

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
багатофункціонального менеджера завантажень мультимедіа
даних з мережі Internet”

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

Керівник проекту
кандидат технічних наук
_____ Буравченко К.О.
« ____ » _____ 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

Мікіньову Ігорю Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи	<u>Дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet</u>
2. Керівник роботи	<u>Буравченко Костянтин Олегович, канд. техн. наук</u> (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	<u>10.12.2023 р.</u>
4. Мета та завдання випускної кваліфікаційної роботи:	<u>Метою розробки є дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet</u>
5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)	
<u>1. Призначення та область використання.</u>	<u>6. Наукова новизна.</u>
<u>2. Перегляд аналогічних існуючих систем.</u>	<u>7. Економічна ефективність розробленої програми.</u>
<u>3. Опис і обґрунтування проектних рішень.</u>	<u>8. Заходи з охорони праці та техніки безпеки.</u>
<u>4. Етапи програмування системи.</u>	<u>9. Висновки.</u>
<u>5. Впровадження системи в промислову експлуатацію</u>	
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<u>Наукова новизна</u>	<u>1 аркуш</u>
<u>Структурна схема системи</u>	<u>1 аркуш</u>
<u>Функціональна схема системи</u>	<u>1 аркуш</u>
<u>Діаграма процесів</u>	<u>1 аркуш</u>
<u>Блок-схема алгоритму роботи додатку</u>	<u>2 аркуша</u>
<u>Показники економічної ефективності</u>	<u>1 аркуш</u>

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	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

Мікіньов І.Ю. Дослідження та програмна реалізація системи багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet. 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 багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Метою розробки є дослідження та програмна реалізація системи багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Об'єктом дослідження є процес багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Предметом дослідження є методи багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

Результат роботи – програмна реалізація системи багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

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

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

Програму розроблено в середовищі RAD Studio Delphi.

Ключові слова: комп'ютерні науки, менеджера завантажень

ABSTRACT

Mikinov I.Yu. Research and software implementation of a multifunctional manager of downloading multimedia data from the Internet. 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 multi-functional manager system for downloading multimedia data from the Internet.

The goal of the development is the research and software implementation of a system of a multifunctional manager for downloading multimedia data from the Internet.

The object of the study is the process of a multi-functional manager for downloading multimedia data from the Internet.

The subject of the study is the methods of the multifunctional manager of downloading multimedia data from the Internet.

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

The result of the work is the software implementation of a multifunctional manager of downloading multimedia data from the Internet.

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

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

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

The program was developed in the RAD Studio Delphi environment.

Keywords: computer science, download manager

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи	26
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	36
3.4 Розробка діаграми процесів.....	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	41
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	41
4.2 Захист розробленого програмного забезпечення.....	54
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	57
6 НАУКОВА НОВИЗНА	59

					ВКРМ-122.23.0054.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Мікіньов І.Ю.</i>					М	1	100
<i>Перев.</i>	<i>Буравченко К.О.</i>							
Н.контр.	<i>Коваленко А.С.</i>					<i>ЦНТУ КН-22М-2</i>		
Затв.	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	60
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	60
7.2 Розрахунок трудомісткості розробки програмної продукції.....	62
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	64
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	69
7.5 Визначення собівартості розробки та ціни програмної продукції.....	73
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	76
7.7 Визначення експлуатаційних витрат.....	76
7.8 Визначення економічної ефективності програмної продукції.....	78
7.9 Висновок.....	80
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	81
8.1 Вступ.....	81
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	82
8.3 Аналіз умов праці на робочому місці ІТ-фахівця.....	83
8.4 Пропозиції щодо підвищення працездатності ІТ-фахівця.....	85
8.5 Пожежна безпека.....	87
8.6 Розрахункова частина	88
8.7 Висновки до розділу.....	90
9 ОСНОВНІ ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЛОМ	–	локальна обчислювальна мережа
ОП	–	оперативна пам'ять
ОС	–	операційна система
ПК	–	персональний комп'ютер
САЗ	–	система автоматизованого завантажування
ЦП	–	центральний процесор
DNS	–	сервер доменних імен
HTTP	–	HyperText Transfer Protocol – «протокол передачі гіпертексту»
HTTPS	–	Hypertext Transfer Protocol Secure
FTP	–	File Transfer Protocol – протокол передачі файлів
IP	–	Internet Protocol
LAN	–	локальна мережа
MAC	–	Media Access Control – управління доступом до носія
NAT	–	Network Address Translation
NTLM	–	NT LAN Manager – протокол мережної автентифікації
TCP	–	Transmission Control Protocol
UDP	–	User Datagram Protocol – протокол користувальницьких дейтаграм

ВСТУП

Актуальність теми. Як відомо, завантаження файлів спеціальними менеджерами завантаження (download-менеджерами), а не інтернет-браузерами, на практиці виявляється набагато більш зручним і ефективним. І справа отут не тільки в можливості поновлення завантаження, тим більше, що в деяких браузерах вона з недавнього часу підтримується, але й у більше швидкому завантаженні, що досягається завдяки багатопоточності й використанню ряду інших технологій. Тільки от вибрати підходящий "завантажувач" не дуже просто, адже на ринку пропонується чимало подібних рішень, причому з досить близькою на перший погляд функціональністю. Всі download-менеджери підтримують багатопоточне завантаження й докачку файлів після обриву зв'язку й автоматично підбирають оптимальну (залежно від типу з'єднань) швидкість завантаження. І хоча не всі протоколи й сервери підтримують поновлення завантаження (наприклад, воно неможливо на безкоштовних аккаунтах Rapidshare), download-менеджери, на відміну від інтернет-браузерів, забезпечують більше стійке з'єднання із сервером, а виходить, і трохи більшу швидкість завантаження. Менеджери завантажень можуть бути автоматично убудовані в популярні браузери, замінюючи стандартний модуль завантаження, стежать за буфером обміну, самостійно виявляючи посилання, і тому забезпечують швидке й надійне завантаження файлів з FTP– і HTTP-серверів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

- Огляд існуючих систем багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.
- Дослідження системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Програмна реалізація системи багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Об'єктом дослідження є процес багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Предметом дослідження є методи багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

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

– Удосконалено метод багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet.

– Розроблено вітчизняний продукт багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

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

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Убудовані функції Windows не передбачають можливість дозавантаження файлів – у підсумку у випадку обриву зв'язку інформацію доводиться завантажувати заново, затрачаючи на цей зайвий час і нерви й оплачуючи зайвий трафік. І навіть при нормальному з'єднанні, швидким такий варіант завантаження не назвеш, адже завантаження виробляється в один потік. Інша справа – спеціальні менеджери завантаження (download-менеджери), що вміють відновляти завантаження з того місця, де вона була перервана, і що розбивають завантажувальні файли на секції, що завантажуються одночасно, що дозволяє домогтися істотного збільшення швидкості завантаження.

1.2 Область застосування

Між собою менеджери завантаження розрізняються різноманітними нюансами. Зокрема, не всі забезпечують часткове завантаження ZIP-архівів, коректно розпізнають посилання з "файлосховищ", таких як Rapidshare, у безкоштовному режимі (як правило, на файлосховищах підтримка менеджерів завантаження передбачена тільки для VIP-клієнтів, всім іншим користувачам пропонуються завантажувати файли через інтернет-браузер) і дозволяють прямо (тобто просто додавши URL виду <http://www.youtube.com/watch?v=...>) завантажувати кліпи з онлайн-відеосервісів (YouTube, Google Video і ін.). Не всі "завантажувачі" дозволяють оцінити розмір файлу перед його завантаженням і вміють завантажувати файли із захищених FTP- і web-серверів (тобто за протоколами SFTP і HTTPS відповідно). А вже підтримка BitTorrent- і eDonkey-клієнтів взагалі реалізована лише в деяких менеджерах завантаження. Крім того,

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

у різних рішеннях відрізняється список підтримуваних інтернет-браузерів, та й сам принцип подібної інтеграції різний: десь досить просто включити відповідні прапорці в налаштуваннях, а десь буде потрібно попередньо скачати й установити відповідному браузеру плагін. Крім того, є й більше тонкі розходження, що стосуються самого процесу завантаження, – наприклад в одних рішеннях при виявленні активності браузера пріоритет по трафіку автоматично змінюється (а десь це прийде робити вручну), в інші можна призначати завантаженням пріоритети (що в частині продуктів не передбачене) і т.п.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

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

GetRight/GetRight Pro

GetRight – давно присутній на ринку й популярний завдяки зручності й надійній роботі менеджер завантаження, що сьогодні вже далеко не настільки привабливий для широкого кола користувачів, оскільки по ряду параметрів уступає аналогам і пропонується по досить високій ціні.

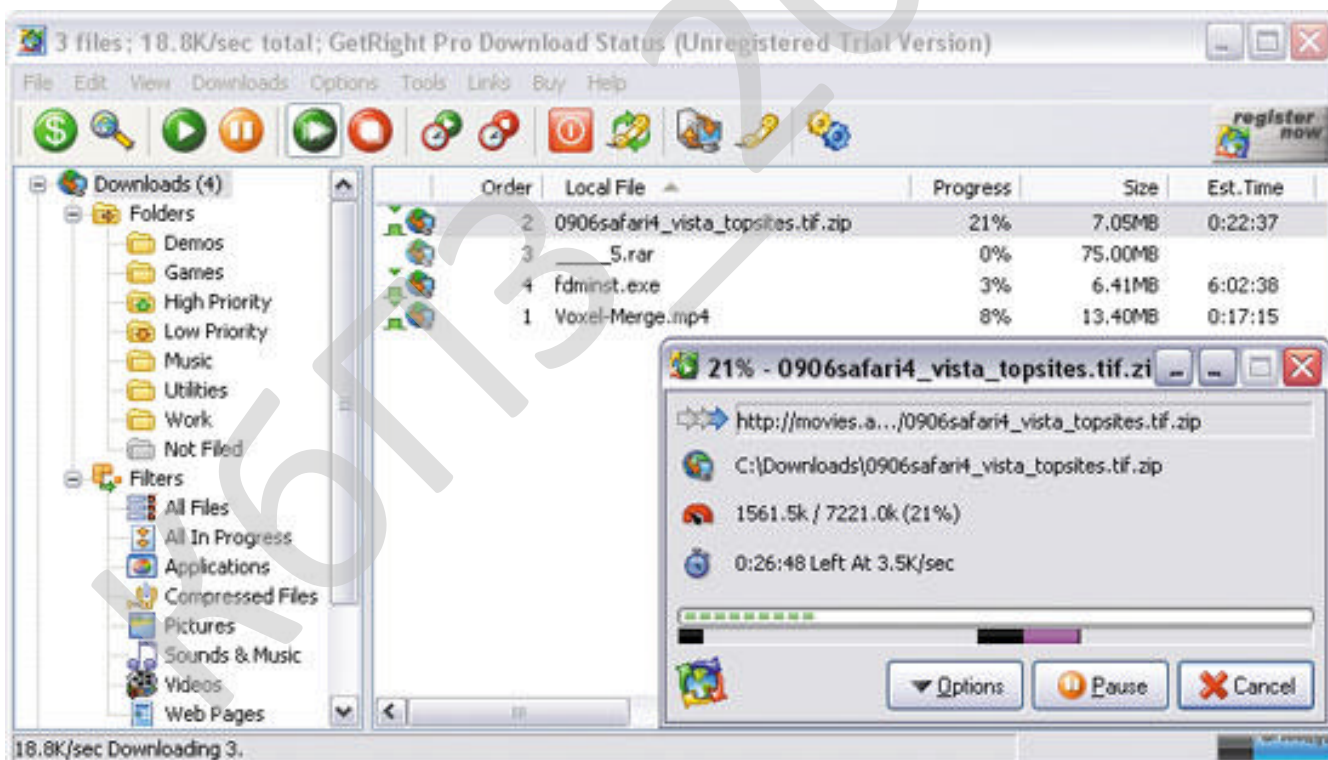


Рисунок 2.1 – Інтерфейс користувача GetRight

Та й подальша розробка цього рішення, по суті, уже не ведеться. Даний download-менеджер дозволяє завантажувати файли (у тому числі за розкладом) за протоколами HTTP, HTTPS, FTP і FTPS і може використовуватися для завантаження файлів з файлообмінних серверів, одержання нових музичних кліпів в Podcast/ RSS-канал і завантаження файлів за протоколом BitTorrent (є можливість одночасного одержання частини даних по FTP/HTTP, а частини – з BitTorrent з наступним «склеюванням» на льоту). Передбачено можливість автоматичного додавання отриманих мультимедіа файлів у плейлисти додатків MediaPlayer і iTunes. Однак утиліта не допоможе при завантаженні відео з відеохостингів і частковому завантаженні архівів. GetRight автоматично інтегрується в провідні браузері, причому вказати перелік браузерів, що цікавлять, можна при установці утиліти. Спеціальний убудований оглядач забезпечує перегляд структури папок FTP-серверів і HTTP-сайтів. Обмеження трафіку виробляється вручну шляхом установки максимально припустимої межі швидкості завантаження, при цьому дозволяється включити прапорець для зняття цього обмеження в моменти активності скринсейвера. Перед завантаженням можна перевірити адресу, що цікавить, на існування й уточнити розмір файлу. Кількість одночасно завантажуваних файлів і число потоків регулюються, а кращі «дзеркала» для завантаження нескладно знайти автоматично або вказати вручну. Залежно від типів завантажуваних файлів можуть розподілятися по різних папках, для папок дозволяється встановлювати пріоритети. По завершенні завантаження можливо автоматичне проведення оцінки цілісності отриманих файлів і перевірки їх на віруси. Програма представлена у двох версіях – базовій GetRight і розширеній GetRight Pro. В Pro-версії передбачена можливість роботи в режимі «клієнт-сервер» (для завантаження файлів одним комп'ютером у мережі, якому з інших просто передаються посилання) і в якості проксі-сервера (дозволяє іншим інтернет-утилітам використовувати можливості GetRight по дозавантаженні файлів). У ній також убудований функціонал для синхронізації вже завантажених файлів у випадку появи на серверах їхніх відновлень і завантаження файлів на

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

сервери, є скриптова мова (забезпечує розширені можливості керування завантаженням), можна лімітувати швидкість завантаження окремих файлів і ін.

Download Master

Download Master – багатофункціональний download-менеджер, що дозволяє завантажувати файли не тільки з FTP– і HTTP-серверів, але також з популярних відеосервісів (YouTube, Google Video, RuTube, Відео@mail.ru, Rambler Vision) і файлохочищ (включаючи Rapidshare) у безкоштовному режимі. У випадку ZIP-архівів їхній зміст може проглядатися перед завантаженням, передбачена також можливість завантаження тільки обраних з архіву файлів. Реалізовано пошук файлів, програм, ігор і музики в каталозі файлів TopDownloads.

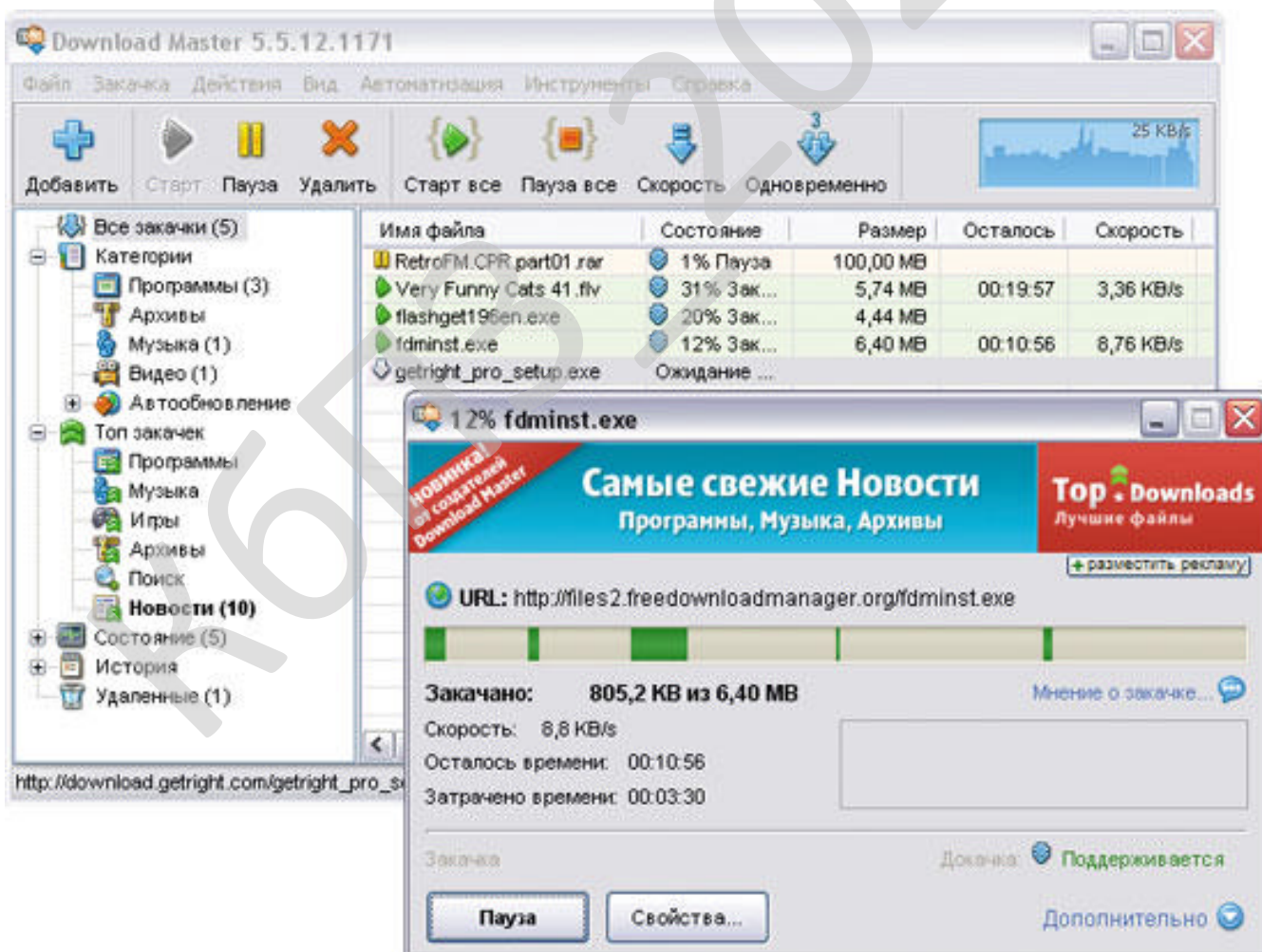


Рисунок 2.2 – Інтерфейс користувача Download Master

Download Master інтегрується в усі самі популярні браузері (для клонів ІЕ буде потрібно налаштування параметрів), а убудований в утиліту FTP Explorer забезпечує зручну навігацію по FTP-серверах. Утиліта вмє самостійно визначати типи файлів, що завантажуються, і розподіляє їх по категоріях (програми, музика, відео та ін.), а при зниженні швидкості автоматично перезапускає завантаження, що дозволяє уникнути простоїв. Завантаження ведеться в кілька потоків. При необхідності для певних сайтів число потоків нескладно обмежити й настроїти програму так, щоб завантажувальні з них файли зберігалися в певних папках і/або категоріях. За закінченням завантаження можливе відключення від Інтернету, перехід у сплячий режим, вимикання комп'ютера й т.п. Передбачене керування швидкістю завантаження й автоматичне її зниження у випадку, якщо користувач зайнявся інтернет-серфінгом. Можлива робота з розкладу, а також перевірка заколисаних файлів на відновлення й синхронізація (автооновлення) файлів на сервері й локальному ПК.

Програма представлена у двох версіях: стандартний Download Master і портативний Download Master Portable. Остання не вимагає установки й може бути записана на флешку або зовнішній диск, що забезпечує можливість завантаження файлів на будь-якому підключеному до Інтернету комп'ютері. У версії Portable немає інтеграції із браузерами й відключена можливість експорту/імпорту налаштувань, тому для застосування на власному комп'ютері більше зручною буде стандартна версія програми.

ReGet Deluxe

ReGet Deluxe – багатфункціональний менеджер завантажень, що дозволяє завантажувати файли (у тому числі за розкладом) не тільки зі звичайних FTP–і HTTP-серверів, але й із захищених файлових (SFTP) і веб (HTTPS)-серверів. Також утиліта вмє завантажувати мультимедіаконтент через поточкові протоколи (MMS, RTSP) і може використовуватися для завантаження з файлообмінних серверів (MySpace.com, iDrive.com і т.д.), однак вона не допоможе скачати флеш-відео з відеохостингів, зокрема з YouTube. Утиліта відображає

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		11

розмір завантажуючого файлу ще перед завантаженням, дозволяє переглядати ZIP-архіви й завантажувати їх частково й уміє автоматично перемикається на більше швидкі «дзеркала».

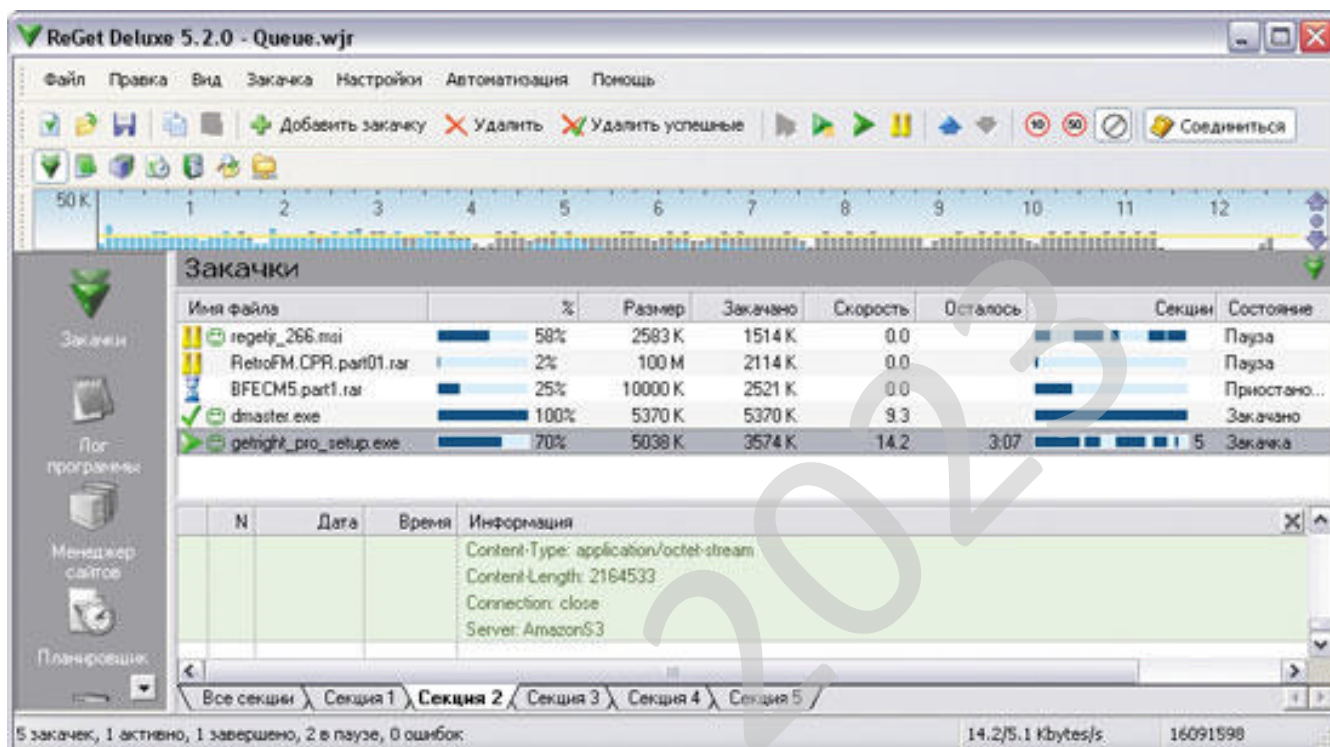


Рисунок 2.3 – Інтерфейс користувача ReGet Deluxe

ReGet Deluxe автоматично інтегрується в Microsoft Edge і ряд його клонів, інтеграція в частину браузерів здійснюється через плагіни сторонніх розроблювачів. Убудований FTP-Браузер спрощує пошук потрібних файлів на FTP-серверах. Завантажуються файли, що, можуть автоматично розкладатися по різних папках з обліком призначеної їм користувачем категорії, причому при використанні макросів допускається навіть автоматичне створення потрібних папок відповідно до розширень завантажених файлів, датою завантаження або ім'ям сервера, з якого був завантажений файл. З конкретного завантаження утиліта автоматично відключається від сервера й з'єднується з ним знову, даючи завантаженню стартовий поштовх. Одночасно може завантажуватися кілька файлів – їхня кількість (як і число потоків) залежить від типу з'єднання. У ході

завантаження утиліта зауважує активність браузера й автоматично знижує свій трафік при передачі браузером даних. По завершенні завантаження вона може автоматично розірвати інтернет-з'єднання, запустити зовнішній додаток, виключити комп'ютер і т.п.

Програма представлена у двох редакціях: ReGet Deluxe і ReGet Deluxe Personal, остання не має мультимовної підтримки й призначена для використання тільки на домашньому комп'ютері. Утиліта може працювати в одному із трьох режимів: спрощеному (у ньому працюють лише базові функції), розширеному й режимі експерта, перемикання між ними здійснюється за допомогою команд головного меню.

Free Download Manager

Free Download Manager – повноцінний і дуже зручний менеджер завантаження, що дозволяє завантажувати файли (у тому числі за розкладом) з FTP– і HTTP-серверів, захищених веб (HTTPS)-серверів, а також з популярних файлохранилищ і відеосервісів (YouTube, Google Video і ін.). Можливе одночасне розвантаження від декількох «дзеркал». При завантаженні відеоконтенту допускається автоматичне перетворення FLV-файлів в інші відеоформати за допомогою убудованого перекодувальника відео. Передбачено часткове завантаження ZIP-архівів, завантаження файлів через мережу BitTorrent і цілком сайтів для їхньої наступного перегляду в офлайн. Є також спеціальний Upload manager, що забезпечує завантаження файлів в онлайн-сховище WikiFortio (<http://www.wikifortio.com/>), де дозволяється до місяця зберігати файли розміром до 100 Мбайт.

Програма інтегрується в усі самі популярні браузери (причому без необхідності установки плагінів) і стежить за буфером обміну. Спеціальний оглядач сайтів дозволяє переглядати структуру папок сайтів, що спрощує завантаження потрібних даних. Обмеження трафіку виробляється вручну шляхом вибору одного із трьох передвстановлених режимів, можливо автоматичне зменшення завантаження каналу при виявленні активності браузера. Одночасно

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

можуть завантажуватися кілька файлів – кожний у кілька потоків, кількість одночасно завантажуваних файлів регулюється. Завантажувальні файли можуть розподілятися по різних папках з обліком зазначених користувачем категорій, і для конкретних завантажень допускається установка обмежень на використання каналу. Перед завантаженням будь-якого файлу його розмір нескладно уточнити. По завершенні завантаження утиліта може автоматично розірвати інтернет-з'єднання й виключити комп'ютер. Допускається керування завантаженнями у віддаленому режимі.

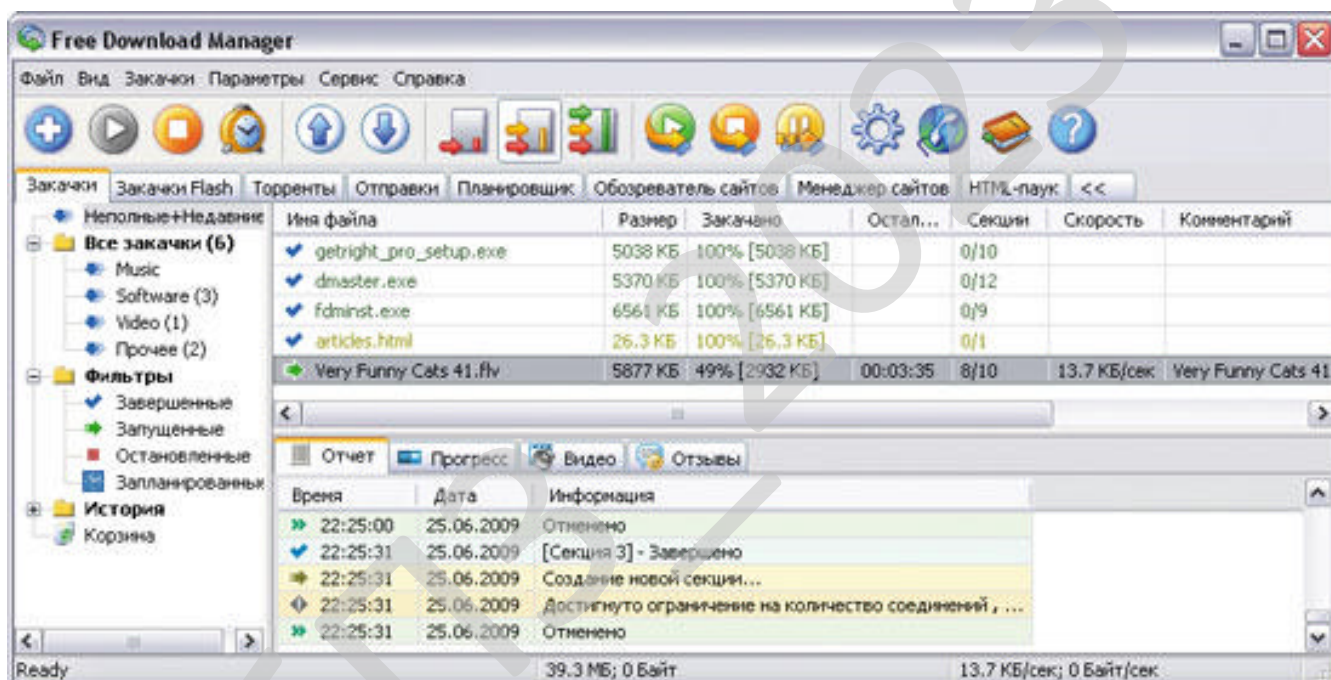


Рисунок 2.4 – Інтерфейс користувача Free Download Manager

Програма представлена у двох редакціях: стандартній Free Download Manager і полегшеній Free Download Manager Lite. В останній немає підтримки протоколу BitTorrent, а також відсутні вбудований відеоконвертор і менеджер завантаження файлів на сервери. В обох редакціях є функціонал для запису портативної версії, що не вимагає установки й може бути записана на флешку або зовнішній диск.

Orbit Downloader

Orbit Downloader – дуже простий у застосуванні, але досить функціональний і зручний менеджер завантаження, що відрізняється розширеною підтримкою соціальних сервісів. Він дозволяє завантажувати не тільки файли за протоколами FTP і HTTP, але й різного роду потоковий флеш-, відео- і аудіоконтент через протоколи RTSP/MMS/RTMP з MySpace, YouTube, Imeem, Pandora і інших подібних сервісів. Можливе завантаження файлів з файлообмінних сервісів (включаючи Rapidshare, MegaUpload, 4shared, Depositfiles та ін.), а також проведення завантаження за технологією Metalink, що дозволяє завантажувати окремі сегменти з мереж практично з будь-яким протоколом. Завантаження медіаконтенту із соціальних сервісів реалізовано в цій утиліті через убудований модуль Grab++, що працює тільки під Microsoft Edge.

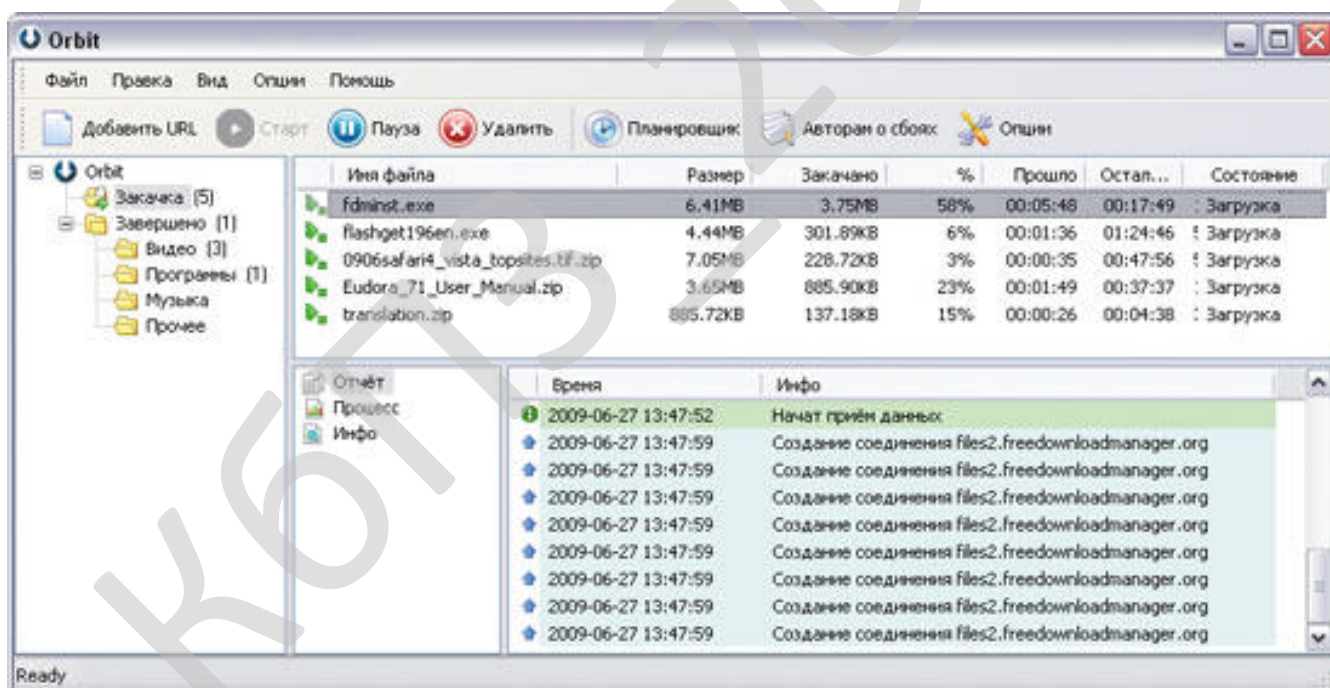


Рисунок 2.5 – Интерфейс користувача Orbit Downloader

Причому технологія завантаження принципово відрізняється від прийнятої в інших download-менеджерах: якщо звичайно менеджеріві

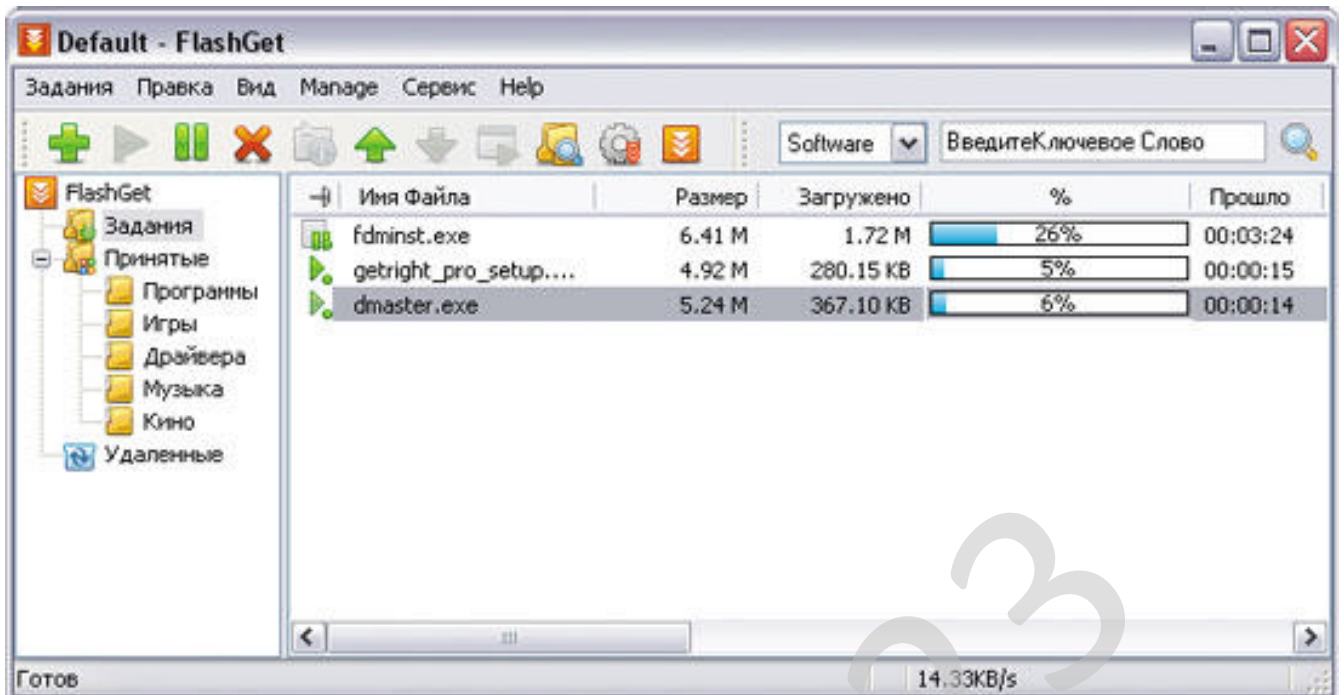


Рисунок 2.6 – Интерфейс користувача FlashGet

FlashGet уміє інтегруватися в популярні браузері, підтримка частини з яких вимагає установки плагінів. Убудовані FTP- і HTTP-оглядачі спрощують пошук потрібних файлів. Оптимальна швидкість з'єднання вказується в налаштуваннях шляхом установки типу з'єднання, можлива установка обмеження трафіку вручну вибором одного із трьох передвстановлених режимів. Завантажувальні файли можуть розподілятися по різних папках з обліком зазначених користувачем категорій. Передбачено автоматичний пошук «дзеркал» і здійснення завантаження з найшвидших і доступних серверів. Нескладно настроїти проведення завантаження в строго призначений час, а по його завершенні утиліта може перевірити файли на віруси й розірвати з'єднання.

ReGet Junior

ReGet Junior – найпростіший у застосуванні download-менеджер, орієнтований на починаючих користувачів. Утиліта має мінімального функціонала, і її привабливість полягає лише в тому, що розбиратися з налаштуваннями не буде потрібно в принципі. Вона дозволяє завантажувати

Downloader, оскільки в інших розглянутих download-менеджерах підтримка відповідних браузерів відсутній. При цьому потрібно мати на увазі, що завантаженням медіаконтенту із соціальних сервісів в Orbit Downloader ним скористатися не вдасться, оскільки вона працює тільки під Microsoft Edge.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

Delphi 10.4 Sydney

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

Основні можливості Delphi 10.4.1:

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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 швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

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

Істотне поліпшення Delphi Code Insight

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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 компонентів на одній

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім додаткової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

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

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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 багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Програмне забезпечення, що розробляється в даному магістерському проекті – універсальний менеджер завантажень файлів, робота якого спрямована на рішення трьох завдань: швидке завантаження, можливість перервати процес завантаження й наступне керування завантаженим матеріалом. Відмітною рисою додатка є зручний користувальницький інтерфейс і разюче висока швидкість завантаження. Чим обумовлені такі можливості? По-перше, оборотна увага на зовнішню оболонку. Робоче вікно програми організоване найкращим для користувача образом. Права половина вікна наочно відображає прогрес файлу, що завантажується, ліва дозволяє з легкістю орієнтуватися в списку завантажень, розсортованих по відповідних папках. Верхню частину робочого стола програми займає панель інструментів з інтуїтивно зрозумілими значками, що допомагає додати файл у список завантаження, видалити його, призупинити або відновити завантаження. Для кожного файлу, що завантажується, відкривається своє інформаційне вікно, що тримає користувача в курсі швидкості й обсягу завантажених даних.

Весь процес завантаження відбувається швидко за рахунок особливої технології поділу файлу на окремі потоки. Ці потоки завантажуються одночасно й у цілому зменшують загальний час завантаження. Якщо ви з якоїсь причини не можете чекати закінчення завантаження, програмне забезпечення, що розробляється в даному магістерському проекті, надає можливість призупинити процес і продовжити його в будь-який зручний час.

Останнім часом у нових версіях деяких браузерів (Microsoft Edge, Firefox, Google Chrome, Apple Safari, Mozilla, Opera) програмне забезпечення, що розробляється в даному магістерському проекті, використовується як

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

альтернатива стандартному завантажувальному клієнтові. Крім того, даний додаток можна використовувати для перегляду відео з популярних хостингів YouTube, Rambler Vision, й так далі. Строгий розподіл завантаженої інформації з тематичних папок стає можливим завдяки автоматичному розпізнаванню програмою безлічі різноманітних типів файлів. При запуску завантаження програмне забезпечення, що розробляється в даному магістерському проєкті, оперативно визначає потрібну папку для збереження завантажуючого файлу в автоматичному режимі.

Менеджер завантажень (download manager) – комп'ютерна програма, призначена для завантаження файлів з Інтернету або локальної мережі.

Раніше основними перевагами менеджерів завантаження перед вбудованими засобами браузерів вважалася можливість дозавантаження, завантаження в кілька потоків і запуск завдань за розкладом.

Специфіка додатка жадає від розроблювачів менеджера завантаження постійного розвитку й удосконалювання, тому що мир Інтернету постійно змінюється, а разом з ним змінюються й переваги користувачів. Якщо ще зовсім недавно для менеджера завантажень досить було забезпечувати підтримку Microsoft Edge, сьогодні відсутність інтеграції з альтернативними браузерами Mozilla, Opera та ін. розцінюється як великий недолік. Конкуруючи між собою, розроблювачі менеджерів завантаження постійно вбудовують у програми нові опції, тому тими можливостями, які сьогодні є для програми унікальними, завтра вже напевно обзаведуться й інші додатки.

Серед найбільш затребуваних опцій менеджерів завантаження можна відзначити: вбудований FTP-клієнт, можливість перегляду Zip-архівів перед завантаженням, одержання розміру файлів до початку завантаження, перевірка й відновлення архівів, установка пріоритетів для завантажень, робота з командним рядком, завантаження HTML-сторінок із зображеннями. Крім цього, недавно розроблювачі програм повідомили про можливість розширення їх можливостей за допомогою модулів, що підключаються – плагінів. Передбачається, що вони допоможуть реалізувати ті функції, які в цей час відсутні в сучасних менеджерах

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

завантажень, наприклад, завантаження файлів з файлообмінних систем на зразок Rapidshare.

Однак область застосування менеджерів завантажень має деякі обмеження:

– Не є вірним твердження у тому, що якщо технологія багатопоточного завантаження така ефективна, то варто максимально збільшити кількість секцій. Будь-яка технологія має свою межу, при досягненні якого, навпаки, починає сповільнювати роботу програми. Для 56 Kbps оптимальним є 4-5 потоків, для 512 Kbps – 20-25 і т.д. Але ніяк не 100. І не слід забувати, що на багатьох Web-серверах встановлене обмеження на кількість одночасних з'єднань (іноді 5-10, а де й 1). Якщо ж користувач порушує подібні правила, він автоматично відправляється в чорний список – у найкращому разі на пару днів, у гіршому – назавжди. Тому незначне прискорення завантаження обмеження доступу до Internet ресурсу не варте цього.

– Не є вірним твердження у тому, що один менеджер завантажень може працювати значно швидше, ніж інший. Всі сучасні менеджери завантаження використовують ті самі технології акселерації з'єднання, так що відмінності у швидкості їхньої роботи (найчастіше незначні) пояснюються якістю вашої телефонної лінії й каналу провайдеру, ступенем завантаженості Web-вузлів, з яких ви завантажувате файли, та ін.

– Не є вірним твердження у тому, що в довідковій системі менеджера завантаження написано, що він дозволяє прискорити процес завантаження до 500%. Не слід сприймати це буквально й сподіватися, що менеджер завантажень перетворить вашу застарілу телефонну лінію у виділений канал 256К. Тут мається на увазі технологія багатопоточного завантаження, що дозволяє більш ефективно використовувати Internet з'єднання. Згадайте, як повільно тягнуться файли за допомогою убудованих засобів браузера (нерідкі величини 1-2 Kbps при пропускній здатності dial-up в 4-5 Kbps). А встановивши самий простенький менеджер завантаження, легко домогтися граничної величини 5 Kbps. От вам і прискорення на 500%.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Убудоване в браузер, програмне забезпечення, що розробляється в даному магістерському проекті, дозволяє вибрати певні розширення файлів і відфільтрувати їх за ключевим словом. Завантаження файлів виробляється всього в кілька кліків, тому поставити відразу кілька десятків файлів на завантаження не важко буде. Убудований планувальник додатково заощадить час, роблячи програмне забезпечення, що розробляється в даному магістерському проекті, незамінною програмою з розряду "Must have".

Стандартний менеджер завантажувань має наступний набір функцій:

- Зручна організація завантажень, зберігання посилань до завантажених файлів.

- Створення списків завантажень у різних форматах.

- Завантаження файлів за розкладом.

- Перевірка завантажених файлів антивірусними програмами.

- Припинення завантаження файлу.

- Поновлення завантаження файлу з останнього місця його переривання (т.зв. «дозавантаження»).

- Завантаження файлу в кілька потоків, що дозволяє збільшити швидкість завантажування, якщо пропускна здатність клієнта вище частки пропускної здатності, виділеної сервером на одне з'єднання.

- Обмеження швидкості завантаження.

- Рекурсивне завантаження (створення локального дзеркала).

Висока швидкість завантаження досягається за рахунок застосування динамічної розбивки файлів на секції, які завантажуються одночасно: це підвищує швидкодію в 3-10 разів. Крім того, у сучасних менеджерів завантажень існують такі можливості, які дозволяють повноцінно працювати із програмою без відкриття головного вікна зі списком завантажень. Серед інших можливостей можна назвати автоматичне дозавантаження після обриву зв'язку, роботу по розкладу, підтримку різних варіантів вигляду інтерфейсу користувача, функції пошуку файлів і можливість роботи через HTTP і FTP проксі-сервера.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Ключові особливості й функції програми:

- можливість швидкого завантаження по протоколах HTTP, HTTPS, FTP;
- перехоплює посилання з буфера обміну;
- автоматично перейменовує файли з повторюваними назвами;
- проводить пошук по завантажених файлах;
- автоматично сортує завантажуваний матеріал по потрібних папках (наприклад, «музика», «відео», «програми», «ігри»);
- завантаження матеріалів з популярних відеохостингів;
- можливість перервати й продовжити процес завантаження.

3.2 Розробка структурної схеми

Система, яка розробляється, у ході виконання магістерського проектування, включає в себе наступні структурні блоки та функціональні можливості.

Простий і зручний у використанні:

- інтуїтивно зрозумілий інтерфейс із можливістю налаштування;
- підтримка списків закахувань із широкими можливостями сортування;
- плаваючий індикатор швидкості закахування, індикатори активних закахувань, керування активними закахуваннями;
- оригінальна технологія Активний Наочний Кошик, що дозволяє вам повноцінно працювати із програмою без відкриття головного вікна зі списком закахувань;
- підтримка модулів, що підключаються, (плагинів);
- підтримка шкір;
- Fast Update – перевірка доступності нових версій програми;
- багатомовність.

Повний набір необхідних функцій:

- динамічне багатопотокове закахування;

- докачування після обриву зв'язку з HTTP, HTTPS і FTP серверів;
- робота з розкладу, відключення ПК після завершення закачування;
- оптимальні налаштування для роботи з різними типами з'єднань (dial-up, ISDN, ADSL, LAN) і на різних швидкостях;
- FTP Explorer;
- потужний і зручний засіб керування категоріями закачувань;
- пошук і додавання дзеркал для закачування;
- можливість закачування відео (у тому числі й HD) з YouTube, Вконтакте й інших популярних відеосервісів;
- робота з ZIP архівами: можливість перегляду вмісту ZIP архівів перед закачуванням, можливість закачувати тільки обрані файли з архіву, можливість перевіряти ZIP архіви й відновлювати ушкоджені файли, а також можливість розпаковувати архіви;
- робота з RAR архівами: можливість перевіряти RAR архіви, можливість розпаковувати архіви;
- докладний лог-файл на кожне завантаження;
- історія закачувань;
- менеджер сайтів для керування паролями й папками для збереження;
- керування швидкістю закачування, автоматичний режим для комфортної роботи із броузером;
- можливість установки пріоритетів для закачувань;
- регульований режим для динамічного керування швидкістю закачування;
- розпізнавання контенту при роботі через HTTP;
- робота з командним рядком;
- дзвонилка для dial-up з'єднань;
- пошук файлів, програм, ігор і музики в каталозі файлів TopDownloads;
- можливість синхронізації (автооновлення) файлів на сервері й локальному ПК;

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- перевірка заколисаних файлів на відновлення;
- можливість послухати/подивитися музичні й відео файли в процесі закачування. Автоматичне одержання інформації про MP3 файлах при старті закачування;
- можливість закачувати html-сторінки разом з картинками;
- Smart Pause. Можливість тимчасової зупинки закачуємих файлів з наступним їхнім стартом у тій же стані;
- широкі можливості налаштування параметрів з'єднання, HTTP, HTTPS і FTP протоколів;
- робота через HTTP і FTP проксі-сервера, підтримка NTLM і NTLM-проксі автентифікації;
- можливість звірення MD5 суми заколисаного файлу.

Найвища швидкість закачування:

- інтелектуальна багатосекційність. Він динамічно розбиває файл на секції й завантажує їх одночасно. Це збільшує швидкість закачування в 3-10 разів;
- при зниженні швидкості автоматично перезапускає закачування, що дозволяє уникнути простоїв.

Широкі можливості інтеграції:

- моніторинг буфера обміну;
- інтеграція в броузери Microsoft Microsoft Edge 4.0 і вище, Firefox, Google Chrome, Mozilla, Opera 4.0 і вище, Netscape Communicator 6.0 і вище, і ін.;
- інтеграція з антивірусними програмами;
- DM Bar – панель інструментів для Microsoft Edge і Mozilla Firefox .

Менеджери завантаження значно полегшують процес завантаження файлів із серверів в інтернеті на локальний комп'ютер. Користувач, відвідуючи різні сайти, може додати в менеджер посилання на файли, які він хотів би скачати й запустити їхнє завантаження після того, як серфінг закінчений. Таким чином, завантаження файлів за допомогою менеджера може бути виконана в той час, коли користувач не працює в інтернеті. Багато менеджерів завантаження

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

дозволяють указати час, коли необхідно скачати файли, що буває зручно при використанні діал-ап доступу в інтернет: список файлів для завантаження можна сформуванати вдень і запланувати завантаження на нічний час, коли діє більш дешевий тариф на послуги доступу в мережу, або більш висока швидкість доступу, у зв'язку з тим, що на так багато користувачів використовують Інтернет вночі. У цьому випадку, менеджер сам додзвониться до провайдеру, скачає файли й виключить комп'ютер. Крім цього, менеджери можуть дозавантажувати файли, тобто, якщо завантаження було перервано й файл не був завантажений повністю, то менеджер після повторного з'єднання із сервером продовжить завантаження файлу з того місця, де була перервана завантаження, а не буде завантажувати весь файл заново. На цьому список можливостей менеджерів завантаження не закінчується, вони можуть інтегруватися із браузером, автоматично знижувати швидкість завантаження, коли користувач працює в інтернеті, зберігати облікові дані для доступу до певних серверів і так далі.

Структурна схема системи зображена на рисунку 3.1. З неї ми бачимо, що система представляє собою взаємодію наступних структурних блоків:

1. Інтерфейс користувача головного вікна програми програмного забезпечення багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

2. Вікно статусів, яке включає в себе:

– Перелік усіх завантажень.

– Категорії файлів, які завантажуються (програми, архіви, музика, відео).

– Топ завантажень (програми, архіви, музика, відео, пошук).

– Новини.

– Стан (завантаження, чекання завантаження, заплановано, помилки, пауза, завантажено).

– Історія.

– Видаленні завантаження.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

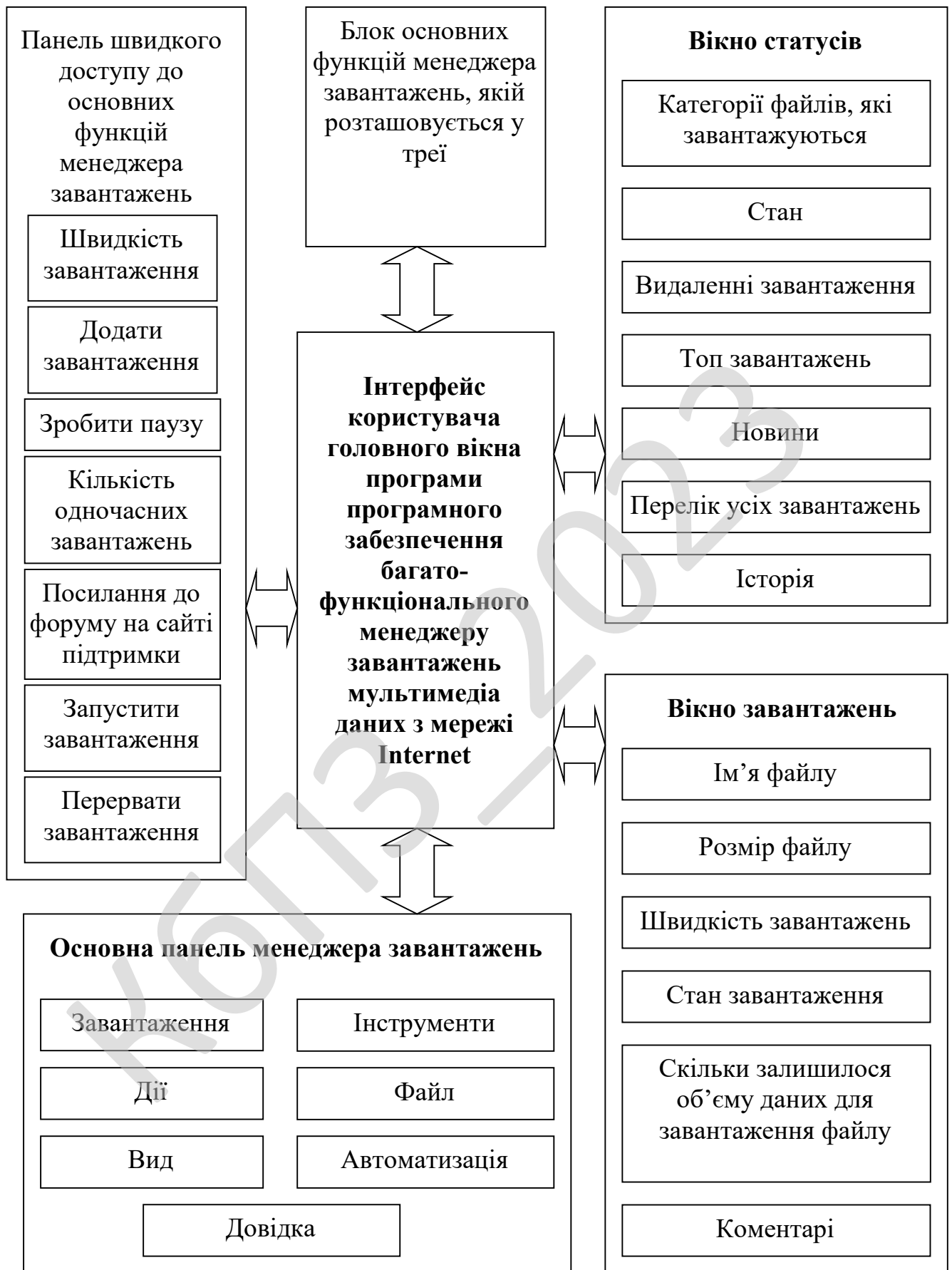


Рисунок 3.1 – Структурна схема системи

3. Панель швидкого доступу до основних функцій менеджера завантажень, яке включає в себе наступні функції:

- Додати завантаження.
- Запустити завантаження.
- Зробити паузу.
- Перервати завантаження.
- Швидкість завантаження.
- Кількість одночасних завантажень.
- Посилання до форуму на сайті підтримки.

4. Вікно завантажень:

- Ім'я файлу.
- Розмір файлу.
- Скільки залишилося об'єму даних для завантаження файлу.
- Швидкість завантажень.
- Коментарі.
- Стан завантаження.

5. Блок основних функцій менеджера завантажень, якій розташовується у треї.

6. Основна панель менеджера завантажень:

- Файл (топ завантажень, імпорт завантажень, імпорт, експорт, вихід).
- Завантаження (додати завантаження, додати групу завантажень, перевірити оновлення, пауза, видалити, видалити разом з файлом, запланувати, перезавантажити заново, копіювати URL, відкрити файл, відкрити папку, скопіювати файл, перемістити файл, меню Windows, робота з архівом, коментарі, знайти дзеркала, додати в менеджер сайтів, властивості).

– Автоматизація (стартувати усі завантаження при запуску програми, стартувати усі завантаження при появі інтернету, стартувати усі завантаження по часу, відновити зв'язок при обриві, відключитися від інтернету після завершення завантажень, перевірити завантажені файли на віруси).

- Інструменти (пошук, історія, менеджер сайтів, розклад, плагіни,

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

налаштування: загальні, з'єднання, завантаження, проксі, автоматизація, менеджер сайтів, розклад, інтерфейс, інші, плагіни).

– Довідка (зміст, домашня сторінка, он-лайн підтримка, повідомити про помилку, форум, перевірити оновлення, про програму).

– Дії (стартувати все, призупинити все, тимчасова зупинка завантажуваних, видалити все, знайти, знайти далі, швидкість).

– Вид (налаштування кнопок, сортування списку, список завантажень, звук, категорії, лог завантажень, плаваюче вікно, скіни, мова інтерфейсу: українська, англійська).

3.3 Розробка функціональної схеми

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

Функціонально система складається з наступних блоків:

- блок керування категоріями завантажень;
- блок керування швидкістю завантаження, автоматичний режим для комфортної роботи із браузером;
- блок можливість установки пріоритетів для завантажень;
- регульований режим для динамічного керування швидкістю завантаження;
- блок розпізнавання контенту при роботі через HTTP;
- блок роботи з командним рядком;
- блок синхронізації (автооновлення) файлів на сервері й локальному ПК;
- блок можливості послухати/подивитися музичні й відео файли в процесі завантаження. Автоматичне одержання інформації про MP3 файли при старті завантаження;
- блок можливості завантажувати html-сторінки разом з картинками;
- блок налаштування параметрів з'єднання, HTTP, HTTPS і FTP протоколів;

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

- блок роботи через HTTP і FTP проксі-сервера, підтримка NTLM і NTLM-проксі автентифікації;
- блок звірення MD5 суми завантаженого файлу.
- блок історії завантажень;
- блок перевірки завантажених файлів на відновлення;
- блок динамічного багатопотокового завантаження;
- блок робота з розкладом;
- блок дозавантаження після обриву зв'язку з HTTP, HTTPS і FTP серверів;
- блок відключення ПК після завершення завантаження;
- блок оптимальних налаштувань для роботи з різними типами з'єднань (dial-up, ISDN, ADSL, LAN) і на різних швидкостях;
- блок тимчасової зупинки завантажуваних файлів з наступним їхнім стартом у тому же стані;
- блок пошуку і додавання дзеркал для завантаження;
- блок завантаження відео з відеосервісів;
- блок роботи з ZIP архівами: можливість перегляду вмісту ZIP архівів перед завантаженням, можливість завантажувати тільки обрані файли з архіву, можливість перевіряти ZIP архіви й відновлювати ушкоджені файли, а також можливість розпаковувати архіви;
- блок роботи з RAR архівами: можливість перевіряти RAR архіви, можливість розпаковувати архіви;
- докладний лог-файл на кожне завантаження;
- блок менеджера сайтів для керування паролями й папками для збереження;

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

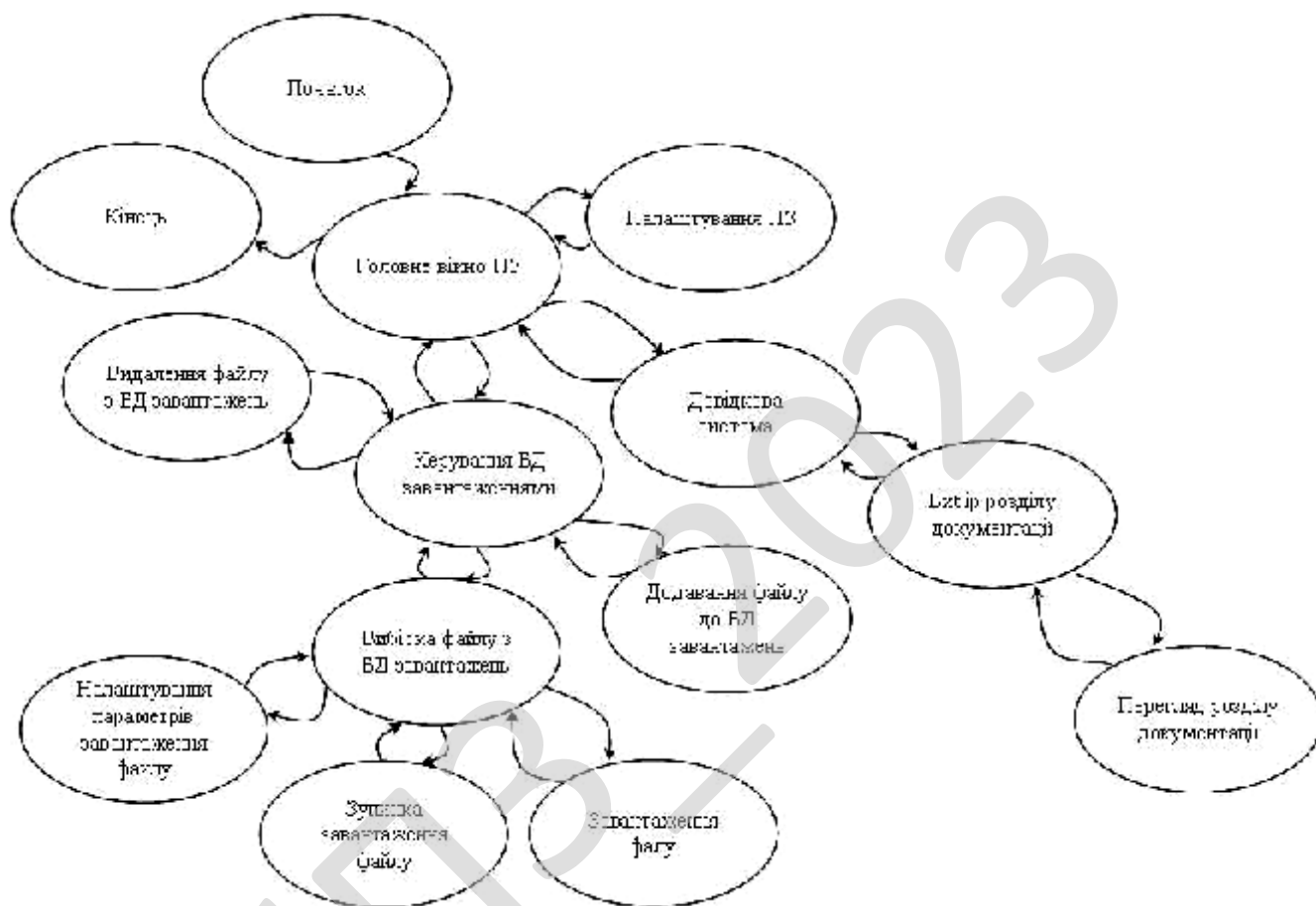


Рисунок 3.3 – Діаграма взаємодії процесів

Після початку роботи ПЗ ми потрапляємо до головного вікно ПЗ звідки можемо потрапити до налаштування ПЗ, довідкової системи з можливістю обрати розділ документації та провести перегляд розділу документації чи через керування БД завантаженнями за допомогою видалення файлу з БД завантажень та додавання файлу до БД завантажень потрапити до вибірки файлу з БД завантажень.

Далі можливо потрапити до налаштування параметрів завантаження файлу, зупинки завантаження файлу, завантаження файлу.

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

КБГПЗ - 2023

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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:

- Виведення головного вікна ПЗ.
- Корегувати чергу завантажень (запит).
- Редагування черги завантажень.
- Запит налаштування?.
- Налаштування параметрів ПЗ.
- Завантаження даних (запит).
- Введення URL адреси.
- Введення локального шляху збереження даних.
- Обрання пріоритету завантаження.
- Обрання параметрів завантаження.
- Підпрограма завантаження даних.
- Виведення вікна завантаження.
- Дані завантажено (запит).
- Збереження даних.
- Повідомлення WM_CLOSE (запит).

На рисунку 4.2 наведено блок-схему підпрограми завантаження даних. Її робота складається з виконання наступних кроків:

- URL валідний (запит).
- Запит розміру файлу.
- Виведення розміру файлу.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

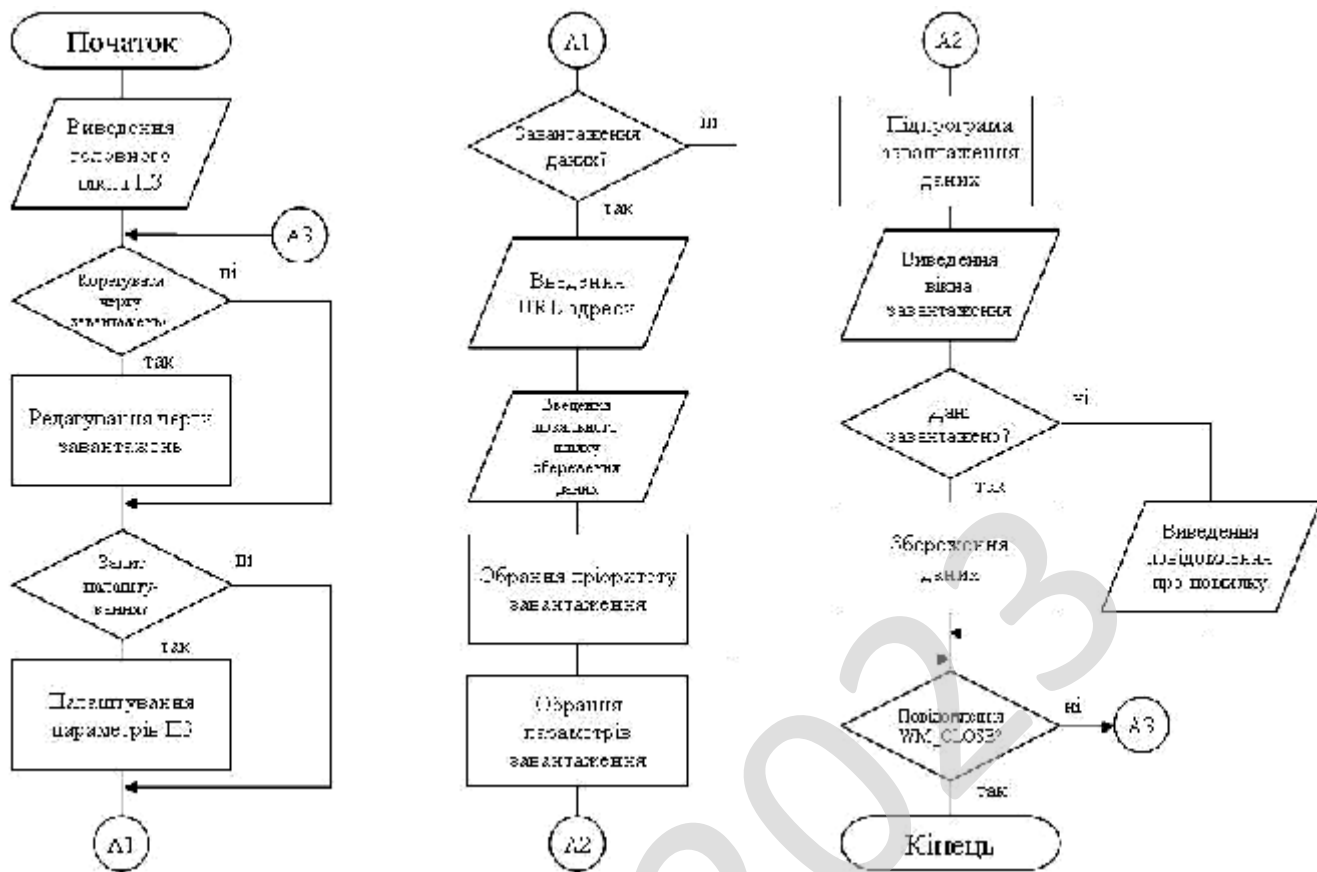


Рисунок 4.1 – Блок-схема основної програми

- Формування файлу завантаження.
- Завантаження блоків файлу.
- Виведення поточної швидкості з'єднання.
- Обчислення приблизного часу завантаження.
- Виведення приблизного часу завантаження.
- Додавання блоків до файлу завантаження.
- Виведення інтервальних даних.
- Завантаження завершено (запит).
- Завантаження пройшло успішно (запит).
- Остаточне збереження файлу на диску у вказаній директорії.
- Виведення повідомлення про завершення файлу.
- Виведення тривалості завантаження.
- Виведення інформації про файл та процес завантаження.

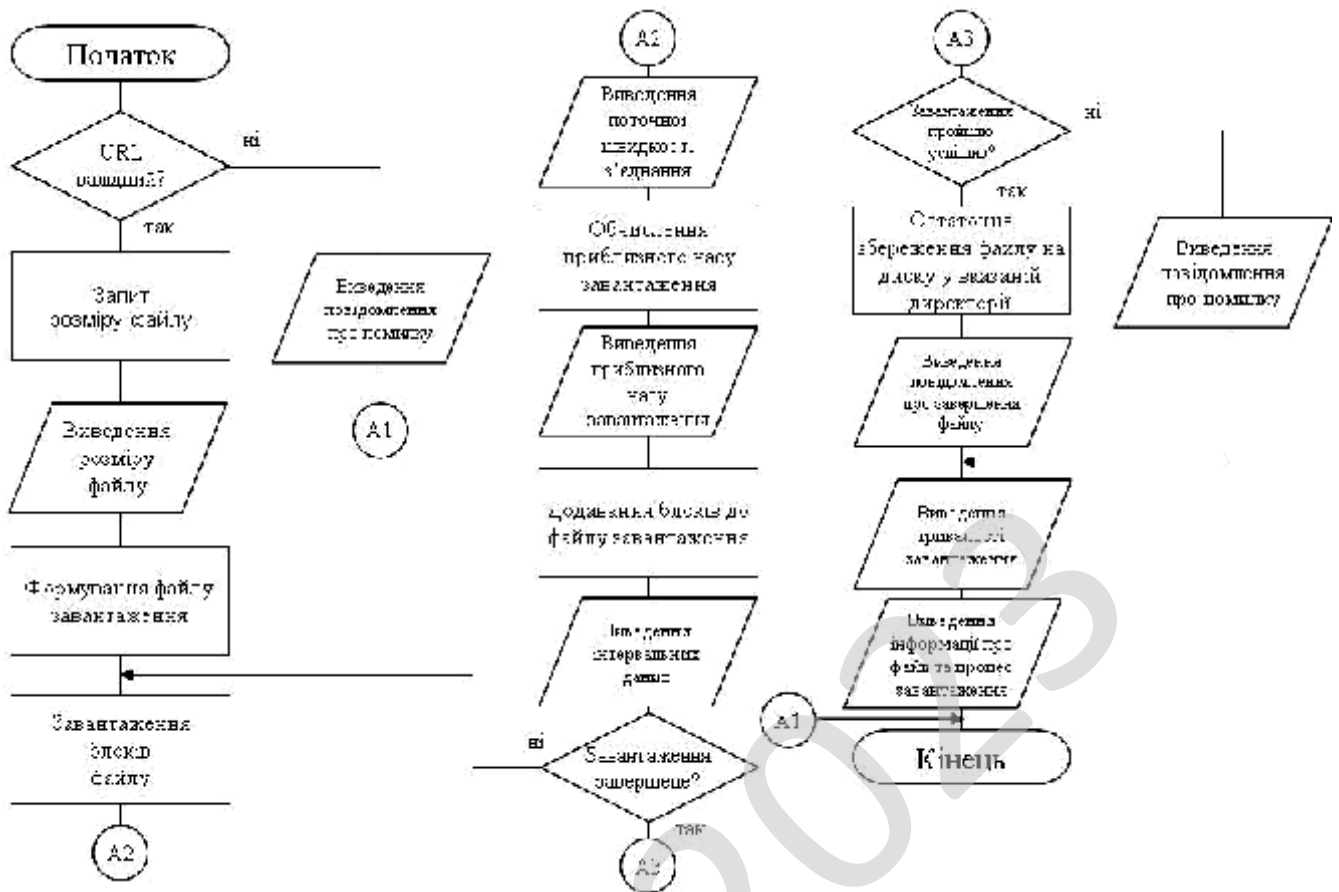


Рисунок 4.2 – Блок-схема підпрограми

Опис алгоритмів функціонування системи

Перевірка існування зазначеної URL.

{... Тут іде заголовок файлу й визначення форми TForm1 і її екземпляра Form1}
 {У форму потрібно помістити кнопку TButton і одне поле TEdit. При натисканні на кнопку викликається оброблювач події OnClick – Button1Click. Перед цим в TEdit потрібно ввести адресу URL.}

```

procedure Button1Click (Sender: TObject);
begin
    {Намагаємося одержати заголовок}
    NMHTTP1.Head(Edit1.Text);
    {Якщо URL невірний, то тут вискочить помилка}
end;
  
```

Одночасне зкачування зазначених URL у заданий каталог.

```

// Тут іде заголовок файлу й визначення форми TForm1 і її екземпляра Form1
// Опис класу окремого процесу
type
  
```

```

THTTPThread = class (TThread)
private
{Для кожного процесу - створюємо свій компонент TNMHTTP}
    FHTTP: TNMHTTP;
protected
// Execute викликається при запуску процесу;
    override - заміняємо
// існуючу процедуру базового класу TThread
    procedure Execute;
    override;
// Do Work - створена нами функція, виконання якої синхронізується в Execute
    procedure Do Work;
public
    // URL - створена нами рядок, що вказує процесу, який URL йому потрібно
    скачати
    URL: string;
    end;
// У форму потрібно помістити три кнопки TButton, одне поле TEdit і один список
// TListBox. При натисканні на кнопку Button1 викликається оброблювач події
// OnClick - Button1Click. Перед цим в TEdit потрібно ввести шлях до каталогу, в
// якому будуть зберігатися закачанні файли, а ListBox1 потрібно заповнити списком
// URL-ів для закачування (за допомогою кнопок Add (Button2) і Delete (Button3)).

    procedure TForm1. Button3Click (Sender: TObject);
    begin
{Видалення виділеного URL зі списку}
        if ListBox1. ItemIndex >= 0 then
            ListBox1. Items. Delete (ListBox1. ItemIndex);
        end;
    procedure TForm1. Button2Click (Sender: TObject);
        var s: string;
        begin
{Додавання URL у список}
            s := InputBox ('Додати', 'Уведіть URL: ', '');
            if s <> '' then
                ListBox1. Items. Add (s);
            end;
        procedure TForm1. Button1Click (Sender: TObject);
            var i: Integer;
            begin
{Перевірка на існування каталогу}
                if Length (Edit1. Text) > 0 then

```

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

```

        if not DirectoryExists (Edit1. Text) then
            Mkdir (Edit1. Text);
{Далі йде створення для кожного URL у списку свого процесу}
        for i := 0 to ListBox1. Items. Count-1 do
            begin
                with THTTPThread. Create (True) do
                    begin
{Створюємо припинене завдання, вказуємо їй її URL і запускаємо її}
                        URL := ListBox1. Items[i];
                        Resume;
                    end;
                end;
            end;
end;
// Оператори процесу THTTPThread
procedure THTTPThread. Execute;
begin
// Робимо так, щоб кожний процес виконувався одночасно з іншими (синхронізація)
Synchronize (Do Work);
end;
procedure THTTPThread. Do Work;
var i: Integer;
begin
{Створюємо компонент TNMHTTP}
    FHTTP := TNMHTTP. Create (Form1);
{Результат треба записувати у файли}
    FHTTP. InputFileMode := True;
{Підбираємо імена для файлів}
    i := 1;
    while FileExists (Form1. Edit1. Text+'\page'+IntToStr (i)+'. htm') do
        Inc (i);
{Вказуємо, у які саме файли класти результат}
    FHTTP. Body := Form1. Edit1. Text+'\body'+IntToStr (i)+'. htm';
    FHTTP. Header := Form1. Edit1. Text+'\header'+IntToStr (i)+'. txt';
{Намагаємося надіслати запит}
    FHTTP. Get (URL);
{Перед завершенням процесу не забуваємо звільнити пам'ять з-під компонента}
    FHTTP. Free;
end;

```

						ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата			45

Менеджер це практично тільки GUI, що одержить у своє розпорядження інтерфейс завантажувача. Сам завантажувач буде створюватися/знищуватися усередині системи автоматизованого завантажування САЗ, назовні будуть виведені тільки її методи Пауза/Продовжити/Скасувати + статистика й лог. От цей інтерфейс:

```
IRegionTilesDownload = interface ['{4BF3D0E8-3971-4EDC-97F3-44461D6A54FD}']
procedure SaveSession (ASessionFileName: string);
procedure Abort;
function GetPaused: Boolean;
procedure SetPaused (APaused: Boolean);
property Paused: Boolean read GetPaused write SetPaused;
function GetFinished: Boolean;
property Finished: Boolean read GetFinished;
function GetTotalInRegion: Int64;
property TotalInRegion: Int64 read GetTotalInRegion;
function GetDownloaded: Int64;
property Downloaded: Int64 read GetDownloaded;
function GetProcessed: Int64;
property Processed: Int64 read GetProcessed;
function GetDownloadSize: Double;
property DownloadSize: Double read GetDownloadSize;
function GetElapsedTime: TDateTime;
property ElapsedTime: TDateTime read GetElapsedTime;
function GetStartTime: TDateTime;
property StartTime: TDateTime read GetStartTime;
function GetZoom: Byte;
property Zoom: Byte read GetZoom;
end;
```

Інтерфейс лога (повідомлення які зараз виводяться в Мемо):

```
ILogForTaskThread = interface (ILogSimple) ['{8583BE 82-1239-4E0 F-9B96-D2810232282A}']
function GetLastMessages (AMaxRowCount: Cardinal; var ALastId: Cardinal;
out AcntLines: Cardinal): WideString; safecall;
end;
```

Опишемо загальний інтерфейс (ISASCommonAPI) з парою методів, щоб можна було стартувати збережені сесії з менеджера (автоматично, при перезапусках САЗ), і можливо, управляти видимістю головного вікна САЗ.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46


```

Presto/2.1.1'#13#10+'Accept:      text/html,      application/xml;      q=0.9,
application/xhtml+xml, image/png, image/jpeg, image/gif, image/ x-xbitmap, */*;
q=0.1'#13#10+' Accept-Language: ua-UA, ua; q=0.9, en; q=0.8'#13#10+' Accept-
Charset: iso-8859-1, utf-8, utf-16, *; q=0.1'#13#10+' Accept-Encoding: deflate,
identity, *; q=0'#13#10+'Connection: Close, TE'#13#10+'TE: deflate, chunked,
identity, trailers'#13#10#13#10);

    send (Sock, Send_buf, strlen (Send_buf), 0);
    Tmp:='';
    ZeroMemory (@buf, 1024);
    RcvLen:=recv (Sock, buf, 1024, 0);
    while RcvLen > 0 do
    begin
    Tmp := Tmp + Copy (buf, 0, RcvLen);
    RcvLen := recv (sock, buf, 1024, 0);
    end;
    log (tmp, clRed);
    StrPCopy (Recv_buf, tmp);
    closesocket (Sock);
    end;
    end;

```

Завантаження файлу з Інтернет можлива декількома шляхами.

За допомогою компонента TNMHTTP. Даний компонент дає можливість працювати по протоколі HTTP.

Зокрема, метод NMHTTP. Get дозволяє виконувати стандартний запит GET протоколу HTTP для завантаження інформації. Крім методу GET підтримуються POST, PUT, DELETE, OPTIONS.

За допомогою компонента TNMFTP – даний компонент дозволяє працювати по протоколі FTP.

TIdFTP і TIdHTTP – компоненти Indy для роботи із протоколу FTP і HTTP відповідно.

Реалізація протоколу FTP і HTTP "вручну" шляхом устанавлення з'єднання із сервером через TClientSocket або TIdTCPClient. Даний метод складніше в реалізації, тому що прийде реалізовувати роботу з обраного протоколу згідно RFC, проводити обробку помилок і т.п.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Використання API функцій бібліотеки wininet. Dll (найбільш уживаної функції InternetOpen, InternetOpenURL, InternetReadFile / InternetWriteFile і InternetCloseHandle).

З описаних методів останній має перевага, тому що його реалізація досить проста й немає потреби замислюватися про тонкості (наприклад, про проксі-сервер і його налаштування). Крім того, в wininet. Dll утримується кілька десятків функцій для роботи з FTP, дозвону до провайдеру.

Завантаження файлу за допомогою TIdHTTP.

Компонент бібліотеки Indy TIdHTTP містить ряд властивостей, які управляють роботою компонента. Розглянемо найцікавіші Властивості:

– HandleRedirects – Якщо дана властивість дорівнює True, те компонент обробляє коди протоколу HTTP (переадресація).

– Port – Порт, по якому виробляється з'єднання із сервером, за замовчуванням 80.

– ProtocolVersion – Версія протоколу. Можливий один із двох варіантів – pv1_0 для HTTP 1.0, pv1_1 для HTTP 1.1 (установлено за замовчуванням).

– RedirectMaximum – Максимальна кількість переадресацій, що може бути виконано.

– Має сенс при HandleRedirects = True, за замовчуванням дорівнює 15. Призначення даного досить очевидно – може виявитися, що при запиті ресурсів відбудеться редирект на x1, звідти – на x2 і так далі нескінченно (у принципі, велика ймовірність, що ланцюжок редиректів може утворити кільце).

– При досягненні заданої граничної кількості редиректів буде викликаний оброблювач Do Request, і Якщо він не обробить дану ситуацію, то виникне виняткова ситуація EIdProtocolReplyError. На мій погляд значення 15 велике і я рекомендую використовувати значення порядку 5.

– Request – Параметри в заголовку HTTP запиту.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

– UseNagle – Використання алгоритму Наггла (Nagle algorithm) при обміні. За замовчуванням використання алгоритму Наггла включене й виключати його не рекомендується.

Методи:

– Do Request – Виконання запиту й прийом відповіді.

– Get – Виконання методу Get.

Нижче наведено короткий опис основних властивостей, методів і подій компонента TNMHTTP.

Властивості:

– Body – рядок, що містить або шлях до файлу, у який буде записане тіло http-документа (Якщо властивість InputFileMode дорівнює True), або безпосередньо саме тіло (Якщо властивість InputFileMode дорівнює False). Тип: string.

– Header – рядок, що містить або шлях до файлу, у який буде записаний заголовок http-документа (Якщо властивість InputFileMode дорівнює True), або безпосередньо сам заголовок (Якщо властивість InputFileMode дорівнює False). Тип: string.

– HeaderInfo – структура, що містить різну інформацію про http-документ. Тип: THeaderInfo.

– InputFileMode – тип запису результату. Значення True – запис у файли, зазначені у властивостях Body і Header, False – запис у самі ці властивості. Тип: Boolean.

– OutputFileMode – тип відсилаємих даних (методами Put, Post і Trace). Значення True – дані для відправлення втримуються у файлах, зазначених при виклику цих методів, а False – у самих аргументах цих методів. Тип: Boolean.

Далі деякі властивості, успадковані від TPowerSock:

– BytesRecvd, BytesSent, BytesTotal – кількість відправлена, прийнята й загальна кількість байтів відповідно. Тип: LongInt.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

утримуватися шлях до файлу, що містить потрібні дані. Delete (URL: string) – надсилає запит на видалення http-документа (з адресою URL).

Далі деякі методи, успадковані від TPowerSock:

- Abort і Cancel – переривають з'єднання й обмін даними.
- Disconnect – від'єднання від HTTP-сервера.

Події:

- OnAuthenticationNeeded – виникає, коли сервер вимагає вказівки імені користувача й пароля. В оброблювачі цієї події (Якщо воно виникне) Ви повинні відповісти серверу, заповнивши потрібними значеннями відповідні змінні. Примітка: Перед установленням з'єднання можна відразу заповнити поля UserID і Password у властивості HeaderInfo.

- OnAboutToSend – виникає, коли компонент TNMHTTP збирається відправляти дані (запит). В оброблювачі цієї події можна заповнити додатковою інформацією властивість SendHeader.

- OnFailure – виникає, коли поточна операція завершилася невдало, тобто відбулася помилка.

- OnRedirect – виникає, сервер переадресував посилання із зазначеної URL на інше посилання. Установивши параметр handled у значення True можна заборонити переадресацію й зупинитися на запитаній URL. Значення за замовчуванням – False.

- OnSuccess – виникає, коли поточна операція завершилася успішно, тобто запит був виконаний без помилок.

Далі деякі методи, успадковані від TPowerSock:

- OnConnect – виникає, коли з'єднання із сервером успішно встановлено.
- OnDisconnect – виникає, коли з'єднання із сервером завершено.
- OnConnectionFailed – виникає, коли з'єднання із сервером установити не вдалося.
- OnError – виникає, коли остання операція була завершена з помилкою.

- OnHostResolved – виникає, коли від DNS отримана IP-адреса зазначеного хосту.
- OnInvalidHost – виникає, коли DNS повернув помилку при спробі визначити IP-адресу зазначеного хосту.
- OnPacketRecvд – виникає, коли значення властивостей BytesRecvд та BytesTotal змінені, тобто була прийнята нова порція даних від сервера.
- OnPacketSent – виникає, коли значення властивостей BytesSent та BytesTotal змінені, тобто була відправлена нова порція даних на сервер.
- OnStatus – виникає, коли статус компонента був змінений (для відновлення візуального оповіщення користувача).

Якщо використовуєте проксі з'єднання

Додавання від 30.10.2023 16:43:

```

procedure InitHttp (Port: Word; IP, Login, Password: String; Http: TidHTTP);
begin
    Http. Request. ProxyPort := Port;
    Http. Request. ProxyServer := IP;
    Http. Request. ProxyUsername := Login+'5';
    Http. Request. ProxyPassword := Password;
end;
function LoadPrice (const URL, FileName: String; Http: TidHTTP): Boolean;
var tmpStream: TFileStream;
begin
    tmpStream := TFileStream. Create (FileName, fmCreate);
    tryHttp. Get (URL, tmpStream);
    tmpStream. Free;
    exceptOn E: Exception do
    begin
        if E.ClassName = 'EIdSocketError'
        then MessageText := 'Невірно зазначений порт або IP з'єднання'
        else
            if E.ClassName = 'EIdProtocolReplyError'
            then MessageText := 'Неправильний логін/пароль';
            Result := MessageError (MessageText);
            tmpStream. Free;
            DeleteFile (PChar (FileName));
        end;
    end;
end;

```

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

молодшого. Генератор ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру схематично приведено у стандарті.

Як впливає з генератора ключових потоків ДСТУ 8845:2019 у режимі генерації гами шифру основними компонентами генератору є реєстр зсуву з лінійним зворотнім зв'язком та скінчений автомат на базі якого виконується нелінійне перетворення T . Вхідні дані (ключ шифрування K та вектор ініціалізації IV) використовуються для ініціалізації змінної стану $S_i (i \geq 0)$, яка складається із двох компонент до складу яких входять [12]:

- 16 змінних $s^{(i)}$ – комірок реєстра зсуву з лінійним зворотнім зв'язком: $s^{(i)} = (s_{15}^{(i)}, s_{14}^{(i)}, s_{13}^{(i)}, s_{12}^{(i)}, s_{11}^{(i)}, s_{10}^{(i)}, s_9^{(i)}, s_8^{(i)}, s_7^{(i)}, s_6^{(i)}, s_5^{(i)}, s_4^{(i)}, s_3^{(i)}, s_2^{(i)}, s_1^{(i)}, s_0^{(i)})$;
- Два реєстри скінченного автомату $r^{(i)} : r^{(i)} = (r_2^{(i)}, r_1^{(i)})$. На виході отримуємо ключовий потік (гамма), який формується з 8-байтних слів Z_i .

З рисунку слідує, що відводи реєстра зсуву з лінійним оберненим зв'язком побудовані за примітивним над полем $GF(2^{64})$ поліномом: $f(x) = x^{16} + x^{13} + \alpha^{-1}x^{11} + \alpha$, де α є коренем примітивного над полем $GF(2^8)$ поліному $g(z) = z^8 + \beta^{170}z^7 + \beta^{166}z^6 + \beta^2 z^5 + \beta^{224} z^4 + \beta^{70}z^3 + \beta^2$. Поле $GF(2^8)$ як і в ДСТУ 7624:2014 побудовано за примітивним на полем $GF(2)$ поліномом $p(y) = x^8 + x^4 + x^3 + x^2 + 1$, а коефіцієнти $g(z)$ подаються через ступінь примітивного елементу β поля $GF(2^8)$, тобто β корінь поліному $p(y)$. Тобто, у нас є вежа полів: $GF(2) \subset GF(2^8) \subset GF(2^{64}) \subset GF(2^{1024})$, де:

- поле $GF(2^{1024})$ задається відводами зворотного зв'язку як фактор кільце $GF(2^{64})[x]/(f(x))$;
- поле $GF(2^{16424})$ задається як фактор кільце $GF(2^8)[z]/(g(z))$;
- поле $GF(2^{1024})$ задається як фактор кільце $GF(2)[y]/(p(y))$.

З вищезазначеного, слідує що період вихідної послідовності становить 2^{1024} .

Структурно в алгоритмі симетричного потокового перетворення ДСТУ 8845:2019 виділяють три основні функції:

– функція ініціалізації, яка приймає в якості вхідних даних 256-бітний вектор ініціалізації IV та 256-бітний або 512-бітний секретний ключ K, і виробляє початкове значення змінної стану $S_0 = (s^{(0)}, r^{(0)})$;

– функція наступного стану Next, яка приймає на вхід змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє наступне значення змінної стану $S_{i+1} = (s^{(i+1)}, r^{(i+1)})$;

– функція ключового потоку Strm, що приймає на вході змінну стану $S_i = (s^{(i)}, r^{(i)})$ та виробляє на виході 64-бітний ключовий потік Z^i .

Також функція Next може виконуватися в двох режимах, в залежності від способу виконання ітерації, як частина реалізації ініціалізації алгоритму ДСТУ 8845:2019 або як частина функції ключового потоку Strm.

КБГПЗ-2023

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

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

- Меню яке складається з пунктів файл, налаштування та довідка.
- Лівої верхньої деревовидної панелі відображення стану програми.
- Лівої нижньої деревовидної панелі відображення папок диску.
- Правої панелі інформації.

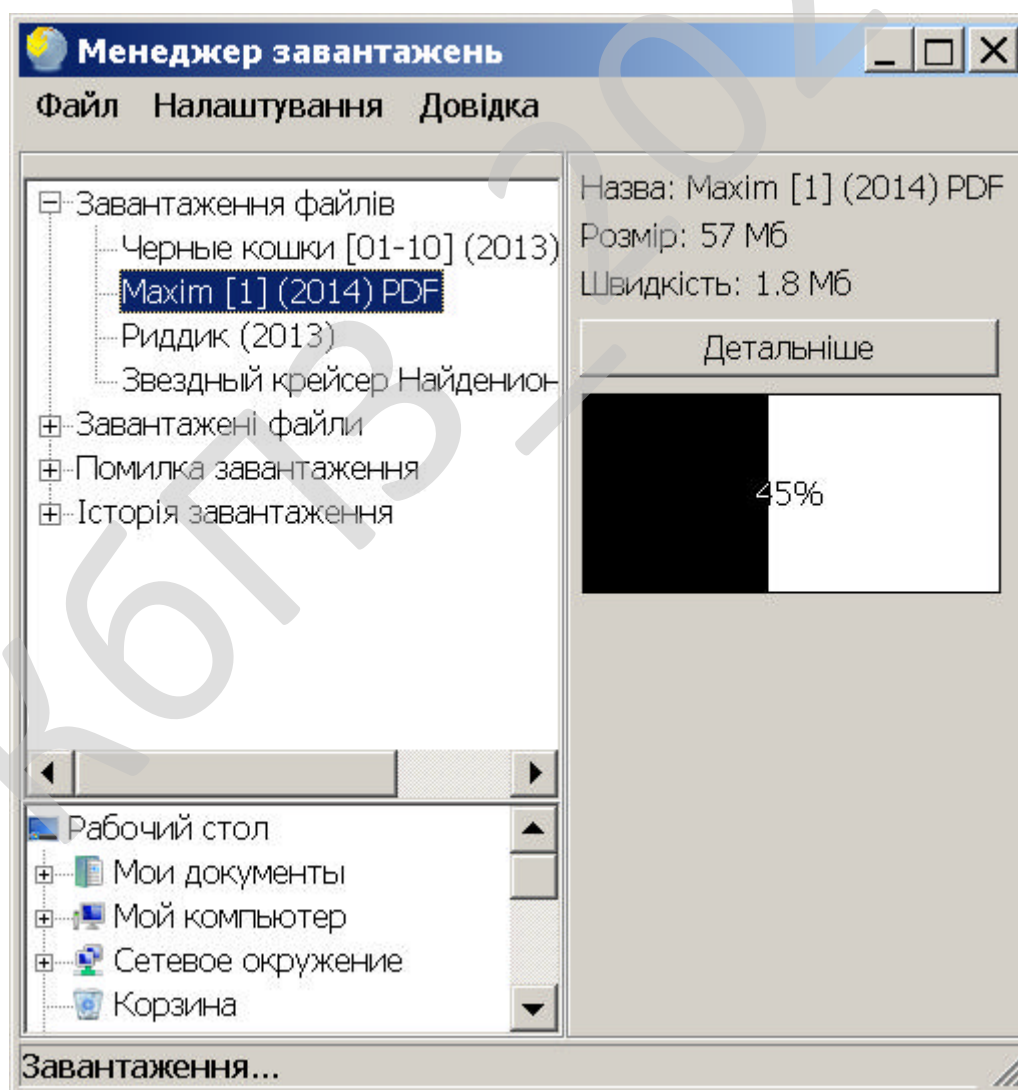


Рисунок 5.1 – Головне вікно програми

Було обрано Shareware умову розповсюдження.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (неповнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

КБГПЗ-2023

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Метою розробки є дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Об'єктом дослідження є процес багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Предметом дослідження є методи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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

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

– Удосконалено метод багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

– Розроблено вітчизняний продукт багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи багатфункціонального менеджера завантажень мультимедіа даних з мережі Internet. 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	50
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
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

1	2	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 табл. 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 ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	50000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – 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.:

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

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

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

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 ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 56 = 95 \text{ люд/день}$$

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

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	95	Ф 7.1-7.4
Впровадження	13	Д13
Всього	136	–

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,

$$Ч = \frac{136 \cdot 1}{60-5} = 2,5 \text{ ставки}$$

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 обладнан ня	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромереж	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки}$$

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	Кількість штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, серверу доступу АДСЛ (ОС Linux), Wi-Fi	2	0,5
	налаштування ADSL, VPN, PPPoE, Frame Relay		
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів д мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість 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:

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	11585	34755
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	2,5	13000	97500
Інженер-електронщик	1,2	12000	43200
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	12000	18000
Всього за період розробки	$R_{cn}=6,95$	-	$\Phi_{роб}=256455$

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.

$$z_{cd} = \frac{256455}{6,95 \cdot 60} = 615 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць. S_y – питома площа на одне робоче місце, m^2 ; C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми hotline за 16.10.23 – джерело <https://hotline.ua>

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	AMD Ryzen 3 3200G (YD3200C5FHMPK) AM4, 4 ядра, 4 потоки, 3.6 GHz, 4.0 GHz, TDP – 65 Вт, 12nm	-
Системна плата	GIGABYTE A520M H DDR4, 64 ГБ, 5100 MHz, LAN – 1 Гбіт/с, DVI, HDMI, 1 x M.2 2280, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon Vega 8	-
Жорсткий диск	SSD M.2 2280 240GB Apacer (AP240GAST280-1) 240 GB, TLC, M.2, SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 2666 MHz Kingston (KVR26N19S8/8)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	Gamemax MT520-450W Мультимедійний, Miditower, ATX, Mini – ITX, PSU – 450 Вт, Слотів розширення – 7	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-
інше	Клавіатура, мишка	-
Система охолодження	Deepcool BETA 10 Для процесорів – AMD, сокет – AM4, AM2, AM2+, AM3, AM3+, FM1, FM2	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	LG W2363V-WF Wide LCD 2ms, 70 000:1, 300кд/м2, 170/160, D-Sub / Glossy White	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	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	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	50000	10	5000
Разом	$K_p = 1606075$		$A_p = 141140$

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.

$$Z_o = 615 \cdot 136 / 50 = 1672 \text{ грн}$$

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, %

$$Z_d = 1672 \cdot 10 \cdot 0,01 = 167,2 \text{ грн}$$

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, %

$$C_{oc} = 0,01 \cdot 22(1672+167,2) = 681 \text{ грн}$$

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

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

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 1672 \cdot 15 \cdot 0,01 = 251 \text{ грн}$$

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.

Згідно виданих норм n_{mic} приймаємо 1/6 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n=210$ грн., визначаємо вартість паперу за період розробки $N_m=3$ міс:

$$Z_{M1} = C_n \cdot N_m \cdot n_{mic}. \quad (7.16)$$

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 10 примірників:

$$Z_{M2} = \sum C_{d.}, \quad (7.17)$$

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 41,2 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 41,2 грн/шт.

$$Z_{M2} = 41,2 \cdot 10 = 412 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_{z.}, \quad (7.18)$$

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

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

$$Z_M = (105 + 412 + 1702) / 50 = 44 \text{ грн.}$$

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

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

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

$$O_n = 1672 \cdot 15 \cdot 0,01 = 251 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 50$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 141140 \cdot 3 / (50 \cdot 12) = 706 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

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

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

$$C_n = 1672 + 167 + 681 + 251 + 44 + 251 + 706 = 3772 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 55%

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

де P_n – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 3772 = 2075 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	Z_o	1672
2. Додаткова зарплата виконавців	Z_d	167
3. Відрахування на соціальні потреби	C_{oc}	681
4. Загальногосподарські витрати	Γ_{ocn}	251
5. Витрати на матеріали	Z_M	44
6. Освоєння нових операційних систем, мов програмування	O_n	251
7. Амортизація основних фондів	A_M	706
8. Повна собівартість програмного забезпечення	C_n	3772
9. Плановий прибуток	P_p	2075
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5847
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{\text{дв}} \cdot C_n$	$ПДВ$	1169,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	7922

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	–	7922
Всього капітальних витрат	–	7922

7.7 Визначення експлуатаційних витрат

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

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування	Z_p	26572	11072
2. Витрати на електроенергію	$Z_{ел}$	2455	2215
3. Витрати на амортизацію	$Z_{ам}$	0	1981
Всього витрат за рік	I	29027	15268

Витрати на оплату праці:

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

де T_p – кількість годин обслуговування системи за рік, год.; Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин обслуговування системи зменшилась з 180 годин до 75 годин на рік, тому витрати на обслуговування склали:

$$Z_{p \text{ баз}} = 180 \cdot 110 \cdot 1,1 \cdot 1,22 = 26572 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 75 \cdot 110 \cdot 1,1 \cdot 1,22 = 11072 \text{ грн.}$$

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 ($C_{ел}$).

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

$$Z_{ел \text{ баз}} = 0,475 \cdot 2455 \cdot 3,8 = 4431 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,475 \cdot 1227 \cdot 3,8 = 2215 \text{ грн}$$

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	–	7922	–	1980,5
Всього відрахувань	-	–	7922	–	1980,5

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.

$$E_e = (5847 - 3772) \cdot 50 - (0,05 \cdot 1408000 + 0,5 \cdot 114885 + 0,25 \cdot 33190 + 0,1 \cdot 50000) \cdot 3/12 = 68465 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

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

де: K_p^* – балансова вартість основних фондів розробника.

$$T_e = \frac{1606075}{(5847 - 3772) \cdot 50 \cdot 12 / 3} = 3,87 \text{ років.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

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

де $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно, $K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються

$$E_{cn} = (29027 - 15268) - 0,25 \cdot 7922 = 11779 \text{ грн.}$$

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

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

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

$$T_{cn} = \frac{7922}{29027 - 15268} = 0,6 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	50
2. Повна собівартість розробленої програми	Грн.	3772
3. Ціна розробленої програми	Грн.	5847
4. Плановий прибуток від реалізації розробленої програми	Грн.	2075
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1606075
7. Загальний прибуток від реалізації програмної продукції	Грн.	103750
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	68465
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	3,87
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	7922
11. Величина економічного ефекту у користувача програмної продукції	Грн.	11779
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,6

7.9 Висновки

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

КБПЗ-2023

					VKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Програмісти у процесі роботи отримують негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки (м'язи рук та суглоби пальців) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій спеціалісти відносять високочастотні електромагнітні коливання роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

праці в робочих місць; проведення атестації робочих місць в умовах праці. Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці [5].

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальні машини (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98 [2].

Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово-емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок.

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юнктивіти, тендовагініти ті інші захворювання.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу(або другу) групу допуску з електробезпеки.

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України “Про затвердження Порядку проведення медичних оглядів працівників певних категорій” від 21 травня 2007 року №246 [8].

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

8.3 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням)

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [21]

8.4 Пропозиції щодо підвищення працездатності ІТ-фахівця

Практичне значення заходів щодо підвищення працездатності впливає із закономірностей її динаміки і зводиться ось до чого:

- збільшення фази стійкого стану у фонді робочого часу;
- прискорення процесу працювання;

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

- віддалення фази розвитку втоми;
- забезпечення високої продуктивності праці за нормальних фізіологічних затрат.

Комплекс заходів щодо підвищення і збереження працездатності працівників на оптимальному рівні реалізується на техніко-організаційному, соціально-економічному, санітарно-гігієнічному, медико-біологічному, психологічному напрямках.

Вагомим фактором високої працездатності і продуктивності праці є оптимізація трудових навантажень на основі механізації і автоматизації виробничих процесів, удосконалення технології, скорочення і ліквідації важкої ручної праці. Доведено, що при правильній організації праці на легких роботах спостерігається найбільша тривалість фази стійкого стану, а на важких роботах вона нетривала.

Високий рівень працездатності безпосередньо залежить від умов праці, оскільки поліпшення їх супроводжується зменшенням енергетичних затрат організму на подолання несприятливого впливу факторів виробничого середовища.

Важливим напрямком підвищення працездатності працюючих є ритмізація трудових процесів, оптимізація темпу роботи, а також раціоналізація трудових рухів на фізіологічній основі, що сприяє формуванню і закріпленню робочих динамічних стереотипів, а отже зменшенню м'язових і вольових зусиль. Ритмічна робота підвищує функціональні можливості організму, сприяє його тренуваності і забезпечує економізацію енергетичних затрат. [1]

Багатьом програмістам постійно доводиться працювати з великою кількістю програм одночасно. Часте перемикання туди-сюди між IDE та довідкою суттєво зменшує продуктивність фахівця. Однак вирішення цієї проблеми досить просте та очевидне: встановлення більшої кількості моніторів.

Оптимальним варіантом є два монітори. Все ж таки це найпростіший з апаратної точки зору варіант. Крім того, якби їх було більше, то ними було б

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

важче керувати, та й столі просто не вистачить місця на ще один монітор. Але тут ще залежить розміру моніторів. Є системи із 4 або 6 відносно невеликими екранами, які кріпляться на кронштейні. Але оптимальним є два 27-дюймові монітори, на яких все добре видно, особливо коли працювати доводиться в основному з текстом [3].

8.4 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [24]

8.6 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення, ширина якого складає 6,1 м, довжина – 6,8 м, висота – 3,2 м.

У зазначеному приміщенні працює 7 людей.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

типом лампам) [8]. Підставимо всі значення у формулу, визначемо світловий потік: $F=90391$ Лм.

Для розрахунку будемо використовувати світлодіодні стельові панелі Призма-72 6400К, світловий потік яких $F_n = 7200$ Лм.

Число ламп визначається по формулі:

$$N=F/F_n$$

де: F – світловий потік,

F_n – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначемо індекса приміщення:

$$N= 90391 / 7200=12,5 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 13 шт.

8.7 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 умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з умов поліпшення охорони праці.

Список використаних джерел інформації

1. Державні будівельні норми України. Природне і штучне освітлення: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення: 16.10.2023).

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> (дата звернення: 16.10.2023).

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 16.10.2023).

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

5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення: 16.10.2023).

6. **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> (дата звернення 19.09.22).

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

8. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://cpo.stu.cn.ua> (дата звернення: 16.10.2023).

9. 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 19.09.22).

10. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. — 19 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення: 16.10.2023).

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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 багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.
- Досліджена система багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.
- На основі отриманих результатів досліджень створена програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 8845:2019.

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 11779 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,6 роки.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мікіньов І.Ю. Дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
5. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
6. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
7. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
8. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

9. 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.

10. 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.

11. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

12. 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.

13. 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.

14. 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.

15. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019,

					BKPM-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

16. 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.

17. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

18. 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.

19. 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.

20. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

21. 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.

22. 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.

23. 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.

24. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

25. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

26. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

27. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

28. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

29. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

30. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

31. 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.

32. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

33. 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.

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

35. 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.

36. 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).

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

37. 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

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

39. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральнотраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

40. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

41. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

42. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

43. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

44. 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.

45. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

46. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

47. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

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

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

50. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник для студентів вищих навчальних закладів напрямів підготовки 8.050102 «Комп'ютерна інженерія». За ред. О.А. Смірнова Гриф «Навчальний посібник» надано у відповідності з листом Міністерства освіти і науки, молоді та спорту України від 17.04.2012 року № 1/11-5249. – Кіровоград: КНТУ 2012. – 250 с.

					ВКРМ-122.23.0054.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

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-122.23.0054.00.00.ТЗ		
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розробив</i>	<i>Мікіньов І.Ю.</i>				<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	<i>Буравченко К.О.</i>						
<i>Н. Контр.</i>	<i>Коваленко А.С.</i>				ЦНТУ КН-22М-2		
<i>Затв.</i>	<i>Смірнов О.А.</i>						
<i>Дослідження та програмна реалізація системи багатофункціонального менеджера завантажень мультимедіа даних з мережі</i>							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

2 Підстава для розробки

Підставою для розробки служить завдання на випуск 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 багатофункціонального менеджера завантажень мультимедіа даних з мережі Internet.

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-122.23.0054.00.00.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 багатofункціонального менеджера завантажень мультимедіа даних з мережі Internet;
- 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-122.23.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi.

					ВКРМ-122.23.0054.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівця.

					ВКРМ-122.23.0054.00.00.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 – 100 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2023 р.

					ВКРМ-122.23.0054.00.00.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
_____ Буравченко К.О.

*Дослідження та програмна реалізація
системи багатофункціонального менеджера завантажень мультимедіа
даних з мережі Internet*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 53

Літера: РП

Кропивницький – 2023 року

Multimedia_Download_Internet_Threads.pas - темн

```

unit uThreads;

interface

uses
  Classes, IdHTTP, IdFTP, IdComponent, IdFTPCommon, Windows, SysUtils, uObjects,
  Forms;

type

// Class TGetFileHttp //

TGetFileHttp = class(TThread)
public
  Item      : Integer;
  Reload    : Boolean;

  constructor Create(CreateSuspended : Boolean; P : Pointer);

private
  Data      : TTask;
  Tick      : Integer;
  StartSize : Integer;
  HTTP      : TIdHTTP;

  procedure OnWork(Sender : TObject; AWorkMode : TWorkMode; const AWorkCount :
Integer);
  procedure OnWorkBegin(Sender : TObject; AWorkMode : TWorkMode; const
AWorkCountMax : Integer);
  procedure OnWorkEnd(Sender: TObject; AWorkMode: TWorkMode);

protected
  procedure Execute; override;
end;

// Class TGetOptionsHttp //

TGetOptionsHttp = class(TThread)
public
  constructor Create(CreateSuspended : Boolean; P : Pointer);

private
  Data : TTask;

protected
  procedure Execute; override;
end;

// Class TGetFileFtp //

TGetFileFtp = class(TThread)
public
  Item      : Integer;
  Reload    : Boolean;

  constructor Create(CreateSuspended : Boolean; P : Pointer);

private
  Data : TTask;
  Tick : Integer;
  FTP  : TIdFTP;

  procedure OnWork(Sender : TObject; AWorkMode : TWorkMode; const AWorkCount :
Integer);

```

```

    procedure OnWorkBegin(Sender : TObject; AWorkMode : TWorkMode; const
AWorkCountMax : Integer);
    procedure OnWorkEnd(Sender: TObject; AWorkMode: TWorkMode);

    protected
    procedure Execute; override;
end;

// Class TGetOptionsFtp //

TGetOptionsFtp = class(TThread)
    public
        constructor Create(CreateSuspended : Boolean; P : Pointer);

    private
        Data : TTask;

    protected
        procedure Execute; override;
end;

implementation

uses uMain, uProcedures;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                     Class 'TGetFileHttp'                                     //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

constructor TGetFileHttp.Create(CreateSuspended : Boolean; P : Pointer);
begin
    Data := P;

    inherited Create(CreateSuspended);
end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

procedure TGetFileHttp.Execute;
var
    FileStream : TFileStream;
begin
    HTTP := TIdHTTP.Create(nil);

    if FileExists(Data.Directory + '\' + Data.FileName) then
    begin
        FileStream := TFileStream.Create(Data.Directory + '\' + Data.FileName,
fmOpenReadWrite);
        FileStream.Position := FileStream.Size;

    end else
    begin
        FileStream := TFileStream.Create(Data.Directory + '\' + Data.FileName,
fmCreate);

    end;

    HTTP.OnWork := OnWork;
    HTTP.OnWorkBegin := OnWorkBegin;

```

```

HTTP.OnWorkEnd := OnWorkEnd;

if Options.HTTPVersion = hvHttp10 then HTTP.ProtocolVersion := pv1_0 else
HTTP.ProtocolVersion := pv1_1;

if (Options.HTTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) = False)
then
begin

    HTTP.ProxyParams.ProxyServer := Options.HTTPProxy.Host;
    HTTP.ProxyParams.ProxyPort := Options.HTTPProxy.Port;
    HTTP.ProxyParams.ProxyUsername := Options.HTTPProxy.UserName;
    HTTP.ProxyParams.ProxyPassword := Options.HTTPProxy.Password;

end;

if Data.UseSpecial then
begin

    HTTP.Port := Data.Port;

    HTTP.Request.Username := Data.Login;
    HTTP.Request.Password := Data.Password;

end;

HTTP.Request.ContentRangeStart := Data.LoadSize;
HTTP.Request.ContentRangeEnd := Data.TotalSize;

HTTP.HandleRedirects := Options.Redirect;

StartSize := Data.LoadSize;

try

    HTTP.Get(Data.LinkToFile, FileStream);

except

    on E : Exception do
    begin

        Data.Status := tsError;
        Data.ErrorText := E.Message;

        MessageBox(Application.Handle, PChar('Помилка при завантаженні файлу.' +
#13#10 + E.Message), PChar(Options.Name), MB_OK or MB_ICONERROR);

    end;

end;

HTTP.Free;
FileStream.Free;

end;

////////////////////////////////////

procedure TGetFileHttp.OnWork(Sender: TObject; AWorkMode: TWorkMode; const
AWorkCount: Integer);
var
    TickCount : Integer;
    Count      : Integer;

begin

    if AWorkMode = wmRead then
    begin

```

```

    Data.LoadSize := StartSize + AWorkCount;

    TickCount := GetTickCount;
    Count := (TickCount - Tick) div 1000;

    if (Data.LoadSize > 0) and (Count > 0) then Data.Speed := Data.LoadSize div
Count;

    end;

    if Data.Status = tsStoped then HTTP.Disconnect;

end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
procedure TGetFileHttp.OnWorkBegin(Sender : TObject; AWorkMode : TWorkMode;
const AWorkCountMax : Integer);
begin
    Tick := GetTickCount;

    Data.TimeBegin := Now;
    Data.Status := tsLoading;

end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
procedure TGetFileHttp.OnWorkEnd(Sender: TObject; AWorkMode: TWorkMode);
begin
    Data.TimeEnd := Now;
    Data.TimeTotal := Data.TimeBegin - Data.TimeEnd;

    if Data.Status <> tsStoped then Data.Status := tsLoad;

end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                     'TGetOptionsHttp'                                     //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
constructor TGetOptionsHttp.Create(CreateSuspended : Boolean; P : Pointer);
begin
    Data := P;

    inherited Create(CreateSuspended);

end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

procedure TGetOptionsHttp.Execute;
var
    HTTP : TIdHTTP;

begin
    HTTP := TIdHTTP.Create(nil);

    if (Options.HTTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) = False)
then
    begin

```

```

HTTP.ProxyParams.ProxyServer := Options.HTTPProxy.Host;
HTTP.ProxyParams.ProxyPort := Options.HTTPProxy.Port;
HTTP.ProxyParams.ProxyUsername := Options.HTTPProxy.UserName;
HTTP.ProxyParams.ProxyPassword := Options.HTTPProxy.Password;

end;

if Data.UseSpecial then
begin

    HTTP.Port := Data.Port;
    HTTP.Request.Username := Data.Login;
    HTTP.Request.Password := Data.Password;

end;

HTTP.HandleRedirects := Options.Redirect;

if Options.HTTPVersion = hvHttp10 then HTTP.ProtocolVersion := pv1_0 else
HTTP.ProtocolVersion := pv1_1;

try

    HTTP.Head(Data.LinkToFile);

    Data.Status := tsReady;
    Data.ErrorText := 'Помилко немає';

    Data.TotalSize := HTTP.Response.ContentLength;
    Data.LastModified := HTTP.Response.LastModified;

except

    on E : Exception do
    begin

        Data.Status := tsError;
        Data.ErrorText := E.Message;

        Data.TotalSize := 0;
        Data.LastModified := 0;

        MessageBox(Application.Handle, PChar('Помилка при завантаженні файлу.' +
#13#10 + E.Message), PChar(Options.Name), MB_OK or MB_ICONERROR);

    end;

end;

HTTP.Free;

fMain.RefreshTasks;

end;

////////////////////////////////////////////////////////////////////////////////
//                                           Class 'TGetFileFtp'                                           //
////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////

constructor TGetFileFtp.Create(CreateSuspended : Boolean; P : Pointer);
begin

    Data := P;

    inherited Create(CreateSuspended);

end;

```

```
////////////////////////////////////  
procedure TGetFileFtp.Execute;  
begin  
    FTP := TIdFTP.Create(nil);  
  
    FTP.OnWork := OnWork;  
    FTP.OnWorkBegin := OnWorkBegin;  
    FTP.OnWorkEnd := OnWorkEnd;  
  
    if (Options.FTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) = False)  
    then  
    begin  
        FTP.ProxySettings.Host := Options.FTPProxy.Host;  
        FTP.ProxySettings.Port := Options.FTPProxy.Port;  
        FTP.ProxySettings.Username := Options.FTPProxy.UserName;  
        FTP.ProxySettings.Password := Options.FTPProxy.Password;  
  
    end;  
  
    if Data.UseSpecial then  
    begin  
        FTP.Port := Data.Port;  
        FTP.Username := Data.Login;  
        FTP.Password := Data.Password;  
  
    end else FTP.Username := 'anonymous';  
  
    try  
        FTP.TransferType := ftBinary;  
        FTP.Host := ExtractAddress(Data.LinkToFile);  
        FTP.Connect;  
        FTP.Get(ExtractFileName(Data.LinkToFile), Data.Directory + '\' +  
Data.FileName, False, True);  
        FTP.Disconnect;  
  
        Data.Status := tsReady;  
        Data.ErrorText := 'Помилка немає';  
  
    except  
        on E : Exception do  
        begin  
            Data.Status := tsError;  
            Data.ErrorText := E.Message;  
  
            MessageBox(Application.Handle, PChar('Помилка при завантаженні файлу.' +  
#13#10 + E.Message), PChar(Options.Name), MB_OK or MB_ICONERROR);  
  
        end;  
  
    end;  
  
    FTP.Free;  
  
end;  
  
////////////////////////////////////  
procedure TGetFileFtp.OnWork(Sender: TObject; AWorkMode: TWorkMode; const  
AWorkCount: Integer);  
var  
    TickCount : Integer;
```

```

Count      : Integer;

begin

  if AWorkMode = wmRead then
  begin

    Data.LoadSize := AWorkCount;

    TickCount := GetTickCount;
    Count := (TickCount - Tick) div 1000;

    if (Data.LoadSize > 0) and (Count > 0) then Data.Speed := Data.LoadSize div
    Count;

    end;

    if Data.Status = tsStoped then FTP.Disconnect;

end;

if Data.Status = tsStoped then FTP.Disconnect;

end;

////////////////////////////////////////////////////////////////

procedure TGetFileFtp.OnWorkBegin(Sender : TObject; AWorkMode : TWorkMode; const
AWorkCountMax : Integer);
begin

  Tick := GetTickCount;

  Data.TimeBegin := Now;
  Data.Status := tsLoading;

end;

////////////////////////////////////////////////////////////////

procedure TGetFileFtp.OnWorkEnd(Sender: TObject; AWorkMode: TWorkMode);
begin

  Data.TimeEnd := Now;
  Data.TimeTotal := Data.TimeBegin - Data.TimeEnd;

  if Data.Status <> tsStoped then Data.Status := tsLoad;

end;

////////////////////////////////////////////////////////////////
//                                     'TGetOptionsFtp'                                     //
////////////////////////////////////////////////////////////////

constructor TGetOptionsFtp.Create(CreateSuspended : Boolean; P : Pointer);
begin

  Data := P;

  inherited Create(CreateSuspended);

end;

////////////////////////////////////////////////////////////////

procedure TGetOptionsFtp.Execute;
var
  FTP : TIdFTP;
  Details : TStrings;

begin

```

```

FTP := TIdFTP.Create(nil);
Details := TStringList.Create;

if (Options.FTPProxy.UseProxy) and (LocalAddress(Data.LinkToFile) = False)
then
begin
    FTP.ProxySettings.Host := Options.FTPProxy.Host;
    FTP.ProxySettings.Port := Options.FTPProxy.Port;
    FTP.ProxySettings.Username := Options.FTPProxy.UserName;
    FTP.ProxySettings.Password := Options.FTPProxy.Password;

end;

if Data.UseSpecial then
begin
    FTP.Port := Data.Port;
    FTP.Username := Data.Login;
    FTP.Password := Data.Password;

end else FTP.Username := 'anonymous';

try
    FTP.TransferType := ftBinary;
    FTP.Host := ExtractAddress(Data.LinkToFile);
    FTP.Connect;

    FTP.List(Details, ExtractFileName(Data.LinkToFile), True);
    Data.TotalSize := FTP.DirectoryListing.Items[0].Size;
    Data.LastModified := FTP.DirectoryListing.Items[0].ModifiedDate;

    FTP.Disconnect;

    Data.Status := tsReady;
    Data.ErrorText := 'Помилка немає';

except
    on E : Exception do
    begin
        Data.Status := tsError;
        Data.ErrorText := E.Message;

        Data.TotalSize := 0;
        Data.LastModified := 0;

        MessageBox(Application.Handle, PChar('Помилка при завантаженні файлу.' +
        #13#10 + E.Message), PChar(Options.Name), MB_OK or MB_ICONERROR);

        end;

    end;

    FTP.Free;
    Details.Free;

    fMain.RefreshTasks;

end;

end.

```

**Multimedia_Download_Internet_Main.pas - головна програма
багатофункціонального менеджера завантажень мультимедіа даних з
мережі Internet**

```
unit uMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ToolWin, ComCtrls, ImgList, StdActns, Menus, ExtCtrls, ActnList,
  XPStyleActnCtrls, ActnMan, ActnCtrls, ActnMenus, StdCtrls, XPMan, Clipbrd,
  DateUtils, ShellAPI;

type
  TfMain = class(TForm)
    ilMenu: TImageList;
    ActionMainMenuBar1: TActionMainMenuBar;
    ActionManager1: TActionManager;
    actExit: TAction;
    actAdd: TAction;
    actOptions: TAction;
    lvTasks: TListView;
    StatusBar: TStatusBar;
    actStatusBar: TAction;
    actInfo: TAction;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Splitter1: TSplitter;
    tvFolders: TTreeView;
    Splitter2: TSplitter;
    Panel2: TPanel;
    lvParams: TListView;
    XPManifest1: TXPManifest;
    ilTasks: TImageList;
    actLoad: TAction;
    actDelete: TAction;
    ilTree: TImageList;
    actParams: TAction;
    actStop: TAction;
    ilTray: TImageList;
    actClearDel: TAction;
    actEdit: TAction;
    actAbout: TAction;
    actAddCategory: TAction;
    actDeleteCategory: TAction;
    actEditCategory: TAction;
    pmTasks: TPopupMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N5: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    pmTree: TPopupMenu;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N14: TMenuItem;
    tmUpdate: TTimer;
    N15: TMenuItem;
    N16: TMenuItem;
    ActionToolBar1: TActionToolBar;
    pmTray: TPopupMenu;
```

```

N17: TMenuItem;
N18: TMenuItem;
N20: TMenuItem;
N21: TMenuItem;
N22: TMenuItem;
N19: TMenuItem;
N23: TMenuItem;
N24: TMenuItem;
actOpenFolder: TAction;
N4: TMenuItem;
N6: TMenuItem;
actOpenFile: TAction;
N13: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure actExitExecute(Sender: TObject);
procedure actAddExecute(Sender: TObject);
procedure actOptionsExecute(Sender: TObject);
procedure actStatusBarExecute(Sender: TObject);
procedure actInfoExecute(Sender: TObject);
procedure actLoadExecute(Sender: TObject);
procedure actDeleteExecute(Sender: TObject);
procedure lvParamsCustomDrawItem(Sender: TCustomListView; Item: TListItem;
State: TCustomDrawState; var DefaultDraw: Boolean);
procedure actParamsExecute(Sender: TObject);
procedure lvTasksSelectItem(Sender: TObject; Item: TListItem; Selected:
Boolean);
procedure tvFoldersChange(Sender: TObject; Node: TTreeNode);
procedure actStopExecute(Sender: TObject);
procedure TrayIconClick(Sender: TObject);
procedure lvTasksCustomDrawItem(Sender: TCustomListView; Item: TListItem;
State: TCustomDrawState; var DefaultDraw: Boolean);
procedure actClearDelExecute(Sender: TObject);
procedure actEditExecute(Sender: TObject);
procedure actAboutExecute(Sender: TObject);
procedure Panel2Resize(Sender: TObject);
procedure Panel1Resize(Sender: TObject);
procedure actAddCategoryExecute(Sender: TObject);
procedure lvTasksKeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
procedure actDeleteCategoryExecute(Sender: TObject);
procedure actEditCategoryExecute(Sender: TObject);
procedure tmUpdateTimer(Sender: TObject);
procedure N17Click(Sender: TObject);
procedure actOpenFolderExecute(Sender: TObject);
procedure actOpenFileExecute(Sender: TObject);
private
  NextViewerHandle : THandle;
  procedure OnDrawClipboard(var Message : TMessage); Message WM_DRAWCLIPBOARD;
  procedure OnChangeCBChain(var Message : TMessage); Message WM_CHANGECHAIN;
public
  procedure InsertItem(P : Pointer);
  procedure UpdateItems;
  procedure RefreshTasks;
  procedure SaveTreeNode(Node : TTreeNode);
end;

var
  fMain: TfMain;

implementation

uses uAddTask, uObjects, uProcedures, uOptions, uLoading, uThreads, uEditTask,
uAbout, uAddCategory, uEditCategory,
uDownload;

{$R *.dfm}

procedure TfMain.UpdateItems;

```

```

var
  Data: TTask;
  i: integer;
begin
  for i:=0 to lvTasks.Items.Count-1 do
  begin
    Data := lvTasks.Items[i].Data;
    if Data.Status=tsLoading
    then
      begin
        if Data.TotalSize>0
        then lvTasks.Items[i].SubItems[1]:=BytesToText(Data.TotalSize)
        else lvTasks.Items[i].SubItems[1]='?';
        lvTasks.Items[i].SubItems[2]:=BytesToText(Data.LoadSize);
        if Data.TotalSize>0
        then
          lvTasks.Items[i].SubItems[3]:=FloatToStr((Data.LoadSize/Data.TotalSize)*100,
          ffFixed, 18, 0)
          else lvTasks.Items[i].SubItems[3]='?';
          if Data.Speed>0
          then lvTasks.Items[i].SubItems[4]:=BytesToText(Data.Speed)+'/'
          else lvTasks.Items[i].SubItems[4]='?';
        end;
      end;
    end;
  end;

  //Створення головної форми

  procedure TfMain.FormCreate(Sender: TObject);
  begin
    Options:=TOptions.Create;
    Options.Path:=ExtractFileDir(Application.ExeName);
    Options.Version:=GetFileVersion(Application.ExeName);
    Options.Name:='Менеджер завантажень файлів з мережі';
    Options.Load;
    NextViewerHandle:=SetClipboardViewer(Handle);
    Application.Title:=Options.Name+' ';
    Caption:=Options.Name;
    tvFolders.FullExpand;
    actParams.Checked:=Panel2.Visible;
    actStatusBar.Checked:=StatusBar.Visible;
    tvFoldersChange(Self, tvFolders.Selected);
  end;

  //Закриття головної форми

  procedure TfMain.FormClose(Sender: TObject; var Action: TCloseAction);
  begin
    ChangeClipboardChain(Handle, NextViewerHandle);
    Options.Save;
    Options.Free;
  end;

  //закінчення роботи

  procedure TfMain.actExitExecute(Sender: TObject);
  begin
    Close;
  end;

  //додання нового посилання на навантаження

  procedure TfMain.actAddExecute(Sender: TObject);
  begin
    fAddTask:=TfAddTask.Create(Application);
    fAddTask.ShowModal;
  end;

  //Встановлення опцій

```

```

procedure TfMain.actOptionsExecute(Sender: TObject);
begin
  fOptions:=TfOptions.Create(Application);
  fOptions.ShowModal;
end;

//Виведеннястатусбара

procedure TfMain.actStatusBarExecute(Sender: TObject);
begin
  StatusBar.Visible:=actStatusBar.Checked;
end;

//Виведення інформації про завантаження

procedure TfMain.actInfoExecute(Sender: TObject);
var
  ThreadHttp: TGetOptionsHttp;
  ThreadFtp: TGetOptionsFtp;
begin
  if lvTasks.SelCount=0
  then Exit;
  if (TTask(lvTasks.Selected.Data).Protocol=ptHttp)
    or (TTask(lvTasks.Selected.Data).Protocol=ptHttps)
  then
  begin
    ThreadHttp:=TGetOptionsHttp.Create(true, lvTasks.Selected.Data);
    ThreadHttp.Priority:=Options.Priority;
    ThreadHttp.FreeOnTerminate:=true;
    ThreadHttp.Resume;
  end;
  if TTask(lvTasks.Selected.Data).Protocol=ptFtp
  then
  begin
    ThreadFtp:=TGetOptionsFtp.Create(true, lvTasks.Selected.Data);
    ThreadFtp.Priority:=Options.Priority;
    ThreadFtp.FreeOnTerminate:=true;
    ThreadFtp.Resume;
  end;
end;

//Редагування завантаження

procedure TfMain.actLoadExecute(Sender: TObject);
var
  LoadOne: TLoadOne;
begin
  if lvTasks.SelCount=0
  then Exit;
  if Options.MinOnRun
  then Application.Minimize;
  if TTask(lvTasks.Selected.Data).Status=tsLoad
  then
  begin
    if MessageBox(Application.Handle, PChar('Файл
    '+TTask(lvTasks.Selected.Data).FileName + ' вже був завантажений. Завантажити
    його знову?'), PChar(Options.Name), MB_YESNO or MB_ICONERROR)=IDYES
    then
    begin
      TTask(lvTasks.Selected.Data).LoadSize:=0;
      TTask(lvTasks.Selected.Data).Status:=tsReady;
    end
    else Exit;
  end;
  if Options.ShowLoadingForm
  then
  begin
    fLoading:=TfLoading.Create(Application);
  end;
end;

```

```

    fLoading.Data:=lvTasks.Selected.Data;
    fLoading.Show;
end;
LoadOne:=TLoadOne.Create(true, lvTasks.Selected.Data);
LoadOne.FreeOnTerminate:=true;
LoadOne.Resume;
end;

//Видалення завантаження

procedure TfMain.actDeleteExecute(Sender: TObject);
var
    i: integer;
begin
    if not Assigned(lvTasks.Selected)
    then Exit;
    if MessageBox(Application.Handle, 'Видалити виділені елементи?',
PChar(Options.Name), MB_YESNO or MB_ICONWARNING)=IDYES
    then
        begin
            for i:=0 to lvTasks.Items.Count-1 do
                begin
                    if lvTasks.Items[i].Selected
                    then
                        begin
                            if TTask(lvTasks.Selected.Data).Status=tsDeleted
                            then TTask(lvTasks.Selected.Data).Status:=tsDelete
                            else TTask(lvTasks.Selected.Data).Status:=tsDeleted;
                        end;
                    end;
                end;
            lvTasks.Selected.Delete;
            lvTasks.Repaint;
        end;

//Рисуння вікна параметрів

procedure TfMain.lvParamsCustomDrawItem(Sender: TCustomListView;
Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);
begin
    if Item.Index mod 2=0
    then
        begin
            Sender.Canvas.Font.Color:=clBlack;
            Sender.Canvas.Brush.Color:=$F6F6F6;
        end
    else
        begin
            Sender.Canvas.Font.Color:=clBlack;
            Sender.Canvas.Brush.Color:=clWhite;
        end;
    end;

//Виведення параметрів

procedure TfMain.actParamsExecute(Sender: TObject);
begin
    if ActParams.Checked
    then
        begin
            Panel2.Height:=200;
            Panel1.Height:=Panel4.Height-200;
            Splitter2.Visible:=true;
        end
    else
        begin
            Panel2.Height:=0;
            Panel1.Height:=Panel4.Height;
            Splitter2.Visible:=false;
        end;
    end;
end;

```

```

    end;
end;

//Вибір вікна у якому потрібно працювати

procedure TfMain.lvTasksSelectItem(Sender: TObject; Item: TListItem; Selected:
Boolean);
var
    Data: TTask;
begin
    if (lvParams.Items[0]=nil) or (Item<>lvTasks.Selected)
    then Exit;
    Data:=Item.Data;
    lvParams.Items[0].SubItems[1]:=Data.LinkToFile;
    lvParams.Items[1].SubItems[1]:=Data.Directory;
    lvParams.Items[2].SubItems[1]:=Data.FileName;
    if Data.TotalSize>0
    then lvParams.Items[3].SubItems[1]:=IntToStr(Data.TotalSize)
    else lvParams.Items[3].SubItems[1]='?';
    lvParams.Items[4].SubItems[1]:=IntToStr(Data.LoadSize);
    if Data.LastModified<>0
    then lvParams.Items[5].SubItems[1]:=FormatDateTime('dd.mm.yyyy hh:mm:ss',
Data.LastModified)
    else lvParams.Items[5].SubItems[1]='?';
    if Data.TotalSize>0
    then lvParams.Items[6].SubItems[1]='Так'
    else lvParams.Items[6].SubItems[1]='Немає';
    if Data.TimeBegin<>0
    then lvParams.Items[7].SubItems[1]:=FormatDateTime('dd.mm.yyyy hh:mm:ss',
Data.TimeBegin)
    else lvParams.Items[7].SubItems[1]='?';
    if Data.TimeEnd<>0
    then lvParams.Items[8].SubItems[1]:=FormatDateTime('dd.mm.yyyy hh:mm:ss',
Data.TimeEnd)
    else lvParams.Items[8].SubItems[1]='?';
    if Data.TimeTotal<>0
    then lvParams.Items[9].SubItems[1]:=FormatDateTime('hh:mm:ss', Data.TimeTotal)
    else lvParams.Items[9].SubItems[1]='?';
    if Data.Speed>0
    then lvParams.Items[10].SubItems[1]:=IntToStr(Data.Speed)
    else lvParams.Items[10].SubItems[1]='?';
    lvParams.Items[11].SubItems[1]:=Data.ErrorText;
    lvParams.Items[12 ].SubItems[1]:=Data.Description;
end;

//Зміна папки куди записувати завантажений файл

procedure TfMain.tvFoldersChange(Sender: TObject; Node: TTreeNode);
var
    i: integer;
    Data: TTask;
    SortStatus: TTaskStatus;
    SortDate: TDate;
begin
    if (Node.Level=0) or (Node<>tvFolders.Selected)
    then Exit;
    lvTasks.Items.Clear;
    SortStatus:=tsReady;
    SortDate:=0;
    if (Node.Level=1) and (Node.Index=0)
    then
        begin
            for i:=0 to Options.Task.Count-1 do
                begin
                    Data:=Options.Task[i];
                    if Data.Status=tsLoad
                    then InsertItem(Data);
                end;
            end;
        end;
end;

```

```

if (Node.Level=1) and (Node.Index=1)
then
begin
for i:=0 to Options.Task.Count-1 do
begin
Data:=Options.Task[i];
if (Data.Status<>tsDelete) and (Data.Status<>tsDeleted)
then InsertItem(Data);
end;
end;
if (Node.Parent.Level=1) and (Node.Parent.Index=1)
then
begin
case Node.Index of
0: SortStatus:=tsReady;
1: SortStatus:=tsLoad;
2: SortStatus:=tsLoading;
3: SortStatus:=tsStopped;
4: SortStatus:=tsError;
end;
for i:=0 to Options.Task.Count-1 do
begin
Data:=Options.Task[i];
if Data.Status=SortStatus
then InsertItem(Data);
end;
end;
if (Node.Level=1) and (Node.Index=3)
then
begin
for i:=0 to Options.Task.Count-1 do
begin
Data:=Options.Task[i];
if Data.Status=tsDeleted
then InsertItem(Data);
end;
end;
if (Node.Parent.Level=1) and (Node.Parent.Index=2)
then
begin
if (Node.Index=0) or (Node.Index=1)
then
begin
case Node.Index of
0: SortDate:=Date;
1: SortDate:= Date-1;
end;
for i:=0 to Options.Task.Count-1 do
begin
Data:=Options.Task[i];
if (SameDate(Data.TimeEnd, SortDate)) and (Data.Status=tsLoad)
then InsertItem(Data);
end;
end;
if (Node.Index=2) or (Node.Index=3)
then
begin
case Node.Index of
2: SortDate:= Now-8;
3: SortDate:= Now-31;
end;
for i:=0 to Options.Task.Count-1 do
begin
Data:=Options.Task[i];
if (Data.TimeEnd>SortDate) and (Data.Status=tsLoad)
then InsertItem(Data);
end;
end;
end;
end;

```

```

end;

//Остановка завантаження
procedure TfMain.actStopExecute(Sender: TObject);
begin
  if not Assigned(lvTasks.Selected)
  then Exit;
  if TTask(lvTasks.Selected.Data).Status=tsLoading
  then TTask(lvTasks.Selected.Data).Status:=tsStopped;
end;

//Виведення вікна кліпбоарда
procedure TfMain.OnDrawClipboard(var Message : TMessage);
begin
  if Options.HookClipboard
  then
    begin
      if Clipboard.HasFormat(CF_TEXT)
      then
        begin
          if (Pos('http://', Trim(Clipboard.AsText))=1)
          or (Pos('https://', Trim(Clipboard.AsText))=1)
          or (Pos('ftp://', Trim(Clipboard.AsText))=1)
          then
            begin
              SetForegroundWindow(fMain.Handle);
              fAddTask:=TfAddTask.Create(Application);
              fAddTask.ShowModal;
            end;
          end;
        end;
      Message.Result:=SendMessage(WM_DRAWCLIPBOARD, NextViewerHandle, 0, 0);
    end;

  procedure TfMain.OnChangeCBChain(var Message: TMessage);
  begin
    if Message.wParam=Integer(NextViewerHandle)
    then
      begin
        NextViewerHandle:=Message.lParam;
        Message.Result:=0;
      end
    else Message.Result:=SendMessage(NextViewerHandle, WM_CHANGECHAIN,
    Message.wParam, Message.lParam);
  end;

  //Звертання системи у трей
  procedure TfMain.TrayIconClick(Sender: TObject);
  begin
    SetForegroundWindow(Handle);
  end;

  //вибір вікна поточних завдань
  procedure TfMain.lvTasksCustomDrawItem(Sender: TCustomListView;
  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);
  begin
    if Item.Index mod 2=0
    then
      begin
        Sender.Canvas.Font.Color:=clBlack;
        Sender.Canvas.Brush.Color:=$F6F6F6;
      end
    else
      begin

```

```

    Sender.Canvas.Font.Color:=clBlack;
    Sender.Canvas.Brush.Color:=clWhite;
end;
end;

//Очищення вікна завантаження

procedure TfMain.actClearDelExecute(Sender: TObject);
var
  i: integer;
begin
  if MessageBox(Application.Handle, 'Ви дійсно хочете очистити папку
"Вилучені"?', PChar(Options.Name), MB_OKCANCEL or MB_ICONWARNING)=ID_OK
  then
    begin
      for i:=0 to Options.Task.Count-1 do
        begin
          if TTask(Options.Task[i]).Status=tsDeleted
            then TTask(Options.Task[i]).Status:=tsDelete;
          end;
        lvTasks.Clear;
      end;
    end;

//додавання даних про завантаження

procedure TfMain.InsertItem(P : Pointer);
var
  ListItem: TListItem;
  Data: TTask;
begin
  Data:=P;
  ListItem:=lvTasks.Items.Insert(0);
  ListItem.SubItems.Add(Data.LinkToFile);
  if Data.TotalSize>0
  then ListItem.SubItems.Add(BytesToText(Data.TotalSize))
  else ListItem.SubItems.Add('?');
  ListItem.SubItems.Add(BytesToText(Data.LoadSize));
  if Data.TotalSize>0
  then ListItem.SubItems.Add(FloatToStr((Data.LoadSize/Data.TotalSize)*100,
fffixed, 18, 0))
  else ListItem.SubItems.Add('?');
  if Data.Speed>0
  then ListItem.SubItems.Add(BytesToText(Data.Speed)+' /с')
  else ListItem.SubItems.Add('?');
  ListItem.ImageIndex:=1;
  ListItem.Data:=Data;
end;

//Додавання завантаження

procedure TfMain.actEditExecute(Sender: TObject);
begin
  if lvTasks.SelCount=0
  then Exit;
  fEditTask:=TfEditTask.Create(Application);
  fEditTask.Data:=lvTasks.Selected.Data;
  fEditTask.ShowModal;
end;

//Інформація про завантаження

procedure TfMain.actAboutExecute(Sender: TObject);
begin
  fAbout:=TfAbout.Create(Application);
  fAbout.ShowModal;
end;

//Встановлення розмірів вікон

```

```

procedure TfMain.Panel2Resize(Sender: TObject);
var
  i: integer;
begin
  if lvParams.Width>400
  then
    begin
      if lvParams.Height<240
      then i:=40
      else i:=24;
      lvParams.Columns[2].Width:=lvParams.Width-lvParams.Columns[1].Width-i;
    end;
end;

procedure TfMain.PanellResize(Sender: TObject);
var
  i: integer;
begin
  if lvTasks.Width>500
  then
    begin
      if lvTasks.Height<lvTasks.Items.Count*16
      then i:=40
      else i:=24;
      lvTasks.Columns[1].Width:=lvTasks.Width-lvTasks.Columns[2].Width -
lvTasks.Columns[3].Width - lvTasks.Columns[4].Width-lvTasks.Columns[5].Width-i;
    end;
end;

//Додавання категорії завантаження

procedure TfMain.actAddCategoryExecute(Sender: TObject);
begin
  fAddCategory:=TfAddCategory.Create(Application);
  fAddCategory.ShowModal;
end;

procedure TfMain.lvTasksKeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
begin
  if Key=VK_DELETE
  then actDeleteExecute(nil);
end;

//Видалення категорії завантаження

procedure TfMain.actDeleteCategoryExecute(Sender: TObject);
begin
  if MessageBox(Application.Handle, PChar('Ви дійсно хочете видалити папку "' +
tvFolders.Selected.Text + '"?'), PChar(Options.Name), MB_OKCANCEL or
MB_ICONWARNING)=ID_OK
  then tvFolders.Selected.Delete;
end;

//Редагування категорії завантаження

procedure TfMain.actEditCategoryExecute(Sender: TObject);
begin
  fEditCategory:=TfEditCategory.Create(Application);
  fEditCategory.ShowModal;
end;

procedure TfMain.tmUpdateTimer(Sender: TObject);
begin
  UpdateItems;
end;

procedure TfMain.N17Click(Sender: TObject);

```

```

begin
  SetForegroundWindow(Handle);
end;

procedure TfMain.RefreshTasks;
var
  Data: TTask;
  i: integer;
begin
  for i:=0 to lvTasks.Items.Count-1 do
    begin
      Data:=lvTasks.Items[i].Data;
      if Data.TotalSize>0
      then lvTasks.Items[i].SubItems[1]:=BytesToText(Data.TotalSize)
      else lvTasks.Items[i].SubItems[1]='?';
      lvTasks.Items[i].SubItems[2]:=BytesToText(Data.LoadSize);
      if Data.TotalSize>0
      then
        lvTasks.Items[i].SubItems[3]:=FloatToStr((Data.LoadSize/Data.TotalSize)*100,
        ffFixed, 18, 0)
        else lvTasks.Items[i].SubItems[3]='?';
        if Data.Speed>0
        then lvTasks.Items[i].SubItems[4]:=BytesToText(Data.Speed)+' /с'
        else lvTasks.Items[i].SubItems[4]='?';
      end;
    end;
end;

procedure TfMain.SaveTreeNode(Node : TTreeNode);
var
  i: integer;
begin
  for i:=0 to Node.Count-1 do
    begin
      ShowMessage(Node.Item[i].Text);
      SaveTreeNode(Node.Item[i]);
    end;
  end;
end;

//Відкриття папки завантаження

procedure TfMain.actOpenFolderExecute(Sender: TObject);
begin
  if not Assigned(lvTasks.Selected)
  then Exit;
  ShellExecute(0, 'open', PChar(TTask(lvTasks.Selected).Data).Directory), '',
  '', SW_SHOWNORMAL);
end;

//Відкриття файлу завантаження

procedure TfMain.actOpenFileExecute(Sender: TObject);
begin
  if not Assigned(lvTasks.Selected)
  then Exit;
  ShellExecute(0, 'open', PChar(TTask(lvTasks.Selected).Data).Directory + '\'
+ TTask(lvTasks.Selected).Data).FileName), '', '', SW_SHOWNORMAL);
end;

end.

```

Multimedia_Download_Internet_Procedures.pas - основні процедури

```

unit uProcedures;

interface

uses
  SysUtils, Windows, ShellAPI, ShlObj;

function BytesToText(Bytes : Integer) : String;
function GetFileVersion(FileName : String) : String;
function BrowserFolder(Owner : THandle) : String;
function CreateFileName(Url : String) : String;
function LocalAddress(Url : String) : Boolean;
function ExtractAddress(Url : String) : String;
function ExtractFileName(Url : String) : String;
function GetFreeSpace(Disk : String) : Int64;
function IsRun : Boolean;
function GetTimeStr(Secs : Integer) : String;

implementation

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function BytesToText(Bytes : integer) : String;
begin
  if Bytes div 1000 < 1 then Result := IntToStr(Bytes) + ' байт';

  if Bytes div 1000 >= 1 then Result := FloatToStr(Bytes/1000, ffNumber, 18, 1)
+ ' Кб';

  if Bytes div 1000 >= 1000 then Result := FloatToStr(Bytes/1000000, ffNumber,
18, 1) + ' Мб';

end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function GetFileVersion(FileName : String) : String;
var
  Data : Pointer;
  DataSize, InfoSize : Dword;
  Dummy : Cardinal;
  Buffer: array [0..MAX_PATH] of Char;
  Major1, Major2, Minor1, Minor2 : Integer;
  FileInfo : PVSEFixedFileInfo;
begin
  StrCat(Buffer, PChar(FileName));
  DataSize := GetFileVersionInfoSize(Buffer, Dummy);

  if DataSize > 0 then
  begin
    GetMem(Data, DataSize);
    GetFileVersionInfo(Buffer, 0, DataSize, Data);
    VerQueryValue(Data, '\', Pointer(FileInfo), InfoSize);

    Major1 := FileInfo.dwFileVersionMS shr 16;
    Major2 := FileInfo.dwFileVersionMS and $FFFF;
    Minor1 := FileInfo.dwFileVersionLS shr 16;
    Minor2 := FileInfo.dwFileVersionLS and $FFFF;
  end;
end;

```

```

    Result := IntToStr(Major1) + '.' + IntToStr(Major2) + '.' + IntToStr(Minor1)
+ ' build ' + IntToStr(Minor2);

    FreeMem(Data, DataSize);

end;

end;

/////////////////////////////////////////////////

//Дерево каталогів

function BrowserFolder(Owner : THandle) : String;
var
    TitleName    : String;
    lpItemID     : PItemIDList;
    BrowseInfo   : TBrowseInfo;
    DisplayName  : Array [0..MAX_PATH] of char;
    TempPath     : Array [0..MAX_PATH] of char;

begin
    FillChar(BrowseInfo, SizeOf(TBrowseInfo), #0);

    BrowseInfo.hwndOwner := Owner;
    BrowseInfo.pszDisplayName := @DisplayName;

    TitleName := 'Виберіть директорію';

    BrowseInfo.lpszTitle := PChar(TitleName);
    BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;

    lpItemID := SHBrowseForFolder(BrowseInfo);

    if lpItemID <> nil then
    begin
        SHGetPathFromIDList(lpItemID, TempPath);
        GlobalFreePtr(lpItemID);

        Result := TempPath;

    end else
    begin
        Result := '';

    end;

end;

/////////////////////////////////////////////////

//Створення імені файлу

function CreateFileName(Url : String) : String;
var
    i : Integer;

begin
    Result := '';

    if Pos('///', Url) > 0 then Delete(Url, 1, Pos('///', Url) + 1);

    if Url = '' then Exit;

    if Pos('/', Url) > 0 then Delete(Url, 1, Pos('/', Url))

```


Multimedia_Download_Internet_Download.pas - модуль завантажень

```

unit uDownload;

interface

uses
  Classes, uThreads, uObjects, uProcedures;

type
  // Class TLoadOne //

TLoadOne = class(TThread)
  public
    constructor Create(CreateSuspended : Boolean; P : Pointer);
  private
    Data : TTask;
  protected
    procedure Execute; override;
end;

implementation

uses uMain;

//Створення завантаження

constructor TLoadOne.Create(CreateSuspended : Boolean; P : Pointer);
begin
  Data:=P;
  inherited Create(CreateSuspended);
end;

//Редагування завантаження

procedure TLoadOne.Execute;
var
  ThreadHttp: TGetFileHttp;
  ThreadHttpOpt: TGetOptionsHttp;
  ThreadFtp: TGetFileFtp;
  ThreadFtpOpt: TGetOptionsFtp;
begin
  if (Data.Protocol=ptHttp) or (Data.Protocol=ptHttps)
  then
  begin
    ThreadHttpOpt:=TGetOptionsHttp.Create(True, Data);
    ThreadHttpOpt.Priority:=Options.Priority;
    ThreadHttpOpt.Resume;
    ThreadHttpOpt.WaitFor;
    ThreadHttpOpt.Free;
    if Data.Status=tsError
    then Exit;
    if (Data.TotalSize<0) and not (Options.ResumeLoad)
    then Exit;
    if Data.TotalSize>0
    then
    begin
      if Data.TotalSize>GetFreeSpace(Data.Directory)
      then Exit;
    end;
    ThreadHttp:=TGetFileHttp.Create(true, Data);
    ThreadHttp.Priority:=Options.Priority;
    ThreadHttp.Resume;
    ThreadHttp.WaitFor;
    ThreadHttp.Free;
  end;
end;

```

```
end;

if Data.Protocol=ptFtp
then
begin
  ThreadFtpOpt:=TGetOptionsFtp.Create(True, Data);
  ThreadFtpOpt.Priority:=Options.Priority;
  ThreadFtpOpt.Resume;
  ThreadFtpOpt.WaitFor;
  ThreadFtpOpt.Free;
  if Data.Status=tsError
  then Exit;
  if Data.TotalSize>0
  then
  begin
    if Data.TotalSize>GetFreeSpace(Data.Directory)
    then Exit;
  end;
  ThreadFtp:=TGetFileFtp.Create(True, Data);
  ThreadFtp.Priority:=Options.Priority;
  ThreadFtp.Resume;
  ThreadFtp.WaitFor;
  ThreadFtp.Free;
end;
fMain.tmUpdate.Enabled:=false;
fMain.RefreshTasks;
end;

end.
```

**Multimedia_Download_Internet>Loading.pas - візуалізація процесу
завантаження**

```

unit uLoading;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Gauges, ExtCtrls, uObjects, uProcedures;

type
  TfLoading = class(TForm)
    ProgressBar: TGauge;
    Button1: TButton;
    Button2: TButton;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Timer: TTimer;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure TimerTimer(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    Data : TTask;
  end;

var
  fLoading: TfLoading;

implementation

{$R *.dfm}

procedure TfLoading.Button1Click(Sender: TObject);
begin
  Close;
end;

procedure TfLoading.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action := caFree;
end;

procedure TfLoading.TimerTimer(Sender: TObject);
var
  TimeRemind : Integer;
begin
  Label11.Caption := FormatDateTime('hh:mm:ss', Data.TimeBegin - Now);
  if Data.TotalSize > 0 then
  begin
    Label7.Caption := BytesToText(Data.LoadSize) + ' з ' +
    BytesToText(Data.TotalSize);
    ProgressBar.MinValue := 0;
  end;
end;

```

```
ProgressBar.MaxValue := Data.TotalSize;
ProgressBar.Progress := Data.LoadSize;
if Data.Speed > 0 then
begin
    TimeRemind := (Data.TotalSize - Data.LoadSize) div Data.Speed;
    Label13.Caption := GetTimeStr(TimeRemind);
end;
end
else
begin
    ProgressBar.MinValue := 0;
    ProgressBar.MaxValue := 8;
    ProgressBar.AddProgress(1);
    if ProgressBar.Progress = 8 then ProgressBar.Progress := 0;
    Label7.Caption := BytesToText(Data.LoadSize);
end;
Label9.Caption := BytesToText(Data.Speed) + '/c';
if (Data.Status = tsLoad) then
begin
    Timer.Enabled := False;
    ProgressBar.Progress := ProgressBar.MaxValue;
    if Options.AutoCloseLoadingForm then Close;
end;
if (Data.Status = tsError) then
begin
    Timer.Enabled := False;
    ProgressBar.BackColor := clRed;
    if Options.AutoCloseLoadingForm then Close;
end;
end;

procedure TfLoading.FormShow(Sender: TObject);
begin
    Timer.Enabled := True;
    Label2.Caption := Data.LinkToFile;
end;

procedure TfLoading.Button2Click(Sender: TObject);
begin
    Data.Status := tsStoped;
    Close;
end;

end.
```

Multimedia_Download_Internet_Objects.pas - основні об'єкти

```

unit uObjects;

interface

uses
  Classes, XMLDoc, XMLIntf, SysUtils, Windows, ComCtrls, Menus;

type

// Class TTaskStatus //

TTaskStatus = (tsReady, tsError, tsLoad, tsLoading, tsStoped, tsDelete,
tsDeleted);

// Class THTTPVersion //

THTTPVersion = (hvHttp10, hvHttp11);

// Class TProtocol //

TProtocolType = (ptHttp, ptHttps, ptFtp);

// Class TCategory //

TCategory = record
  Name      : String;
  Directory : String;
  Extension : String;
end;

// Class TProxy //

TProxy = record
  Host       : String;
  Port       : Integer;
  UserName   : String;
  Password   : String;
  UseProxy   : Boolean;
end;

// Class TTask //

TTask = class(TObject)
public
  LinkToFile      : String;
  FileName        : String;
  Directory       : String;
  TotalSize       : Integer;
  LoadSize        : Integer;
  StartPosition   : Integer;
  EndPosition     : Integer;
  Login           : String;
  Password        : String;
  Port            : Integer;
  LastModified    : TDateTime;
  TimeBegin       : TDateTime;
  TimeEnd         : TDateTime;
  TimeTotal       : TDateTime;
  ScheduleOn      : Boolean;
  Speed           : Integer;
  Status          : TTaskStatus;
  Protocol        : TProtocolType;

```

```
        UseSpecial      : Boolean;
        ErrorText       : String;
        Category        : String;
        Description      : String;
end;

// Class TOptions //

TOptions = class(TObject)
public
    Name           : String;
    Version        : String;
    Path           : String;
    AgentName      : String;
    ShowLoadingForm : Boolean;
    AutoCloseLoadingForm : Boolean;
    HookClipboard  : Boolean;
    RunWithWindows : Boolean;
    MinToTray      : Boolean;
    MinOnRun       : Boolean;
    AlwaysInTray   : Boolean;
    UseProxyLocal  : Boolean;
    ResumeLoad     : Boolean;
    HTTPVersion    : THTTPVersion;
    Redirect       : Boolean;
    Priority        : TThreadPriority;
    HotKey         : TShortCut;
    HTTPProxy      : TProxy;
    FTPProxy       : TProxy;
    Url            : TStringList;
    Directory      : TStringList;
    Task           : TList;

    procedure Save;
    procedure Load;
end;

var
    Options : TOptions;

implementation

uses uMain;

////////////////////////////////////
//                               Class 'TOptions'                               //
////////////////////////////////////

////////////////////////////////////

procedure TOptions.Save;
var
    Xml      : TXMLDocument;
    Parent   : IXMLNode;
    Child    : IXMLNode;
    Value    : IXMLNode;
    i, n     : Integer;
    Data     : TTask;

begin

    Xml := TXMLDocument.Create(nil);
    Xml.Active := True;
```

```
if Xml.IsEmptyDoc then Xml.DocumentElement := Xml.CreateElement('XMLOptions',
'');
```

```
Xml.DocumentElement.Attributes['Name'] := Options.Name;
Xml.DocumentElement.Attributes['Version'] := Options.Version;
```

```
Parent := Xml.DocumentElement.AddChild('Options');
Child := Parent.AddChild('AgentName');
Child.Text := AgentName;
Child := Parent.AddChild('ShowLoadingForm');
Child.Text := BoolToStr(ShowLoadingForm);
Child := Parent.AddChild('AutoCloseLoadingForm');
Child.Text := BoolToStr(AutoCloseLoadingForm);
Child := Parent.AddChild('HookClipboard');
Child.Text := BoolToStr(HookClipboard);
Child := Parent.AddChild('MinToTray');
Child.Text := BoolToStr(MinToTray);
Child := Parent.AddChild('AlwaysInTray');
Child.Text := BoolToStr(AlwaysInTray);
Child := Parent.AddChild('MinOnRun');
Child.Text := BoolToStr(MinOnRun);
Child := Parent.AddChild('RunWithWindows');
Child.Text := BoolToStr(RunWithWindows);
Child := Parent.AddChild('UseProxyLocal');
Child.Text := BoolToStr(UseProxyLocal);
Child := Parent.AddChild('Redirect');
Child.Text := BoolToStr(Redirect);
Child := Parent.AddChild('ThreadPriority');
Child.Text := IntToStr(Integer(Priority));
Child := Parent.AddChild('HotKey');
Child.Text := ShortCutToText(HotKey);
Child := Parent.AddChild('HTTPVersion');
Child.Text := IntToStr(Integer(HTTPVersion));
Child := Parent.AddChild('ResumeLoad');
Child.Text := BoolToStr(ResumeLoad);
```

```
Parent := Xml.DocumentElement.AddChild('MainForm');
Child := Parent.AddChild('MainWindowTop');
Child.Text := IntToStr(fMain.Top);
Child := Parent.AddChild('MainWindowLeft');
Child.Text := IntToStr(fMain.Left);
Child := Parent.AddChild('MainWindowHeight');
Child.Text := IntToStr(fMain.Height);
Child := Parent.AddChild('MainWindowWidth');
Child.Text := IntToStr(fMain.Width);
Child := Parent.AddChild('CategoryPanelWidth');
Child.Text := IntToStr(fMain.Panel3.Width);
Child := Parent.AddChild('TasksPanelHeight');
Child.Text := IntToStr(fMain.Pane1.Height);
Child := Parent.AddChild('ParamsVisible');
Child.Text := BoolToStr(fMain.actParams.Checked);
Child := Parent.AddChild('StatusBarVisible');
Child.Text := BoolToStr(fMain.actStatusBar.Checked);
Child := Parent.AddChild('TasksColumn1Width');
Child.Text := IntToStr(fMain.lvTasks.Columns[1].Width);
Child := Parent.AddChild('TasksColumn2Width');
Child.Text := IntToStr(fMain.lvTasks.Columns[2].Width);
Child := Parent.AddChild('TasksColumn3Width');
Child.Text := IntToStr(fMain.lvTasks.Columns[3].Width);
Child := Parent.AddChild('TasksColumn4Width');
Child.Text := IntToStr(fMain.lvTasks.Columns[4].Width);
Child := Parent.AddChild('TasksColumn5Width');
Child.Text := IntToStr(fMain.lvTasks.Columns[5].Width);
Child := Parent.AddChild('ParamsColumn1Width');
Child.Text := IntToStr(fMain.lvParams.Columns[1].Width);
Child := Parent.AddChild('ParamsColumn2Width');
Child.Text := IntToStr(fMain.lvParams.Columns[2].Width);
Child := Parent.AddChild('SelectedTreeItem');
```

```

    if Assigned(fMain.tvFolders.Selected) then Child.Text :=
IntToStr(fMain.tvFolders.Selected.AbsoluteIndex) else Child.Text := '0';

    Parent := Xml.DocumentElement.AddChild('Proxy');

    Child := Parent.AddChild('HTTPProxy');

    Value := Child.AddChild('Host');
    Value.Text := HTTPProxy.Host;
    Value := Child.AddChild('Port');
    Value.Text := IntToStr(HTTPProxy.Port);
    Value := Child.AddChild('User');
    Value.Text := HTTPProxy.UserName;
    Value := Child.AddChild('Password');
    Value.Text := HTTPProxy.Password;
    Value := Child.AddChild('UseProxy');
    Value.Text := BoolToStr(HTTPProxy.UseProxy);

    Child := Parent.AddChild('FTPProxy');

    Value := Child.AddChild('Host');
    Value.Text := FTPProxy.Host;
    Value := Child.AddChild('Port');
    Value.Text := IntToStr(FTPProxy.Port);
    Value := Child.AddChild('User');
    Value.Text := FTPProxy.UserName;
    Value := Child.AddChild('Password');
    Value.Text := FTPProxy.Password;
    Value := Child.AddChild('UseProxy');
    Value.Text := BoolToStr(FTPProxy.UseProxy);

    if Url <> nil then
    begin

        Parent := Xml.DocumentElement.AddChild('Url');

        if Url.Count > 15 then n := 15 else n := Url.Count - 1;

        for i := 0 to n do
        begin

            Child := Parent.AddChild('Url');
            Child.Text := Url.Strings[i];

        end;
    end;

    if Directory <> nil then
    begin

        Parent := Xml.DocumentElement.AddChild('Directory');

        if Directory.Count > 15 then n := 15 else n := Directory.Count - 1;

        for i := 0 to n do
        begin

            Child := Parent.AddChild('Directory');
            Child.Text:= Directory.Strings[i];

        end;
    end;

    if Task <> nil then
    begin

        Parent := Xml.DocumentElement.AddChild('Tasks');

```

```

for i := 0 to Task.Count - 1 do
begin

    Data := Task.Items[i];

    if Data.Status <> tsDelete then
    begin

        Child := Parent.AddChild('Task');

        Value := Child.AddChild('LinkToFile');
        Value.Text := Data.LinkToFile;
        Value := Child.AddChild('FileName');
        Value.Text := Data.FileName;
        Value := Child.AddChild('Directory');
        Value.Text := Data.Directory;
        Value := Child.AddChild('TotalSize');
        Value.Text := IntToStr(Data.TotalSize);
        Value := Child.AddChild('LoadSize');
        Value.Text := IntToStr(Data.LoadSize);
        Value := Child.AddChild('StartPosition');
        Value.Text := IntToStr(Data.StartPosition);
        Value := Child.AddChild('EndPosition');
        Value.Text := IntToStr(Data.EndPosition);
        Value := Child.AddChild('Login');
        Value.Text := Data.Login;
        Value := Child.AddChild('Password');
        Value.Text := Data.Password;
        Value := Child.AddChild('Port');
        Value.Text := IntToStr(Data.Port);
        Value := Child.AddChild('LastModified');
        Value.Text := DateTimeToStr(Data.LastModified);
        Value := Child.AddChild('TimeBegin');
        Value.Text := DateTimeToStr(Data.TimeBegin);
        Value := Child.AddChild('TimeEnd');
        Value.Text := DateTimeToStr(Data.TimeEnd);
        Value := Child.AddChild('TimeTotal');
        Value.Text := DateTimeToStr(Data.TimeTotal);
        Value := Child.AddChild('ScheduleOn');
        Value.Text := BoolToStr(Data.ScheduleOn);
        Value := Child.AddChild('Speed');
        Value.Text := IntToStr(Data.Speed);
        Value := Child.AddChild('Status');
        Value.Text := IntToStr(Integer(Data.Status));
        Value := Child.AddChild('Protocol');
        Value.Text := IntToStr(Integer(Data.Protocol));
        Value := Child.AddChild('UseSpecial');
        Value.Text := BoolToStr(Data.UseSpecial);
        Value := Child.AddChild('ErrorText');
        Value.Text := Data.ErrorText;
        Value := Child.AddChild('Category');
        Value.Text := Data.Category;
        Value := Child.AddChild('Description');
        Value.Text := Data.Description;

    end;

end;

end;

    Xml.SaveToFile(Options.Path + '\ ' + Options.Name + '.xml');
    Xml.Free;

end;

////////////////////////////////////

```

```

procedure TOptions.Load;
var
  Xml      : IXMLDocument;
  Parent   : IXMLNode;
  Child    : IXMLNode;
  Value    : IXMLNode;
  i        : Integer;
  Data     : TTask;

begin
  Url := TStringList.Create;
  Directory := TStringList.Create;
  Task := TList.Create;

  if not FileExists(Options.Path + '\' + Options.Name + '.xml') then Exit;

  Xml := TXMLDocument.Create(nil);
  Xml.Active := True;
  Xml.LoadFromFile(Options.Path + '\' + Options.Name + '.xml');

  Parent := Xml.DocumentElement.ChildNodes['Options'];
  Child := Parent.ChildNodes['AgentName'];
  AgentName := Child.Text;
  Child := Parent.ChildNodes['ShowLoadingForm'];
  ShowLoadingForm := StrToBool(Child.Text);
  Child := Parent.ChildNodes['AutoCloseLoadingForm'];
  AutoCloseLoadingForm := StrToBool(Child.Text);
  Child := Parent.ChildNodes['HookClipboard'];
  HookClipboard := StrToBool(Child.Text);
  Child := Parent.ChildNodes['MinToTray'];
  MinToTray := StrToBool(Child.Text);
  Child := Parent.ChildNodes['MinOnRun'];
  MinOnRun := StrToBool(Child.Text);
  Child := Parent.ChildNodes['AlwaysInTray'];
  AlwaysInTray := StrToBool(Child.Text);
  Child := Parent.ChildNodes['RunWithWindows'];
  RunWithWindows := StrToBool(Child.Text);
  Child := Parent.ChildNodes['UseProxyLocal'];
  UseProxyLocal := StrToBool(Child.Text);
  Child := Parent.ChildNodes['Redirect'];
  Redirect := StrToBool(Child.Text);
  Child := Parent.ChildNodes['HotKey'];
  HotKey := TextToShortCut(Child.Text);
  Child := Parent.ChildNodes['ThreadPriority'];
  Priority := TThreadPriority(StrToInt(Child.Text));
  Child := Parent.ChildNodes['HTTPVersion'];
  HTTPVersion := THTTPVersion(StrToInt(Child.Text));
  Child := Parent.ChildNodes['ResumeLoad'];
  ResumeLoad := StrToBool(Child.Text);

  Parent := Xml.DocumentElement.ChildNodes['MainForm'];
  Child := Parent.ChildNodes['MainWindowTop'];
  fMain.Top := StrToInt(Child.Text);
  Child := Parent.ChildNodes['MainWindowLeft'];
  fMain.Left := StrToInt(Child.Text);
  Child := Parent.ChildNodes['MainWindowHeight'];
  fMain.Height := StrToInt(Child.Text);
  Child := Parent.ChildNodes['MainWindowWidth'];
  fMain.Width := StrToInt(Child.Text);
  Child := Parent.ChildNodes['CategoryPanelWidth'];
  fMain.Panel3.Width := StrToInt(Child.Text);
  Child := Parent.ChildNodes['TasksPanelHeight'];
  fMain.Panel1.Height := StrToInt(Child.Text);
  Child := Parent.ChildNodes['ParamsVisible'];
  fMain.Panel3.Visible := StrToBool(Child.Text);
  Child := Parent.ChildNodes['StatusBarVisible'];
  fMain.StatusBar.Visible := StrToBool(Child.Text);
  Child := Parent.ChildNodes['TasksColumn1Width'];

```

```

fMain.lvTasks.Columns[1].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['TasksColumn2Width'];
fMain.lvTasks.Columns[2].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['TasksColumn3Width'];
fMain.lvTasks.Columns[3].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['TasksColumn4Width'];
fMain.lvTasks.Columns[4].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['TasksColumn5Width'];
fMain.lvTasks.Columns[5].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['ParamsColumn1Width'];
fMain.lvParams.Columns[1].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['ParamsColumn2Width'];
fMain.lvParams.Columns[2].Width := StrToInt(Child.Text);
Child := Parent.ChildNodes['SelectedTreeItem'];
fMain.tvFolders.Items[StrToInt(Child.Text)].Selected := True;

Parent := Xml.DocumentElement.ChildNodes['Proxy'];

Child := Parent.ChildNodes[0];

Value := Child.ChildNodes['Host'];
HTTPProxy.Host := Value.Text;
Value := Child.ChildNodes['Port'];
HTTPProxy.Port := StrToInt(Value.Text);
Value := Child.ChildNodes['User'];
HTTPProxy.UserName := Value.Text;
Value := Child.ChildNodes['Password'];
HTTPProxy.Password := Value.Text;
Value := Child.ChildNodes['UseProxy'];
HTTPProxy.UseProxy := StrToBool(Value.Text);

Child := Parent.ChildNodes[1];

Value := Child.ChildNodes['Host'];
FTPProxy.Host := Value.Text;
Value := Child.ChildNodes['Port'];
FTPProxy.Port := StrToInt(Value.Text);
Value := Child.ChildNodes['User'];
FTPProxy.UserName := Value.Text;
Value := Child.ChildNodes['Password'];
FTPProxy.Password := Value.Text;
Value := Child.ChildNodes['UseProxy'];
FTPProxy.UseProxy := StrToBool(Value.Text);

Parent := Xml.DocumentElement.ChildNodes['Url'];

for i := 0 to Parent.ChildNodes.Count - 1 do
begin
    Child := Parent.ChildNodes[i];
    Url.Add(Child.Text);
end;

Parent := Xml.DocumentElement.ChildNodes['Directory'];

for i := 0 to Parent.ChildNodes.Count - 1 do
begin
    Child := Parent.ChildNodes[i];
    Directory.Add(Child.Text);
end;

Parent := Xml.DocumentElement.ChildNodes['Tasks'];

for i := 0 to Parent.ChildNodes.Count - 1 do
begin

```

```

Data := TTask.Create;

Child := Parent.ChildNodes[i];
Value := Child.ChildNodes['LinkToFile'];
Data.LinkToFile := Value.Text;
Value := Child.ChildNodes['FileName'];
Data.FileName := Value.Text;
Value := Child.ChildNodes['Directory'];
Data.Directory := Value.Text;
Value := Child.ChildNodes['TotalSize'];
Data.TotalSize := StrToInt(Value.Text);
Value := Child.ChildNodes['LoadSize'];
Data.LoadSize := StrToInt(Value.Text);
Value := Child.ChildNodes['StartPosition'];
Data.StartPosition := StrToInt(Value.Text);
Value := Child.ChildNodes['EndPosition'];
Data.EndPosition := StrToInt(Value.Text);
Value := Child.ChildNodes['Login'];
Data.Login := Value.Text;
Value := Child.ChildNodes['Password'];
Data.Password := Value.Text;
Value := Child.ChildNodes['Port'];
Data.Port := StrToInt(Value.Text);
Value := Child.ChildNodes['LastModified'];
Data.LastModified := StrToDateTime(Value.Text);
Value := Child.ChildNodes['TimeBegin'];
Data.TimeBegin := StrToDateTime(Value.Text);
Value := Child.ChildNodes['TimeEnd'];
Data.TimeEnd := StrToDateTime(Value.Text);
Value := Child.ChildNodes['TimeTotal'];
Data.TimeTotal := StrToDateTime(Value.Text);
Value := Child.ChildNodes['ScheduleOn'];
Data.ScheduleOn := StrToBool(Value.Text);
Value := Child.ChildNodes['Speed'];
Data.Speed := StrToInt(Value.Text);
Value := Child.ChildNodes['Status'];
Data.Status := TTaskStatus(StrToInt(Value.Text));
Value := Child.ChildNodes['Protocol'];
Data.Protocol := TProtocolType(StrToInt(Value.Text));
Value := Child.ChildNodes['UseSpecial'];
Data.UseSpecial := StrToBool(Value.Text);
Value := Child.ChildNodes['ErrorText'];
Data.ErrorText := Value.Text;
Value := Child.ChildNodes['Category'];
Data.Category := Value.Text;
Value := Child.ChildNodes['Description'];
Data.Description := Value.Text;

Task.Add(Data);

end;

Xml := nil;

end;

end.

```

Multimedia_Download_Internet_AddCategory.pas - створення нової категорії

```

unit uAddCategory;

interface

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

type
  TfAddCategory = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    edName: TEdit;
    edPath: TEdit;
    Label3: TLabel;
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
  public
    { Public declarations }
  end;

var
  fAddCategory: TfAddCategory;

implementation

uses uProcedures, uMain, uObjects;

{$R *.dfm}

procedure TfAddCategory.Button3Click(Sender: TObject);
begin
  Close;
end;

procedure TfAddCategory.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TfAddCategory.Button1Click(Sender: TObject);
begin
  edPath.Text:=BrowserFolder(Handle);
end;

procedure TfAddCategory.Button2Click(Sender: TObject);
var
  Node: TTreeNode;
begin
  if Trim(edName.Text)=''
  then
  begin
    MessageBox(Application.Handle, 'Не вказана назва категорії!',
    PChar(Options.Name), MB_OK or MB_ICONERROR);
    Exit;
  end;
end;

```

```
end;
if Trim(edPath.Text)=''
then
begin
  MessageBox(Application.Handle, 'Не вказане розміщення папки!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
  Exit;
end;
if not DirectoryExists(Trim(edPath.Text))
then
begin
  if MessageBox(Application.Handle, PChar('Папки "' + edPath.Text + '" не існує!
Створити?'), PChar(Options.Name), MB_OKCANCEL or MB_ICONERROR)=ID_OK
  then
  begin
    MkDir(Trim(edPath.Text));
  end
  else Exit;
end;
Node:=fMain.tvFolders.Items.AddChild(fMain.tvFolders.Selected, edName.Text);
Node.ImageIndex:=2;
Node.SelectedIndex:=2;
Close;
end;

end.
```

**Multimedia_Download_Internet_EditCategory.pas - редагування
категорії**

```

unit uEditCategory;

interface

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

type
  TfEditCategory = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    edName: TEdit;
    edPath: TEdit;
    Label3: TLabel;
    Memol: TMemo;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private

  public
    { Public declarations }
  end;

var
  fEditCategory: TfEditCategory;

implementation

uses uProcedures, uMain, uObjects;

{$R *.dfm}

procedure TfEditCategory.Button3Click(Sender: TObject);
begin
  Close;
end;

procedure TfEditCategory.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TfEditCategory.Button1Click(Sender: TObject);
begin
  edPath.Text:=BrowserFolder(Handle);
end;

procedure TfEditCategory.Button2Click(Sender: TObject);
begin
  if Trim(edName.Text)=''
  then
  begin
    MessageBox(Application.Handle, 'Не вказана назва категорії!',
    PChar(Options.Name), MB_OK or MB_ICONERROR);
    Exit;
  end;
end;

```

```
if Trim(edPath.Text)=''
then
begin
begin
MessageBox(Application.Handle, 'Не вказане розміщення папки!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
Exit;
end;
if not DirectoryExists(Trim(edPath.Text))
then
begin
if MessageBox(Application.Handle, PChar('Папки "' + edPath.Text + '" не
існує! Створити?'), PChar(Options.Name), MB_OKCANCEL or MB_ICONERROR)=ID_OK
then
begin
MkDir(Trim(edPath.Text));
end
else Exit;
end;
fMain.tvFolders.Selected.Text := Trim(edName.Text);
Close;
end;

procedure TfEditCategory.FormCreate(Sender: TObject);
begin
edName.Text:=fMain.tvFolders.Selected.Text;
end;

end.
```

Multimedia_Download_Internet_AddTask.pas - Створення задачі

```

unit uAddTask;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  StdCtrls, Clipbrd, Mask, Spin, Dialogs;

type
  TfAddTask = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    btnOK: TButton;
    btnCancel: TButton;
    Label3: TLabel;
    Label4: TLabel;
    edLogin: TEdit;
    edPassword: TEdit;
    cbxSpecial: TCheckBox;
    Label5: TLabel;
    Label9: TLabel;
    cbCategory: TComboBox;
    cbDirectory: TComboBox;
    Label10: TLabel;
    edFileName: TEdit;
    cbUrl: TComboBox;
    Label8: TLabel;
    edDescription: TEdit;
    sePort: TSpinEdit;
    GroupBox1: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnCancelClick(Sender: TObject);
    procedure cbxSpecialClick(Sender: TObject);
    procedure cbDirectoryClick(Sender: TObject);
    procedure cbDirectoryDropDown(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure cbUrlChange(Sender: TObject);
    procedure btnOKClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fAddTask: TfAddTask;

implementation

uses uObjects, uMain, uProcedures;

{$R *.dfm}

procedure TfAddTask.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TfAddTask.btnCancelClick(Sender: TObject);
begin
  Close;
end;

```

```

end;

procedure TfAddTask.cbxSpecialClick(Sender: TObject);
begin
  if cbxSpecial.Checked
  then
    begin
      edLogin.Enabled:=true;
      edPassword.Enabled:=true;
      sePort.Enabled:=true;
    end
  else
    begin
      edLogin.Enabled:=false;
      edPassword.Enabled:=false;
      sePort.Enabled:=false;
    end;
end;

procedure TfAddTask.cbDirectoryClick(Sender: TObject);
begin
  if cbDirectory.ItemIndex=0
  then
    begin
      cbDirectory.Items.Strings[0]:=BrowserFolder(Handle);
      cbDirectory.ItemIndex:=0;
    end;
end;

procedure TfAddTask.cbDirectoryDropDown(Sender: TObject);
begin
  cbDirectory.Items.Strings[0]:='Огляд папок...';
end;

procedure TfAddTask.FormCreate(Sender: TObject);
var
  ClipboardTemp: string;
  i: Integer;
begin
  if Clipboard.HasFormat(CF_TEXT)
  then
    begin
      ClipboardTemp:=Trim(Clipboard.AsText);
      if (Pos('http://', ClipboardTemp)=1)
        or (Pos('https://', ClipboardTemp)=1)
        or (Pos('ftp://', ClipboardTemp)=1)
      then cbUrl.Text:=ClipboardTemp;
      if Pos('www.', ClipboardTemp)=1
      then cbUrl.Text:='http://'+ClipboardTemp;
      if Pos('ftp.', ClipboardTemp)=1
      then cbUrl.Text:='ftp://'+ClipboardTemp;
    end;
  edFileName.Text:=CreateFileName(cbUrl.Text);
  cbUrl.Items:=Options.Url;
  sePort.Enabled:=false;
  for i:=0 to Options.Directory.Count-1 do
    begin
      cbDirectory.Items.Insert(1, Options.Directory.Strings[i]);
    end;
end;

procedure TfAddTask.cbUrlChange(Sender: TObject);
begin
  edFileName.Text:=CreateFileName(cbUrl.Text);
end;

procedure TfAddTask.btnOKClick(Sender: TObject);
var
  i, n: integer;

```

```

Data: TTask;
begin
  if cbUrl.Text=''
  then
    begin
      MessageBox(Application.Handle, 'Не вказане посилання!', PChar(Options.Name),
MB_OK or MB_ICONERROR);
      Exit;
    end;
  if cbDirectory.Text=''
  then
    begin
      MessageBox(Application.Handle, 'Не вказаний каталог для збереження файлу!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
  if edFileName.Text=''
  then
    begin
      MessageBox(Application.Handle, 'Не вказане ім'я файлу!', PChar(Options.Name),
MB_OK or MB_ICONERROR);
      Exit;
    end;
  if cbCategory.Text=''
  then
    begin
      MessageBox(Application.Handle, 'Не вказана категорія!', PChar(Options.Name),
MB_OK or MB_ICONERROR);
      Exit;
    end;
  if (Pos('http://', Trim(cbUrl.Text))=1)
  or (Pos('https://', Trim(cbUrl.Text))=1)
  or (Pos('ftp://', Trim(cbUrl.Text))=1)
  then
  else
    begin
      MessageBox(Application.Handle, 'Не вказаний протокол (http://, https://,
ftp://)!', PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
  if not DirectoryExists(Trim(cbDirectory.Text))
  then
    begin
      if MessageBox(Application.Handle, PChar('Папки "'+cbDirectory.Text + '" не
існує! Створити?'), PChar(Options.Name), MB_OKCANCEL or MB_ICONERROR)=IDOK
      then Mkdir(Trim(cbDirectory.Text))
      else Exit;
    end;
  if FileExists(Trim(cbDirectory.Text)+'\'+Trim(edFileName.Text))
  then
    begin
      if MessageBox(Application.Handle, PChar('Файл "'+cbDirectory.Text + '\'+
edFileName.Text + '" вже існує! Хотите перезаписати його?'),
PChar(Options.Name), MB_OKCANCEL or MB_ICONERROR)=IDCANCEL
      then Exit;
    end;
  n:=0;
  for i:=0 to Options.Url.Count-1 do
    begin
      if Options.Url.Strings[i]=Trim(cbUrl.Text)
      then n:=n+1;
    end;
  if n=0
  then Options.Url.Insert(0, Trim(cbUrl.Text));
  n:=0;
  for i:=0 to Options.Directory.Count-1 do
    begin
      if Options.Directory.Strings[i]=Trim(cbDirectory.Text)
      then n:=n+1;
    end;
  end;
end;

```

```
    end;
  if n=0
  then Options.Directory.Insert(0, Trim(cbDirectory.Text));
  Data:=TTask.Create;
  Data.LinkToFile:=Trim(cbUrl.Text);
  Data.FileName:=Trim(edFileName.Text);
  Data.Directory:=Trim(cbDirectory.Text);
  Data.Login:=Trim(edLogin.Text);
  Data.Password:=Trim(edPassword.Text);
  Data.Port:=sePort.Value;
  Data.UseSpecial:=cbxSpecial.Checked;
  Data.Description:=Trim(edDescription.Text);
  Data.Category:=Trim(edDescription.Text);
  Data.Status:=tsReady;
  Data.TotalSize:=0;
  Data.LoadSize:=0;
  Data.StartPosition:=0;
  Data.EndPosition:=0;
  Data.LastModified:=0;
  Data.TimeBegin:=0;
  Data.TimeEnd:=0;
  Data.TimeTotal:=0;
  Data.ErrorText:='';
  if Pos('http://', Data.LinkToFile)=1
  then Data.Protocol:=ptHttp;
  if Pos('https://', Data.LinkToFile)=1
  then Data.Protocol:=ptHttps;
  if Pos('ftp://', Data.LinkToFile)=1
  then Data.Protocol:=ptFtp;
  Options.Task.Add(Data);
  Close;
end;

end.
```

Multimedia_Download_Internet_EditTask.pas - редагування задачі

```

unit uEditTask;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  StdCtrls, Clipbrd, Mask, Spin, Dialogs, uObjects;

type
  TfEditTask = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    btnOK: TButton;
    btnCancel: TButton;
    Label3: TLabel;
    Label4: TLabel;
    edLogin: TEdit;
    edPassword: TEdit;
    cbxSpecial: TCheckBox;
    Label5: TLabel;
    Label9: TLabel;
    cbCategory: TComboBox;
    cbDirectory: TComboBox;
    Label10: TLabel;
    edFileName: TEdit;
    cbUrl: TComboBox;
    Label8: TLabel;
    edDescription: TEdit;
    sePort: TSpinEdit;
    GroupBox1: TGroupBox;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    rbSchedule: TRadioButton;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnCancelClick(Sender: TObject);
    procedure cbxSpecialClick(Sender: TObject);
    procedure cbDirectoryClick(Sender: TObject);
    procedure cbDirectoryDropDown(Sender: TObject);
    procedure cbUrlChange(Sender: TObject);
    procedure btnOKClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    Data : TTask;
  end;

var
  fEditTask: TfEditTask;

implementation

uses uMain, uProcedures;

{$R *.dfm}

procedure TfEditTask.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TfEditTask.btnCancelClick(Sender: TObject);
begin

```

```

Close;
end;

procedure TfEditTask.cbxSpecialClick(Sender: TObject);
begin
  if cbxSpecial.Checked
  then
    begin
      edLogin.Enabled:=true;
      edPassword.Enabled:=true;
      sePort.Enabled:=true;
    end
  else
    begin
      edLogin.Enabled:=false;
      edPassword.Enabled:=false;
      sePort.Enabled:=false;
    end;
end;

procedure TfEditTask.cbDirectoryClick(Sender: TObject);
begin
  if cbDirectory.ItemIndex=0
  then
    begin
      cbDirectory.Items.Strings[0]:=BrowserFolder(Handle);
      cbDirectory.ItemIndex:=0;
    end;
end;

procedure TfEditTask.cbDirectoryDropDown(Sender: TObject);
begin
  cbDirectory.Items.Strings[0]:='Перегляд папок...';
end;

procedure TfEditTask.cbUrlChange(Sender: TObject);
begin
  edFileName.Text:=CreateFileName(cbUrl.Text);
end;

procedure TfEditTask.btnOKClick(Sender: TObject);
var
  i, n: integer;
begin
  if cbUrl.Text=''
  then
    begin
      MessageBox(Application.Handle, 'Не зазначене посилання!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
  if cbDirectory.Text = '' then
    begin
      MessageBox(Application.Handle, 'Не зазначений каталог для збереження файлу!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
  if edFileName.Text = '' then
    begin
      MessageBox(Application.Handle, 'Не зазначене ім'я файлу!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
  if cbCategory.Text = '' then
    begin
      MessageBox(Application.Handle, 'Не зазначена категорія!',
PChar(Options.Name), MB_OK or MB_ICONERROR);
      Exit;
    end;
end;

```

```

if (Pos('http://', Trim(cbUrl.Text)) = 1)
  or (Pos('https://', Trim(cbUrl.Text)) = 1)
  or (Pos('ftp://', Trim(cbUrl.Text)) = 1)
then
else
  begin
    MessageBox(Application.Handle, 'Не зазначений протокол (http://, https://,
ftp://)!', PChar(Options.Name), MB_OK or MB_ICONERROR);
    Exit;
  end;
if not DirectoryExists(Trim(cbDirectory.Text)) then
  begin
    if MessageBox(Application.Handle, PChar('Папки "' + cbDirectory.Text + '" не
існує! Створити?'), PChar(Options.Name), MB_OKCANCEL or MB_ICONWARNING) = IDOK
      then MkDir(Trim(cbDirectory.Text)) else Exit;
    end;
    n := 0;
  for i := 0 to Options.Url.Count - 1 do
    begin
      if Options.Url.Strings[i] = Trim(cbUrl.Text) then n := n + 1;
    end;
    if n = 0 then Options.Url.Insert(0, Trim(cbUrl.Text));
    n := 0;
    for i := 0 to Options.Directory.Count - 1 do
      begin
        if Options.Directory.Strings[i] = Trim(cbDirectory.Text) then n := n + 1;
      end;
      if n = 0 then Options.Directory.Insert(0, Trim(cbDirectory.Text));
      Data.LinkToFile := Trim(cbUrl.Text);
      Data.FileName := Trim(edFileName.Text);
      Data.Directory := Trim(cbDirectory.Text);
      Data.Login := Trim(edLogin.Text);
      Data.Password := Trim(edPassword.Text);
      Data.Port := sePort.Value;
      Data.UseSpecial := cbxSpecial.Checked;
      Data.Description := Trim(edDescription.Text);
      if Pos('http://', Data.LinkToFile) = 1 then Data.Protocol := ptHttp;
      if Pos('https://', Data.LinkToFile) = 1 then Data.Protocol := ptHttps;
      if Pos('ftp://', Data.LinkToFile) = 1 then Data.Protocol := ptFtp;
      Close;
    end;

////////////////////////////////////

procedure TfEditTask.FormShow(Sender: TObject);
var
  i : Integer;
begin
  cbUrl.Text := Data.LinkToFile;
  edFileName.Text := Data.FileName;
  cbUrl.Items := Options.Url;
  for i := 0 to Options.Directory.Count - 1 do
    begin
      cbDirectory.Items.Insert(1, Options.Directory.Strings[i]);
    end;
    cbDirectory.Text := Data.Directory;
    cbCategory.Text := Data.Category;
    edDescription.Text := Data.Description;
    cbxSpecial.Checked := Data.UseSpecial;
  if Data.UseSpecial
  then
    begin
      edLogin.Enabled := True;
      edPassword.Enabled := True;
      sePort.Enabled := True;
    end
  else
    begin
      edLogin.Enabled := False;

```

```
edPassword.Enabled := False;  
sePort.Enabled := False;  
end;  
edLogin.Text := Data.Login;  
edPassword.Text := Data.Password;  
sePort.Value := Data.Port;  
if Data.ScheduleOn then rbSchedule.Checked:=true;  
end;  
  
end.
```

К6П3-2023

Multimedia_Download_Internet_Options.pas - параметри програми

```

unit uOptions;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls, Spin, Registry;

type
  TfOptions = class(TForm)
    btnOK: TButton;
    btnCancel: TButton;
    PageControll: TPageControl;
    TabSheet1: TTabSheet;
    Label1: TLabel;
    cbAgent: TComboBox;
    TabSheet3: TTabSheet;
    cbxRunWithWindows: TCheckBox;
    cbxHookClipboard: TCheckBox;
    cbxMinToTray: TCheckBox;
    Label23: TLabel;
    mmExtention: TMemo;
    Label15: TLabel;
    TrackBar: TTrackBar;
    Label16: TLabel;
    CheckBox4: TCheckBox;
    CheckBox5: TCheckBox;
    cbxAutoCloseLoadingForm: TCheckBox;
    cbxShowLoadingForm: TCheckBox;
    cbxMinOnRun: TCheckBox;
    cbxRedirect: TCheckBox;
    Label6: TLabel;
    hkApplication: THotKey;
    Label7: TLabel;
    cbHttpVersion: TComboBox;
    cbxResumeLoad: TCheckBox;
    TabSheet2: TTabSheet;
    cbxUseProxyLocal: TCheckBox;
    cbUseHTTPProxy: TCheckBox;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    Label3: TLabel;
    edHTTPProxyHost: TEdit;
    edHTTPProxyPort: TEdit;
    Label4: TLabel;
    Label5: TLabel;
    edHTTPProxyUser: TEdit;
    edHTTPProxyPass: TEdit;
    GroupBox2: TGroupBox;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    edFTPProxyHost: TEdit;
    edFTPProxyPort: TEdit;
    edFTPProxyUser: TEdit;
    edFTPProxyPass: TEdit;
    cbUseFTPProxy: TCheckBox;
    cbxAlwaysInTray: TCheckBox;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnCancelClick(Sender: TObject);
    procedure btnOKClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure TrackBarChange(Sender: TObject);
  private

```

```

    { Private declarations }
public
    { Public declarations }
end;

var
    fOptions: TfOptions;

implementation

uses uObjects, uMain;

{$R *.dfm}

procedure TfOptions.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action:=caFree;
end;

procedure TfOptions.btnCancelClick(Sender: TObject);
begin
    Close;
end;

procedure TfOptions.btnOKClick(Sender: TObject);
var
    Registry : TRegistry;
begin
    Options.AgentName := Trim(cbAgent.Text);
    Options.ShowLoadingForm := cbxShowLoadingForm.Checked;
    Options.AutoCloseLoadingForm := cbxAutoCloseLoadingForm.Checked;
    Options.HookClipboard := cbxHookClipboard.Checked;
    Options.MinToTray := cbxMinToTray.Checked;
    Options.MinOnRun := cbxMinOnRun.Checked;
    Options.AlwaysInTray := cbxAlwaysInTray.Checked;
    Options.RunWithWindows := cbxRunWithWindows.Checked;
    Options.HotKey := hkApplication.HotKey;
    Options.Redirect := cbxRedirect.Checked;
    Options.UseProxyLocal := cbxUseProxyLocal.Checked;
    Options.ResumeLoad := cbxResumeLoad.Checked;
    Options.HTTPProxy.UseProxy := cbUseHTTPProxy.Checked;
    Options.HTTPProxy.Host := Trim(edHTTPProxyHost.Text);
    Options.HTTPProxy.Port := StrToInt(Trim(edHTTPProxyPort.Text));
    Options.HTTPProxy.UserName := Trim(edHTTPProxyUser.Text);
    Options.HTTPProxy.Password := Trim(edHTTPProxyPass.Text);
    Options.FTPProxy.UseProxy := cbUseFTPProxy.Checked;
    Options.FTPProxy.Host := Trim(edFTPProxyHost.Text);
    Options.FTPProxy.Port := StrToInt(Trim(edFTPProxyPort.Text));
    Options.FTPProxy.UserName := Trim(edFTPProxyUser.Text);
    Options.FTPProxy.Password := Trim(edFTPProxyPass.Text);

    if cbHttpVersion.Text = 'HTTP 1.0' then Options.HTTPVersion := hvHttp10 else
Options.HTTPVersion := hvHttp11;

    case TrackBar.Position of

        0 : Options.Priority := tpIdle;
        1 : Options.Priority := tpLowest;
        2 : Options.Priority := tpLower;
        3 : Options.Priority := tpNormal;
        4 : Options.Priority := tpHigher;
        5 : Options.Priority := tpHighest;
        6 : Options.Priority := tpTimeCritical;

    end;

Options.Save;

Registry := TRegistry.Create(KEY_ALL_ACCESS);

```

```

try
    Registry.RootKey := HKEY_LOCAL_MACHINE;
    Registry.OpenKey('\Software\Microsoft\Windows\CurrentVersion\Run', True);

    if cbxRunWithWindows.Checked then Registry.WriteString('Download',
Application.ExeName)
    else Registry.DeleteValue('Download');

    Registry.CloseKey;

finally

    Registry.Free;

end;

Close;

end;

procedure TfOptions.FormCreate(Sender: TObject);
begin

    cbAgent.Text := Options.AgentName;
    cbxShowLoadingForm.Checked := Options.ShowLoadingForm;
    cbxAutoCloseLoadingForm.Checked := Options.AutoCloseLoadingForm;
    cbxHookClipboard.Checked := Options.HookClipboard;
    cbxMinToTray.Checked := Options.MinToTray;
    cbxMinOnRun.Checked := Options.MinOnRun;
    cbxAlwaysInTray.Checked := Options.AlwaysInTray;
    cbxRunWithWindows.Checked := Options.RunWithWindows;
    hkApplication.HotKey := Options.HotKey;
    cbxRedirect.Checked := Options.Redirect;
    cbxUseProxyLocal.Checked := Options.UseProxyLocal;
    cbxResumeLoad.Checked := Options.ResumeLoad;
    cbUseHTTPProxy.Checked := Options.HTTPProxy.UseProxy;
    edHTTPProxyHost.Text := Options.HTTPProxy.Host;
    edHTTPProxyPort.Text := IntToStr(Options.HTTPProxy.Port);
    edHTTPProxyUser.Text := Options.HTTPProxy.UserName;
    edHTTPProxyPass.Text := Options.HTTPProxy.Password;
    cbUseFTPProxy.Checked := Options.FTPProxy.UseProxy;
    edFTPProxyHost.Text := Options.FTPProxy.Host;
    edFTPProxyPort.Text := IntToStr(Options.FTPProxy.Port);
    edFTPProxyUser.Text := Options.FTPProxy.UserName;
    edFTPProxyPass.Text := Options.FTPProxy.Password;

    if Options.HTTPVersion = hvHttp10 then cbHttpVersion.ItemIndex := 0 else
cbHttpVersion.ItemIndex := 1;

    case Options.Priority of

        tpIdle          : begin TrackBar.Position := 0; Label16.Caption := 'Низький';
end;
        tpLowest        : begin TrackBar.Position := 1; Label16.Caption := 'Нижче
середнього'; end;
        tpLower         : begin TrackBar.Position := 2; Label16.Caption :=
'Середній'; end;
        tpNormal        : begin TrackBar.Position := 3; Label16.Caption := 'Вище за
середне'; end;
        tpHigher        : begin TrackBar.Position := 4; Label16.Caption := 'Високий';
end;
        tpHighest       : begin TrackBar.Position := 5; Label16.Caption := 'Вільш
високий'; end;
        tpTimeCritical : begin TrackBar.Position := 6; Label16.Caption := 'Реального
часу'; end;
        end;
end;
end;

```

```
procedure TfOptions.TrackBarChange(Sender: TObject);
begin
  case TrackBar.Position of
    0: Label16.Caption := 'Низький';
    1: Label16.Caption := 'Нижче середнього';
    2: Label16.Caption := 'Середній';
    3: Label16.Caption := 'Вище за середнє';
    4: Label16.Caption := 'Високий';
    5: Label16.Caption := 'Більш високий';
    6: Label16.Caption := 'Реального часу';
  end;
end;

end.
```

КБГПЗ - 2023

Multimedia_Download_Internet_Threads.pas - довідка

```

unit uAbout;

interface

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

type
  TfAbout = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Image1: TImage;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fAbout: TfAbout;
implementation

uses uObjects;
{$R *.dfm}
procedure TfAbout.Button1Click(Sender: TObject);
begin
  Close;
end;
procedure TfAbout.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
end;

procedure TfAbout.FormCreate(Sender: TObject);
begin
  Label1.Caption:=Options.Name;
  Memo1.Clear;
  Memo1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' Дослідження та програмна реалізація системи
багатофункціонального менеджера завантажень мультимедіа даних з мережі
Internet');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Буравченко К.О. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Мікіньов Ігор Юрійович');
  Memo1.Lines.Add(' гр. КН-22М-2');
  Memo1.Lines.Add('');
  Memo1.Lines.Add(' Кропивницький 2023');
  Memo1.Lines.Add('');
end;
end;
end.

```