

## Лабораторна робота №3

### Аналіз структури програмної системи та зворотне проєктування

**Мета:** навчитися виконувати початковий аналіз програмного проєкту, виявляти його структуру, виділяти компоненти та будувати UML-діаграми для кращого розуміння архітектури.

#### *Література*

*Bass L., Clements P., Kazman R. Software Architecture in Practice (SEI Series in Software Engineering). Print2print, 2022. 442 с. ISBN 978-0-13-688609-9.*

### Зміст роботи

#### **Завдання:**

1. Обрати проєкт з відкритим кодом (не більше 5000 рядків).  
Завантажити архів ZIP або клонувати репозиторій:

*git clone https://github.com/ім'я\_користувача/назва\_проєкту.git*

2. Запустити систему в середовищі розробки, ознайомитися з її функціональністю. Коротко описати: мову програмування; призначення програми; основні модулі та можливості.

3. Ознайомитися зі структурою каталогів (src, test, config тощо). Зробити скріншот і пояснити призначення основних папок.

4. Переглянути вихідний код. Скласти список основних класів і модулів (назва, роль). Зробити короткий опис (2–3 речення для кожного модуля).

5. Використовуючи інструмент для візуалізації (draw.io / PlantUML тощо) побудувати UML-діаграму класів: відобразити класи, їх атрибути та методи; показати зв'язки між ними (асоціація, агрегація, наслідування).

6. Побудувати UML-діаграму компонентів: відобразити основні модулі та їх залежності.

7. Провести аналіз проблемних місць програмної системи. На основі діаграм визначити: надмірні залежності між класами; дублювання функціоналу; «божественні класи» (класи з надто великою відповідальністю). Зробити висновки про сильні та слабкі сторони архітектури.