read(MW.Byte1, SizeOf(Byte));
    Result := MW.Value;
end;
procedure GetJPGSize(const sFile: string; var wWidth, wHeight: Word);
const
    ValidSig: array[0..1] of Byte = ($FF, $D8);
    Parameterless = [$01, $D0, $D1, $D2, $D3, $D4, $D5, $D6, $D7];
var
    Sig: array[0..1] of byte;
    f: TFileStream;
    x: integer;
    Seg: byte;
    Dummy: array[0..15] of byte;
    Len: word;
    ReadLen: LongInt;

```

```

begin
  FillChar(Sig, SizeOf(Sig), #0);
  f := TFileStream.Create(sFile, fmOpenRead);
  try
    ReadLen := f.read(Sig[0], SizeOf(Sig));
    for x := Low(Sig) to High(Sig) do
      if Sig[x] <> ValidSig[x] then ReadLen := 0;
    if ReadLen > 0 then
      begin
        ReadLen := f.read(Seg, 1);
        while (Seg = $FF) and (ReadLen > 0) do
          begin
            ReadLen := f.read(Seg, 1);
            if Seg <> $FF then
              begin
                if (Seg = $C0) or (Seg = $C1) then
                  begin
                    ReadLen := f.read(Dummy[0], 3); { Ці байти не потрібні }
                    wHeight := ReadMWord(f);
                    wWidth := ReadMWord(f);
                  end
                else
                  begin
                    if not (Seg in Parameterless) then
                      begin
                        Len := ReadMWord(f);
                        f.Seek(Len - 2, 1);
                        f.read(Seg, 1);
                      end
                    else
                      Seg := $FF; { Це неправдиво, для підтримки циклу. }
                    end;
                  end;
                end;
              end;
            end;
          end;
        finally
          f.Free;
        end;
      end;
    end;
  procedure GetPNGSize(const sFile: string; var wWidth, wHeight: Word);
  type
    TPNGSig = array[0..7] of Byte;

```

```

const
  ValidSig: TPNGSig = (137,80,78,71,13,10,26,10);
var
  Sig: TPNGSig;
  f: tFileStream;
  x: integer;
begin
  FillChar(Sig, SizeOf(Sig), #0);
  f := TFileStream.Create(sFile, fmOpenRead);
  try
    f.read(Sig[0], SizeOf(Sig));
    for x := Low(Sig) to High(Sig) do
      if Sig[x] <> ValidSig[x] then Exit;
    f.Seek(18, 0);
    wWidth := ReadMWord(f);
    f.Seek(22, 0);
    wHeight := ReadMWord(f);
  finally
    f.Free;
  end;
end;
procedure GetGIFSize(const sGIFFile: string; var wWidth, wHeight: Word);
type
  TGIFHeader = record
    Sig: array[0..5] of char;
    ScreenWidth, ScreenHeight: Word;
    Flags, Background, Aspect: Byte;
  end;
  TGIFImageBlock = record
    Left, Top, Width, Height: Word;
    Flags: Byte;
  end;
var
  f: file;
  Header: TGifHeader;
  ImageBlock: TGifImageBlock;
  nResult: integer;
  x: integer;
  c: char;
  DimensionsFound: boolean;
begin
  wWidth := 0;

```

						БКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			65

```

wHeight := 0;
if sGifFile = '' then
  Exit;
{$ I-I-}
FileMode := 0; { тільки читання }
AssignFile(f, sGifFile);
reset(f, 1);
if IOResult <> 0 then
  { Не вдалося відкрити файл }
  Exit;
{ читає заголовок для забезпечення правильності файлу. }
BlockRead(f, Header, SizeOf(TGifHeader), nResult);
if (nResult <> SizeOf(TGifHeader)) or (IOResult <> 0) or
  (StrLComp('GIF', Header.Sig, 3) <> 0) then
begin
  { файл зображення недійсний }
  Close(f);
  Exit;
end;
{ пропускаємо карту кольорів якщо вона присутня }
if (Header.Flags and $80) > 0 then
begin
  x := 3 * (1 shl ((Header.Flags and 7) + 1));
  Seek(f, x);
  if IOResult <> 0 then
  begin
    { карта кольорів зруйнована }
    Close(f);
    Exit;
  end;
end;
DimensionsFound := False;
FillChar(ImageBlock, SizeOf(TGIFImageBlock), #0);
{ беремо блоками. }
BlockRead(f, c, 1, nResult);
while (not EOF(f)) and (not DimensionsFound) do
begin
  case c of
    ',': { пошук зображення }
    begin
      BlockRead(f, ImageBlock, SizeOf(TGIFImageBlock), nResult);
      if nResult <> SizeOf(TGIFImageBlock) then

```

```

begin
    { блокуємо помилкові зображення }
    Close(f);
    Exit;
end;
wWidth := ImageBlock.Width;
wHeight := ImageBlock.Height;
DimensionsFound := True;
end;
'y': { пропускаємо }
begin
    { NOP }
end;
end;
BlockRead(f, c, 1, nResult);
end;
Close(f);
{$I+}
end;
end.

```

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму SHA-3 (Кессак) – алгоритм гешування змінної розрядності, розроблений групою авторів на чолі з Йоаном Дайменом, співавтором Rijndael, автором шифрів MMB, SHARK, Noekeon, SQUARE і BaseKing. 2 жовтня 2012 року Кессак став переможцем конкурсу криптографічних алгоритмів, проведеним Національним інститутом стандартів і технологій США. 5 серпня 2015 року алгоритм затверджено та опубліковано в якості стандарту FIPS 202¹. У програмній реалізації автори заявляють про 12,5 циклах на байт при виконанні на ПК з процесором Intel Core 2. Проте в апаратних реалізаціях Кессак виявився набагато швидшим, ніж всі інші фіналісти.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

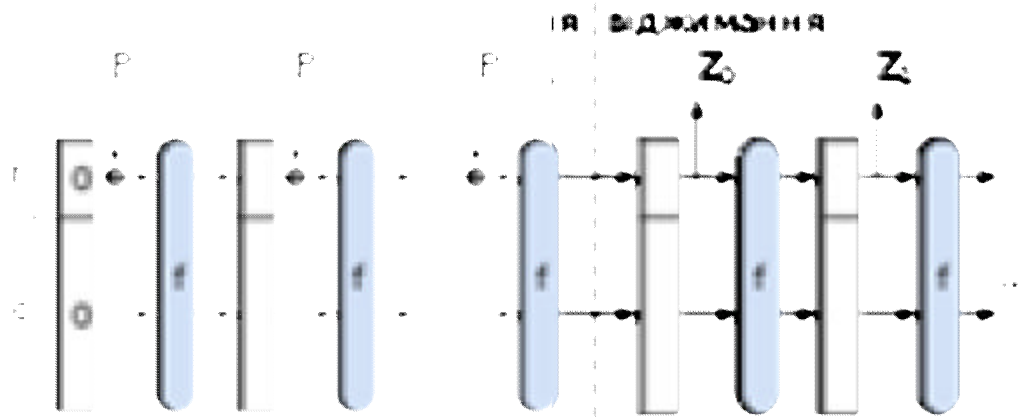


Рисунок 4.3 – Конструкція функції губки, використана в хеш-функції

Конструкція функції губки, використана в хеш-функції. P_i – вхідні блоки, Z_j – вихід алгоритму. Невикористаний для виведення набір бітів c («capacity») повинен мати значний розмір для досягнення стійкості до атак.

Алгоритм SHA-3 побудований за принципом криптографічної губки (дана структура криптографічних алгоритмів була запропонована авторами алгоритму Кесак раніше).

Хеш-функції сімейства SHA-3 побудовані на основі конструкції криптографічної губки, в якій дані спочатку «вбираються» в губку, при якому початкове повідомлення M піддається багатораундовим перестановкам f , потім результат Z «віджимається» з губки. На етапі «вбирання» блоки повідомлення додаються за модулем 2 з підмножиною стану, який потім перетворюється з допомогою функції перестановки f . На етапі «віджимання» вихідні блоки зчитуються з одного і того ж підмножинного стану, зміненого функцією перестановки f . Розмір частини стану, який записується і зчитується, називається «швидкістю» (англ. rate) і позначається r , а розмір частки, яка незаймана введенням / виведенням, називається «ємністю» (англ. capacity) і позначається c .

Алгоритм отримання значення хеш-функції можна розділити на кілька етапів:

– Вихідне повідомлення M додається до рядка P довжини, кратній r , за допомогою функції доповнення (pad-функції).

– Рядок P ділиться на n блоків довжини r : P_0, P_1, \dots, P_{n-1}

– «Всмоктування»: кожен блок P_i доповнюється нулями до рядка довжини b біт і підсумовується по модулю 2 з рядком стану S , де S – рядок довжини b біт ($b = r + c$). Перед використанням цієї функції всі елементи S дорівнюють нулю. Для кожного наступного блоку стан – рядок, отриманий застосуванням функції перестановок f до результату попереднього кроку.

– «Віджимання»: поки довжина Z менша d (d – кількість біт в результаті геш-функції), до Z додається r перших біт стану S , після кожного додавання до S , застосовується функція перестановок f . Потім S обрізається до довжини d біт

– Рядок Z довжини d біт повертається в якості результату

Завдяки тому, що стан містить c додаткових біт, алгоритм стійкий до атаки подовженням повідомлення, до якої прийняті алгоритми SHA-1 і SHA-2.

У SHA-3 стан S – це масив 5×5 слів довжиною $w = 64$ біта, всього $5 \times 5 \times 64 = 1600$ біт. Також в Кессак можуть використовуватися довжини w , рівні меншим ступеням 2 (від $w = 1$ до $w = 32$).

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що головне вікно складається з наступних блоків:

- Блок меню.
- Вікно відкритих зображень.
- Кнопки швидкого доступу до елементів меню.
- Параметри збереження.
- Якість зображення після стиснення.
- Розміри зображення.
- Вікно стану процесу при конвертуванні.

Блок меню складається з наступних елементів:

- Файл.
- Конвертація.
- Параметри.
- Довідка.

Параметри збереження складаються з наступних елементів:

- Замінити файл.
- Копіювати в папку.
- Упакувати в архівний файл.

На рисунку 5.2 зображено процес конвертації файлу.

На рисунку 5.3 зображено процес перегляду зображень. З рисунку видно, що виводиться вікно, у якому відображене зображення й визначені параметри зображення перед та після конвертації:

- Розмір файлу.
- Формат файлу.
- Масштаб.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

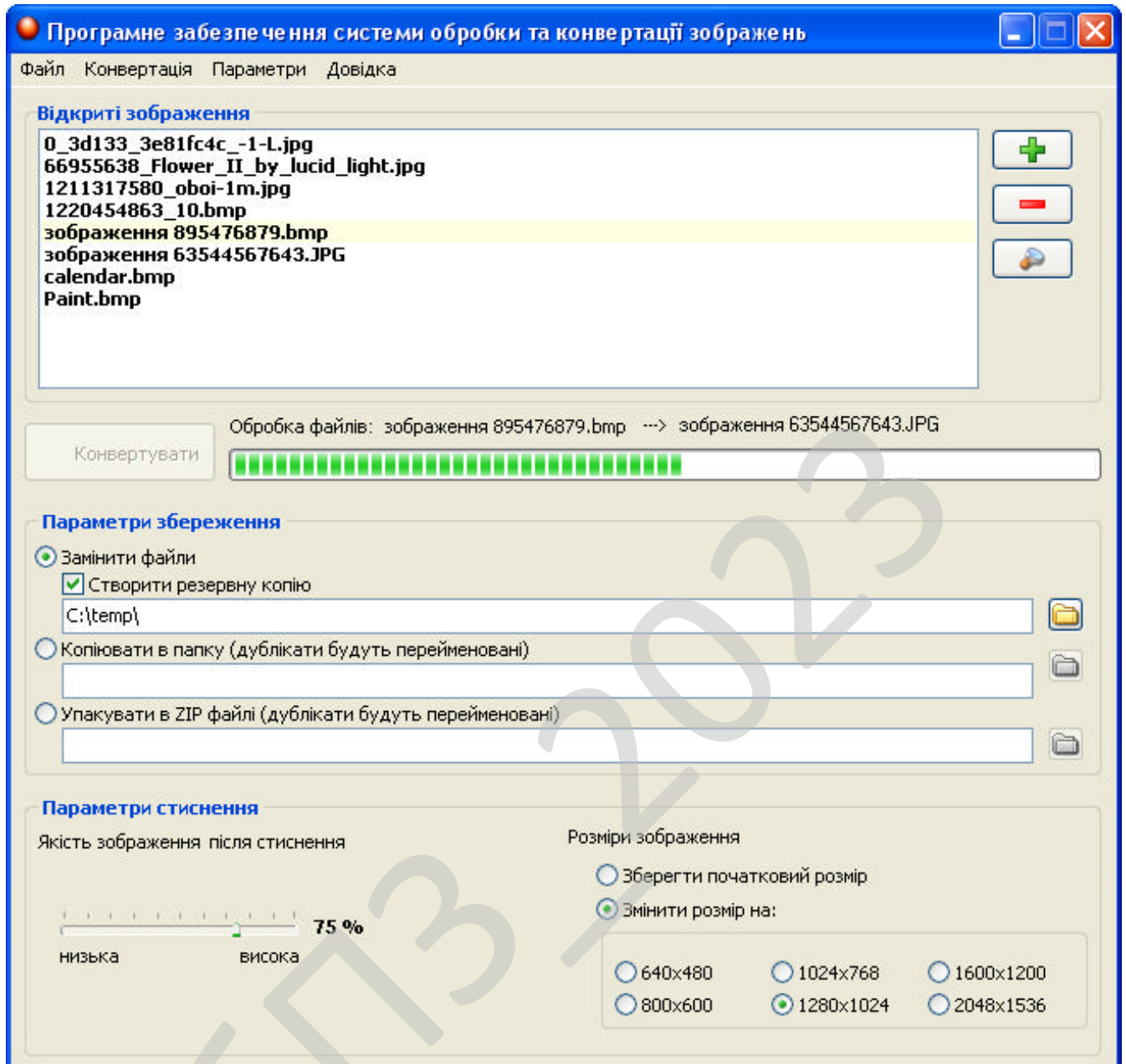


Рисунок 5.2 – Конвертація файлів

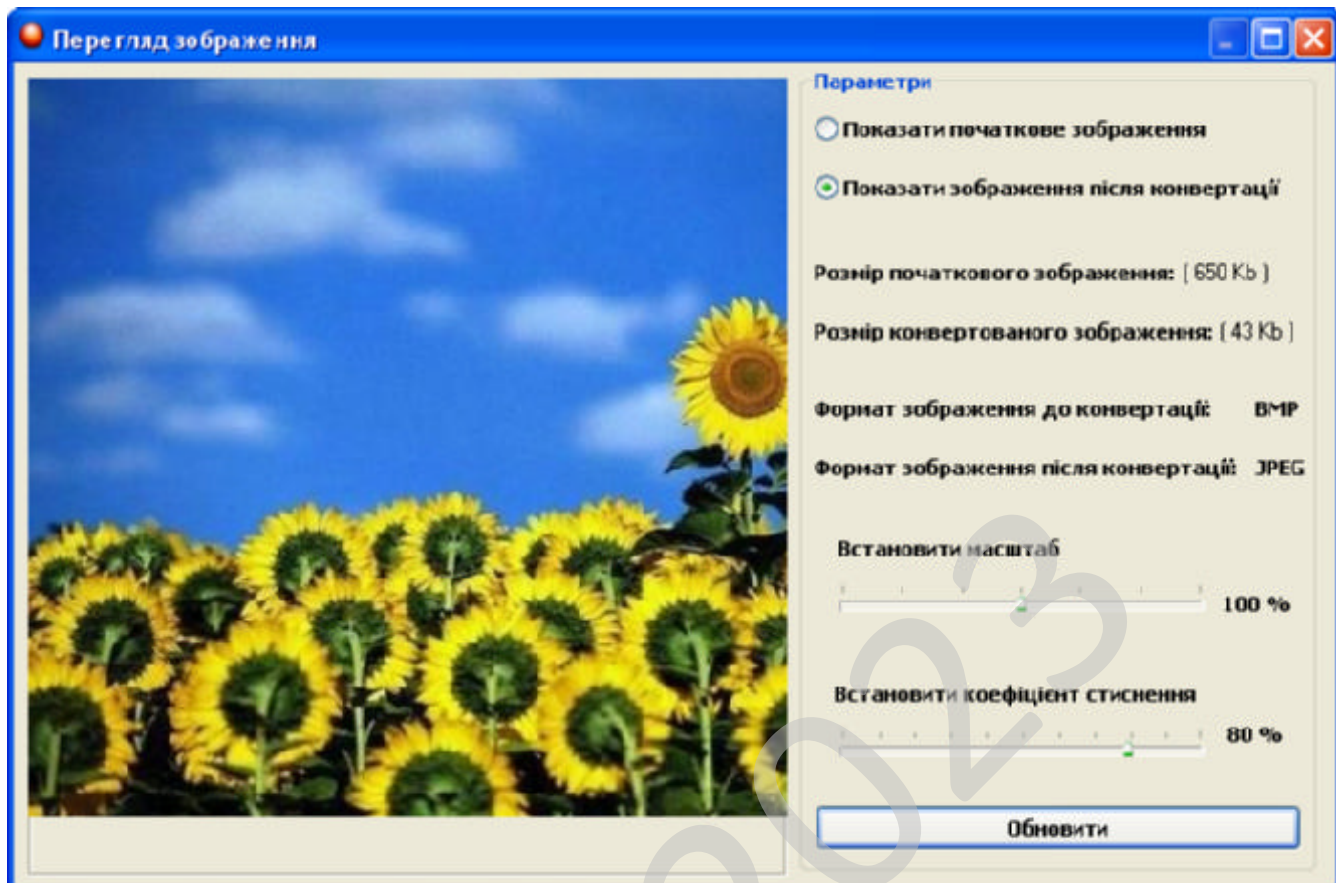


Рисунок 5.3 – Перегляд зображень

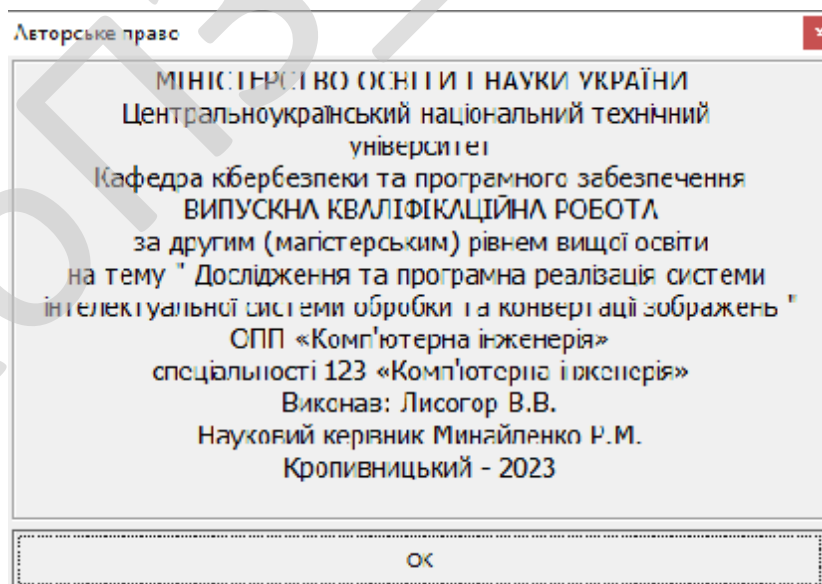


Рисунок 5.4 – Довідка

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної системи обробки та конвертації зображень.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

Об'єктом дослідження є процес інтелектуальної системи обробки та конвертації зображень.

Предметом дослідження є методи інтелектуальної системи обробки та конвертації зображень.

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

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

– Удосконалено метод інтелектуальної системи обробки та конвертації зображень.

– Розроблено вітчизняний продукт інтелектуальної системи обробки та конвертації зображень, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи обробки та конвертації зображень.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	120
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	120000
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-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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.:

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

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 50 = 84 \text{ люд/день.}$$

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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	84	Ф 7.1-7.4
Впровадження	13	Д13
Всього	125	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;
 T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{125 \cdot 1}{60 - 5} = 2,3 \text{ ставки.}$$

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	6	540	9
Монітор	60	6	360	6
Клавіатура	30	6	180	3
Маніпулятор «мишка»	30	6	180	3
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	1	30	0,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	260	650	10,83
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	37,99

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{\text{ор}}^c = \frac{38 \cdot 3}{1,2} = 95 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	0,5	18100	27150
Продакт-менеджер	0,25	10000	7500
Інженер-програміст	2,3	16000	110400
Інженер-електронщик	0,2	10000	6000
Інженер-системотехнік	0,25	10000	7500
Адміністратор мережі	0,5	10000	15000
Системний програміст	0,25	10000	7500
Дизайнер WEB	0,25	10000	7500
Інженер-верстальник	0,25	10000	7500
Бухгалтер-економіст	0,5	10000	15000
Всього за період розробки	$R_{cn} = 5,25$	-	$\Phi_{роб} = 211050$

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.

$$Z_{cd} = \frac{211050}{5,25 \cdot 60} = 670 \text{ грн.}$$

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-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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

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

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

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

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу Інтернет магазину Компбест за 20.10.23 – джерело <https://compbest.com.ua>.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Core i5-8400 (6 ядер по 2.8 - 4.0 GHz), MB Smart Cache	-
Системна плата	MB Intel s1151 mATX, 4x USB 2.0, 2x USB 3.0, 2x PS/2, 2x DVI, 1x HDMI, 5x Audio, 1 LAN (RJ-45)	-
Відеокарта	nVidia GeForce GTX 750 Ti, 2 GB DDR5 128-bit	-
Жорсткий диск	SSD: 240 Gb	-
Оперативна пам'ять	16 GB DDR4	-
DVD-привод	DVDRW Pioneer DVR-TD10RS SATA Slim Black Bulk (DVR-TD10RS)	-
Корпус	ATX Terra Tower, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm) black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear) 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 3.0 Card reader. 3.5", 2*USB3. +AUDIO+1394, multi: All Type Cards, black	-
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 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	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	199177

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

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	199177	-	-
Всього по групі	199177	50	99588,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$K_p = 1760367$		$A_p = 190286$

Згідно прийнятих норм на підприємстві $n_{\text{вип}}$ приймаємо 0,5 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=210$ грн., визначаємо вартість паперу за період розробки:

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

$$З_{M1} = 210 \cdot 0,5 = 105 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 60):

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

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 24 грн./шт., DVD-R box – 35 грн./шт.

$$З_{M2} = 60 \cdot 24 = 1440 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

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

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

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

$$З_M = (105 + 1440 + 1702) / 120 = 27 \text{ грн.}$$

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

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

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

$$O_n = 698 \cdot 15 \cdot 0,01 = 105 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 120$ прим.):

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

$$A_m = 190286 \cdot 3 / (120 \cdot 12) = 396 \text{ грн.}$$

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

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

$$C_n = 698 + 70 + 288 + 105 + 27 + 105 + 396 = 1689 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	Z_o	698
2. Додаткова зарплата виконавців	Z_d	70
3. Відрахування на соціальні потреби	C_{oc}	288
4. Загальногосподарські витрати	Γ_{ocn}	105
5. Витрати на матеріали	Z_m	27
6. Освоєння нових операційних систем, мов програмування	O_n	105
7. Амортизація основних фондів	A_m	396
8. Повна собівартість програмного забезпечення	C_n	1689
9. Плановий прибуток	P_p	845
10. Ціна підприємства $C_n = C_n + P_p$	C_n	2534
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	506,8
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3040,8

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 1689 = 845 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

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

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

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3041
Всього капітальних витрат	–	3041

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	56364	32208
2. Витрати на електроенергію	$Z_{ел}$	439	251
3. Витрати на амортизацію	$Z_{ам}$	0	760
Всього витрат за рік	I	56803	33219

Витрати на обслуговування роботи системи:

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

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

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин на обслуговування системи зменшилось з 350 год до 200 год на рік.

$$Z_{p\text{ баз}} = 350 \cdot 120 \cdot 1,1 \cdot 1,22 = 56364 \text{ грн},$$

$$Z_{p\text{ нов}} = 200 \cdot 120 \cdot 1,1 \cdot 1,22 = 32208 \text{ грн}.$$

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	–	3041	–	760,25
Всього відрахувань	-	–	3041	–	760,25

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

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

$$Z_{ел \text{ баз}} = 0,545 \cdot 350 \cdot 2,3 = 439 \text{ грн.}$$

$$Z_{ел \text{ нов}} = 0,545 \cdot 200 \cdot 2,3 = 251 \text{ грн.}$$

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:

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

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

$$E_e = (2534 - 1689) \cdot 120 - (0,05 \cdot 1408000 + 0,4 \cdot 199177 + 0,25 \cdot 33190 + 0,1 \cdot 120000) \cdot 3/12 = 53828,5 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p – балансова вартість основних фондів розробника.

$$T_e = \frac{1760367}{(2534 - 1689) \cdot 120 \cdot 12 / 3} = 4 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

де: I_b, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_b, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (56803 - 33219) - 0,25 \cdot 3041 = 22824 \text{ грн.}$$

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.	120
2. Повна собівартість розробленої програми	Грн.	1689
3. Ціна розробленої програми	Грн.	2534
4. Плановий прибуток від реалізації розробленої програми	Грн.	845
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	101400
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	53828,5
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Рік	4
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3041
11. Величина економічного ефекту у користувача програмної продукції	Грн.	22824
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	0,13

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

$$T_{cn} = \frac{K_n - K_b}{I_b - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{3041}{56803 - 33219} = 0,13 \text{ року.}$$

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.

					VKPM-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Персональні електронно-обчислювальні машини (ПЕОМ) відіграють важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ПЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ПЕОМ у професійних цілях, оскільки завдяки ПЕОМ зявилося багато нових професій.

В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв'язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов'язаних з постійною роботою за комп'ютером.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат;

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

8.3 Аналіз умов праці програміста

Умови праці в приміщенні, в якому знаходиться робоче місце програміста є сприятливими. Приміщення обладнане автономною системою газового опалення, основною перевагою якого є програмування режиму роботи в залежності від погодних умов, оскільки клімат є нестійким. Використовується система природної та штучної вентиляції, що забезпечує ефективну циркуляцію повітря. В кабінеті знаходиться кондиціонер АКАІ АК-АС7010-OF.

Засоби копіювальної техніки знаходяться на достатньо далекій відстані від робочих місць, оскільки приміщення складає 20 м², а у відділі налічується два працівники, тобто концентрація озону та оксиду азоту в повітрі є невисокою. Таким чином на кожного програміста приходиться 10,0 м² що відповідає нормам Державним санітарним правилам і нормам ДСанПіН 3.3.2.007-98 [1]. Висота стелі приміщення складає 2,9 метри, що також не порушує нормативні вимоги. Прибиральники підтримують порядок в службових приміщеннях, дотримуються санітарно-гігієнічних норм по прибиранню приміщень, витирають пил, підмітають підлогу наприкінці кожного робочого дня.

В цілому потрібно відмітити застарілість офісної техніки та відсутність клавіатур з ергономічною розкладкою та рідкокристалічних моніторів, які здійснюють менш негативний вплив на стан здоров'я працівників відділу.

Оформлення інтер'єру приміщення є відповідне вимогам з ергономіки та стимулює працівників до підвищення працездатності та зниження втоми. Стеля білого кольору створює оптичний ефект збільшення висоти приміщення, підлога пофарбована коричневим кольором, а стіни – у жовтий. Перевагами даного кольору є створення відчуття теплоти, здатність привертати увагу без додаткової втоми.

Висота столу складає 71,5 см, до того ж його можна регулювати відповідно до власних потреб. Стіл має достатній внутрішній об'єм, завдяки ширині у 73 см та висоті простору під столом – у 63 см, є достатньо важким для забезпечення

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

Для запобігання поразки електричним струмом в приміщенні відділу використовується ряд організаційно-технічних заходів: розташування проводів живлення поза зоною пересування людей; допуск до роботи електроприладів тільки тих робітників, що знайомі із технікою безпеки; використання мережних продовжувачів з вбудованими запобіжниками на 0,1 А; при ремонті обладнання персонал попереджується.

Устаткування, що працює в приміщенні живиться від мережі 220В та частотою 50Гц. Споживачами цієї напруги є також джерела штучного освітлення. Вони розташовуються на висоті 3 м, що задовольняє нормі, відповідно до якого джерела освітлення повинні розташовуватися на висоті 2,5 м від підлоги.

Проводка схована. У якості розеток для підключення устаткування застосовуються розетки з заземленим кожухом, захищеного від випадкового доторку до струмоведучих частин. Електроустаткування, що знаходиться в приміщенні відділу відноситься до установок напругою до 1000В.

На робочому місці програміста з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартам фірми ІВМ, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

Основні причини ураження людини електричним струмом на робочому місці:

– дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції:

- нерегламентоване використання електричних приладів:
- відсутність інструктажу співробітників з правил електробезпеки.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

Отже за результатами проведеного аналізу можна зробити висновки, що всі показники знаходяться у межах запропонованих значень

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

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

8.5 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток $45 \cdot 45 \cdot 5$ мм., довжиною $L=2,5$ м., та горизонтальний електрод – металева полоса з перетином $40 \cdot 4$ мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=3$ м. Глибина закладення горизонтального контура заземлення $t=0,8$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр. 13-16 [6], або аналогічної.

Зелёное выкинуть,

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,8+2,5/2=2,05 \text{ м.}$$

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуальної системи обробки та конвертації зображень.

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 інтелектуальної системи обробки та конвертації зображень.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуальної системи обробки та конвертації зображень.

– Досліджена система інтелектуальної системи обробки та конвертації зображень.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної системи обробки та конвертації зображень.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної системи обробки та конвертації зображень.

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

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

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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

Програма реалізована на мові високого рівня RAD Studio Delphi. 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 SHA-3.

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 22824 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,13 роки.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лисогор В.В. Дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. G.K.Wallace. “The JPEG still picture compression standard.” Communication of ACM. Volume 34. Number 4 April 1991.
3. B.Smith, L.Rowe. “Algorithm for manipulating compressed images.” Computer Graphics and applications. September 1993.
4. A. Jacquin, “Fractal image coding based on a theory of iterated contractive image transformations”, Visual Comm. and Image Processing, vol. SPIE-1360, 1990.
5. “Progressive Bi-level Image Compression, Revision 4.1”, ISO/IEC JTC1/SC2/WG9, CD 11544, September 16, 1991.
6. W.B. Pennebaker, J.L. Mitchell, G.G. Langdon, R.B. Arps, “An overview of the basic principles of the Q-coder adaptive binary arithmetic coder”, IBM Journal of research and development, Vol.32, No.6, November 1988, pp. 771-726.
7. D.A. Huffman, “A method for the construction of minimum redundancy codes.” In processing. IRE val.40 1962 pp. 1098-1101.
8. Standardisation of Group 3 Facsimile apparatus for document transmission. CCITT Recommendations. Fascicle VII.2. T.4. 1980.
9. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press. 2022 1677 p.
10. Will Grant. 101 UX Principles. Packt Publishing. 2022. 432 p.
11. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
12. Henry Lloyd. Interactive Computer Graphics. States Academic Press. 2022. 247 p.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

13. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
14. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
15. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
16. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
17. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
18. 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.
19. 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.
20. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph.. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
21. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
22. 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.
23. 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,
24. 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-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

25. 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>

26. 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.

27. 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.

28. 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.

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

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

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

49. 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.

50. 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.

51. 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.

52. 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.

					ВКРМ-123.23.0014.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.23.0014.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Лисогор В.В.				Дослідження та програмна реалізація системи інтелектуальної системи обробки та конвертації зображень	Літ.	Аркуш	Аркушів
Перевірів	Минайленко Р.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інтелектуальної системи обробки та конвертації зображень.

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. № 34-13 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 інтелектуальної системи обробки та конвертації зображень.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-123.23.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи інтелектуальної системи обробки та конвертації зображень;
- 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-123.23.0014.00.00.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 RAD Studio Delphi.

					ВКРМ-123.23.0014.00.00.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 повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.23.0014.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 112 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 12.12.2023 р.

					ВКРМ-123.23.0014.00.00.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
_____ Минайленко Р.М.

***Дослідження та програмна реалізація
системи інтелектуальної системи обробки та конвертації зображень***

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 28

Літера: РП

Кропивницький – 2023 року

EasyCompress_Obrobka_ta_konvertacia_zobragen.dpr - файл проекту

```

program EasyCompress_Obrobka_ta_konvertacia_zobragen;

uses
  Windows, SysUtils, Classes, Forms, Messages,
  Main in 'Main.pas' {MainForm}, // головна програма
  SfxUtils in 'TZIP\SfxUtils.pas', // підпрограма центру роботи з зображеннями
  Zip in 'TZIP\Zip.pas', // підпрограма стиснення зображення
  ZipDlls in 'TZIP\ZipDlls.pas'; // бібліотека стиснення зображення
  Preview in 'Preview.pas' {DisplayForm}, // підпрограма перегляду зображень
  Error in 'Error.pas' {ErrorForm}, // підпрограма виявлення помилок
  About in 'About.pas' {AboutForm}, // підпрограма виведення даних про розробника

var
  ClassName : Array[0..255] of char;
  WM_PARAM_ATOM : integer;
  result : integer;
  h: THandle;
  Param_Atom: Atom;
  nb : integer;

{$R EasyCompress_Obrobka_ta_konvertacia_zobragen.res}

begin
  {
    Application.Initialize;
    // Application.Title := 'Starting
    EasyCompress_Obrobka_ta_konvertacia_zobragen...';
    GetClassName(Application.Handle, ClassName, 254);
    // result := FindWindow(ClassName,
    'EasyCompress_Obrobka_ta_konvertacia_zobragen ');
    h := FindWindow(nil, 'EasyCompress_Obrobka_ta_konvertacia_zobragen ');

    if result <> 0 then begin
      WM_PARAM_ATOM := RegisterWindowMessage('WM_PARAM_ATOM');
      if ParamCount > 0 then begin
        for nb := 1 to ParamCount do begin
          Param_Atom := GlobalAddAtom(PChar(ParamStr(nb)));
          SendMessage(h, WM_PARAM_ATOM, Param_Atom, 0);
        end;
      end;
      ShowWindow(result, SW_RESTORE);
      SetForegroundWindow(result);
      Application.Terminate;
    end else begin
      Application.Title := '';
      // Application.Icon.Assign("chinaz23.ico");
      Application.CreateForm(TMainForm, MainForm);
      Application.CreateForm(TErrorForm, ErrorForm);
      Application.CreateForm(TAboutForm, AboutForm);
      Application.CreateForm(TDisplayForm, DisplayForm);
      Application.Run;
    // end;
  end.

```

Preview.pas - файл передперегляду зображень

```

unit Preview;

interface

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

type
  TDisplayForm = class(TForm)
    Panel_DisplayImage: TPanel;
    DisplayImage: TImage;
    Panel_DisplaySettings: TGroupBox;
    RBtn_ShowOriginalImage: TRadioButton;
    RBtn_ShowPreviewImage: TRadioButton;
    CompressionBar: TTrackBar;
    Label_CompressionRate: TLabel;
    Label_Title1: TLabel;
    ZoomFactorIndex: TTrackBar;
    Label_Title2: TLabel;
    Label_ZoomFactor: TLabel;
    RefreshAdvice_Label: TLabel;
    RefreshAdvice_Fond: TShape;
    RefreshAdvice_Image: TImage;
    ProcessingFile_Image: TImage;
    Label_OriginalFileSize: TLabel;
    Label_CompressedFileSize: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ImageMouseDown(Sender: TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
    procedure ImageMouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
    procedure CompressionRateChange(Sender: TObject);
    procedure RefreshDisplay(Sender: TObject);
    procedure ZoomFactorIndexChange(Sender: TObject);
    procedure LoadPreview(FileName: string);
    procedure Button1Click(Sender: TObject);
  private
    { Private Declarations }
    function GetZoomFactorFromIndex(Index: integer): integer;
  public
    { Public Declarations }
  end;

var
  DisplayForm: TDisplayForm;
  CursorStartPosX, CursorStartPosY : integer;

  CURRENTLY_DISPLAYED_FILE_NAME : string; // \
  CURRENTLY_USED_COMPRESSION_RATE : integer; // >- Збережені
частини інформації
  CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE : string; // /

implementation

uses Main;

{$R *.dfm}
procedure TDisplayForm.FormCreate(Sender: TObject);
begin
  // DoubleBuffered := true;
  // Panel_DisplayImage.DoubleBuffered := true;
end;
function TDisplayForm.GetZoomFactorFromIndex(Index: integer): integer;
begin

```

```

case Index of
  1: result:=12;
  2: result:=25;
  3: result:=50;
  5: result:=200;
  6: result:=400;
  7: result:=800;
else result:=100;
end;
end;
// ----- //
procedure TDisplayForm.ImageMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
  CursorStartPosY := Y;
  CursorStartPosX := X;
end;
// ----- //
procedure TDisplayForm.ImageMouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
  if (ssLeft in Shift) then begin
    DisplayImage.Top := DisplayImage.Top - (CursorStartPosY-Y);
    DisplayImage.Left := DisplayImage.Left - (CursorStartPosX-X);
    if DisplayImage.Top < Panel_DisplayImage.Height - DisplayImage.Height then
DisplayImage.Top := Panel_DisplayImage.Height - DisplayImage.Height;
    if DisplayImage.Left < Panel_DisplayImage.Width - DisplayImage.Width then
DisplayImage.Left := Panel_DisplayImage.Width - DisplayImage.Width;
    if DisplayImage.Top > 0 then DisplayImage.Top := 0;
    if DisplayImage.Left > 0 then DisplayImage.Left := 0;
  end;
end;
// ----- //
procedure TDisplayForm.ZoomFactorIndexChange(Sender: TObject);
var
  ZoomFactor: integer;
begin
  ZoomFactor := GetZoomFactorFromIndex(ZoomFactorIndex.Position);
  Label_ZoomFactor.Caption := IntToStr(ZoomFactor)+' %';
  DisplayImage.Height := round((ZoomFactor*DisplayImage.Picture.Height)/100);
  DisplayImage.Width := round((ZoomFactor*DisplayImage.Picture.Width)/100);
  if (DisplayImage.Height + DisplayImage.Top) < Panel_DisplayImage.Height then
DisplayImage.Top := (Panel_DisplayImage.Height - DisplayImage.Height);
  if (DisplayImage.Width + DisplayImage.Left) < Panel_DisplayImage.Width then
DisplayImage.Left := (Panel_DisplayImage.Width - DisplayImage.Width);
  if DisplayImage.Top > 0 then DisplayImage.Top := 0;
  if DisplayImage.Left > 0 then DisplayImage.Left := 0;
end;
// ----- //
procedure TDisplayForm.CompressionRateChange(Sender: TObject);
var
  AdviceControlsAvailability: boolean;

begin
  Label_CompressionRate.Caption := IntToStr(CompressionRate.Position) + ' %';

  AdviceControlsAvailability :=
(CURRENTLY_USED_COMPRESSION_RATE<>CompressionRate.Position) and
RBtn_ShowPreviewImage.Checked;
  RefreshAdvice_Image.Visible := AdviceControlsAvailability;
  RefreshAdvice_Label.Visible := AdviceControlsAvailability;
  RefreshAdvice_Fond.Visible := AdviceControlsAvailability;

  if not AdviceControlsAvailability then
    Label_CompressedFileSize.Caption :=
CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE
  else Label_CompressedFileSize.Caption := '( ??? Kb )';

```

```

MainForm.CompressionRate.Position:=DisplayForm.CompressionRate.Position;
end;
procedure TDisplayForm.LoadPreview(FileName: string);
begin
    CURRENTLY_DISPLAYED_FILE_NAME := FileName;
    CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE := '( ??? Kb )';
    Label_CompressedFileSize.Caption := CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE;
    RefreshDisplay(MainForm);
end;
procedure TDisplayForm.RefreshDisplay(Sender: TObject);
var
    LoadingFailure : boolean;
    JPEGImg1 : TJpegImage;
    JPEGImg2 : TJpegImage;
    BMPTempImg : TBitmap;
    TempStream : TMemoryStream;
    ImgFile : file of Byte;
begin
    LoadingFailure := false;
    DisplayForm.Cursor := crHourGlass;

    RefreshAdvice_Image.Visible := false;
    RefreshAdvice_Label.Visible := false;
    RefreshAdvice_Fond.Visible := false;
    ProcessingFile_Image.Visible := true;
    Application.ProcessMessages;

    CURRENTLY_USED_COMPRESSION_RATE := CompressionRate.Position;

    if RBtn_ShowOriginalImage.Checked then begin
CompressionRate.Enabled:=False;
        try
            DisplayImage.Picture.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME);
        except
            MessageBoxA(MainForm.Handle, Pchar('Не вдається прочитати зображення.
Помилка формату. '), Pchar('Помилка'), MB_ICONSTOP + MB_SYSTEMMODAL +
MB_SETFOREGROUND + MB_TOPMOST);
            LoadingFailure := true;
        end;
        if LoadingFailure then begin
            if not (Sender = MainForm) then Close;
            DisplayForm.Cursor := crDefault;
            exit;
        end;
    end else begin
CompressionRate.Enabled:=True;
        JPEGImg1 := TJpegImage.Create;
        BMPTempImg := TBitmap.Create;
        JPEGImg2 := TJpegImage.Create;
        TempStream := TMemoryStream.Create;
        try
            if (UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.JPEG') or
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.OBROBKA_TA_KONVERTA
CIA_ZOBRAGEN') or
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.SHM') then begin
//-----
                JPEGImg1.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME); // Завантаження
файлу
                Application.ProcessMessages;
                BMPTempImg.Assign(JPEGImg1);
            end else if
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.BMP') then begin
                BMPTempImg.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME); //
Завантаження файлу
            end;
            JPEGImg2.Assign(BMPTempImg);

```

```

Application.ProcessMessages;
JPEGImg2.CompressionQuality:= CompressionRate.Position;
JPEGImg2.Compress;
JPEGImg2.SaveToStream(TempStream);
TempStream.Position:=0;
JPEGImg2.LoadFromStream(TempStream);
CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE := '(
'+Inttostr(TempStream.Size div 1024)+' Kb )';
Label_CompressedFileSize.Caption :=
CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE;
DisplayImage.Picture.Assign(JPEGImg2);
finally
TempStream.Free;
JPEGImg2.Free;
BMPTempImg.Free;
JPEGImg1.Free;
end;
except
MessageBoxA(MainForm.Handle,Pchar('Не вдається прочитати зображення.
Помилка формату. '),Pchar('Помилка'),MB_ICONSTOP + MB_SYSTEMMODAL +
MB_SETFOREGROUND + MB_TOPMOST);
LoadingFailure := true;
end;
if LoadingFailure then begin
if not (Sender = MainForm) then Close;
DisplayForm.Cursor := crDefault;
exit;
end;

end;
DisplayForm.Cursor := crDefault;
ProcessingFile_Image.Visible := false;
if (Sender = MainForm) then begin
AssignFile(ImgFile, CURRENTLY_DISPLAYED_FILE_NAME);
Reset(ImgFile);
Label_OriginalFileSize.Caption := '( '+Inttostr(FileSize(ImgFile) div
1024)+' Kb )';
CloseFile(ImgFile);
DisplayImage.Width := DisplayImage.Picture.Width;
DisplayImage.Height := DisplayImage.Picture.Height;
DisplayImage.Top := 0;
DisplayImage.Left := 0;
ZoomFactorIndex.Position := 4; // Первісний розмір зображення
ShowModal;
end;

end;
procedure TDisplayForm.Button1Click(Sender: TObject);
var
LoadingFailure : boolean;
JPEGImg1 : TJpegImage;
JPEGImg2 : TJpegImage;
BMPTempImg : TBitmap;
TempStream : TMemoryStream;
ImgFile : file of Byte;
begin
LoadingFailure := false;
DisplayForm.Cursor := crHourGlass;

RefreshAdvice_Image.Visible := false;
RefreshAdvice_Label.Visible := false;
RefreshAdvice_Fond.Visible := false;
ProcessingFile_Image.Visible := true;
Application.ProcessMessages;

CURRENTLY_USED_COMPRESSION_RATE := CompressionRate.Position;
if RBtn_ShowOriginalImage.Checked then begin
try
DisplayImage.Picture.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME);

```

```

except
    MessageBoxA(MainForm.Handle, Pchar('Не вдається прочитати зображення.
Помилка формату. '), Pchar('Помилка'), MB_ICONSTOP + MB_SYSTEMMODAL +
MB_SETFOREGROUND + MB_TOPMOST);
    LoadingFailure := true;
end;
if LoadingFailure then begin
    if not (Sender = MainForm) then Close;
    DisplayForm.Cursor := crDefault;
    exit;
end;

end else begin
    JPEGImg1 := TJpegImage.Create;
    BMPTempImg := TBitmap.Create;
    JPEGImg2 := TJpegImage.Create;
    TempStream := TMemoryStream.Create;
    try
        if (UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.JPEG') or
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='._OBROBKA_TA_KONVERTA
CIA_ZOBRAGEN') or
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.SHM') then begin
//-----
        JPEGImg1.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME); // Завантаження
файлу
        Application.ProcessMessages;
        BMPTempImg.Assign(JPEGImg1);
        end else if
(UpperCase(ExtractFileExt(CURRENTLY_DISPLAYED_FILE_NAME))='.BMP') then begin
        BMPTempImg.LoadFromFile(CURRENTLY_DISPLAYED_FILE_NAME); //
Завантаження файлу
        end;
        JPEGImg2.Assign(BMPTempImg);
        Application.ProcessMessages;
        JPEGImg2.CompressionQuality:= CompressionRate.Position;
        JPEGImg2.Compress;
        JPEGImg2.SaveToStream(TempStream);
        TempStream.Position:=0;
        JPEGImg2.LoadFromStream(TempStream);
        CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE := '(
'+Inttostr(TempStream.Size div 1024)+' Kb)';
        Label_CompressedFileSize.Caption :=
CURRENTLY_DETERMINED_COMPRESSED_FILE_SIZE;
        DisplayImage.Picture.Assign(JPEGImg2);
        finally
            TempStream.Free;
            JPEGImg2.Free;
            BMPTempImg.Free;
            JPEGImg1.Free;
        end;
    except
        MessageBoxA(MainForm.Handle, Pchar('Не вдається прочитати зображення.
Помилка формату.'+#13+'Le fichier peut avoir йтй effacй ou ktre
corrompu. '), Pchar('Помилка'), MB_ICONSTOP + MB_SYSTEMMODAL + MB_SETFOREGROUND +
MB_TOPMOST);
        LoadingFailure := true;
    end;
    if LoadingFailure then begin
        if not (Sender = MainForm) then Close;
        DisplayForm.Cursor := crDefault;
        exit;
    end;

end;

DisplayForm.Cursor := crDefault;
ProcessingFile_Image.Visible := false;

if (Sender = MainForm) then begin

```

```
AssignFile(ImgFile, CURRENTLY_DISPLAYED_FILE_NAME);
Reset(ImgFile);
Label_OriginalFileSize.Caption := '(' +Inttostr(FileSize(ImgFile) div
1024)+' Kb)';
CloseFile(ImgFile);
DisplayImage.Width := DisplayImage.Picture.Width;
DisplayImage.Height := DisplayImage.Picture.Height;
DisplayImage.Top := 0;
DisplayImage.Left := 0;
ZoomFactorIndex.Position := 4; // Первісний розмір зображення
ShowModal;
end;

end;

end.
```

КБПЗ - 2023

Error.pas - файл обработки ошибок

```
unit Error;

interface

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

type
  TErrorForm = class(TForm)
    ErrorTitleMsg: TLabel;
    FilesConcernedMemo: TMemo;
    CloseBtn: TBitBtn;
    ClipboardCopyBtn: TBitBtn;
    procedure ClipboardCopyBtnClick(Sender: TObject);
    procedure CloseBtnClick(Sender: TObject);
  private
    { Private Declarations }
  public
    { Public Declarations }
  end;

var
  ErrorForm: TErrorForm;

implementation

{$R *.dfm}

procedure TErrorForm.CloseBtnClick(Sender: TObject);
begin
  Close;
end.
```

Main.pas - основна програма

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, ExtCtrls, Buttons, ShellCtrls, XPMan, Types,
  ShellAPI, FileCtrl, Math, Jpeg, OleCtrls, StrUtils, DateUtils, Zip, SfxUtils,
  IniFiles;
type
  TMainForm = class(TForm)
    Panel_Parameters: TGroupBox;
    Panel_Files: TGroupBox;
    Panel_Compression: TGroupBox;
    Panel_Destination: TGroupBox;
    Btn_CloseApp: TBitBtn;
    Btn_AboutApp: TBitBtn;
    Btn_BeginCompression: TBitBtn;
    Btn_AddFile: TBitBtn;
    Btn_DeleteFile: TBitBtn;
    Btn_ViewFile: TBitBtn;
    CompressionRate: TTrackBar;
    RBtn_KeepOGSize: TRadioButton;
    RBtn_ChangeOGSize: TRadioButton;
    RBtn_ModifiedSize: TRadioGroup;
    RBtn_OverwriteFiles: TRadioButton;
    RBtn_CreateNewFolder: TRadioButton;
    RBtn_CreateZIPFile: TRadioButton;
    CBox_CreateBackup: TCheckBox;
    RBtn_KeepAllMetadata: TRadioButton;
    RBtn_RemoveAllMetadata: TRadioButton;
    CBox_ChangeDateData: TCheckBox;
    FilesSelectedDate: TDateTimePicker;
    CBox_AddCopyright: TCheckBox;
    CopyrightText: TEdit;
    Btn_ChangeCopyrightTextFormat: TBitBtn;
    PBar_ProgressionStatus: TProgressBar;
    Label_Title1: TLabel;
    Label_Title2: TLabel;
    Label_Advice1: TLabel;
    Label_Advice2: TLabel;
    Label_CompressionRate: TLabel;
    Label_ProgressionStatus: TLabel;
    SaveZIPFileDialog: TSaveDialog;
    CopyrightPositionTL: TShape;
    CopyrightPositionTR: TShape;
    CopyrightPositionBL: TShape;
    CopyrightPositionBR: TShape;
    XPManifest: TXPManifest;
    Label_Title3: TLabel;
    AddImageDialog: TOpenDialog;
    FileList: TListBox;
    BackupPath: TEdit;
    CopyToDirectoryPath: TEdit;
    ExportZipFilePath: TEdit;
    Btn_BrowseBackupPath: TBitBtn;
    Btn_BrowseCopyToDirectoryPath: TBitBtn;
    Btn_BrowseExportZipFilePath: TBitBtn;
    CopyrightFontDialog: TFontDialog;
    FilesSelectedTime: TDateTimePicker;
    CBox_DeleteThumbnails: TCheckBox;
    CBox_AddGlobalComment: TCheckBox;
    CommentText: TEdit;
    Labell: TLabel;
    Imagel: TImage;
```

```

Timer1: TTimer;
procedure RBtn_ModifiedSizeClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Btn_AboutAppClick(Sender: TObject);
procedure CBox_AddGlobalCommentClick(Sender: TObject);
procedure Btn_BeginCompressionClick(Sender: TObject);
procedure Btn_ChangeCopyrightTextFormatClick(Sender: TObject);
procedure SelectedProfileChange(Sender: TObject);
procedure Btn_BrowseExportZipFilePathClick(Sender: TObject);
procedure Btn_BrowseCopyToDirectoryPathClick(Sender: TObject);
procedure Btn_BrowseBackupPathClick(Sender: TObject);
procedure CopyrightPositionMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure CBox_AddCopyrightClick(Sender: TObject);
procedure RBtn_MetadataChoiceClick(Sender: TObject);
procedure RBtn_DestinationPathChoiceClick(Sender: TObject);
procedure RBtn_OGSizeChoiceClick(Sender: TObject);
procedure CompressionRateChange(Sender: TObject);
procedure Btn_CloseAppClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Btn_ViewFileClick(Sender: TObject);
procedure Btn_DeleteFileClick(Sender: TObject);
procedure Btn_AddFileClick(Sender: TObject);
procedure ChangeControlsDisponibility(EnableControls: boolean);
procedure FileListDrawItem(Control: TWinControl; Index: Integer; Rect:
TRect; State: TOwnerDrawState);
procedure FileListKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);
procedure FileListClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure AddFilesFromPath(Path: string);
procedure FilesSelectedTimeChange(Sender: TObject);
procedure LoadProfiles();
procedure ReadProfile(ProfileIndex: integer);
procedure Timer1Timer(Sender: TObject);
private
  { Private Declarations }
  ZipComponent : TZip;
public
  { Public Declarations }
  procedure DefaultHandler(var msg); override;
end;

var
  MainForm: TMainForm;

  WM_PARAM_ATOM: cardinal;
  ErrorsList: TStringList;
  JPEGImage1: TJpegImage;
  JPEGImage2: TJpegImage;
  //      \
  //      /      BMPTempImage1: TBitmap;
  BMPTempImage2: TBitmap;
  FIRST_EXECUTION: boolean = true;
  PROFILE_APPLYING: boolean = false;
  false; //      \      IS_PROCESSING_IMAGES: boolean =
  ABORT_REQUESTED: boolean = false;
  APP_PATH: string;
  EXIFTOOL_PATH: string;
  TEMP_PATH: string;
  PROFILES_CFGFILE: string;
  //      /
  //      /

implementation

uses Error, About, Preview;

{$R *.dfm}

```

```

//Функція запуску та чекання завантаження

function LaunchAndWait(CommandLine: String; WShowWin: Word): Boolean;
var
  StartInfo    : TStartupInfo;
  ProcessInfo  : TProcessInformation;
  Ended        : Boolean;
begin
  FillChar(StartInfo, SizeOf(StartInfo), #0);
  StartInfo.cb := SizeOf(StartInfo);
  StartInfo.dwFlags := STARTF_USESHOWWINDOW;
  StartInfo.wShowWindow := WShowWin;
  if CreateProcess(nil, PChar(CommandLine), nil, nil, False, 0, nil, nil,
  StartInfo, ProcessInfo) then begin
    Ended := False;
    repeat
      case WaitForSingleObject(ProcessInfo.hProcess, 200) of
        WAIT_OBJECT_0: Ended := True;
        WAIT_TIMEOUT  : ;
      end;
      Application.ProcessMessages;
    until Ended;
    result := true;
  end else result := false;
end;

// ----- //
//Функція видалення директорії
// ----- //
function DeleteDirectory(Path: string): Boolean;
var
  fos: TSHFileOpStruct;
begin
  Path := ExcludeTrailingPathDelimiter(Path);
  ZeroMemory(@fos, SizeOf(fos));
  with fos do begin
    wFunc := FO_DELETE;
    fFlags := FOF_SILENT or FOF_NOCONFIRMATION;
    pFrom := PChar(Path + #0);
  end;
  Result := (0=ShFileOperation(fos));
end;

// ----- //
Зміна дати та часу файлу
// ----- //
function CorruptFileDateTime(const FileName: string; NewDate: TDateTime):
boolean;
var
  fHandle : integer;
  Succeed : boolean;
  FinalDate, TempFileTime : TFileTime;
  TempSystemTime : TSystemTime;
begin
  fHandle := FileOpen(FileName, fmShareDenyWrite or fmOpenWrite);
  if fHandle < 0 then Succeed := false
  else begin
    DecodeDateTime(NewDate, TempSystemTime.wYear, TempSystemTime.wMonth,
    TempSystemTime.wDay,
    TempSystemTime.wHour, TempSystemTime.wMinute,
    TempSystemTime.wSecond, TempSystemTime.wMilliseconds);
    SystemTimeToFileTime(TempSystemTime, TempFileTime);
    LocalFileTimeToFileTime(TempFileTime, FinalDate);
    Succeed := SetFileTime(fHandle, @FinalDate, @FinalDate, @FinalDate);
    FileClose(fHandle);
  end;
  Result := Succeed;
end;

```

```

// ----- //
Визначення імені файлу
// ----- //
function ExtractFileNameOnly(FileName:TFileName): TFileName;
var
  ExtensionPart : TFileName;
  ExtensionLength : Integer;
begin
  FileName := ExtractFileName(FileName);
  ExtensionPart := ExtractFileExt(FileName);
  ExtensionLength := Length(ExtensionPart);
  Delete(FileName, Length(FileName)-ExtensionLength+1,ExtensionLength);
  Result:=FileName;
end;
// ----- //
Пошук імені доступного файлу
// ----- //
function FindAvailableFileName(Directory, FileName, FileExtension: string):
string;
var
  FileDuplicationChars : string;
  FileDuplicationIndex : integer;
begin
  Directory := IncludeTrailingPathDelimiter(Directory);
  if FileExists(Directory + FileName + FileExtension) then begin
    FileDuplicationIndex := 2;
    FileDuplicationChars := ' ('+IntToStr(FileDuplicationIndex)+)';
    while FileExists(Directory + FileName + FileDuplicationChars +
FileExtension) do begin
      Inc(FileDuplicationIndex);
      FileDuplicationChars := ' ('+IntToStr(FileDuplicationIndex)+)';
    end;
    end else FileDuplicationChars := '';

    result := Directory + FileName + FileDuplicationChars + FileExtension;
end;
// ----- //
Створення рядка
// ----- //
function RandomString(GeneratedStringLength: integer): string;
var
  i: integer;
  BaseChars, TempStr: string;
begin
  BaseChars := 'abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNPQRSTUVWXYZ';
  // Available Chars (they need to be compatible with a FileName)
  for i:=0 to GeneratedStringLength-1 do TempStr := TempStr +
BaseChars[Random(61)+1];
  result := TempStr;
end;

// Procedures Concerning Buttons Management
procedure TMainForm.Btn_AddFileClick(Sender: TObject);
var
  i: integer;
begin
  if AddImageDialog.Execute then begin
    ErrorsList := TStringList.Create;
    ErrorsList.Clear;
    for i:=0 to AddImageDialog.Files.Count-1 do begin
      if FileExists(AddImageDialog.Files[i]) then
FileList.Items.Add(AddImageDialog.Files[i]);
    end;
    if (ErrorsList.Text<>'') then begin
      ErrorForm.Caption := 'Помилка при додаванні файлів';
      ErrorForm.ErrorTitleMsg.Caption := 'Ці файли не можуть бути додані в список
зображень для стиснення '+ #13 +', тому що їх формат не підтримується
(підтримуються формати _OBROBKA_TA_KONVERTACIA_ZOBRAGEN, JPEG, BMP)';
      ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
    end;
  end;
end;

```

```

        ErrorForm.ShowModal;
    end;
    ErrorsList.Free;
end;
end;
// ----- //
Додавання шляху до файлу
// ----- //
procedure TMainForm.AddFilesFromPath(Path: string);
var
    SearchInfo: TSearchRec;
begin
    if FileExists(Path) then begin
        if (UpperCase(ExtractFileExt(Path)) = '_OBROBKA_TA_KONVERTACIA_ZOBRAGEN')
            or (UpperCase(ExtractFileExt(Path)) = '.JPEG')
            or (UpperCase(ExtractFileExt(Path)) = '.BMP')
            or (UpperCase(ExtractFileExt(Path)) = '.SHM')
        then FileList.Items.Add(Path)
        else ErrorsList.Add(Path);
        exit;
    end;
    Path := IncludeTrailingPathDelimiter(Path);
    if DirectoryExists(Path) then begin
        if FindFirst(Path+'*.*', faAnyFile, SearchInfo)=0 then begin
            repeat
                if (SearchInfo.Name[1]<>'.') then
                    AddFilesFromPath(Path+SearchInfo.FindData.cFileName);
            until FindNext(SearchInfo)<>0;
            FindClose(SearchInfo);
        end;
    end;
end;
// ----- //
Знищення файлу
// ----- //
procedure TMainForm.Btn_DeleteFileClick(Sender: TObject);
begin
    FileList.DeleteSelected;
    Btn_DeleteFile.Enabled := false;
    Btn_ViewFile.Enabled := false;
    FileList.ItemIndex:=-1;
end;
// ----- //
Перегляд файлу
// ----- //
procedure TMainForm.Btn_ViewFileClick(Sender: TObject);
begin
    // Preview of the focused image
    if FileList.ItemIndex = -1 then exit;
    DisplayForm.LoadPreview(FileList.Items[FileList.ItemIndex]);
end;
//=====//
// Виведення списку файлів та вибір файлу
// ----- //
procedure TMainForm.FileListClick(Sender: TObject);
begin
    Btn_DeleteFile.Enabled := (FileList.SelCount <> 0);
    Btn_ViewFile.Enabled := (FileList.SelCount <> 0);
end;
// ----- //
Поле списку файлів
// ----- //
procedure TMainForm.FileListDrawItem(Control: TWinControl; Index: Integer; Rect:
TRect; State: TOwnerDrawState);
begin
    if odSelected in State then begin
        FileList.Canvas.Brush.Color := clInfoBk;
        FileList.Canvas.FillRect(Rect);
    end else begin

```

```

    FileList.Canvas.Brush.Color := clWhite;
    FileList.Canvas.FillRect(Rect);
end;
FileList.Canvas.Font.Name := 'Tahoma';
FileList.Canvas.Font.Size := 8;
FileList.Canvas.Font.Color := clblack;
FileList.Canvas.TextOut(Rect.Left+2, Rect.Top,
ExtractFileName(FileList.Items[Index]));
end;
// ----- //
procedure TMainForm.FileListKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
begin
    if (Key=VK_DELETE) then Btn_DeleteFileClick(nil);
    if (Key=VK_INSERT) then Btn_AddFileClick(nil);
    if (Key=VK_RETURN) then Btn_ViewFileClick(nil);
end;
// ----- //
Робота з файлами зображень
// ----- //
procedure TMainForm.DefaultHandler(var msg);
var
    ReceivedParam: PChar;
    ReceivedAtom: atom;
    params: TStringList;
    i, NumberOfFiles: integer;
    StrFileName : string;
    FileName : array[0..255] of char;
begin
    inherited DefaultHandler(Msg);
    if IS_PROCESSING_IMAGES then exit;
    if TMessage(msg).Msg = WM_PARAM_ATOM then begin
        ReceivedAtom := TMessage(msg).wParam;
        GetMem(ReceivedParam, 256);
        try
            GlobalGetAtomName(ReceivedAtom, ReceivedParam, 256);
            try
                params := TStringList.Create;
                ErrorsList := TStringList.Create;
                try
                    params.Clear;
                    params.Add(ReceivedParam);
                    ErrorsList.Clear;
                    for i := 0 to params.Count-1 do AddFilesFromPath(params[i]);
                    if (ErrorsList.Text<>') then begin
                        ErrorForm.Caption := 'Помилка при додаванні файлів';
                        ErrorForm.ErrorTitleMsg.Caption := 'Ці файли не можуть бути додані в
список зображень для стиснення '+ #13 '+', тому що їх формат не підтримується
(підтримуються формати OBROBKA TA KONVERTACIA ZOBRAGEN, JPEG, BMP):';
                        ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
                        ErrorForm.ShowModal;
                    end;
                    finally ErrorsList.Free; params.Free; end;
                    finally GlobalDeleteAtom(ReceivedAtom); end;
                    finally FreeMem(ReceivedParam); end;
                end;
            end;
        // ----- . .
        // Перетаскування файлів
        if TMessage(msg).Msg=WM_DROPFILES then begin
            NumberOfFiles := DragQueryFile(TMessage(msg).wParam, $FFFFFFFF, FileName,
sizeof(FileName));
            ErrorsList := TStringList.Create;
            ErrorsList.Clear;
            for i := 0 to NumberOfFiles-1 do
                begin
                    DragQueryFile(TMessage(msg).wParam, i, FileName, sizeof(FileName));
                    StrFileName := FileName;
                    AddFilesFromPath(StrFileName); // Calling for recursive addition procedure
                end;
        end;
    end;
end;

```

```

    if (ErrorsList.Text<>')') then begin
        ErrorForm.Caption := 'Помилка при додаванні файлів';
        ErrorForm.ErrorTitleMsg.Caption := 'Ці файли не можуть бути додані в
        список зображень для стиснення '+ #13 +', тому що їх формат не підтримується
        (підтримуються формати _OBROBKA_TA_KONVERTACIA_ZOBRAGEN, JPEG, BMP):';
        ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
        ErrorForm.ShowModal;
    end;
    ErrorsList.Free;
end;
end;
// ----- //
Виведення вікна
// ----- //
procedure TMainForm.FormShow(Sender: TObject);
var
    i: integer;
begin
    if not FIRST_EXECUTION then exit
    else FIRST_EXECUTION := false;
    WM_PARAM_ATOM := RegisterWindowMessage('WM_PARAM_ATOM');
    if ParamCount = 0 then exit;
    ErrorsList := TStringList.Create;
    try
        ErrorsList.Clear;
        for i := 1 to ParamCount do AddFilesFromPath(ParamStr(i));    if
        (ErrorsList.Text<>')') then begin
            ErrorForm.Caption := 'Помилка при додаванні файлів';
            ErrorForm.ErrorTitleMsg.Caption := 'Ці файли не можуть бути додані в
            список зображень для стиснення '+ #13 +', тому що їх формат не підтримується
            (підтримуються формати _OBROBKA_TA_KONVERTACIA_ZOBRAGEN, JPEG, BMP):';
            ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
            ErrorForm.ShowModal;
        end;
        finally ErrorsList.Free; end;
    end;
    procedure TMainForm.CompressionRateChange(Sender: TObject);
    begin

        DisplayForm.CompressionRate.Position:=MainForm.CompressionRate.Position;

        Label_CompressionRate.Caption := IntToStr(CompressionRate.Position)+' %';
    end;
    // ----- //
    Вибір коефіцієнту стискання
    // ----- //
    procedure TMainForm.RBtn_OGSizeChoiceClick(Sender: TObject);
    begin
        RBtn_ModifiedSize.Enabled := RBtn_ChangeOGSize.Checked;
    end;
    // ----- //
    procedure TMainForm.RBtn_ModifiedSizeClick(Sender: TObject);
    begin
        // if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
    end; //=====//
    procedure TMainForm.RBtn_DestinationPathChoiceClick(Sender: TObject);
    begin
        // if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
        CBox_CreateBackup.Enabled := RBtn_OverwriteFiles.Checked;
        BackupPath.Enabled := CBox_CreateBackup.Checked and
        RBtn_OverwriteFiles.Checked;
        Btn_BrowseBackupPath.Enabled := CBox_CreateBackup.Checked and
        RBtn_OverwriteFiles.Checked;
        CopyToDirectoryPath.Enabled := RBtn_CreateNewFolder.Checked;
        Btn_BrowseCopyToDirectoryPath.Enabled := RBtn_CreateNewFolder.Checked;
        ExportZipFilePath.Enabled := RBtn_CreateZIPFile.Checked;
        Btn_BrowseExportZipFilePath.Enabled := RBtn_CreateZIPFile.Checked;
    end;
end;

```

```

// ----- //
Вибір місця розташування резервної копії
// ----- //
procedure TMainForm.Btn_BrowseBackupPathClick(Sender: TObject);
var
  SelectedPath: string;
begin
  SelectedPath := BackupPath.Text;
  {$IFDEF VER170}
  if not SelectDirectory('Місце розташування резервної копії', '', SelectedPath,
[sdNewUI, sdNewFolder, sdShowEdit, sdValidateDir, sdShowShares], nil) then exit;
  {$ELSE}
  if not SelectDirectory('Місце розташування резервної копії', '', SelectedPath)
then exit;
  {$ENDIF}
  BackupPath.Text := IncludeTrailingPathDelimiter(SelectedPath);
end;
// ----- //
Місце розташування зображення
// ----- //
procedure TMainForm.Btn_BrowseCopyToDirectoryPathClick(Sender: TObject);
var
  SelectedPath: string;
begin
  SelectedPath := CopyToDirectoryPath.Text;
  {$IFDEF VER170}
  if not SelectDirectory('Місце розташування зображення', '', SelectedPath,
[sdNewUI, sdNewFolder, sdShowEdit, sdValidateDir, sdShowShares], nil) then exit;
  {$ELSE}
  if not SelectDirectory('Місце розташування зображення', '', SelectedPath) then
exit;
  {$ENDIF}
  CopyToDirectoryPath.Text := IncludeTrailingPathDelimiter(SelectedPath);
end;
// ----- //
procedure TMainForm.Btn_BrowseExportZipFilePathClick(Sender: TObject);
begin
  SaveZIPFileDialog.FileName := ExportZipFilePath.Text;
  if not SaveZIPFileDialog.Execute then exit;
  ExportZipFilePath.Text := SaveZIPFileDialog.FileName;
end;
//=====//
procedure TMainForm.RBtn_MetadataChoiceClick(Sender: TObject);
begin
// if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
  CBox_ChangeDateData.Enabled := RBtn_RemoveAllMetadata.Checked;
  CBox_DeleteThumbnails.Enabled := RBtn_RemoveAllMetadata.Checked;
  FilesSelectedDate.Enabled := CBox_ChangeDateData.Checked and
RBtn_RemoveAllMetadata.Checked;
  FilesSelectedTime.Enabled := CBox_ChangeDateData.Checked and
RBtn_RemoveAllMetadata.Checked;
end;
// ----- //
procedure TMainForm.CBox_AddCopyrightClick(Sender: TObject);
begin
// if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
  CopyrightText.Enabled := CBox_AddCopyright.Checked;
  Btn_ChangeCopyrightTextFormat.Enabled := CBox_AddCopyright.Checked;

  if CBox_AddCopyright.Checked then begin

    Label_Title3.Font.Color := clBlack;

    if CopyrightPositionTL.Tag=0 then begin
      CopyrightPositionTL.Cursor := crHandPoint; CopyrightPositionTL.Brush.Color
:= clWhite;
    end else begin
      CopyrightPositionTL.Cursor := crArrow; CopyrightPositionTL.Brush.Color :=
clRed;

```

```

end;

if CopyrightPositionBL.Tag=0 then begin
    CopyrightPositionBL.Cursor := crHandPoint; CopyrightPositionBL.Brush.Color
:= clWhite;
end else begin
    CopyrightPositionBL.Cursor := crArrow; CopyrightPositionBL.Brush.Color :=
clRed;
end;

if CopyrightPositionTR.Tag=0 then begin
    CopyrightPositionTR.Cursor := crHandPoint; CopyrightPositionTR.Brush.Color
:= clWhite;
end else begin
    CopyrightPositionTR.Cursor := crArrow; CopyrightPositionTR.Brush.Color :=
clRed;
end;

if CopyrightPositionBR.Tag=0 then begin
    CopyrightPositionBR.Cursor := crHandPoint; CopyrightPositionBR.Brush.Color
:= clWhite;
end else begin
    CopyrightPositionBR.Cursor := crArrow; CopyrightPositionBR.Brush.Color :=
clRed;
end;

end else begin

    Label_Title3.Font.Color := clMedGray;
    CopyrightPositionTL.Brush.Color := clBtnFace;
    CopyrightPositionTR.Brush.Color := clBtnFace;
    CopyrightPositionBL.Brush.Color := clBtnFace;
    CopyrightPositionBR.Brush.Color := clBtnFace;

    CopyrightPositionTL.Cursor := crArrow; CopyrightPositionTR.Cursor :=
crArrow;
    CopyrightPositionBL.Cursor := crArrow; CopyrightPositionBR.Cursor :=
crArrow;

end;
end;
// ----- //
procedure TMainForm.CopyrightPositionMouseUp(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
// if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
if not CBox_AddCopyright.Checked then exit;
CopyrightPositionTL.Tag := 0; CopyrightPositionTR.Tag := 0;
CopyrightPositionBL.Tag := 0; CopyrightPositionBR.Tag := 0;
(Sender as TShape).Tag := 1;
CBox_AddCopyrightClick(nil);
end;
// ----- //
procedure TMainForm.CBox_AddGlobalCommentClick(Sender: TObject);
begin
// if not PROFILE APPLYING then SelectedProfile.ItemIndex := 0;
CommentText.Enabled := CBox_AddGlobalComment.Checked;
end;
// ----- //
procedure TMainForm.Btn_ChangeCopyrightTextFormatClick(Sender: TObject);
begin
CopyrightFontDialog.Execute;
end;
// ----- //
procedure TMainForm.FilesSelectedTimeChange(Sender: TObject);
begin
FilesSelectedDate.Time := FilesSelectedTime.Time;
end;
//=====//

```

```

// Profiles Management
procedure TMainForm.LoadProfiles;
var
  NumberOfProfiles, DefaultIndex, i: integer;
  TempProfileName: string;
begin
  // SelectedProfile.Items.Text := 'Налагодження...';
  if not FileExists(PROFILES_CFGFILE) then begin
    exit;
  end;

  with TIniFile.Create(PROFILES_CFGFILE) do try
    if (ReadString('HEADER','ID','error') <> '17D5B2702D39F1A6C1E3ACEDAE99CC45')
  then exit;
    NumberOfProfiles := ReadInteger('HEADER','NumberOfProfiles_int',0);
    if NumberOfProfiles <= 0 then begin
      // SelectedProfile.ItemIndex := 0;
      exit;
    end;
    for i:=1 to NumberOfProfiles do begin
      TempProfileName := ReadString('PROFILE'+IntToStr(i),'Name_str','');
      if TempProfileName='' then TempProfileName:='Невідомий профіль';
      // SelectedProfile.Items.Add(TempProfileName);
      end;
      // Default ПроЗавантаження файлу
      DefaultIndex := ReadInteger('HEADER','DefaultProfile_int',0);
      // if (DefaultIndex>0) and (DefaultIndex<SelectedProfile.Items.Count) then
      SelectedProfile.ItemIndex := DefaultIndex
      // else SelectedProfile.ItemIndex := 0;
      // ReadProfile(SelectedProfile.ItemIndex);

    finally
      Free;
    end;
  end;
  // ----- //
procedure TMainForm.ReadProfile(ProfileIndex: integer);
var
  ProfileID: string;
  TempRate: integer;
begin
  if (ProfileIndex=-1) or (ProfileIndex=0) then exit;
  if not FileExists(PROFILES_CFGFILE) then begin
    LoadProfiles;
    exit;
  end;
  with TIniFile.Create(PROFILES_CFGFILE) do try
    if ReadString('HEADER','ID','error') <> '17D5B2702D39F1A6C1E3ACEDAE99CC45'
  then exit;
    ProfileID := 'PROFILE'+IntToStr(ProfileIndex);

    if not SectionExists(ProfileID) then begin
      LoadProfiles;
      exit;
    end;

    PROFILE APPLYING := true;
    TempRate := ReadInteger(ProfileID,'Compression_Rate_int',80);
    if (TempRate>0) and (TempRate<=100) then CompressionRate.Position :=
TempRate
    else CompressionRate.Position := 80;
    if ReadBool(ProfileID,'Compression_ReduceSize_bool',false) then begin
      RBtn_ChangeOGSize.Checked := true;
      RBtn_ModifiedSize.ItemIndex :=
ReadInteger(ProfileID,'Compression_NewSizeIndex_int',2);
    end else RBtn_KeepOGSize.Checked := true;
    case ReadInteger(ProfileID,'Destination_Index_int',0) of
      1: RBtn_CreateNewFolder.Checked := true;
      2: RBtn_CreateZIPFile.Checked := true;
    end;
  end;
end;

```

```

{0:} else begin
    RBtn_OverwriteFiles.Checked := true;
    CBox_CreateBackup.Checked :=
ReadBool(ProfileID, 'Destination_MakeBackup_bool', false);
    end;
end;
if ReadBool(ProfileID, 'Metadata_DeleteMetadata_bool', false) then begin
    RBtn_RemoveAllMetadata.Checked := true;
    CBox_DeleteThumbnails.Checked :=
ReadBool(ProfileID, 'Metadata_DeleteThumbnails_bool', false);
    if ReadBool(ProfileID, 'Metadata_ChangeDate_bool', false) then begin
        CBox_ChangeDateData.Checked := true;
        FilesSelectedDate.Date :=
ReadDate(ProfileID, 'Metadata_NewDate_date', now);
        FilesSelectedTime.Time :=
ReadTime(ProfileID, 'Metadata_NewTime_time', now);
        end else CBox_ChangeDateData.Checked := false;
    end else RBtn_KeepAllMetadata.Checked := true;
    if ReadBool(ProfileID, 'Metadata_AddComment_bool', false) then begin
        CBox_AddGlobalComment.Checked := true;
        CommentText.Text := ReadString(ProfileID, 'Metadata_Comment_str', '');
    end else CBox_AddGlobalComment.Checked := false;
    if ReadBool(ProfileID, 'Metadata_AddCopyright_bool', false) then begin
        CBox_AddCopyright.Checked := true;
        CopyrightText.Text := ReadString(ProfileID, , '© Copyright ');
        case ReadInteger(ProfileID, 'Metadata_CopyrightPosition_int', 4) of
            // 1=TopLeft, 2=TopRight, 3=BottomLeft, 4=BottomRight
            1: CopyrightPositionMouseUp(CopyrightPositionTL, mbLeft, [], 1, 1);
            2: CopyrightPositionMouseUp(CopyrightPositionTR, mbLeft, [], 1, 1);
            3: CopyrightPositionMouseUp(CopyrightPositionBL, mbLeft, [], 1, 1);
            {4:} else CopyrightPositionMouseUp(CopyrightPositionBR, mbLeft, [], 1,
1);
        end;
        CopyrightFontDialog.Font.Name :=
ReadString(ProfileID, 'Metadata_CopyrightFontName_str', 'Tahoma');
        // FontName Validity Check :
        if (Screen.Fonts.IndexOf(CopyrightFontDialog.Font.Name) = -1) or
(CopyrightFontDialog.Font.Name='') then CopyrightFontDialog.Font.Name:='Tahoma';
        try CopyrightFontDialog.Font.Size :=
ReadInteger(ProfileID, 'Metadata_CopyrightFontSize_int', 10);
        except CopyrightFontDialog.Font.Size := 10 end;
        try CopyrightFontDialog.Font.Color :=
StringToColor(ReadString(ProfileID, 'Metadata_CopyrightFontColor_cl', 'clBlack'));
        except CopyrightFontDialog.Font.Color := clBlack end;
        CopyrightFontDialog.Font.Style := [];
        if ReadBool(ProfileID, 'Metadata_CopyrightFontStyleBold_bool', true) then
CopyrightFontDialog.Font.Style := CopyrightFontDialog.Font.Style + [fsBold];
        if ReadBool(ProfileID, 'Metadata_CopyrightFontStyleItalic_bool', false) then
CopyrightFontDialog.Font.Style := CopyrightFontDialog.Font.Style + [fsItalic];
        if ReadBool(ProfileID, 'Metadata_CopyrightFontStyleUnderline_bool', false)
then CopyrightFontDialog.Font.Style := CopyrightFontDialog.Font.Style +
[fsUnderline];
        if ReadBool(ProfileID, 'Metadata_CopyrightFontStyleStrikeOut_bool', false)
then CopyrightFontDialog.Font.Style := CopyrightFontDialog.Font.Style +
[fsStrikeOut];
        end else CBox_AddCopyright.Checked := false;
        finally
            Free;
            PROFILE_APPLYING := false;
        end;
    end;
end;
// ----- //
procedure TMainForm.SelectedProfileChange(Sender: TObject);
begin
    // ReadProfile(SelectedProfile.ItemIndex);
end;
//=====//
procedure TMainForm.ChangeControlsDisponibility(EnableControls: boolean);
begin

```

```

Panel_Files.Enabled := EnableControls;
Panel_Parameters.Enabled := EnableControls;
Panel_Destination.Enabled := EnableControls;
Panel_Compression.Enabled := EnableControls;
// SelectedProfile.Enabled := EnableControls;
Btn_BeginCompression.Enabled := EnableControls;

if EnableControls then Btn_CloseApp.Caption := 'Quarter'
else Btn_CloseApp.Caption := 'Annuler';

Application.ProcessMessages;
end;
// ----- //
procedure TMainForm.Btn_CloseAppClick(Sender: TObject);
begin
  if not IS_PROCESSING_IMAGES then Application.Terminate
  else ABORT_REQUESTED := true;
end;
// ----- //
procedure TMainForm.Btn_AboutAppClick(Sender: TObject);
begin
  AboutForm.ShowModal;
end;
//=====//
procedure TMainForm.Btn_BeginCompressionClick(Sender: TObject);
var
  i,j: integer;
  FileName, FinalDestinationFileName, FinalDestinationPath,
  OriginalFileExtension, ZIPFileName,
  BackupDestinationFileName, BackupDestinationPath, MetaDataSourceFileName:
string;
  xOG, yOG, xRED, yRED, tmp: integer;
  xratio, yratio, finalratio: extended;
  xText, yText: integer;
  FieldsCorrectlyCompleted, IsCorrectlyLoaded, IsCorrectlySaved,
IsLandscapeFormat: boolean;
  BMPTempImage3 : TBitmap;
  FormattedNewFileDateTime: string;
  ZIPPendingFiles: TStringList;
begin
  if FileList.Items.Count=0 then begin
    MessageBoxA(Handle, Pchar('Немає вибраних файлів'), Pchar('Помилка'),
MB_ICONSTOP + MB_SYSTEMMODAL + MB_SETFOREGROUND + MB_TOPMOST);
    exit;
  end;

  if ( (RBtn_OverwriteFiles.Checked) and (CBox_CreateBackup.Checked) and
(BackupPath.Text='') )
  or ( (RBtn_CreateNewFolder.Checked) and (CopyToDirectoryPath.Text='') )
  or ( (RBtn_CreateZIPFile.Checked) and (ExportZipFilePath.Text ='' ) ) then
begin
  MessageBoxA(Handle, Pchar('Вказаного для збереження файлів каталогу не
існує'), Pchar('Помилка'), MB_ICONSTOP + MB_SYSTEMMODAL + MB_SETFOREGROUND +
MB_TOPMOST);
  exit;
end;

Label_ProgressionStatus.Caption := 'Ініціалізація процесу стиснення...';
ChangeControlsDisponibility(false);

IS_PROCESSING_IMAGES := true;
ABORT_REQUESTED := false;

PBar_ProgressionStatus.Position := 0;
PBar_ProgressionStatus.Max := FileList.Count;
Application.ProcessMessages;

ErrorsList := TStringList.Create;
ErrorsList.Clear;

```

```

.....
if RBtn_OverwriteFiles.Checked then begin
  // "Перезаписуємі файли" вибір
  if CBox_CreateBackup.Checked then begin
    BackupDestinationPath := BackupPath.Text;
  end else begin // Навіть якби це не було запрошено споживачем, нам потрібно
робити копію файлу для того, щоб бути здатним перемістити метадані в новий файл
якщо необхідно
    BackupDestinationPath := TEMP_PATH;
  end;
end else begin
  if RBtn_CreateNewFolder.Checked then begin
    FinalDestinationPath :=
IncludeTrailingPathDelimiter(CopyToDirectoryPath.Text);
  end else if RBtn_CreateZIPFile.Checked then begin
    ZIPFileName := ExportZipFilePath.Text;
    ZIPPendingFiles := TStringList.Create;
    ZIPPendingFiles.Clear;
    FinalDestinationPath := TEMP_PATH + ExtractFileNameOnly(ZIPFileName) +
RandomString(10);
  end;
end;
FOR i := 0 TO FileList.Count-1 DO BEGIN
  if ABORT_REQUESTED then begin
    Label_ProgressionStatus.Caption := 'Скасування...';
    Application.ProcessMessages;
    for j := i to FileList.Count-1 do
ErrorsList.Add(FileList.Items[j]+' : Припинено користувачем. ');
      Break;
    end;
    FileName := FileList.Items[i];
    Label_ProgressionStatus.Caption := 'Обробка файлів
'+ExtractFileName(FileName);
    Application.ProcessMessages;
    if not FileExists(FileName) then begin
      ErrorsList.Add(FileName+' : файл недоступний. ');
      Continue; // Compression Process Abort : Next Picture
    end;

    IsCorrectlyLoaded := true;
    try
      if (UpperCase(ExtractFileExt(FileName))='.JPEG') or
(UpperCase(ExtractFileExt(FileName))='._OBROBKA_TA_KONVERTACIA_ZOBRAGEN') then
begin
        JPEGImage1.LoadFromFile(FileName);
        Application.ProcessMessages;
        BMPTempImage1.Assign(JPEGImage1);
      end else if (UpperCase(ExtractFileExt(FileName))='.BMP') then
begin
        BMPTempImage1.LoadFromFile(FileName);
      end;
      if ((BMPTempImage1.Height<=1) or (BMPTempImage1.Width=0)) then
IsCorrectlyLoaded := false;
    except
      IsCorrectlyLoaded := false;
    end;
    if not IsCorrectlyLoaded then begin
      ErrorsList.Add(FileName+' : пошкоджений файл або некоректний
формат файлу');
      Continue;
    end;
    if RBtn_ChangeOGSize.Checked then begin
      xOG := BMPTempImage1.Width; yOG := BMPTempImage1.Height;

      if xOG>=yOG then IsLandscapeFormat:=true
      else begin //
        IsLandscapeFormat := false;
        tmp := xOG; xOG := yOG; yOG := tmp;
      end;
    end;
  end;
end;

```

```

xRED:=0; yRED:=0;
case RBtn_ModifiedSize.ItemIndex of
0:begin xRED:=640; yRED:=480; end;
1:begin xRED:=800; yRED:=600; end;
2:begin xRED:=1024; yRED:=768; end;
3:begin xRED:=1280; yRED:=1024; end;
4:begin xRED:=1600; yRED:=1200; end;
5:begin xRED:=2048; yRED:=1536; end;
end;
xratio:=(100*xRED)/xOG; yratio:=(100*yRED)/yOG;

if (xratio>=100) and (yratio>=100) then
BMPTempImage2.Assign(BMPTempImage1)
else begin
finalratio := MinValue([xratio, yratio]);
if not IsLandscapeFormat then begin
tmp := xOG; xOG := yOG; yOG := tmp;
end;
xRED := round(xOG*finalratio/100); yRED :=
round(yOG*finalratio/100);

if xRED=0 then xRED := 1; if yRED<=1 then yRED := 2;

BMPTempImage3 := TBitmap.Create;
try
BMPTempImage3.Width := xRED;
BMPTempImage3.Height := yRED;
BMPTempImage3.PixelFormat := BMPTempImage1.PixelFormat;
SetStretchBltMode(BMPTempImage3.Canvas.Handle, HALFTONE); //
Напівтон використаний для кращої вихідної якості образу
StretchBlt(BMPTempImage3.Canvas.Handle, 0, 0, xRED, yRED,
BMPTempImage1.Canvas.Handle, 0, 0, xOG, yOG, SRCCOPY);
BMPTempImage2.Width := xRED;
BMPTempImage2.Height := yRED;
BMPTempImage2.PixelFormat := BMPTempImage1.PixelFormat;
BMPTempImage2.Assign(BMPTempImage3);
finally
BMPTempImage3.Free;
end;
end;
end else begin
BMPTempImage2.Assign(BMPTempImage1);
Reduction Required : Direct Assignment
end;
if CBox_AddCopyright.Checked then begin
BMPTempImage2.Canvas.Brush.Style := BsClear;
BMPTempImage2.Canvas.Font := CopyrightFontDialog.Font;
xText:=30; yText:=30;
if CopyrightPositionTL.Tag=1 then begin
xText := 30;
yText := 30;
end else if CopyrightPositionBL.Tag=1 then begin
xText := 30;
yText := BMPTempImage2.Height-
(BMPTempImage2.Canvas.TextHeight(CopyrightText.Text)+30);
end else if CopyrightPositionTR.Tag=1 then begin
xText := BMPTempImage2.Width-
(BMPTempImage2.Canvas.TextWidth(CopyrightText.Text)+30);
yText := 30;
end else if CopyrightPositionBR.Tag=1 then begin
xText := BMPTempImage2.Width-
(BMPTempImage2.Canvas.TextWidth(CopyrightText.Text)+30);
yText := BMPTempImage2.Height-
(BMPTempImage2.Canvas.TextHeight(CopyrightText.Text)+30);
end;
BMPTempImage2.Canvas.TextOut(xText, yText, CopyrightText.Text);
end;
end;

```

```

        if RBtn_OverwriteFiles.Checked then begin
            OriginalFileExtension := ExtractFileExt(FileName);
            ForceDirectories(BackupDestinationPath);
            BackupDestinationFileName :=
FindAvailableFileName(BackupDestinationPath, ExtractFileNameOnly(FileName),
OriginalFileExtension);
            if not
CopyFile(PChar(FileName), PChar(BackupDestinationFileName), true) then begin
                ErrorsList.Add(FileName+' : Пошкоджений файл, або некоректний
формат файлу');
                Continue;
            end;
            if (OriginalFileExtension='.jpeg') then
RenameFile(FileName, ChangeFileExt(FileName, '_Obrobka_ta_konvertacia_zobragen'))
;
                Application.ProcessMessages;
                FinalDestinationFileName := ChangeFileExt(FileName, '.shm');
            end else begin

                ForceDirectories(FinalDestinationPath);
                Application.ProcessMessages;
                FinalDestinationFileName :=
FindAvailableFileName(FinalDestinationPath, ExtractFileNameOnly(FileName),
'.shm');

            end;
            IsCorrectlySaved := true;
            JPEGImage2 := TJpegImage.Create;
            try
                try
                    JPEGImage2.Assign(BMPTempImage2);
                    Application.ProcessMessages;
                    JPEGImage2.CompressionQuality := CompressionRate.Position;
                    JPEGImage2.Compress;
                    JPEGImage2.SaveToFile(FinalDestinationFileName);
                except
                    IsCorrectlySaved:= false;
                end;
            finally
                JPEGImage2.Free;
            end;
            if not IsCorrectlySaved then begin
                ErrorsList.Add(FileName+' : Файл не може бути стиснутий. Помилка
збереження.');
```

```

        LaunchAndWait(''+EXIFTOOL_PATH+''' -
Copyright=''+CopyrightText.Text+''' '''+FinalDestinationFileName+''' -P -q -m -L -
overwrite_original',SW_HIDE);
        Application.ProcessMessages;
    end;
    if RBtn_RemoveAllMetadata.Checked then begin
        if not CBox_DeleteThumbnails.Checked then begin
            //
            LaunchAndWait(''+EXIFTOOL_PATH+''' -TagsFromFile
'''+MetaDataSourceFileName+''' -ThumbnailImage '''+FinalDestinationFileName+''' -P -
q -m -overwrite_original',SW_HIDE);
            Application.ProcessMessages;
        end;
        if (RBtn_RemoveAllMetadata.Checked) and
(CBox_ChangeDateData.Checked) then begin
            FormattedNewFileDateTime := FormatDateTime('yyyy:mm:dd
hh:mm:ss', FilesSelectedDate.DateTime);
            LaunchAndWait(''+EXIFTOOL_PATH+''' -
DateTimeOriginal=''+FormattedNewFileDateTime+''' '
CreateDate=''+FormattedNewFileDateTime+'Z" '
MetadataDate=''+FormattedNewFileDateTime+'Z" '
ModifyDate=''+FormattedNewFileDateTime+''' '
'''+FinalDestinationFileName+''' -P -q -m -overwrite_original',SW_HIDE);
            Application.ProcessMessages;
            // File DateTime Corruption (CreationDate, LastModifiedDate,
LastAccessedDate)
            CorruptFileDateTime(FinalDestinationFileName,
FilesSelectedDate.DateTime);
        end;
    end;
    if (RBtn_OverwriteFiles.Checked) and (not CBox_CreateBackup.Checked)
then DeleteFile(BackupDestinationFileName);
    PBar_ProgressionStatus.Position := i+1;
END;

    if (RBtn_CreateZIPFile.Checked) and (not ABORT_REQUESTED) then begin

        if FileExists(ZIPFileName) and (not DeleteFile(ZIPFileName)) then begin
            ErrorsList.Add('Фатальна помилка : не вдалося створити ZIP архів
(''+ExtractFileName(ZIPFileName)+'')');
            ErrorsList.Add('Переконайтеся в тому, що шлях збереження фалів задано
вірно');
            ABORT_REQUESTED := false;
        end else begin

            ZipComponent := TZip.create(self);
            try
                try
                    ZipComponent.FileName := ZIPFileName;
                    if CBox_AddGlobalComment.Checked then ZipComponent.ZipComment :=
CommentText.Text
                else ZipComponent.ZipComment := '';
                ZipComponent.FileSpecList := ZipPendingFiles;
                ZipComponent.AddOptions := [aoUpdate];
                if (ZipComponent.Add <> ZipPendingFiles.Count) then begin
                    if ZipComponent.Cancelled then begin
                        ABORT_REQUESTED := true;
                        ErrorsList.Add(ZIPFileName+' : помилка створення ZIP архіву');
                    end else begin
                        ABORT_REQUESTED := false;
                        ErrorsList.Add('деякі файли не вдалося додати в ZIP
архів'''+ExtractFileName(ZIPFileName)+'''');
                    end;
                end;
            end;
        end;
    end;
end;

```

```

except
    ErrorsList.Add(('+ExtractFileName(ZIPFileName)+'));
    ErrorsList.Add();
    ABORT_REQUESTED := false;
end;
finally
    ZipComponent.Free;
end;
if CBox_ChangeDateData.Checked then CorruptFileDateTime(ZIPFileName,
FilesSelectedDate.DateTime);
end;
end;
PBar_ProgressionStatus.Position := PBar_ProgressionStatus.Max;
Label_ProgressionStatus.Caption := 'Відміна процесу стиснення користувачем';
Application.ProcessMessages;
ErrorForm.Caption := 'Користувач відмінив процес стиснення';
if RBtn_OverwriteFiles.Checked then ErrorForm.ErrorTitleMsg.Caption :=
'Деякі файли не були стиснуті '+ #13 +' вже оброблені файли не будуть
відновлені.'
else ErrorForm.ErrorTitleMsg.Caption := 'Деякі файли не були стислі '+ #13
+' вже оброблені файли не будуть видалені';
ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
ErrorForm.ShowModal;
IS_PROCESSING_IMAGES := false;
ABORT_REQUESTED := false;
ChangeControlsDisponibility(true);
if RBtn_CreateZIPFile.Checked then ZipPendingFiles.Free;
ErrorsList.Free;
Application.ProcessMessages;
DeleteDirectory(TEMP_PATH);
Application.ProcessMessages;
exit;
end else begin
    Btn_CloseApp.Caption := 'Скасування'#10#13'неможливе';
    Btn_CloseApp.Enabled := false;
end;
PBar_ProgressionStatus.Position := PBar_ProgressionStatus.Max;
if (ErrorsList.Text<>'') then begin
    Label_ProgressionStatus.Caption := 'Стиснення відбулося з помилками';
    Application.ProcessMessages;
    ErrorForm.Caption := 'Помилка при компресії файлів';
    ErrorForm.ErrorTitleMsg.Caption := 'Деякі файли не можуть бути стиснуті
правильно '+ #13 +' Перевірте формат і доступність:';
    ErrorForm.FilesConcernedMemo.Lines := ErrorsList;
    ErrorForm.ShowModal;
end else begin
    Label_ProgressionStatus.Caption := 'Стиснення успішно завершилося';
    Application.ProcessMessages;
    MessageBoxA(MainForm.Handle, Pchar('Всі файли було успішно
стиснуто'), Pchar('Стиснення заверрене'), MB_ICONINFORMATION + MB_SYSTEMMODAL +
MB_SETFOREGROUND + MB_TOPMOST);
end;
IS_PROCESSING_IMAGES := false;
Btn_CloseApp.Enabled := true;
ChangeControlsDisponibility(true);
Btn_CloseApp.Enabled := true;
if RBtn_CreateZIPFile.Checked then ZipPendingFiles.Free;
ErrorsList.Free;
Application.ProcessMessages;
DeleteDirectory(TEMP_PATH);
Application.ProcessMessages;
end;
procedure TMainForm.FormCreate(Sender: TObject);
var
    RES: TResourceStream;
begin
    DragAcceptFiles(FileList.Handle, true); // Допускаються Drag'n Drop процедури
    FilesSelectedDate.Date := now;
    FilesSelectedTime.Time := now;

```

```

APP_PATH :=
IncludeTrailingPathDelimiter(ExtractFilePath(Application.ExeName));
EXIFTOOL_PATH := APP_PATH+'exiftool.exe';
TEMP_PATH :=
IncludeTrailingPathDelimiter(ExtractFilePath(Application.ExeName)) + '$temp$\';
PROFILES_CFGFILE := APP_PATH+'CompressionProfiles.jcfg';

if not FileExists(EXIFTOOL_PATH) then begin
  try
    RES := TResourceStream.Create(0, 'EXIFTOOL', 'EXEFILE');
    try
      RES.SaveToFile(EXIFTOOL_PATH);
      Application.ProcessMessages;
    finally
      RES.Free;
    end;
  except
    MessageBox(0, Pchar('Виникла помилка при ініціалізації програми, яка
перешкоджає її виконанню. '+ #13 +' Файл "exiftool.exe" не було
знайдено. '), Pchar('Помилка'), MB_ICONSTOP + + MB_ICONINFORMATION +
MB_SYSTEMMODAL + MB_SETFOREGROUND + MB_TOPMOST);
    Application.Terminate;
  end;
end;
if not FileExists(APP_PATH+'ZipDll.dll') then begin
  try
    RES := TResourceStream.Create(0, 'ZIPDLL', RT_RCDATA);
    try
      RES.SaveToFile(APP_PATH+'ZipDll.dll');
      Application.ProcessMessages;
    finally
      RES.Free;
    end;
  except
    MessageBox(0, Pchar('Виникла помилка при ініціалізації програми, яка
перешкоджає її виконанню. '+ #13 +' Файл "ZipDll.dll" не було
знайдено. '), Pchar('Помилка'), MB_ICONSTOP + + MB_ICONINFORMATION + MB_SYSTEMMODAL
+ MB_SETFOREGROUND + MB_TOPMOST);
    Application.Terminate;
  end;
end;

LoadProfiles();

end;
// ----- //
procedure TMainForm.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
  if IS_PROCESSING_IMAGES then CanClose:=false;
end;
// ----- //
procedure TMainForm.Timer1Timer(Sender: TObject);
begin
  //DisplayForm.CompressionRate
end;
initialization
  Randomize;
  JPEGImage1 := TJpegImage.Create;
  BMPTempImage1 := TBitmap.Create;
  BMPTempImage2 := TBitmap.Create;
// ----- //
finalization
  BMPTempImage2.Free;
  BMPTempImage1.Free;
  JPEGImage1.Free;
  Application.ProcessMessages;
  DeleteDirectory(TEMP_PATH);
end.

```

about.pas - файл довідки

```
unit about;

interface

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

type
  TForm5 = class(TForm)
    Mem1: TMemo;
    Button1: TButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

{$R *.dfm}

procedure TForm5.FormCreate(Sender: TObject);
begin
  Mem1.Clear;
  Mem1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Mem1.Lines.Add('');
  Mem1.Lines.Add(' на тему:');
  Mem1.Lines.Add('');
  Mem1.Lines.Add(' Дослідження та програмна реалізація інтелектуальної системи  
обробки та конвертації зображень');
  Mem1.Lines.Add('');
  Mem1.Lines.Add(' Керівник: Минайленко Р.М. ');
  Mem1.Lines.Add('');
  Mem1.Lines.Add(' Розробив: студент Лисогор Владислав Вікторович');
  Mem1.Lines.Add(' гр. КІ-22М-1');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('Кропивницький 2023');
  Mem1.Lines.Add('');
end;

procedure TForm5.Button1Click(Sender: TObject);
begin
  Form5.Close;
end;
end.
```