

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 перетворення
мультимедіа даних із збереженням логічно-структурного змісту
даних”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-2
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Шевченко Д.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.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 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Шевченку Денису Валерійовичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 33-13 від 04.08.2023 року | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <i>10.12.2023 р.</i> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <i>Метою розробки є дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних</i> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Економічна ефективність розробленої програми.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | <i>5. Впровадження системи в промислову експлуатацію</i> | |
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | | | | | | | | | | |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | | | | | | | | | | |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | | | | | | | | | | |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | | | | | | | | | | |
| <i>5. Впровадження системи в промислову експлуатацію</i> | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <i>Наукова новизна</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> | | | | | | | | | | |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> | | | | | | | | | | |

6. Консультанти розділів роботи

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

7. Дата видачі завдання « 6 » вересня 2023 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Шевченко Д.В. Дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Метою розробки є дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Об'єктом дослідження є процес перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Предметом дослідження є методи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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

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

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4.

Ключові слова: комп'ютерні науки, перетворення мультимедіа даних

ABSTRACT

Shevchenko D.V. Research and software implementation of a multimedia data conversion system with preservation of the logical and structural content of the data. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for the multimedia data conversion system with preservation of the logical and structural content of the data.

The purpose of the development is the research and software implementation of the multimedia data conversion system while preserving the logical and structural content of the data.

The object of research is the process of transforming multimedia data while preserving the logical and structural content of the data.

The subject of research is methods of multimedia data transformation while preserving the logical and structural content of the data.

Research methods are based on methods of information theory and coding, methods of mathematical statistics, software development methods.

The result of the work is the software implementation of the multimedia data conversion system with preservation of the logical and structural content of the data.

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

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

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

The program was developed in the Delphi 10.4 environment.

Keywords: computer science, multimedia data transformation

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	4
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	27
2.3 Розгорнута постановка завдання	33
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	35
3.1 Опис функціонування системи	35
3.2 Розробка структурної схеми.....	39
3.3 Розробка функціональної схеми	46
3.4 Розробка діаграми процесів.....	65
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	67
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	67
4.2 Захист розробленого програмного забезпечення.....	77
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	79
6 НАУКОВА НОВИЗНА	82

						ВКРМ-122.23.0050.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних	Літ.	Аркуш	Аркушів
Розроб.	Шевченко Д.В.					М	1	121
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22М-2		
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	83
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	83
7.2 Розрахунок трудомісткості розробки програмної продукції.....	85
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	87
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	92
7.5 Визначення собівартості розробки та ціни програмної продукції.....	96
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	99
7.7 Визначення експлуатаційних витрат.....	100
7.8 Визначення економічної ефективності програмної продукції.....	101
7.9 Висновок.....	103
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	104
8.1 Вступ.....	104
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	105
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	106
8.4 Розробка заходів з умов поліпшення охорони праці.....	109
8.5 Розрахункова частина	110
9 ОСНОВНІ ВИСНОВКИ.....	112
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	114

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

КБПЗ – 2023

ВСТУП

Актуальність теми. Програми для перегляду й простої обробки зображень часто поставляються разом із цифровими фотоапаратами й сканерами. Більш складні й потужні програми (Adobe Photoshop, Corel PHOTO-PAINT, Paint Shop Pro, Microsoft Picture It!, Visualizer Photo Studio, Pixel image editor, PixBuilder Photo Editor, Fo2Pix ArtMaster та ін.) потрібно діставати окремо, і як правило за гроші. Виключення становить GIMP, безкоштовно розповсюджувана програма, можливості якої порівнянні з можливостями Adobe Photoshop. Сучасні редактори не позбавлені недоліків, однак грамотне їхнє використання дозволяє вирішити більшість завдань, що виникають при редагуванні зображень. Вони дозволяють, у якимсь ступені, виправляти технічні дефекти, допущені при проведенні фотозйомки. Конвертери можна підрозділити на дві групи:

– Класичні, засновані на принципі рендерінгу (всі налаштування застосовуються до зменшеної копії зображення – preview, після чого програма конвертує вихідний файл).

– Працюючі із зображеннями в реальному часі (Adobe Lightroom, Apple Aperture).

В обох підходів є свої плюси. У першому випадку мінімальне навантаження на ресурси комп'ютера, що дозволяє з комфортом обробляти більші файли. Такий метод звичайно використовують конвертери, здатні працювати зі студійними цифровими задниками, наприклад Capture One. Другий підхід з'явився порівняно недавно й більше ефективний, якщо у вашому розпорядженні є потужна робоча станція й розмір файлів не перевищує 100 Мб.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.
- Дослідження системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.
- Програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Об'єктом дослідження є процес перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Предметом дослідження є методи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.
- Розроблено вітчизняний продукт перетворення мультимедіа даних із збереженням логічно-структурного змісту даних, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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

					ВКРМ-122.23.0050.00.00.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 перетворення мультимедіа даних із збереженням логічно-структурного змісту даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					VKPM-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Програмне забезпечення, яке розробляється у даному магістерському проекті призначено для системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних. Пакетна обробка файлів незамінна при роботі з великою кількістю зображень, що вимагають однотипного редагування. Практично будь-яку рутинну операцію, пов'язану з обробкою (англ. batch mode), у такий спосіб можна виконати набагато швидше.

Для графічних форматів можливі різні варіанти перетворень: від зміни розміру й формату зображення, додавання водяного знака й тексту до накладення ефектів і фільтрів. Примітно, що в багато програм для перегляду зображень уже убудовані подібні функції. Крім того, як альтернатива існують окремі програми для обробки – конвертери.

Важливий і сам метод роботи з даними. У моді так звана недеструктивна обробка. При відкритті файлу програма автоматично створює текстовий документ (звичайно – файл із розширенням ХМР), де записує всю інформацію про хід роботи з ісходником. Через час можна повернутися до параметрів конвертування, а не робити обробку із самого початку або працювати з експортованими у форматах TIFF або JPEG даними. Це дозволяє одержати кращу якість на виході.

Більшість розходжень у результатах конвертування пов'язане з установками за замовчуванням, від яких відштовхується користувач. Розходження між програмами одного покоління, як правило, ставляться до передачі кольору – вони суб'єктивні. Конвертери різних поколінь можуть розрізнятися й на рівні технологій: старі програми гірше працюють із деталями, тому що алгоритми обробки постійно вдосконалюються.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Областю застосування є перетворення або конвертація мультимедіа даних. При заливанні зображень на хост, найчастіше з'являється необхідність конвертації зображення в строго певний тип, наприклад в jpg або png. Конвертери можуть конвертувати зображення наступних форматів:

1. Hasselblad CFV/H3D39II (.3FR).
2. PFS: 1st Publisher (.ART).
3. Sony Digital Camera Alpha Raw Image Format (.ARW).
4. Microsoft Audio/Visual Interleaved (.AVI).
5. AVS X image (.AVS).
6. Microsoft Windows bitmap (.BMP).
7. Computer Graphics Format (.CGM).
8. Kodak Cineon Image Format (.CIN).
9. Raw cyan, magenta, yellow, and black samples (.CMYK).
10. Raw cyan, magenta, yellow, black, and alpha samples (.CMYKA).
11. Canon Digital Camera Raw Image Format (.CR2).
12. Canon Digital Camera Raw Image Format (.CRW).
13. Microsoft Cursor Icon (.CUR).
14. DR Halo (.CUT).
15. Digital Imaging and Communications in Medicine (DICOM) image (.DCM).
16. Kodak Digital Camera Raw Image File (.DCR).
17. ZSoft IBM PC multi-page Paintbrush image (.DCX).
18. Microsoft DirectDraw Surface (.DDS).
19. Multi-face font package (Freetype 2.3.5) (.DFONT).
20. Digital Negative (.DNG).
21. SMPTE Digital Moving Picture Exchange 2.0 (SMPTE 268 M-2003) (.DPX).

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

22. Encapsulated Portable Document Format (.EPDF).
23. Adobe Encapsulated PostScript Interchange format (.EPI).
24. Adobe Encapsulated PostScript (.EPS).
25. Adobe Encapsulated PostScript (.EPSF).
26. Adobe Encapsulated PostScript Interchange format (.EPSI).
27. Adobe Encapsulated PostScript Interchange format with TIFF preview (.EPT).
28. Encapsulated PostScript Level II with TIFF preview (.EPT2).
29. Encapsulated PostScript Level III with TIFF preview (.EPT3).
30. Epson RAW Format (.ERF).
31. High dynamic-range (HDR) file format developed by Industrial Light & Magic (.EXR).
32. Group 3 TIFF (.FAX).
33. Flexible Image Transport System (.FITS).
34. FlashPix Format (.FPX).
35. Plasma fractal image (.FRACTAL).
36. Flexible Image Transport System (.FTS).
37. Group 3 FAX (.G3).
38. CompuServe Graphics Interchange Format (.GIF).
39. CompuServe graphics interchange format (version 87a) (.GIF87).
40. Raw gray samples (.GRAY).
41. Raw green, red, and blue samples (.GRB).
42. High dynamic-range (HDR) file format developed by Industrial Light & Magic (.HDR).
43. Slow Scane TeleVision (.HRZ).
44. Truevision Targa image (.ICB).
45. Microsoft icon (.ICO).
46. Microsoft icon (.ICON).
47. IPL Image Sequence (.IPL).

48. Joint Bi-level Image experts Group interchange format (2.0) (.JBG).
49. Joint Bi-level Image experts Group interchange format (2.0) (.JBIG).
50. Multiple-image Network Graphics (.JNG).
51. JPEG-2000 JP2 File Format Syntax (.JP2).
52. JPEG-2000 Code Stream Syntax (.JPC).
53. JPEG/JIFF Image (.JPE).
54. Joint Photographic Experts Group JFIF format (.JPEG).
55. Joint Photographic Experts Group JFIF format (.JPG).
56. JPEG-2000 File Format Syntax (.JPX).
57. Kodak Digital Camera Raw Image Format (.K25).
58. Kodak Digital Camera Raw Image Format (.KDC).
59. Motion Picture Experts Group file interchange format (version 2)
(.M2V).
60. Raw MPEG-4 Video (.M4V).
61. MATLAB image format (.MAT).
62. Magick image file format (.MIFF).
63. Multiple-image Network Graphics (.MNG).
64. Bi-level bitmap in least-significant-byte first order (.MONO).
65. QuickTime Movie (.MOV).
66. MPEG-4 Video Stream (.MP4).
67. Magick Persistent Cache image file format (.MPC).
68. Motion Picture Experts Group file interchange format (version 1)
(.MPEG).
69. MPEG Video Stream (.MPG).
70. Sony (Minolta) Raw Image File (.MRW).
71. Magick Scripting Language (.MSL).
72. ImageMagick's own SVG internal renderer (.MSVG).
73. MTV Raytracing image format (.MTV).
74. Magick Vector Graphics. (.MVG).

75. Nikon Digital SLR Camera Raw Image File (.NEF).
76. Nikon Digital SLR Camera Raw Image File (.NRW).
77. Olympus Digital Camera Raw Image File (.ORF).
78. On-the-air Bitmap (.OTB).
79. Open Type font (Freetype 2.3.7) (.OTF).
80. 16bit/pixel interleaved YUV (.PAL).
81. Palm pixmap (.PALM).
82. Common 2-dimensional bitmap format (.PAM).
83. Portable bitmap format (black and white) (.PBM).
84. Photo CD (.PCD).
85. Photo CD (.PCDS).
86. HP Page Control Language (.PCL).
87. Apple Macintosh QuickDraw/PICT (.PCT).
88. ZSoft IBM PC Paintbrush file (.PCX).
89. Palm Database ImageViewer Format (.PDB).
90. Portable Document Format (.PDF).
91. Portable Document Archive Format (.PDFA).
92. Pentax Electronic File (.PEF).
93. Embrid Embroidery Format (.PES).
94. Postscript Type 1 font (ASCII) (.PFA).
95. Postscript Type 1 font (binary) (.PFB).
96. Portable float map format (.PFM).
97. Portable graymap format (gray scale) (.PGM).
98. Personal Icon (.PICON).
99. Apple Macintosh QuickDraw/PICT file (.PICT).
100. Alias/Wavefront RLE image format (.PIX).
101. Joint Photographic Experts Group JFIF format (62) (.PJPEG).
102. Plasma fractal image (.PLASMA).
103. Portable Network Graphics (.PNG).

104. Portable Network Graphics (.PNG24).
105. Portable Network Graphics (.PNG32).
106. Portable Network Graphics (.PNG8).
107. Portable anymap (.PNM).
108. Portable pixmap format (color) (.PPM).
109. Adobe PostScript file (.PS).
110. Adobe Photoshop bitmap file (.PSD).
111. Pyramid encoded TIFF (.PTIF).
112. Seattle File Works multi-image file (.PWP).
113. Fuji CCD-RAW Graphic File (.RAF).
114. SUN Rasterfile (.RAS).
115. Raw red, green, and blue samples (.RGB).
116. Raw red, green, blue, and alpha samples (.RGBA).
117. Alias/Wavefront image file (.RLA).
118. Utah Run length encoded image file (.RLE).
119. Scitex Continuous Tone Picture (.SCT).
120. Seattle File Works image (.SFW).
121. Irix RGB image (.SGI).
122. Sony Raw Format 2 (.SR2).
123. Sony Raw Format (.SRF).
124. SUN Rasterfile (.SUN).
125. Truevision Targa image (.TGA).
126. Tagged Image File Format (.TIF).
127. Tagged Image File Format (.TIFF).
128. PSX TIM file (.TIM).
129. TrueType font collection (Freetype 2.3.7) (.TTC).
130. TrueType font file (.TTF).
131. Raw text file (.TXT).
132. Truevision Targa image (.VDA).

133. VICAR rasterfile format (.VICAR).
134. Visual Image Directory (.VID).
135. Khoros Visualization Image File Format (.VIFF).
136. Truevision Targa image (.VST).
137. Wireless bitmap (.WBMP).
138. Google Web (.WEBP).
139. Windows Meta File (.WMF).
140. Compressed Windows Meta File (.WMZ).
141. X Image (.X).
142. Sigma Camera RAW Picture File (.X3F).
143. X Windows system bitmap, black and white only (.XBM).
144. Constant image uniform color (.XC).
145. GIMP image (.XCF).
146. X Windows system pixmap (.XPM).
147. Microsoft XML Paper Specification (.XPS).
148. Khoros Visualization image (.XV).
149. X Windows system window dump (.XWD).
150. Raw Y, Cb, and Cr samples (.YCBCR).
151. Raw Y, Cb, Cr, and alpha samples (.YCBCRA).
152. CCIR 601 4:1:1 (.YUV).

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

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

Image Tuner

Проста програма, що працює за принципом «одного вікна» і не занадто ускладнена налаштуваннями. У лівій частині вікна додаються файли для обробки, праворуч – вказуються параметри конвертації. Доступний передогляд по клічці на ескіз зображення.

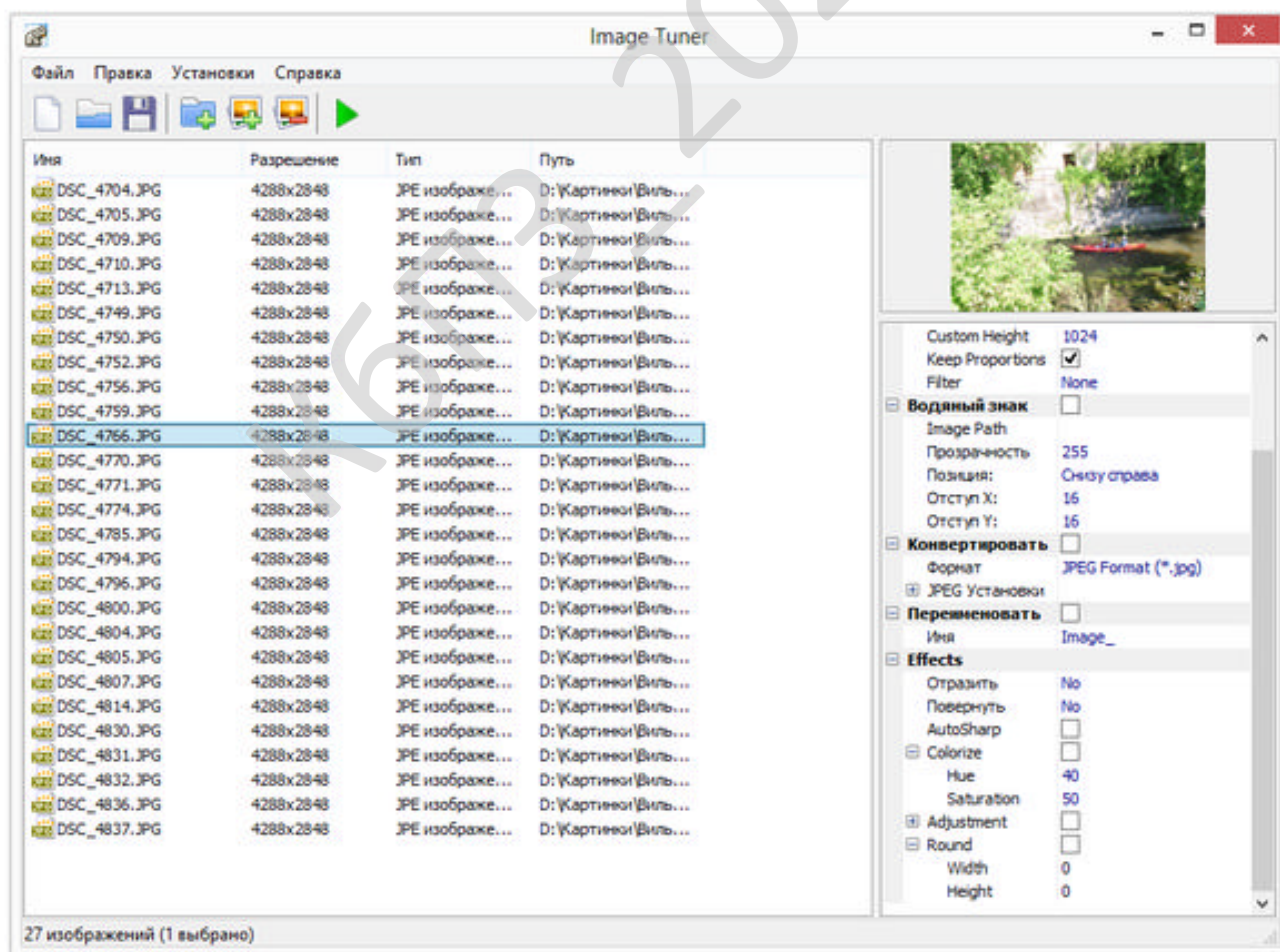


Рисунок 2.1 – Интерфейс Image Tuner

Серед можливостей перетворення – зміна відтінку, розміру, орієнтації зображення, додавання водяного знака. У наявності є фільтри, але вони настільки прості, що навіть не мають налаштувань і чомусь заховані в секції «Змінити розмір». Половина налаштувань – на англійській, що залишилася частина переведена на російську.

Кількість форматів для збереження невелике – усього 5. На вході Image Tuner підтримує такі популярні формати, як JPEG, BMP, PNG, GIF, TIFF, а також RAW, NEF і інші.

Таким чином, дуже простий конвертер для самої базової обробки зображень.

IrfanView

Переглядач IrfanView, при своїй безкоштовності й невеликому розмірі, підтримує значну кількість форматів і містить у собі функції конвертера, доступні через меню «Batch Conversion/Rename...». На вибір – один із трьох режимів: пакетне перейменування, перетворення й змішаний.

Для збереження доступно близько 20 форматів (меню «Output format»), при цьому параметри доступні не для кожного з них.

Вибір інших перетворень можливий тільки при активації «Use advanced options...», по натисканню кнопки «Advanced». Налаштування дозволяють змінити розміри, кадрувати зображення, відбити по горизонталі або вертикалі, додати водяний знак – стандартний набір для переглядача графіки. Однак на практиці стає зрозуміло, що «просунуті» опції зовсім не інтуїтивні: вся справа в тому, що вікна передогляду не передбачається (передогляд у головному вікні ставиться до вихідного зображення). Звичайно, кадрувати фото або змінити його розмір можна попільсьельно. А як обійтися з фільтрами налаштування яскравості, балансу й інших колірних параметрів, коли потрібно вказувати цифри? Виходячи з вищесказаного, після застосування перетворень і виходу з «Advanced» зміни можна буде побачити тільки по закінченню конвертації. А от для пакетного перейменування файлів доступний тестовий режим.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

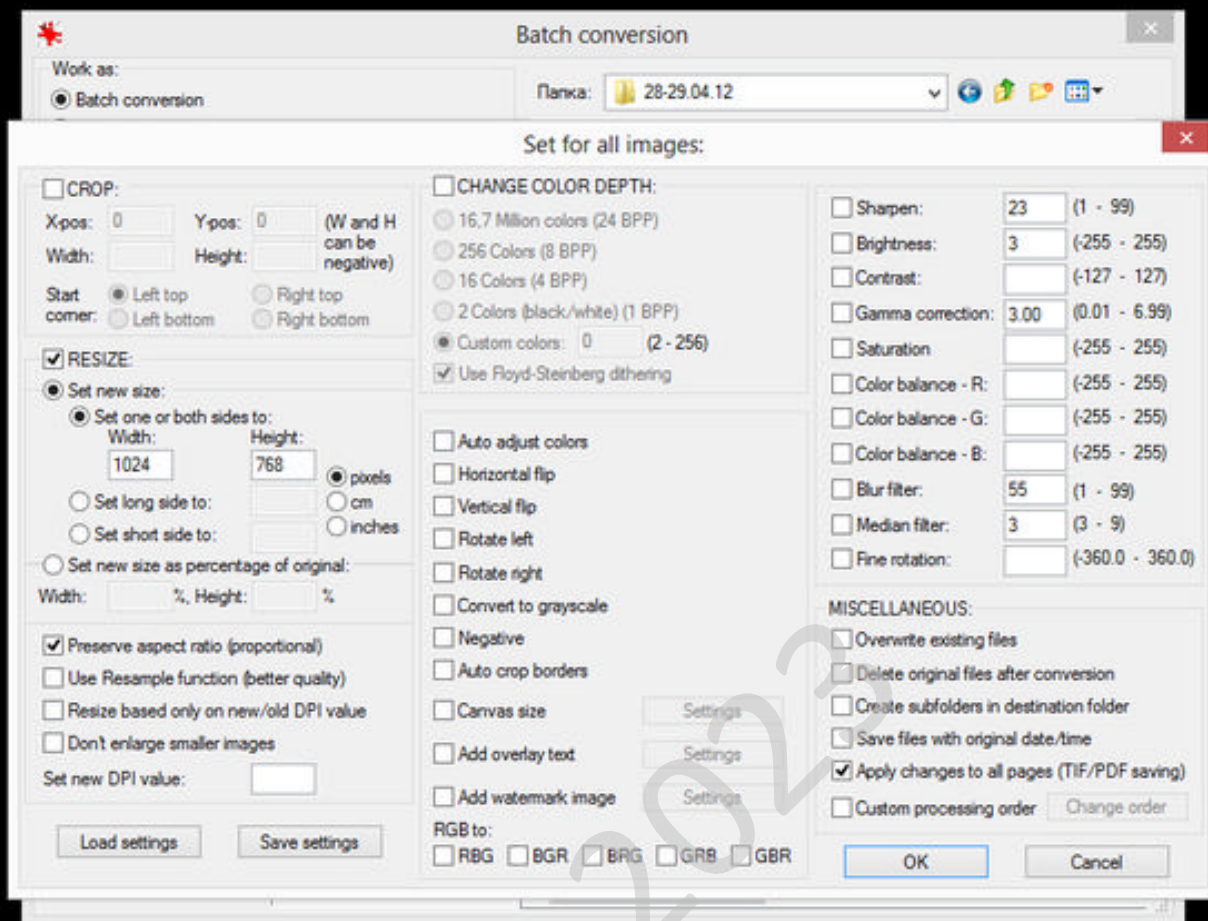


Рисунок 2.2 – Налаштування перетворення в IrfanView

Таким чином, назвати IrfanView зручним конвертером зображень не можна. Його можна використовувати лише для базових перетворень, що не вимагають передогляду: зміна орієнтації зображення, розміру й формату збереження.

AVS Image Converter

AVS Image Converter є частиною набору програм для роботи із зображеннями AVS4You. Видимо, тому в дистрибутив розміром в 27 МБ (що не так і мало для конвертера), входять допоміжні утиліти Software Navigator і Update Manager.

Підтримуваних форматів зображень для запису – 8, для читання – ледве більше 20. Для обробки можна імпортувати фото з аккаунтів Flickr або Facebook (після попередньої авторизації).

Графічний інтерфейс дружелюбно розташований до недосвідченого користувача: деякі функції спрощені або замінені передумовками. При зміні розміру зображення доступні передналаштування для різних моніторів і стандартних пропорцій фотознімків, що особливо корисно при публікації в Інтернеті.

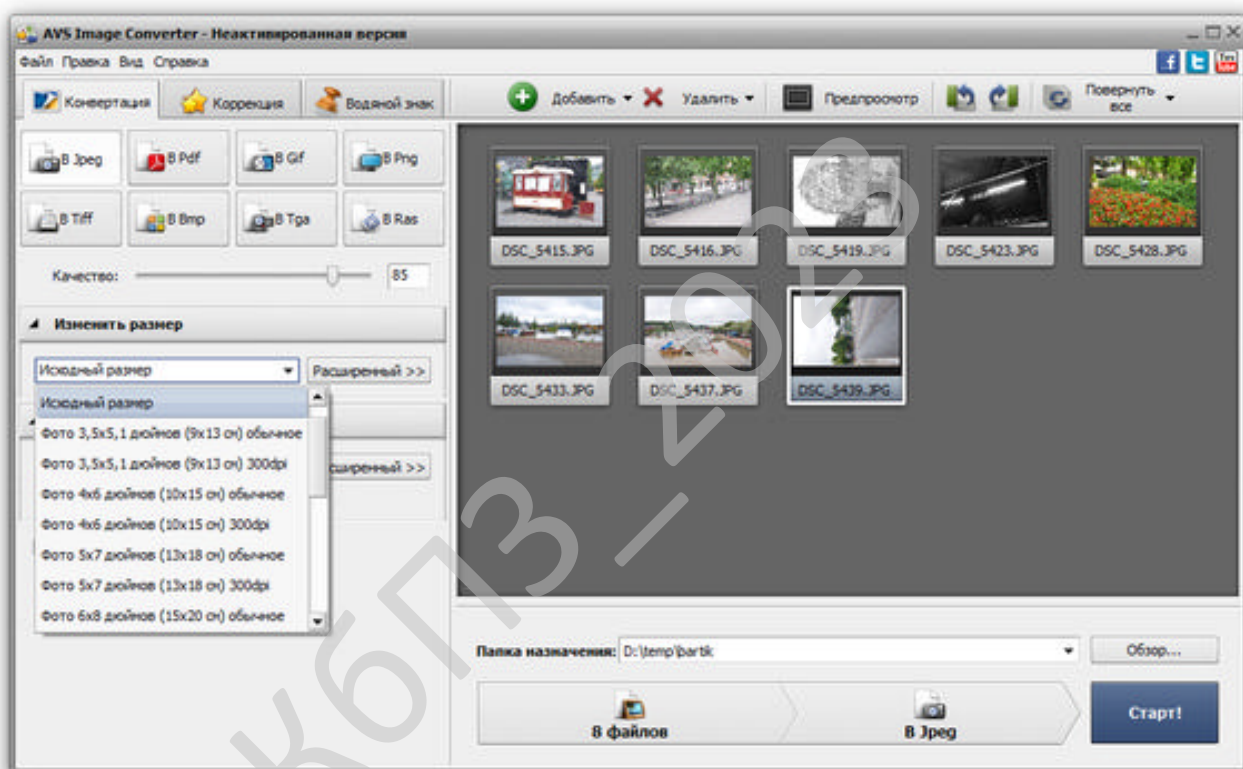


Рисунок 2.3 – Вибір пресету в AVS Image Converter

Налаштувань перетворення (вкладка «Корекція») небагато: зміна яскравості, контрасту, колірної температури; ефекти розмиття/різкості, додавання текстури. Крайня вкладка – «Водяной знак», що дозволяє накласти зображення або текст на фото.

Інтерфейс розділений на дві частини – що не дуже раціонально, оскільки область для вибору файлів займає більшу частину робочого простору.

Додавши зображення в чергу конвертації, можна вказати формат вихідного зображення («Output Format») і папку збереження («Output Folder»).

Для доступу до інструментів перетворень активується опція «Use Advanced Options (Resize...)». Набір перетворень суцільно базовий: зміна розмірів, перспективи, додавання тексту, водяного зображення, рамки. Налаштування можна зберегти або завантажити з файлу конфігурації, так само як у більшості програм, що обдивляються. Вікна передогляду немає.

В окрему вкладку «Batch Rename» винесений інструмент для пакетного перейменування файлів. Крім стандартної зміни ім'я по масці, цікава функція «Search and Replace», з її допомогою легко підкоректувати назву файлів без повної зміни ім'я.

XnConvert

XnConvert – «відбруньковування» від популярного переглядача зображень XnView. По суті, всі ті ж самі дії, вироблені в середовищі XnConvert, можна зробити й в XNView (про що нижче), однак автономний конвертер має ряд відмінностей. Насамперед, він не призначений для перегляду. З одного боку, нічого не відволікає від функціонала. З іншого боку – якщо потрібно довідатися саму базову інформацію про вихідний файл, доводиться звертатися до іншої програмі-переглядачеві.

Спочатку роботи із програмою потрібно вказати файли для обробки (підтримується перетаскування, можна також вказати за допомогою кнопок). Варто звернути увагу, що зверху перебувають стовпці для сортування. При клічі але кожному з них у контекстному меню можна вибрати з безлічі інших варіантів – список величезний. Втім, було б більш зручно, якби файли відображалися у вигляді таблиці, а не ескізів – тоді було б зручно переглядати інформацію про вихідних файлів, а так її попросту ні, і сортування носить трохи «умовний» характер – незрозуміло, для чого вона.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

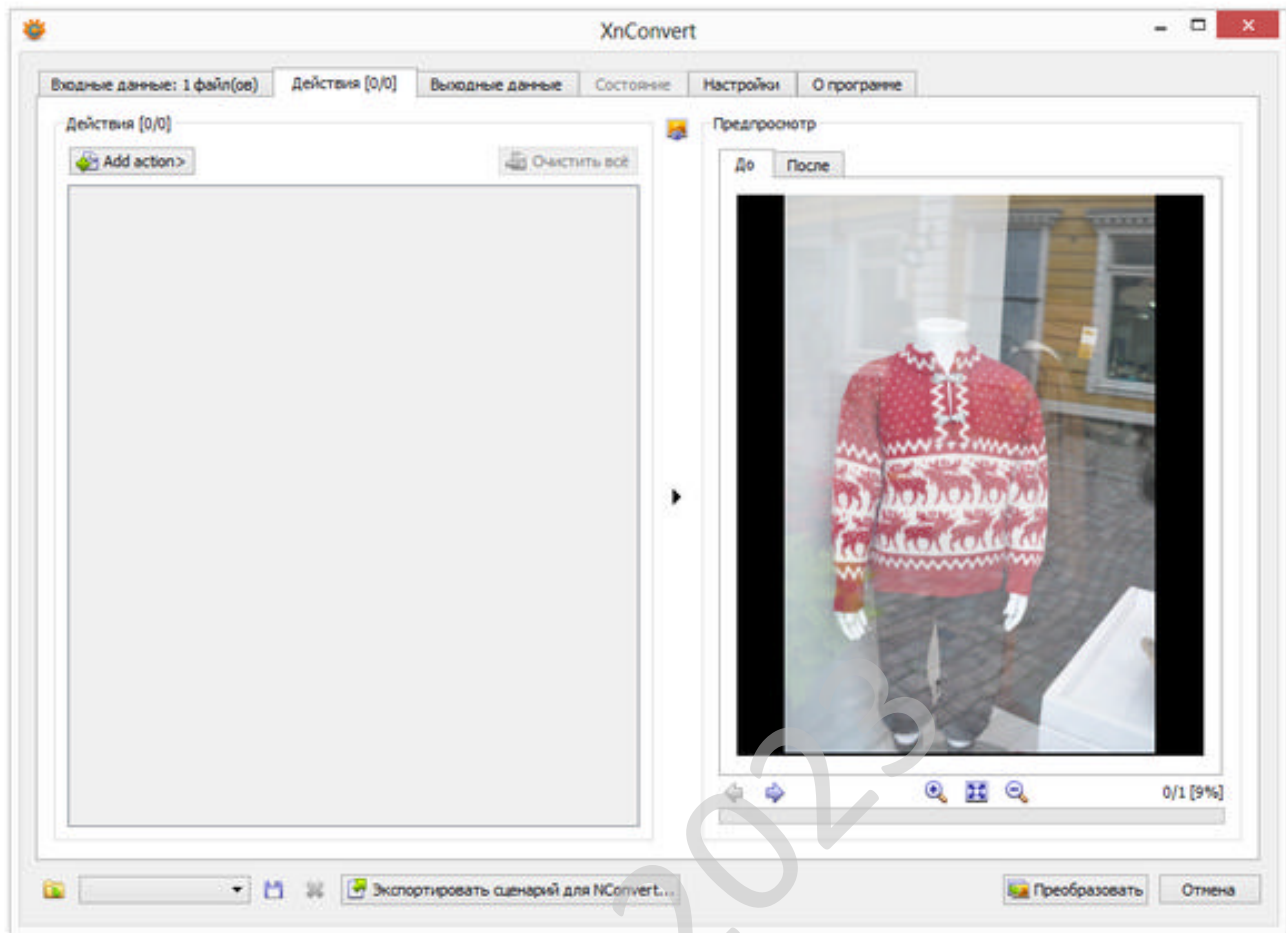


Рисунок 2.5 – Пакетна обробка в XnConvert

Основна вкладка, де зібрана список перетворень – «Дії». Для безкоштовної програми кількість налаштувань приємно дивує. Діляться перетворення на 4 групи:

- Зображення – по більшій частині, робота із трансформацією й властивостями файлу.
- Корекція – робота з рівнями, кольором.
- Фільтр – застосування ефектів розмиття, посилення різкості, зміни фокуса.
- Різне – по більшій частині, всі ті ж ефекти.

У вкладці вихідні дані вказуються опції збереження оброблених файлів: маска найменування, розташування й графічний формат. Для популярних форматів (JPG, GIF, PNG і т.п.) їсти можливість указати додаткові параметри

збереження. Є малопримітна, але досить корисна опція по експорті – можна завантажити оброблені фото на Picasa– або Flickr-акаунт, упакувати або відправити по FTP або на email.

Імпорт/експорт сценаріїв (налаштувань перетворень) доступні в нижній частині вікна, кнопка «Завантажити сценарій».

Для читання також доступно величезна кількість форматів – більше 500 (з тих, які підтримуються переглядачем XNView), з тим нюансом, що специфічні вимагають установки GhostScript або плагіна CAD.

XnView

Коротенько – про можливості пакетної обробки фото в XnView. Вікно налаштувань можна викликати через меню «Інструменти – Пакетна обробка...»...

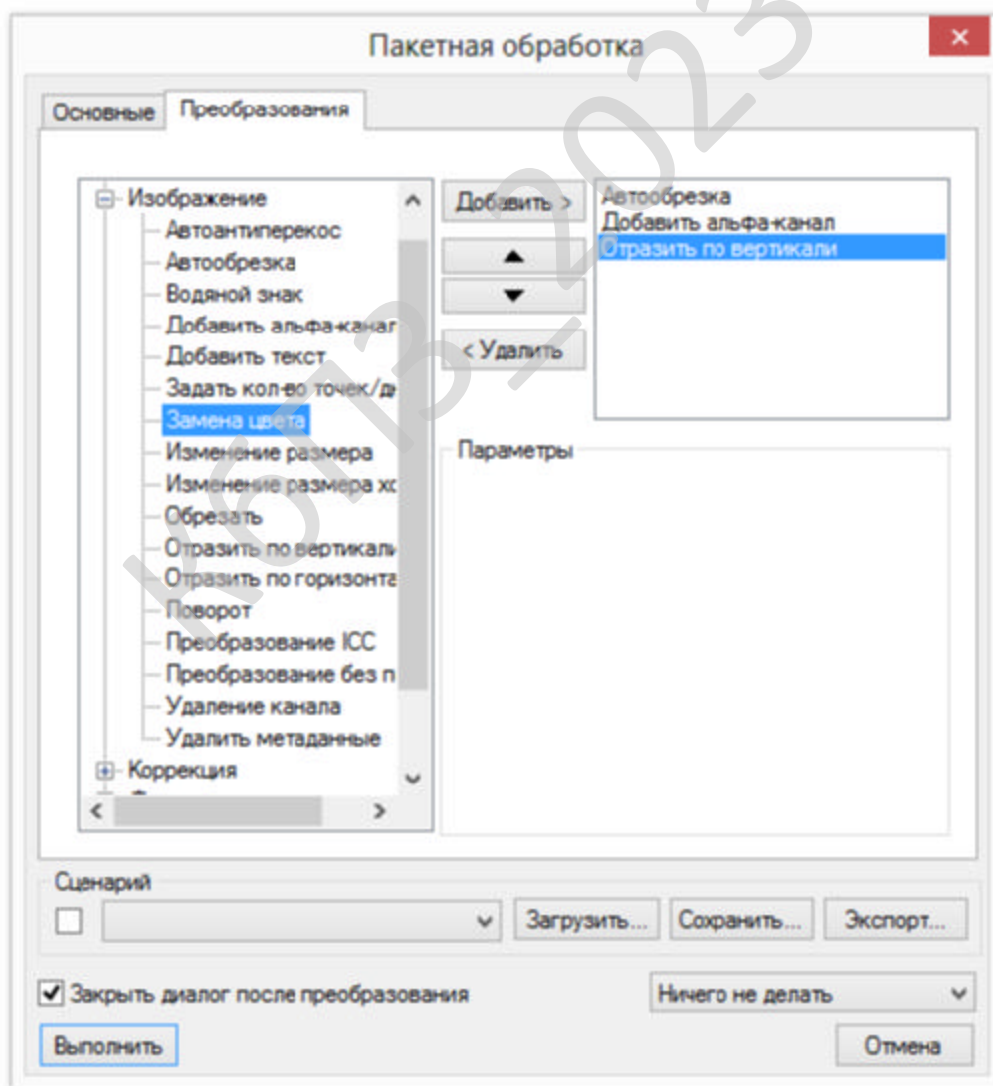


Рисунок 2.6 – Пакетна обробка в XnView

Насамперед, інтерфейс відрізняється від XNConvert. Складається всього із двох вкладок, у першій вказується формат і параметри збереження. У другій вкладці – список перетворень, які, для активації й налаштування, потрібно додати в праву частину вікна. Можливість збереження сценаріїв також передбачена, але формат не сполучимо з XnConvert: тут XBS, в XNConvert – BAT.

Фотоконвертер

Конвертація в «Фотоконвертері» примітна зручним покроковим майстром налаштування. На першому етапі потрібно вибрати зображення для конвертації й додати в праву частину вікна. На другому кроці – вибір перетворень, є вікно передогляду. Для виходу в меню вибору на панелі редагування потрібно натиснути кнопку «Додати дія». Серед функцій редагування, доступних у мінімальній версії конвертера, можна відзначити ефекти розмиття/різкості, усунення ефекту червоних очей. Водяний знак – тільки в стандартній версії. Професійна редакція «Фотоконвертера» дозволяє підключати ефекти Adobe Photoshop. Зібрано перетворення більш логічно, ніж у вищезгаданих програмах (приміром, XnConvert), хоча й тут не все ідеально. Наприклад, у групі налаштувань «Автоматичні» – корекція рівнів і кадрування, хоча ці операції логічніше зарахувати до редагування. Фільтри, такі як розмиття, у подібних конвертерах відносилися до ефектів. Однак звикнути до такого суб'єктивного сортування неважко. Третій і останній крок налаштування – вибір формату збереження. Якщо натиснути на стрілки вниз, з'являться додаткові опції, зокрема, перейменування файлів по масці. Після цього можна натискати кнопку «Старт».

Довідка чомусь доступна тільки англійською мовою, на сайті доступний список питань і відповідей і опис можливостей на російській. Усього на вибір надають три редакції програми – професійна, стандартна й домашня. Професійний «Фотоконвертер» підтримує більше 400+ графічних форматів, функціонує з командного рядка. Базовий містить стандартні функції й розпізнає тільки самі затребувані формати зображень (JPEG, TIFF, GIF, PNG, BMP), за винятком, ні багато ні мало, 380+ що залишилися.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

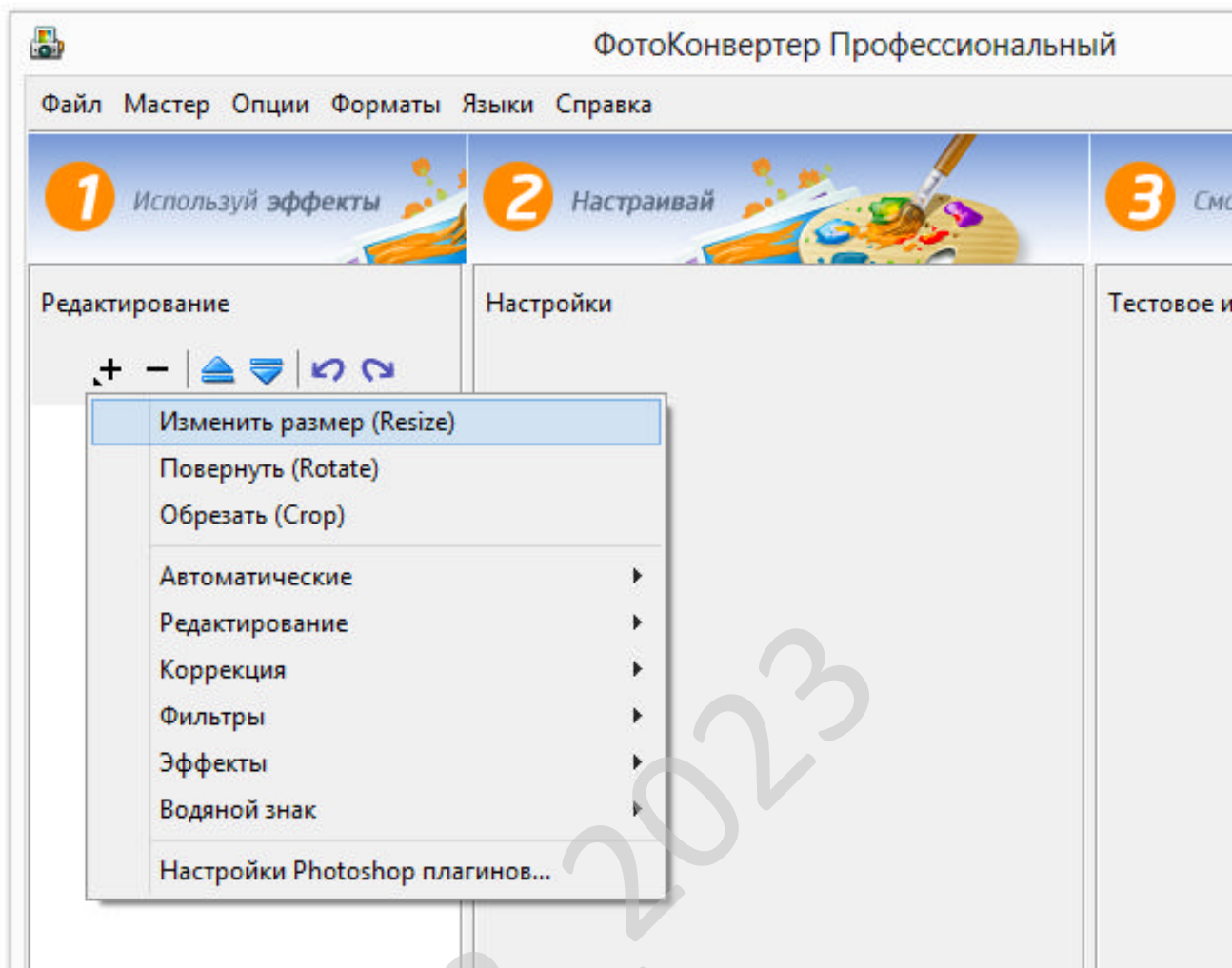


Рисунок 2.7 – Налаштування перетворень в «Фотоконвертері»

Adobe Photoshop

Безумовно, у графічний пакет Adobe Photoshop убудовані необхідні інструменти по пакетному перетворенню фото. Актуальна на даний момент версія – CS6, в огляді розглядається CS5. Обробку можна робити за допомогою екшнів (англ. «actions» – дії), операції Batch («File – Automate – Batch...») або скрипта Image Processor («File – Scripts – Image Processor...»).

Перший спосіб

Спочатку, через палітру Actions записується дія, що буде застосовуватися до кожного зображення. Найкраще для цієї мети взяти «тестовий зразок». На даному кроці список можливих перетворень обмежений лише засобами

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- Image Sizing – розміри й розв’язна здатність зображення на виході.
- Output Sharpening – посилення різкості.
- Metadata – зберігати метадані чи файлу ні.
- Watermarking – накладення водяного знака.

У лівій бічній панелі («Preset») доступні передумовки, є можливість додавання власних.

AhaView

Зручний переглядач і конвертер цифрових зображень.

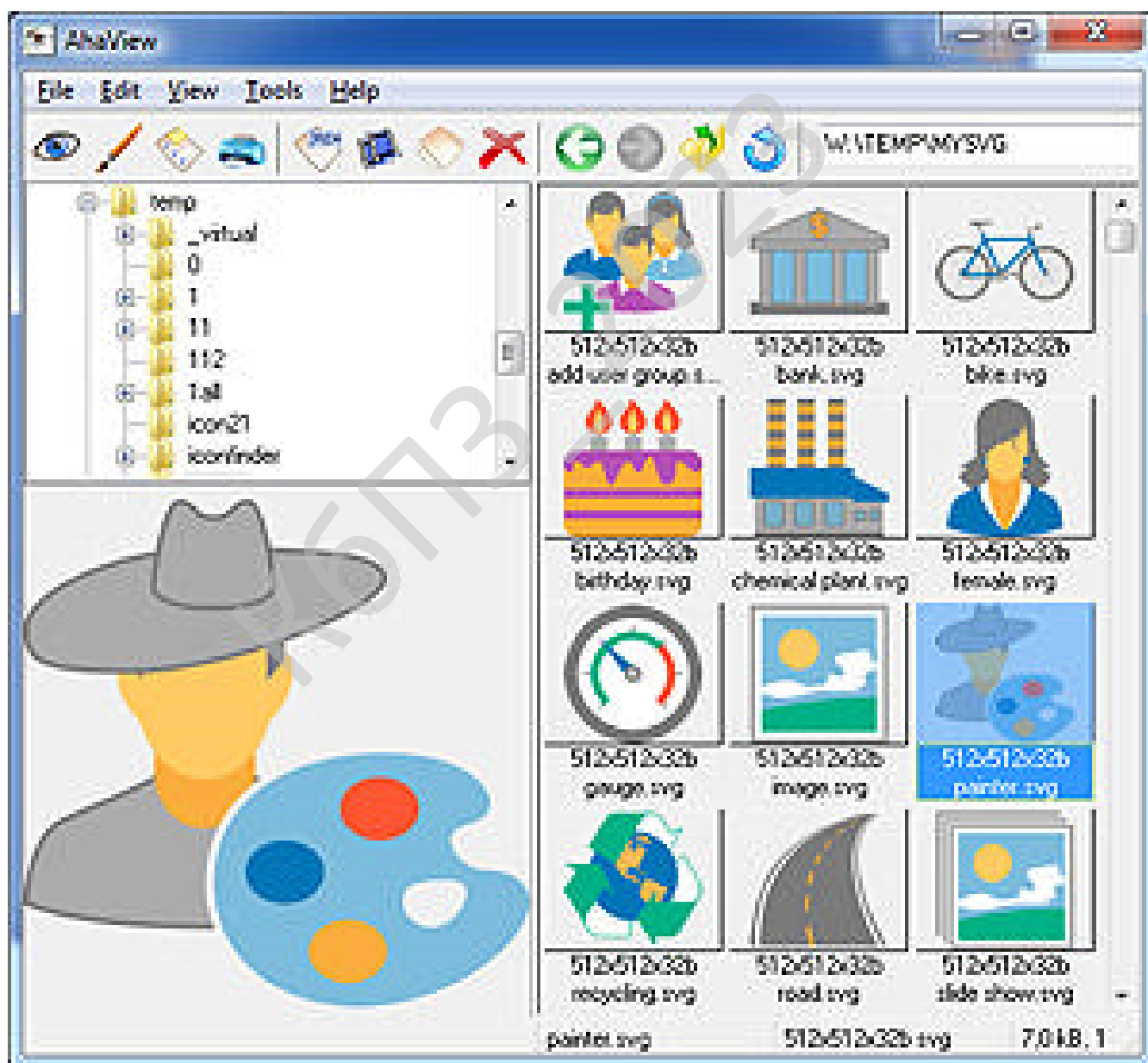


Рисунок 2.10 – Інтерфейс користувача AhaView

З AhaView ви можете:

- Переглядати каталоги із зображеннями.
- Перегортати картинки в повноекраному режимі.
- Конвертувати графічні файли у формати BMP, PNG, GIF, ICO, TIFF, TGA, XPM, і JPEG.
- Конвертувати й переглядати SVG, AI, PDF, ICNS файли.
- Створювати іконки з фотографій.
- Привласнювати опис файлам.
- Запускати слайд-шоу.
- Досліджувати властивості зображень.
- Копіювати рисунки в буфер.
- Робити операції з файлами.
- Використовувати інтерфейс командного рядка.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

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 підтримуватиме довільні ініціалізацію, фіналізацію й операції копіювання.

- Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

- Відладник Win 64 (на LLDB) і збирач для C++.

- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

- Підтримка Metal Driver GPU для macOS і iOS.

- Вбудований Fmxlinux.

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

- Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

- Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проєктах. 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 швидковістю. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCL, 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 Snake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані 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 стилізуємий 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.
- 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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

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 перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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

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

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

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

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

д) розробити рекомендації по організаційних та методичних заходах, які

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ - 2023

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Обробка зображень – будь-яка форма обробки інформації, для якої вхідні дані представлені зображенням, наприклад, фотографіями або відеокадрами. Обробка зображень може здійснюватися як для одержання зображення на виході (наприклад, підготовка до поліграфічного тиражування, до телетрансляції й т.д.), так і для одержання іншої інформації (наприклад, розпізнання тексту, підрахунок числа й типу кліток у полі мікроскопа й т.д.). Крім статичних двомірних зображень, обробляти потрібно також зображення, що змінюються згодом, наприклад відео.

Ще в середині ХХ століття обробка зображень була по більшій частині аналоговою й виконувалася оптичними пристроями. Подібні оптичні методи дотепер важливі, у таких областях як, наприклад, голографія. Проте, з різким ростом продуктивності комп'ютерів, ці методи все в більшій мірі витіснялися методами цифрової обробки зображень. Методи цифрової обробки зображень звичайно є більше точні, надійні, гнучкі і прості в реалізації, ніж аналогові методи. У цифровій обробці зображень широко застосовується спеціалізоване встаткування, таке як процесори з конвеєрною обробкою інструкцій і багатопроцесорні системи. В особливій мірі це стосується систем обробки відео. Обробка зображень виконується також за допомогою програмних засобів комп'ютерної математики, наприклад, MATLAB, Mathcad, Maple, Mathematica і ін. Для цього в них використовуються як базові засоби, так і пакети розширення Image Processing.

Більшість методів обробки одномірних сигналів (наприклад, медіанний фільтр) застосовні й до двомірних сигналів, якими є зображення. Деякі із цих одномірних методів значно ускладнюються з переходом до двомірного сигналу.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Обробка зображень вносить сюди кілька нових понять, таких як зв'язність і ротаційна інваріантність, які мають сенс тільки для двомірних сигналів. В обробці сигналів широко використовуються перетворення Фур'є, а також вейвлет-перетворення й фільтр Габора. Обробку зображень розділяють на обробку в просторовій області (перетворення яскравості, гама корекція й т.д.) і частотній (перетворення Фур'є, і т.д.). Перетворення Фур'є дискретної функції (зображення) просторових координат є періодичним по просторових частотах з періодом 2π .

Типові завдання:

- Геометричні перетворення, такі як обертання й масштабування.
- Колірна корекція: зміна яскравості й контрасту, квантування кольору, перетворення в інший колірний простір.
- Порівняння двох і більше зображень. Як окремий випадок – знаходження кореляції між зображенням і зразком, наприклад, у детекторі банкнот.
- Комбінування зображень різними способами.
- Інтерполяція й згладжування.
- Поділ зображення на області (сегментація зображень).
- Редагування й ретушування.
- Розширення динамічного діапазону шляхом комбінування зображень із різною експозицією (HDR).
- Компенсація втрати різкості, наприклад, шляхом нерізкого маскуванню.

Обробка зображень у прикладних і наукових цілях:

- Розпізнавання тексту.
- Обробка супутникових знімків.
- Машинний зір.
- Обробка даних для виділення різних характеристик.
- Обробка зображень у медицині.
- Ідентифікація особистості (по особі, радужці, дактилоскопічним даним).

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

- Автоматичне керування автомобілями.
- Визначення форми об'єкта, що цікавить нас.
- Визначення переміщення об'єкта.
- Накладення фільтрів.

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.
- 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, гістограма, 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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- Передача EXIF, IPTC інформації й альфа-каналу.
- Робота в багатозадачному режимі.
- Конвертація з контекстного меню.
- Убудований планувальник.
- Робота з командного рядка.

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-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

3.2 Розробка структурної схеми

Існують наступні види й цілі 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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

кольору, наприклад, у випадку друку на паперовому носії, визначення кількості фарби для передачі кожного кольору.

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.

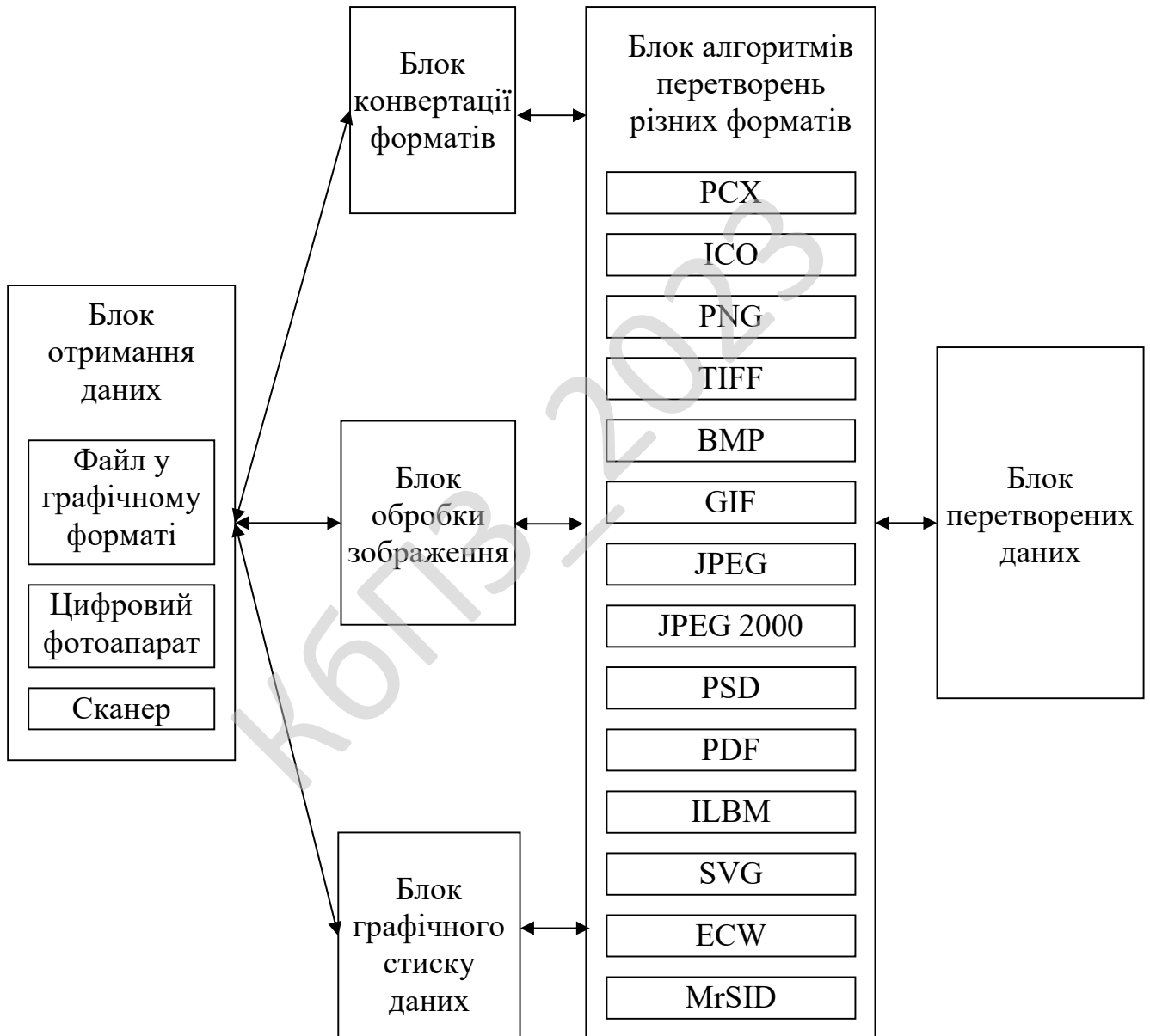


Рисунок 3.1 – Структурна схема системи

Структурна схема системи, яка зображена на рисунку 3.1, складається з наступних блоків.

1. Блок алгоритмів перетворень різних форматів.
2. Блок конвертації форматів – призначений для конвертації графічних файлів з одного формату у інший.
3. Блок графічного стиску даних – призначений для стиснення даних, отриманих зі сканера, у графічний файл.
4. Блок перетворених даних – призначений для реалізації іншого алгоритму стиску даних, ніж той, що був, до конвертації.
5. Блок отримання даних:
 - Отримання даних з файлу у графічному форматі.
 - Отримання даних зі сканера або цифрового фотоапарату.
6. Блок обробки зображення – призначений для обробки зображення, тобто являє собою невеликий графічний редактор.

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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 часто 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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

– Ретуш. У вузькому змісті ретуш – усунення непотрібних деталей зображення, шумів, зміна композиції. Часто до ретуші доводиться прибигати 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.

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 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).

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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 у край 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 і 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.

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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 кольорокоррекцією. Інші причини застосування кольорокоррекції: недостатній або надлишковий контраст зображення, вуаль, вицвітання зображення. Також необхідно відзначити кольорокоррекцію 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, кольорокоррекцію можна застосувати як інструмент реалізації творчого задуму або побажань замовника.

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 кольорокоррекції 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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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.

3.3 Розробка функціональної схеми

Конвертуються наступні формати:

– 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 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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

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 відобразити 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.

– Формат TIFF – формат зберігання растрових графічних зображень. Був розроблений компанією Aldus у співробітництві з Microsoft для використання з PostScript. TIFF став популярним форматом для зберігання зображень із великою глибиною кольору. Він використовується при скануванні, відправленні факсів, розпізнаванні тексту, у поліграфії, широко підтримується графічними додатками.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

TIFF був обраний у якості основного графічного формату операційної системи NeXTStep і з неї підтримка цього формату перейшла в Mac OS X. Компанія-власник специфікацій – Aldus Corporation – згодом об'єдналася з Adobe, що володіє в цей час авторським правом на ці специфікації. Файли формату TIFF, як правило, мають розширення .tiff або .tif.

– Формат ECW – пропрієтарний формат файлів растрових зображень, оптимізований для зберігання аерофотознімків і космічних знімків, що використовує техніку сплеск-стиску із втратами даних. Формат розроблений компанією Earth Resource Mapping. Формат ефективно стискає надвеликі зображення із прекрасним динамічним контрастом. Формат ECW дозволяє зберігати дані про систему координат зображення місцевості (картографічну проекцію та ін.) безпосередньо в самому файлі зображення. Графічні дані можуть стискуватися зі швидкістю 1,5 Мб/с процесором із частотою 1 Гц. Процес стиску даних не вимагає використання великого обсягу оперативної пам'яті для великих зображень. Даний формат файлу дозволяє одержати ступінь стиску від 1:10 до 1:100. Формат дозволяє витягати ділянки зображення без необхідності буферизації й розпакування всього зображення. Існує безкоштовна бібліотека програм для стиску у форматі ECW зображень обсягом до 500 Мб. Вихідний код доступний тільки для ознайомлення. Для інших застосувань користувачам потрібно виплатити винагороду, якщо обсяг даних перевищує 500 Мб. Більшість стандартних в індустрії геоінформатики програмних продуктів підтримують формат ECW (наприклад, сімейство продуктів GeoMedia корпорації Intergraph).

– Формат ILBM – різновид файлів формату IFF використовуваний для зберігання даних растрової графіки. Форматом підтримується черезрядкова розверстка, кольорове зображення представлене сукупністю однобітних масивів, кожний з яких зберігає 1 біт даних для одного пікселя в зображенні. Такі масиви називають біт-планами, або бітмапами. Також, форматом підтримується горизонтальний і вертикальний стиск дані зображення алгоритмом RLE. Формат ILBM був розроблений для найбільш повного розкриття можливостей

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

чипсета класичної Amiga.

– Формат MrSID – формат графічних файлів (з розширенням .sid), розроблений і запатентований компанією LizardTech. Використовується для стиску растрової графіки, використовуваної в ГІС, наприклад, для ортографічно скоректованого аерофотознімання.

– Формат PSD – растровий формат зберігання графічної інформації, що використовує стиск без втрат, створений спеціально для програми Adobe Photoshop і підтримуючий всі його можливості.

– Формат SVG – мова розмітки масштабованої векторної графіки, створена Консорціумом Всесвітньої павутини (W3C) і вхідна у підмножину розширюваної мови розмітки XML, призначеної для опису двовимірної векторної й змішаної векторно/растрової графіки у форматі XML. Підтримує як нерухливу, так анімовану й інтерактивну графіку – або, в інших термінах, декларативну й скриптову. Це відкритий стандарт, є рекомендацією консорціуму W3C, – організації, що розробила такі стандарти, як HTML і XHTML. Розробляється з 1999 року, в 2001 році вийшла 1.1 версія, що залишається актуальною до сьогоднішнього дня, в активній розробці версія 1.2. В основу SVG лягли мови розмітки VML і PGML.

– Формат PDF – кроссплатформений формат електронних документів, створений фірмою Adobe Systems з використанням ряду можливостей мови PostScript. У першу чергу призначений для подання в електронному виді поліграфічної продукції, – значна кількість сучасного професійного друкованого встаткування може обробляти PDF безпосередньо. Для перегляду можна використовувати офіційну безкоштовну програму Adobe Reader, а також програми сторонніх розроблювачів. Традиційним способом створення PDF-документів є віртуальний принтер, тобто документ як такий готується у своїй 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

						ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			49

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.

– Формат JPEG – один з популярних графічних форматів, застосовуваний для зберігання фотозображень і подібних їм зображень. Файли, що містять дані JPEG, звичайно мають розширення .jpeg, .jif, .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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

– 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.

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функціональна схема складається з наступних блоків:

1. Головне вікно інтерфейсу користувача.
2. Блок функцій по конвертуванню графічних форматів з одного в інший.
3. Конвертація прямо з буфера обміну.
4. Зняття скріншотів.
5. Створення власного слайд-шоу з ефектами.
6. Блок пакетної обробки.
7. Підтримка декількох мов.
8. Блок функцій по редагуванню зображень:
 - авто-обрізка;
 - кадрування;
 - видалення нахилу;
 - фліп;

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- дзеркальне відображення;
- зміна розміру зображення;
- поворот зображення;
- зрушення зображення;
- додавання шуму;
- вигин зображення;
- видалення границь.

9. Блок ефектів:

- тиснення;
- розмиття;
- гравірування;
- корекція гама;
- 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;

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

- ефект старих фотографій;
- пікселізація;
- полярність;
- постеризовування;
- панч, радіальний хвильовий;
- хвильовість;
- насиченість;
- різкість;
- гладкість.

10. 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.

11. Панель меню користувача:

- Файл.
- Виправлення.
- Зображення.
- Ефекти.
- Вид.
- Масштаб.
- Утиліти.
- Вибране.
- Довідка.

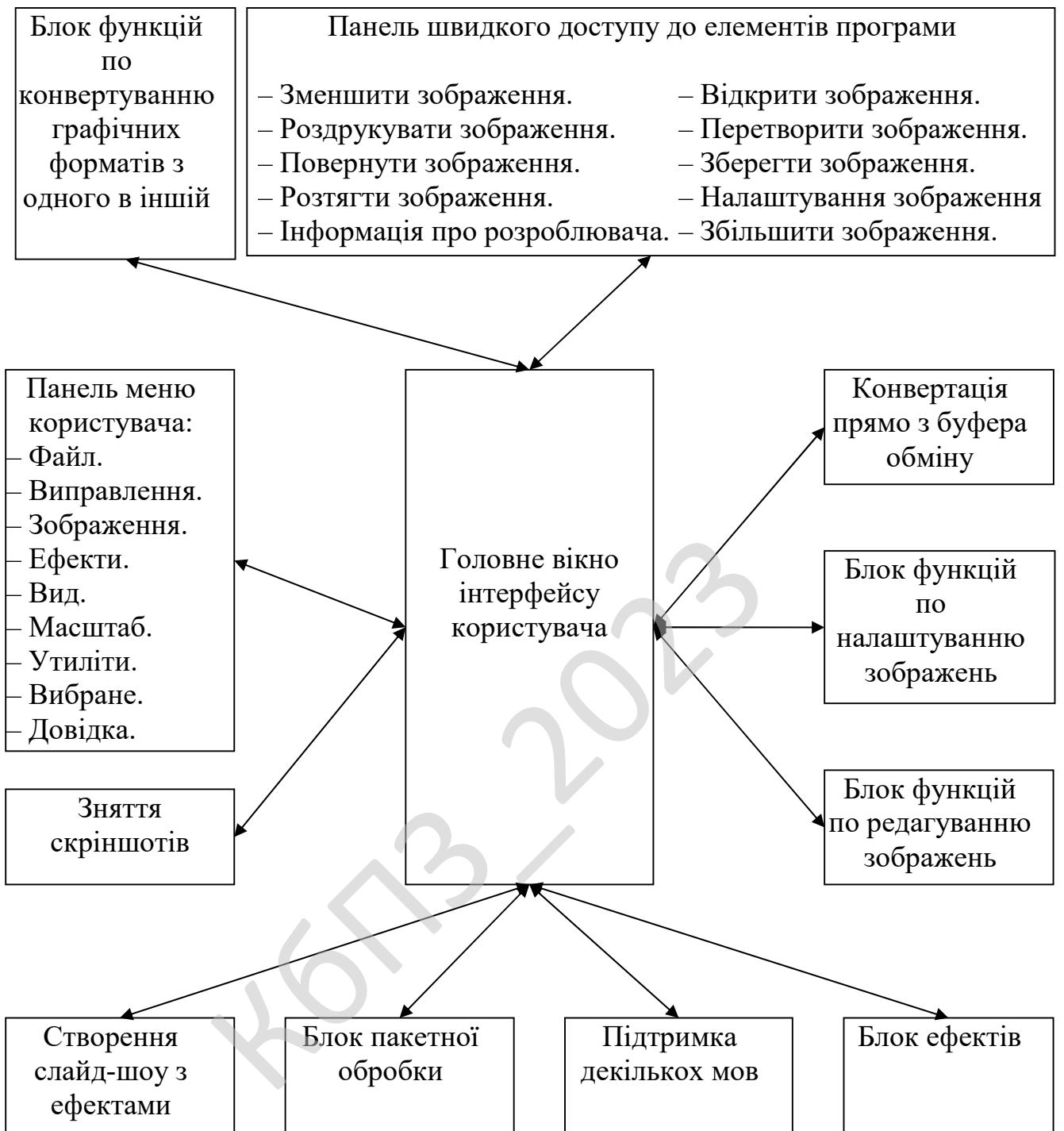


Рисунок 3.2 – Функціональна схема системи

12. 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.

Наведемо опис структури найбільш розповсюджених форматів даних.

GIF

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).

Розглянутий нами нижче варіант алгоритму буде використовувати дерево для подання й зберігання ланцюжків. Очевидно, що це досить сильне обмеження на вид ланцюжків, і далеко не всі однакові підланцюжки в зображенні будуть використані при стиску. Однак у пропонуваному алгоритмі вигідно стискати навіть ланцюжки, що складаються з 2 байт.

Процес стиску виглядає досить просто. Ми зчитуємо послідовно символи вхідного потоку й перевіряємо, є чи в створеній нами таблиці рядків такий рядок. Якщо рядок є, то ми зчитуємо наступний символ, а якщо рядка немає, то ми заносимо в потік код для попереднього знайденого рядка, заносимо рядок у таблицю й починаємо пошук знову.

Функція InitTable, реалізована на псевдокоді, очищає таблицю й поміщає в неї всі рядки одиничної довжини.

```

InitTable;
CompressedFile.WriteCode(ClearCode);
CurStr:=порожній рядок;
While (не ImageFile.EOF) begin //Поки не кінець файлу
    C:=ImageFile.ReadNextByte;

```

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```

If (CurStr+C є в таблиці) then
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.

Алгоритм декомпресії, що здійснює цю операцію, реалізований на псевдокоді, виглядає в такий спосіб:

```
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.

Алгоритм JPEG

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.

JPEG – один із самих нових і досить потужних алгоритмів. Практично він є стандартом де-факто для повнокольорових зображень [1]. Оперує алгоритм областями 8x8, на яких яскравість і колір міняються порівняно плавно. Внаслідок цього, при розкладанні матриці такої області в подвійний ряд по косинусах (див. формули нижче) значимими виявляються тільки перші коефіцієнти. Таким чином, стиск в JPEG здійснюється за рахунок плавності зміни кольорів у зображенні. Алгоритм розроблений групою експертів в області фотографії спеціально для стиску 24-бітних зображень. JPEG – Joint Photographic Expert Group – підрозділ у рамках ISO – Міжнародної організації по стандартизації. У цілому алгоритм заснований на дискретному косинусоїдальному перетворенні (надалі ДКП), застосовуваному до матриці зображення для одержання деякої 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.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Крок 1.

Переводимо зображення з колірного 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 колірного 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 виконувється 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:

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

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

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).

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

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 .

Алгоритм декомпресії, реалізований на псевдокодi, 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 лічильник, що може приймати значення від 1 до 64. Рядок з 64 повторюваних байтів ми перетворюємо у два байти, тобто стиснемо в 32 рази.

Алгоритм розрахований на ділову графіку – зображення з великим областями повторюваного кольору. Ситуація, коли файл збільшується, для цього простого алгоритму не така рідка. Її можна легко одержати, застосовуючи групове кодування до оброблених кольорових фотографій. Для того щоб збільшити зображення у два рази, його треба застосувати до зображення, у якому значення всіх пікселів більше двійкового 11000000 і підряд попарно не повторюються.

Даний алгоритм реалізований у форматі РСХ.

Другий варіант алгоритму

Другий варіант цього алгоритму має більший максимальний коефіцієнт архівації й менше збільшує в розмірах вихідний файл.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Алгоритм декомпресії, реалізований на псевдокоді, для нього виглядає

так:

```
Function Initialization;
do
  byte := ImageFile.ReadNextByte;
  counter := Low7bits(byte)+1;
  if (якщо ознака повтору(byte)) then
    begin
      value := ImageFile.ReadNextByte;
      for I := 1 to counter do
        CompressedFile.WriteByte(value);
      end
    else
      for i:=1 to counter) do
        begin
          value := ImageFile.ReadNextByte;
          CompressedFile.WriteByte(value);
        end
      CompressedFile.WriteByte(byte);
    while (ImageFile.EOF);
```

Ознакою повтору в даному алгоритмі є одиниця в старшому розряді відповідного байта. Як можна легко підрахувати, у найкращому разі цей алгоритм стискає файл в 64 рази (а не в 32 рази, як у попередньому варіанті), у гіршому збільшує на 1/128.

Середні показники ступеня компресії даного алгоритму перебувають на рівні показників першого варіанта.

Схожі схеми компресії використані в якості одного з алгоритмів, підтримуваних форматом TIFF, а також у форматі TGA.

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.5.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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 головного вікна ПЗ та запит відкриття файлу. Після чого проходить завантаження обраних файлів зображень, виведення списку зображень та запит обрання файлу.

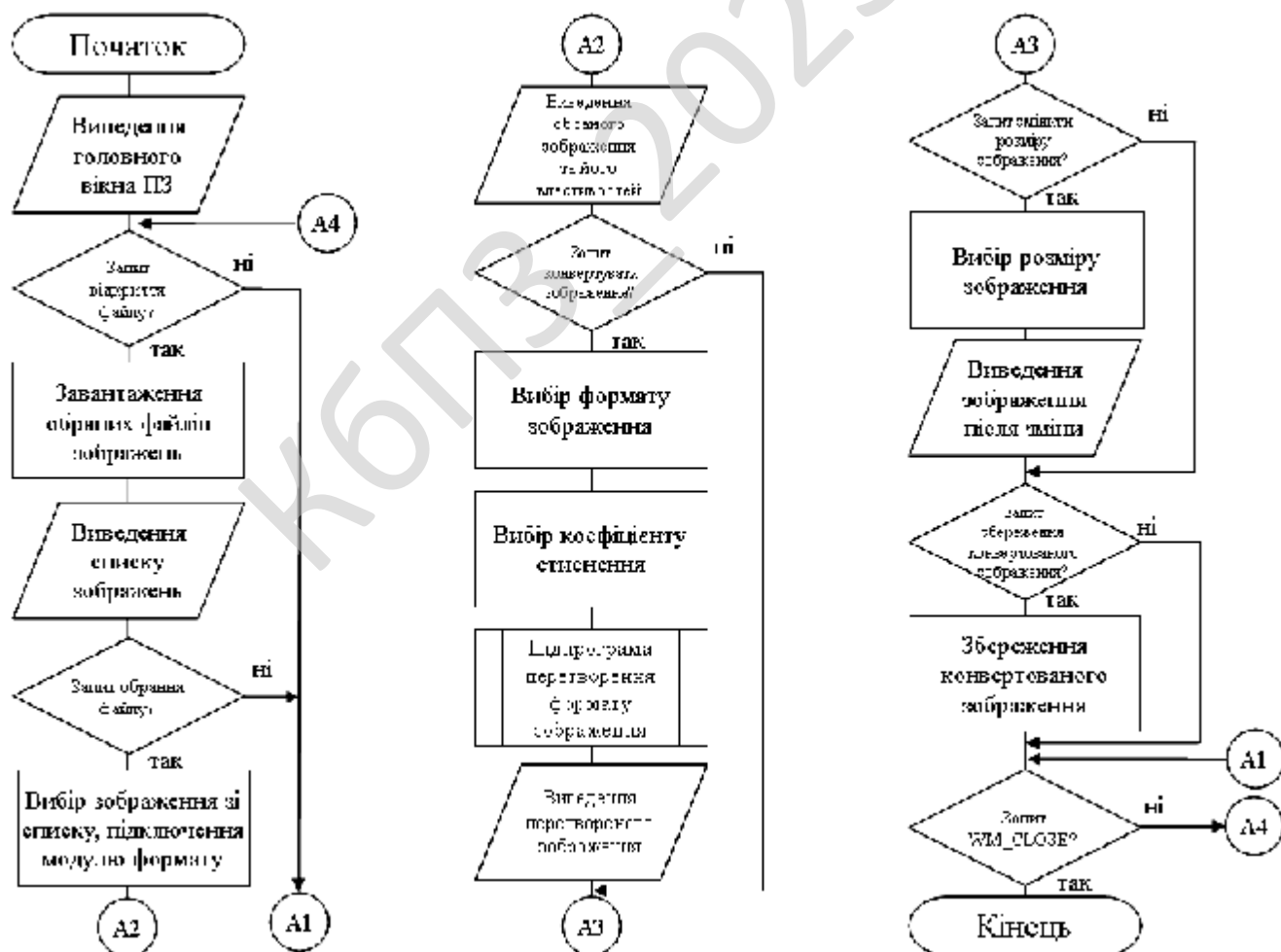


Рисунок 4.1 – Блок-схема основної програми

Далі вибір зображення зі списку, підключення модулю формату. виведення обраного зображення та його властивостей та запит конвертувати зображення. При виконанні запиту проходить вибір формату зображення, вибір коефіцієнту стиснення та виконання підпрограми перетворення формату зображення яка зображена на рисунку 4.2.

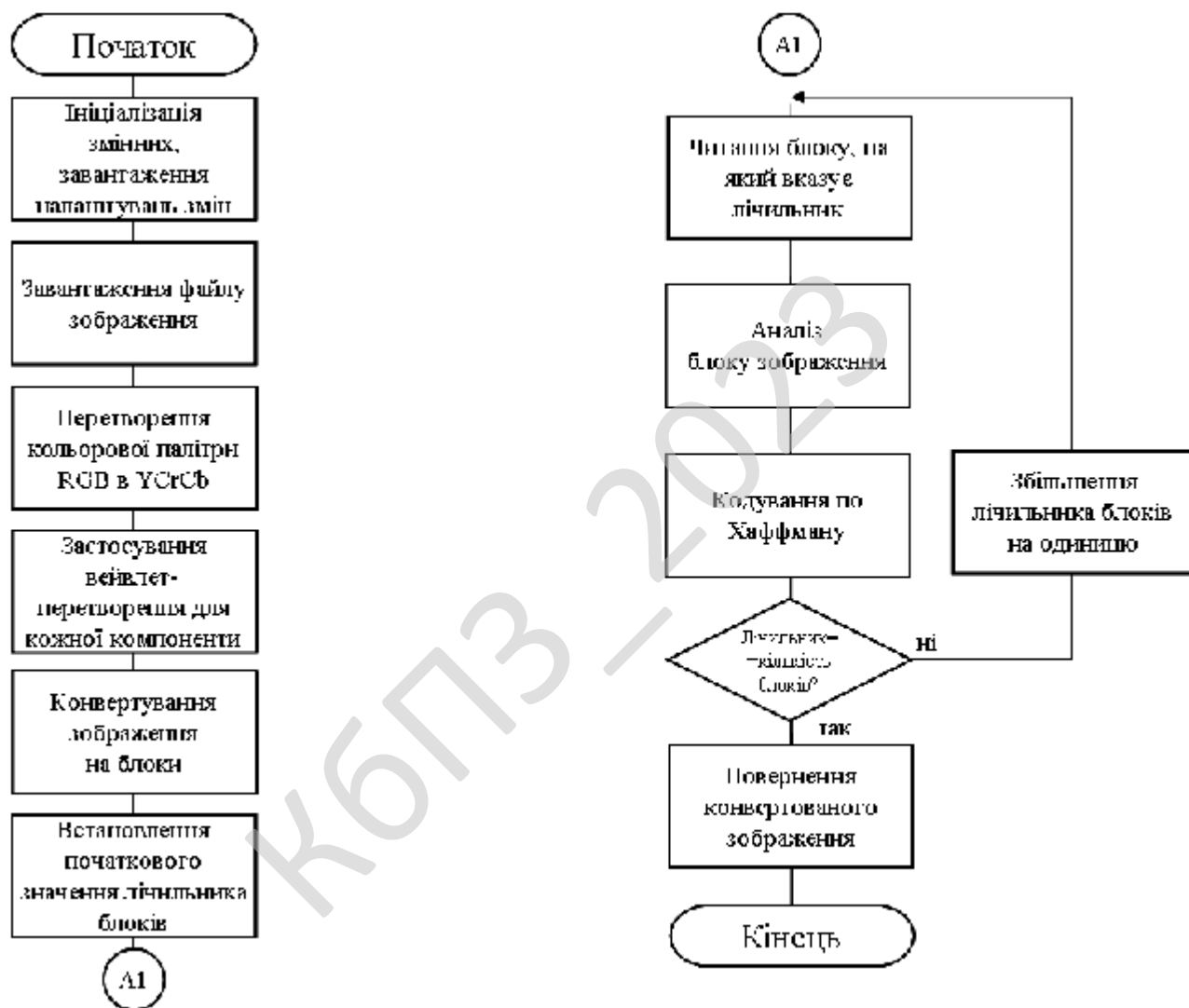


Рисунок 4.2 – Блок-схема підпрограми

Спочатку проходить ініціалізація змінних, завантаження налаштувань змін, завантаження файлу зображення, перетворення кольорової палітри RGB в YCrCb та застосування вейвлет-перетворення для кожної компоненти. Далі проходить

конвертування зображення на блоки, встановлення початкового значення лічильника блоків, читання блоку, на який вказує лічильник, аналіз блоку зображення та кодування по Хаффману. Якщо лічильник буде дорівнювати кількості блоків проходить повернення конвертованого зображення до головної блок схеми. Далі проходить виведення перетвореного зображення, запит змінити розміру зображення з подальшим вибором розміру зображення та виведенням зображення після зміни. Запит збереження конвертованого зображення з подальшим збереженням конвертованого зображення. Та запит WM_CLOSE який при виконанні завершує роботу програми.

Розглянемо розроблену частину вихідного коду яка проводить 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. }

```

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70


```

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.

```

Розглянемо розроблену частину вихідного коду яка реалізує деякі функції кодування файлу у заданий графічний формат.

Конвертування BMP в EMF.

```

function bmp2emf(const SourceFileName: TFileName): Boolean;
var
  Metafile: TMetafile;
  MetaCanvas: TMetafileCanvas;

```

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

```

    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
    bmp2emf('C:\TestBitmap.bmp');
end;

```

Розглянемо розроблену частину вихідного коду яка проводить конвертування JPG до формату 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

```

					БКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74


```

{Створюємо маску "And"}
  AndMask := TBitmap.Create;
  AndMask.Monochrome := true;
  AndMask.Width := IconSize;  AndMask.Height := IconSize;
{Малюємо на масці "And"}
  AndMask.Canvas.Brush.Color := clWhite;
  AndMask.Canvas.FillRect(Rect(0, 0, IconSize, IconSize));
  AndMask.Canvas.Brush.Color := clBlack;
  AndMask.Canvas.Ellipse(4, 4, IconSize - 4, IconSize - 4);
{Малюємо для тесту}
  Form1.Canvas.Draw(IconSize * 2, IconSize, AndMask);
{Створюємо маску "XOr"}
  XOrMask := TBitmap.Create;
  XOrMask.Width := IconSize;
  XOrMask.Height := IconSize;
{Малюємо на масці "XOr"}
  XOrMask.Canvas.Brush.Color := clBlack;
  XOrMask.Canvas.FillRect(Rect(0, 0, IconSize, IconSize));
  XOrMask.Canvas.Pen.Color := clRed;  XOrMask.Canvas.Brush.Color := clRed;
  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;

```

Розглянемо розроблену частину вихідного коду яка проводить конвертування BMP в JPG.

uses

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

```

    Jpeg, ClipBrd;
procedure TfrmMain.ConvertBMP2JPEG;
var
    jpgImg: TJPEGImage;
begin
    // копіюємо bitmap 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');
end;

```

Розглянемо розроблену частину вихідного коду яка проводить конвертування ICO до формату 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;

```

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На кожному етапі половина об'єднується за допомогою функції F[1] з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N-го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

					VKPM-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми інтуїтивно зрозумілий та складається з таких логічних блоків:

- Блоку обрання папки та файлів;

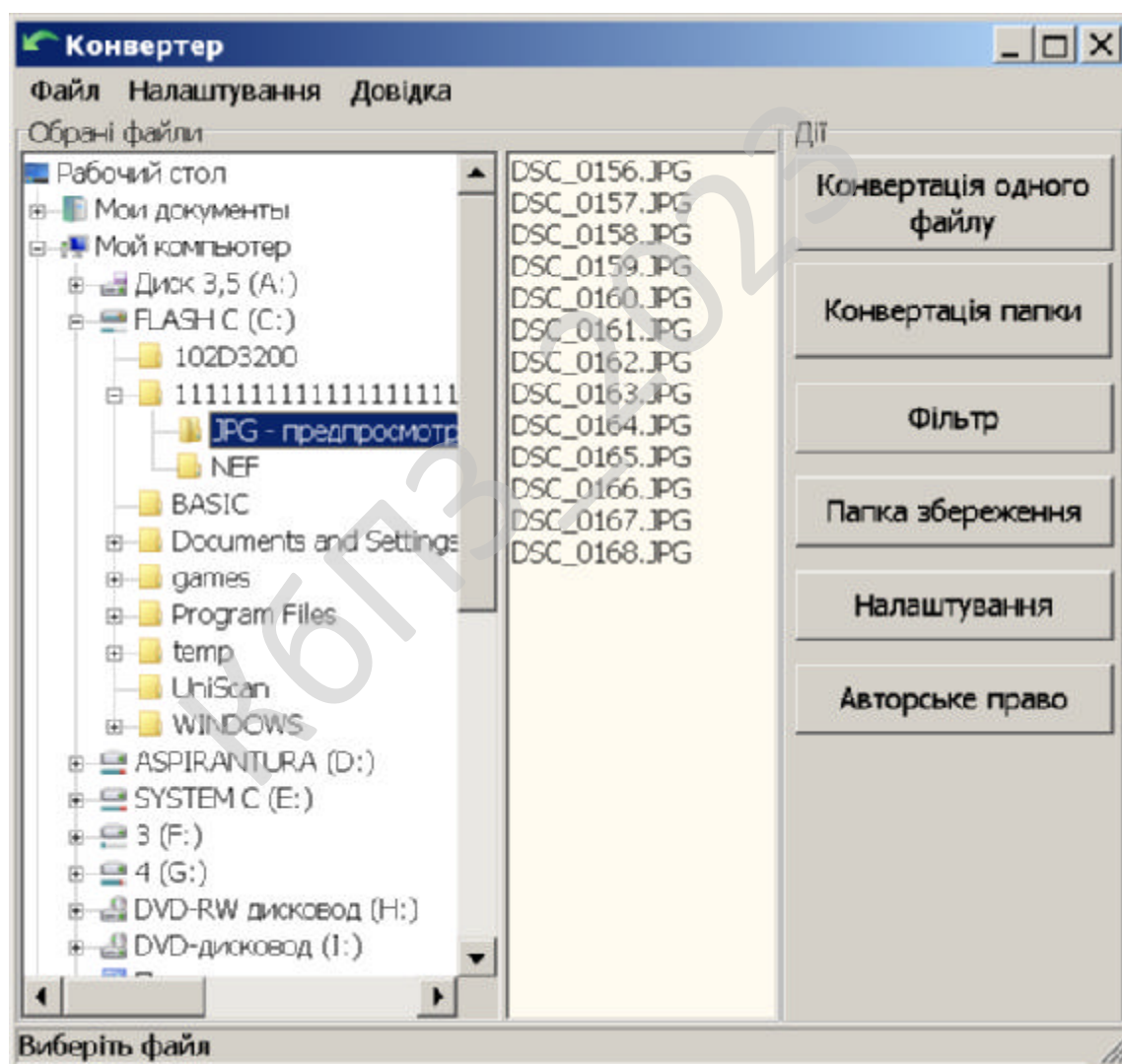


Рисунок 5.1 – Головне вікно програми

- Блоку введення змісту папки;
- Блоку функцій програми.

На рисунку 5.2 зображено вікно налаштувань програми з обраною мовою інтерфейсу програми – англійська. Також є можливість встановити українську мову.

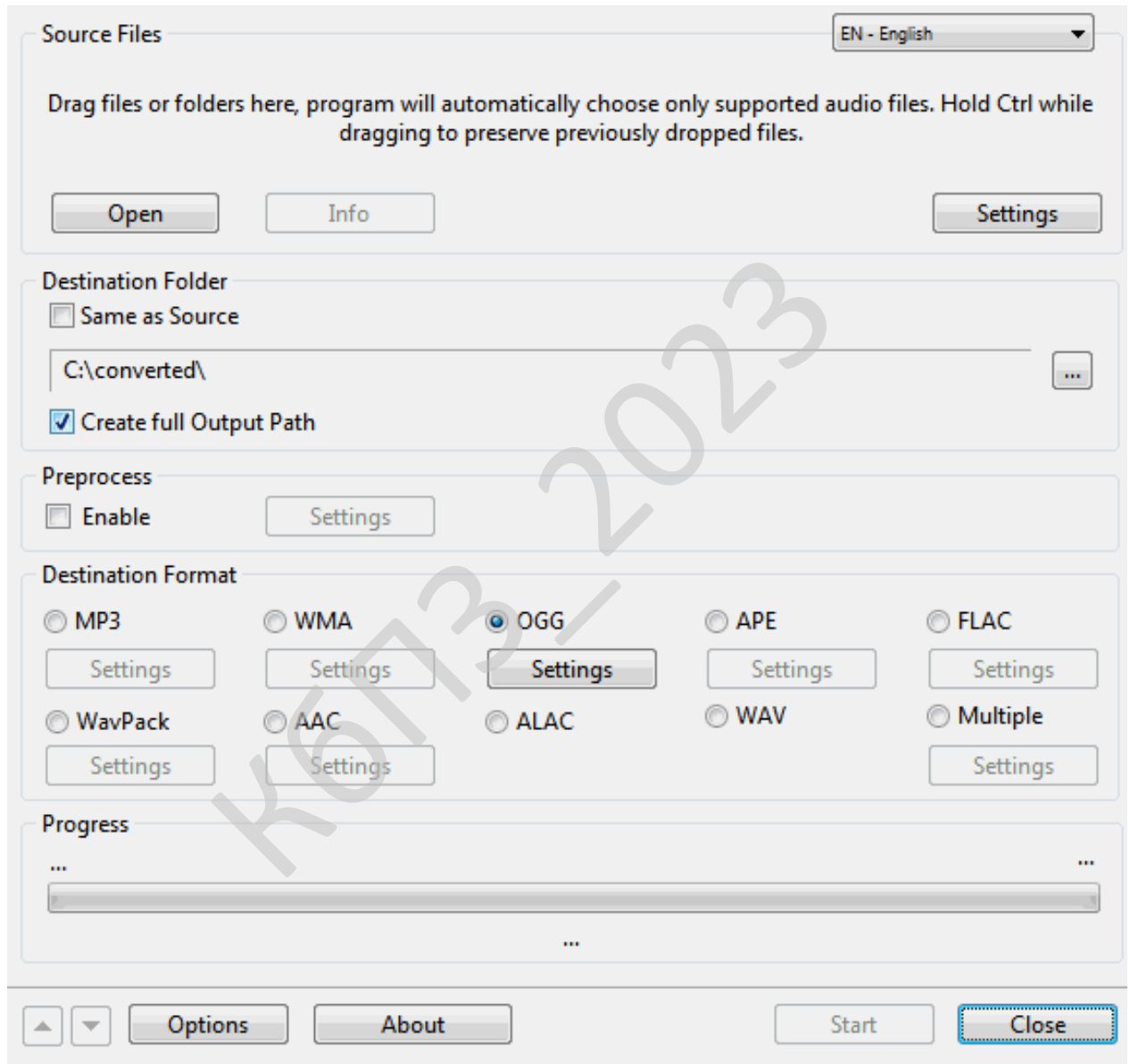


Рисунок 5.2 – Вікно налаштувань програми, встановлення мови інтерфейсу

На рисунку 5.3 зображено форма авторського права. Обрана ліцензія розповсюдження програми – freeware (вільне розповсюдження).

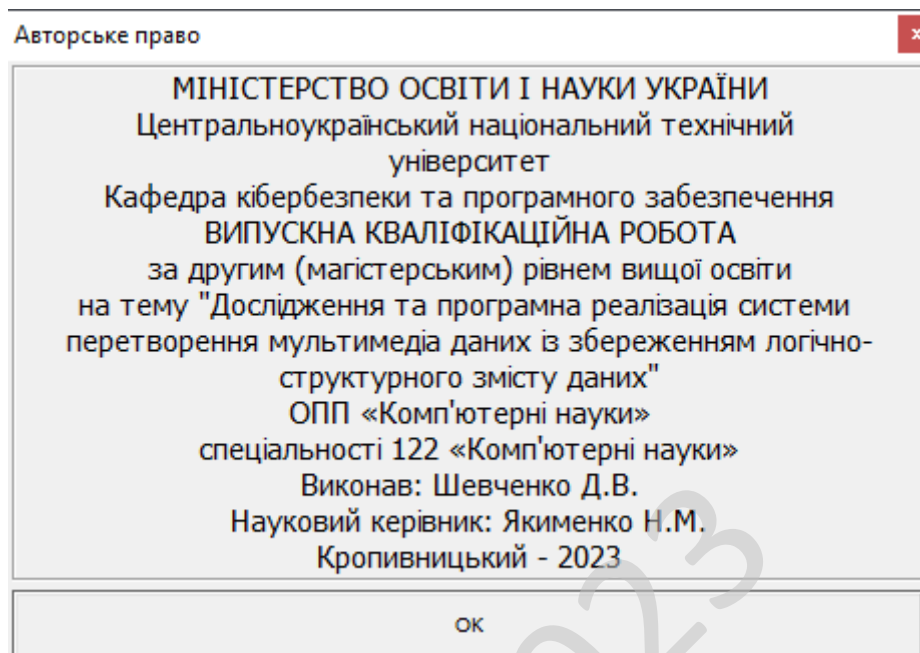


Рисунок 5.3 – Довідка

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Метою розробки є дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Об'єктом дослідження є процес перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Предметом дослідження є методи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

– Розроблено вітчизняний продукт перетворення мультимедіа даних із збереженням логічно-структурного змісту даних, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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 були проведені дослідження та виконана програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних. 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	20
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.	–	20000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

B – показник ступеня, що визначається співвідношенням:

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

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

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

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

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

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

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

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

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

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

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

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

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

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33 + 0,2(1,026 - 1,01)} \cdot 35 = 59 \text{ люд/день.}$$

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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	59	Ф 7.1-7.4
Впровадження	13	Д13
Всього	100	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{100 \cdot 1}{60-5} = 1,82 \text{ ставки.}$$

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	8	720	12
Монітор	60	8	480	8
Клавіатура	30	8	240	4
Маніпулятор «мишка»	30	8	240	4
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	2	60	1
Кабельні господарства ЛОМ на 1 м.п.	2,5	150	375	6,25
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	40,91

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:

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

$$\Phi_{\text{др}}^c = \frac{41 \cdot 3}{1,2} = 102,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 102,5 / (60 \cdot 8) = 0,2 \text{ ставки.}$$

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2022, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

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	18213,3	27320
Продакт-менеджер	0,25	14000	10500
Інженер-програміст	1,82	13000	70980
Інженер-електронщик	0,2	12000	7200
Інженер-системотехнік	0,25	12000	9000
Адміністратор мережі	0,5	12000	18000
Системний програміст	0,25	12000	9000
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	12000	9000
Бухгалтер-економіст	0,5	12213,3	18320
Всього за період розробки	$R_{cn} = 4,77$	-	$\Phi_{роб} = 188320$

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.

$$z_{cd} = \frac{188320}{4,77 \cdot 60} = 658 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

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

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

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

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 400...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:

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

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

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

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 18.10.23 – джерело <https://compbest.com.ua>.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
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

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	12	10947	13136,4	144500,4
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	163051,9

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	163051,9	-	-
Всього по групі	163051,9	50	81525,95
Група 5, 6			
4. Вимірювальні пристрої	9031	25	2257,75
5. Транспортні засоби	143000	20	28600
6. Господарський інвентар	28000	25	7000
Всього по групі 5,6	180031	-	37857,75
Нематеріальні активи			
7. Нематеріальні активи	20000	10	2000
Разом	$K_p = 1771082,9$		$A_p = 189783,7$

Примітка: вартість автомобіля взята за даними електронного ресурсу, джерело https://auto.ria.com/uk/auto_renault_scenic_33598032.html, складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 658 \cdot 100 / 20 = 3292 \text{ грн.}$$

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

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

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 3292 \cdot 10 \cdot 0,01 = 329 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(3292 + 329) = 797 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 3292 \cdot 15 \cdot 0,01 = 494 \text{ грн.}$$

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

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

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

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 2):

$$З_{M2} = \sum Ц_d, \quad (7.17)$$

де: $Ц_d$ – вартість дисків CD/DVD: CDR box – 21,5 грн./шт., DVD-R box – 39 грн./шт.

$$З_{M2} = 21,5 \cdot 2 = 43 \text{ грн.}$$

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

$$З_{M3} = \sum Ц_z, \quad (7.18)$$

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

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

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

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

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

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

$$O_n = 3292 \cdot 15 \cdot 0,01 = 494 \text{ грн.}$$

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

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

$$A_m = 189783,7 \cdot 3 / (20 \cdot 12) = 2372 \text{ грн.}$$

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

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

$$C_n = 3292 + 392 + 797 + 494 + 93 + 494 + 2372 = 7934 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 55%.

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

де: P_n – рівень рентабельності, %.

$$\Pi_p = 0,01 \cdot 55 \cdot 7934 = 4364 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	3292
2. Додаткова зарплата виконавців	Z_d	392
3. Відрахування на соціальні потреби	C_{oc}	797
4. Загальногосподарські витрати	Γ_{ocn}	494
5. Витрати на матеріали	Z_m	93
6. Освоєння нових операційних систем, мов програмування	O_n	494

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	2372
8. Повна собівартість програмного забезпечення	C_n	7934
9. Плановий прибуток	P_p	4364
10. Ціна підприємства $C_n = C_n + P_p$	C_n	12298
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	2459,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	14757,6

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	–	14758
Всього капітальних витрат	–	14758

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	26840	12078
2. Витрати на електроенергію	$Z_{ел}$	4104	3648
3. Витрати на амортизацію	$Z_{ам}$	0	3690
Всього витрат за рік	I	30944	19416

Витрати на обслуговування 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 системи за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 200 годин на рік до 90 годин на рік, тому витрати на технічне обслуговування зменшилися з:

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

до:

$$Z_{p \text{ нов}} = 90 \cdot 100 \cdot 1,1 \cdot 1,22 = 12078 \text{ грн.}$$

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

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

$$Z_{ел баз} = 0,45 \cdot 2400 \cdot 3,8 = 4104 \text{ грн.}$$

$$Z_{ел нов} = 0,40 \cdot 2400 \cdot 3,8 = 3648 \text{ грн.}$$

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	–	14758	–	3689,5
Всього відрахувань	-	–	14758	–	3689,5

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.

$$E_e = (12298 - 7934) \cdot 20 - (0,05 \cdot 1408000 + 0,5 \cdot 163051,9 + 0,2 \cdot 143000 + 0,25 \cdot 37031 + 0,1 \cdot 20000) \cdot 3/12 = 39334 \text{ грн.}$$

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 без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_6 = \frac{363083}{(12298 - 7934) \cdot 20 \cdot 12 / 3} = 1 \text{ рік.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	20
2. Повна собівартість розробленої програми	Грн.	7934
3. Ціна розробленої програми	Грн.	12298
4. Плановий прибуток від реалізації розробленої програми	Грн.	4364
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1771083
7. Загальний прибуток від реалізації програмної продукції	Грн.	87280
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	39334
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Рік	1
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	14281
11. Величина економічного ефекту у користувача програмної продукції	Грн.	7839
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Рік	1,28

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

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

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

$$E_{cn} = (30944 - 19416) - 0,25 \cdot 14758 = 7839 \text{ грн.}$$

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

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

$$T_{cn} = \frac{14758}{29108 - 17664} = 1,28 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Стрімке впровадження комп'ютерів не тільки в сфері управління виробництвом, в банківській системі, бізнесі, системі освіти, але також на транспорті, сфері обслуговування призвело до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння.

Техніка безпеки – це система правил і заходів, які допомагають запобігти травмам, хворобам і аваріям на робочому місці або в повсякденному житті. Знання техніки безпеки дуже важливе, бо воно рятує життя і здоров'я людей.

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207 [4], який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження [2]. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста вулючають наступні фактори:

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Можна виділити наступні основні фактори, що впливають на стан здоров'я людей, які працюють за комп'ютером:

- сидяче положення на протязі тривалого періоду;
- вплив електромагнітного випромінювання монітора;
- втома очей, навантаження на зір;
- перевантаження суглобів кистей;
- стрес при втраті інформації.

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп'ютером і поблизу нього. В сучасних умовах взаємодія людини

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

з технікою значно ускладнилась, що вимагає комплексного підходу, який передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищесказане в повній мірі відноситься й до системи «людина–комп'ютер–середовище».

Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму, випромінювання.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам та запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини необхідно здійснити санітарно-гігієнічну характеристику умов працівника, який працює з програмним продуктом.

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

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	6,2
Довжина	8
Висота	3,4

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

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6,2
Обсяг, V	м ³	не менше 20.0	21,1

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин) [2].

У зазначеному приміщенні працює 8 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа приміщення та а об'єм у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2], НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови Головного державного санітарного лікаря України [5], робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 кКал. у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

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

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$ [6].

Обчислимо індекс приміщення за формулою:

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

де: S – площа приміщення, $S = 49,6 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3,4 \text{ м}$;

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

B – довжина приміщення, $B = 8 \text{ м}$.

Підставимо всі значення у формулу та визначемо індекса приміщення:
 $i = 0,57$.

Знаючи індекс приміщення, знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників відповідного типу). Підставимо всі значення у формулу, визначимо світловий потік: $F = 53374 \text{ Лм}$.

Для штучного освітлення приміщення використовуються LED панель MAXUS ASSISTANCE PRO 80W 5000K WHITE (M1052480531), світловий потік якої $F_{л} = 8000 \text{ Лм}$.

Число світильників визначається по формулі:

$$N = F / F_{л}$$

де: F – світловий потік,

$F_{л}$ – світловий потік одного світильника.

$$N = 53374 / 8000 = 6,7 \text{ шт.}$$

Приймаємо необхідну кількість світильників 7 шт.

Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

негативно впливають на програмістів під час роботи. Виконано розрахунок захисного штучного освітлення. Розроблено заходи з охорони праці.

Список використаних джерел інформації

1. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2.007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
4. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>
6. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення: 16.06.2023).
7. Методичні рекомендації до виконання розділу «Заходи з охорони праці та техніки безпеки» у магістерській дисертації / Л.Д. Третьякова; М-во освіти і науки України, Національний технічний університет України «Київський політехнічний інститут» – Київ, КПІ, 2014. – 26 с. – Режим доступу до ресурсу: <http://surl.li/dhulo> (дата звернення: 16.06.2023).

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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 перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.
- Досліджена система перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.
- На основі отриманих результатів досліджень створена програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

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

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм FEAL.

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 7839 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 1,28 роки.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шевченко Д.В. Дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
4. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
5. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
6. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
7. 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.
8. 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. ун-т. – Д.: НГУ, 2016. – 187 с.
9. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph.. — 2011. — Vol. 30, no. 4. — P. 99:1--99:8.
10. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. — 2011. — Vol. 20, no. 10. — P. 2760—2768.
11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

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

12. 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,

13. 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.

14. 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>

15. 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.

16. 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.

17. 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.

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

					БКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

41. 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.

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

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

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

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

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

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

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120

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

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

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

51. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

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

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

54. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

					ВКРМ-122.23.0050.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		121

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0050.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Шевченко Д.В.				<i>Дослідження та програмна реалізація системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних</i>	Літ.	Аркуш	Аркушів
Перевірів	Якименко Н.М.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-2			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

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. № 33-13 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 перетворення мультимедіа даних із збереженням логічно-структурного змісту даних.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-122.23.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи перетворення мультимедіа даних із збереженням логічно-структурного змісту даних;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.

					ВКРМ-122.23.0050.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0050.00.00.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 – 121 аркуш.

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 21.12.2023 р.

					ВКРМ-122.23.0050.00.00.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

_____ Якименко Н.М.

***Дослідження та програмна реалізація
системи перетворення мультимедіа даних із збереженням логічно-
структурного змісту даних***

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 29

Літера: РП

Кропивницький – 2023 року

Додаток Б
(обов'язковий)

Міністерство освіти і науки
Кіровоградський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник магістерського проекту
_____ Якименко Н.М.

*Програмне забезпечення системи перетворення мультимедіа даних із
збереженням логічно-структурного змісту даних*

Лістинг програми

Код документу 12

Носій: дискета

Загальна кількість аркушів: 28

Літера: РП

Кропивницький 2023

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;
  end;
end;

```

```

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; IS_PROCESSING_IMAGES: boolean =
false; // /
  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 := 'abcdefghijklmnopqrstuvwxy1234567890ABCDEFGHIJKLMNQPQRSTUVWXYZ';
  // 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 _Peretvorenya_multimedia_data, JPEG, BMP):';
      ErrorForm.FilesConcernedMemo.Lines := ErrorsList;

```

```

        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)) = '_Peretvorenya_multimedia_data')
            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 _Peretvorenya_multimedia_data, 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;
        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 _Peretvorenyya_multimedia_data, 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 _Peretvorenyya_multimedia_data, 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;
  finally
    Free;
  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 := 'Quitter'
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))='._Peretvorennya_multimedia_data') 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, '_Peretvorennya_multimedia_data'));
            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. ');
            Continue;
        end;

        // if (RBtn_OverwriteFiles.Checked) and
        (OriginalFileExtension='.bmp') then DeleteFile(FileName);
        if RBtn_CreateZIPFile.Checked then
ZIPPendingFiles.Add(FinalDestinationFileName);
        Application.ProcessMessages;

        if RBtn_OverwriteFiles.Checked then
MetadataSourceFileName:=BackupDestinationFileName
        else MetadataSourceFileName := FileName;
        if RBtn_KeepAllMetadata.Checked then begin
            LaunchAndWait(''+EXIFTOOL_PATH+' -TagsFromFile
''+MetadataSourceFileName+' -all:all ''+FinalDestinationFileName+' -P -q -m -
overwrite_original', SW_HIDE);
            Application.ProcessMessages;
        end;
        if CBox_AddGlobalComment.Checked then begin
            LaunchAndWait(''+EXIFTOOL_PATH+' -
UserComment='''+CommentText.Text+' -XPCComment='''+CommentText.Text+' -
xmp:UserComment='''+CommentText.Text+' -Comment='''+CommentText.Text+'
''+FinalDestinationFileName+' -P -q -m -L -overwrite_original', SW_HIDE);
            Application.ProcessMessages;
        end;
        if CBox_AddCopyright.Checked then begin

```

```

        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;
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.

```

EasyCompress_Peretvorenyya_multimedia_data.dpr - файл проекту

```

program EasyCompress_Peretvorenyya_multimedia_data;

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_Peretvorenyya_multimedia_data.res}

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

  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;
    CompressionRate: 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))='.Peretvorenya_multim
edia_data') 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))='._Peretvorenya_multim
edia_data') 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 effacy 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.
```

about.pas - файл довідки

```
unit about;

interface

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

type
  TForm5 = class(TForm)
    Memo1: 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
  Memo1.Clear;
  Memo1.Lines.Add(' МАГІСТЕРСЬКИЙ ПРОЕКТ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' Дослідження та програмна реалізація системи перетворення  
мультимедіа даних із збереженням логічно-структурного змісту даних ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' Керівник: Якименко Н.М. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' Розробив: студент Шевченко Денис Валерійович');
  Memo1.Lines.Add(' гр. КН-22М-2');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Кропивницький 2023');
  Memo1.Lines.Add('');
end;

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