

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи
інтелектуального завантаження даних з мережі”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-1
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Гладин В.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Гладину Віталію Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі

2. Керівник роботи Смірнова Тетяна Віталіївна, канд. техн. наук

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

затверджені наказом вищого навчального закладу № 32-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

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

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

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

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Гладин В.В. Дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального завантаження даних з мережі.

Метою розробки є дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі.

Об'єктом дослідження є процес інтелектуального завантаження даних з мережі.

Предметом дослідження є методи інтелектуального завантаження даних з мережі.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуального завантаження даних з мережі.

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

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

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

Програму розроблено в середовищі RAD Studio Delphi 10.

Ключові слова: комп'ютерні науки, інтелектуальне завантаження, дані, мережі

ABSTRACT

Hladyn V.V. Research and software implementation of the system of intelligent download of data from the network. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of intelligent data download from the network.

The purpose of the development is research and software implementation of a system of intelligent data download from the network.

The object of research is the process of intelligently downloading data from the network.

The subject of research is the methods of intelligent data download from the network.

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

The result of the work is the software implementation of the intelligent data download system from the network.

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

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

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

The program was developed in the RAD Studio Delphi 10 environment.

Keywords: computer science, intelligent download, data, networks

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	32
3.4 Розробка діаграми процесів.....	35
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	37
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	37
4.2 Захист розробленого програмного забезпечення.....	53
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	56
6 НАУКОВА НОВИЗНА	62

						ВКРМ-122.23.0028.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Гладиш В.В.				Дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі	Літ.	Аркуш	Аркушів
Перев.	Смірнова Т.В.					М	1	101
Н.контр.	Коваленко А.С.				ЦНТУ КН-22М-1			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	63
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	63
7.2 Розрахунок трудомісткості розробки програмної продукції.....	65
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	67
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	71
7.5 Визначення собівартості розробки та ціни програмної продукції.....	75
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	79
7.7 Визначення експлуатаційних витрат.....	79
7.8 Визначення економічної ефективності програмної продукції.....	81
7.9 Висновок.....	83
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	84
8.1 Вступ.....	84
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	85
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	86
8.4 Розробка заходів з умов поліпшення охорони праці.....	89
8.5 Розрахункова частина	90
9 ОСНОВНІ ВИСНОВКИ.....	93
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Сучасне суспільство неможливо представити без Інтернету. Всі ми що-небудь качаємо з Інтернету. Однак використовувати для цього вбудовані засоби браузера хоч і просто, але не завжди надійно – навіть якщо ви є власником виділеної двомегабітної лінії, однаково ймовірність одержання битих файлів залишається. Що ж говорити про тих, хто підключається до Інтернету по dial-up, – їм необхідно застосувати які-небудь, нехай самі мінімальні, заходи щодо захисту своїх завантажень.

І якщо на якісний канал грошей немає, то єдиний спосіб убезпечити файли, що завантажуються, від прикрих збоїв і обривів лінії – скористатися яким-небудь менеджером завантажень.

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.

Подібними можливостями сьогодні вже не здивувати інтернет-користувача, адже завантажувати що-небудь за допомогою стандартного браузера нікому просто не спадає на думку. Тому перед розроблювачами менеджерів завантаження коштує завдання постачити свою програму як можна більшим числом корисних опцій, а перед користувачем – вибрати кращу.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі.

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

- Огляд існуючих систем інтелектуального завантаження даних з мережі.
- Дослідження системи інтелектуального завантаження даних з мережі.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Програмна реалізація системи інтелектуального завантаження даних з мережі.

Об'єктом дослідження є процес інтелектуального завантаження даних з мережі.

Предметом дослідження є методи інтелектуального завантаження даних з мережі.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод інтелектуального завантаження даних з мережі.
- Розроблено вітчизняний продукт інтелектуального завантаження даних з мережі, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуального завантаження даних з мережі.

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

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

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для завантаження файлів з інтернету, а у принципі, й з довільної мережі.

Стандартний менеджер завантажувань має наступний набір функцій:

- 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, виділюваної сервером на одне з'єднання.
- Обмеження швидкості завантаження.
- Рекурсивне завантаження (створення локального дзеркала).
- Зручна організація завантажень, зберігання посилань до завантажених файлів.
- Створення списків завантажень у різних форматах.
- Завантаження файлів за розкладом.
- Перевірка завантажених файлів антивірусними програмами.

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1.2 Область застосування

Областю застосування менеджера завантажень є довільні задачі, пов'язані з завантаженням файлів з мережі, як локальної або корпоративної, так і глобальної, тобто з Інтернету.

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.

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 браузером 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.

Однак область застосування менеджерів завантажень має деякі обмеження:

– 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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

відмінності у швидкості їхньої роботи (найчастіше незначні) пояснюються якістю вашої телефонної лінії й каналу провайдеру, ступенем завантаженості Web-вузлів, з яких ви завантажуйте 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%.

– 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 установлене 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.

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

У рамках виконання магістерського проектування проведемо дослідження існуючих менеджерів завантажень. Не секрет що користувачеві, який буває в інтернеті часто доводиться зберігати на жорсткий диск свого комп'ютера ті або інші файли, виявлені на веб-просторах. Робити це можна вбудованими засобами windows, але набагато зручніше користуватися так званими менеджерами завантажень. Ці програми дозволяють крім простій завантаження контенту з'єднуватися з інтернетом за розкладом, поновлювати зв'язок при його обриві, дозавантажувати файли в будь-який зручний користувачеві час, вести зручне сортування завантажень по їхньому типу й багато чого іншого. Нижче ми приведемо невеликий огляд існуючих на даний момент платних і безкоштовних менеджерів завантажень і спробуємо з'ясувати, у чому в них відмінності.

Free Download Manager

Колись невелика й проста програма обростила купою функціональних доповнень – тут і робота з торрентами й завантаження flash-відео, з'явилися додаткові мови. Існує й lite версія меншого розміру, до якої можна підключити тільки необхідні цікаві користувачеві плагіни.

Основні можливості:

- Протоколи: HTTP, HTTPS, FTP.
- Убудовані FTP і WEB браузері.
- Планувальник завантажень.
- 3 режими для обмеження швидкості всіх завантажень.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні завантаження, персональні налаштування для завантажень із певних серверів).
- Завантаження сайтів цілком або тільки файлів з певними розширеннями.
- Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні завантаження, персональні налаштування для завантажень із певних серверів).
- Завантаження сайтів цілком або тільки файлів з певними розширеннями.

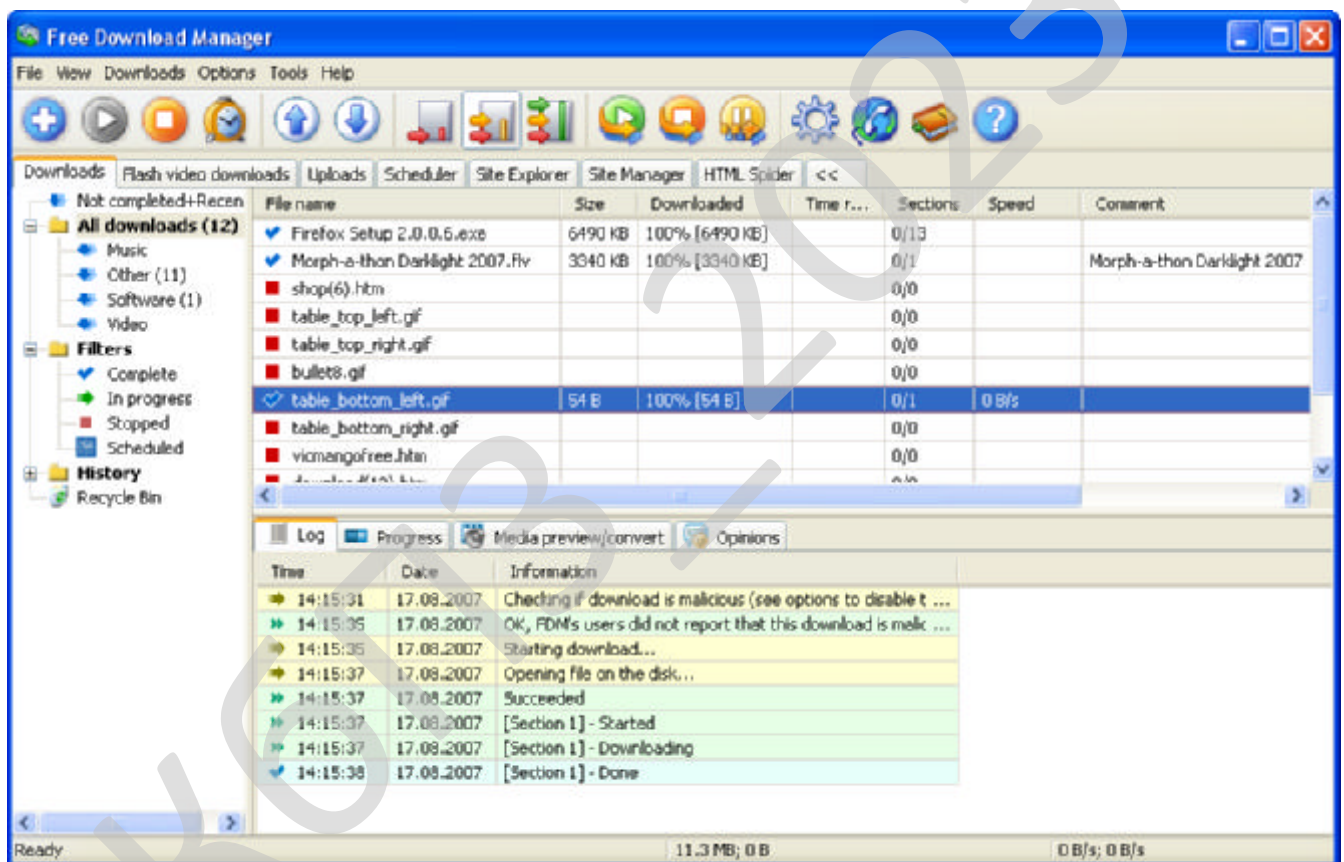


Рисунок 2.1 – Інтерфейс користувача Free Download Manager

ReGet Deluxe

ReGet Deluxe – це швидкий і надійний менеджер завантажень для Windows XP і Vista, що пропонує багато функціонала для збільшення

швидкості й організації ваших завантажень, поза залежністю від того, відбувається завантаження чи з веб-сайтів, FTP-серверів, локальної мережі або серверів, що пропонують потоковий мультимедіа-контент.

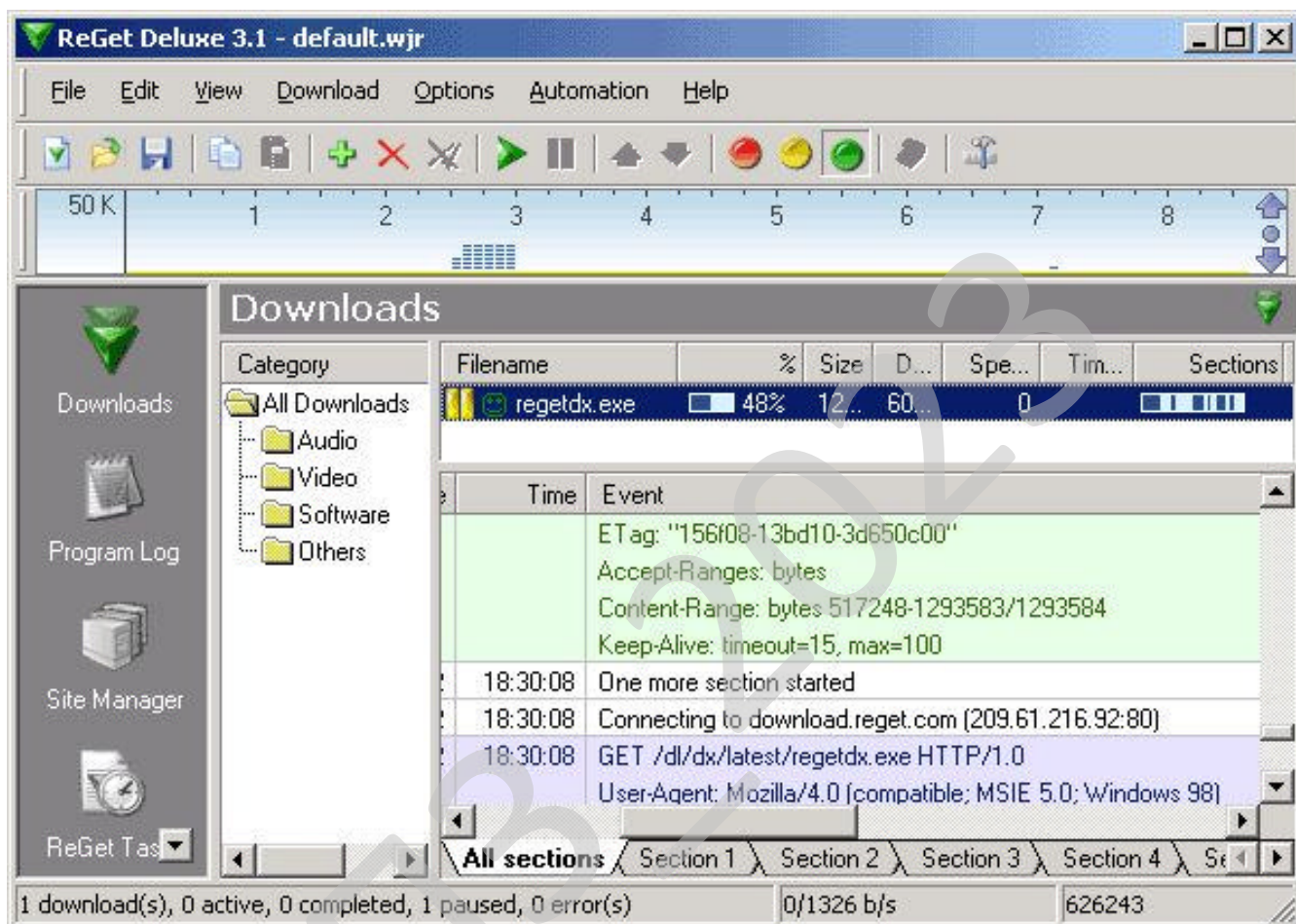


Рисунок 2.2 – Інтерфейс користувача ReGet Deluxe

Основні можливості:

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

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

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

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

– Можливість використання скриптів для автоматизації.

– Автоматична перевірка на віруси завантажених файлів.

– Планувальник завантажень.

– Вбудований FTP браузер.

– Часткове завантаження ZIP-архівів, перегляд їхнього вмісту перед завантаженням.

– Інтеграція в браузери.

Download Master

Основними рисами, що виділяють Download Master серед інших програм, є найвища ефективність роботи й зручний користувальницький інтерфейс.

Download Master дозволяє значно підвищити швидкість завантаження файлів через Інтернет з використанням HTTP, HTTPS і FTP протоколів. Для цього використовується розбивка файлу на потоки, які завантажуються одночасно. Програма також підтримує докачку файлу з поточної позиції після обриву зв'язку.

Для підвищення зручності використання, Download Master інтегрується в Internet Explorer, Firefox, Mozilla, Opera, Netscape Communicator і інші, замінюючи стандартний модуль завантаження. Крім того, програма стежить за буфером обміну, виявляючи посилання.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

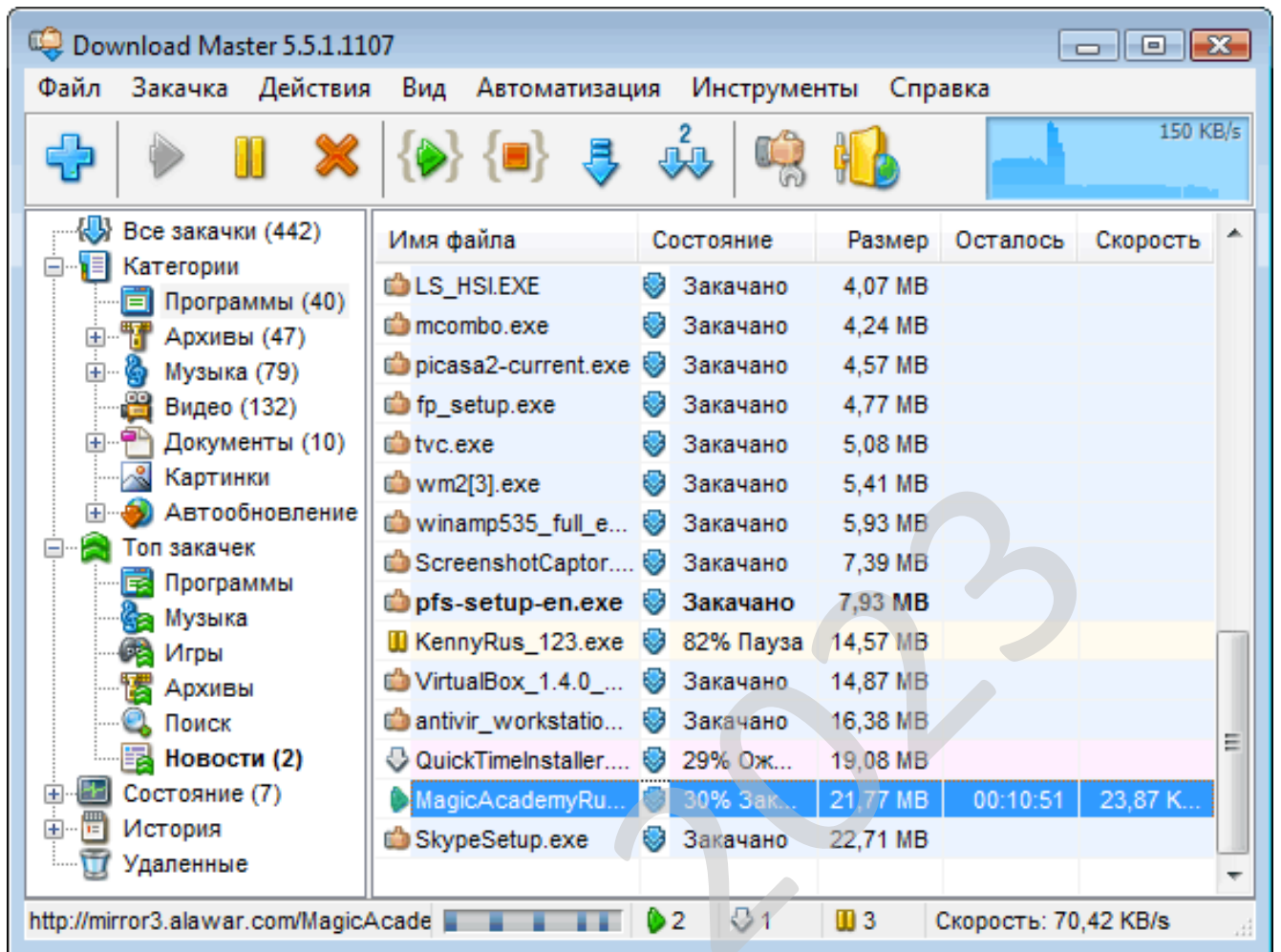


Рисунок 2.3 – Інтерфейс користувача Download Master

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).
- Часткове завантаження ZIP-архівів, перегляд їхнього вмісту перед завантаженням.
- Підтримка архівів RAR: можливість перевіряти й розпаковувати.

– Менеджер сайтів (зберігання аккаунтів з наступною автоматичною авторизацією при додаванні завантаження, персональні налаштування для завантажень із певних серверів).

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

– Можливість змінювати пріоритет завантаження.

– Підтримка командного рядка.

– Синхронізація файлів на комп'ютері із сервером.

– Перевірка завантажених файлів на відновлення

– Можливість прослухати/переглянути музичні й відео файли в процесі завантаження. Автоматичне одержання інформації про MP 3-файлах при старті завантаження.

– Автоматична перевірка на віруси завантажених файлів.

– Інтеграція в браузері.

HiDownload

За заявою розроблювача – програма HiDownload уміє працювати не тільки із завантаженням файлів з мережі, але й з потоковим відео й аудіо, одним кліком користувач може підписатися на підкасти – програма сама завантажить новий випуск, тільки він з'явиться на сервері.

Основні можливості:

– Протоколи: HTTP, HTTPS, FTP, RTSP, MMS.

– Робота через HTTP, FTP, SOCKS проксі-сервера.

– Можливість указати ім'я завантажуючого файлу, з яким він буде збережений на диску.

– Робота з потоковими форматами: Windows Media™.

– Real Audio™ and Real Video™.

– QuickTime™.

– Flash Video (FLV).

– Streaming MP3 (Shoutcast™).

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

- Підтримка метафайлів: ram, smil, asx, wmx, wvx, m3u, pls.
- Завантаження безлічі файлів із сайту або FTP-сервера.
- Підтримка інтернет радіомовлення.
- Планувальник завантажень.
- Категорії файлів (можливість зберегти файли різних типів у різних папках).
- Можливість вказати Referrer для кожного завантаження.
- Захват і збереження на диску потокового відео, аудіо.
- Вбудований FTP браузер.
- Автоматична перевірка на віруси завантажених файлів.
- Інтеграція в браузери.

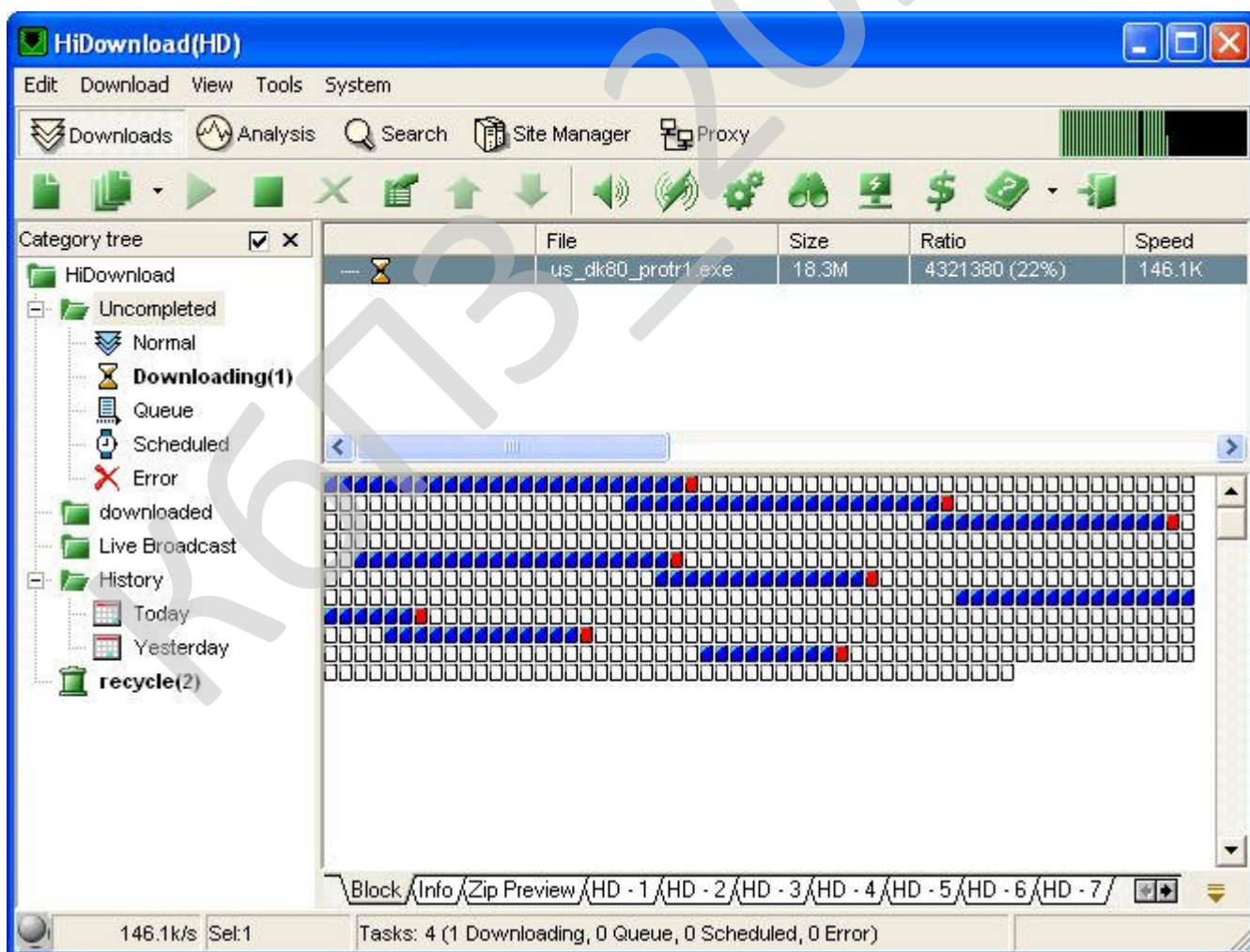


Рисунок 2.4 – Інтерфейс користувача HiDownload

- Автоматична перевірка на віруси завантажених файлів.
- Автоматичний запуск завантажених файлів.
- Часткове завантаження ZIP-архівів, перегляд їхнього вмісту перед завантаженням.
- Інтеграція в браузері.

Деякі висновки

У даному огляді розглянуто лише незначну частину із усього достатку програм для завантаження файлів з мережі. Проте, неважко помітити, що основні функції подібного софта дуже схожі – коректне збереження з можливістю дозавантаження будь-якого виду контенту. Додатки відрізняються лише функціональністю у вигляді додаткових можливостей, необхідність яких для себе вибирає сам користувач. Незайвою буває функція роботи з розкладу. Багато менеджерів завантаження дозволяють указати час, коли необхідно скачати файли, що буває зручно при використанні dial-up доступу в інтернет: список файлів для завантаження можна скласти вдень і запланувати завантаження на нічний час, коли діє більш дешевий тариф на послуги доступу в мережу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

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

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

Win 64-відладник і збирач для C++

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

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка 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.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

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

Поліпшена кроссплатформеність

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

						ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			22

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуального завантаження даних з мережі.

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

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

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

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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) сформуванати висновки про виконаний обсяг робіт та одержані результати.

КБГПЗ-2023

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

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 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.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

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 завантажень;

– менеджер сайтів для керування паролями й папками для збереження;

– керування швидкістю завантаження, автоматичний режим для комфортної роботи із браузером;

– можливість установки пріоритетів для завантажень;

– регульований режим для динамічного керування швидкістю завантаження;

– розпізнавання контенту при роботі через HTTP;

– робота з командним рядком;

– звонилка 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 завантажених файлів на відновлення;

– можливість послухати/подивитися музичні й відео файли в процесі завантаження. Автоматичне одержання інформації про MP3 файлах при старті завантаження;

– можливість завантажувати html-сторінки разом з картинками;

– Smart Pause. Можливість тимчасової зупинки завантажуваних файлів з наступним їхнім стартом у тім же стані;

– широкі можливості налаштування параметрів з'єднання, HTTP, HTTPS і FTP протоколів;

– робота через HTTP і FTP проксі-сервера, підтримка NTLM і NTLM-проксі автентифікації;

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

– можливість звірення MD5 суми завантаженого файлу.

Розроблювальний менеджер завантажень реалізує високу швидкість завантаження, за рахунок використання наступних технологій:

– інтелектуальна багатосекційність. Менеджер завантажень 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.

Крім того, у розроблювальному менеджері завантажень реалізуються 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;

– Bar – панель інструментів для Internet Explorer і Mozilla Firefox.

Розглянемо як реалізується «активний наочний кошик». Концепція полягає в тім, що реалізовано повноцінну роботу з менеджером завантажень без відкриття головного вікна програми.

Основним елементом даної технології є плаваюче віконце, що містить у собі наступні об'єкти:

– індикатор швидкості завантаження;

– кошик додавання завантажень;

– індикатори активних завантажень із можливістю керування цими завантаженнями;

– меню керування програмою;

– короткий список завантажень із можливістю їхнього старту.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

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

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

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

4. Вікно статусів, яке включає в себе:

- Перелік усіх завантажень.
- Категорії файлів, які завантажуються (програми, архіви, музика, відео).
- Топ завантажень (програми, архіви, музика, відео, пошук).
- Новини.
- Стан (завантаження, чекання завантаження, заплановано, помилки, пауза, завантажено).

- Історія.
- Видаленні завантаження.

5. Вікно завантажень:

- Ім'я файлу.
- Стан завантаження.

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

6. Основна панель менеджера завантажень:

- Файл (топ завантажень, імпорт завантажень, імпорт, експорт, вихід).
- Завантаження (додати завантаження, додати групу завантажень, перевірити оновлення, пауза, видалити, видалити разом з файлом, запланувати, перезавантажити заново, копіювати URL, відкрити файл, відкрити папку, скопіювати файл, перемістити файл, меню Windows, робота з архівом, коментарі, знайти дзеркала, додати в менеджер сайтів, властивості).
- Дії (стартувати все, призупинити все, тимчасова зупинка завантажуваних, видалити все, знайти, знайти далі, швидкість).
- Вид (налаштування кнопок, сортування списку, список завантажень, звук, категорії, лог завантажень, плаваюче вікно, скіни, мова інтерфейсу: українська, 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).
- 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).

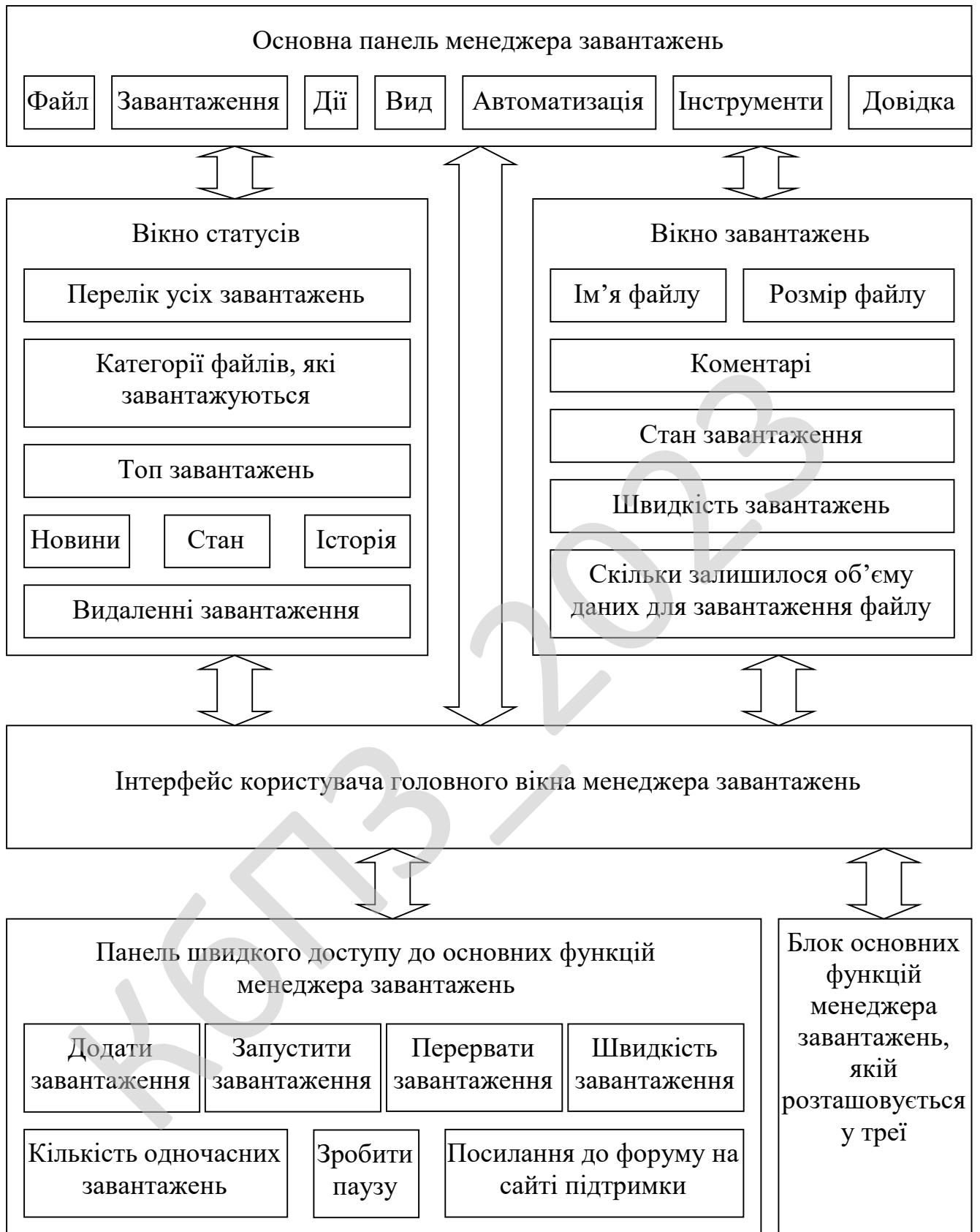


Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Дано визначення функціональної схеми. Функціональна схема – схема що роз'ясняє певні процеси, що протікають в окремих функціональних блоках програмного продукту або в програмному продукті в цілому.

Схемами функціональними користуються для вивчення принципів роботи програмного продукту, а також при його впровадженні й супроводі.

Нижче розглянемо її більш докладно.

Функціонально система взаємодіє з наступними блоками:

- функціональний блок динамічного багатопотокового завантаження;
- функціональний блок дозавантаження після обриву зв'язку з HTTP, HTTPS і FTP серверів;
- функціональний блок робота з розкладом;
- 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 керування категоріями завантажень;
- 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;
- докладний лог-файл на кожне завантаження;
- функціональний блок історії завантажень;

- 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 перевірки завантажених файлів на відновлення;
- 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 тимчасової зупинки завантажуваних файлів з наступним їхнім стартом у тому же стані;
- 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.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33



Рисунок 3.2 – Функціональна схема системи

3.4 Розробка діаграми процесів

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



Рисунок 3.3 – Діаграма взаємодії процесів

З діаграми видно, що першим процесом, який запускається у системі, є процес виведення вікна завантажень.

Від цього процесу запускаються наступні процеси:

- Процес завантаження.
- Процес встановлення параметрів.
- Процес роботи з категоріями завантажуваних файлів.

Процес завантаження взаємодіє з наступними процесами:

- Процес видалення завантаження.
- Процес виведення параметрів завантаження.
- Процес створення нової задачі.

– Процес запуску завантаження.

Останній процес у свою чергу взаємодіє з наступними процесами:

– Процес виведення вікна завантаження.

– Процес призупинки завантаження.

Процес встановлення параметрів взаємодіє з наступними процесами:

– Процес зміни параметрів програми.

– Процес зміни параметрів завантажень.

– Процес зміни параметрів проксі-сервера.

Процес роботи з категоріями завантажуваних файлів взаємодіє з наступними процесами:

– Процес редагування категорій.

– Процес створення нової категорії для завантажень.

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

КБГІЗ-2023

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

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 відбувається вибір однієї з наступних дій:

- Завантажити файл.
- Редагувати категорії.
- Змінити параметри.

Розглянемо ці дії більш детально.

Якщо необхідно завантажити файл, то виконується наступна послідовність дій:

- Вводиться посилання на файл.
- Вводиться директорія для збереження, яка створюється, або обирається з тих, які існують.
- Відбувається вибір категорії.
- Відбувається вибір параметрів завантаження.
- Запускається процес завантаження.
- Виводиться вікно процесу завантаження.

Якщо файл завантажено то відбувається збереження файлу.

У іншому випадку виводиться повідомлення про помилку.

Якщо потрібно редагувати категорії, то відбувається редагування існуючих категорій, їх видалення, або створення нових категорій.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Якщо є необхідність у заміні параметрів програми, завантажень та проксі-сервера відбувається ця зміна параметрів.

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

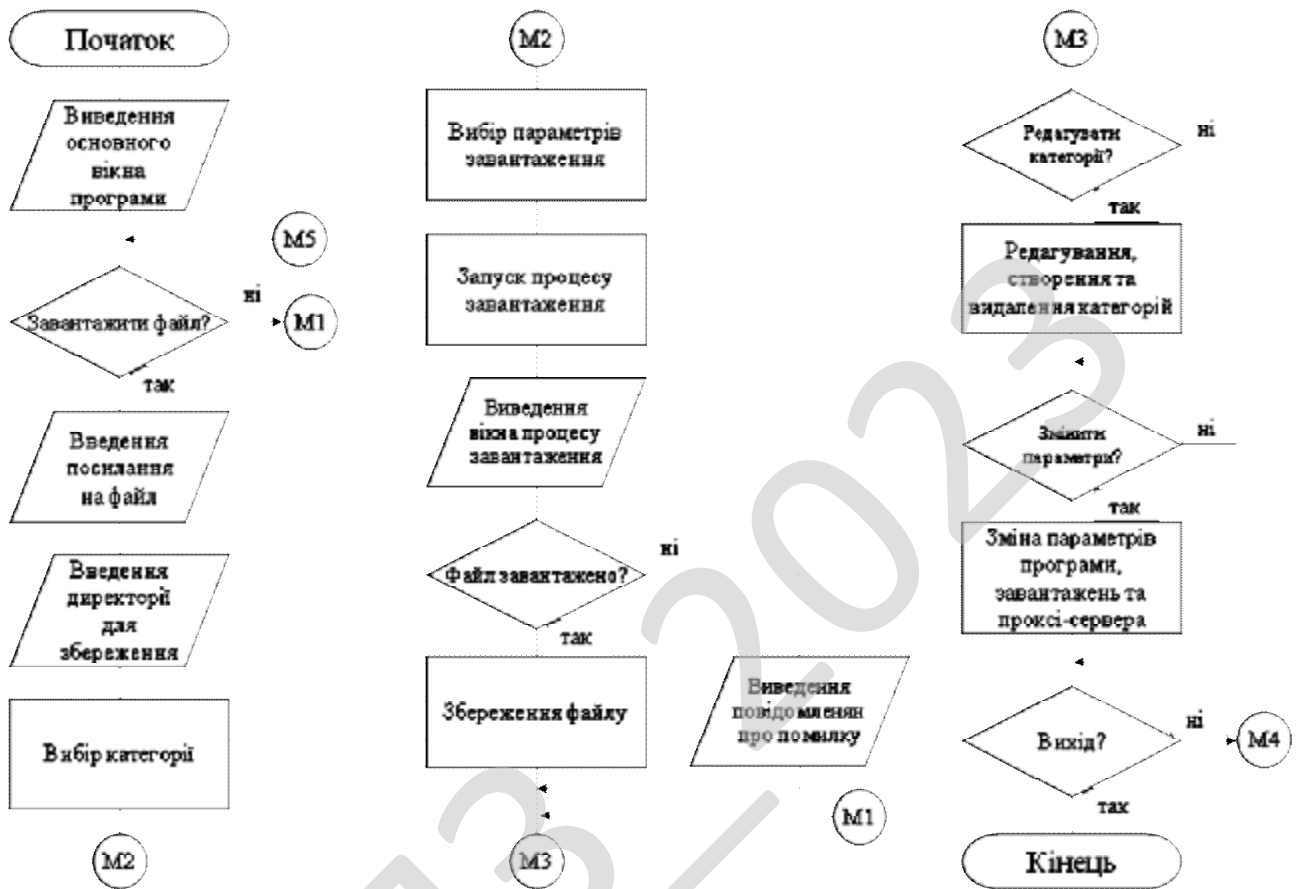


Рисунок 4.1 – Блок-схема роботи основної програми

На рисунку 4.2 зображено блок-схему роботи підпрограми процесу завантаження. З блок-схеми ми бачимо, що спершу відбувається перевірка, чи є файл на сервері.

Якщо він є то відбувається наступна послідовність ітерацій:

- Вноситься завантаження у категорію «Завантажуються».
- Одержується розмір файлу.
- Виводиться розмір файлу.
- Одержуються дані про швидкість даного з'єднання.

- Виводяться дані про швидкість з'єднання.
- Обчислюється час завантаження.
- Виводиться приблизно час завантаження.
- Відбувається завантаження файлу.
- Виводиться процес бар витраченого часу, та часу, що залишився.

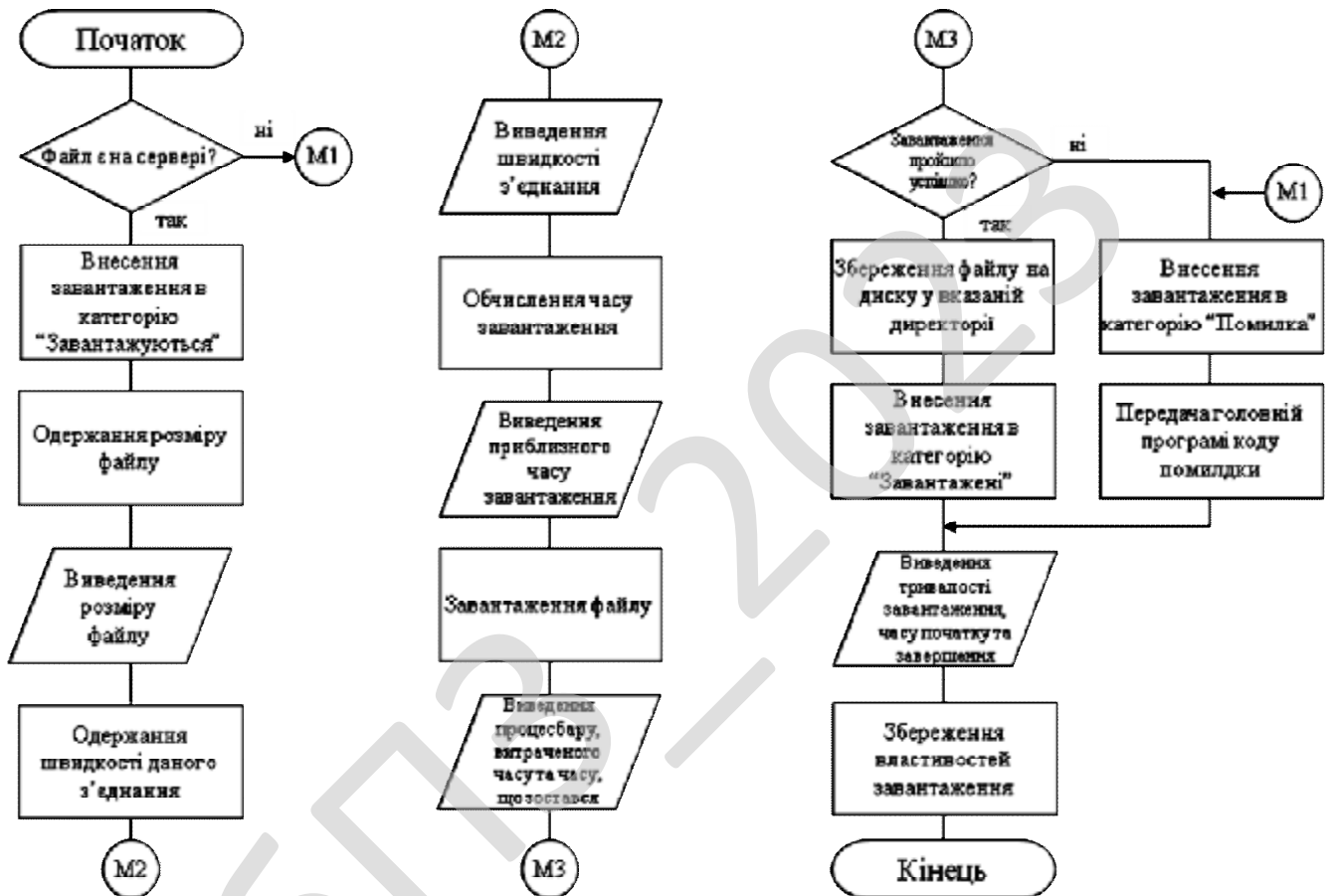


Рисунок 4.2 – Блок-схема роботи підпрограми процесу завантаження

Якщо завантаження пройшло успішно то відбуваються наступні дії:

- Зберігається файл на диску у вказаній директорії.
- Вноситься завантаження в категорію «Завантажені».

У іншому випадку відбуваються наступні дії:

- Вноситься завантаження у категорію «Помилка».
- Головній програмі передається код помилки.


```

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;

//Виведення статусбара

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;

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

```

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;
        end;
    end;
end;

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

```

        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;
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;
end;

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

```

//Рисування 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;

//Вибір вікна у якому потрібно працювати

procedure TfMain.lvTasksSelectItem(Sender: TObject; Item: TListItem; Selected:
Boolean);
var

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

```

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

```

					БКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

```

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;

```

Робимо потік

З оброблювачем поки почекаємо, а напишемо саме складне – клас для потоку.

```

{$R *.dfm}
// -----
type
  TDownloader = class(TThread)
  protected
    procedure Execute; override;
  public
    property URL:string read FURL write FURL;
    property ToFolder:string read FToFolder write FToFolder;
  end;

```

Перші два рядки зроблені для того, щоб було видно, де вписати код. І натискаємо Ctrl+Shift+C. Delphi допише небагато коду. Він тепер буде виглядати так:

```

type
  TDownloader = class(TThread)
  private
    FToFolder: string;
    FURL: string;
  protected
    procedure Execute; override;
  published
  public
    property URL:string read FURL write FURL;
    property ToFolder:string read FToFolder write FToFolder;
  end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    SaveDialog1.FileName:=copy(Edit1.Text,LastDelimiter('\? ',Edit1.Text)+1,maxint);

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

    if SaveDialog1.Execute then
        Edit2.Text:=SaveDialog1.FileName;
end;
{ TDownloader }

```

```

procedure TDownloader.Execute;
begin
    //тут ми почнемо писати код роботи
end;

```

Компонент `idHTTP` був кинутий на форму тільки з однією метою – щоб Delphi додала всі заголовні файли в `uses`. Потім його можна буде видалити. Але можна й самостійно вписати в `uses` файл `idHTTP`.

Головний код потоку

Отже, код оброблювача:

```

procedure TDownloader.Execute;
var
    http:TIdHTTP;
    str:TFileStream;
begin
    //Створимо клас для завантаження
    http:=TIdHTTP.Create(nil);
    //каталог, куди файл покласти
    ForceDirectories(ExtractFileDir(ToFolder));
    //Потік для збереження
    str:=TFileStream.Create(ToFolder, fmCreate);
    try
        //Качаємо
        http.Get(url, str);
    finally
        //Нас учили чистити за собою
        http.Free;
        str.Free;
    end;
end;

```

В цій процедурі не можна прямо звертатися до компонентів форми й інших потоків. Це робиться спеціальним чином, що називається синхронізацією.

Запускаємо потік

І нарешті, оброблювач для кнопки:

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

procedure TForm1.Button2Click(Sender: TObject);
var d:TDownloader;
begin
    //Створимо клас потоку.
    //Потік для початку буде зупинений
    d:=TDownloader.Create(true);
    //Передамо параметри потоку
    d.URL:=Edit1.Text;
    d.ToFolder:=Edit2.Text;
    //Потік повинен видалити себе по завершенню своєї роботи
    d.FreeOnTerminate:=true;
    //І запустимо його на завантаження.
    d.Resume;
    //Тепер із процедури ми вийдемо, але потік працює
    //і живе своїм життям
end;

```

Тепер наш код уже може качати, але в нас немає прогресу, немає повідомлення про завершення завантаження.

Додаткові можливості

Додамо для почала повідомлення про завантаження. Додамо в public частину форми додамо рядок:

```

public
    { public declarations }
    procedure thrTerminate(Sender:TObject);
end;

```

і натиснемо Ctrl+Shift+C.

З'явиться новий оброблювач, що ми доповнимо одним рядком. Я виводжу просто повідомлення про готовність. Ця процедура буде викликатися, коли потік виконає всю свою роботу (завершиться процедура Execute). Тільки в цій процедурі можна одночасно звертатися до компонентів форми й даним потоку.

```

procedure TForm1.thrTerminate(Sender: TObject);
begin
    ShowMessage('Готово');
end;

```

І додамо її виклик в оброблювачі кнопки запуску:

```

//Потік повинен видалити себе по завершенню своєї роботи
d.FreeOnTerminate:=true;

```

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50


```

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.ru');
connect(Sock,sockAddr,sizeof(TSockA ddr));
if(Sock <> INVALID_SOCKET) then
begin
StrPCopy(Send_buf,'GET /111.jpg HTTP/1.1'#13#10 +
'Host: ismily.ru'#13#10+
'User-Agent: Opera/9.63 (Windows NT 5.1; U; ru) 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: ru-RU,ru;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;

```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Blowfish, який є симетричним алгоритмом шифрування, тобто таким, у якому ключ шифрування дорівнює ключу дешифрування. Він є мережею Фейштеля, у якій кількість ітерацій дорівнює 16. Довжина блоку дорівнює 64 бітам, ключ може мати будь-яку довжину в межах

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

448 біт. Хоча перед початком будь-якого шифрування виконується складна фаза ініціалізації, саме шифрування даних виконується досить швидко.

Алгоритм призначений в основному для додатків, у яких ключ міняється нечасто, до того ж існує фаза початкового рукостискання, під час якої відбувається автентифікація сторін і узгодження загальних параметрів і секретів. При реалізації на 32-бітних мікропроцесорах з більшим кешем даних Blowfish значно швидше DES.

Алгоритм складається із двох частин: розширення ключа й шифрування даних. Розширення ключа перетворює ключ довжиною, принаймні, 448 біт у кілька масивів підключів загальною довжиною 4168 байт.

В основі алгоритму лежить мережа Фейштеля з 16 ітераціями. Кожна ітерація складається з перестановки, що залежить від ключа, і підстановки, що залежить від ключа й даних. Операціями є XOR і додавання 32-бітних слів.

Blowfish використовує велику кількість підключів. Ці ключі повинні бути обчислені заздалегідь, до початку будь-якого шифрування або дешифрування даних. Елементи алгоритму:

1. P – масив, що складається з вісімнадцяти 32-бітних підключів:

$$P_1, P_2, \dots, P_{18}.$$

2. Чотири 32-бітних S -boxes с 256 входами кожний. Перший індекс означає номер S -box, другий індекс – номер входу.

$$S_{1,0}, S_{1,1}, \dots, S_{1,255};$$

$$S_{2,0}, S_{2,1}, \dots, S_{2,255};$$

$$S_{3,0}, S_{3,1}, \dots, S_{3,255};$$

$$S_{4,0}, S_{4,1}, \dots, S_{4,255};$$

Шифрування

Входом є 64-бітний елемент даних X , що ділиться на дві 32-бітні половини, X_l і X_r .

$$X_l = X_l \text{ XOR } P_i$$

$$X_r = F(X_l) \text{ XOR } X_r$$

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Swap X_l and X_r

Функція F

Розділити X_l на чотири 8-бітних елементи A, B, C, D .

$$F(X_l) = ((S_{1,A} + S_{2,B} \bmod 2^{32}) \text{ XOR } S_{3,C}) + S_{4,D} \bmod 2^{32}$$

Дешифрування відрізняється від шифрування тим, що P_i використовуються у зворотному порядку.

Генерація підключів

Підключи обчислюються з використанням самого алгоритму Blowfish.

1. Ініціалізувати перший P -масив і чотири S -boxes фіксовані рядки.
2. Виконати операцію XOR P_1 з першими 32 бітами ключа, операцію XOR P_2 із другими 32 бітами ключа й т.д. Повторювати цикл доти, поки весь P -масив не буде побітово складний з усіма бітами ключа. Для коротких ключів виконується конкатенація ключа із самим собою.
3. Зашифрувати нульовий рядок алгоритмом Blowfish, використовуючи підключи, описані в пунктах (1) і (2).
4. Замінити P_1 і P_2 виходом, отриманим на кроці (3).
5. Зашифрувати вихід кроку (3), використовуючи алгоритм Blowfish з модифікованими підключами.
6. Замінити P_3 і P_4 виходом, отриманим на кроці (5).
7. Продовжити процес, замінюючи всі елементи P -масиву, а потім всі чотири S -boxes, виходами відповідним чином модифікованого алгоритму Blowfish.

Для створення всіх підключів потрібна 521 ітерація.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено користувацький інтерфейс головного вікна програми.

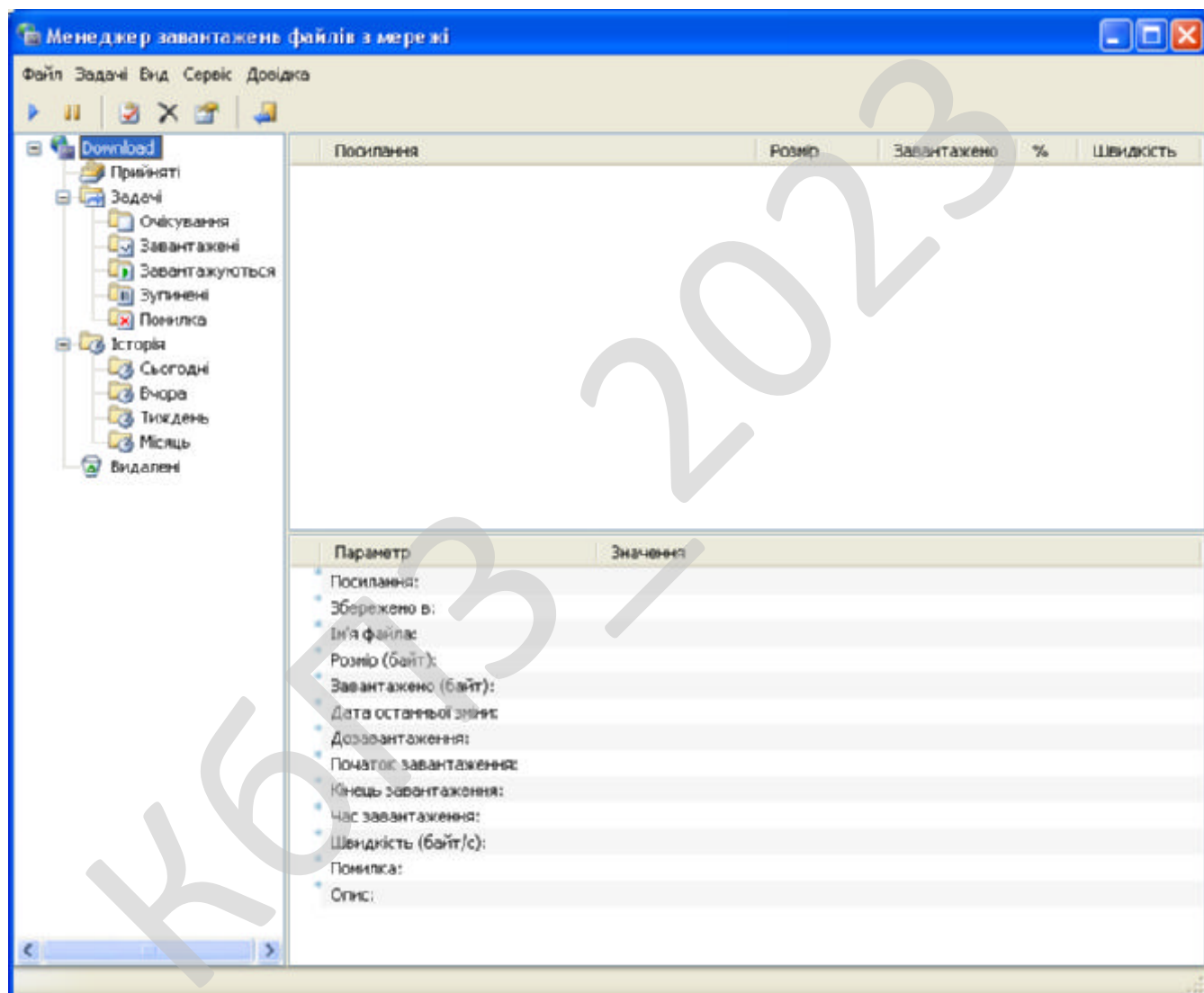


Рисунок 5.1 – Головне вікно програми

З нього ми бачимо, що на головній формі знаходяться наступні функціональні блоки:

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

- Блок роботи з параметрами програми.
- Блок швидкого доступу до основних функцій програми.
- Блок категорій.
- Блок вікна візуалізації завантаження файлів.
- Блок обирання параметрів.

На рисунку 5.2 наведено вікно створення нової задачі. З нього видно, що задаються наступні параметри:

- Адреса посилання.
- Директорія зберігання.
- Ім'я файлу, який завантажується.
- Вибір категорії.
- Опис.
- Особливі параметри входу на сервер.
- Завдання часу початку завантаження.

Рисунок 5.2 – Вікно створення нової задачі

На рисунках 5.3-5.8 наведені наступні скріншоти:

- Головне вікно програми (завантаження файлу).
- Вікно процесу завантаження файлу.
- Вікно параметрів програми (вкладка «Загальні»).
- Вікно параметрів програми (вкладка «Завантаження»).
- Вікно довідки

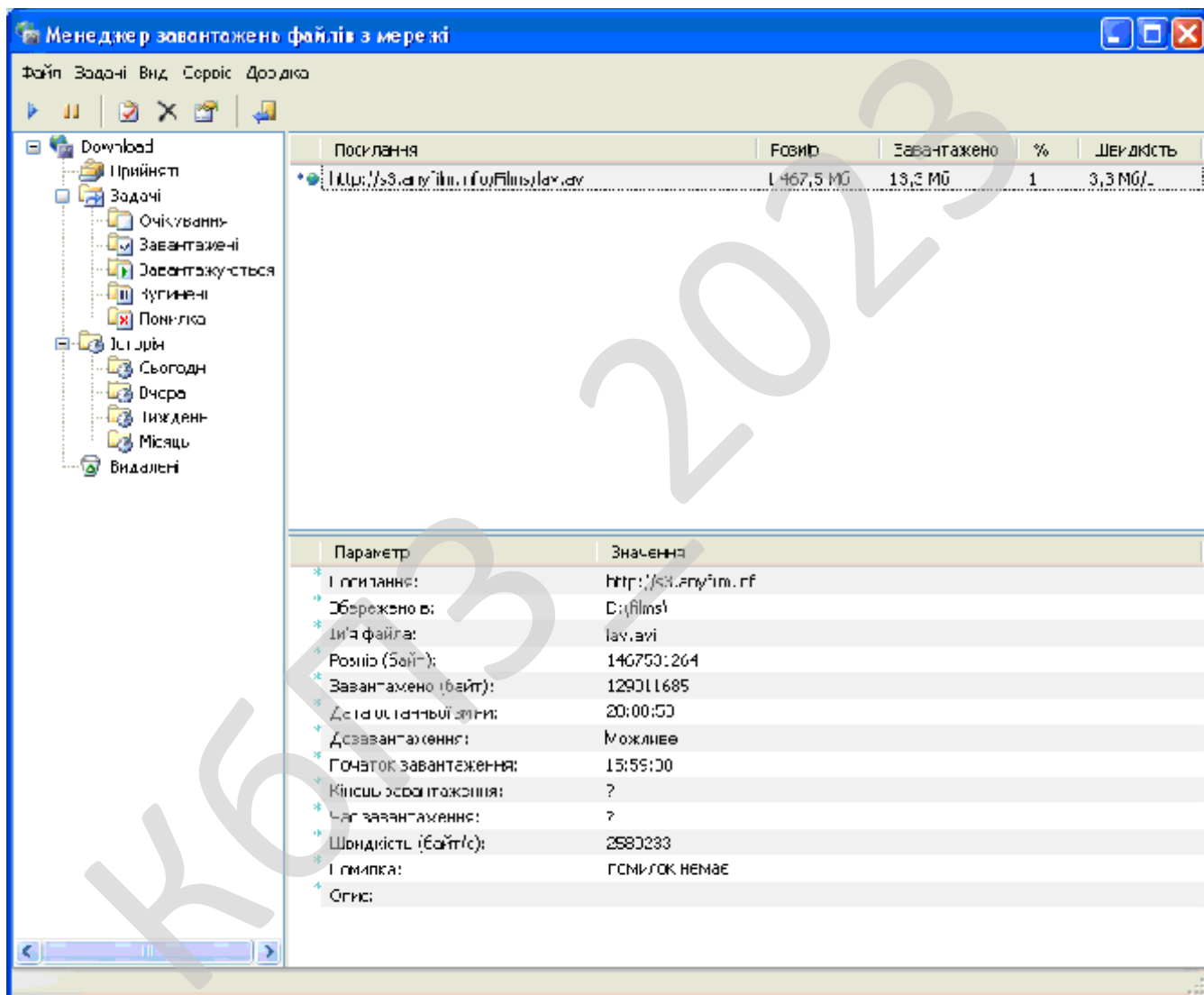


Рисунок 5.3 – Головне вікно програми (завантаження файлу)

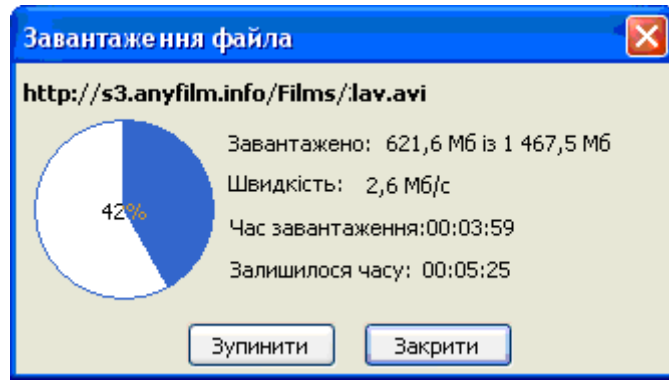


Рисунок 5.4 – Вікно процесу завантаження файлу

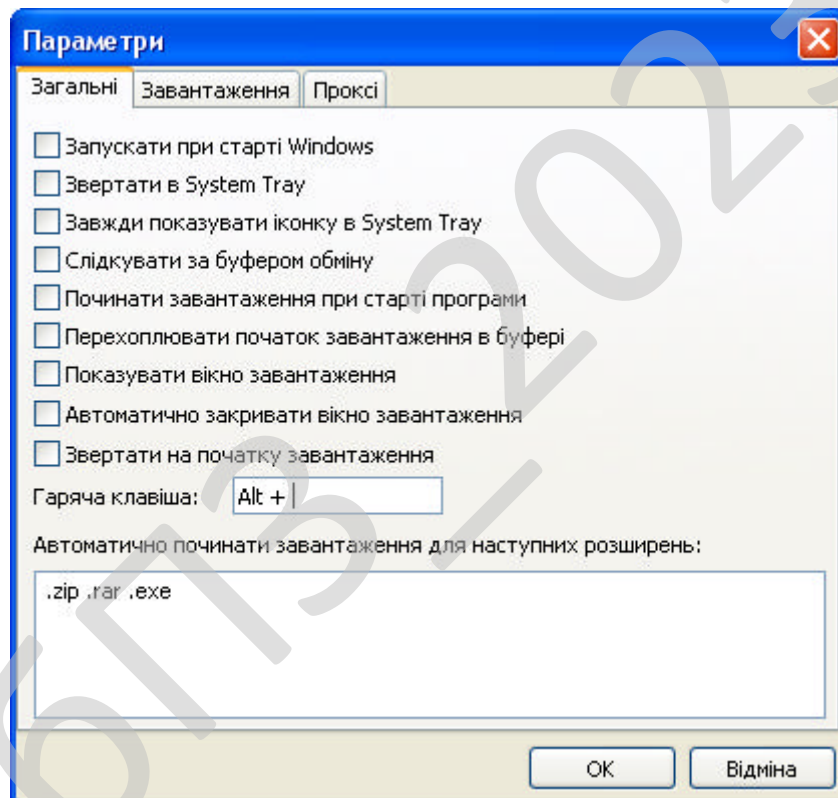


Рисунок 5.5 – Вікно параметрів програми (вкладка «Загальні»)

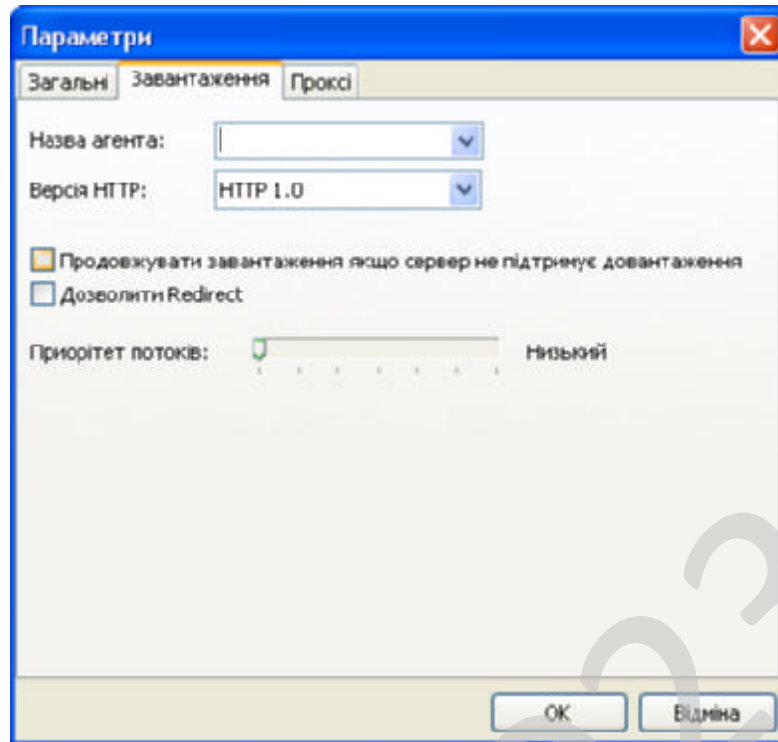


Рисунок 5.6 – Вікно параметрів програми (вкладка «Завантаження»)

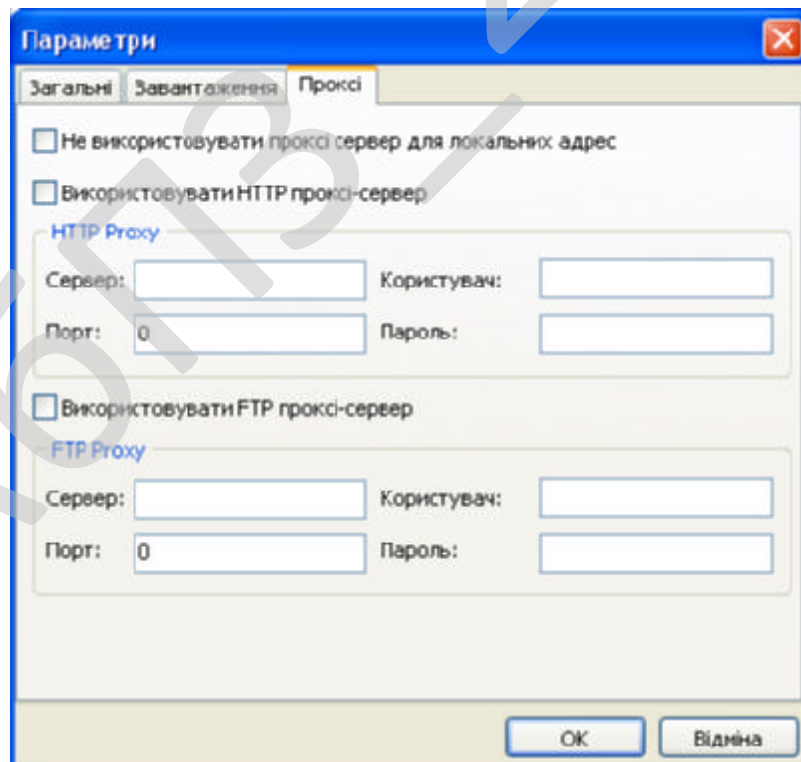


Рисунок 5.7 – Вікно параметрів програми (вкладка «Проксі»)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Центральноукраїнський національний технічний
університет
Кафедра кібербезпеки та програмного забезпечення
ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему " Дослідження та програмна реалізація системи
інтелектуального завантаження даних з мережі "
ОПП «Комп'ютерні науки»
спеціальності 122 «Комп'ютерні науки»
Виконав: Гладин В.В.
Науковий керівник: Смірнова Т.В.
Кропивницький - 2023

ок

Рисунок 5.8 – Вікно довідки

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		61

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуального завантаження даних з мережі.

Метою розробки є дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі.

Об'єктом дослідження є процес інтелектуального завантаження даних з мережі.

Предметом дослідження є методи інтелектуального завантаження даних з мережі.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод інтелектуального завантаження даних з мережі.
- Розроблено вітчизняний продукт інтелектуального завантаження даних з мережі, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

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

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

В магістерській роботі було проведене дослідження та виконана програмна реалізація системи інтелектуального завантаження даних з мережі. Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	40
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

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

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	40000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням

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

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

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

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

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

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

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

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

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

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

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

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

$$T_{РП} = 0,3 \cdot 3,22 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 90 = 183 \text{ люд/день}$$

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	183	Ф 7.1-7.4
Впровадження	13	Д13
Всього	224	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

$$Ч = \frac{224 \cdot 1}{60 - 5} = 4 \text{ ставки}$$

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

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

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	7	630	10,5
Монітор	60	7	420	7
Клавіатура	30	7	210	3,5
Маніпулятор «мишка»	30	7	210	3,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	450	1125	18,75
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	54,75

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

$$\Phi_{\text{др}}^c = \frac{55 \cdot 3}{1,2} = 137,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

$$Ч_{ел}=137,5/(60\cdot 8)=0,3 \text{ ставки}$$

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

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

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

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

Продовження таблиці 7.4

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

Складемо штатний розклад виконавців:

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	15000	45000
Продакт-менеджер	0,25	14000	10500
Інженер-програміст	4,0	14648,5	175782
Інженер-електронщик	0,3	14000	12600
Інженер-системотехнік	0,25	14000	10500
Адміністратор мережі	0,5	14000	21000
Системний програміст	0,25	14000	10500
Дизайнер WEB	0,25	14000	10500
Інженер-верстальник	0,25	14000	10500
Бухгалтер-економіст	0,5	14966	22449
Всього за період розробки	$R_{cn}=7,55$	-	$\Phi_{роб}=329331$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{329331}{7,55 \cdot 60} = 727 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

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

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

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми Компбест за 16.11.23 – джерело <https://compbest.com.ua/>.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	Intel Core™ i3 10105 (BX8070110105) 1200, 4 ядра, 8 потоків, 3.7 GHz, 4.4 GHz, TDP – 65 Вт, 14nm, BOX	-
Системна плата	ASUS PRIME H510M-K сокет – 1200, DDR4, 3200 MHz, LAN – 1 Гбіт/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	Intel UHD Graphics 630	-
Жорсткий диск	SSD M.2 2280 512GB LEVEN (JP600PCIE512GB) Серія – JP600, 512 GB, 3D TLC NAND, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Fury Beast Black Kingston Fury (ex.HyperX) (KF432C16BB/8)	-
Блок живлення	Gamemax 500W (GM-500B) ATX 12V v2.3, 500 Вт, 20+4 pin, CPU – 4+4pin, GPU – 1x6 pin, SATA – 3, Peripheral – 2, +12V1 – 20A, 1x120 мм, 150 x 140 x 86 мм	-
Корпус	Vinga CS210B, Miditower, ATX, Micro – ATX, Mini – ITX, Слотів розширення – 7	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4- B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	-

						БКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			73

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	LG W2363V-WF Wide LCD 2ms, 70 000:1, 300кд/м2, 170/160, D-Sub / Glossy White	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	114885,1

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	40000	10	4000
Разом	$K_p = 1596075$		$A_p = 140140$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців

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

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 727 \cdot 224 / 40 = 4071 \text{ грн}$$

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

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

де H_q – норматив додаткової зарплати, %

$$Z_d = 4071 \cdot 10 \cdot 0,01 = 407 \text{ грн}$$

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

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

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(4071+407) = 985 \text{ грн}$$

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

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

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 4071 \cdot 15 \cdot 0,01 = 611 \text{ грн}$$

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

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

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

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

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

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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

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

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 49,2 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 49,2 грн/шт.

$$Z_{M2} = 49,2 \cdot 10 = 492 \text{ грн.}$$

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

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

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

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

$$Z_M = (105 + 492 + 1702) / 40 = 57 \text{ грн.}$$

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

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

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

$$O_n = 4071 \cdot 15 \cdot 0,01 = 611 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 40$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 140140 \cdot 3 / (40 \cdot 12) = 876 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

$$C_n = 4071 + 407 + 985 + 611 + 57 + 611 + 876 = 7618 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 55%

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

де P_n – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 7618 = 4190 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	Z_o	4071
2. Додаткова зарплата виконавців	Z_d	407
3. Відрахування на соціальні потреби	C_{oc}	985
4. Загальногосподарські витрати	Γ_{ocn}	611
5. Витрати на матеріали	Z_M	57
6. Освоєння нових операційних систем, мов програмування	O_n	611
7. Амортизація основних фондів	A_M	876
8. Повна собівартість програмного забезпечення	C_n	7618
9. Плановий прибуток	P_p	4190
10. Ціна підприємства $C_n = C_n + P_p$	C_n	11808
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{дв} \cdot C_n$	$ПДВ$	2361,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	15998

Витрати на оплату праці:

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

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

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 400 годин на рік до 250 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 400 \cdot 100 \cdot 1,1 \cdot 1,22 = 53680 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 250 \cdot 100 \cdot 1,1 \cdot 1,22 = 33550 \text{ грн.}$$

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

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

Після впровадження програмного продукту витрати на електроенергію не змінюються і складають $Z_{ел \text{ баз}} = Z_{ел \text{ нов}}$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	15998	–	3999,5
Всього відрахувань	-	–	15998	–	3999,5

$$T_{cn} = \frac{15998}{53690 - 37550} = 0,99 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	40
2. Повна собівартість розробленої програми	Грн	7618
3. Ціна розробленої програми	Грн.	11808
4. Плановий прибуток від реалізації розробленої програми	Грн.	4190
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1596075
7. Загальний прибуток від реалізації програмної продукції	Грн.	167600
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	132565
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,4
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	15998
11. Величина економічного ефекту у користувача програмної продукції	Грн.	12141
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,99

7.9 Висновки

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

КБГПЗ-2023

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки.

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

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

Ці шкідливі фактори можуть привести до професійних захворювань.

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп'ютером і поблизу нього. В сучасних умовах взаємодія людини з технікою значно ускладнилась, що вимагає комплексного підходу, який передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищесказане в повній мірі відноситься й до системи «людина–комп'ютер–середовище».

Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище [3]. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму, випромінювання.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

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

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

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

Таблиця 8.1 – Розміри приміщення

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

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

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6
Обсяг, V	м ³	не менше 20.0	20,4

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

У зазначеному приміщенні працює 3 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та а об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам

(Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2], НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °С	Воло- гість,%	Швидкість повітря, м/с	Температура, °С	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	21-23	40-58	0,11
Тепла	23-25	50-70	0,1	23-25	50-65	0,1

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер Canon LBP-6030B (8468B006), електродвигуни вентиляторів ЕОМ.

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

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

8.5 Розрахункова частина

Завдання: розрахувати штучне освітлення робочого приміщення.

Початкові дані: ширина робочого приміщення: 3 м.; довжина – 6 м.; висота – 3,4 м.

Розрахунок штучного освітлення проведемо за методом коефіцієнта використання світлового потоку.

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

$$F=ESKZ/n,$$

де:

F – світловий потік, що розраховується, Лм;

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

S – площа освітлюваного приміщення (у нашому випадку $S=3 \times 6 = 18$ м²);

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

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

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін} = 50\%$ і $\rho_{стелі} = 50\%$ [6].

Обчислимо індекс приміщення за формулою:

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

де:

S – площа приміщення, $S = 18 \text{ м}^2$;

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

A – ширина приміщення, $A = 3 \text{ м}$;

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

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

Знаючи індекс приміщення, за знаходимо $n = 0,29$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників відповідного типу). Підставимо всі значення у формулу, визначимо світловий потік: $F=19370 \text{ Лм}$.

Для штучного освітлення приміщення використовуються *LED панель MAXUS ASSISTANCE PRO 40W 5000K WHITE (M1051440531)*, світловий потік яких $F_{\text{л}} = 4000 \text{ Лм}$.

Число світильників визначається по формулі:

$$N=F/F_{\text{л}}$$

де: F – світловий потік,

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

$$N= 19370/ 4000=4,84 \text{ шт.}$$

Приймаємо необхідну кількість світильників 5 шт.

Висновки до розділу

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

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок захисного штучного освітлення. Розроблено заходи з охорони праці.

					БКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Список використаних джерел інформації

1. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПН 3.3.2.007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
4. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>
6. Оришака О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення: 16.06.2023).
7. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / Оришака О. В., Марченко К.М.; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення. – Кропивницький : ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу (URI): <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення: 16.06.2023).

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуального завантаження даних з мережі.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтелектуального завантаження даних з мережі.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуального завантаження даних з мережі.
- Досліджена система інтелектуального завантаження даних з мережі.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуального завантаження даних з мережі.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуального завантаження даних з мережі.

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

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

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

Програма реалізована на мові високого рівня RAD Studio Delphi 10. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Blowfish.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 12141 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,99 роки.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гладин В.В. Дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
8. 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.
9. 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.
10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools».

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

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

12. 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.

13. 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.

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

15. 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.

16. 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.

17. 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.

					БКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

18. Smirnov O., Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.*

27. 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.

28. 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.

29. 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.

30. 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.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

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

32. 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.*

33. 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*

34. 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.*

35. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.*

36. Смірнов О.А., 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.

37. 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.*

38. 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.*

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

39. 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.

40. 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).**

41. 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

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

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

44. 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.

45. 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.

					ВКРМ-122.23.0028.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

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

47. 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.

48. 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.

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

50. 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.

51. 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.

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.23.0028.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Гладин В.В.				Дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М-1			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інтелектуального завантаження даних з мережі.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 32-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інтелектуального завантаження даних з мережі.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

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

					ВКРМ-122.23.0028.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи інтелектуального завантаження даних з мережі;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.23.0028.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.

					ВКРМ-122.23.0028.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-122.23.0028.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 101 аркуш.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 11.12.2023 р.

					ВКРМ-122.23.0028.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи інтелектуального завантаження даних з мережі*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 53

Літера: РП

Кропивницький – 2023 року

uMain.pas - головна програма

```
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;

```

```
//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;
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;
end;

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

//Відкриття папки завантаження

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

//Відкриття файлу завантаження

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

end.

```

uDownload.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.
```

uProcedures.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;

```

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

FreeMem(Data, DataSize);

end;

end;

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

//Дерево каталогів

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

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

  TitleName := 'Виберіть директорію';

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

  lpItemID := SHBrowseForFolder(BrowseInfo);

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

    Result := TempPath;
  end else
  begin
    Result := '';
  end;
end;

end;

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

//Створення імені файлу

function CreateFileName(Url : String) : String;
var
  i : Integer;
begin
  Result := '';

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

  if Url = '' then Exit;

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

```

uObjects.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            : TString;
    Directory      : TString;
    Task           : TList;

    procedure Save;
    procedure Load;
end;

var
    Options : TOptions;

implementation

uses uMain;

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

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

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

begin

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

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

```

```

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

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

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

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

```

```

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

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

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

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

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

if Url <> nil then
begin
    Parent := Xml.DocumentElement.AddChild('Url');

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

    for i := 0 to n do
    begin
        Child := Parent.AddChild('Url');
        Child.Text := Url.Strings[i];

    end;

end;

if Directory <> nil then
begin
    Parent := Xml.DocumentElement.AddChild('Directory');

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

    for i := 0 to n do
    begin
        Child := Parent.AddChild('Directory');
        Child.Text := Directory.Strings[i];

    end;

end;

if Task <> nil then
begin
    Parent := Xml.DocumentElement.AddChild('Tasks');

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

```



```

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.

```

uLoading.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;
    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.
```

uAddCategory.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;
    Memol: TMemo;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  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 - 2023

uEditCategory.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.
```

uAddTask.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! Хочете 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.
```

uEditTask.pas - редагування задачі

```

unit uEditTask;

interface

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

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

var
  fEditTask: TfEditTask;

implementation

uses uMain, uProcedures;

{$R *.dfm}

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

procedure TfEditTask.btnCancelClick(Sender: TObject);
begin

```

```

Close;
end;

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

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

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

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

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

```

```

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

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

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

```

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

end.
```

К6П3-2023

uOptions.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;

```

```
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 - 2023

uThreads.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 := 'Помилко немає';

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

except

    on E : Exception do
    begin

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

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

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

    end;

end;

HTTP.Free;

fMain.RefreshTasks;

end;

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

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

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

    Data := P;

    inherited Create(CreateSuspended);

end;

```

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

```

Count      : Integer;

begin

  if AWorkMode = wmRead then
  begin

    Data.LoadSize := AWorkCount;

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

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

    end;

    if Data.Status = tsStoped then FTP.Disconnect;

end;

if Data.Status = tsStoped then FTP.Disconnect;

end;

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

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

  Tick := GetTickCount;

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

end;

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

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

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

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

end;

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

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

  Data := P;

  inherited Create(CreateSuspended);

end;

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

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

begin

```

```

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

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

end;

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

end else FTP.Username := 'anonymous';

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

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

    FTP.Disconnect;

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

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

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

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

        end;

end;

FTP.Free;
Details.Free;

fMain.RefreshTasks;

end;

end.

```

uThreads.pas - довідка

```
unit uAbout;

interface

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

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

var
  fAbout: TfAbout;

implementation

uses uObjects;

{$R *.dfm}

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

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

procedure TfAbout.FormCreate(Sender: TObject);
begin
  Label1.Caption:=Options.Name;
  Memo1.Clear;
  Memo1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Дослідження та програмна реалізація системи інтелектуального
завантаження даних з мережі');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Смірнова Т.В. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Білорус Ярослав Олександрович');
  Memo1.Lines.Add(' гр. КН-22М-1');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Кропивницький 2023');
  Memo1.Lines.Add('');
end;
end;
end.
```