



Міністерство освіти і науки України  
Національний технічний університет України “Київський  
політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №2  
**Технологія розроблення програмного забезпечення**  
**«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ**  
**ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.**  
**КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»**  
Варіант 25

Виконала  
студентка групи ІА-24  
Мелешко Юлія Сергіївна

Перевірив:  
Мягкий М.Ю.

Київ 2024р.

**Тема:** Діаграма варіантів використання. Сценарії варіантів використання. Діаграми uml. Діаграми класів. Концептуальна модель системи

### **Теоретичні відомості Діаграма прецедентів (Use Case Diagram)**

Діаграма прецедентів використовується для візуалізації функціональних вимог до системи. Вона ілюструє, як користувачі (актори) взаємодіють із системою через конкретні сценарії використання (прецеденти). Основними елементами цієї діаграми є актори, прецеденти та зв'язки між ними. Прецеденти дозволяють виявити основні функції системи та забезпечити їхнє розуміння на високому рівні.

### **Діаграма класів (Class Diagram)**

Діаграма класів моделює статичну структуру системи. Вона демонструє класи, їх атрибути, методи (операції) та зв'язки між класами, такі як асоціації, агрегації, композиції та успадкування. Класи представляють основні компоненти системи, їхні характеристики (атрибути) та поведінку (методи), а зв'язки між ними показують, як ці класи взаємодіють. **База даних та її структура**

База даних — це організований набір інформації, що зберігається у структурованому вигляді, зазвичай у формі таблиць. Таблиці складаються з рядків (записів) і стовпців (полів), де поля містять атрибути даних. Структура бази даних визначає, як дані пов'язані між собою. Основні елементи бази даних включають таблиці, ключі (первинні та зовнішні), а також зв'язки між таблицями, які можуть бути одного з типів: один-доодного, один-до-багатьох або багато-до-багатьох.

### **Шаблон Репозиторію (Repository Pattern)**

Шаблон Репозиторію використовується для абстрагування доступу до даних. Він

дозволяє працювати з базою даних через клас-репозиторій, що інкапсулює операції зі

збереження, отримання, оновлення та видалення даних. Це знижує залежність між бізнес-логікою та логікою доступу до даних, роблячи систему більш гнучкою до змін або оновлень.

### **Теоретичні відомості Діаграма прецедентів (Use Case Diagram)**

Діаграма прецедентів використовується для візуалізації функціональних вимог до системи. Вона ілюструє, як користувачі (актори) взаємодіють із системою через конкретні сценарії використання (прецеденти). Основними елементами цієї діаграми є актори, прецеденти та зв'язки між ними.

Прецеденти дозволяють виявити основні функції системи та забезпечити їхнє розуміння на високому рівні.

### **Діаграма класів (Class Diagram)**

Діаграма класів моделює статичну структуру системи. Вона демонструє класи, їх атрибути, методи (операції) та зв'язки між класами, такі як асоціації, агрегації, композиції та успадкування. Класи представляють основні компоненти системи, їхні характеристики (атрибути) та поведінку (методи), а зв'язки між ними показують, як ці класи взаємодіють.

### **База даних та її структура**

База даних — це організований набір інформації, що зберігається у структурованому вигляді, зазвичай у формі таблиць. Таблиці складаються з рядків (записів) і стовпців (полів), де поля містять атрибути даних. Структура бази даних визначає, як дані пов'язані між собою. Основні елементи бази даних включають таблиці, ключі (первинні та зовнішні), а також зв'язки між таблицями, які можуть бути одного з типів: один-доодного, один-до-багатьох або багато-до-багатьох.

### **Шаблон Репозиторію (Repository Pattern)**

Шаблон Репозиторію використовується для абстрагування доступу до даних. Він дозволяє працювати з базою даних через клас-репозиторій, що інкапсулює операції зі збереження, отримання, оновлення та видалення даних. Це знижує залежність між бізнес-логікою та логікою доступу до даних, роблячи систему більш гнучкою до змін або оновлень.

# Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт. Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

## **..25 Installer generator (iterator, builder, factory method, bridge, interpreter, client-server)**

Генератор інсталяційних пакетів повинен мати якийсь спосіб налаштування файлів, що входять в установку, установки вікон з інтерактивними можливостями (галочка - створити ярлик на робочому столі; ввести в текстове поле деякі дані, наприклад, ліцензійний ключ і т.д.). Генератор повинен вивести один файл .exe або .msi.

Посилання на код:

<https://github.com/jlmp3mp44/trpz/tree/lab2>

## Хід роботи

### Діаграма прецендентів:

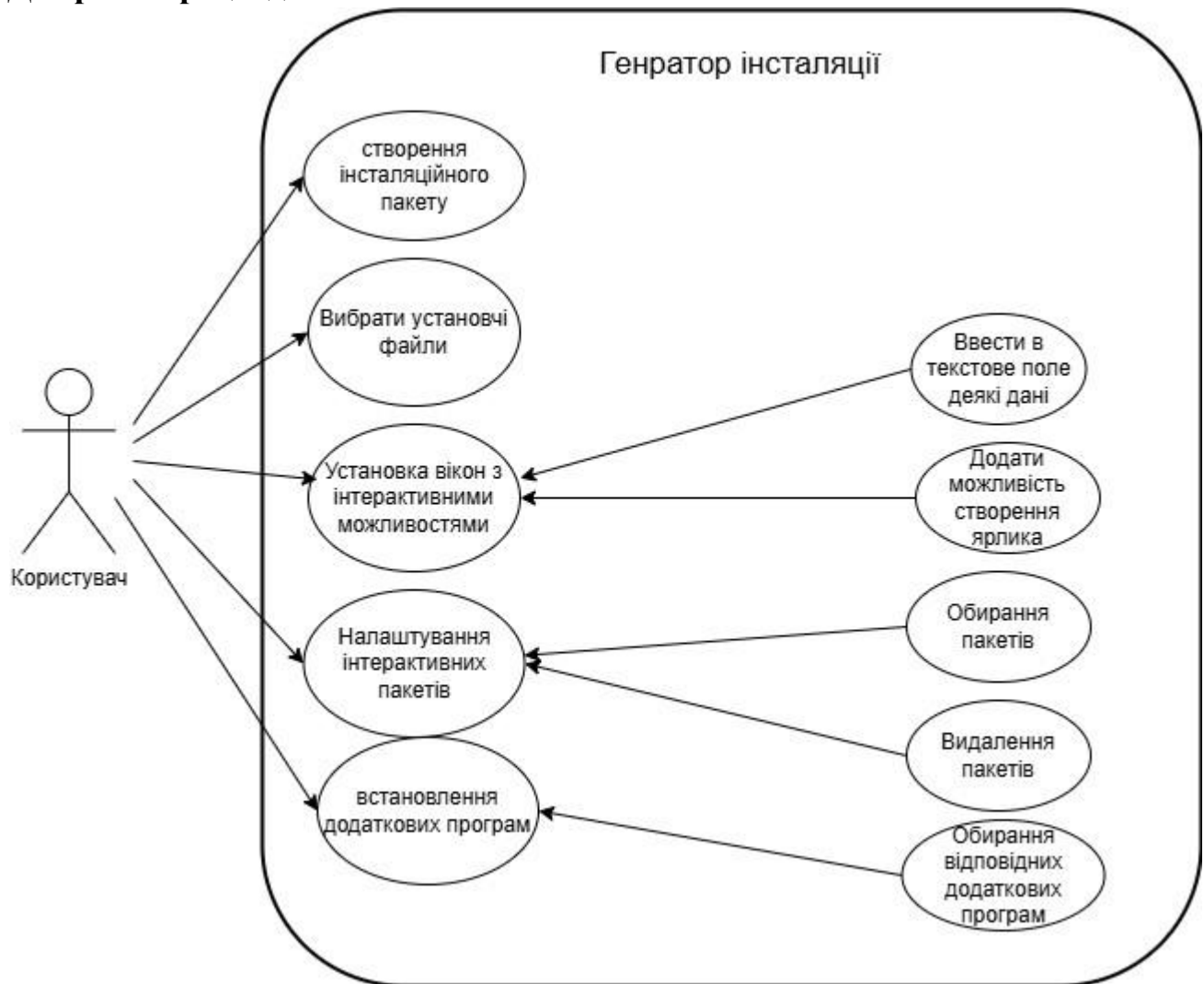


Рисунок 1. Діаграма прецендентів

Користувач запускає генератор інсталяції він може:

- створити інсталяційний пакет (діє інструкції по інсталяції)
- вибрати установчі файли (клікає, які хоче щоб були встановленні файли) також тут він може ввести ліцензійний код продукта, та може натиснути автоматичного створення ярлика
- налаштувати інтерактивні пакети (вибрати пакети, які хоче, обрати пакети за замовчуванням, видалити якісь)

- встановити додаткові файли/програми (натиснути на додаткові файли, які хоче завантажити)

## **Преценденти використання**

### **Прецендент 1: Вибір установчих файлів** Передумови:

- Користувач запустив генератор інсталяції.
- Користувач має доступ до каталогу з установчими файлами.

#### Післяумови:

- Користувач вибрав необхідні файли для інсталяції.
- Вибрані файли збережено для подальшого використання під час інсталяції.

#### Основний хід подій:

1. Користувач відкриває інтерфейс вибору установчих файлів.
2. Користувач переглядає доступний список установчих файлів.
3. Користувач обирає необхідні файли, які бажає встановити.
4. Система підтверджує вибір і зберігає його для подальшого застосування.

#### Виключення:

- Якщо каталог з установчими файлами порожній, система повідомляє про відсутність файлів і пропонує вибрати інший каталог.
- Якщо вибраний файл пошкоджений або недоступний, система видає помилку та просить користувача вибрати інший файл.

### **Прецендент 2:**

#### **Налаштування інтерактивних пакетів** Передумови:

- Користувач запустив генератор інсталяції.
- Користувач має доступ до списку доступних інтерактивних пакетів.

#### Післяумови:

- Вибрані інтерактивні пакети налаштовані відповідно до вибору користувача.
- Пакети збережені для подальшої інсталяції або видалення.

Основний хід подій:

1. Користувач відкриває інтерфейс налаштування інтерактивних пакетів.
2. Користувач обирає пакети, які він хоче встановити.
3. Користувач видаляє непотрібні пакети зі списку.
4. Система підтверджує вибір і готує пакети для подальшої інсталяції або видалення.

Виключення:

- Якщо користувач не вибрав жодного пакета, система видає попередження про необхідність вибору хоча б одного пакета для продовження.
- Якщо обраний пакет несумісний із системою, система видає повідомлення

про помилку та пропонує вибрати інший пакет. **Прецедент 3: Установка додаткових програм** Передумови:

- Користувач запустив генератор інсталяції.
- Користувач має доступ до списку додаткових програм.

Післяумови:

- Обрані додаткові програми встановлені на комп'ютер користувача.
- Програми з'явилися у відповідному розділі меню після встановлення.

Основний хід подій:

1. Користувач відкриває інтерфейс для установки додаткових програм.
2. Користувач обирає програми, які він хоче додати під час інсталяції.
3. Користувач підтверджує вибір.
4. Система інсталює обрані програми на пристрій користувача.

Виключення:

- Якщо обрана програма вже встановлена, система повідомляє про це користувача та пропонує вибрати іншу програму.
- Якщо під час інсталяції виникла помилка, система видає повідомлення про помилку та припиняє інсталяцію обраних програм.

Діаграма класів:

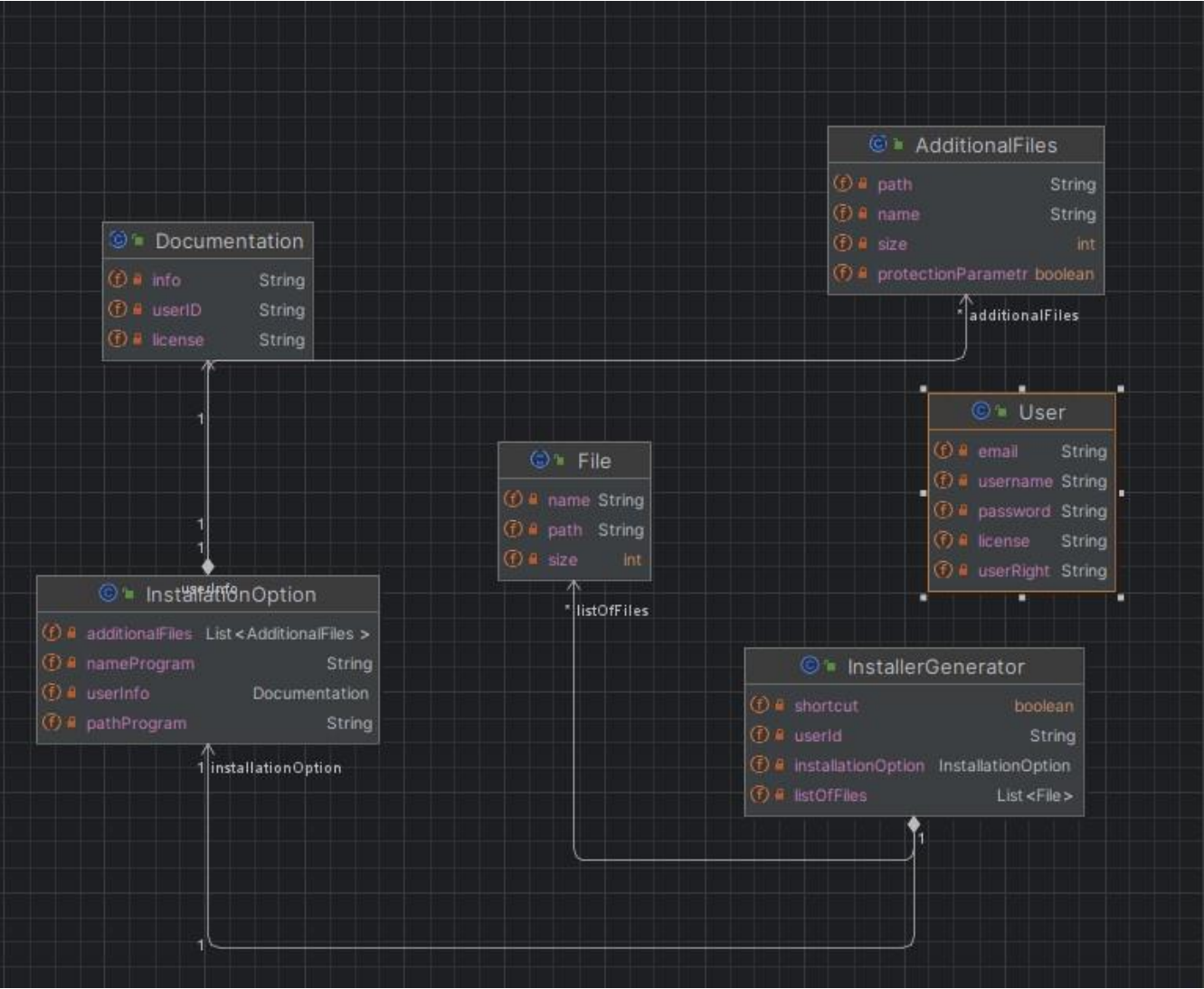


Рисунок 2. Діаграма класів



## Структура бази даних

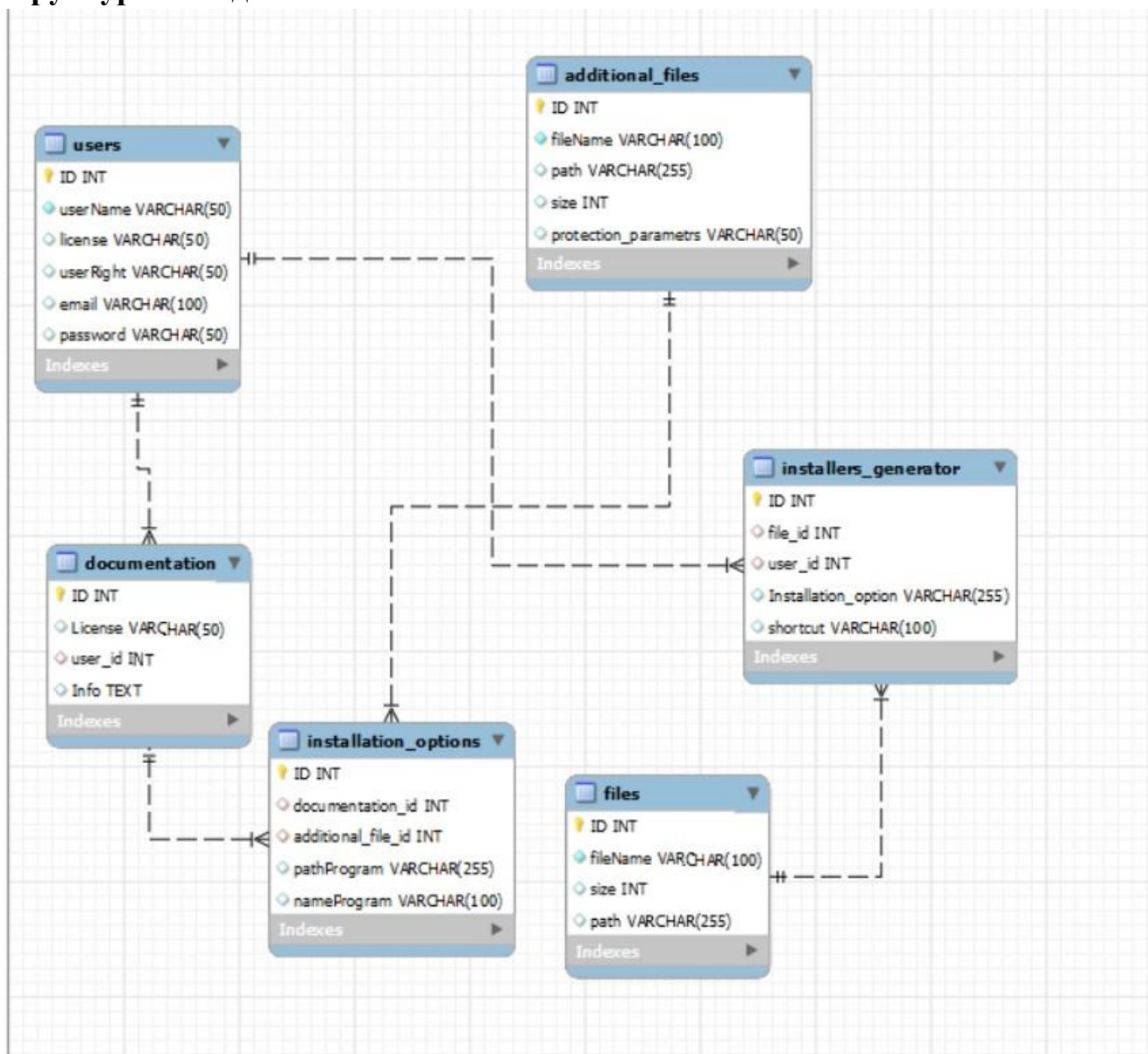


Рисунок 3. Структура бази даних

**Висновок :**Отже, у ході виконання лабораторної роботи було проведено ознайомлення з теоретичними відомостями та розроблено діаграму прецедентів, для розуміння основних сценаріїв взаємодії користувача із системою, та діаграму класів для системи керування завданнями, для розуміння структури системи,

взаємозв'язки між об'єктами. Розробила структуру бази даних, яка буде відповідати за зберігання і організацію даних для роботи системи.