



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 №3

*«ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY», «STATE»,
«STRATEGY»»*

Варіант 23

Виконала студентка групи ІА–13:

Сиваченко Дар'я Євгенівна

Перевірив :

Мягкий Михайло Юрійович

Київ 2023

Тема: Project Management software

Хід роботи

В даній лабораторній роботі були представлені такі шаблони, як: «SINGLETON», «ITERATOR», «PROXY», «STATE», «STRATEGY». Я обрала для реалізації шаблон «STRATEGY» для реалізації зміни статусу завдання в класі Task:

Інтерфейс TaskStatusUpdateStrategy: Це інтерфейс, який визначає метод updateStatus(Task task). Він слугує узагальненим способом оновлення статусу завдання.

Реалізація стратегій:

DefaultTaskStatusUpdateStrategy: Реалізує інтерфейс TaskStatusUpdateStrategy і має внутрішню логіку оновлення статусу для стандартного сценарію.

UrgentTaskStatusUpdateStrategy: Інша реалізація інтерфейсу, яка має внутрішню логіку оновлення статусу для термінових завдань.

Клас Task: У цьому класі є методи для оновлення статусу завдання та встановлення стратегії. Він має поле status, що відображає поточний статус завдання.

Метод updateTaskStatus(TaskStatus newStatus) встановлює новий статус завдання.

Метод setStatusUpdateStrategy(TaskStatusUpdateStrategy statusUpdateStrategy) приймає реалізацію інтерфейсу стратегії і викликає метод updateStatus, передаючи йому поточний об'єкт завдання. Цей метод дозволяє динамічно змінювати стратегію оновлення статусу.

Основна ідея використання шаблону "Стратегія" полягає в тому, що можна динамічно змінювати алгоритм оновлення статусу без прив'язки до конкретних реалізацій. Це дає гнучкість у виборі поведінки об'єкта Task відповідно до поточних вимог або умов. Вибір конкретної стратегії передається через метод setStatusUpdateStrategy, де можна встановити потрібну логіку для оновлення статусу. Цей підхід робить код більш масштабованим і легким у зміні, оскільки нові стратегії можуть бути додані чи видалені без зміни основної логіки класу Task.

Реалізація даного шаблону буде надана до звіту

Висновок

У цій лабораторній роботі були створені різні класи, які моделюють систему керування завданнями. Кожен клас має свої унікальні характеристики та методи для взаємодії з іншими класами. Взаємодія між об'єктами моделюється через взаємні відносини, наприклад, завдання може мати коментарі, користувачі можуть мати завдання та здійснювати певні операції. У реалізації також були використані шаблони проектування. Був розглянутий шаблон "Стратегія", який дозволяє динамічно змінювати стратегію оновлення статусу завдання. Деякі методи та логіка у класах залишилися нереалізованими з метою показати загальну структуру програми. Ці класи можна доробити, додавши необхідну функціональність. Також існують можливості для вдосконалення та додаткових оптимізацій у системі. Лабораторна робота демонструє потенціал створення класів для моделювання системи та використання патернів проектування для покращення структури програми.