

Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

**Їнь Фейюй**

**АНАЛІЗ ДОДАТКІВ ДЛЯ ВВІМКНЕННЯ ОБЧИСЛЮВАЛЬНИХ  
МАШИН**

**кваліфікаційна робота**

**здобувача вищої освіти другого (магістерського) рівня**

**освітньої програми «Комп'ютерні мережі»**

**за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис \_\_\_\_\_ **Їнь ФЕЙЮЙ**

Науковий керівник \_\_\_\_\_ **Геннадій МОГИЛЬНИЙ,**  
кандидат технічних наук, доцент  
кафедри інформаційних технологій  
та систем

Завідувач кафедри \_\_\_\_\_ **Микола СЕМЕНОВ,**  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

**Інь Фейюй**

**Тема:** Аналіз додатків для ввімкнення обчислювальних машин.

**Спеціальність:** 123 «Комп'ютерна інженерія».

**Установа:** ЛНУ імені Тараса Шевченка, 2025 р.

**Магістерська робота містить:** 91 с., 28 рис., 24 джерела, 5 додатків.

**Об'єктом дослідження** є технології дистанційного управління комп'ютерами.

**Предмет дослідження** – мережева технологія Wake-on-LAN для ввімкнення комп'ютерів.

**Мета роботи** – аналіз та впровадження технології Wake-on-LAN із розробкою програмного додатку, що забезпечує дистанційне ввімкнення комп'ютерів із підвищеним рівнем безпеки та автоматизації.

**Результати роботи.** Проведено порівняльний аналіз існуючих програм для реалізації Wake-on-LAN, досліджено особливості налаштувань BIOS, мережевих карт, а також методи інтеграції з Windows Task Scheduler і MikroTik. Було розроблено додаток із зручним інтерфейсом, який підтримує відправлення WoL-пакетів, автоматизацію завдань і захист від несанкціонованого ввімкнення за допомогою пароля BIOS. Додаток пройшов функціональне тестування, перевірку продуктивності та аналіз безпеки.

**Висновки.** У результаті розроблено інноваційний додаток, що спрощує керування комп'ютерами в навчальних лабораторіях та підвищує ефективність їх використання завдяки автоматизації та інтеграції з сучасними мережевими інструментами.

**Ключові слова:** WAKE-ON-LAN, WOL, LAN, ВІДДАЛЕНЕ ВВІМКНЕННЯ, БІОС, МАГІЧНИЙ ПАКЕТ, ПЛАНУВАЛЬНИК ЗАВДАНЬ, МІКРОТІК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЗАЦІЯ.

**Yin Feiyu**

**Topic:** Analysis of additional computer processing machines.

**Specialty:** 123 "Computer Engineering".

**Institution:** Luhansk Taras Shevchenko National University, 2025.

**Master's thesis includes:** 91 pages, 28 figures, 24 references, 5 appendices.

**Object of the study:** technologies for remote computer management.

**Subject of the study:** Wake-on-LAN network technology for powering on computers.

**Purpose of the study:** to analyze and implement the Wake-on-LAN technology by developing a software application that enables secure and automated remote power-on of computers.

**Results of the study.** A comparative analysis of existing Wake-on-LAN software was conducted, and the configuration features of BIOS, network adapters, and integration methods with Windows Task Scheduler and MikroTik were examined. An application with a user-friendly interface was developed, supporting WoL packet transmission, task automation, and protection against unauthorized power-on via BIOS password. The application underwent functional testing, performance evaluation, and security analysis.

**Conclusions.** As a result, an innovative application was developed that simplifies the management of computers in educational laboratories and enhances their efficiency through automation and integration with modern networking tools.

**Keywords:** WAKE-ON-LAN, WOL, LAN, REMOTE POWER-ON, BIOS, MAGIC PACKET, TASK SCHEDULER, MIKROTIK, SOFTWARE, AUTOMATION.

## ЗМІСТ

<b>ВСТУП</b>	<b>5</b>
<b>РОЗДІЛ І ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ</b>	<b>7</b>
1.1. Принципи роботи та налаштування технології WAKE-ON-LAN	7
1.2. Порівняльна характеристика програмних аналогів	11
Висновки до розділу	25
<b>РОЗДІЛ ІІ ОПИС ПРОЦЕСІВ СТВОРЕННЯ ТА ОНОВЛЕННЯ ПРОГРАМНОГО ДОДАТКУ</b>	<b>28</b>
2.1. Загальні вимоги до програмного додатку	28
2.3. Розробка та впровадження програмного додатку	31
2.4. Огляд інтерфейсу користувача комп'ютерного додатку	32
Висновок до розділу	35
<b>РОЗДІЛ ІІІ ДОСЛІДЖЕННЯ БЕЗПЕКИ ТА ШЛЯХИ ПІДВИЩЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМНОГО ДОДАТКУ</b>	<b>37</b>
3.1. Обмеження доступу до BIOS	37
3.2. Приклад налаштування обмеження доступу до BIOS	41
3.3. Використання пароля на BIOS у процесі відправки магнітного пакета	44
3.4. Інтеграція WoL-додатка з WINDOWS TASK SCHEDULER	48
3.5. Інтеграція WoL-додатка з MIKROTIK.	52
4.4. Тестування додатку	55
Висновки до розділу	56
<b>ЗАГАЛЬНІ ВИСНОВКИ</b>	<b>58</b>
<b>СПИСОК ЛІТЕРАТУРИ</b>	<b>60</b>
<b>ДОДАТОКИ</b>	<b>63</b>
ДОДАТОК А Лістинг програмного коду. Основна форма №1	63
ДОДАТОК Б Лістинг програмного коду. Основна форма №2	81
ДОДАТОК В Лістинг програмного коду. Основна форма №3	84
ДОДАТОК Г Лістинг програмного коду. Основна форма безпеки №4	88
ДОДАТОК Д Стислий опис файлів програмного додатку	91

## ВСТУП

В умовах відкритої збройної агресії російської федерації та постійних повітряних тривог велика кількість підприємств розпочала запроваджувати різноманітні методи віддаленої роботи співробітників. Безумовно, ці заходи розпочала значна кількість структурних підрозділів, шкіл, коледжів та університетів була змушена перейти на дистанційну роботу, дистанційне навчання або змішану форму організації освітнього процесу, що поєднує онлайн-заняття та, за можливістю, очні зустрічі. Часті повітряні тривоги, небезпека життю змушують навчально-допоміжний персонал, студентів та викладачів переміщуватися між аудиторіями та укриттями. Значний вплив мають постійні перебої зі світлом, які загалом призводять до не продуктивного навчання та безперервних перебоїв у стабільній роботі з комп'ютерною технікою. Повсякчасна необхідність вручну вмикати та вимикати обладнання є не лише трудомістким і незручним процесом, а й суттєво уповільнює навчання, а за відсутності навчально-допоміжного персоналу це стає практично неможливим.

Для вирішення цієї проблеми доцільно впровадити технологію Wake-on-LAN (WoL). Ця інноваційна технологія дозволяє дистанційно вмикати комп'ютери через мережу, що значно спрощує організацію роботи комп'ютерних лабораторій. З цієї точки зору, постає необхідність розробки інноваційних технологічних рішень, які б дозволили ефективно керувати ресурсами комп'ютерних лабораторій на відстані [1,2]. Для підвищення ефективності використання ресурсів комп'ютерних лабораторій необхідно розробити інноваційні рішення. Зокрема, створення спеціального додатку для дистанційного вмикання комп'ютерів дозволить значно спростити роботу адміністраторів, забезпечити безперебійний доступ студентів та викладачів до необхідного обладнання та покращить освітній процес.

Наукова новизна роботи полягає в інтеграції WoL із сучасними засобами автоматизації та безпеки, такими як «Планувальник задач» (Scheduler) та

маршрутизатори MikroTik. Це дозволяє вирішити низку завдань, пов'язаних із масштабованістю мереж, автоматизацією освітніх і технічних процесів та захистом від несанкціонованого доступу.

**Метою роботи** є аналіз підходів до створення програмного додатку для управління обчислювальними машинами в навчальних лабораторіях, що поєднує зручність та автоматизацію використання, збільшення функціональності, продуктивності та безпеки.

**Об'єктом дослідження** є технології мереж.

**Предмет дослідження** – мережева технологія ввімкнення комп'ютерів.

**Методи дослідження:** аналіз та впровадження технології Wake-on-LAN із розробкою програмного додатку, що забезпечує дистанційне ввімкнення комп'ютерів із підвищеним рівнем безпеки та автоматизації.

**Для досягнення поставленої мети** необхідно вирішити наступні завдання:

1. Дослідити технологію WoL, принципи її роботи та проаналізувати існуючі програмні рішення для реалізації технології Wake-on-LAN;
2. Розробити вимоги та впровадити розроблений програмний додаток із підтримкою віддаленого ввімкнення комп'ютерів, «планувальником задач» та ширшим функціоналом ніж у аналогів;
3. Проаналізувати особливості налаштування BIOS, обчислювальних машин при використанні технології WoL;
4. Проаналізувати шляхи підвищення функціональних можливостей роботи додатку та безпеки його застосування.

Теоретична значущість роботи полягає у розробці моделі інтеграції WoL із сучасними мережевими та системними інструментами, яка може бути використана для оптимізації адміністрування в освітніх та корпоративних середовищах. Практична значущість полягає у створенні програмного додатку, який може бути впроваджений у навчальних закладах та організаціях для підвищення ефективності управління комп'ютерними ресурсами.

# **РОЗДІЛ І**

## **ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ**

Ефективне управління комп'ютерними системами є важливим аспектом у сучасних навчальних та корпоративних інфраструктурах. Однією з ключових технологій, яка дозволяє дистанційно керувати обчислювальними машинами, є Wake-on-LAN (WoL). Ця технологія відкриває можливості для автоматизації процесів, зменшення енергоспоживання та підвищення ефективності адміністрування мереж.

У цьому розділі розглядаються принципи роботи WoL, методи її налаштування, а також проводиться аналіз існуючих програмних рішень для реалізації цієї технології. Це дослідження дозволяє визначити переваги та недоліки різних підходів, створюючи базу для подальшого розроблення власного програмного продукту.

### **1.1. Принципи роботи та налаштування технології Wake-on-LAN**

Технологія Wake-on-LAN (WoL) є одним із методів дистанційного управління комп'ютерами, який дозволяє виводити пристрій зі сплячого або вимкненого стану через локальну мережу (LAN). Це забезпечує зручність в управлінні обчислювальними ресурсами на відстані, що є надзвичайно корисним у навчальних комп'ютерних лабораторіях, де часом необхідно масово включати машини перед заняттями або для технічного обслуговування [13].

Основні принципи роботи Wake-on-LAN базуються на здатності мережевої карти реагувати на спеціальний "магічний пакет" (magic packet), коли комп'ютер перебуває в режимі низького енергоспоживання, такому як сплячий режим (S3) або режим глибокого сну (S4/S5) [16]. Також мережевий адаптер, підключений до джерела живлення навіть у сплячому стані, постійно

слухає мережу. Коли він отримує широкомовний пакет, що містить свою MAC-адресу, адаптер ініціює процес пробудження системи.

Налаштування WoL має декілька основних кроків, а саме:

### **1. Увімкнення WoL у BIOS/UEFI;**

Спершу необхідно перевірити та увімкнути підтримку WoL на апаратному рівні через BIOS або UEFI [15]. Ця функція не завжди увімкнена за замовчуванням і може бути названа по-різному (рис. 1.1) в різних моделях комп'ютерів:

- Wake-on-LAN (WoL)
- Power On By PCI-E
- Resume by PCI-E Device
- Resume by PME
- S4/S5 Wake on LAN
- ErP (енергозберігаючі режими)

Залежно від версії BIOS або UEFI, для доступу до цієї опції необхідно увійти до налаштувань, зазвичай натискаючи клавішу `Del`, `F2` або `Esc` під час завантаження комп'ютера [3].

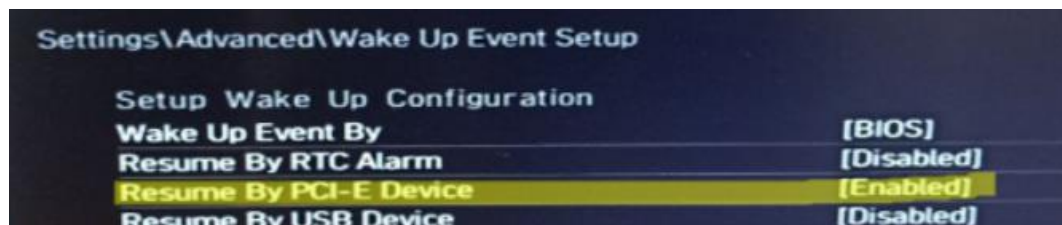


Рис. 1.1. Приклад назви функції

### **2. Налаштування мережевого адаптера у Windows;**

Після увімкнення WoL на рівні BIOS/UEFI потрібно переконатися, що мережева карта підтримує цю функцію, і налаштувати її у операційній системі Windows:

- Крок 1: Відкрити Диспетчер пристроїв, натиснувши `Win + X` та обравши «Device Manager» (або виконавши команду `devmgmt.msc`).



- Крок 2: Розгорнути секцію «Network Adapters» та відкрити властивості фізичного мережевого адаптера [17].
- Крок 3: На вкладці «Advanced» знайти параметр, пов'язаний із Wake-on-LAN. Зазвичай він називається «Wake on magic packet», але в мережевих адаптерах Intel може називатися «PME (Power Management Event)» (рис 1.2).

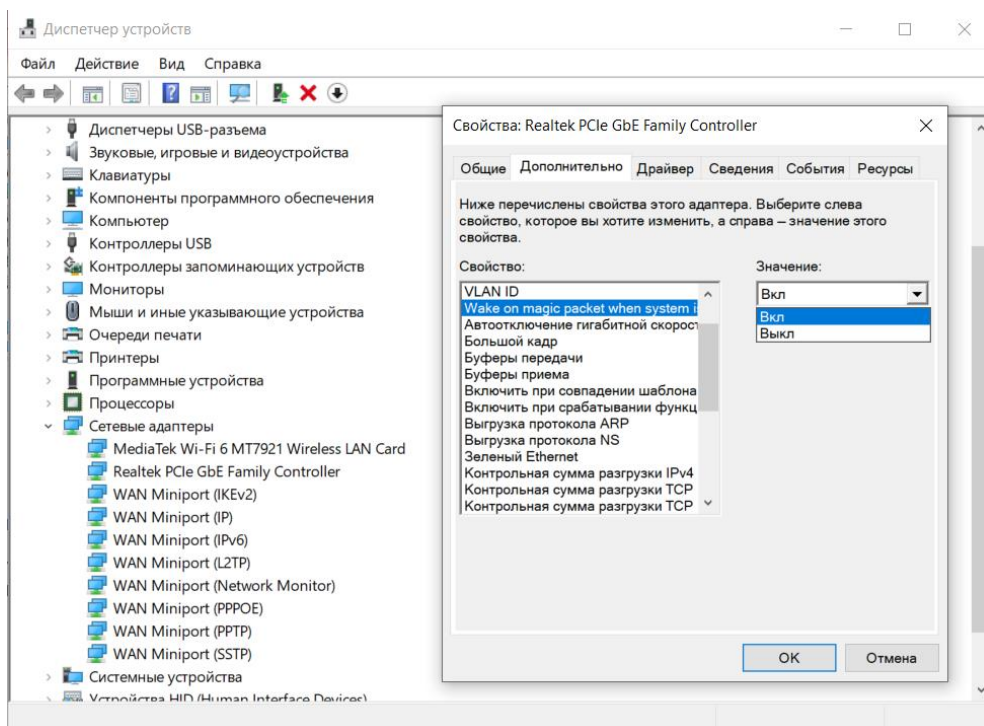


Рис. 1.2. Налаштування мережевого адаптера

- Крок 4: На вкладці «Power Management» активувати опції «Allow this device to wake the computer» і «Only allow a magic packet to wake the computer» [19]. Це гарантує, що пристрій виходитиме зі сплячого режиму лише при отриманні "магічного пакета" (рис 1.3).

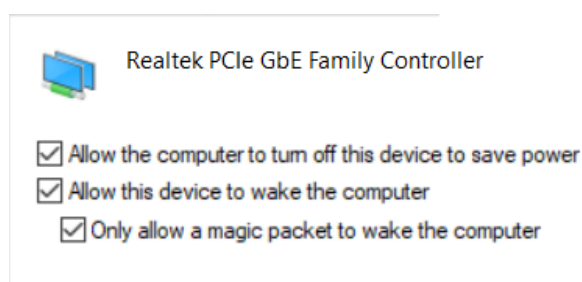


Рис. 1.3. Активація опцій

Щоб почати використовувати Wake-on-LAN та успішно запустити комп'ютер, необхідно знати MAC-адресу комп'ютера та IP-адресу (не обов'язково, якщо ПК знаходиться в межах однієї мережі, а саме сегменті L2), який потрібно ввімкнути [18]. Ці дані можна знайти у властивостях мережевого адаптера або за допомогою командного рядка, натиснувши комбінацію клавіш «Win + R» та ввівши «cmd» (рис. 1.4), ввівши команду «ipconfig /all» (рис. 1.5).

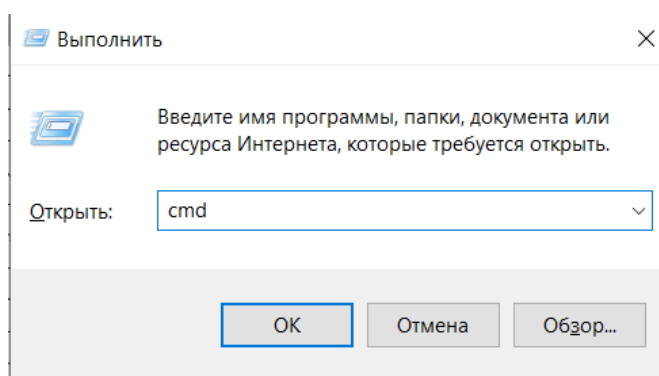


Рис. 1.4. Відкриття командної строки

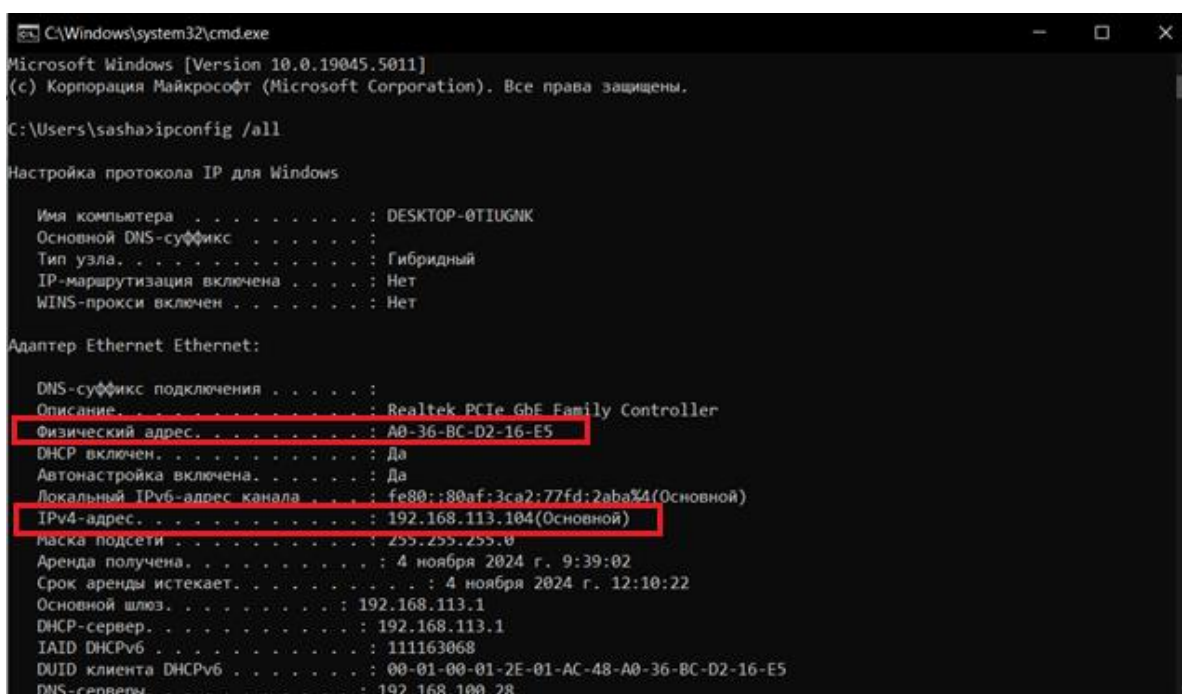


Рис. 1.5. Знаходження MAC та IP адреси

Наступні дії, які слід виконати, щоб відправити "магічний пакет":

1. Використовувати спеціалізоване програмне забезпечення, таке як: «NirSoft WakeMeOnLan» або «WakeOnLanGui», «Depicus Wake on LAN» або «SolarWinds WoL» "HM Software NL WoL", "WinBox від MikroTik"
2. Використовувати мережеві маршрутизатори або комутатори, що підтримують відправку WoL пакетів.
3. Використовувати оновлену програму для відправки магічних пакетів або налаштувати автоматизовану систему (наприклад, за допомогою MikroTik), що надсилатиме WoL пакети згідно з розкладом або запитом адміністратора.

Отже, Wake-on-LAN є потужним інструментом для ІТ-відділів, особливо у великих інфраструктурах на кшталт комп'ютерних лабораторій та аудиторій, де регулярно віддалене ввімкнення комп'ютерів дозволяє заощаджувати час і енергоресурси, роблячи процес навчання більш ефективним.

## **1.2. Порівняльна характеристика програмних аналогів**

Сучасні рішення для реалізації технології Wake-on-LAN (WoL) включають широкий спектр програмних інструментів, кожен із яких пропонує певні можливості для керування обчислювальними ресурсами в локальній мережі [5]. Завдяки різноманіттю програмного забезпечення адміністратори можуть обирати саме ті інструменти, що найкраще відповідають їхнім потребам. Деякі з них орієнтовані на простоту використання та базові функції, інші ж — на гнучке налаштування та розширені можливості автоматизації. У цьому підрозділі розглянемо основні програмні засоби для надсилання WoL-пакетів: NirSoft WakeMeOnLan, WakeOnLanGui, SolarWinds WoL, HM Software NL WoL, та WinBox від MikroTik. Для кожного з них буде проаналізовано такі аспекти, як інтерфейс, функціональність, підтримка автоматизації, а також особливості налаштування та використання.

### ***Огляд програми «WakeMeOnLan» від NirSoft.***

**NirSoft WakeMeOnLan** — це легка та безкоштовна утиліта, розроблена компанією NirSoft, що дозволяє віддалено пробуджувати комп'ютери в локальній мережі за допомогою Wake-on-LAN [12]. Основні особливості NirSoft WakeMeOnLan:

- **Інтерфейс.** Програма має простий та інтуїтивно зрозумілий інтерфейс, який не потребує складного налаштування (рис. 1.6).

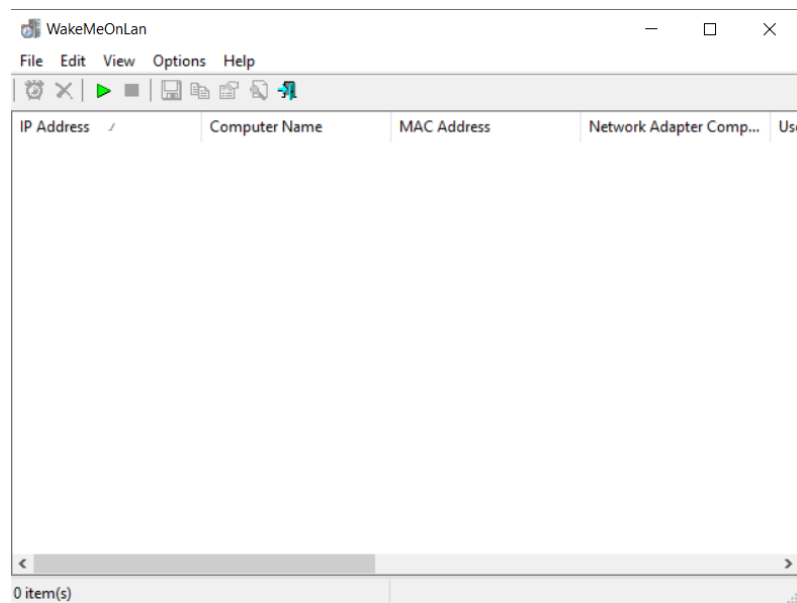


Рис. 1.6. Початкове вікно програми

Вікно програми почне відображати список пристроїв у мережі з можливістю сортування та фільтрації після натискання на зелений трикутник, але пристрої відображаються лише в межах роутера (рис. 1.7).

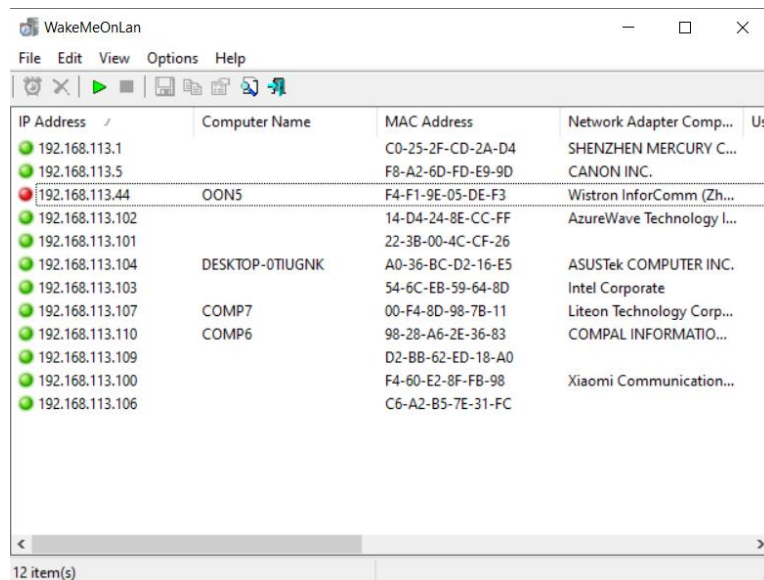


Рис. 1.7. Пошук доступних адрес

- **Функціональність.** WakeMeOnLan сканує мережу, автоматично виявляючи пристрої, які підтримують Wake-on-LAN, і додає їх у список. Користувач може пробуджувати обрані пристрої, обравши потрібну строку(-ки) та натиснувши на іконку годинника і підтвердивши дію (рис 1.8).

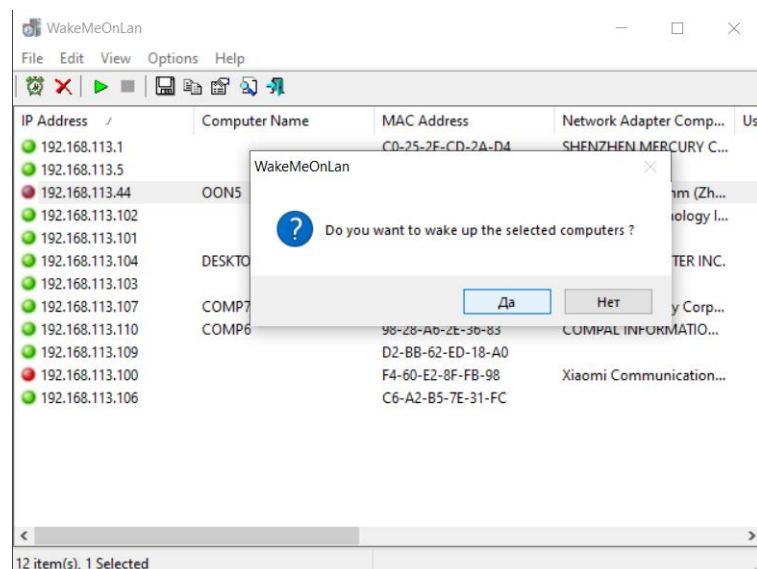


Рис. 1.8. Підтвердження відправки сигналу

- **Підтримка автоматизації.** Програма дозволяє зберігати список пристроїв у файлі та використовувати його для майбутніх запусків (рис.

1.9), що робить її зручною для періодичних завдань, але без вбудованих засобів автоматизації на рівні розкладу [12]. Всі файли програми можуть зберігатися в архіві або в папці на комп'ютері.

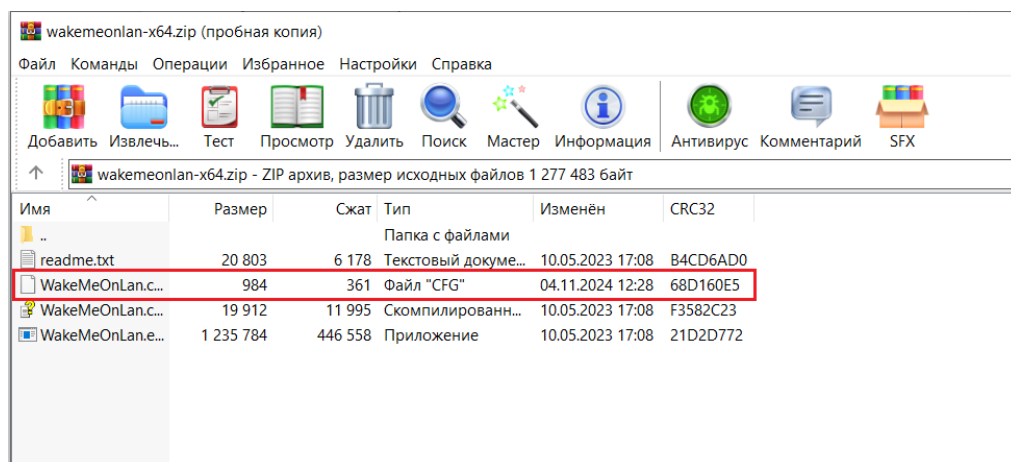


Рис. 1.9. Збереження даних «WakeMeOnLan»

- **Налаштування та використання.** Налаштування NirSoft WakeMeOnLan є мінімальним, оскільки програма створена для швидкого й ефективного запуску WoL. Вона не має додаткових можливостей для розширеної конфігурації мережевих параметрів.

Недоліки утиліти: неможливість посилення сигналу в інші мережі, а саме вузьке направлення дії тільки в межі сегменту L2. Також немає можливості планування ввімкнення за розкладом, а саме планувальника задач (англ. scheduler) [12].

### *Огляд програми «WakeOnLanGui» від Depicus.*

**WakeOnLanGui** — це базовий програмний додаток від Depicus з простим графічним інтерфейсом для виконання Wake-on-LAN. Даний інструмент є популярним серед користувачів, які шукають простий спосіб тільки надіслати магічний пакет [6].

- **Інтерфейс.** Графічний інтерфейс є простим і дозволяє швидко вводити MAC-адреси та IP-адреси пристроїв для відправлення WoL-пакетів (рис. 1.10).

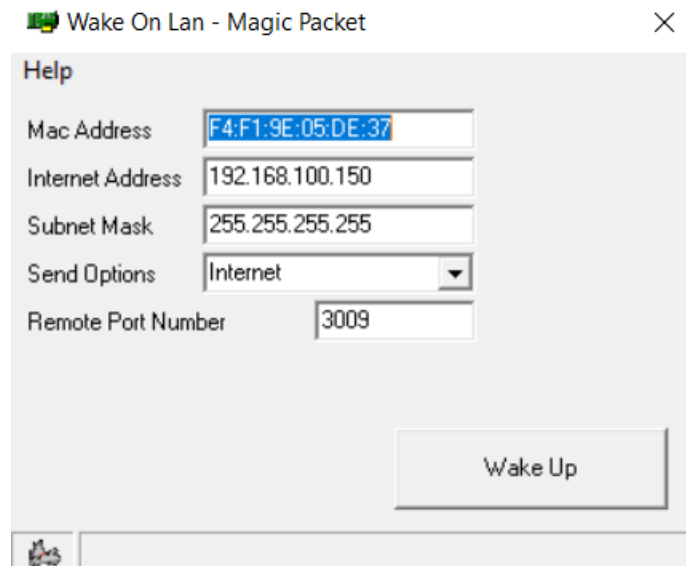


Рис. 1.10. Графічний інтерфейс

- **Функціональність.** Основна функція — пробудження пристроїв за MAC-адресою та IP-адресою; інші опції або додаткові функції не передбачені, що обмежує програму базовим використанням.
- **Підтримка автоматизації.** WakeOnLanGui не надає можливостей для автоматизованого пробудження комп'ютерів за розкладом. Його функціонал обмежується прямим ручним використанням. Також відсутня база даних про комп'ютери, що ускладнює швидке використання та досягнення ввімкнення групи обчислювальних машин.
- **Налаштування та використання.** Додаток простий у налаштуванні та використанні. Користувач має ввести лише MAC та IP адреси цільового комп'ютера, щоб відправити WoL-пакет [6].

Після відправки пакету у рядку з'являється повідомлення про те, що пакет був надісланий (рис. 1.11). Також було проведено тестування відправки пакета за межі роутера — результати невдалі, адже комп'ютер, знаходившись в іншому сегменті L2 не отримав сигнал.

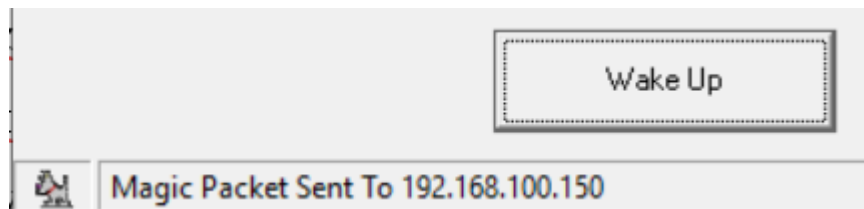


Рис. 1.11. Повідомлення про відправку

Недоліки програмного додатка зв'язані з тим, що він дуже вузько направлений і, якщо порівнювати його з іншими програмними рішеннями, то можна виокремити відсутність планувальника задач, неможливість відправки сигналу в іншу мережу, немає можливості створити список комп'ютерів, для швидкого ввімкнення одночасно.

### *Огляд програми «SolarWinds WoL»*

**SolarWinds WoL** — це на вигляд простий додаток, але з великим набором інструментів, що дозволяє надсилати WoL-пакети (рис. 1.13) [8].

Встановлення програми відбувається просто та швидко (рис. 1.12), але треба бути уважним, адже установник пропонує встановити сторонні програми. Вже після завантаження потрібно зайти в папку компонентів, куди відбулось встановлення, та відкрити «WakeOnLAN.exe» (рис. 1.13).

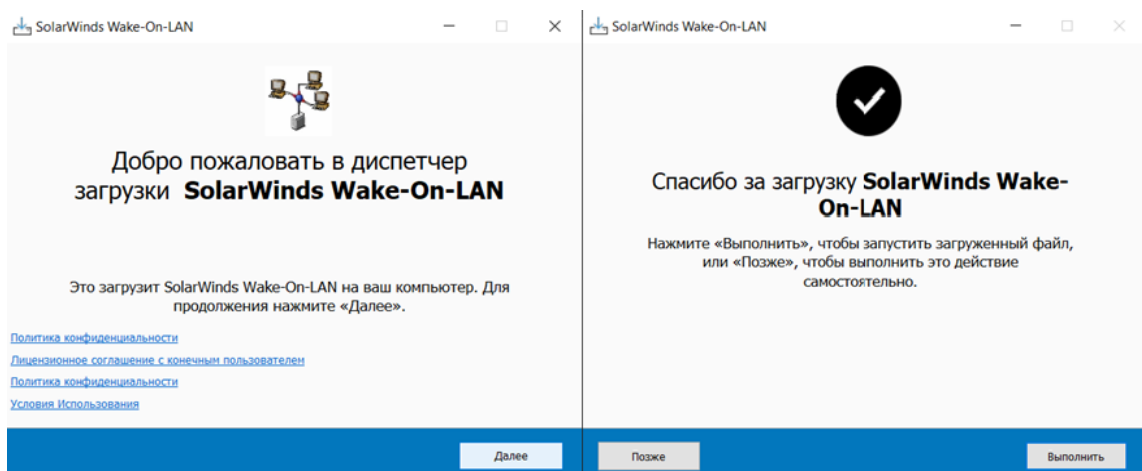


Рис. 1.12. Встановлення програми



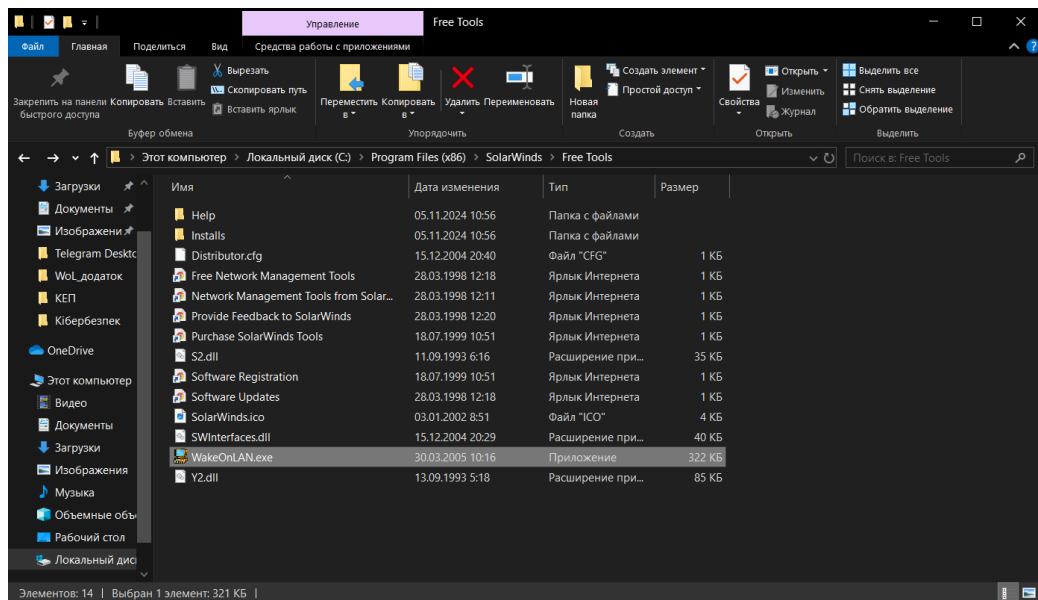


Рис. 1.13. Папка з компонентами

- **Интерфейс.** SolarWinds надає користувачеві дуже простий та небагатофункціональний інтерфейс, орієнтований на звичайних користувачів (рис. 1.14).

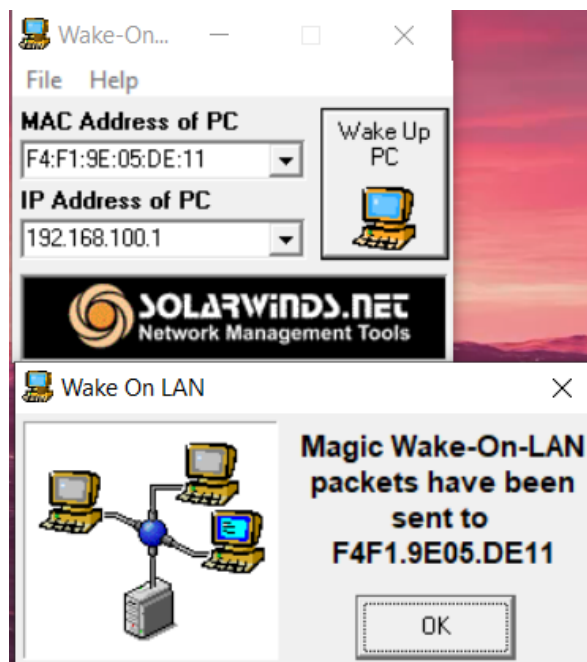


Рис. 1.14. Інтерфейс додатка

- **Функціональність.** Програма немає розширені функції моніторингу мережі, не дозволяє відстежувати стан комп'ютерів та не може

комплексно керувати приладами. Вона має тільки деякі звичайні налаштування (рис. 1.15) та функцію автоматичного знаходження MAC-адреса ПК де встановлений додаток [8].

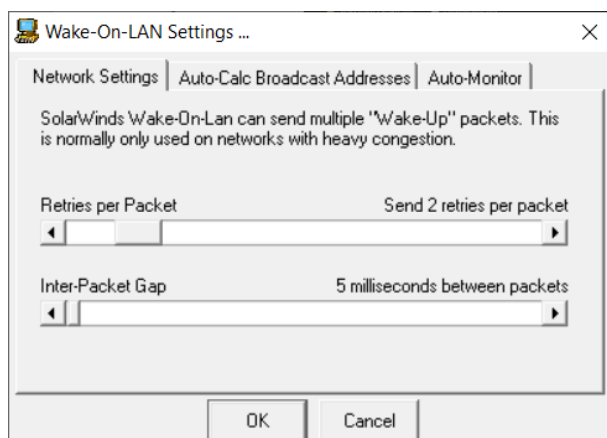


Рис. 1.15. Налаштування

- **Підтримка автоматизації.** SolarWinds не підтримує автоматизацію самостійно, а лише тільки завдяки інтеграції з іншими компонентами, і це дозволяє налаштовувати розклад і використовувати сценарії для автоматичного запуску WoL, але функції не були знайдені у вільному доступі.
- **Налаштування та використання.** Налаштування є більш складним і потребує певних навичок у мережевому адмініструванні, що може бути менш зручним для звичайних користувачів [8].

Щодо недоліків програми так це те, що для повного користування всіма можливостями і всім функціоналом потрібно встановлювати додаткові компоненти, яких немає у відкритому доступі на момент написання даної роботи. Версія без додатків дуже вузько направлена та виконує тільки основну функцію – посилення магічних пакетів і, до речі, це відбувається тільки в межах однієї мережі (сегменті L2).

### *Огляд програми «HM Software NL WoL»*

**HM Software NL WoL** — легка програма для Windows, орієнтована на базову WoL-функціональність. Завантажити можна в Microsoft Store безкоштовно (рис. 1.16) [7].

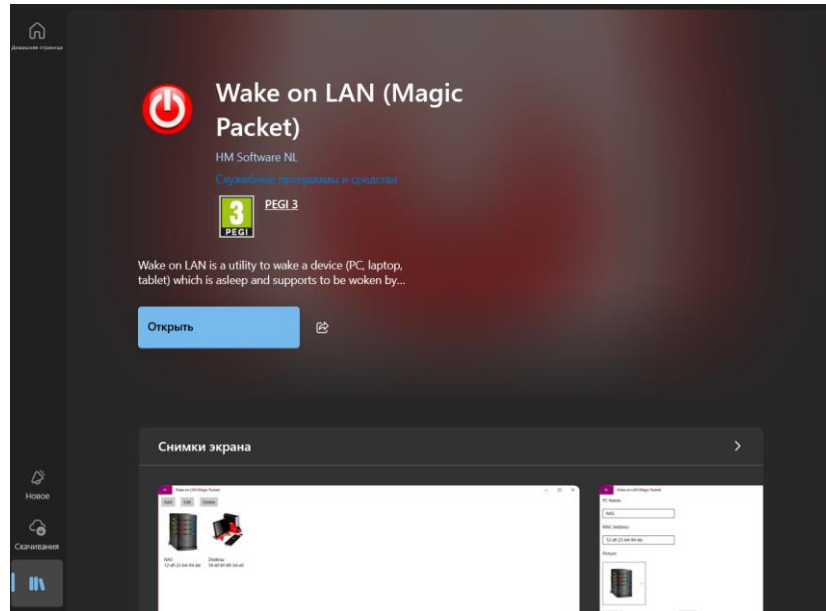


Рис. 1.16. Вигляд «HM Software NL WoL» в Microsoft Store

- **Інтерфейс.** Має мінімалістичний інтерфейс, що дозволяє швидко відправити WoL-пакет без додаткових налаштувань. Натиснувши ПКМ по доданому ПК, обираємо «Send WOL» і сигнал успішно відправляється (рис. 1.17).

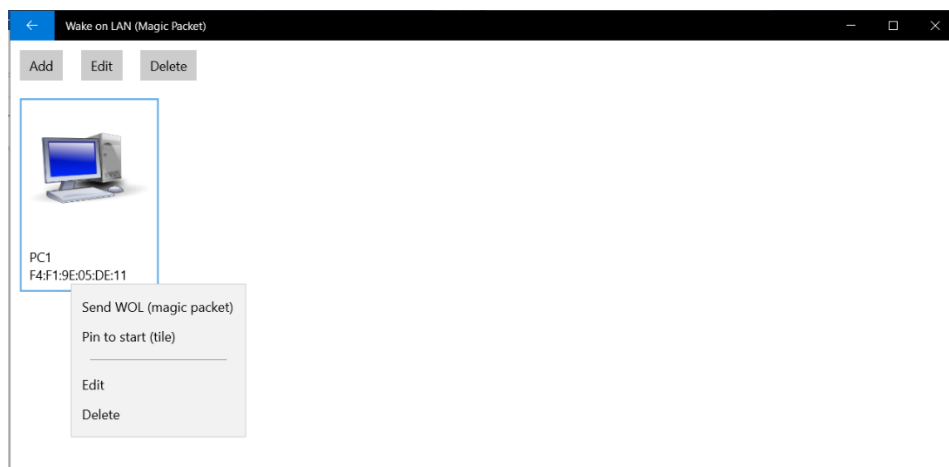


Рис. 1.17. Інтерфейс та відправка сигналу

- **Функціональність.** Програма здатна надсилати WoL-пакети всередині локальної мережі та підтримує збереження налаштувань пристроїв для подальшого використання [7].
- **Підтримка автоматизації.** NM Software NL WoL не має вбудованої підтримки автоматизації або інтеграції з іншими інструментами. Треба обирати кожен ПК і відправляти окремо кожному.
- **Налаштування та використання.** Використання максимально просте, програма підходить для тих, кому потрібен швидкий доступ до основних функцій WoL без налаштувань. Додати дані про комп'ютери просто, потрібен лише тільки MAC-адреса та дати назву (рис. 1.18) [7].

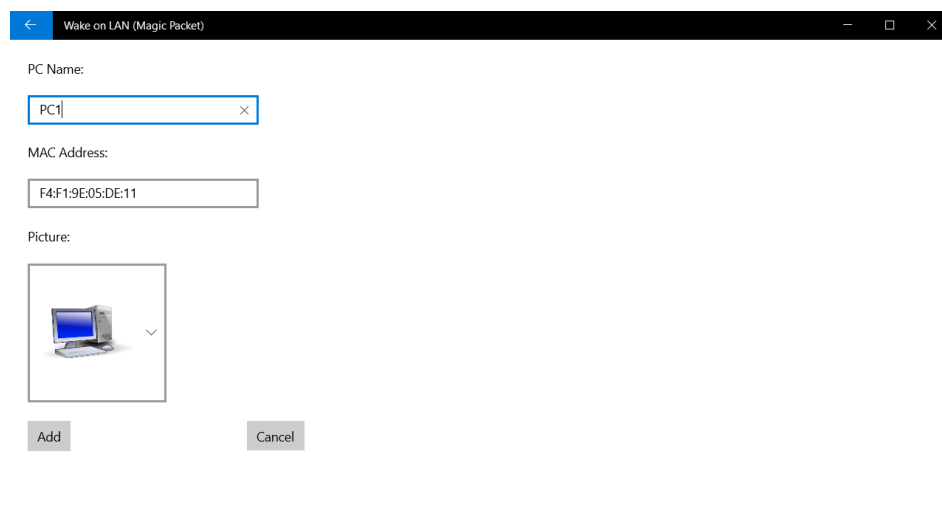


Рис. 1.18. Налаштування програми

З недоліків програми можна виділити застарілий інтерфейс, адже «NM Software NL WoL» має спрощений інтерфейс, який виглядає застарілим і менш інтуїтивним, що може впливати на зручність використання для сучасних користувачів. Також програма орієнтована тільки на просте надсилення WoL-пакетів і не підтримує розширені мережеві функції, такі як пробудження через Інтернет, що може ускладнити використання в складних мережах або великих мережевих інфраструктурах та відсутність планування ввімкнення в запланований час. «NM Software NL WoL» не отримує регулярних оновлень і,

схоже, не має активної підтримки, що може призводити до проблем із сумісністю на нових версіях Windows і підвищує ризик безпекових уразливостей.

### ***Огляд програми «Easy WOL (Wake on LAN)»***

**Easy WOL** — це зручний додаток для Windows, що дозволяє швидко та ефективно пробуджувати комп'ютери в локальній мережі за допомогою Wake-on-LAN. Завантажити додаток можна в Microsoft Store безкоштовно (рис. 1.19) [10].

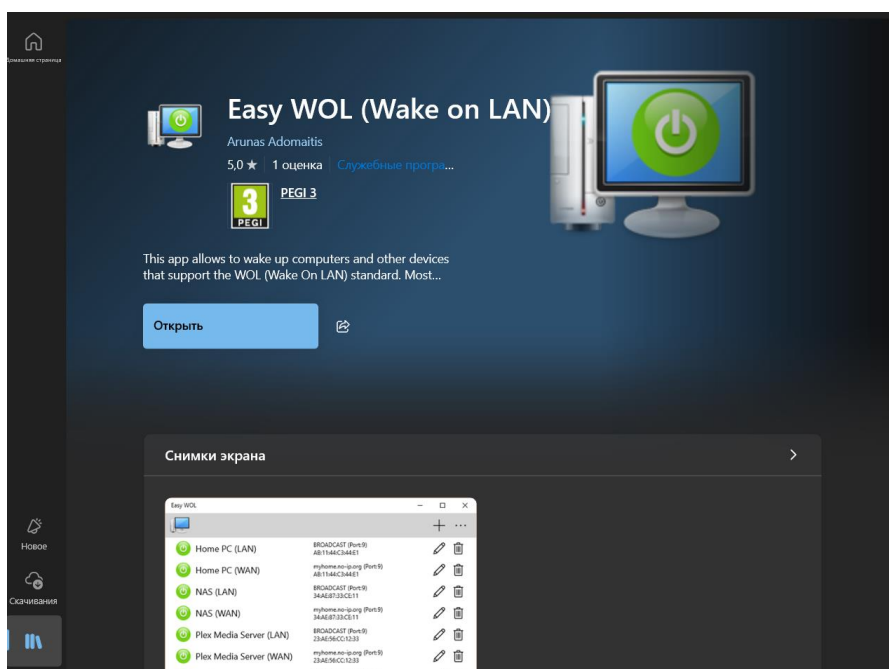


Рис. 1.19. Вигляд «Easy WOL» в Microsoft Store.

- **Інтерфейс.** «Easy WOL» має простий та зрозумілий інтерфейс, оптимізований для початківців та тих, хто шукає швидкий спосіб запуску WoL-пакетів (рис. 1.20). Інтерфейс дозволяє додавати пристрої за їх MAC-адресами, а також зберігати їх для подальшого використання (рис. 1.21).



Рис. 1.20. Початковий інтерфейс

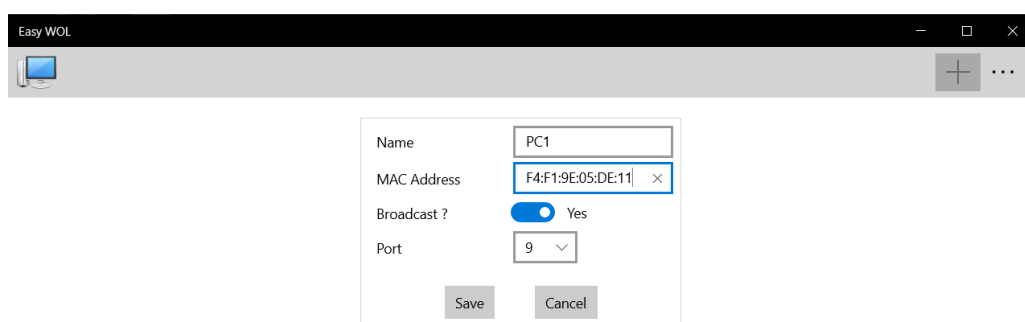


Рис. 1.21. Спроба додавання даних

- **Функціональність.** Програма забезпечує базову можливість пробудження пристроїв у локальній мережі, але не підтримує розширені налаштування для відправки WoL-пакетів через Інтернет або з додатковими параметрами (порт або IP-адреса) [10].
- **Підтримка автоматизації.** «Easy WOL» не має вбудованих функцій для автоматизованого пробудження пристроїв за розкладом. Користувачі можуть вручну пробуджувати пристрої, збережені в списку, але автоматизація або інтеграція з іншими сервісами не підтримуються.
- **Налаштування та використання.** Додаток не потребує складних налаштувань, що робить його зручним для користувачів будь-якого рівня. Налаштування обмежуються введенням MAC-адрес, ідеально підходить для тих, кому потрібен швидкий доступ до основних функцій

WoL без додаткових конфігурацій [10]. Щоб відправити магічний пакет треба натиснути на зелений значок навпроти доданого ПК (рис. 1.22).

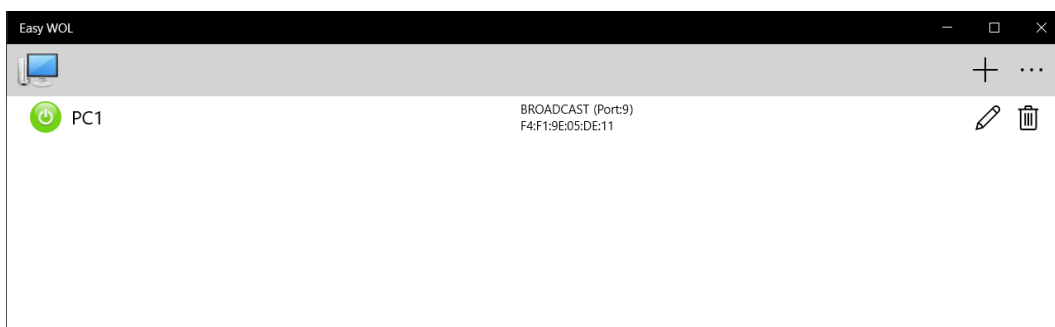


Рис. 1.22. Відправка сигналу

З недоліків можна виокремити обмежений функціонал «Easy WOL» надає лише базові можливості для пробудження комп'ютерів в межах локальної мережі. Вона не підтримує складніші налаштування, такі як вибір альтернативного порту чи IP-адреси для надсилання WoL-пакетів через Інтернет. Також додаток не має функцій автоматичного пробудження за розкладом, що обмежує його використання в великих інфраструктурах, де бажано мати можливість планувати увімкнення комп'ютерів заздалегідь. А також вузька спеціалізація додатку, а саме створення його виключно для відправки сигналу WoL, без інтеграції додаткових функцій для мережевого моніторингу чи керування пристроями. Щодо незручності використання, було досліджено, що після натискання на кнопку і відправки сигналу, ніякого повідомлення не відображається і неможливо дізнатися про успішність операції.

### ***Огляд програми «WinBox від MikroTik»***

**WinBox від MikroTik** — це багатофункціональний професійний інструмент, що використовується для управління пристроями MikroTik, зокрема, з можливістю надсилання WoL-пакетів [9].

- **Інтерфейс.** Інтерфейс WinBox орієнтований на професіонального адміністратора мережі та включає всі налаштування MikroTik в одному

вікні, що може бути складним для новачків, але ефективним для досвідчених користувачів (рис 1.23).

- **Функціональність.** Окрім WoL, WinBox дозволяє повністю налаштувати маршрутизатори MikroTik, включаючи маршрутизацію, безпеку, моніторинг мережі та незлічену низку інших корисних функцій.
- **Підтримка автоматизації.** Підтримує сценарії та автоматизацію, що робить його потужним інструментом для адміністрування великих мереж.
- **Налаштування та використання.** Використання потребує знань про маршрутизатори MikroTik, що може бути викликом для користувачів без досвіду.

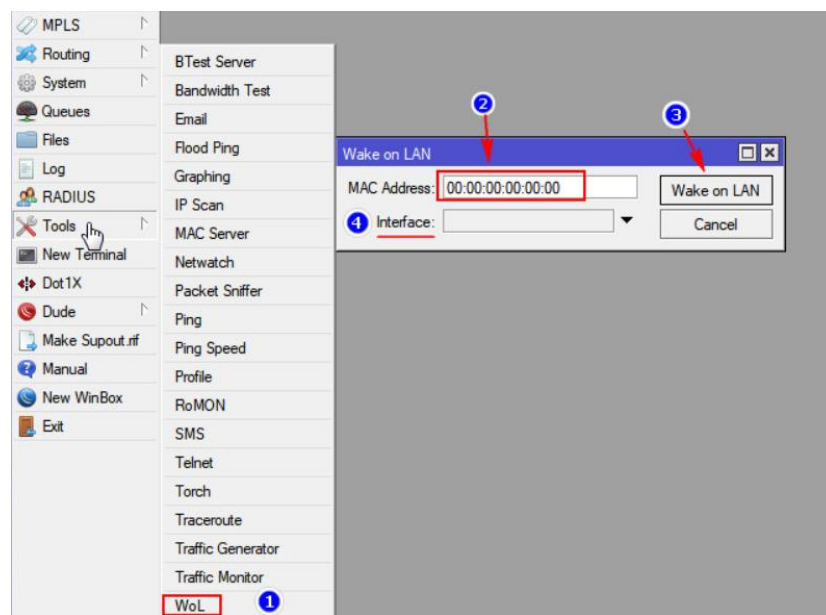


Рис. 1.23. Інтерфейс WinBox

Можна виокремити деякі недоліки, такі як:

- Високий поріг входу. WinBox розроблений для професіоналів та просунутих користувачів, що може ускладнити роботу для тих, хто не знайомий із системами MikroTik. Інтерфейс, хоча й потужний, може бути незручним для новачків.



- Надмірність для простих задач. Для базових потреб WoL WinBox може виявитися надто складним. Більшість функцій цього ПЗ пов'язані з налаштуванням маршрутизації, VPN, фільтрації трафіку, що не завжди виправдано для простої відправки WoL-пакетів [9].
- Обмеження на доступність лише для обладнання MikroTik. Для максимальної функціональності WinBox потрібні маршрутизатори та точки доступу MikroTik, що знижує універсальність і обмежує використання в мережах з іншим обладнанням.

Отже, у ході аналізу програмних засобів для реалізації WoL було розглянуто різні рішення, кожне з яких має свої особливості, переваги та обмеження. «NirSoft WakeMeOnLan» та «WakeOnLanGui» виділяються простим інтерфейсом, але мають обмеження в автоматизації та роботі поза локальною мережею. «SolarWinds WoL» пропонує розширену функціональність, проте для повного використання можливостей потребує додаткових компонентів. «HM Software NL WoL» та «Easy WOL» орієнтовані на базові потреби, що обмежує їхню придатність для великих інфраструктур. WinBox від MikroTik, хоча й надає потужні можливості для адміністраторів, є складним для користувачів-початківців. Таким чином, вибір конкретного програмного рішення має базуватись на обсягах інфраструктури, потребах автоматизації та рівні підготовки персоналу.

### **Висновки до розділу**

Розділ був присвячений аналізу технології Wake-on-LAN (WoL) та існуючих програмних додатків для її реалізації. Це дослідження дозволило з'ясувати, що WoL – важливий інструмент для дистанційного управління комп'ютерами та у великих навчальних або корпоративних інфраструктурах. За допомогою цієї технології можна віддалено контролювати увімкнення комп'ютерів у мережах, що особливо корисно для навчальних комп'ютерних

лабораторій, де потрібно швидко та зручно керувати великим парком обчислювальних машин.

У підрозділі 1.1 детально описано принципи роботи WoL та етапи налаштування, починаючи від увімкнення підтримки WoL у BIOS/UEFI та завершуючи активацією функції в операційній системі Windows. Це забезпечує базову підготовку для коректної роботи технології в локальній мережі. Визначено також ключову роль налаштувань мережевого адаптера для отримання та обробки "магічного пакета", який і ініціює процес пробудження. Правильне налаштування кожного з етапів є обов'язковою умовою для коректної реалізації технології, що робить WoL ефективним і стабільним інструментом управління комп'ютерними ресурсами на відстані.

У підрозділі 1.2 було проведено порівняльний аналіз програмних засобів для реалізації WoL. Було розглянуто шість основних програм: «NirSoft WakeMeOnLan», «WakeOnLanGui», «Depicus Wake on LAN», «SolarWinds WoL», «NM Software NL WoL» та «WinBox від MikroTik». Кожен із цих інструментів має свої унікальні особливості та сфери застосування. Деякі з них, наприклад, «NirSoft WakeMeOnLan» і «WakeOnLanGui», пропонують прості рішення для управління комп'ютерами у межах локальної мережі, тоді як «WinBox від MikroTik» надає більш комплексні можливості налаштування, що робить його підходящим для адміністраторів мереж MikroTik, але складним для загального використання. «Depicus Wake on LAN» підтримує пробудження через Інтернет, що розширює можливості WoL, тоді як «SolarWinds WoL» пропонує гнучку інтеграцію з іншими інструментами для побудови масштабної інфраструктури управління.

Результати порівняння виявили низку важливих особливостей програмного забезпечення для WoL:

- **Автоматизація.** Далеко не всі програми підтримують автоматизацію на рівні розкладу, що є важливим для ефективного адміністрування великих мереж.
- **Гнучкість налаштувань.** Програми відрізняються за можливостями налаштування мережевих параметрів і підтримкою віддалених мереж, що може впливати на зручність їх застосування в конкретних умовах.
- **Зручність та доступність.** Простота інтерфейсу та доступність додаткових функцій також є визначальними факторами, особливо для

початківців або тих, хто потребує базових функцій без глибоких налаштувань.

Таким чином, аналіз WoL та існуючих програмних засобів показав, що для ефективного вибору рішення необхідно враховувати специфіку мережевої інфраструктури та потреби користувачів – інженерів комп'ютерних лабораторій. У випадку великих лабораторій доцільним буде вибір комплексних інструментів з можливостями автоматизації та гнучкого налаштування. Водночас для невеликих мереж або локальних завдань цілком підходять простіші програмні додатки, які не потребують складних конфігурацій.

## РОЗДІЛ II

### ОПИС ПРОЦЕСІВ СТВОРЕННЯ ТА ОНОВЛЕННЯ ПРОГРАМНОГО ДОДАТКУ

#### 2.1. Загальні вимоги до програмного додатку

Розробка програмного додатку для ввімкнення обчислювальних машин у навчальних комп'ютерних лабораторіях базується на чітко визначеному технічному завданні [13]. Основною метою є створення зручного та ефективного інструменту, здатного працювати в умовах мінімальних технічних вимог. **Шифр ПР:** WOL-EZ-13. **Підставою для виконання** даної розробки є завдання на дипломний проект. **Терміни розробки:** початок - 20 січня 2024р., закінчення - 15 жовтня 2024р. **Фінансується** за рахунок коштів замовника. Програма призначена для використання в навчальному процесі в НН ІМІТ ЛНУ ім. Т. Шевченка.

Основними мінімальними технічними характеристиками є підтримка операційних систем Windows 10/11, мінімальна вимога до апаратних ресурсів: Intel Core i3, 4 ГБ ОЗУ, HDD не менше 64 Гбайт; підключення до локальної мережі установи; швидкість з'єднання не менше ніж 10 Мбит/с; та використання компоненту .NET Framework 3.5. Також потрібен ярлик (посилання) для відкриття та розпочинання роботи додатку та для стабільності роботи додатку потрібне резервування оперативної пам'яті у 50 МБ.

Для забезпечення функціональності передбачено зручний та інтуїтивно зрозумілий інтерфейс із підтримкою ввімкнення одного або кількох комп'ютерів одночасно, додавання, редагування, видалення даних про пристрої та можливість відправки сигналу в іншу мережу (за сегмент L2). Реалізована можливість інтеграції з Windows Task Scheduler та WinBox від MikroTik.

Алгоритм роботи програми передбачає попереднє попередньо підготувати MAC-адреси комп'ютерів, завантажити додаток, запустити ярлик

«WoL\_exe.exe», внести MAC-адрес через спеціальне вікно програми, які потім будуть використовуватися для відправлення Wake-on-LAN пакетів та перезапустити додаток. Після налаштувань програма автоматично оновлює список доступних пристроїв, забезпечуючи швидкий доступ до основних функцій.

Техніко-економічні вимоги до кінцевого продукту такі:

- 1) Вартість робіт по розробці даної ПР визначається згідно договору на розробку.
- 2) Вартість запропонованих аналогів повинна забезпечити економічну доцільність їх застосування.
- 3) Програма повинна працювати краще існуючих аналогів.

Процес розробки передбачає декілька етапів, серед яких аналіз існуючих рішень, детальне дослідження технології, розробка та тестування основних модулів, створення остаточної версії програмного забезпечення та підготовка звітних матеріалів. Для прийняття роботи передбачено комплексне тестування, що включає функціональне тестування, перевірку коректності введених даних, продуктивність та аналіз безпеки [14]. Завершення роботи включає оцінку результатів замовником, подання коду програми, документації та звітів. Такий підхід дозволяє гарантувати високу якість розробки та її відповідність потребам користувачів.

## **2.2. Основні функціональні вимоги до програмного додатку**

Розробка програмного додатку для реалізації функції Wake-on-LAN (WoL) базувалася на аналізі потреб користувачів, технічних обмежень та подібних програм. Основні вимоги першої версії програми включають:

1. **Сумісність з Windows.** Додаток повинен працювати на платформі Windows, яка є найбільш поширеною серед користувачів.

2. **Зручний інтерфейс.** Інтуїтивно зрозумілий дизайн дозволить швидко знаходити потрібні функції та уникати складних інструкцій.
3. **Ввімкнення комп'ютерів через LAN.** Основна функція полягає в надсиланні WoL-пакетів для дистанційного ввімкнення пристроїв.
4. **Додавання нових комп'ютерів.** Користувачі повинні мати можливість легко додавати записи з назвами та MAC-адресами.
5. **Збереження даних.** Список комп'ютерів має зберігатися локально для подальшого використання після перезапуску.
6. **Вибір одного або кількох комп'ютерів.** Додаток має підтримувати одночасне ввімкнення кількох пристроїв.
7. **Валідація даних.** Введені MAC-адреси перевіряються для уникнення помилок.
8. **Простота використання.** Легкість встановлення та мінімальні вимоги до технічних знань забезпечать швидкий старт.
9. **Стабільність роботи.** Додаток повинен бути відмовостійким та надавати зрозумілі повідомлення про помилки.

В процесі аналізу попередніх версій цього додатку встановлено додаткові функціональні вимоги, а саме:

1. Запуск програми з командної строки, а саме відправка магічного пакета без ввімкнення самої програми.
2. Можливість планувати ввімкнення комп'ютерів зарозкладом, або у бажаний час.
3. Забезпечити безпеку, а саме встановити паролі на кожен BIOS, та відправляти магічний пакет з паролем, захистившись від несанкціонованого ввімкнення ПК третіми особами.
4. Можливість ввімкнення обчислювальних машин за межами однієї мережі, а саме за межами сегменту L2.

Дотримання початкових вимог та вимог у другій версії додатка забезпечує надійність, зручність та ефективність розробленого продукту.

### 2.3. Розробка та впровадження програмного додатку

Розроблений додаток для Wake-on-LAN (WoL) забезпечує дистанційне ввімкнення комп'ютерів через локальну мережу за допомогою "магічних пакетів". Основні функції включають:

1. **Відправлення WoL-пакетів.** Додаток формує та надсилає магічні пакети, що містять MAC-адресу цільового комп'ютера. Реалізація виконується у формі №1.
2. **Збереження даних.** Інформація про комп'ютери (назви та MAC-адреси) зберігається у текстовому файлі, що забезпечує доступність даних після перезапуску. Реалізація виконується у формі №3.
3. **Керування списком комп'ютерів.** Користувачі можуть додавати, редагувати та видаляти записи. Додаток забезпечує простий доступ до цих функцій через форму №2.
4. **Одночасне ввімкнення кількох пристроїв.** Можливість вибору декількох комп'ютерів для одночасного ввімкнення спрощує управління великими мережами.
5. **Валідація MAC-адрес.** Додаток перевіряє правильність введених MAC-адрес, використовуючи регулярні вирази, щоб уникнути помилок [17].
6. **Планування ввімкнення.** Присутня можливість ввімкнення як одного так і декількох комп'ютерів у потрібний час, запланувавши заздалегідь.
7. **Відправка сигналу з командного рядка.** Реалізована змога відправки сигналу не вмикаючи додаток, а просто ввівши потрібну команду в командний рядок.
8. **Сигнал за межі сегменту L2.** Доступна функція відправки магічного пакета за межі власної мережі реалізована за допомогою MikroTik.
9. **Безпека.** Поява підґрунтя називати додаток безпечним, адже тепер для ввімкнення будь-якого ПК потрібно ввести пароль.

Тепер функціонал додатку забезпечує надійну та просту взаємодію, відповідаючи потребам як корпоративних, так і домашніх користувачів та має

захист від небажаного втручання в систему. Код додатку (Додаток А, Додаток Б, Додаток В, Додаток Г) має чотири класи, багато полів та методів. Для зображення структури додатку було розроблено детальну діаграму класів розробленого додатку (рис. 2.1).

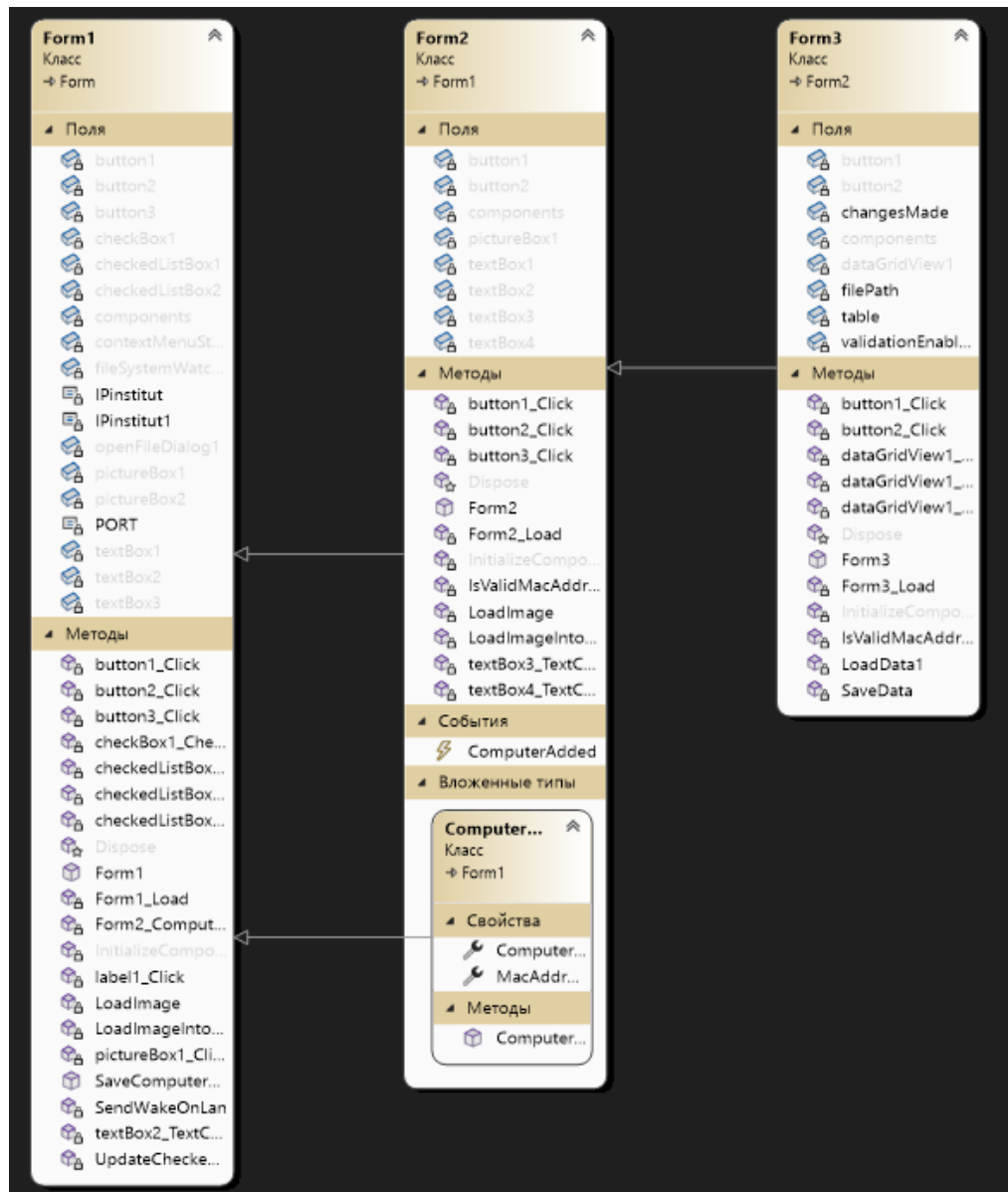


Рис. 2.1. Діаграма класів додатку

## 2.4. Огляд інтерфейсу користувача комп'ютерного додатку

Інтерфейс додатку було розроблено з акцентом на простоту та зручність використання. Основні елементи інтерфейсу включають:



1. **Список комп'ютерів.** Таблиця, що дозволяє вибирати пристрої для відправлення WoL-пакетів (рис. 2.2).

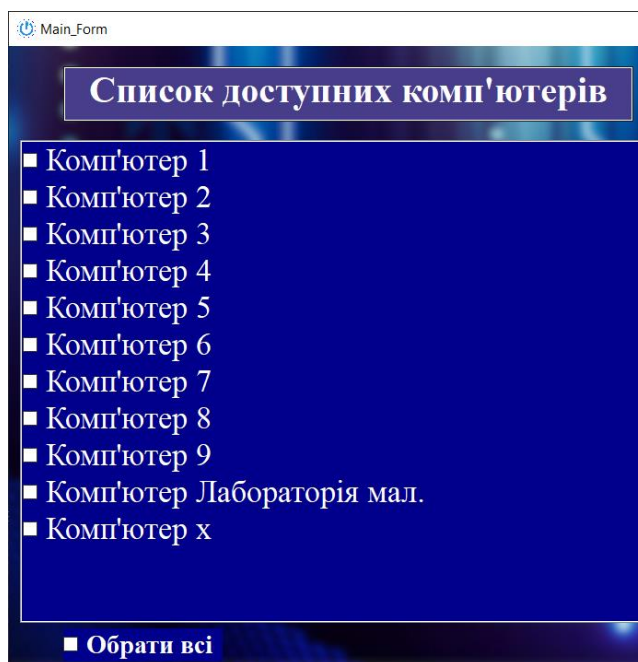


Рис. 2.2. Список комп'ютерів

2. **Кнопки управління.** Додаток надає кнопки для додавання, редагування, видалення комп'ютерів, а також для надсилення WoL-пакетів як миттєво так і з можливістю планування (рис. 2.3).

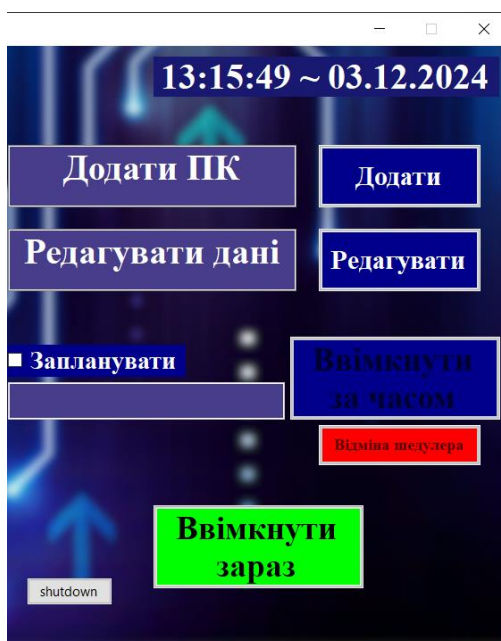
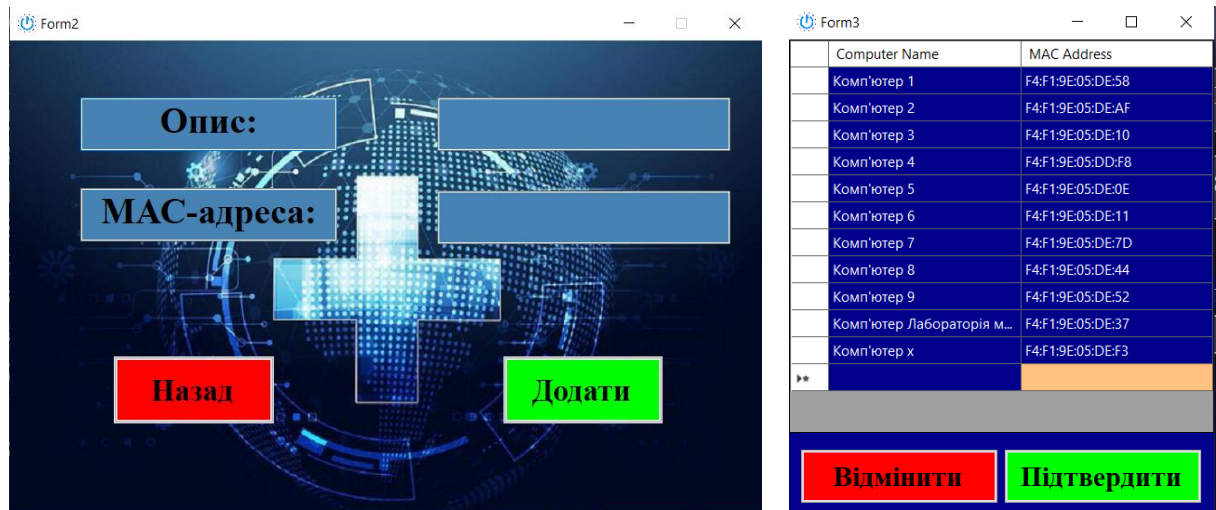


Рис. 2.3. Кнопки управління

3. **Форми додавання та редагування.** Інтуїтивно зрозуміле вікно для введення назв комп'ютерів та їхніх MAC-адрес із перевіркою валідності введених даних (рис. 2.4).



Computer Name	MAC Address
Комп'ютер 1	F4:F1:9E:05:DE:58
Комп'ютер 2	F4:F1:9E:05:DE:AF
Комп'ютер 3	F4:F1:9E:05:DE:10
Комп'ютер 4	F4:F1:9E:05:DD:F8
Комп'ютер 5	F4:F1:9E:05:DE:0E
Комп'ютер 6	F4:F1:9E:05:DE:11
Комп'ютер 7	F4:F1:9E:05:DE:7D
Комп'ютер 8	F4:F1:9E:05:DE:44
Комп'ютер 9	F4:F1:9E:05:DE:52
Комп'ютер Лабораторія м...	F4:F1:9E:05:DE:37
Комп'ютер x	F4:F1:9E:05:DE:F3

Рис. 2.4. Форми додавання та редагування

4. **Повідомлення та підказки.** Додаток інформує про успішність виконання операцій або помилки, що допомагає користувачам орієнтуватися під час роботи (рис. 2.5).

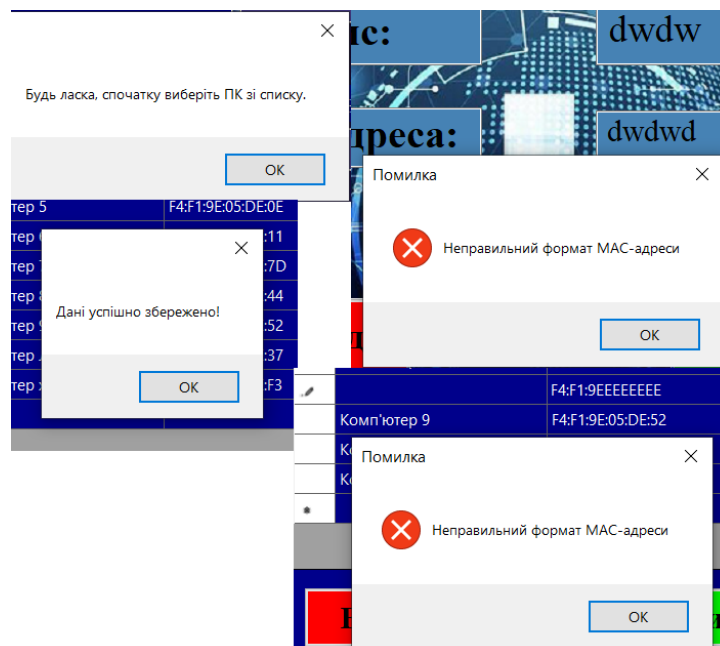


Рис. 2.5. Повідомлення про помилки та підказки

Для зручності запуску додаток можна перенести на робочий стіл та створити ярлик, що спрощує доступ до основного виконуваного файлу WoL\_exe.exe (рис. 2.6). Інтерфейс спрямований на забезпечення максимальної простоти використання навіть для користувачів без технічних знань.

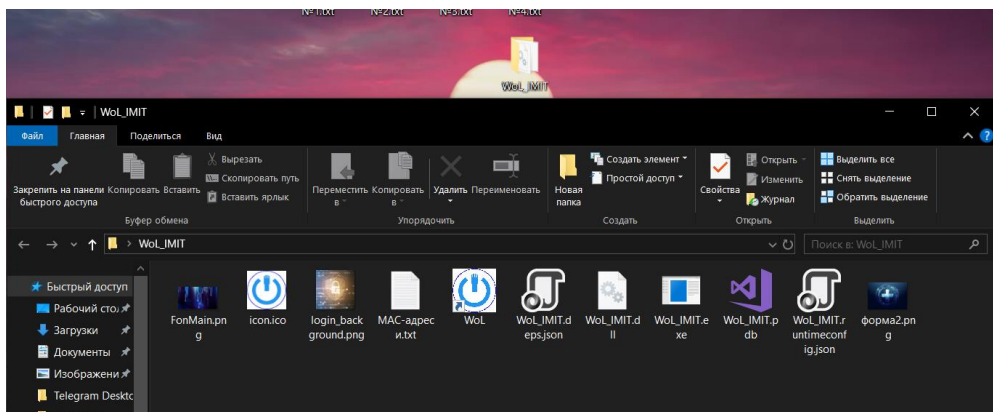


Рис. 2.6. Папка для роботи програми

## Висновок до розділу

У цьому розділі було описано розробку програмного додатку для реалізації функції Wake-on-LAN (WoL), що забезпечує дистанційне ввімкнення комп'ютерів у локальній мережі. Було визначено основні вимоги до програмного забезпечення, включаючи сумісність із Windows, зручність інтерфейсу, стабільність роботи та можливість збереження даних.

Функціональність додатку охоплює відправлення WoL-пакетів, керування списком комп'ютерів, перевірку валідності MAC-адрес та одночасне ввімкнення кількох пристроїв. Реалізація цих можливостей сприяє простоті використання та надійності додатку.

Інтерфейс розроблений для інтуїтивної взаємодії, з акцентом на доступність та зрозумілість. Користувачам надається можливість швидко керувати комп'ютерами через таблицю, кнопки управління та форми редагування. Система повідомлень допомагає уникати помилок і підвищує ефективність роботи.

Результатом роботи стало створення зручного та функціонального інструменту, який відповідає сучасним вимогам до програм для адміністрування комп'ютерних мереж. Додаток демонструє надійність, простоту використання та високий рівень адаптивності для корпоративних і навчальних середовищ.

## РОЗДІЛ III

### ДОСЛІДЖЕННЯ БЕЗПЕКИ ТА ШЛЯХИ ПІДВИЩЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМНОГО ДОДАТКУ

#### 3.1. Обмеження доступу до BIOS

Базова система вводу-виводу (BIOS) або її сучасний аналог — розширений інтерфейс Unified Extensible Firmware Interface (UEFI) — є програмним забезпеченням, яке забезпечує основне управління апаратним забезпеченням комп'ютера на момент його завантаження. Одним із ключових механізмів захисту BIOS/UEFI є використання пароля. Це захисний параметр, який обмежує доступ до налаштувань системи, зокрема до функцій ввімкнення комп'ютера за допомогою Wake-on-LAN (WoL).

Пароль на BIOS виконує наступні завдання:

1. **Обмеження доступу.** Запобігає несанкціонованому доступу до налаштувань BIOS/UEFI, захищаючи критичні параметри системи.
2. **Підвищення безпеки мережі.** Перешкоджає використанню WoL для несанкціонованого включення комп'ютерів, які можуть бути під'єднані до корпоративної мережі.
3. **Контроль доступу до апаратного забезпечення.** Забезпечує обмеження доступу до функцій апаратного рівня, таких як налаштування енергоспоживання чи порядку завантаження.

Використання пароля на BIOS забезпечує додатковий рівень безпеки у випадках, коли фізичний доступ до комп'ютера може бути небажаним або коли він використовується у великій мережевій інфраструктурі. Завдяки цьому WoL стає більш захищеним інструментом для віддаленого управління, оскільки навіть отримання доступу до мережевої карти не дозволяє обійти налаштування BIOS.

Залежно від виробника материнської плати або ноутбука, структура меню BIOS/UEFI та методи налаштування пароля можуть дещо відрізнятися. У цьому підрозділі розглянемо специфіку налаштування для таких популярних виробників, як **ASUS, Dell, HP, Lenovo, MSI** та **ACER**.

### **ASUS**

#### **Доступ до BIOS:**

- Під час завантаження системи натисніть клавішу Del або F2.

#### **Налаштування пароля:**

1. У меню BIOS перейдіть до вкладки «**Advanced**» або «**Security**» (залежить від версії прошивки).
2. Оберіть опцію «**Set Supervisor Password**».
3. Введіть пароль і підтвердьте його.
4. Якщо потрібен пароль для запуску системи, увімкніть параметр «**Set User Password**».

#### **Особливості для WoL:**

- Перейдіть до розділу **Advanced > APM Configuration**.
- Активуйте параметр «**Power On By PCI-E/PCI**».
- Переконайтеся, що опція WoL увімкнена, навіть якщо пароль активний.

### **DELL**

#### **Доступ до BIOS:**

- Натисніть F2 під час завантаження системи.

#### **Налаштування пароля:**

1. У меню BIOS виберіть розділ «**Security**».
2. Оберіть «**Admin Password**» для захисту налаштувань BIOS або «**System Password**» для обмеження доступу до запуску системи.
3. Введіть пароль і підтвердьте його.

### **Особливості для WoL:**

- Увімкніть параметр WoL у розділі **Power Management > Wake on LAN**.
- Якщо ввімкнений пароль, переконайтеся, що активована функція «**Enable WOL even when Admin Password is set**».

## **HP**

### **Доступ до BIOS:**

- Натисніть Esc або F10 під час завантаження.

### **Налаштування пароля:**

1. У меню BIOS знайдіть розділ «**Security**».
2. Оберіть опцію «**Administrator Password**» або «**Power-on Password**» залежно від цілей захисту.
3. Введіть бажаний пароль та збережіть зміни.

### **Особливості для WoL:**

- Увімкніть опцію WoL у розділі **Advanced > Built-In Devices Options**.
- Переконайтеся, що параметр «**Allow WOL with BIOS Password**» активний.

## **LENOVO**

### **Доступ до BIOS:**

- Натисніть F1 або Fn + F2 (залежно від моделі) під час завантаження.

### **Налаштування пароля:**

1. У меню BIOS виберіть розділ «**Security**».
2. Оберіть опцію «**Set Supervisor Password**» [20].
3. Для захисту доступу до системи додатково встановіть «**Power-On Password**».

### **Особливості для WoL:**

- Перейдіть до розділу «**Power Management**».

- Активуйте опцію WoL.
- У деяких моделях Lenovo необхідно також увімкнути «**Deep Sleep Control**» для коректної роботи WoL.

### **MSI**

#### **Доступ до BIOS:**

- Натисніть Del або F2 під час завантаження.

#### **Налаштування пароля:**

1. У меню BIOS перейдіть до розділу **Settings > Security**.
2. Оберіть «**Administrator Password**» для обмеження доступу до налаштувань або «**User Password**» для захисту запуску системи [21].
3. Введіть пароль та підтвердьте його.

#### **Особливості для WoL:**

- Перейдіть до розділу **Settings > Advanced > Power Management Setup**.
- Активуйте опцію «**Resume by PCI-E Device**».
- Переконайтеся, що налаштування мережевої карти у Windows відповідають вимогам WoL.

#### **Загальні рекомендації для всіх виробників**

##### **1. Збереження пароля:**

- Рекомендується зберігати паролі в надійному місці, оскільки їх втрата може ускладнити доступ до системи.

##### **2. Оновлення BIOS:**

- Перед увімкненням пароля переконайтеся, що прошивка BIOS/UEFI оновлена до останньої версії для забезпечення стабільної роботи.

##### **3. Тестування налаштувань:**

- Перевірте, чи працює WoL після встановлення пароля. Якщо функція не працює, зверніться до документації виробника для додаткових налаштувань.



Такий підхід допомагає врахувати особливості кожного виробника та зробити налаштування максимально адаптивним до конкретних умов.

### 3.2. Приклад налаштування обмеження доступу до BIOS

Процес впровадження пароля на BIOS є базовим етапом забезпечення безпеки комп'ютера. Адже маючи MAC-адресу цільового ПК та будучи з ним в одній мережі, можна несанкціоновано вмикати його, що неприпустимо для навчальних лабораторій. Отже, потрібно запобігти тому, щоб хтось дізнався MAC-адресу ПК, тому треба поставити пароль на BIOS. Щоб поставити пароль, потрібно виконати наступні кроки:

#### 1. Доступ до BIOS/UEFI:

- Під час завантаження системи натисніть спеціальну клавішу (зазвичай Del, F2, F10 або Esc), щоб увійти до налаштувань BIOS/UEFI (рис. 3.1).
- Для деяких сучасних систем із швидким завантаженням може знадобитися перезапуск системи в режимі **Advanced Startup** через меню операційної системи.

Після входу в BIOS на екрані з'явиться основне меню зображене на рисунку 3.2.

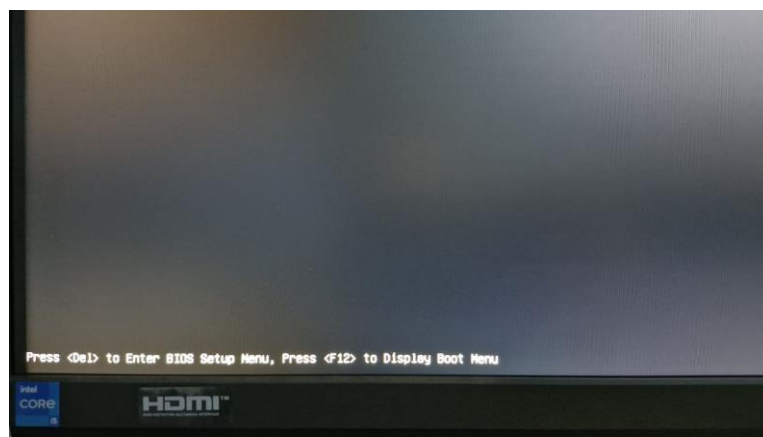


Рис. 3.1. Вхід в BIOS

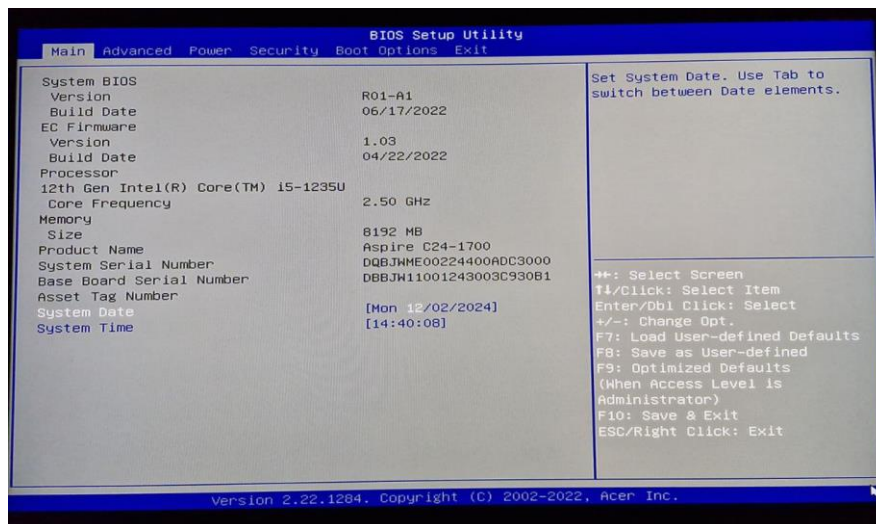


Рис. 3.2. BIOS

## 2. Активація пароля:

- У меню BIOS знайдіть розділ «**Security**» або схожий параметр (назва може відрізнятись залежно від версії прошивки та виробника материнської плати) (рис. 3.3).
- У розділі «**Set Supervisor Password**» або «**Set User Password**» введіть бажаний пароль. «**Supervisor Password**» зазвичай використовується для захисту налаштувань BIOS, а «**User Password**» — для обмеження доступу до запуску системи.

У з'явившомуся вікні потрібно ввести пароль та підтвердити його, ввівши ще раз ті самі символи (рис. 3.4).

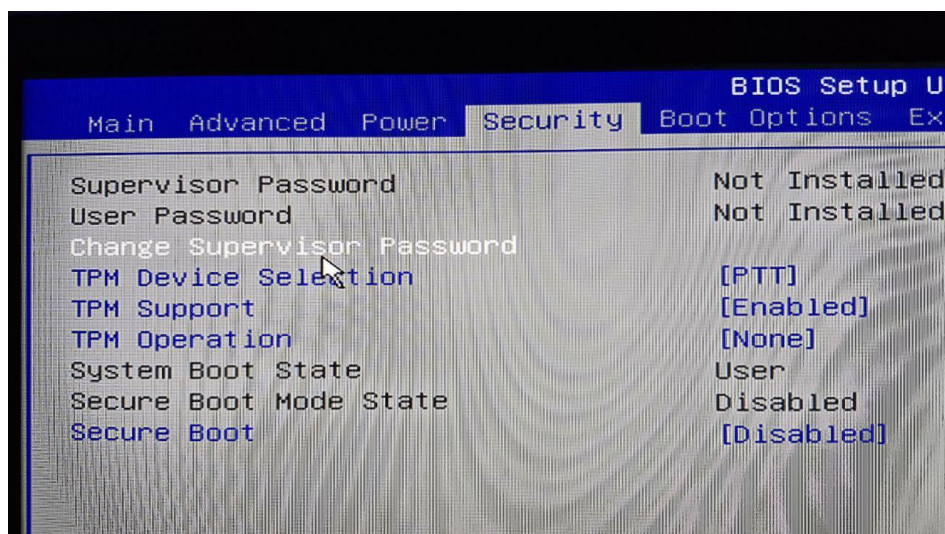


Рис. 3.3. Налаштування паролю на BIOS

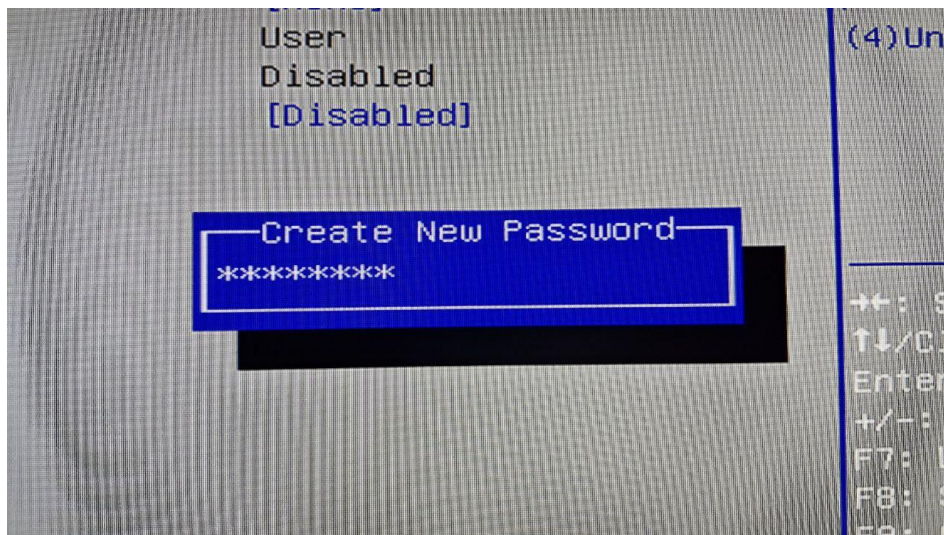


Рис. 3.4. Введення паролю

### 3. Збереження налаштувань:

- Після встановлення пароля обов'язково збережіть зміни, натиснувши клавішу F10 або вибравши опцію «**Save and Exit**» (рис. 3.5).

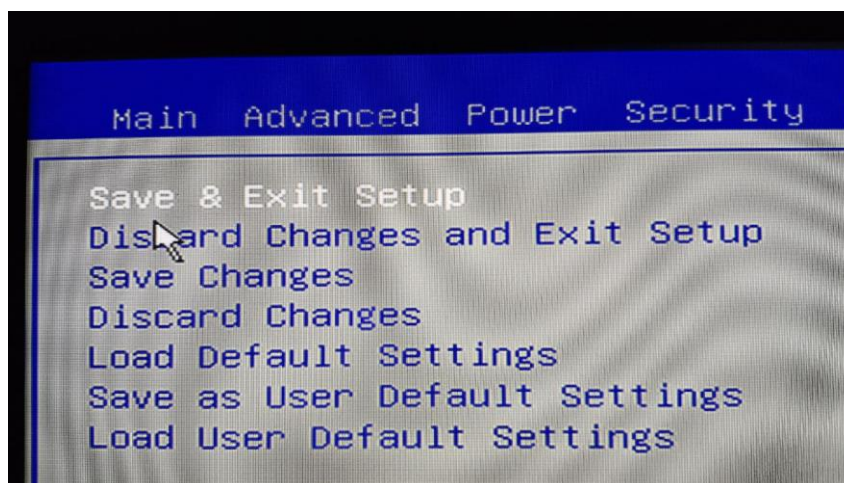


Рис. 3.5. Збереження та вихід

### 4. Тестування захисту:

- Перезавантажте систему, щоб перевірити, чи вимагає BIOS введення пароля для доступу до налаштувань або запуску комп'ютера.

Після перезавантаження системи та повторній спробі входу в BIOS вже потрібно буде вводити встановлений пароль (рис. 3.6).

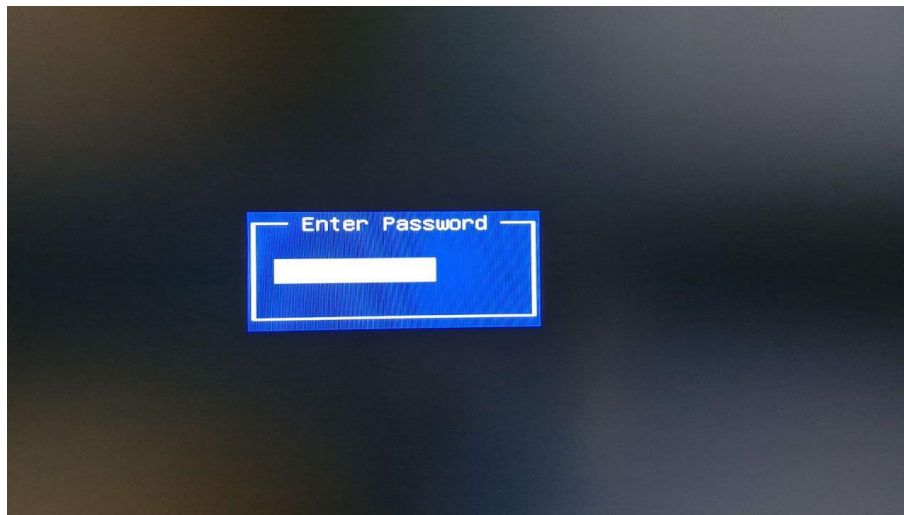


Рис. 3.6. Результат – встановлений пароль на BIOS

Таким чином ми захистили ПК від навмисних спроб дізнатися MAC-адресу нашого пристрою, тим самим захистили комп'ютер від несанкціонованого ввімкнення по мережі.

#### ***Особливості налаштувань пароля BIOS для підтримки Wake-on-LAN:***

- У деяких системах активований пароль може блокувати автоматичну ініціацію WoL. У такому випадку необхідно:
  - Перевірити, чи доступна опція «**Allow WoL When Password Is Set**» (або схожа), і активувати її.
  - Переконаватися, що параметри живлення мережевої карти налаштовані правильно для прийому "магічного пакета", навіть за умов активного пароля.

### **3.3. Використання пароля на BIOS у процесі відправки магічного пакета**

Пароль на BIOS впливає на функціональність Wake-on-LAN, додаючи рівень безпеки, але також ускладнюючи процес пробудження пристрою. Розглянемо, як це працює:

#### **1. Прийом магічного пакета:**



- Мережевий адаптер продовжує виконувати моніторинг мережі навіть у вимкненому стані, якщо WoL активовано в BIOS/UEFI.
- Якщо пароль на BIOS налаштовано правильно, адаптер приймає "магічний пакет" і передає сигнал на центральний процесор.

## **2. Аутентифікація:**

- У разі активного «Supervisor Password» або «User Password», BIOS перевіряє, чи дозволено пробудження системи без прямого введення пароля.
- Якщо налаштування дозволяють пробудження, система починає стандартний процес завантаження ОС.

## **3. Відмова у пробудженні:**

- У разі, якщо пароль не дозволяє Wake-on-LAN, система залишається у вимкненому стані. Це запобігає можливим спробам використання WoL для несанкціонованого доступу [23].

### **Переваги використання пароля BIOS у поєднанні з WoL:**

- Забезпечується багаторівневий захист: навіть якщо зломисник отримав доступ до локальної мережі, без знання пароля BIOS пробудження системи неможливе.
- Уникаються випадки випадкового або помилкового включення систем, що може бути проблемою у великих мережах.

### **Можливі обмеження:**

- У складних інфраструктурах із сотнями машин адміністрування паролів може бути трудомістким.
- Неправильно налаштований пароль може повністю блокувати функціональність WoL, що вимагає повторного налаштування [23].

В рамках дослідження було проаналізовано можливості використання пароля BIOS для забезпечення додаткового рівня захисту комп'ютерів, що працюють із технологією Wake-on-LAN (WoL). Основна ідея полягала в тому,

щоб запобігти несанкціонованому ввімкненню пристроїв третіми особами через локальну мережу.

#### Ключові аспекти дослідження:

1. **Механізм SecureOn.** SecureOn є функцією в деяких BIOS, що дозволяє додатково захистити ввімкнення комп'ютера через WoL шляхом використання спеціального пароля. Після отримання магічного пакета система перевіряє, чи збігається зазначений у ньому пароль із налаштуваннями BIOS, і лише тоді пробуджує пристрій. Це забезпечує двофакторний захист: фізичний (через BIOS) і мережевий (через WoL).
2. **Результати тестування на Acer.** У ході тестування на тестовій моделі Acer було встановлено, що BIOS цієї моделі не підтримує функцію SecureOn. Це стало перешкодою для інтеграції WoL із використанням пароля, що вказує на апаратні обмеження. Виявлено, що стандартний пароль BIOS забезпечує захист лише доступу до налаштувань BIOS або запуску системи, але не контролює прийом магічних пакетів.

Для захисту комп'ютерів від несанкціонованого ввімкнення було розроблено захист програми, де реалізовано вхід в систему. Для входу в програму та функціонувати в ній, потрібно ввести логін та пароль, встановлений раніше (рис. 3.7).

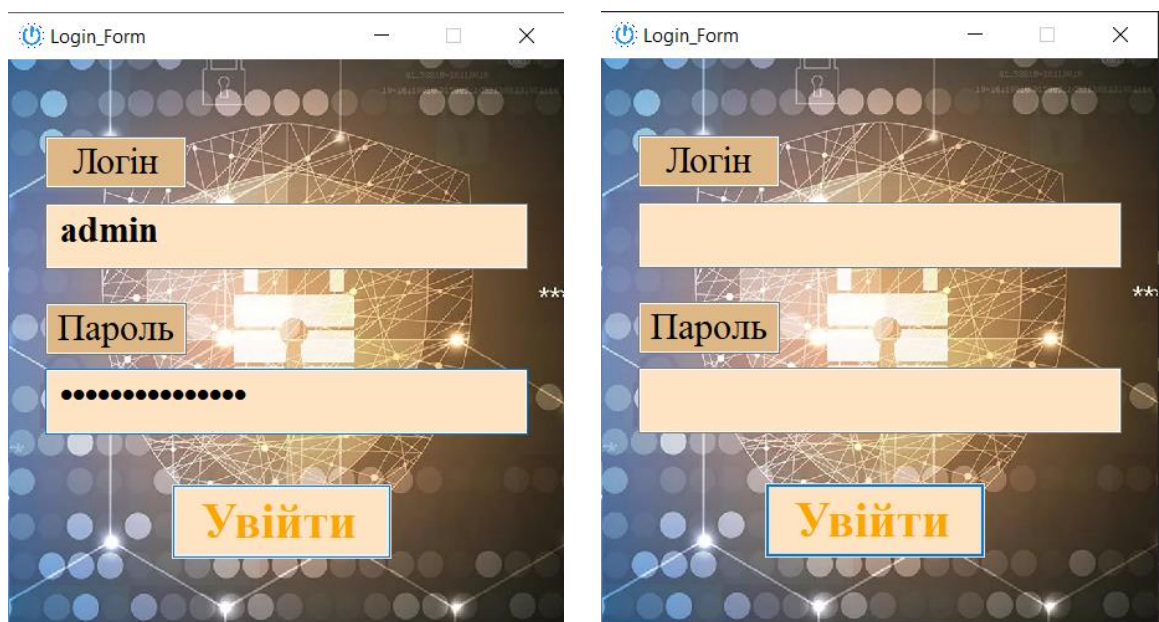


Рис. 3.7 Форма для входу в програму

У випадку, коли дані були введені невірно – виникає сповіщення про помилку та проханням спробувати знову (рис. 3.8).



Рис. 3.8. Сповіщення про неправильні дані

Кількість спроб – 3, це зроблено для того, щоб не можна було підбирати пароль до нескінченності. Після трьох хибних спроб програма припиняє роботу. Реалізація в коді показана на рисунку 3.9.

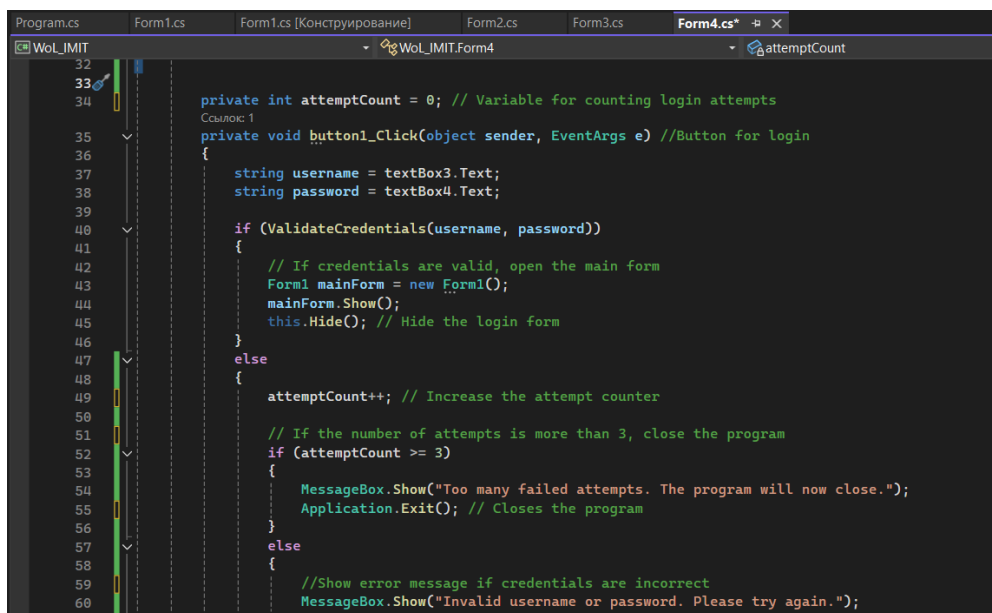


Рис. 3.9. Захист від підбору пароля

Дане рішення захищає комп'ютери не повністю, адже знаючи MAC-адресу будь-якого ПК в системі і знаходячись в одній мережі з комп'ютером

можна за допомогою іншої програми надіслати сигнал та успішно ввімкнути комп'ютер. Було досліджено в другому розділі, що для того, щоб захиститися від таких ситуацій, потрібно встановити пароль на BIOS і тим самим треті особи не дізнаються MAC-адресу обчислювальної машини і не зможуть несанкціоновано вмикати її.

### **3.4. Інтеграція WoL-додатка з Windows Task Scheduler**

Інтеграція WoL-додатка з «Планувальником завдань» Windows (Windows Task Scheduler) дозволяє автоматизувати процес надсилання WoL-сигналів у визначений час. Це дає можливість запланувати ввімкнення комп'ютерів без необхідності ручного втручання, що значно підвищує ефективність управління мережею.

Також є особливості інтеграції, а саме використання Windows Task Scheduler для WoL-додатка передбачає:

1. **Створення запланованих завдань.** Додаток додає завдання до Планувальника завдань, яке запускає виконуваний файл WoL\_IMIT.exe із вказаною MAC-адресою.
2. **Налаштування часу виконання.** Користувач задає точний час у форматі «HH:mm:ss, dd.MM.yyyy», після чого завдання автоматично виконується у заданий момент.
3. **Гнучкість управління.** Завдання можна додавати, редагувати або видаляти як через додаток, так і безпосередньо в Windows Task Scheduler.

Щодо реалізації інтеграції, для цього було використано бібліотеку «Microsoft.Win32.TaskScheduler», яка надає API для взаємодії з «Планувальником завдань». Процес створення завдання включає наступні етапи:

1. **Парсинг часу.** Введені дані у текстовому полі (наприклад, textBox4) конвертуються у формат DateTime для подальшого використання.



2. **Налаштування завдання.** Завдання реєструється у Windows Task Scheduler із такими параметрами:

- Тригери (Triggers) - час виконання.
- Дія (Action) - запуск WoL\_IMIT.exe із параметром MAC-адреси.

3. **Додавання опису.** Завдання супроводжується описом, який вказує на його функцію (наприклад, "Wake-on-LAN завдання для MAC-адреси").

Код для створення завдання зображено на рисунку 3.10.

```
using (TaskService ts = new TaskService())
{
    TaskDefinition td = ts.NewTask();
    td.RegistrationInfo.Description = "Wake-on-LAN завдання для MAC-адреси: " + macAddress;
    td.Triggers.Add(new TimeTrigger { StartBoundary = scheduledTime, Enabled = true });
    td.Actions.Add(new ExecAction("WoL_IMIT.exe", macAddress, null));
    ts.RootFolder.RegisterTaskDefinition(taskName, td);
}
```

Рис. 3.10. Код для створення завдання

Переваги інтеграції:

- **Автоматизація:** Користувач може заздалегідь планувати час ввімкнення комп'ютерів та не хвилюватись за те, щоб комп'ютер, з якого було подано завдання, був завжди ввімкнутий.
- **Універсальність:** Завдання додаються до Windows Task Scheduler, що забезпечує їх доступність для перегляду та редагування та гарантує виконання.

Щоб реалізувати інтеграцію програми з «Планувальником завдань (Windows Task Scheduler)», потрібно використовувати бібліотеку «Microsoft.Win32.TaskScheduler». Вона забезпечує API для створення, редагування та видалення завдань у Windows Task Scheduler.

Ось покрокова інструкція:

1. Додайте бібліотеку «Microsoft.Win32.TaskScheduler» за допомогою команди: «Install-Package TaskScheduler».

2. Імпортуємо бібліотеку у проєкт до файлу «using» за допомогою строки: «using Microsoft.Win32.TaskScheduler;»
3. Далі для створення завдань у Windows Task Scheduler потрібно реалізувати цей код на рисунку 3.11

```
{
    private void AddTaskToScheduler(string taskName, string time, string macAddress)
    {
        try
        {
            // Парсимо час
            DateTime scheduledTime = DateTime.ParseExact(time, "HH:mm:ss, dd.MM.yyyy", null);

            using (TaskService ts = new TaskService())
            {
                // Створюємо нове завдання
                TaskDefinition td = ts.NewTask();
                td.RegistrationInfo.Description = "Wake-on-LAN завдання для MAC-адреси: " + macAddress;

                // Вказуємо час запуску
                td.Triggers.Add(new TimeTrigger
                {
                    StartBoundary = scheduledTime,
                    Enabled = true
                });

                // Додаток, який потрібно виконати
                td.Actions.Add(new ExecAction("WoL_IMIT.exe", macAddress, null)); // Замініть на шлях до вашого exe

                // Зберігаємо завдання
                ts.RootFolder.RegisterTaskDefinition(taskName, td);
            }

            MessageBox.Show("Завдання додано до Планувальника завдань.", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Помилка при додаванні завдання: {ex.Message}", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Рис. 3.11. Програмування коду програми

4. Виклик методу з форми реалізований через натискання кнопки на формі програми та вводу даних у форматі - «HH:mm:ss, dd.MM.yyyy», за допомогою строк коду на рисунку 3.12.

```
private void buttonAddTask_Click(object sender, EventArgs e)
{
    string taskName = "WakeOnLAN_" + DateTime.Now.Ticks;
    string time = textBox4.Text; // Припустимо, час вводиться у textBox4
    string macAddress = "00AABBCCDDEE"; // Замініть на вибрану MAC-адресу з інтерфейсу

    AddTaskToScheduler(taskName, time, macAddress);
}
```

Рис. 3.12. Виклик методу

5. Після запуску програми завдання з'явиться в Task Scheduler у папці "Task Scheduler Library". Там можливо перевірити його налаштування і протестувати.
6. Рекомендації щодо тестування. Перевірте права адміністратора для програми, оскільки Планувальник завдань може вимагати підвищених привілеїв. Використовуйте реальні шляхи до файлу WoL\_IMIT.exe, якщо цей файл знаходиться поза проектом.

Додатково можна реалізувати функції:

- Видалення завдань через додаток (рис. 3.13).
- Виведення списку запланованих завдань у інтерфейс програми(рис. 3.14).

```
using (TaskService ts = new TaskService())
{
    ts.RootFolder.DeleteTask("НазваЗавдання", false);
}
```

Рис. 3.13. Видалення завдань в планувальнику завдань

```
using (TaskService ts = new TaskService())
{
    foreach (var task in ts.AllTasks)
    {
        Console.WriteLine(task.Name);
    }
}
```

Рис. 3.14. Виведення списку запланованих завдань

Отже, інтеграція WoL-додатка з Windows Task Scheduler відкриває широкі можливості для автоматизації управління комп'ютерними системами, підвищуючи зручність та ефективність роботи в мережевих середовищах.

### 3.5. Інтеграція WoL-додатка з MikroTik.

На основі отриманих результатів дослідження та тестування з'явилася ідея модернізувати програму, зокрема зробити інтеграцію з MikroTik, за умови, що він є центром мережі в системі. Ідея полягає в тому, щоб додаток надсилав магічний пакет до MikroTik, який, у свою чергу, через скрипт передавав сигнал до цільового пристрою в іншому сегменті L2. Цей підхід забезпечить:

- Можливість роботи з пристроями, які знаходяться в іншій мережі, іншому сегменті L2.
- Гарантійне доставлення сигналу до цільового ПК.

Інтеграція WoL-додатка з маршрутизаторами MikroTik передбачає, що додаток буде надсилати сигнал на MikroTik, а маршрутизатор передаватиме WoL-пакет у потрібний сегмент мережі. Нижче поетапно показано, як це реалізувати.

#### *1. Налаштування маршрутизатора MikroTik*

##### **1.1. Налаштування для прийому WoL-сигналів**

1. Відкрити інтерфейс **WinBox** або веб-інтерфейс MikroTik [22].
2. Перейти до **IP > Firewall > NAT**:
  - Додати нове правило.
  - **Chain**: dstnat.
  - **Protocol**: UDP.
  - **Dst. Port**: 9 (порт для WoL).
  - **Action**: dst-nat.
  - **To Address**: IP-адреса комп'ютера (або широкомова адреса сегмента мережі, наприклад, 192.168.102.255).

- **To Ports:** 9 [4].

### 1.2. Додавання скрипта для передачі WoL-пакета

1. У **System > Scripts** додайте новий скрипт, наприклад, `send-wol`.
2. Код скрипта: `«/tool wol mac-address=XX:XX:XX:XX:XX:XX interface=ether2»`.

Треба замінити «XX:XX:XX:XX:XX:XX» на MAC-адресу комп'ютера, який потрібно пробудити, а «ether2» — на інтерфейс маршрутизатора, через який здійснюється передача [4].

Для реалізації того, щоб MikroTik отримав список MAC-адрес та відправив їм сигнали успішно краще використовувати скрипт, зображений на рисунку 3.15.

```
:foreach mac in=[list of MAC addresses] do={  
    /tool wol mac-address=$mac interface=ether2  
}
```

Рис. 3.15. Скрипт для отримання MAC-адрес

### 1.3. Додавання API доступу

- У **IP > Services** увімкніть **API** (порт за замовчуванням: 8728).
- Переконайтеся, що доступ до API дозволений для вашої локальної мережі у **IP > Firewall > Filter Rules** [20].

## 2. Особливості використання у програмному додатку

Додаток повинен використовувати API MikroTik для запуску відповідного скрипта. Для цього можна скористатися бібліотекою **Tik4Net** — це популярна бібліотека для взаємодії з MikroTik через API [5].

### 2.1. Встановлення бібліотеки

Встановіть **Tik4Net** через NuGet: «dotnet add package Tik4Net»

### 2.2. Приклад коду для виконання скрипта

Потрібно додати метод для надсилання команди на MikroTik (рис. 3.16).

```
using Tik4Net;
using Tik4Net.Objects;

private void SendWolViaMikroTik(string ip, string username, string password, string scriptName)
{
    try
    {
        using (TikConnection connection = ConnectionFactory.OpenConnection(TikConnectionType.Api))
        {
            connection.Open(ip, username, password);

            // Виклик скрипта
            connection.CallCommandSync("/system/script/run", $"=.id={scriptName}");

            MessageBox.Show("Сигнал успішно відправлено через MikroTik.", "Vcnix", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка при надсиланні команди на MikroTik: {ex.Message}", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Рис. 3.16. Метод для надсилання команди на MikroTik

### 2.3. Виклик методу

У програмі метод можна прив'язати до кнопки (рис. 3.17).

```
private void buttonSendToMikroTik_Click(object sender, EventArgs e)
{
    string ip = "192.168.100.1"; // IP маршрутизатора MikroTik
    string username = "admin"; // Логін MikroTik
    string password = "password"; // Пароль MikroTik
    string scriptName = "send-wol"; // Назва скрипта у MikroTik

    SendWolViaMikroTik(ip, username, password, scriptName);
}
```

Рис. 3.17. Прив'язання до кнопки

Отже, можна підсумувати і зобразити основну схему роботи на рисунку 3.18.

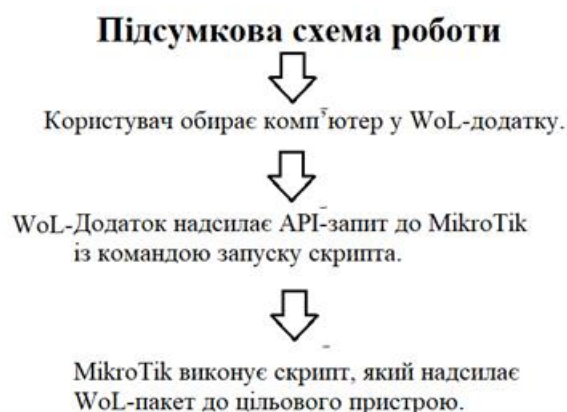


Рис. 3.18. Підсумкова схема роботи

### ***3. Переваги такого підходу***

- **Робота через сегменти L2.** MikroTik передає WoL-сигнали навіть у різних VLAN або сегментах мережі.
- **Централізоване управління.** Всі WoL-сигнали проходять через маршрутизатор.
- **Безпека.** Ви можете налаштувати обмежений доступ до API та запобігти зловживанню.

### **3.6. Тестування додатку**

Для перевірки стабільності та надійності роботи додатку було проведено стрес-тестування за допомогою спеціалізованих інструментів діагностики у середовищі Visual Studio (рис. 3.19). Тестування включало моделювання високого навантаження: часті запити на відправлення WoL-пакетів та оновлення списку комп'ютерів. Додатково на тестовій машині було запущено процеси копіювання файлів між каталогами, а також передачу даних по мережі з мережевого диска. Протягом випробувань додаток продемонстрував високу стійкість: жодних збоїв чи помилок не було виявлено навіть при значному навантаженні. Це свідчить про високу надійність програмного забезпечення та його готовність до реального використання.

Тестування продуктивності показало, що додаток для Wake-on-LAN має достатню швидкодію, здатний ефективно працювати з великими списками комп'ютерів і зберігати стабільність під навантаженням. Отримані результати дозволяють рекомендувати програму для використання в умовах віддаленого керування комп'ютерами, забезпечуючи надійне та ефективне виконання основних функцій.

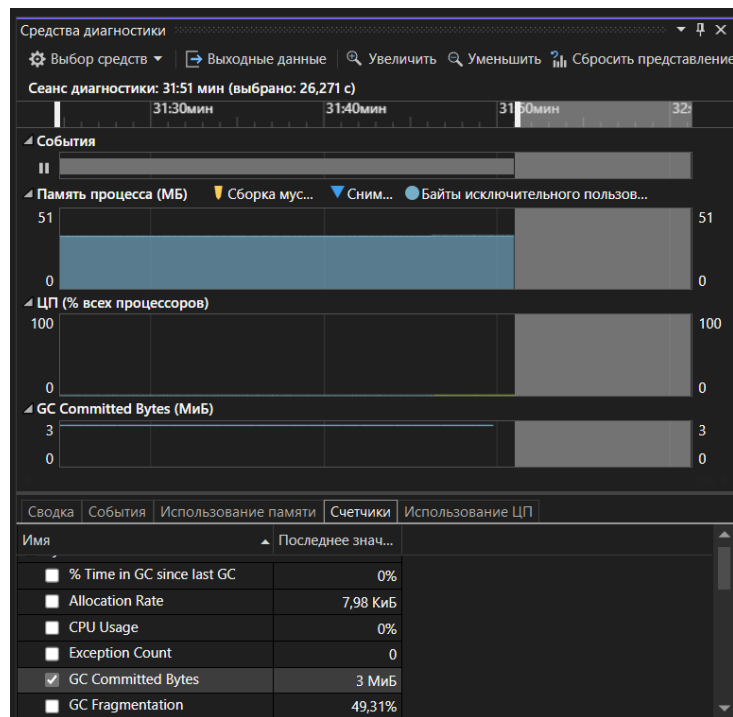


Рис. 3.19. Діагностика додатку

## Висновки до розділу

У третьому розділі було проведено дослідження безпеки та розроблено шляхи підвищення функціональних можливостей програмного додатку для Wake-on-LAN (WoL). Розглянуто методи захисту налаштувань BIOS шляхом встановлення пароля, що запобігає несанкціонованому доступу до обчислювальних машин і забезпечує контроль над їхнім дистанційним увімкненням.

Окремо було розглянуто специфіку налаштування BIOS для таких популярних виробників, як ASUS, Dell, HP, Lenovo та MSI. Вивчено особливості активації функції WoL у цих системах, а також налаштування паролів і параметрів живлення для забезпечення коректної роботи WoL навіть за умов активного захисту BIOS.

**Було проведено дослідження функціоналу Windows Task Scheduler.** Було показано механізм автоматизації відправлення Wake-on-LAN сигналів через створення завдань у планувальнику завдань. Це забезпечує можливість запуску WoL-пакетів у визначений час без участі користувача.



**Також було детально вивчено можливості інтеграції з MikroTik.** Розроблено механізм взаємодії програми з маршрутизатором MikroTik через API, що дозволяє передавати WoL-пакети до пристроїв у різних сегментах мережі. Було налаштовано скрипти на MikroTik для коректної обробки сигналів і визначено алгоритм їх передачі до цільових пристроїв.

Продемонстровано, що впровадження додаткових функцій і засобів захисту суттєво підвищує ефективність і безпеку використання WoL-додатку в корпоративних і навчальних середовищах.

#### **Результати роботи:**

- Забезпечено підтримку відправлення сигналів до кількох комп'ютерів одночасно.
- Встановлено, що інтеграція з MikroTik дозволяє вирішити проблему доступу до пристроїв у віддалених сегментах L2-мережі.
- Запропоновано рішення для підвищення безпеки передачі сигналів через маршрутизатор.

Було проведено тестування додатку в стресових умовах, яке було завершено успішно. Всі функції працювали стабільно, пакети відправлялися, дані оброблювалися коректно, а сигнал доходив до цільових ПК.

Отже, у результаті створено інструмент, який поєднує автоматизацію, мережеву інтеграцію та безпеку, що робить його ефективним засобом для управління комп'ютерними мережами в освітніх та корпоративних середовищах.

## ЗАГАЛЬНІ ВИСНОВКИ

Метою даного дослідження було створення програмного додатку для дистанційного керування обчислювальними машинами в навчальних комп'ютерних лабораторіях, що реалізує функцію ввімкнення пристроїв через технологію Wake-on-LAN (WoL). У сучасному світі проблема автоматизації та ефективного управління комп'ютерними мережами є надзвичайно актуальною, особливо в умовах постійних повітряних тривог, небезпеки життю. Все це змушує студентів та викладачів переміщуватися між аудиторіями та укриттями, а постійні перебої зі світлом загалом призводять до не продуктивного навчання та безперервних перебоїв у стабільній роботі з комп'ютерною технікою і тому потрібно масово використовувати дистанційні рішення. Технологія WoL є інструментом, що дозволяє оптимізувати процес управління, знижуючи витрати часу і ресурсів та вирішує проблему з повсякчасною необхідністю вручну вмикати та вимикати обладнання, що є не лише трудомістким і незручним процесом, а й суттєво уповільнює навчання, особливо за відсутності адмінів.

У процесі роботи було проведено детальне дослідження основ роботи WoL, виявлено апаратні та програмні вимоги, а також встановлено обмеження, пов'язані з її використанням. Особливу увагу приділено аналізу існуючих програмних рішень, серед яких NirSoft WakeMeOnLan, WakeOnLanGui, SolarWinds WoL, HM Software NL WoL, WinBox від MikroTik. Аналіз включав оцінку архітектури та функціоналу цих додатків, що дозволило виявити їх переваги та недоліки. На основі отриманих даних було сформульовано вимоги до функціональності та зручності власного додатку.

У ході роботи розроблено програмний додаток, який забезпечує:

- відправлення магічних пакетів WoL для ввімкнення комп'ютерів;
- управління списком пристроїв із можливістю їх додавання, редагування та видалення;
- підтримку одночасного ввімкнення кількох пристроїв;

- інтеграцію із Windows Task Scheduler для автоматизації завдань;
- передачу WoL-сигналів через MikroTik у різні сегменти мережі.

Додаток було протестовано на стабільність роботи, продуктивність і коректність обробки даних. Реалізовані функції дозволили забезпечити інтуїтивно зрозумілий інтерфейс та високу продуктивність навіть при роботі з великими списками пристроїв.

У результаті дослідження встановлено, що інтеграція додатку з Windows Task Scheduler дозволяє виконувати завдання автоматично за розкладом більш надійніше за створений власноруч шедулер, а використання MikroTik розширює можливості роботи з різними сегментами мережі. Було розроблено скрипти для маршрутизаторів MikroTik, які ефективно виконують функцію пересилання WoL-пакетів, забезпечуючи стабільну роботу системи.

Таким чином, мета роботи була досягнута — створено функціональний, продуктивний та зручний додаток, який відповідає сучасним вимогам до управління комп'ютерними системами. Впровадження цього рішення буде корисним у навчальних закладах і корпоративних середовищах, де потрібне ефективне дистанційне управління.

Подальший розвиток додатку може включати:

- реалізацію модулю для моніторингу стану комп'ютерів у реальному часі;
- інтеграцію з хмарними сервісами для дистанційного керування з будь-якого пристрою;
- покращення механізмів безпеки, включаючи використання двофакторної автентифікації;
- вдосконалення інтерфейсу на основі зворотного зв'язку від користувачів.

## СПИСОК ЛІТЕРАТУРИ

1. Могильний Г.А., Аналіз програмно-апаратних засобів створення системи з віддаленим доступом до навчальних комп'ютерних лабораторій закладів середньої освіти. № 1 (277) (2023): Вісник Східноукраїнського національного університету імені Володимира Даля. URL: <https://doi.org/10.33216/1998-7927-2019-256-8-5-19> (дата звернення: 10.06.2024).
2. Могильний Г.А., Семенов М.А., Кіреєв В.Ю. Впровадження системи віддаленого доступу до інформаційних ресурсів комп'ютерних лабораторій. № 2 (272) (2022): Вісник Східноукраїнського національного університету імені Володимира Даля. URL: <https://doi.org/10.33216/1998-7927-2022-272-2-7-14> (дата звернення: 10.06.2024).
3. Як увімкнути та настроїти Wake-on-LAN (WoL) у Windows? [Електронний ресурс] - URL: <https://tausoft.com.ua/yak-uvimknuty-ta-nastroyity-wake-on-lan-wol-u-windows/> (дата звернення: 10.10.2024).
4. Детально о WOL. [Електронний ресурс] - URL: <https://habr.com/ru/articles/242143/> СКРИПТЫ МИКРО ТИК (дата звернення: 10.07.2024).
5. Wake On LAN (WOL). [Електронний ресурс] - URL: <https://www.codeproject.com/Articles/11469/Wake-On-LAN-WOL> (дата звернення: 11.07.2024).
6. Depicus wol gui download. [Електронний ресурс] - URL: <https://www.depicus.com/wake-on-lan/wake-on-lan-gui> (дата звернення: 11.04.2024).
7. Wake on LAN (Magic Packet) download. [Електронний ресурс] - URL: <https://apps.microsoft.com/detail/9nblggh51pb3?cid=igromania2&hl=ru-ru&gl=RU> (дата звернення: 11.08.2024).
8. SolarWinds Wake-On-LAN download. [Електронний ресурс] - URL: <https://solarwinds-wake-on-lan.softonic.ru/> (дата звернення: 11.09.2024).
9. Mikrotik download. [Електронний ресурс] - URL: <https://mikrotik.com/download> (дата звернення: 11.09.2024).

10. Easy WOL (Wake on LAN). [Електронний ресурс] - URL: <https://apps.microsoft.com/detail/9nblggh5q03z?hl=uk-ua&gl=UA> (дата звернення: 11.09.2024).
11. Computer Networks: A Systems Approach / Брюс С. Деві, Ларрі Л. Петерсон. Elsevier, 2021. 848 с.
12. WakeMeOnLan v1.91 Copyright (c) 2011 - 2023 Nir Sofer. [Електронний ресурс] - URL: [https://www.nirsoft.net/utils/wake\\_on\\_lan.html](https://www.nirsoft.net/utils/wake_on_lan.html) (дата звернення: 21.08.2024).
13. Технічне завдання системи. [Електронний ресурс] - URL: [https://access.dp.ua/more/example\\_tz/](https://access.dp.ua/more/example_tz/) (дата звернення: 22.02.2024).
14. Види тестування. [Електронний ресурс] - URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/page13.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/page13.html) (дата звернення: 22.10.2024).
15. Wake-on-LAN. [Електронний ресурс] - URL: <https://uk.wikipedia.org/wiki/Wake-on-LAN> (дата звернення: 24.04.2024).
16. Комп'ютерні мережі / Коробейнікова Т.І., Захарченко С.М. Львівська політехніка, 2022. 228 с.
17. MAC адреси всі правила валідації. [Електронний ресурс] - URL: <http://surl.li/cmfwlz> (дата звернення: 29.05.2024).
18. Глобальна обчислювальна мережа (гвп). [Електронний ресурс] URL: <https://studfile.net/preview/5258946/page:3/> (дата звернення: 29.04.2024).
19. Що таке хмарні сервіси - ТОП 3 кращих від UCloud. [Електронний ресурс] URL: <https://ucloud.ua/shho-take-hmarni-servisy/> (дата звернення: 02.05.2024).
20. Налаштування BIOS. Set Supervisor Password. [Електронний ресурс] URL: <http://surl.li/rfdueb> (дата звернення: 05.10.2024).

21. How to change administrator password? [Електронний ресурс] URL: <https://learn.microsoft.com/en-us/answers/questions/1574782/how-to-change-administrator-password-when-you-forg> (дата звернення: 06.10.2024).
22. Корисні скрипти для mikrotik RouterOS. [Електронний ресурс] URL: <https://unix.ck.ua/content/skripy-dlya-mikrota> (дата звернення: 05.09.2024).
23. Комп'ютерні мережі. Книга 1. Технології комп'ютерних мереж / Євсєєв С.П., Дженюк Н.В. Новий світ-2000, 2024. 471 с.
24. Systems Programming: Designing and Developing Distributed Applications / Річард Ентоні. Elsevier, 2015. 548 с.

## ДОДАТОКИ

### ДОДАТОК А

#### Лістинг програмного коду. Основна форма №1

```
using System;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Windows.Forms;
using System.Diagnostics;
using System.Runtime.InteropServices;

namespace WoL_IMIT
{
    public partial class Form1 : Form
    {
        private System.Windows.Forms.Timer timer1;
        private System.Threading.Timer schedulerTimer;
        private DateTime scheduledTime;

        private string desktopPath;
        private string projectFolder;
        private string filePath;

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```

        desktopPath =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
        projectFolder = Path.Combine(desktopPath, "WoL_IMIT");
        filePath = Path.Combine(projectFolder, "MAC-адреса.txt");

        // Проверяем, не было ли уже создано поле timer1
        if (timer1 == null)
        {
            // Настройка таймера
            timer1 = new System.Windows.Forms.Timer();
            timer1.Interval = 1000; // Интервал в миллисекундах (1 секунда)
            timer1.Tick += Timer_Tick; // Привязка обработчика события Tick
            timer1.Start(); // Запуск таймера

            // Настройка Label для отображения времени и даты
            SetupLabel();
        }

        // Настройка Label для отображения времени и даты
        SetupLabel();

        this.Load += new System.EventHandler(this.Form1_Load);
        this.FormClosed += new FormClosedEventHandler(Form1_FormClosed);

        schedulerTimer = new System.Threading.Timer(CheckSchedule, null,
Timeout.Infinite, 1000);

    }

```



```

private void SetupLabel()
{
    // Настройка dateTimeLabel
    dateTimeLabel.AutoSize = true;
    dateTimeLabel.Font = new Font("Times New Roman", 24,
FontStyle.Bold);
    //dateTimeLabel.Location = new Point(10, 10); // Установите нужное
расположение
    dateTimeLabel.Text = DateTime.Now.ToString("HH:mm:ss ~
dd.MM.yyyy");
}

private void Timer_Tick(object sender, EventArgs e)
{
    // Обновление данных в Label
    dateTimeLabel.Text = DateTime.Now.ToString("HH:mm:ss ~
dd.MM.yyyy");
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    Application.Exit();
}

private const int PORT = 9;
public void SaveComputerData(string ComputerName, string MacAddress)
{
    string filePath = "MAC-адреса.txt";

```

```

        //додаємо нові дані до файлу
        using (StreamWriter writer = new StreamWriter(filePath, true))
        {
            writer.WriteLine(ComputerName);
            writer.WriteLine(MacAddress);
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        LoadImageIntoPictureBox();
        string txt_NAME_only_read = "MAC-адреси.txt"; //Імена

        if (File.Exists(txt_NAME_only_read))
        {
            string[] lines = File.ReadAllLines(txt_NAME_only_read);
            for (int i = 0; i < lines.Length; i++)
            {
                //додаємо лише непарні рядки (назви) до checkedListBox1
                if (i % 2 == 0)
                {
                    checkedListBox1.Items.Add(lines[i]);
                    checkedListBox2.Items.Add(lines[i + 1].Replace(":", "").Trim());
                }
                //додаємо MAC-адреси до другого checkedListBox, без розділювачів
            }
        }
        else
    }

```

```

{
    MessageBox.Show("Файл не знайдено");
}
}

```

//Послание сигнала

```

private const string IPinstitut = "192.168.100.255";
private const string IPinstitut1 = "192.168.100.255";

```

```

public bool SendWakeOnLan(string MacAddress)

```

```

{
    try
    {
        if (string.IsNullOrEmpty(MacAddress) || MacAddress.Length != 12)
        {
            throw new ArgumentException("MAC-адреса має бути 12
шістнадцяткових символів.");
        }
    }
}

```

```

byte[] MacBytes = new byte[6];

```

```

for (int i = 0; i < 6; i++)

```

```

{
    MacBytes[i] = Convert.ToByte(MacAddress.Substring(i * 2, 2), 16);
}

```

```

byte[] MagicPacket = new byte[6 + 16 * 6];

```

```

for (int i = 0; i < 6; i++)

```

```

{

```

```

        MagicPacket[i] = 0xFF;
    }
    for (int i = 6; i < MagicPacket.Length; i += 6)
    {
        MacBytes.CopyTo(MagicPacket, i);
    }

    using (UdpClient client = new UdpClient())
    {
        client.Send(MagicPacket, MagicPacket.Length, new
        IPEndPoint(IPAddress.Parse(IPinstitut), PORT));

        client.Send(MagicPacket, MagicPacket.Length, new
        IPEndPoint(IPAddress.Parse(IPinstitut1), PORT));

        client.Send(MagicPacket, MagicPacket.Length, new
        IPEndPoint(IPAddress.Broadcast, PORT));

        return true;
    }
}

catch (FormatException ex)
{
    MessageBox.Show("Неправильний формат MAC-адреси: " +
ex.Message);
}

catch (ArgumentException ex)
{
    MessageBox.Show("MAC-адреса має бути 12 шістнадцяткових
символів: " + ex.Message);
}

catch (Exception ex)
{

```

```

        MessageBox.Show("Помилка відправки Wake-on-LAN пакета: " +
ex.Message);
    }
    return false;
}

```

```

private void UpdateCheckedListBox2Selection()
{
    checkedListBox2.ClearSelected();

    foreach (int index in checkedListBox1.SelectedIndices)
    {
        if (index < checkedListBox2.Items.Count)
        {
            checkedListBox2.SetItemChecked(index, true);
        }
    }
}

```

```

private void LoadImageIntoPictureBox()
{
    try
    {
        string image1 = @"FonMain.png";

        pictureBox1.Image = LoadImage(image1);
    }
    catch (Exception ex)

```

```

        {
            MessageBox.Show($"Сталася помилка при завантаженні
зображення: {ex.Message}");
        }
    }

    private Image LoadImage(string imagePath)
    {
        if (File.Exists(imagePath))
        {
            return Image.FromFile(imagePath);
        }
        else
        {
            MessageBox.Show($"Файл зображення '{imagePath}' не існує");
            return null;
        }
    }

    private void checkedListBox2_SelectedIndexChanged(object sender,
EventArgs e)
    {
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Form2 AddForm = new Form2();
        AddForm.ShowDialog();
    }

```

```

private void button3_Click(object sender, EventArgs e)
{
    Form3 EditForm = new Form3();
    EditForm.Show();
}

private void button1_Click(object sender, EventArgs e)
{
    // Проверка, есть ли выбранные элементы в checkedListBox2
    if (checkedListBox2.CheckedItems.Count > 0)
    {
        // Создание массива для хранения выбранных MAC-адресов
        string[] arraySelMACAddr = new
string[checkedListBox2.CheckedItems.Count];

        // Заполнение массива выбранными MAC-адресами
        for (int i = 0; i < checkedListBox2.CheckedItems.Count; i++)
        {
            arraySelMACAddr[i] = checkedListBox2.CheckedItems[i].ToString();
        }

        // Отправка Wake-on-LAN пакетов для каждого MAC-адреса
        foreach (string macAddress in arraySelMACAddr)
        {
            SendWakeOnLan(macAddress);
        }
        MessageBox.Show("Сигнал успішно надісланий!");
    }
}

```

```

    }
    else
    {
        MessageBox.Show("Будь ласка, спочатку виберіть ПК зі списку.");
    }
}

```

```

private void checkedListBox1_SelectedIndexChanged(object sender,
EventArgs e)
{
    UpdateCheckedListBox2Selection();
}

```

```

private void checkedListBox1_ItemCheck(object sender,
ItemCheckEventArgs e)
{
    UpdateCheckedListBox2Selection();
    int selectedI = e.Index;

    if (e.NewValue == CheckState.Checked)
    {
        checkedListBox2.SetItemChecked(selectedI, true);
    }
    else if (e.NewValue == CheckState.Unchecked)
    {
        checkedListBox2.SetItemChecked(selectedI, false);
    }
}

```



```

private void checkBox1_CheckedChanged_1(object sender, EventArgs e)
{
    bool isChecked = checkBox1.Checked;

    for (int i = 0; i < checkedListBox1.Items.Count; i++)
    {
        checkedListBox1.SetItemChecked(i, isChecked);
    }

    for (int i = 0; i < checkedListBox2.Items.Count; i++)
    {
        checkedListBox2.SetItemChecked(i, isChecked);
    }
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

private void timer1_Tick(object sender, EventArgs e)
{
}

```

```

//Для Шедулерра
private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
    textBox4.Enabled = checkBox2.Checked;
    button4.Enabled = checkBox2.Checked;
    button1.Enabled = !checkBox2.Checked;
    button6.Enabled = checkBox2.Checked;
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

private void button4_Click(object sender, EventArgs e)
{
    if (checkedListBox2.CheckedItems.Count > 0)
    {
        if (DateTime.TryParseExact(textBox4.Text, "HH:mm:ss, dd.MM.yyyy",
null, System.Globalization.DateTimeStyles.None, out DateTime result))
        {
            scheduledTime = result;

            MessageBox.Show("Час і дата успішно збережені!\nПІК
ввімкнеться у запланований час.", "Успіх", MessageBoxButtons.OK,
MessageBoxIcon.Information);

            schedulerTimer.Change(0, 1000); // Запускаем таймер
        }
        else
    }
}

```

```

        {
            MessageBox.Show("Неправильний формат часу та дати. Будь ласка, введіть у форматі 00:00:00, 11.11.1111", "Помилка",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("Будь ласка, спочатку виберіть ПК зі списку.");
    }
}

private void CheckSchedule(object state)
{
    if (DateTime.Now >= scheduledTime)
    {
        this.Invoke(new Action(() =>
        {
            SendWakeOnLanToSelected();
        }));
        schedulerTimer.Change(Timeout.Infinite, Timeout.Infinite); //
        Останавливаем таймер после отправки
    }
}

private void SendWakeOnLanToSelected()
{
    if (checkedListBox2.CheckedItems.Count > 0)
    {

```

```

List<string> selectedMACAddresses = new List<string>();

foreach (var item in checkedListBox2.CheckedItems)
{
    selectedMACAddresses.Add(item.ToString());
}

bool signalSent = false;

foreach (string macAddress in selectedMACAddresses)
{
    bool result = SendWakeOnLan(macAddress);
    if (result)
    {
        signalSent = true;
    }
}

//MessageBox.Show("Сигнал успешно отправлен!");

}
else
{
    MessageBox.Show("Пожалуйста, сначала выберите ПК из списка.");
}
}

```

```

private void dateTimeLabel_Click(object sender, EventArgs e)
{

}

//ОТМЕНА ШЕДУЛЛЕРА
private void button6_Click(object sender, EventArgs e)
{
    schedulerTimer.Change(Timeout.Infinite, Timeout.Infinite);
    scheduledTime = DateTime.MinValue;

    MessageBox.Show("Запланированное задание отменено.", "Отмена",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

//Планировщик заданий (Windows Task Scheduler)

//ВЫКЛЮЧЕНИЕ ПК

//string ipAddress = "192.168.113.44";
//private void ShutdownSelectedComputers()
//{
//    if (checkedListBox2.CheckedItems.Count > 0)
//    {
//        foreach (var item in checkedListBox2.CheckedItems)
//        {
//            string ipAddress = item.ToString(); // Предполагается, что в
checkedListBox2 хранятся IP-адреса
//            ShutdownComputer(ipAddress);

```

```

//    }
//  }
//  else
//  {
//    MessageBox.Show("Пожалуйста, сначала выберите ПК из
списка.");
//  }
//}

//private void button5_Click(object sender, EventArgs e)
//{
//  string ipAddress = "192.168.113.44"; // Временный IP адрес

//  ShutdownComputer(ipAddress);
//}

//private void ShutdownComputer(string ipAddress)
//{
//  try
//  {
//    Process.Start("shutdown", $"/m \\\\:ipAddress} /s /f /t 0
/u:oon5\\ADMIN /p:Vjubkf_2023");
//    MessageBox.Show($"Команда на выключение отправлена на
{ipAddress}", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);
//  }
//  catch (Exception ex)
//  {
//    MessageBox.Show($"Ошибка при отправке команды на
выключение: {ex.Message}", "Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
//  }

```

```

    //}

    //private void ShutdownComputer(string ipAddress)
    //{
    //    try
    //    {
    //        var psi = new ProcessStartInfo("shutdown", $"/m \\\\:ipAddress} /s /f /t
0")
    //        {
    //            CreateNoWindow = true,
    //            UseShellExecute = false,
    //            RedirectStandardOutput = true,
    //            RedirectStandardError = true
    //        };
    //        using (Process process = Process.Start(psi))
    //        {
    //            process.WaitForExit();
    //        }

    //        MessageBox.Show($"Команда на выключение отправлена на
{ipAddress}", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);
    //    }
    //    catch (Exception ex)
    //    {
    //        MessageBox.Show($"Ошибка при отправке команды на
выключение: {ex.Message}", "Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    //    }
    //}

```

}

}



## ДОДАТОК Б

### Лістинг програмного коду. Основна форма №2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WoL_IMIT
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

            LoadImageIntoPictureBox();

        }
        public          event          EventHandler<ComputerAddedEventArgs>
ComputerAdded;
        private void LoadImageIntoPictureBox()
        {
            try
            {
                //шлях до зображення
                string image1 = @"форма2.png";
                pictureBox1.Image = LoadImage(image1);
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Сталася помилка при завантаженні
зображення: {ex.Message}");
            }
        }

        private bool IsValidMacAddress(string macAddress)
        {

```

```

        return System.Text.RegularExpressions.Regex.IsMatch(macAddress,
"^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$");
    }

    public class ComputerAddedEventArgs : EventArgs
    {
        public string ComputerName { get; }
        public string MacAddress { get; }

        public ComputerAddedEventArgs(string computerName, string
macAddress)
        {
            ComputerName = computerName;
            MacAddress = macAddress;
        }
    }

    private Image LoadImage(string imagePath)
    {
        // Перевіряємо, чи файл існує
        if (System.IO.File.Exists(imagePath))
        {
            // Завантажуємо зображення з файлу
            return Image.FromFile(imagePath);
        }
        else
        {
            MessageBox.Show($"Файл зображення '{imagePath}' не існує");
            return null;
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {
    }

    private void button2_Click(object sender, EventArgs e)
    {

```

```

        this.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        // Отримуємо назву комп'ютера та MAC-адрес з textBox3 та
textBox4
        string computerName = textBox2.Text;
        string macAddress = textBox4.Text;

        //перевірка на валідність MAC-адреси
        if (!IsValidMacAddress(macAddress))
        {
            MessageBox.Show("Неправильний формат MAC-адреси",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        //зберігаємо дані у файлі
        Form1 mainForm =
Application.OpenForms.OfType<Form1>().FirstOrDefault();
        if (mainForm != null)
        {
            mainForm.SaveComputerData(computerName, macAddress);
        }

        // Очистите текстовые поля
        textBox2.Text = string.Empty;
        textBox4.Text = string.Empty;

        // Покажите сообщение об успешном сохранении данных
        MessageBox.Show("Данні успішно збережено", "Успіх",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

## ДОДАТОК В

### Лістинг програмного коду. Основна форма №3

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WoL_IMIT
{
    public partial class Form3 : Form
    {
        //private string filePath = "C:\\Users\\ifmit\\source\\repos\\WoL_exe\\MAC-
        адреси.txt";
        private string filePath = "MAC-адреси.txt";
        private DataTable table;
        private bool changesMade = false; // ДЛЯ ПЕРЕВІРКИ ЧИ БУЛИ
        ВНЕСЕНІ ЗМІНИ
        private bool validationEnabled = true;
        private void dataGridView1_CellValueChanged(object sender,
        DataGridViewCellEventArgs e)
        {
            // Позначаємо, що були внесені зміни
            changesMade = true;
        }

        public Form3()
        {
            InitializeComponent();
        }

        private void Form3_Load(object sender, EventArgs e)
        {
            LoadData1();
        }
    }
}
```

```

        dataGridView1.CellValidating += dataGridView1_CellValidating;
//обробник подій для забезпечення того, що MAC-адреси відповідають
вимогам
    }

```

```

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

```

```

    private void dataGridView1_CellValidating(object sender,
DataGridViewCellValidatingEventArgs e)
    {
        if (e.ColumnIndex == 1 && e.RowIndex >= 0) // Перевіряємо, чи
редагується колонка MAC-адрес
        {
            string newMacAddress = e.FormattedValue.ToString();

            // Перевіряємо, чи новий MAC-адрес відповідає правильному
формату
            if (!string.IsNullOrEmpty(newMacAddress) &&
!IsValidMacAddress(newMacAddress))
            {
                MessageBox.Show("Неправильний формат MAC-адреси",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                e.Cancel = true; // Відхиляємо зміни
            }
        }
    }

```

```

    private bool IsValidMacAddress(string macAddress)
    {
        return System.Text.RegularExpressions.Regex.IsMatch
(macAddress, "^[0-9A-Fa-f]{2}[:-]{5}([0-9A-Fa-f]{2})$");
    }

```

```

    private void SaveData()
    {
        if (table != null)
        {
            using (StreamWriter writer = new StreamWriter(filePath))
            {

```

```

        foreach (DataRow row in table.Rows)
        {
            writer.WriteLine(row["Computer Name"]);
            writer.WriteLine(row["MAC Address"]);
        }
    }
}

private void LoadData1()
{
    if (File.Exists(filePath))
    {
        table = new DataTable();
        table.Columns.Add("Computer Name");
        table.Columns.Add("MAC Address");

        string[] lines = File.ReadAllLines(filePath);
        for (int i = 0; i < lines.Length; i += 2)
        {
            string computerName = lines[i];
            string macAddress = lines[i + 1];
            table.Rows.Add(computerName, macAddress);
        }

        dataGridView1.DataSource = table;
    }
    else
    {
        MessageBox.Show("Файл не знайдено");
    }
}

private void button1_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Ви впевнені, що хочете відмінити зміни?", "Підтвердження", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        changesMade = false;
        this.Close();
    }
}

```

```
}  
  
private void button2_Click(object sender, EventArgs e)  
{  
    SaveData();  
    MessageBox.Show("Дані успішно збережено!");  
    this.Close();  
}  
}  
}
```

## ДОДАТОК Г

### Лістинг програмного коду. Основна форма безпеки №4

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WoL_IMIT
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void Form4_Load(object sender, EventArgs e)
        {
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }

        private int attemptCount = 0; // Variable for counting login attempts
        private void button1_Click(object sender, EventArgs e) //Button for login
        {
            string username = textBox3.Text;
            string password = textBox4.Text;

            if (ValidateCredentials(username, password))
```



```

    {
        // If credentials are valid, open the main form
        Form1 mainForm = new Form1();
        mainForm.Show();
        this.Hide(); // Hide the login form
    }
    else
    {
        attemptCount++; // Increase the attempt counter

        // If the number of attempts is more than 3, close the program
        if (attemptCount >= 3)
        {
            MessageBox.Show("Too many failed attempts. The program will now
close.");
            Application.Exit(); // Closes the program
        }
        else
        {
            //Show error message if credentials are incorrect
            MessageBox.Show("Invalid username or password. Please try
again.");
        }
    }
}

private bool ValidateCredentials(string username, string password)
{
    //user_1 data
    return username == "1" && password == "1";
}

private void textBox3_TextChanged(object sender, EventArgs e) //User name
{

}

private void textBox4_TextChanged(object sender, EventArgs e) //Password
{

}

private void groupBox1_Enter(object sender, EventArgs e)
{

```

```

    }

    private void LoadImageIntoPictureBox()
    {
        try
        {
            //шлях до зображення
            string image1 = @"login_background.png";
            //завантаження зображення
            pictureBox1.Image = LoadImage(image1);
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Сталася помилка при завантаженні зображення: {ex.Message}");
        }
    }

    private Image LoadImage(string imagePath)
    {
        // Перевіряємо, чи файл існує
        if (System.IO.File.Exists(imagePath))
        {
            // Завантажуємо зображення з файлу
            return Image.FromFile(imagePath);
        }
        else
        {
            MessageBox.Show($"Файл зображення '{imagePath}' не існує");
            return null;
        }
    }
}

```

## ДОДАТОК Д

### Стислий опис файлів програмного додатку

**WoL\_exe.exe** – основний модуль для роботи додатка.

**WoL.exe** – ярлик (посилання) для зручного запуску додатка.

**MAC-адреси.txt** – текстовий документ для збереження даних про комп'ютери. Містить назви та MAC-адреси персональних комп'ютерів.

**FonMain.png, форма2.png, icon.ico, login\_background.png**– файли зображень, потрібні для кращого виду та візуального сприйняття програми.

**WoL\_exe.deps.json, WoL\_exe.dll, WoL\_exe.pdb, WoL\_exe.runtimeconfig.json** – другорядні модулі для функціонування роботи додатка.