

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  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“Програмне забезпечення системи download менеджера”**

Виконав здобувач вищої освіти  
IV курсу, групи КІ-20  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Абдурашітов Ш.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат фізико-математичних наук, доцент  
\_\_\_\_\_ Петренюк В.І.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

м. Кропивницький

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2024 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Абдурашітову Шахруху

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи download менеджера

2. Керівник роботи Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент

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

затверджені наказом вищого навчального закладу № 131-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту 23.05.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи download менеджера

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 17 » січня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання  
« 17 » січня 2024 р.

Підпис керівника

Петренюк В.І.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2024 р.

Підпис здобувача

Абдурашітов Ш.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Абдурашітов Ш. Програмне забезпечення системи download менеджера. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи download менеджера.

Метою розробки є програмне забезпечення системи download менеджера.

Результат роботи – програмна реалізація системи download менеджера.

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 з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4 Sydney.

**Ключові слова:** комп'ютерна інженерія, менеджер завантажень

## ABSTRACT

**Abdurashitov Sh. Software download manager system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the first (bachelor) level of higher education, software designed for the download manager system was developed.

The goal of the development is the download manager system software.

The result of the work is the software implementation of the download manager system.

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

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

The program can be used on a PC with OS Windows 10/11.

The program was developed in the Delphi 10.4 Sydney environment.

**Keywords:** computer engineering, download manager

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	24
2.3 Розгорнута постановка завдання .....	30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	31
3.1 Опис функціонування системи .....	31
3.2 Розробка структурної схеми.....	34
3.3 Розробка функціональної схеми .....	37
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	59
6 ОСНОВНІ ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	64

						ВКРБ-123.24.0083.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Абдураїмов Ш.				Програмне забезпечення системи download менеджера	Літ.	Аркуш	Аркушів
Перев.	Петренко В.І.					Б	1	70
Н.контр.	Коваленко А.С.				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

## 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

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ВСТУП

**Актуальність теми.** Менеджери закачування (download менеджери часто називані й менеджерами завантаження) значно полегшують процес закачування 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 закачування після того, як серфінг закінчений. Таким чином, закачування файлів за допомогою менеджера може бути виконана в той час, коли користувач не працює в інтернеті. Багато менеджерів закачування дозволяють указати час, коли необхідно скачати файли, що буває зручно при використанні діал-ап доступу в інтернет: список файлів для закачування можна сформувати вдень і запланувати закачування на нічний час, коли діє більше дешевий тариф на послуги доступу в мережу. У цьому випадку, менеджер сам додзвониться до провайдера, скачає файли й виключить комп'ютер. Крім цього, менеджери можуть докачувати файли, тобто, якщо закачування було перервано й файл не був закачаний повністю, те менеджер після повторного з'єднання із сервером продовжить закачування файлу з того місця, де була перерване закачування, а не буде закачувати весь файл заново. На цьому список можливостей менеджерів закачування не закінчується, вони можуть інтегруватися із браузером, автоматично знижувати швидкість закачування, коли користувач працює в інтернеті, зберігати облікові дані для доступу до певних серверів і так далі.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи download менеджера.

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

- Огляд існуючих систем download менеджера.
- Дослідження системи download менеджера.
- Програмна реалізація системи download менеджера.

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі download менеджера.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи download менеджера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ\_2024

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для завантаженням файлів з мережі, як локальної або корпоративної, так і глобальної, тобто з Інтернету.

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 Internet Explorer, 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: убудований 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 завантажень, наприклад, завантаження файлів з файлообмінних систем на зразок Rapidshare.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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 якої, 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 завантажуете 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%.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 1.2 Область застосування

Областю застосування є завантаження файлів з інтернету, а у принципі, й з довільної мережі. Стандартний менеджер завантажувач 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.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи download менеджера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	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 першим (бакалаврським) рівнем вищої освіти

У цьому розділі розглянемо основні менеджери закахувань і їхні характеристики

#### **Download Accelerator Plus**

Розмір дистрибутива 3 мегабайти, безкоштовна версія з обмеженням по кількості потоків (10), з рекламою, відключеними можливостями по роботі з ZIP-архівами й іншими обмеженнями.

Основні можливості:

- Протоколи: FTP, HTTP.
- Робота через HTTP і FTP проксі-сервера.
- Інтеграція в браузер.
- Автоматичний пошук дзеркал.
- Часткове завантаження ZIP-архівів, перегляд їхнього вмісту.
- Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні закахування, персональні настроювання для закахувань із певних серверів).
- FTP-браузер.
- Вбудований плеєр для перегляду або прослуховування закахуємих відео або аудіо файлів.
- Планувальник закахувань.
- Автоматична перевірка на віруси закаханих файлів.
- Можливість зберігати завантажуються файли, що, на сайті [dapdrive.com](http://dapdrive.com)
- Каталогізація файлів за допомогою Медіа-центра.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

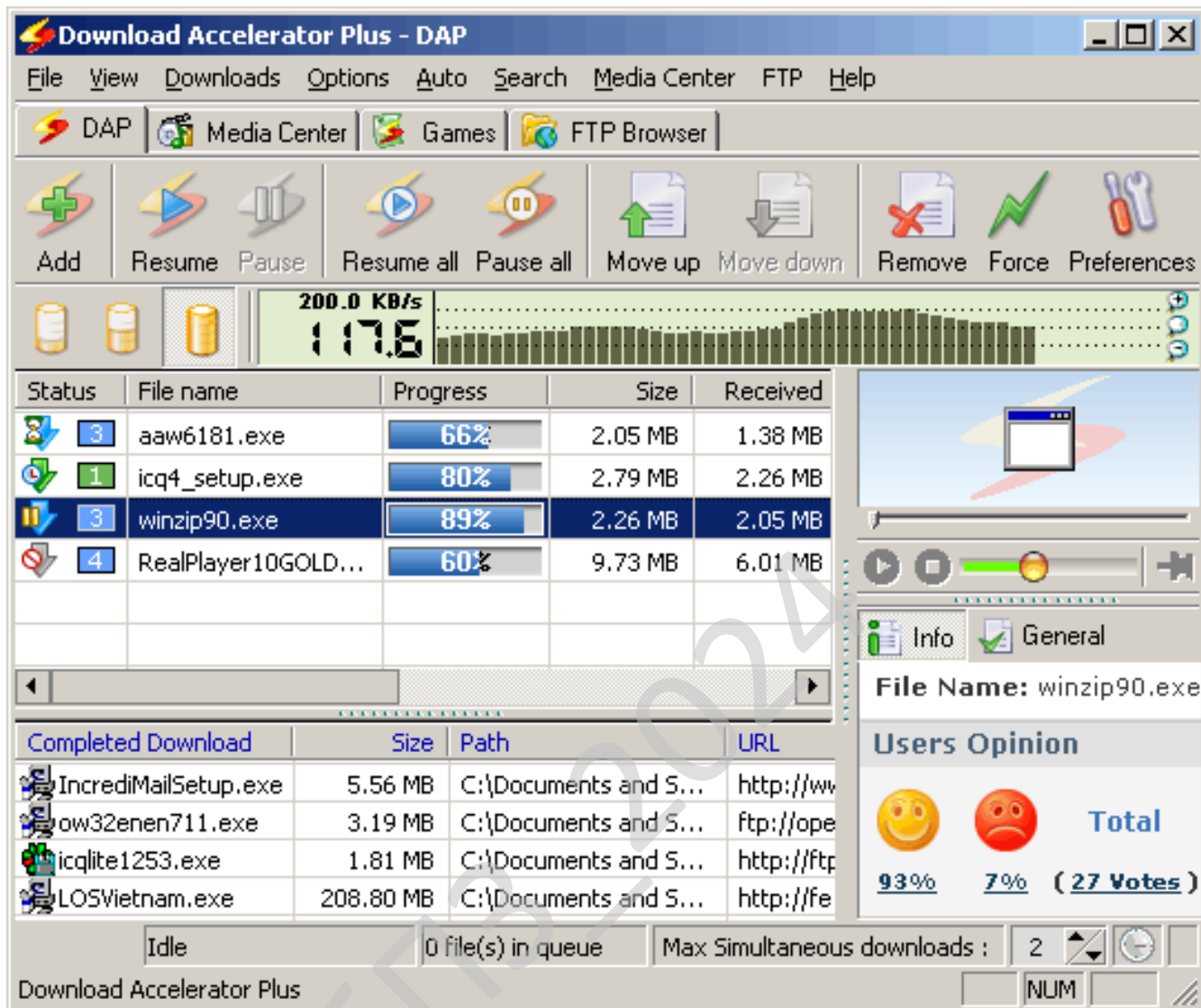


Рисунок 2.1 – Інтерфейс користувача Download Accelerator Plus

### Download Master

Розмір дистрибутива 2 мегабайти, безкоштовний.

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP.
- Робота через HTTP і FTP проксі-сервера, підтримка NTLM і NTLM-проксі автентифікації.
- Вбудований FTP-браузер.

– Автоматичний пошук дзеркал (інших серверів, звідки можна скачати файл).

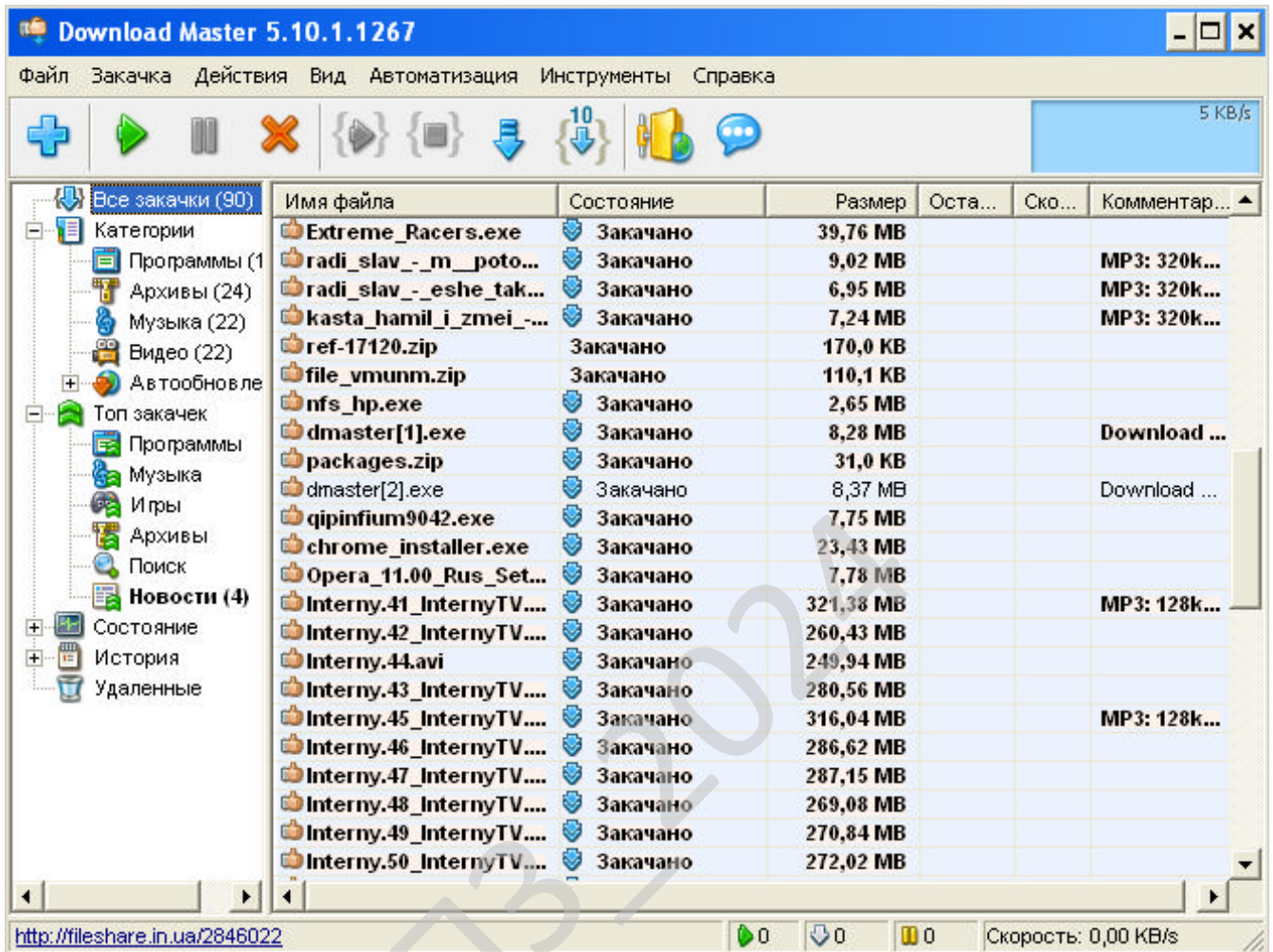


Рисунок 2.2 – Інтерфейс користувача Download Master

- Часткове закачування ZIP-архівів, перегляд їхнього вмісту перед закачуванням.
- Підтримка архівів RAR: можливість перевіряти й розпаковувати.
- Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні закачування, персональні налаштування для закачувань із певних серверів).
- Зміна швидкості для одночасного з закачуванням перегляду сторінок, автоматичне визначення активності ІЕ і зниження швидкості закачувань.

- Можливість змінювати пріоритет закачування.
- Підтримка командного рядка.
- Синхронізація файлів на комп'ютері із сервером.
- Перевірка заколисаних файлів на відновлення.
- Можливість прослухати/переглянути музичні й відео файли в процесі закачування. Автоматичне одержання інформації про MP3-файлах при старті закачування.
- Автоматична перевірка на віруси закачаних файлів.
- Інтеграція в браузері.

### Free Download Manager

Розмір дистрибутива 1,3 мегабайти, безкоштовний.

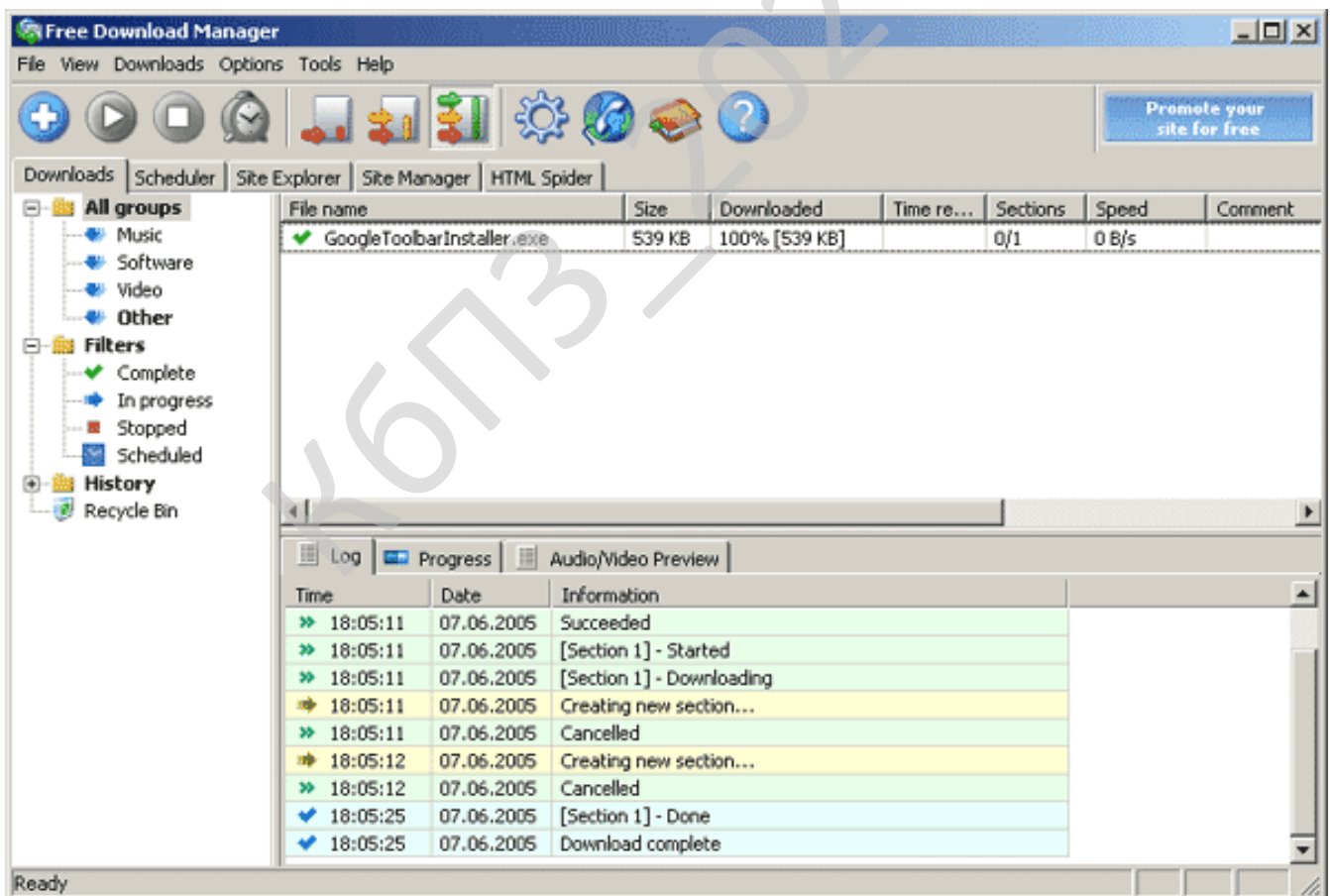


Рисунок 2.3 – Інтерфейс користувача Free Download Manager

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP.
- Вбудовані FTP і WEB браузері.
- Планувальник закичувань.
- 3 режими для обмеження швидкості всіх закичувань.
- Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні закичування, персональні настроювання для закичувань із певних серверів).
- Закичування сайтів цілком або тільки файлів з певними розширеннями.

### Fresh Download

Розмір дистрибутива 1,5 мегабайта, безкоштовний.

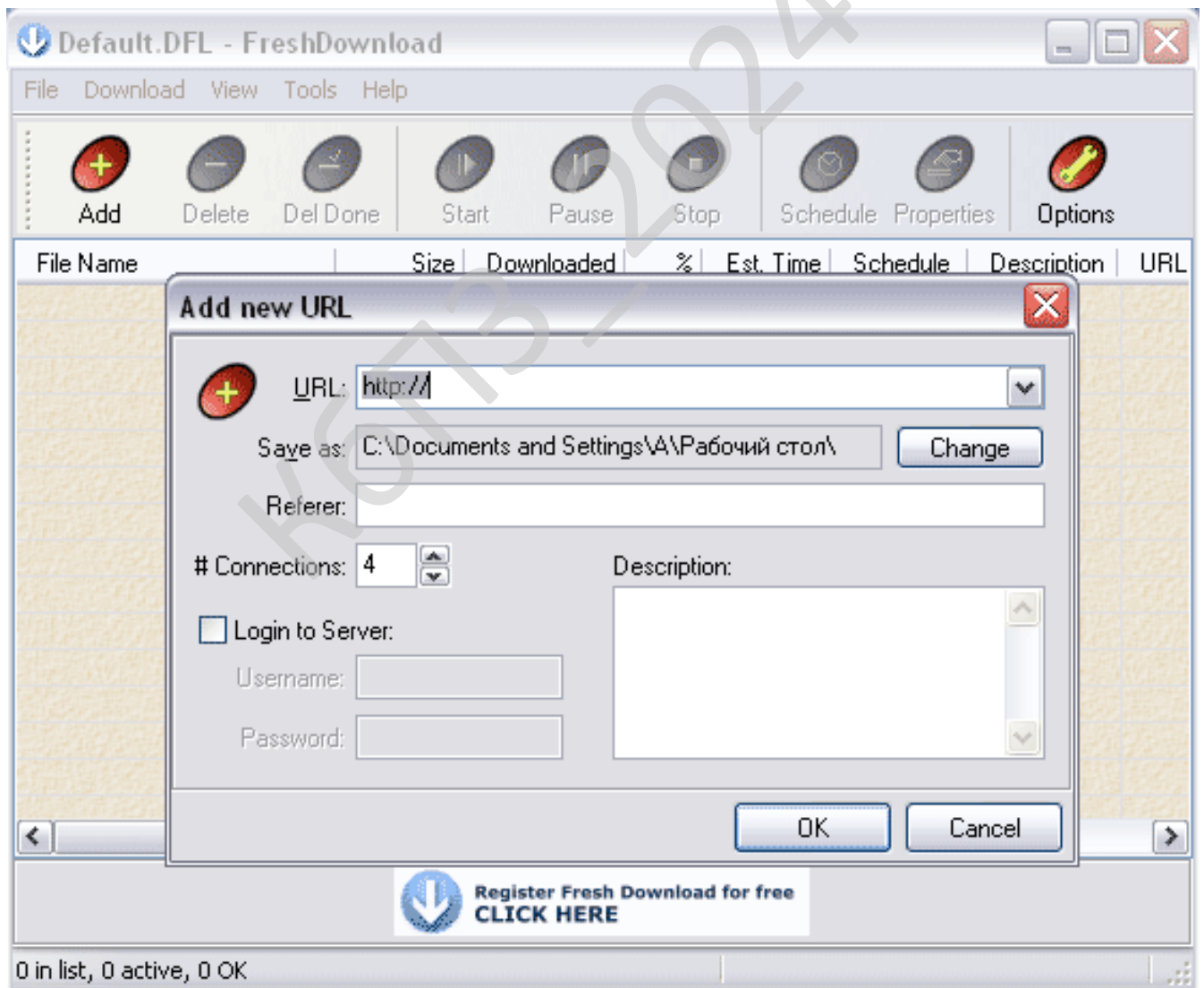


Рисунок 2.4 – Інтерфейс користувача Fresh Download

Основні можливості:

- Протоколи: HTTP, FTP.
- Робота через проксі-сервера.
- Планувальник закичувань.
- Автоматична перевірка на віруси закичаних файлів.
- Інтеграція в браузер.
- Часткове закичування ZIP-архівів, перегляд їхнього вмісту перед закичуванням.
- Можливість укаати Referrer для кожного закичування.

### FlashGet

Розмір дистрибутива 1,6 мегабайта, безкоштовна версія з рекламою й нагадуваннями.

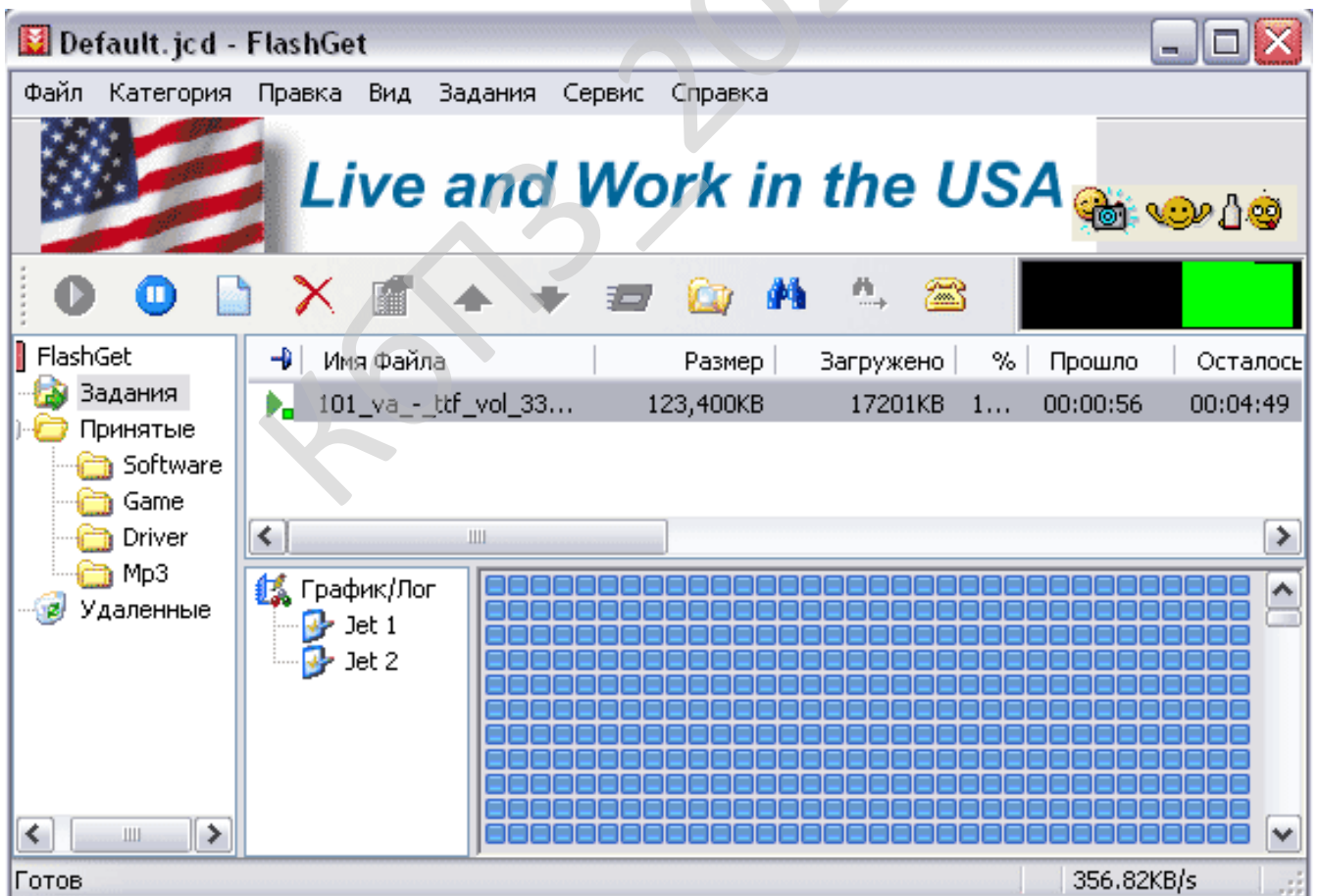


Рисунок 2.5 – Інтерфейс користувача FlashGet

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP, MMS, RTSP.
- Робота через проксі-сервери.
- Вбудовані FTP і WEB браузери.
- Підтримка командного рядка.
- Обмеження швидкості завантажень.
- Автоматичний пошук дзеркал (інших серверів, звідки можна скачати файл).
- Категорії файлів.
- Правила автоматичного сортування файлів (при завантаженні з певного сайту й/або з певним розширенням).
- Завантаження файлів по масці.
- Планувальник завантажень.
- Можливість указати Referrer для кожного завантаження (у русифікації помилка – це поле у властивостях завантаження назване *Посилання*).
- Вибір проксі-сервера для кожного завантаження.
- Можливість указати ім'я завантаженого файлу, з яким він буде збережений на диску.
- Інтеграція в браузері.

### **GetRight**

Розмір дистрибутива 3,2 мегабайти, безкоштовна версія з обмеженням розміру черги завантаження, щоденним обмеженням кількості завантаженої інформації й нагадуваннями.

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP.
- Робота через проксі-сервери.
- Автоматичний пошук дзеркал (інших серверів, звідки можна скачати файл).
- Вбудовані FTP і WEB браузері.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

- Автоматизація (закачування декількох файлів за раз, початок закачування одного файлу після закінчення закачування іншого й т.д.).
- Планувальник закачувань.
- Категорії файлів (можливість зберегти файли різних типів у різних папках).
- Закачування багатьох файлів із сайту або FTP-серверу.
- Обмеження швидкості закачувань.
- Щоденні заплановані закачування (відновлення антивірусних баз).
- Заплановане автоматичне закачування нових файлів (перевірка версій).
- Інтеграція в браузері.

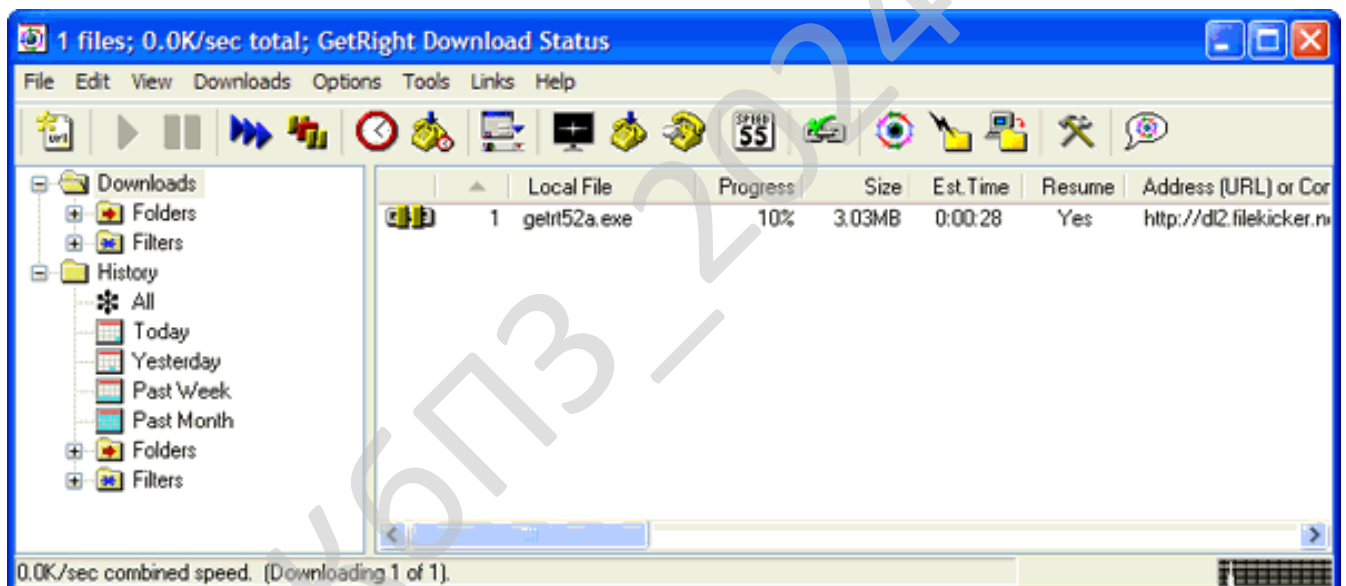


Рисунок 2.6 – Інтерфейс користувача GetRight

### Go!Zilla

Розмір дистрибутива 4 мегабайти, демонстраційна версія (30 днів) містить рекламний модуль.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15



Рисунок 2.7 – Інтерфейс користувача Go!Zilla

Основні можливості:

- Автоматичний пошук дзеркал (інших серверів, звідки можна скачати файл).
- Автоматичний запуск установки закачаного програмного забезпечення.

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- Категорії файлів (можливість зберегти файли різних типів у різних папках).
- Пошук файлів в інтернеті.
- Пошук посилання на файл, якщо була додана в чергу біте посилання або запитуваний файл було віддалено на сервері.
- Інтеграція в браузері. Інтеграція в плеєр (у вибраному автоматично з'являються посилання на закачані аудіо й відео файли).
- Розархівація ZIP-архівів.
- Закачування всіх файлів зі сторінки.
- Планувальник закачувань.

### HiDownload

Розмір дистрибутива 2,2 мегабайти, демонстраційна версія (30 днів).

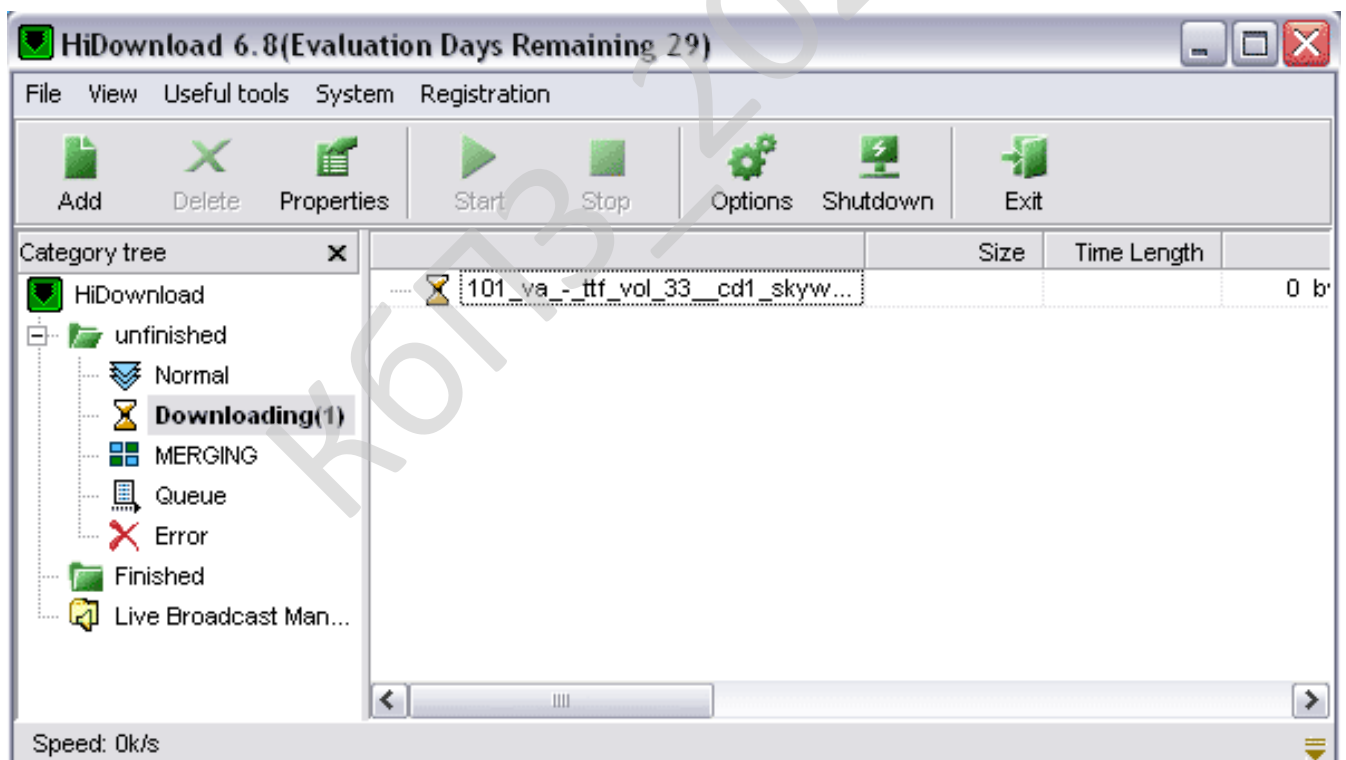


Рисунок 2.8 – Інтерфейс користувача HiDownload

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP, RTSP, MMS.
- Робота через HTTP, FTP, SOCKS проксі-сервера.
- Можливість указати ім'я завантажуваного файлу, з яким він буде збережений

на диску.

- Завантаження безлічі файлів із сайту або FTP-сервера.
- Планувальник завантажень.
- Категорії файлів (можливість зберегти файли різних типів у різних папках).
- Можливість указати Referrer для кожного завантаження.
- Захват і збереження на диску потокового відео, аудіо.
- Вбудований FTP браузер.
- Автоматична перевірка на віруси завантажених файлів.
- Інтеграція в браузер.

### **Internet Download Manager**

Розмір дистрибутива 1,2 мегабайти, демонстраційна версія (30 днів).

Основні можливості:

- Робота через проксі-сервери, підтримка Basic, Negotiate, NTLM і Keberos автентифікації.
- Завантаження всіх файлів зі сторінки.
- Категорії файлів.
- Планувальник завантажень.
- Автоматична перевірка на віруси завантажених файлів.
- Інтеграція в браузер.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

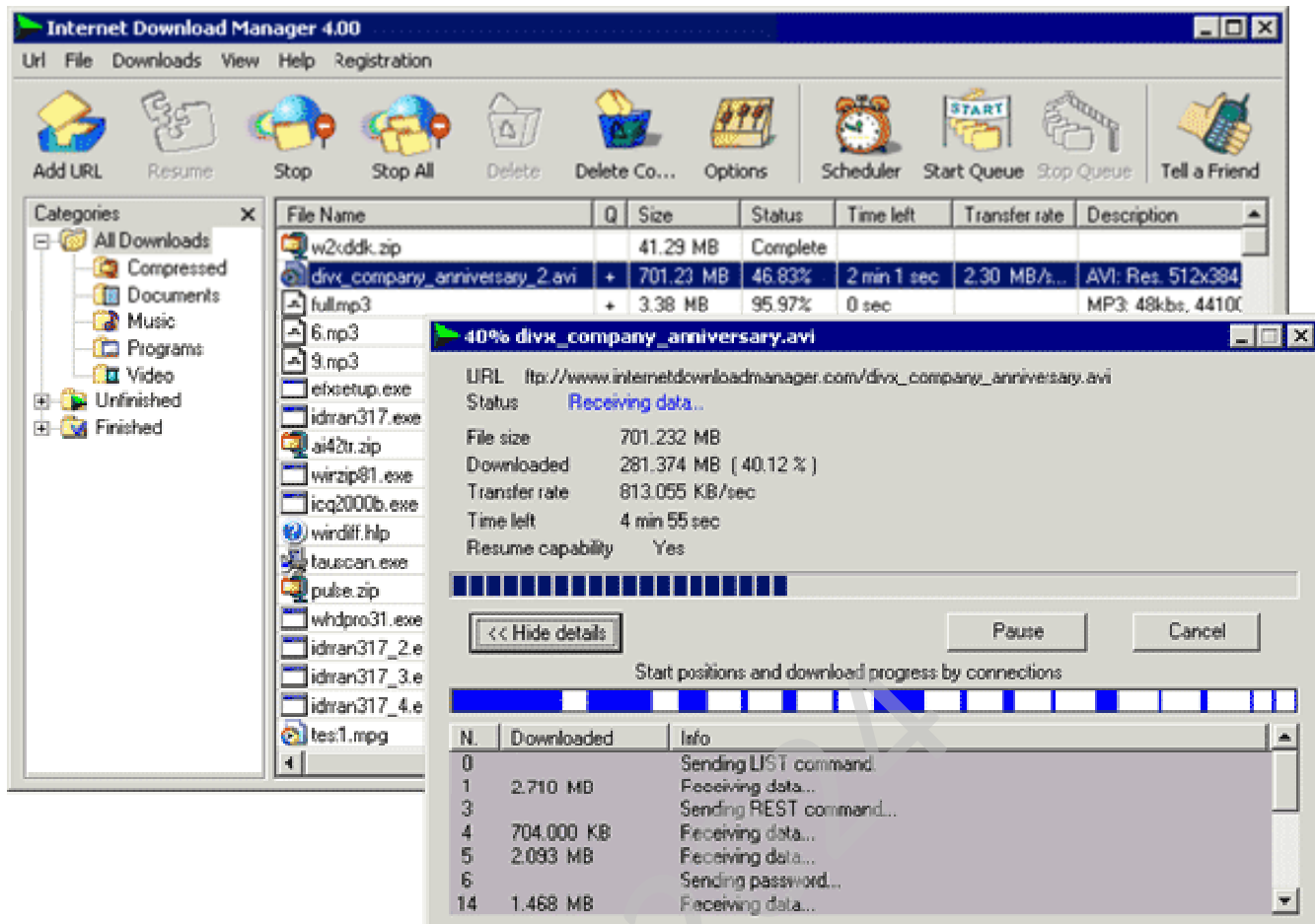


Рисунок 2.9 – Інтерфейс користувача Internet Download Manager

### Mass Downloader

Розмір дистрибутива 1,5 мегабайта, демонстраційна версія (30 днів).

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP, RTSP, MMS.
- Робота через HTTP, SOCKS, RTSP проксі-сервера.
- Категорії файлів (можливість зберегти файли різних типів у різних папках).
- Вбудований FTP браузер.
- Часткове закачування ZIP-архівів, перегляд їхнього вмісту перед закачуванням.
- Планувальник закачувань.
- Інтеграція в браузери.

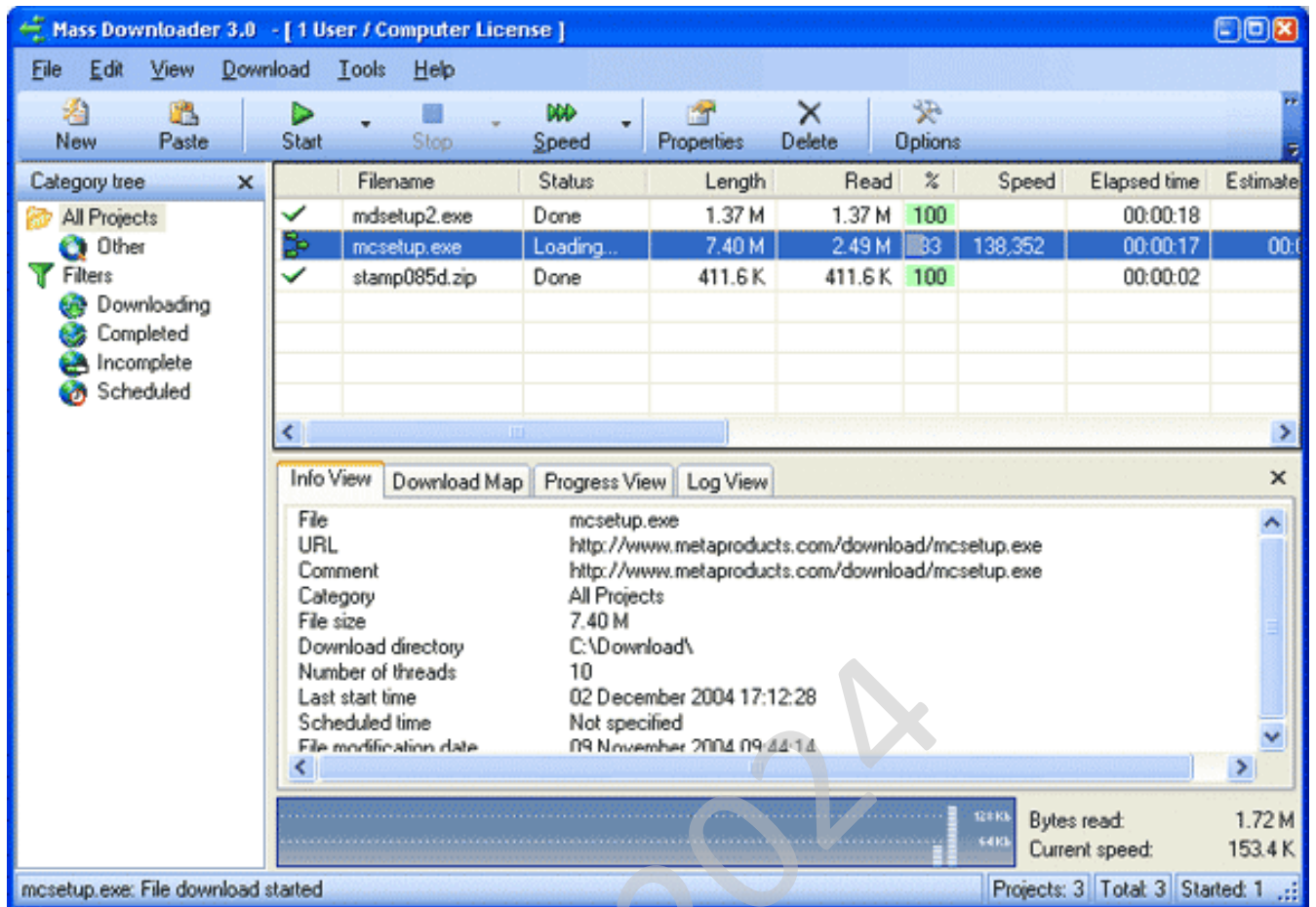


Рисунок 2.10 – Інтерфейс користувача Mass Downloader

### Net Vampire

Розмір дистрибутива 750 кілобайт, демонстраційна версія (30 днів).

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP.
- Робота через HTTP, FTP, SOCKS проксі-сервера.
- Вбудований FTP браузер.
- Закачування всіх файлів зі сторінки.
- Підтримка командного рядка.
- Планувальник закачувань.
- Категорії файлів (можливість зберегти файли різних типів у різних папках).

- Автоматична перевірка на віруси закачаних файлів.
- Автоматичний запуск закачаних файлів.
- Часткове закачування ZIP-архівів, перегляд їхнього вмісту перед закачуванням.
- Інтеграція в браузері.

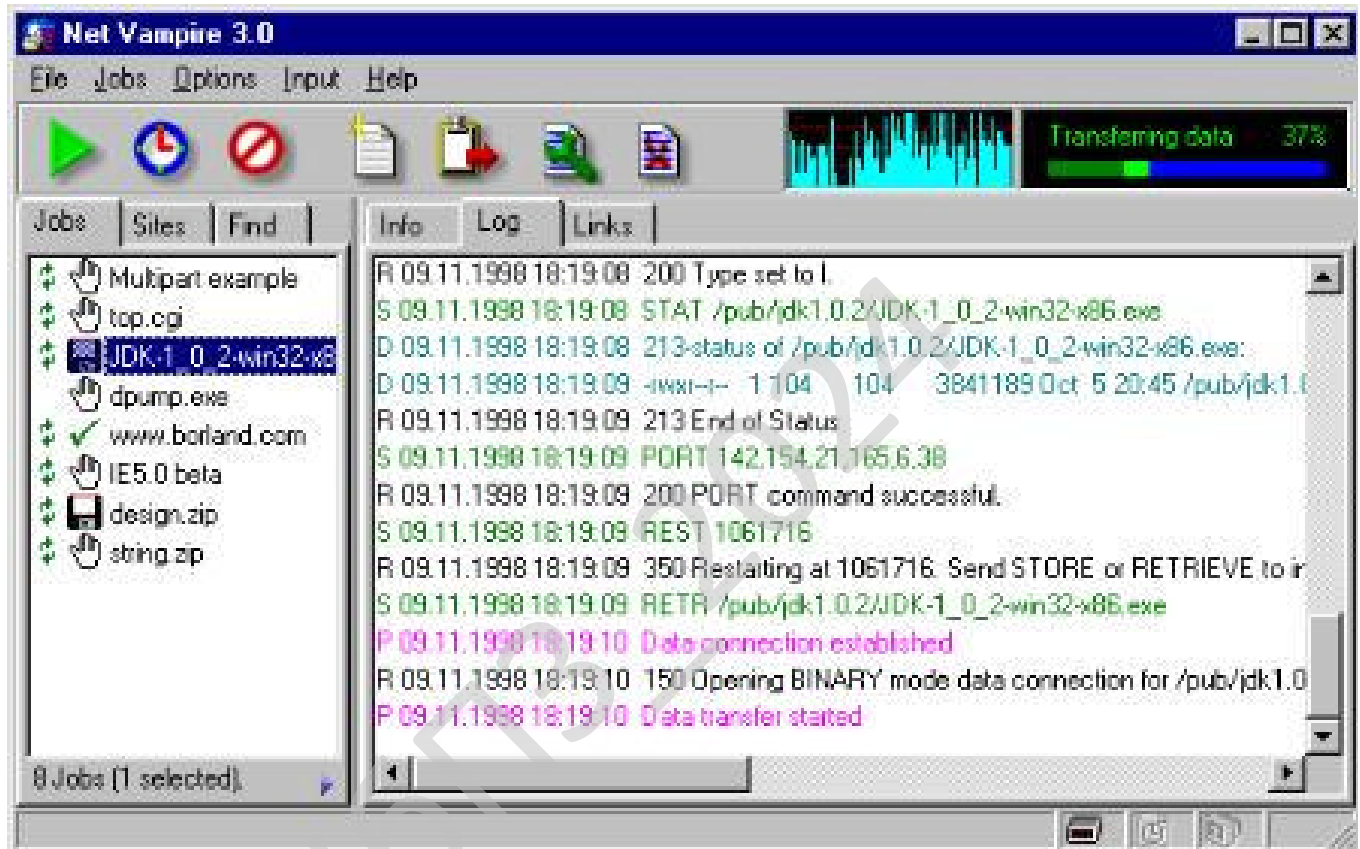


Рисунок 2.11 – Інтерфейс користувача Net Vampire

### ReGet Deluxe

Розмір дистрибутива 1,9 мегабайт, демонстраційна версія (30 днів).

Основні можливості:

- Протоколи: FTP, HTTP, HTTPS, FTPS, MMS, RTSP.
- Робота через проксі, авторизація: базова, NTLM.
- Автоматичний пошук дзеркал (інших серверів, звідки можна скачати файл).

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

– Менеджер сайтів (зберігання аккаунтів на серверах з наступною автоматичною авторизацією при додаванні заочування, персональні налаштування для заочувань із певних серверів).

– Створення списку файлів для завантаження про масці (file0.zip, file1.zip, file.zip).

– Зміна швидкості для одночасного з заочуванням перегляду сторінок, автоматичне визначення активності ІЕ і зниження швидкості заочувань.

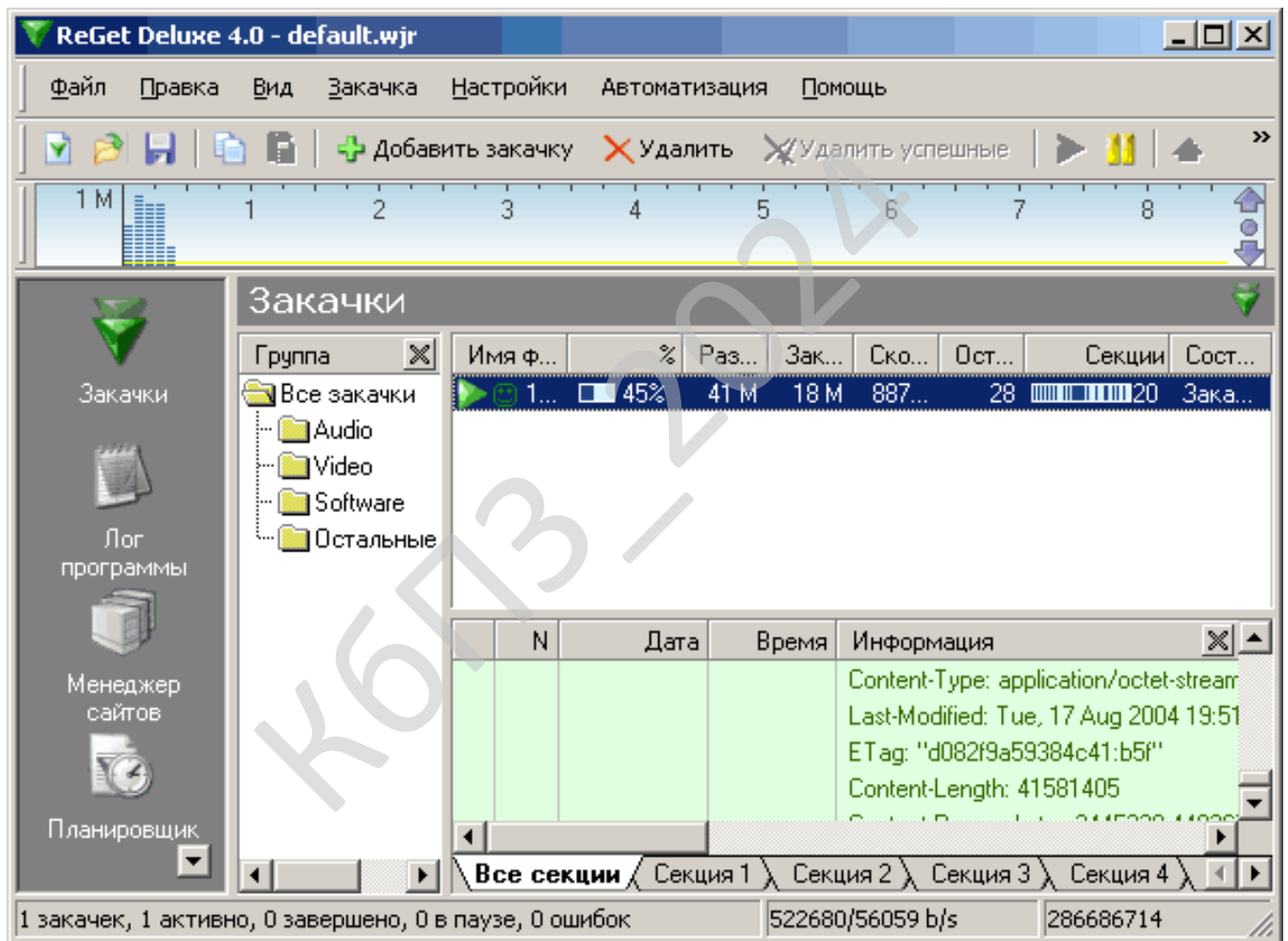


Рисунок 2.12 – Інтерфейс користувача ReGet Deluxe

– Категорії файлів (можливість збереження файлів різних типів у різних папках).

– Можливість використання скриптів для автоматизації.



- Вбудований FTP браузер.
- Інтеграція в браузери.

## 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 належить й розроблюється Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

### Delphi 10.4 Sydney

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

#### Основні можливості Delphi 10.4.1:

- Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.
- Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовуючи класичну реалізацію керування пам'яттю об'єктів.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

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

#### RAD Studio 10.4 Короткий огляд:

- Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

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

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

### **Істотне поліпшення Delphi Code Insight**

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

### **Delphi Custom Managed Records**

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомоги вашого коду, який буде виконуватися у відповідний момент.

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

### **Єдине керування пам'яттю**

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

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

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

### **Розширена підтримка бібліотек C++**

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

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

### **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відлагодочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відлагодочна інформація має інший внутрішній формат, сприяючи більш

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

### **Підвищення якості й швидкодії інструментів**

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

### **Змінені стилі VCL для High DPI**

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

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

### **Нові High DPI стилі й стилізація окремих VCL компонент**

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

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

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

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 ІМЕ.
- 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.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи download менеджера.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Система, яка розробляється, у ході виконання бакалаврського проектування, включає в себе наступні функціональні можливості.

Простий і зручний у використанні:

- інтуїтивно зрозумілий інтерфейс із можливістю налаштування;
- підтримка списків завантажень із широкими можливостями сортування;
- плаваючий індикатор швидкості завантаження, індикатори активних завантажень, керування активними завантаженнями;

- оригінальна технологія Активний Наочний Кошик, що дозволяє вам повноцінно працювати із програмою без відкриття головного вікна зі списком завантажень;

- підтримка модулів, що підключаються, (плагінів);
- підтримка шкір;
- Fast Update – перевірка доступності нових версій програми;
- багатомовність.

Повний набір необхідних функцій:

- динамічне багатопотокове завантаження;
- докачування після обриву зв'язку з HTTP, HTTPS і FTP серверів;
- робота з розкладу, відключення ПК після завершення завантаження;
- 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 завантажень;
- пошук і додавання дзеркал для завантаження;

- 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 браузером;
- 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;
- 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 тім же стані;

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– 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 завантаження:

– інтелектуальна багатосекційність. Він динамічно розбиває файл на секції й завантажує їх одночасно. Це збільшує швидкість завантаження в 3-10 разів;

– при зниженні швидкості автоматично перезапускає завантаження, що дозволяє уникнути простоїв.

Широкі можливості інтеграції:

– моніторинг буфера обміну;

– інтеграція в браузері Microsoft Internet Explorer 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 Internet Explorer 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 дозволяють указати час, коли необхідно скачати файли, що буває зручно при використанні діал-ап доступу в інтернет: список файлів для завантаження можна сформувати 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

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

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.

### 3.2 Розробка структурної схеми

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.
2. 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).
    - Дії (стартувати все, призупинити все, тимчасова зупинка завантажуваних, видалити все, знайти, знайти далі, швидкість).
    - Вид (настроювання кнопок, сортування списку, список завантажень, звук, категорії, лог завантажень, плаваюче вікно, скіни, мова інтерфейсу: українська, російська, англійська).
    - 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).

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

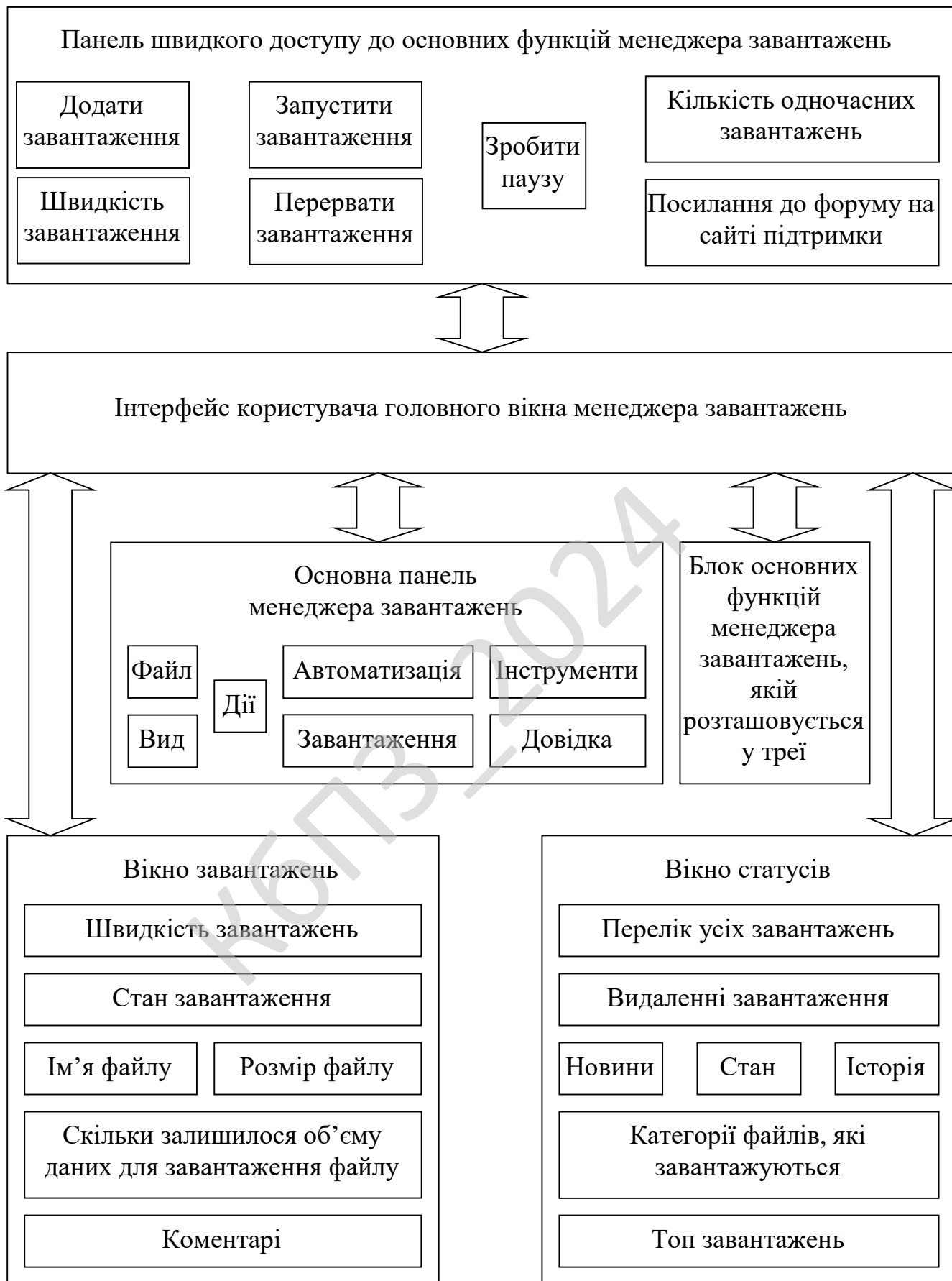


Рисунок 3.1 – Структурна схема системи

– Інструменти (пошук, історія, менеджер сайтів, розклад, плагіни, 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).

### 3. Вікно завантажень:

– Ім'я файлу.

– Стан завантаження.

– Розмір файлу.

– Скільки залишилося об'єму даних для завантаження файлу.

– Швидкість завантажень.

– Коментарі.

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

5. 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.

### 6. 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,

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

пауза, завантажено).

- Історія.
- Видаленні завантаження.

### 3.3 Розробка функціональної схеми

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

Функціонально система складається з наступних блоків:

- функціональний блок керування категоріями завантажень;
- функціональний блок історії завантажень;
- функціональний блок перевірки завантажених файлів на відновлення;
- функціональний блок динамічного багатопотокового завантаження;
- функціональний блок робота з розкладом;
- функціональний блок дозавантаження після обриву зв'язку з HTTP, HTTPS і FTP серверів;
- функціональний блок відключення ПК після завершення завантаження;
- функціональний блок оптимальних налаштувань для роботи з різними типами з'єднань (dial-up, ISDN, ADSL, LAN) і на різних швидкостях;
- функціональний блок тимчасової зупинки завантажуваних файлів з наступним їхнім стартом у тому же стані;
- функціональний блок пошуку і додавання дзеркал для завантаження;
- функціональний блок завантаження відео з відеосервісів;
- функціональний 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;
- функціональний блок роботи з RAR архівами: можливість перевіряти RAR архіви, можливість розпаковувати архіви;

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

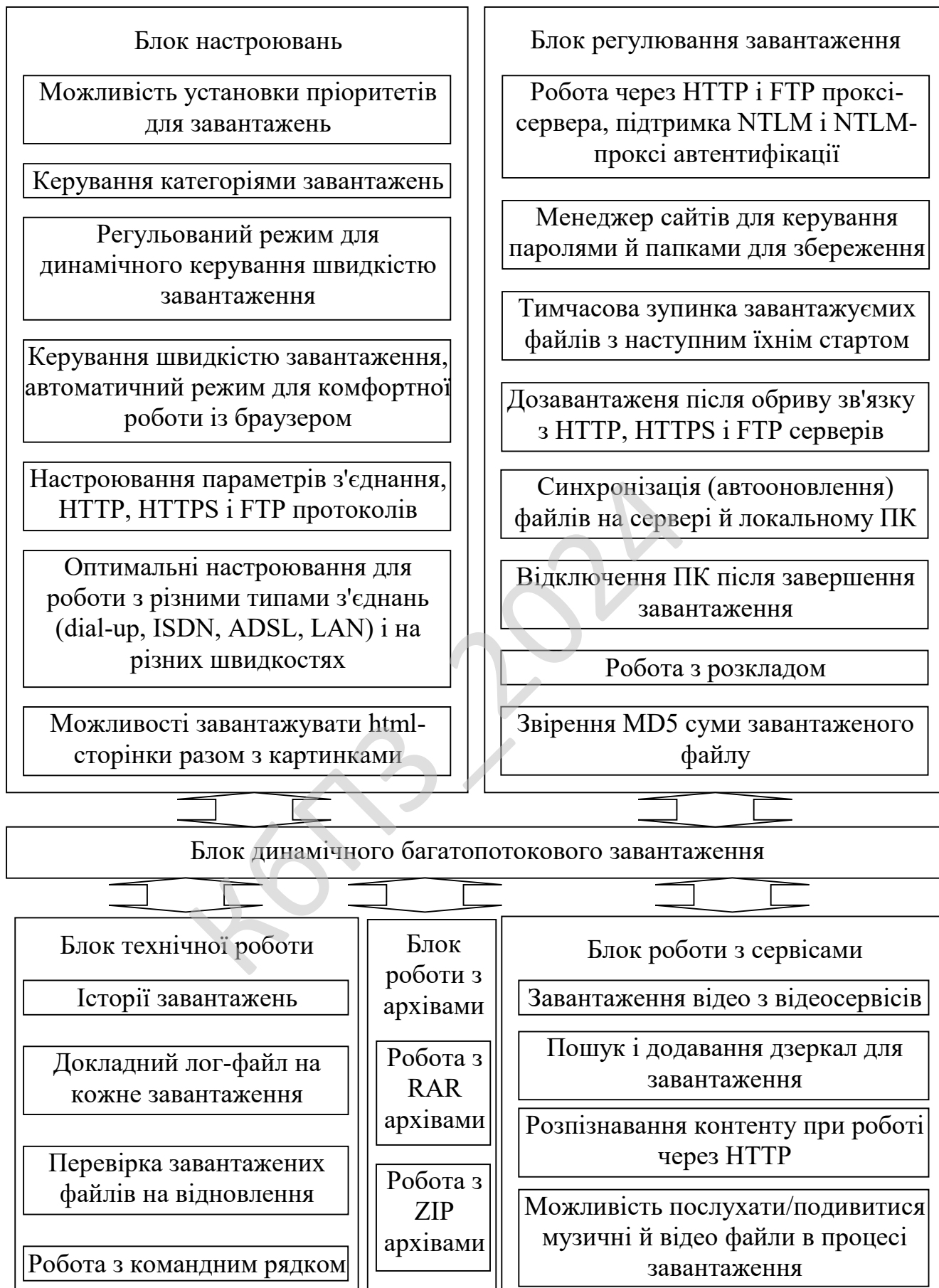


Рисунок 3.2 – Функціональна схема системи

- докладний лог-файл на кожне завантаження;
  - функціональний блок менеджера сайтів для керування пароллями й папками для збереження;
  - функціональний блок керування швидкістю завантаження, автоматичний режим для комфортної роботи із браузером;
  - функціональний блок можливість установки пріоритетів для завантажень;
  - регульований режим для динамічного керування швидкістю завантаження;
  - функціональний блок розпізнавання контенту при роботі через HTTP;
  - функціональний блок роботи з командним рядком;
  - функціональний блок синхронізації (автооновлення) файлів на сервері й локальному ПК;
  - функціональний блок можливості послухати/подивитися музичні й відео файли в процесі завантаження. Автоматичне одержання інформації про MP3 файлах при старті завантаження;
  - функціональний блок можливості завантажувати html-сторінки разом з картинками;
  - функціональний блок налаштування параметрів з'єднання, HTTP, HTTPS і FTP протоколів;
  - функціональний блок роботи через HTTP і FTP проксі-сервера, підтримка NTLM і NTLM-проксі автентифікації;
  - функціональний блок звірення MD5 суми завантаженого файлу.
- Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

### 3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3. З нього видно, що процеси, які відбуваються у системі взаємодіють наступним чином.

Першим процесом, який завантажується у системі є процес виведення головного вікна. Він взаємодіє з наступними процесами:

- Процес зміни параметрів.
- Процес управління категоріями.
- Процес управління завантаженнями.

Процес зміни параметрів взаємодіє з наступними процесами:

- Процес зміни параметрів завантажень.
- Процес зміни параметрів проксі-серверу.
- Процес зміни параметрів програми.

Процес управління категоріями взаємодіє з наступними процесами:

- Процес створення нової категорії.
- Процес вибору категорії зі списку.

Останній процес взаємодіє з наступними процесами:

- Процес редагування категорії.
- Процес видалення категорії.

Процес перегляду вмісту категорії, який у свою чергу, взаємодіє з процесом відкриття завантажених файлів.

Процес управління завантаженнями взаємодіє з наступними процесами:

- Процес додавання файлу у список завантажень.
- Процес вибору файлів зі списку завантажень.

Останній процес, у свою чергу взаємодіє з наступними процесами:

- Процес запуску завантаження.
- Процес паузи в завантаженні.
- Процес визначення параметрів в завантаженні.

- Процес виведення вікна процесу завантаження.
- Процес видалення файлу зі списку завантажень.

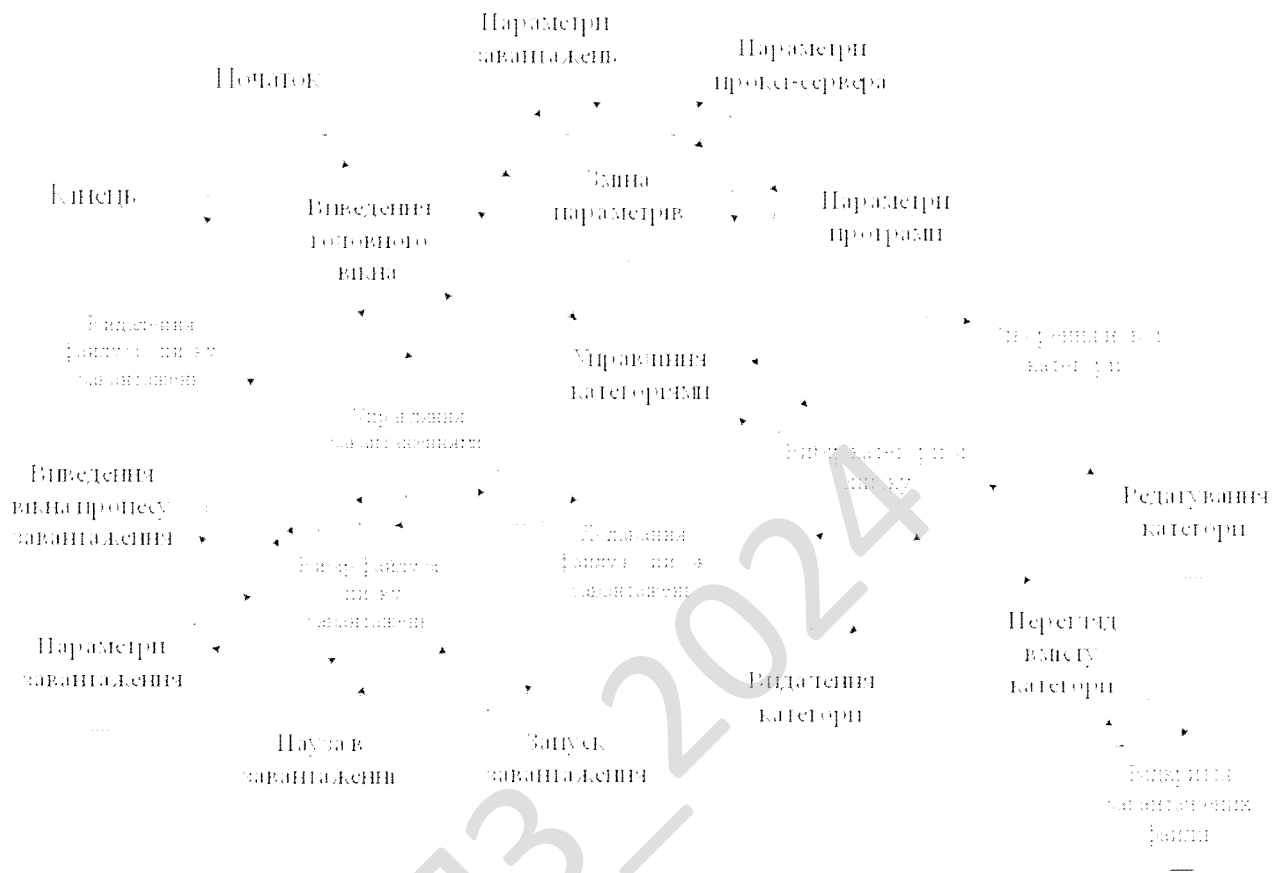


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

### 4.1 Блок–схеми та опис алгоритмів функціонування системи

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

Спершу відбувається виведення основного вікна програми. Після цього користувач обирає яку дію йому виконувати:

- Редагувати категорію.
- Змінити параметри.
- Завантажити файл.

Якщо він обирає редагування категорії, тоді відбувається редагування, створення та видалення категорій.

Якщо він обирає зміну параметрів, тоді відбувається зміна параметрів програми, проксі-сервера та завантажень.

Якщо він обирає завантаження файлу, тоді виконуються наступні дії:

- Виводиться посилання на файл.
- Вводиться папка для збереження.
- Обираються категорії.
- Обираються параметри завантаження.
- Запускається процедура завантаження файлу з мережі.
- Виводиться вікно процесу завантаження.

Якщо файл завантажено, тоді відбувається збереження файлу.

Якщо файл не завантажено, тоді відбувається виведення повідомлення про помилку.

Після цього користувач обирає, працювати йому далі з програмою, або ні.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

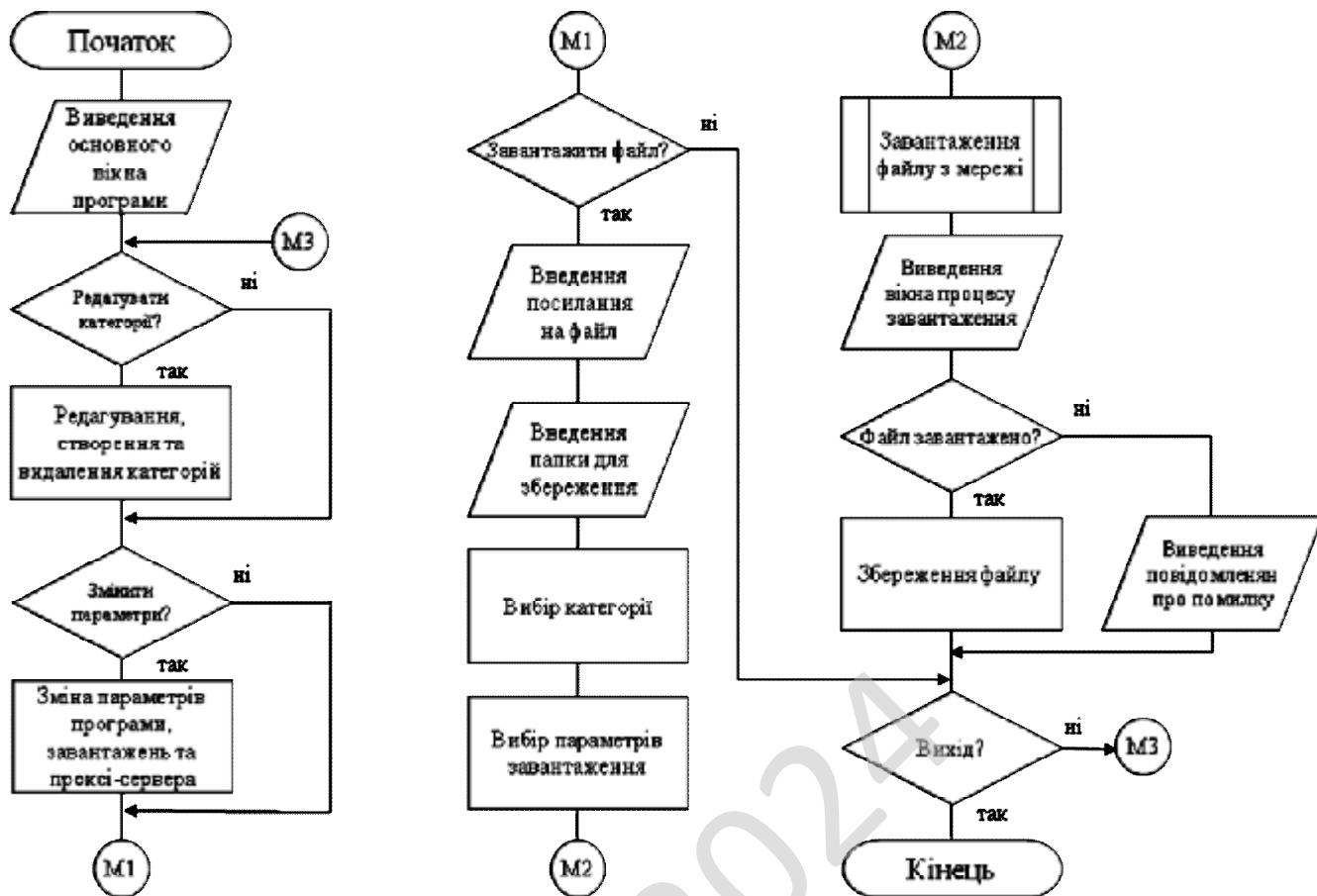


Рисунок 4.1 – Блок-схема роботи основної програми

На рисунку 4.2 зображено блок-схему роботи підпрограми завантаження файлу. Вона працює наступним чином.

Якщо файлу немає на сервері, тоді відбувається виведення повідомлення про недійсність введеного посилання.

Якщо файл є на сервері, тоді відбувається наступні дії:

- Внесення у категорію «Завантажується».
- Одержується розмір файлу.
- Виводиться розмір файлу.
- Одержується швидкість даного з'єднання.
- Виводиться швидкість з'єднання.
- Обчислюється приблизний час завантаження.
- Виводиться приблизний час завантаження.

– Відбувається завантаження файлу.

– Виводиться затрачений час на завантаження, та час, який залишився до кінця завантаження.

Якщо завантаження не завершилось, тоді відбувається перехід на одержання швидкості даного з'єднання.

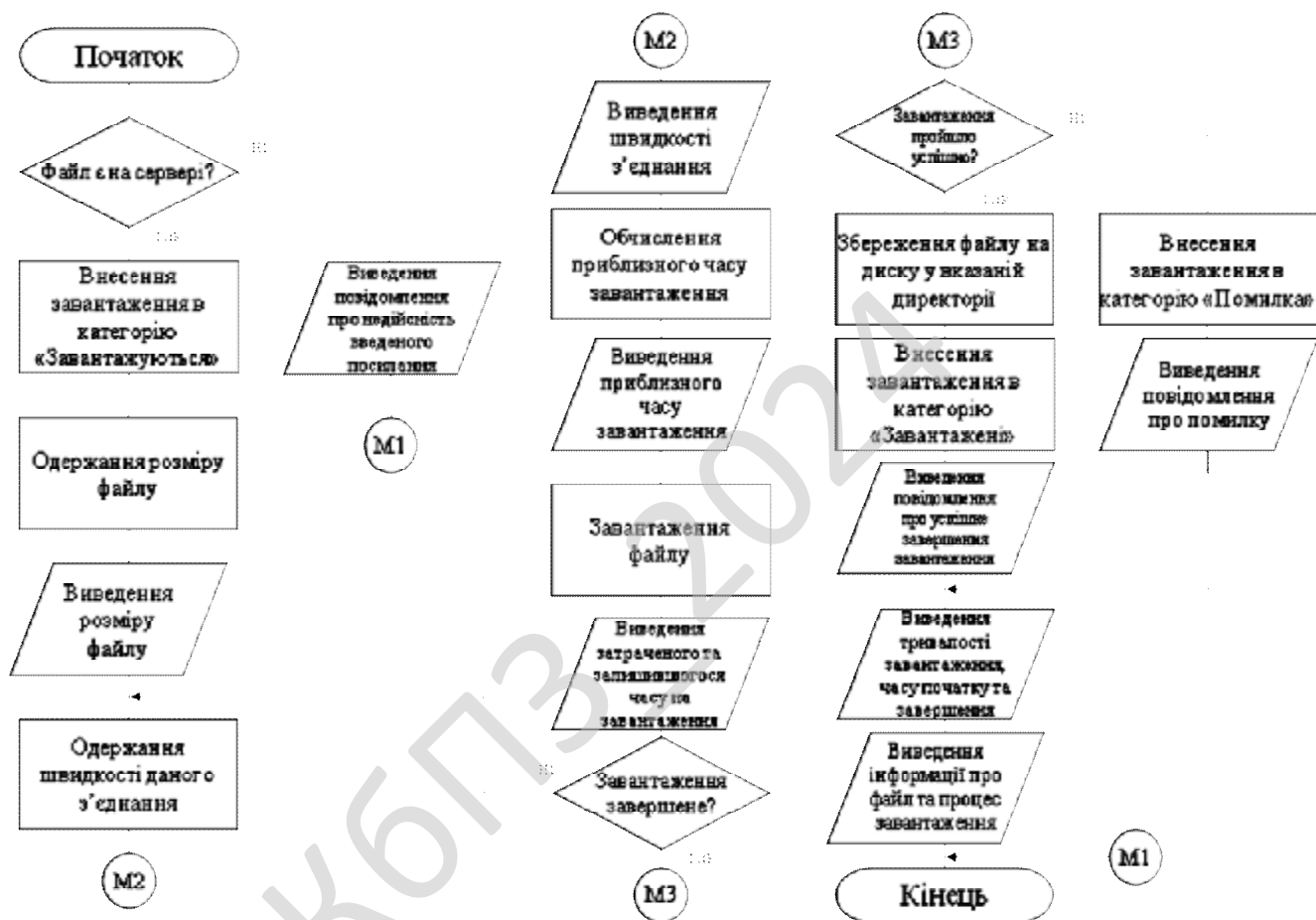


Рисунок 4.2 – Блок-схема роботи підпрограми завантаження файлу

Якщо ж завантаження закінчилось, тоді відбувається перевірка, чи успішно закінчилось завантаження.

Якщо воно закінчилось успішно, тоді виконуються наступні ітерації:

– Файл зберігається на диску у вказаній директорії.

– Вносяться завантаження у категорію «Завантажені».

– Виводиться повідомлення про успішне завершення завантаження.

Якщо ж воно закінчилось не успішно, тоді виконуються наступні ітерації:

– Завантаження вноситься у категорію «Помилки».

– Виводяться повідомлення про помилку.

Після цього відбуваються наступні дії:

– Виводяться відомості про тривалість завантаження, час початку та завершення.

– Виводиться інформація про файл та процес завантаження.

На цьому підпрограма закінчує свою роботу.

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 (у такому виді він зараз у САЗ):

```
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;
```

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45



```

Sock:=socket (AF_INET,SOCK_STREAM,IP_PROTO_TCP);
sockAddr. sin_family := AF_INET;
sockAddr. sin_port := htons (80);
sockAddr. sin_addr. S_addr := Resolve ('ismily.ua');
connect (Sock, sockAddr, sizeof (TSock ddr));
if (Sock <> INVALID_SOCKET) then
begin
    StrPCopy (Send_buf, 'GET /111. jpg HTTP/1.1'#13#10 + 'Host:
ismily.ua'#13#10+' User-Agent: Opera/9.63 (Windows NT 5.1; U;
ua) 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.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

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.

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 настроювання). 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 – Максималана 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.

- 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 наведене 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:

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

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

– Connected – показує, чи встановлено в цей момент з'єднання. Тип: Boolean.

– BeenCanceled – показує, чи було перерване з'єднання із сервером. Тип: Boolean.

– Host – рядок, що містить хост-ім'я віддаленого комп'ютера. Заповнювати не треба, тому що ця властивість устанавлюється автоматично при виклику методів Get, Put, Postи т.д. Тип: string. Port-Integer, що містить порт віддаленого комп'ютера (заповнюється теж автоматично).

– Timeout – таймаут у мілісекундах. Тип: Integer.

Методи:

– Get (URL: string) – надсилає запит на зазначений URL. Дані після виконання цього запиту записуються у файли або в самі властивості Body і Header (залежно від значення властивості InputFileMode).

– Head (URL: string) – надсилає запит на зазначений URL. Дані після виконання цього запиту записуються у файл або в саме властивість Header (залежно від значення властивості InputFileMode). На відміну від методу Get, при виклику Head запит відсилається тільки на заголовок http-документа.

– Post (URL, postData: string) – надсилає запит на зміну http-документа (з адресою URL) на дані, що втримуються в параметрі postData. Якщо OutputFileMode дорівнює True, то в postData повинен утримуватися шлях до файлу, що містить потрібні дані. Put (URL, postData: string) – надсилає запит на створення http-документа (з адресою URL), що містить дані, передані в параметрі postData. Якщо OutputFileMode дорівнює True, то в postData повинен утримуватися шлях до файлу, що містить потрібні дані. Trace (URL, traceData: string) – надсилає запит на одержання відладочних даних (для налагодження з'єднання з HTTP-сервером). Дані для запиту потрібно вказати в параметрі traceData. Якщо OutputFileMode дорівнює True, то в traceData повинен

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.

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

– 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 27.01.2012 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;
```

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

```

end;

end;

Додавання від 27.01.2012 16:44:

Uses OleCtnrs, IdHTTP;
const Price = 'price.xls';
URL = 'http://kirovohrad.ua/upload/File/price.xls';
Port = 3128;
IP = '192.168.1.2';
Login = 'internet';
Password = 'www';
Var Http: TIdHTTP;
begin
    Http := TIdHTTP.Create(nil);
    InitHttp(Port, IP, Login, Password, Http);
    if LoadPrice(URL, Price, Http) then Convert;
    Http.Free;
    Readln;
end.

```

### 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. НЕ ЗАБУДЬТЕ ПОМІСТИТИ У ФОРМУ КОМПОНЕНТИВ TNMHTTP!}

```

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

```

### Однчасне 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;

```

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

```

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
        if not DirectoryExists (Edit1. Text) then
            Mkdir (Edit1. Text);

```

						<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			54

```

        {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, указуємо їй її URL і
                запускаємо її}
                URL := ListBox1. Items[i];
                Resume;
            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);
        {Указуємо, 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;

```

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

## 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192- або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири  $8 \times 8$ -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавіш. Шар дифузії використовує лінійне перетворення на основі матриці з номером галузей 5.

### Аналіз безпеки

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

NESSIE і Японським CRYPTREC проект. Японський шифр має рівні безпеки й можливості обробки, порівнянні із шифром AES/Rijndael.

Camellia – це блоковий шифр, який може бути повністю визначені мінімальними системами багатомірних багаточленів:

– Камелія (а також AES) S-блоки можуть бути описані системою 23 квадратних рівнянь в 80 членах.

– Розклад ключів можна описати 1120 рівняннями в 768 змінні з використанням 3328 лінійних і квадратичних членів.

– Увесь блоковий шифр можна описати 5104 рівняннями в 2816 змінні з використанням 14 592 лінійних і квадратичних членів.

– Усього потрібно 6224 рівняння з 3584 змінними з використанням 17 920 лінійних і квадратичних членів.

– Кількість вільних членів становить 11 696, що приблизно таке ж число, що й для AES.

Теоретично, такі властивості можуть дозволити зламати Camellia (і AES) за допомогою алгебраїчної атаки, такий як розширена розріджена лінеаризація, у т Майбутнє за умови, що атака стане можливою.

Хоча Camellia запатентована, вона доступна за безоплатною ліцензією. Це дозволило шифру Camellia стати частиною проекту OpenSSL під ліцензією з відкритим вихідним кодом з листопада 2006 року. Це також дозволило йому стати частиною Mozilla Модуль NSS (Служби мережної безпеки).

Підтримка Camellia була додана в остаточний випуск Mozilla Firefox 3 в 2008 році (за замовчуванням відключене починаючи з Firefox 33 в 2014 році в дусі «Пропозиції по зміні стандартних наборів шифрів TLS, пропонованих браузерами», який був виключено з версії 37 в 2015 році). Pale Moon, відгалуження Mozilla / Firefox, продовжує пропонувати Camellia і розширив свою підтримку, включивши в нього набори Galois / Counter mode (GCM) із шифром, але вилучив GCM знову у випуску 27.2.0, пославшись на очевидну відсутність інтересу до них.

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Нemoжлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

### **Продуктивність**

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс користувача програми. З нього видно, що інтерфейс розбито на декілька блоків:

- Блок меню.
- Блок клавiш швидкого доступу.
- Блок відображення категорiй.
- Вікно завантаження.

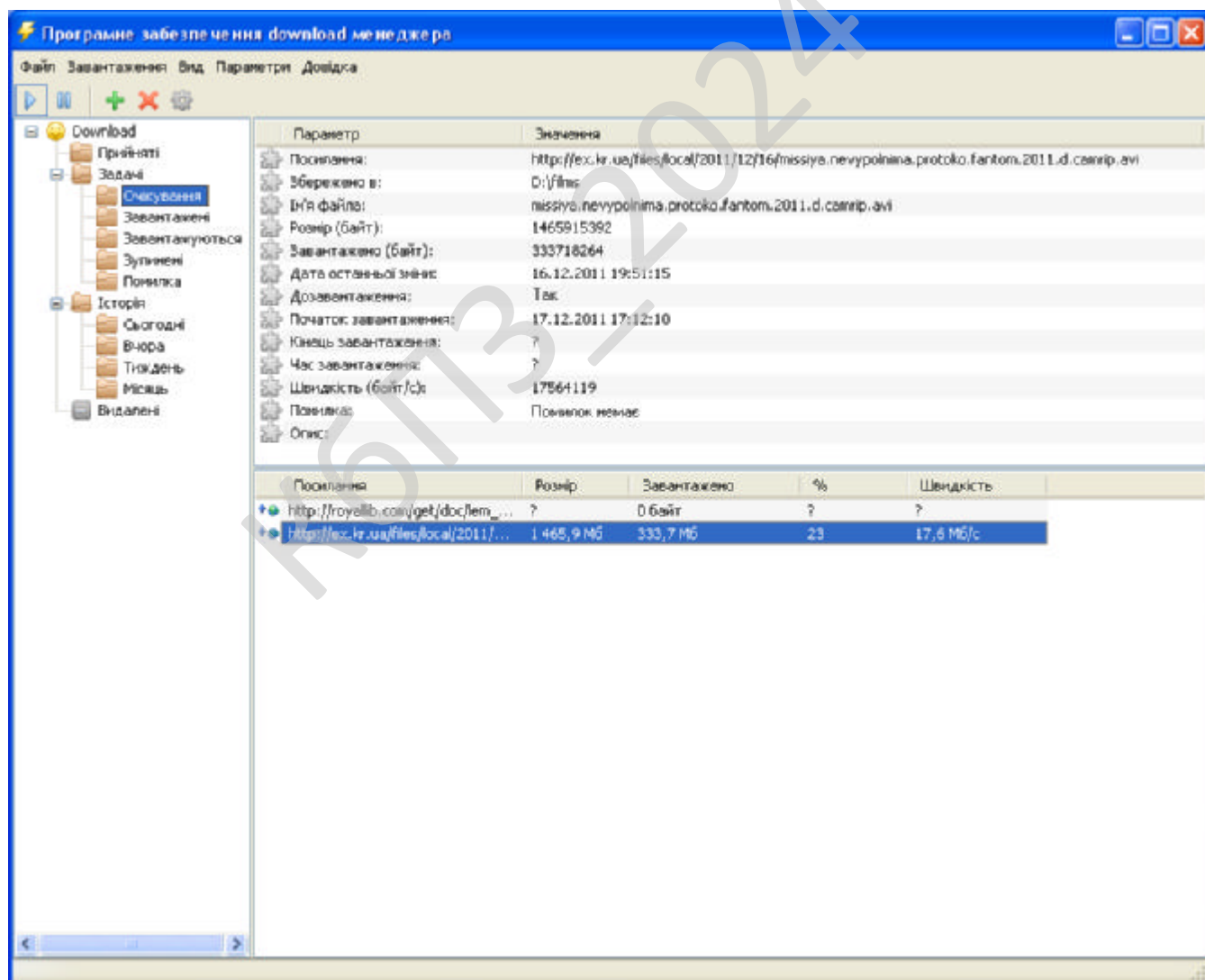


Рисунок 5.1 – Головне вікно програми

Блок меню включає в себе наступні елементи:

- Файл.
- Завантаження.
- Вид.
- Параметри.
- Довідка.

Блок відображення категорій включає в себе наступні категорії:

- Прийняті.
- Задачі.
- Історія.
- Видалені.

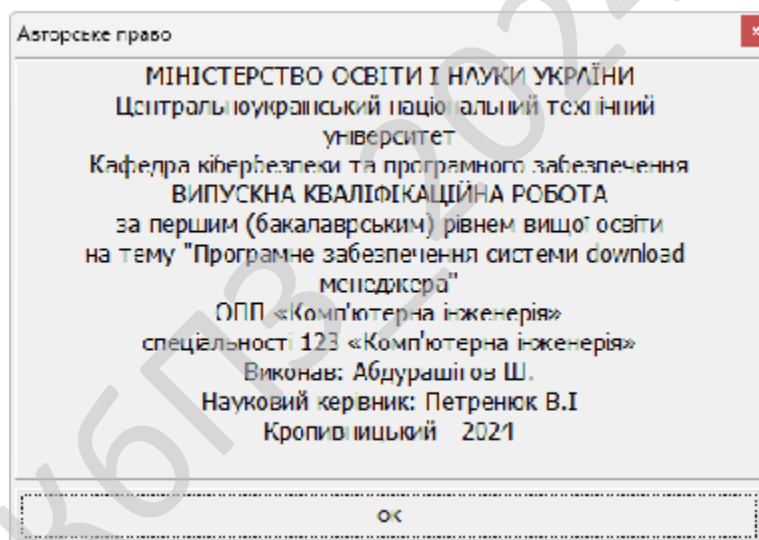


Рисунок 5.2 – Вікно довідки

Задачі включають у себе наступні елементи:

- Очікування.
- Завантаження.
- Завантажуються.
- Зупинені.
- Помилка.

Історія:

- Сьогодні.
- Вчора.
- Тиждень.
- Місяць.

На рисунку 5.2 зображено вікно довідки, у якому визначений розробник програмного забезпечення, тема бакалаврського проекту, керівник та місце виконання.

КБПЗ\_2024

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи download менеджера.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем download менеджера.
- Досліджена система download менеджера.
- На основі отриманих результатів досліджень створена програмна реалізація системи download менеджера.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання download менеджера.

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

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

Програма реалізована на мові високого рівня Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи download менеджера. 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

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

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 Camellia.

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.

КБПЗ\_2024

					ВКРБ-123.24.0083.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
2. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
3. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
4. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
5. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
6. 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.
7. 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.
8. 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.

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

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

11. 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.

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

13. 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.

14. 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.

15. 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.

16. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties».

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

*International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

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

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.*

25. 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.

26. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

27. 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.

28. 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.

29. 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.

30. 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

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.*

31. 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.*

32. 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*

33. 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.*

34. 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.*

35. 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.

36. 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.*

37. 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.*

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

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

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

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

43. 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.

44. 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.

					<b>ВКРБ-123.24.0083.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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

46. 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.

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

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

49. 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.

50. 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.

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

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	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					<b>ВКРБ-123.24.0083.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Абдурашітов Ш.				Програмне забезпечення системи download менеджера	Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.					Б	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-20			
Затв.	Смірнов О.А.							

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи download менеджера.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 131-02 від 01.04.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи download менеджера.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) 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;

					ВКРБ-123.24.0083.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи download менеджера;
- 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.

					<b>ВКРБ-123.24.0083.00.00.ТЗ</b>	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, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРБ-123.24.0083.00.00.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 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 70 аркушів.

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					<b>ВКРБ-123.24.0083.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 10.06.2024 р.

					ВКРБ-123.24.0083.00.00.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  
першим (бакалаврським) рівнем вищої освіти

\_\_\_\_\_ Петренюк В.І.

*Програмне забезпечення системи download менеджера*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 53

Літера: РП

Кропивницький – 2024 року

**DM\_Main.pas - головна програма download менеджеры**

```
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;
  end;
end;
```

```

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)+' /c'
            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 голвної 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);
    fLoading.Data:=lvTasks.Selected.Data;
    fLoading.Show;
  end;
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;

```

```
//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;
  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:=tsStoped;
    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:=tsStoped;
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;

```

```

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)+' /c')
  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)+' /c'
    else lvTasks.Items[i].SubItems[4]='?';
  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;

//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.

```

## DM\_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 : PVSFixedFileInfo;

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;

```





```

function GetFreeSpace(Disk : String) : Int64;
var
  TotalBytes      : Int64;
  TotalFreeBytes  : PLargeInteger;
  FreeBytesCall   : Int64;

begin
  New(TotalFreeBytes);
  try
    GetDiskFreeSpaceEx(PChar(Disk), FreeBytesCall, TotalBytes, TotalFreeBytes);
    Result := TotalFreeBytes^;
  finally
    Dispose(TotalFreeBytes);

  end;
end;

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

//Запуск завантаження

function IsRun : Boolean;
var
  Mutex : integer;

begin
  Result := False;
  Mutex := CreateMutex(nil , True, 'BankClientServer');
  if GetLastError <> 0 then
  begin
    CloseHandle(Mutex);
    Result := True;
  end;
end;

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

//Рядок з часом

function GetTimeStr(SeCS : Integer) : String;

  function LeadingZero(N:Integer) : String;
  begin
    if N < 10 then Result := '0' + IntToStr(N) else Result := IntToStr(N);

  end;

var
  Hours, Mins : Integer;
begin
  Hours := SeCS div 3600;
  SeCS := SeCS - Hours * 3600;
  Mins := SeCS div 60;
  SeCS := SeCS - Mins * 60;

  Result := LeadingZero(Hours) + ':' + LeadingZero(Mins) + ':' +
  LeadingZero(SeCS);
end;
end.

```

## DM\_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.
```

K6П3\_2024

## DM>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;
    ProgressBar.MaxValue := Data.TotalSize;
    ProgressBar.Progress := Data.LoadSize;
  end;
end;

```

```
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.
```

## DM\_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'                               //
////////////////////////////////////

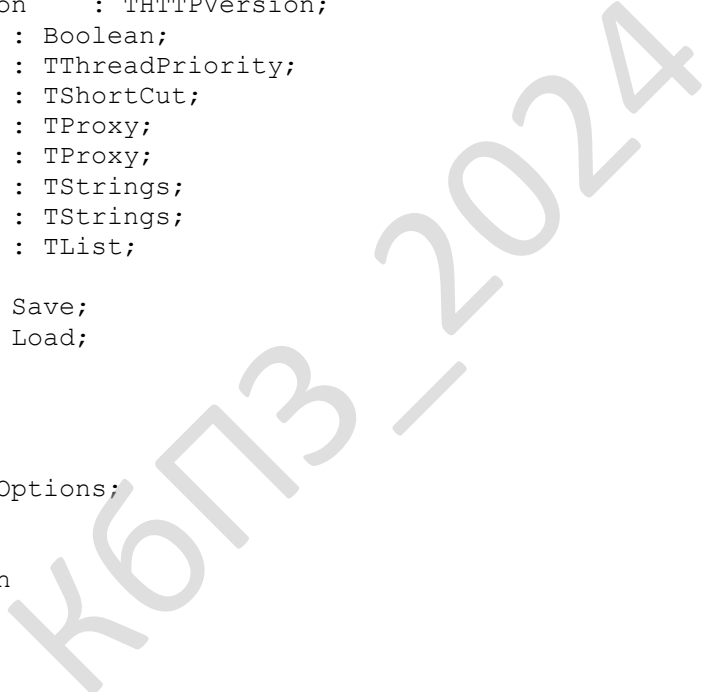
////////////////////////////////////

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.Panel1.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.

```

## DM\_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;
    Mem1: 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;

```

```
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;
Node:=fMain.tvFolders.Items.AddChild(fMain.tvFolders.Selected, edName.Text);
Node.ImageIndex:=2;
Node.SelectedIndex:=2;
Close;
end;
end.
```

КБПЗ\_2024

## DM\_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;
    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);
    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;
  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;
fMain.tvFolders.Selected.Text := Trim(edName.Text);
Close;
end;

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

end.
```

КБПЗ\_2024

## DM\_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.
```

K6П3\_2024

## DM\_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\_2024

**DM\_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;
    PageControl1: 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.
```

КБПЗ\_2024

**DM\_Threads.pas - теми**

```

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 := 'Помилка 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;

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

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.

```

## DM\_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('БАКАЛАВРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Програмне забезпечення download менеджера');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Петренюк В.І. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Абдурашітов Шахрух ');
  Memo1.Lines.Add(' гр. KI-20');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Кропивницький 2024');
  Memo1.Lines.Add('');
end;
end;
end.

```