Таблиця 1 Загальний опис класів та їх призначення.

| **Клас** | **Призначення** |
| --- | --- |
| SystemCleaner | Основний клас, що відповідає за пошук та видалення тимчасових файлів, кешу, журналів, кошика та інших непотрібних файлів у системі. |
| FileScanner | Відповідає за сканування системи та виявлення файлів, які можуть бути безпечно видалені. Працює з каталогами, розширеннями та правилами очищення. |
| CleanerConfig | Зберігає налаштування користувача щодо того, які типи файлів і директорій слід сканувати або ігнорувати під час очищення. |
| ScheduleManager | |  | | --- | |  |  |  | | --- | | Керує автоматичним запуском очищення за розкладом. Дозволяє налаштовувати час, частоту та умови запуску. | |
| UIManager | |  | | --- | |  |  |  | | --- | | Відповідає за взаємодію з користувачем: консольний або графічний інтерфейс, виведення результатів очищення, запит дозволу на видалення. | |
| LogManager | Веде журнал дій: які файли були видалені, коли проводилось очищення, які помилки виникли. Створює текстові звіти. |
| SafetyManager | Перевіряє, чи файли, що підлягають видаленню, не є критичними для системи. Забезпечує безпечне очищення. |
| RestorePointManager | Створює точку відновлення системи перед очищенням, щоб мати можливість повернутись у разі проблем після видалення. |

1.Діаграма класів показує структуру програми очищення системи, визначаючи основні класи, такі як SystemCleaner, FileScanner, CleanerConfig, ScheduleManager, UIManager, LogManager, SafetyManager та RestorePointManager, а також їх методи та взаємозв'язки. Ця діаграма дозволяє зрозуміти, як компоненти взаємодіють між собою і які функції вони виконують у межах програми.

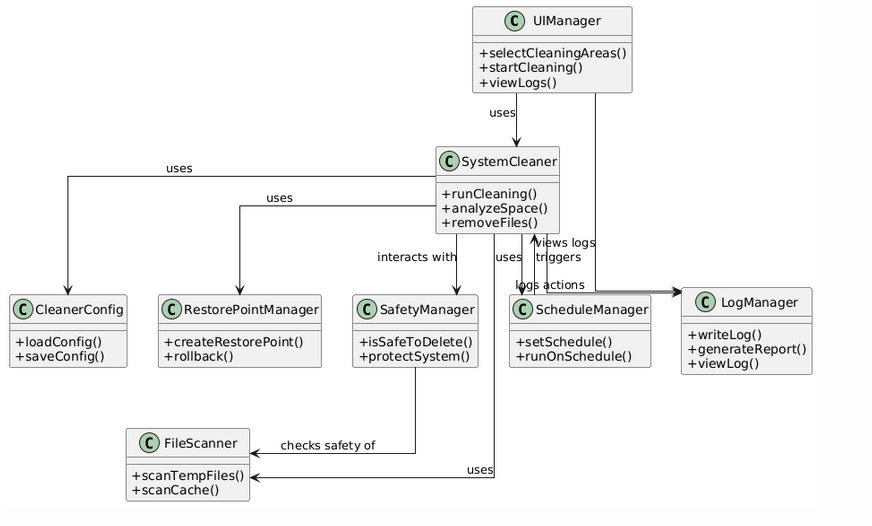


Рис. 1. Діаграма класів.

· **UIManager**: Клас для взаємодії з користувачем. Дозволяє вибирати файли та папки для резервного копіювання, переглядати звіти і журнал операцій.

· **BackupManager**: Клас, що відповідає за створення і відновлення резервних копій. Також здійснює перевірку вільного місця на диску та видаляє застарілі копії.

· **Encryptor**: Клас для шифрування та розшифрування файлів перед збереженням у резервних копіях або після відновлення. Забезпечує захист даних за допомогою пароля.

· **Cleaner**: Клас для перевірки доступного вільного місця на диску та автоматичного видалення старих резервних копій.

· **LogManager**: Клас, який веде журнал дій програми, фіксує результати резервного копіювання, помилки та генерує текстові звіти.

· **Scheduler**: Клас для налаштування та автоматизації процесу резервного копіювання за розкладом. Дозволяє запускати копіювання у визначений час.

2. Структурна схема.

**Структурна схема програми** відображає загальний потік роботи утиліти очищення системи. Вона демонструє, як користувач запускає програму, вибирає області для очищення (тимчасові файли, кеші, журнали), налаштовує автоматичне очищення, а також взаємодіє з іншими функціональними модулями, такими як перевірка безпечності файлів та створення звітів.

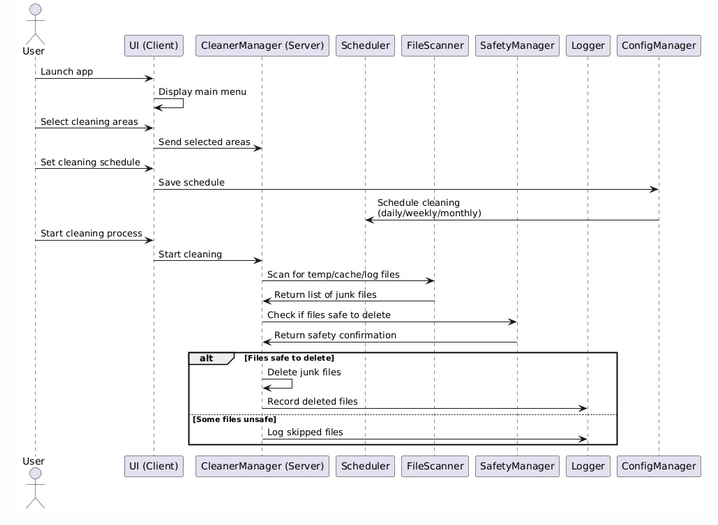


Рис. 2. Структурна схема програми.

* **User:** Користувач, який запускає програму та ініціює процес очищення через інтерфейс.
* **UIManager:** Відповідає за взаємодію з користувачем — відображає меню, приймає вибір зон для очищення, передає дані в інші модулі.
* **Cleaner:** Основний модуль очищення. Шукає непотрібні файли, видаляє їх після перевірки.
* **FileScanner:** Проводить сканування системи для виявлення тимчасових файлів, кешів, логів тощо.
* **SafetyManager:** Перевіряє, чи можна безпечно видалити знайдені файли.
* **LogManager:** Веде журнал усіх дій, створює звіти після сесії очищення.
* **Scheduler:** Дозволяє користувачу налаштувати автоматичне очищення (щодня, щотижня тощо).
* **ConfigManager:** Зберігає обрані користувачем налаштування та передає їх у Scheduler.

### Взаємодія:

* Користувач через UIManager обирає тип очищення або налаштовує автоматичне.
* Cleaner взаємодіє з FileScanner для пошуку непотрібних файлів, а також із SafetyManager для перевірки безпеки видалення.
* LogManager зберігає інформацію про очищення.
* Scheduler працює на основі налаштувань, отриманих через ConfigManager, та запускає очищення у заданий час.

3. Діаграма послідовності для резервного копіювання.

Діаграма послідовності описує покрокову взаємодію об'єктів програми в конкретних сценаріях, таких як резервне копіювання, шифрування файлів або відновлення з резервної копії, і вказує, коли і які функції викликаються в реальному часі.

Як користувач взаємодіє з системою резервного копіювання:

1. **Запуск програми** — користувач відкриває утиліту, з’являється головне меню.
2. **Очищення вручну:**

* Користувач обирає, які зони очищати.
* Система сканує на наявність тимчасових/непотрібних файлів.
* Перевіряється безпека видалення.
* Файли видаляються, а дії записуються в журнал.

3. **Автоматичне очищення:**

* Користувач налаштовує розклад.
* Конфігурація зберігається та Scheduler активує очищення у заданий час.

4. **Перегляд результатів:**

* Користувач переглядає звіт про очищення.

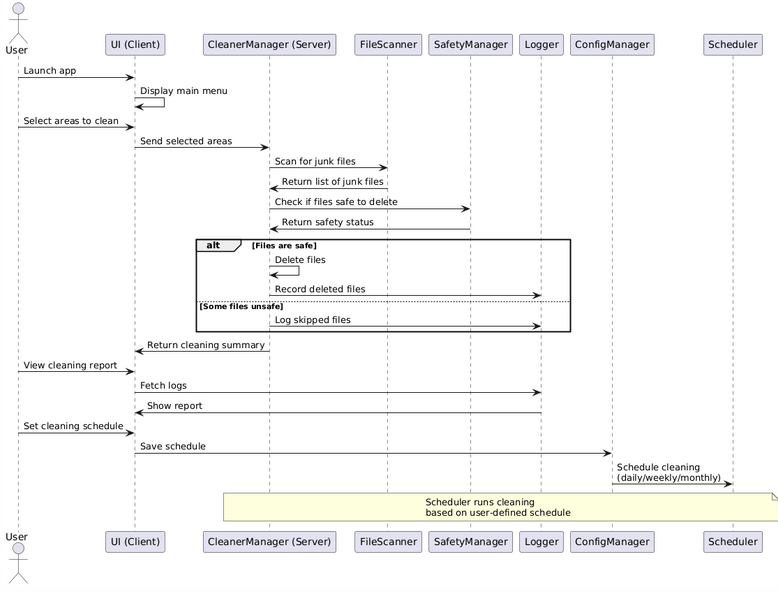


Рис. 3. Діаграма послідовності для резервного копіювання

4. Діаграма варіантів застосування.

**Діаграма варіантів застосування ілюструє основні сценарії використання програми користувачем**, такі як вибір областей для очищення, запуск процесу видалення тимчасових і непотрібних файлів, перегляд логів очищення, планування автоматичної очистки, що дозволяє чітко визначити вимоги до функціоналу системи.

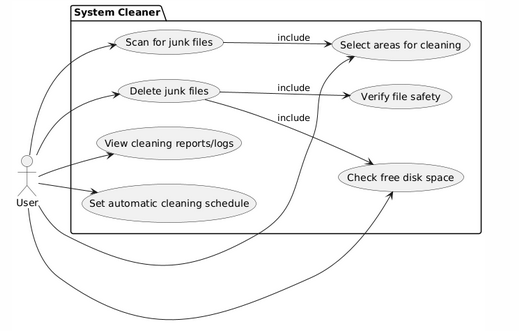


Рис. 4. Діаграма варіантів застосування.

#### **Основні компоненти:**

**UIManager** – інтерфейс користувача, який керує взаємодією з системою.

**CleanerManager** – логіка виявлення та видалення непотрібних файлів

**Scheduler** – модуль автоматизації запуску очищення.

**SafetyManager** – перевірка безпеки файлів перед видаленням.

**LogManager** – запис журналу операцій, перегляд логів користувачем.

**ConfigManager** – керує збереженням налаштувань користувача.

### Зовнішні ресурси:

**FileSystem** — доступ до файлової системи користувача для видалення файлів.

**RecycleBinAPI** — модуль для керування кошиком (видалення з можливістю відновлення).

**WindowsCleanerAPI** — системні API Windows для доступу до тимчасових файлів, кешу браузерів тощо.

5. Діаграма компонентів програми.

**Діаграма компонентів** ілюструє фізичну структуру програми для очищення системи від непотрібних файлів. Вона показує, як основні компоненти утиліти взаємодіють між собою та із зовнішніми ресурсами Windows. Ця діаграма дозволяє побачити модульність системи, що полегшує її супровід і розширення.

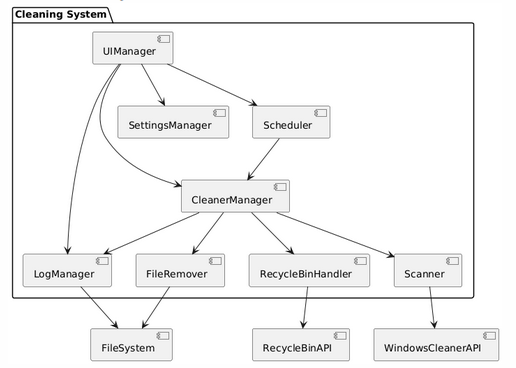


Рис. 5. Діаграма компонентів програми.

#### **1**. **Сканування та очищення (Cleaning Operations)**

* **CleanerManager**: Головний модуль очищення. Координує процес сканування системи, визначає непотрібні файли та передає їх до FileSystem для видалення.
* **Scanner**: Виявляє тимчасові, кешовані та інші непотрібні файли.
* **FileRemover**: Виконує фізичне видалення знайдених файлів із файлової системи.
* **RecycleBinHandler**: Модуль, який дозволяє тимчасове видалення з можливістю відновлення (через кошик).

#### 2. **Інтерфейс користувача (User Interface)**

**UIManager**: Забезпечує інтерфейс для взаємодії з користувачем. Дозволяє запускати сканування, переглядати результати, керувати файлами для видалення.

#### 3. **Журнал і налаштування (Logging & Settings)**

* **LogManager**: Фіксує усі дії програми: які файли були видалені, які ігноровані, які очищення відбулися.
* **SettingsManager**: Зберігає параметри користувача — що сканувати, які папки виключити тощо.
* **Scheduler**: Працює з планувальником Windows для запуску очищення за розкладом (наприклад, щотижня).

### Взаємодія між компонентами:

* **UIManager** взаємодіє з **CleanerManager**, **Scheduler**, **LogManager**.
* **CleanerManager** викликає **Scanner** для сканування, а потім **FileRemover** для видалення.
* **RecycleBinHandler** забезпечує опцію безпечного видалення.
* **Scheduler** запускає **CleanerManager** за заданим розкладом.
* **LogManager** записує всі події, пов’язані з очищенням.