

**Проектування та моделювання програми для моніторингу мережі на
комп'ютері з операційною системою Windows**

Зміст

1. Проектування програми	3
1.1. Опис процесу діяльності	3
1.2. Високорівневий опис реалізації майбутнього програмного рішення згідно розроблених вимог	3
1.3. Архітектура програмного забезпечення	6
2. Моделювання та конструювання програмного забезпечення	9
2.1. Детальний опис класів програми та їх методів	9

1. Проектування програми

1.1. Опис процесу діяльності

Програму для моніторингу мережі на комп'ютері можна запустити на Windows 10 і вище за допомогою виконавчого файлу(.exe).

Після запуску програми відображається чотири вікна-вкладки розташовані по ліву частину вікна. На кожній вкладці відображена інформація і наявний функціонал відповідно до назви вкладки.

1.2. Високорівневий опис реалізації майбутнього програмного рішення згідно розроблених вимог

Програма буде реалізовуватися за допомогою мови програмування Java і матиме графічний інтерфейс написаний за допомогою бібліотеки JavaFX.

Кожна з нище наведених функцій програми матиме відповідну вкладку на графічному інтерфейсі користувача, де буде доступний її функціонал.

Список функцій програми і їх високорівневий опис:

1.2.1. Функція для моніторингу мережевих пристроїв;

При запуску програми на стартовому екрані користувачу відобразиться вміст даної функції, а саме виведеться інформація про мережеві інтерфейси присутні на пристрої, з якого був здійснений запуск програми, у вигляді таблиці з такими колонками:

- Ім'я
- IP-адреса (якщо присутня)
- MAC-адреса (якщо присутня)
- Тип (Фізичний, віртуальний інтерфейс)
- Up (чи запущений він зараз на комп'ютері)
- Loopback (чи є інтерфейсом Loopback – цей інтерфейс дозволяє комп'ютеру надсилати мережеві пакети до себе ж, що дає можливість перевіряти, чи працює мережеве з'єднання коректно.)

Дана інформація оновлюється в режимі реального часу з інтервалом 3 секунди. Назва вкладки на якій буде відображатися вміст таблиці “Мережеві пристрої”.

1.2.2. Функція для відслідковування трафіку та моніторингу мережевих інтерфейсів;

Інформація надана цією функцією відображається на вкладці “Трафік і статистика мережевих інтерфейсів”.

Вміст вкладки:

а. Графік, який повинен відображати інформацію в реальному часі про кількість переданих і отриманих байт по всіх мережевих інтерфейсах включно з локальними у вигляді ламаних ліній на проміжку часу.

б. Стан мережевих інтерфейсів - це статистика використання мережевих протоколів, яка відображає велику кількість параметрів у режимі реального часу(оновлення кожному секунду) доступних для перегляду і аналізу користувачем.

с. Статистика мережевих інтерфейсів. Містить інформацію про загальну кількість відправлених і отриманих байт в поточному з'єднанні включно з байтами переданими локальними інтерфейсами. На вибір користувачу буде доступно можливість відображення байт у кілобайтах, мегабайтах і гігабайтах. Інформація оновлюється кожні 3 секунди.

1.2.3. Функція для автоматичних сповіщень;

Функція необхідна для сповіщення користувача про певні події в мережі(втрата підключення до інтернету, досягнення обмежень трафіку).

Функцію можливо викликати з будь-якого місця програми.

У результаті виклику функції з параметром message(повідомлення, яке необхідно вивести користувачу) буде створене спливаюче вікно у нижній частині екрану справа. Можливе відображення кількох сповіщень одразу. Сповіщення зникне через 5 секунд і з'явиться знову через певний проміжок часу якщо проблема про яку йдеться в сповіщенні не вирішиться.

1.2.4. Функція моніторингу якості зв'язку;

Вміст даної функції відображається на вкладці “Якість з’єднання” у вигляді таблиці з такими стовпчиками:

- a. Адреса сервера;
- b. Пінг до сервера у мілісекундах;
- c. Якість з’єднання (відмінна, хороша, погана, недосяжна);

Список серверів вже заданий в функції. Якщо якийсь з серверів або всі з них отримають статус недосяжних(при втраті з’єднання з інтернетом або інших причинах) на екран користувачеві виведеться відповідне спливаюче сповіщення в якому буде вказана адреса сервера, який отримав такий статус.

1.2.5. Функція обмеження трафіку;

Можливість задання обмежень доступне на вкладці “Обмеження трафіку”.

Дана функція буде сповіщати користувача, коли заданий ліміт вичерпався або скоро вичерпається (90% і більше обмеження використано).

1.3. Архітектура програмного забезпечення

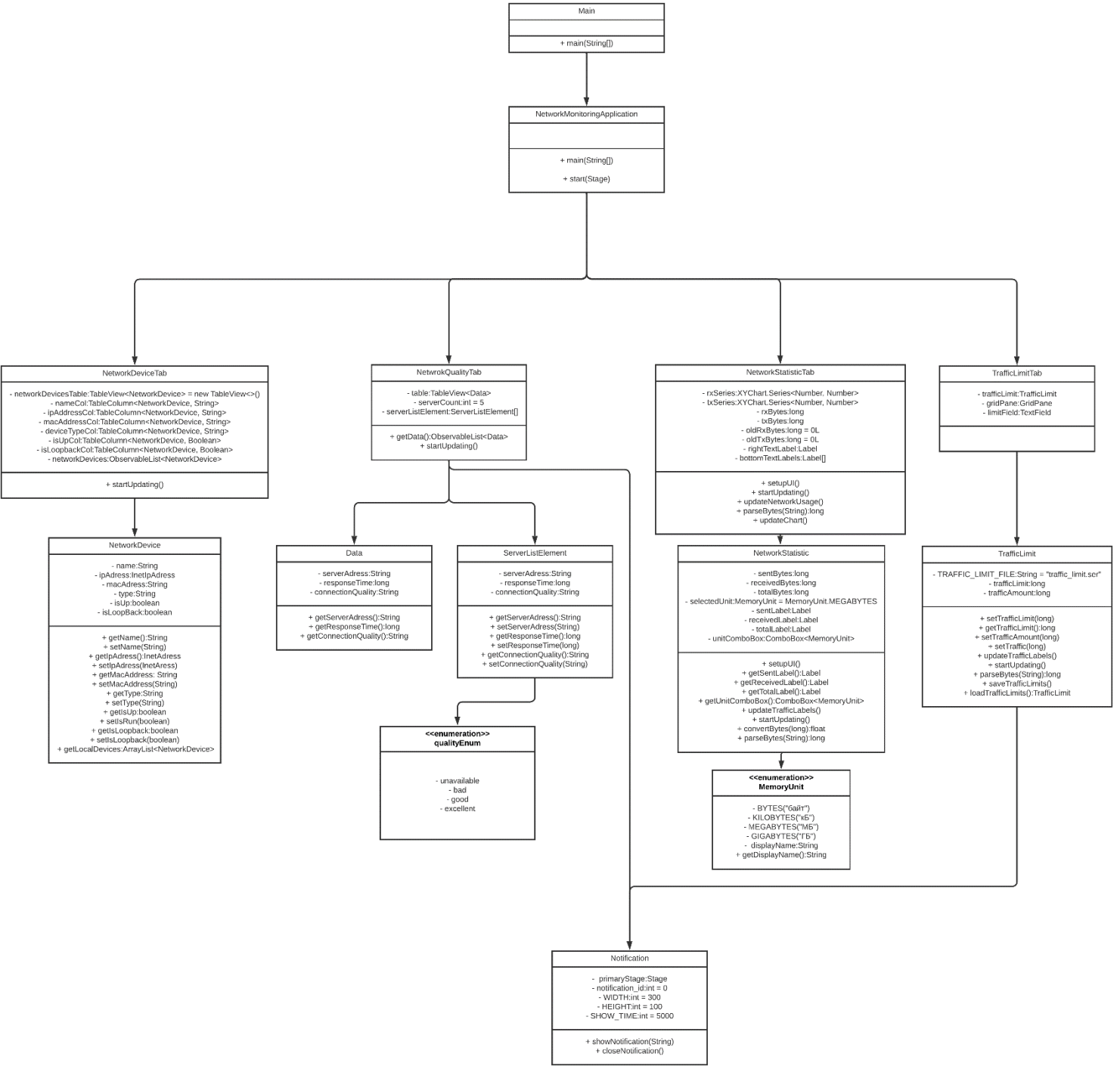


Рис. 1. Діаграма класів

Таблиця 1. Загальний опис класів та їх призначення

Клас	Опис та призначення класу
Main	Головний класу в якому розташовується метод <code>main()</code> – вхідна точка програми.
NetworkMonitoringApplication	Наслідує клас <code>javafx.application.Application</code> і перевизначає метод <code>start()</code> та містить метод <code>main()</code> – вхідну точку для програми з графічним інтерфейсом побудованим на основі JavaFX.
NetworkDeviceTab	Представляє собою вкладку для відображення таблиці мережевих пристроїв.
NetworkDevice	Представляє мережевий пристрій з його властивостями.
FloatingNotification	Клас <code>FloatingNotification</code> представляє собою спливаючі повідомлення на вікні JavaFX. Містить метод <code>showNotification(String)</code> , який можна використати в будь-якому класі програми для демонстрації користувачу вспливаючого сповіщення з текстом. Підтримує показ декількох сповіщень одночасно.
NetworkStatisticTab	Клас <code>NetworkStatisticTab</code> представляє вкладку з графіком використання мережі, який відображає використання мережі у вигляді RX та TX байтів на часовому проміжку та статистику мережі. Графік реалізований з використанням JavaFX.

NetworkStatistic	Клас NetworkStatistic представляє собою моніторинг статистики отриманих і надісланих байтів. Є можливість вибрати одиницю вимірювання в яких буде показуватись дана статистика: байти, кілобайти, мегабайти або гігабайти.
NetworkQualityTab	Клас NetworkQualityTab представляє собою вкладку для моніторингу якості з'єднання з серверами. Використовує клас ServerListElement як елемент таблиці TableView для демонстрації списку серверів і якості з'єднання з ними. Викликає метод showNotification(String) класу FloatingNotification, якщо з'єднання з певним сервером відсутнє.
ServerListElement	Елемент списку серверів, що використовується для моніторингу якості мережевого з'єднання.
TrafficLimitTab	Вкладка, яка містить форму для встановлення обмежень трафіку.
TrafficLimit	Клас, що представляє обмеження трафіку. Викликає метод showNotification(String) класу FloatingNotification, якщо обмеження скоро вичерпається або вже вичерпано.

2. Моделювання та конструювання програмного забезпечення

2.1. Детальний опис класів програми та їх методів

1. Клас Main

Головний клас, який містить точку входу програми – метод `public static void main(String[] args)`

Методи класу:

- **`public static void main(String[] args)`** – точка входу програми, запускає метод `public static void main(String[] args)`, який міститься в класі `NetworkMonitoringApplication`.

2. Клас `NetworkMonitoringApplication`

Клас наслідує `javafx.application.Application`. Основний функціонал класу полягає в побудові головного вікна програми. Містить точку входу для запуску графічного інтерфейсу.

Методи класу:

- **`public static void main(String[] args)`** – запускає графічний інтерфейс методом `launch()` наслідуваним з класу `javafx.application.Application`.
- **`public void start(Stage primaryStage)`** – перевизначений метод класу `javafx.application.Application`, який будує і налаштовує графічний інтерфейс програми.

3. Клас `NetworkDeviceTab`

Клас наслідує `javafx.scene.control.Tab`.

Представляє собою вкладку для відображення таблиці мережевих пристроїв.

Методи класу:

- **`private void startUpdating()`** - запускає процес оновлення таблиці мережевих пристроїв.

4. Клас `NetworkDevice`

Представляє собою мережевий пристрій з його властивостями.

Методи класу:

- **`public static ArrayList<NetworkDevice> getLocalDevices()`** - отримує список локальних мережевих пристроїв.

5. Клас FloatingNotification

Клас FloatingNotification представляє спливаючі повідомлення на вікні JavaFX.

Методи класу:

- **public void showNotification(String message)** - відображає спливаюче повідомлення з заданим текстом.
- **closeNotification()** - закриває вікно спливаючого сповіщення.

6. Клас NetworkQualityTab

Клас наслідує javafx.scene.control.Tab.

Клас NetworkQualityTab представляє собою вкладку для моніторингу якості з'єднання з серверами.

Методи класу:

- **private ObservableList<Data> getData()** - повертає дані для таблиці.
- **private void startUpdating()** - оновлює дані про якість з'єднання з серверами. Вносить оновлені дані в таблицю.

Містить внутрішній статичний клас **Data**, який описує дані, що будуть відображатися в таблиці.

7. Клас ServerListElement

Елемент списку серверів, що використовується для моніторингу якості мережевого з'єднання в клас NetworkQualityTab.

Містить enum(перерахування) **qualityEnum**, що визначає можливі значення якості з'єднання з сервером.

8. Клас NetworkStatisticTab

Клас наслідує `javafx.scene.control.Tab`.

Клас `NetworkStatisticTab` представляє собою графік використання мережі, який відображає використання мережі у вигляді RX та TX байтів на часовому проміжку та статистику мережі. Графік реалізований з використанням `JavaFX`.

Методи класу:

- **`private void setupUI()`** - налаштовує графічний інтерфейс користувача для відображення графіка використання мережі. Встановлює вісі X та Y, створює область графіка та встановлює заголовок. Ініціалізує серії даних для отриманих та надісланих байтів. Додає серії даних до графіка та встановлює його вміст. Створює текстові поля справа та знизу для відображення статистики мережевих інтерфейсів. Використовує клас `NetworkStatistic` для відображення елементу статистики використання трафіку.
- **`private void startUpdating()`** - починає оновлення даних мережевого використання та графіка на проміжках часу. Використовує `Timer` для запуску оновлення в фоновому потоці та `Platform.runLater()` для оновлення в головному потоці `JavaFX`. Оновлює дані мережевого використання та графік кожну секунду.
- **`private void updateNetworkUsage()`** - оновлює використання мережі. Метод виконує запит до командного рядка для отримання даних про використання мережі. Процес будується за допомогою `ProcessBuilder` з командою `"cmd.exe /c netstat -e -s"`. Читає вивід процесу через буферизований читач `BufferedReader`. Обробляє кожен рядок згідно заданого алгоритму для подальшого виведення користувачеві. Якщо знаходиться рядок з відповідною інформацією, розбиває його на частини і оновлює значення. Закриває читач і знищує процес. У разі виникнення виключення `IOException`, виводить стек виключення.
- **`private long parseBytes(String bytesStr)`** - перетворює рядок, що містить значення байтів, у відповідне довге ціле число.
- **`private void updateChart()`** - оновлює графік, додаючи нові дані про використання мережі. Кожне нове значення додається до відповідних серій даних (Rx та Tx) на графіку. Координати X відповідають розміру відповідних серій даних, а координати Y представляють кількість байт.

9. Клас NetworkStatistic

Клас NetworkStatistic представляє собою моніторинг статистики отриманих і надісланих байтів.

Методи класу:

- **private void setupUI()** - налаштовує користувацький інтерфейс для відображення інформації про трафік. Створює мітки для відображення інформації про надіслані, отримані та загальні байти, а також комбінований список для вибору одиниць вимірювання трафіку. Встановлює обробники подій для комбінованого списку для оновлення вибраної одиниці вимірювання та оновлює значення міток.
- **private void updateTrafficLabels()** - оновлює значень міток з інформацією про трафік. Метод виконує запит до командного рядка для отримання даних про статистику використання мережі. Процес будується за допомогою ProcessBuilder з командою "cmd.exe /c netstat -e". Читає вивід процесу через буферизований читач BufferedReader. Обробляє кожен рядок виводу і перевіряє, чи містить він слово "Bytes". Якщо знаходиться рядок з відповідною інформацією, розбиває його на частини і оновлює значення отриманих і надісланих байтів. Закриває читач і знищує процес. У разі виникнення виключення IOException, виводить стек виключення.
- **private void startUpdating()** - починає оновлення даних статистики мережевого використання. Використовує Timer для запуску оновлення в фоновому потоці та Platform.runLater() для оновлення в головному потоці JavaFX. Оновлює дані статистики мережевого використання кожні 3 секунди.
- **private float convertBytes(long bytes)** - конвертує байти в обрану одиницю вимірювання
- **private long parseBytes(String bytesStr)** - перетворює рядок, що містить значення байтів, у відповідне довге ціле число.

Клас містить enum(перерахування) MemoryUnit - enum одиниць вимірювання пам'яті.

10. Клас TrafficLimitTab

Клас наслідує `javafx.scene.control.Tab`.

Вкладка, яка містить форму для встановлення обмежень трафіку. Формує за допомогою конструктора вкладку на якій є поле для встановлення обмеження трафіку в мегабайтах та кнопка зберегти. Він використовує метод `TrafficLimit` для збереження величини обмеження.

11. Клас TrafficLimit

Імплементує інтерфейс `Serializable` і містить методи та поля необхідні для моніторингу і показу сповіщень про досягнення обмежень трафіку.

Методи класу:

- **public void setTraffic(long amount)** - встановлює значення трафіку і перевіряє, чи було досягнуто обмеження трафіку.
- **private void updateTrafficLabels()** - оновлює значення трафіку з використанням команди "netstat -e". Використовується для оновлення даних статистики мережевого використання.
- **public void startUpdating()** - починає оновлення даних статистики мережевого використання. Використовує `Timer` для запуску оновлення в фоновому потоці та `Platform.runLater()` для оновлення в головному потоці JavaFX. Оновлює дані статистики мережевого використання кожні 30 секунд.
- **private long parseBytes(String bytesStr)** - парсить рядок, який представляє значення в байтах, і повертає відповідне числове значення.
- **protected void saveTrafficLimits()** - зберігає обмеження трафіку в файл.
- **public static TrafficLimit loadTrafficLimits()** - завантажує обмеження трафіку з файлу.