



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 №7**

«ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE  
METHOD»»

Варіант 23

Виконала студентка групи ІА–13:

Сиваченко Дар'я Євгенівна

Перевірив :

Мягкий Михайло Юрійович

Київ 2023

## Тема: Project Management software

### Хід роботи

Клас `ProjectManagementFacade``, представляє собою реалізацію шаблону проектування "FACADE". Цей шаблон використовується для надання простого інтерфейсу до складних систем або підсистем, спрощуючи їх використання. У нашому випадку, `ProjectManagementFacade`` використовується для управління різними аспектами системи управління проектами, такими як проекти, завдання, користувачі і ресурси, надаючи методи для роботи з ними через простий інтерфейс. Наприклад, методи такі як `addProject``, `addUser``, `addResource`` дозволяють додавати нові об'єкти проектів, користувачів та ресурсів до відповідних списків у `ProjectManagementFacade``. Методи, які взаємодіють з проектами (`assignManagerToProject``, `addTaskToProject``), завданнями (`assignTaskToUser``, `updateTaskStatus``) та користувачами (`allocateResourceToUser``), надають можливість виконувати дії, такі як призначення менеджера проекту, додавання завдань до проекту, призначення завдань користувачеві, оновлення статусу завдання та виділення ресурсів користувачеві. За допомогою цього шаблону можна здійснювати необхідні операції з управлінням проектами та пов'язаними з ними елементами, приховуючи складність внутрішньої логіки реалізації цих операцій. Такий підхід дозволяє спростити використання функціональності системи управління проектами для зовнішніх користувачів чи інших частин програми. Код до даної лабораторної буде надано разом зі звітом.

**Висновок:** У рамках даної лабораторної роботи було розглянуто структуру та взаємодію класів для системи управління проектами. Було реалізовано класи `Manager`, `Project`, `Task`, `Resource`, `User`, `Comment`, а також використано enum класи `TaskStatus` і `UserRole`, що представляють ключові об'єкти та статуси у системі. Для спрощення взаємодії з цими об'єктами було використано шаблон проектування "FACADE". Клас `ProjectManagementFacade` був реалізований для надання простого інтерфейсу для взаємодії з різними аспектами управління проектами, включаючи додавання/видалення проектів, користувачів, ресурсів, а також управління завданнями та користувачами. Кожен клас мав свою функціональність: `Project` зберігав інформацію про проекти та їх параметри, `Task` відслідковував завдання та їх статуси, `User` представляв користувачів системи, а `Manager` був відповідальним за керування проектами та їх завданнями.

Ця лабораторна робота показала, як за допомогою шаблону "FACADE" можна спростити використання складних систем класів, приховавши деталі їх внутрішньої реалізації за простим інтерфейсом, що дозволяє більш зручно та ефективно управляти системою управління проектами.

### **Відповіді на контрольні запитання**

#### **1. Розкажіть про KISS, DRY, YAGNI, закон Парето.**

- **KISS (Keep It Simple, Stupid):** Принцип KISS вказує на те, що програмне забезпечення повинне бути максимально простим у своїй конструкції. Це означає, що код має бути легким для розуміння, читання та підтримки. Замість складних та заплутаних рішень рекомендується використовувати прості та зрозумілі конструкції.
- **DRY (Don't Repeat Yourself):** Принцип DRY підкреслює необхідність уникати повторення коду. Це означає, що логіка та функціональність програми повинні бути написані лише один раз та використовуватися усюди там, де це потрібно, замість копіювання та вставляння фрагментів коду.
- **YAGNI (You Ain't Gonna Need It):** Принцип YAGNI закликає уникати реалізації функціональності, яка не є наразі потрібною. Він підкреслює важливість уникнення додавання зайвого функціоналу чи можливостей до програми до того моменту, коли це дійсно необхідно. Це дозволяє уникнути зайвого коду та складнощів у майбутньому.
- **Закон Парето (Принцип 80/20):** Закон Парето, також відомий як принцип 80/20, стверджує, що більшість результатів (приблизно 80%) мається в результаті невеликої частини діяльності (приблизно 20%). У розробці програмного забезпечення це може означати, що більшість функцій або витрат припадає на обмежену частину функціоналу. Таким чином, цей принцип нагадує про важливість спрощення та упорядкування функціоналу, фокусуючись на найважливіших задачах.

#### **2. Навіщо використовується шаблон «медіатор» ?**

Шаблон проектування "Mediator" (Медіатор) використовується для спрощення взаємодії між об'єктами, які взаємодіють між собою. Головна мета цього шаблону - уникнути прямих взаємозв'язків між різними класами, замість цього вводячи посередника, який би координував взаємодію між цими класами.

#### **3. Чим відрізняється шаблон «шаблонний метод» від «фабричного методу» ?**

В основному відмінність полягає у тому, що "шаблонний метод" визначає структуру алгоритму, дозволяючи підкласам перевизначати окремі його частини, тоді як "фабричний метод" визначає спосіб створення об'єктів у підкласах.

#### **4. Яку функціональність додає шаблон «міст»?**

Шаблон проектування "Міст" (Bridge) додає в програму можливість розділення абстракції від реалізації таким чином, що обидві частини можуть змінюватись незалежно одна від одної. Шаблон "Міст" вносить важливу функціональність для розробки програмного забезпечення, спрощуючи управління залежностями та розширення програми. Він дозволяє відокремити абстракцію від конкретної реалізації, що полегшує зміни та підтримку коду.

#### **5. Навіщо застосовується шаблон «фасад»?**

Шаблон проектування "Фасад" (Facade) застосовується для надання простого інтерфейсу до складної системи або підсистеми, приховуючи деталі роботи внутрішніх компонентів. Ось основні причини застосування шаблону "Фасад": спрощення використання складних систем, зменшення залежностей, підвищення рівня абстракції, покращення зручності обслуговування та розширюваності.