

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Навчально-науковий інститут комп'ютерного моделювання,  
прикладної фізики та математики

Звіт

З лабораторної роботи №8

З дисципліни:“ Ефективність та якість архітектурних рішень  
інформаційних систем ”

На тему:

«Патерн проєктування Шаблонний метод (Template Method)»

Виконав:

студент групи ІКМ-М225В

Загорулько Віталій Олегович

Харків 2025

## **Мета роботи**

Метою даної лабораторної роботи є ґрунтовне вивчення та практичне засвоєння поведінкового патерна проєктування Шаблонний метод (Template Method), а також набуття навичок його застосування для визначення скелета алгоритму в супер класі, дозволяючи під класам перевизначати певні кроки, не змінюючи загальну структуру алгоритму.

## **Завдання**

У межах виконання лабораторної роботи необхідно виконати такі завдання:

1. Ознайомитися з теоретичними зasadами та основними ідеями патерна проєктування Шаблонний метод.
2. Створити структуру класів, яка демонструє реалізацію патерна Шаблонний метод для стандартизованого процесу оновлення сущностей (Товар, Користувач, Замовлення) через REST API.
3. Реалізувати загальний алгоритм оновлення, що складається з фіксованих кроків: отримання даних, валідація, формування запиту, формування відповіді.
4. За допомогою перевизначення специфічних кроків або хуків (Hooks) реалізувати відмінності дляожної сущності:
  - Товар: Сповіщення адміністратора про невдалу валідацію.
  - Користувач: Заборона зміни поля email.
  - Замовлення: Додавання JSON-подання сущності до відповіді.

## Теоретичні відомості та Опис Проєкту

Патерн Шаблонний метод (Template Method) — це поведінковий патерн проєктування, який визначає незмінний кістяк (скелет) алгоритму в абстрактному базовому класі, дозволяючи підкласам перевизначати конкретні деталі (кроки), не змінюючи при цьому загальну структуру алгоритму . Патерн гарантує, що загальна послідовність виконання кроків завжди дотримується.

У межах цієї лабораторної роботи було реалізовано уніфікований процес оновлення трьох різних сутностей (Товар, Користувач, Замовлення) через абстрактний клас AbstractEntityUpdater.

1. Шаблонний метод: Метод update() в AbstractEntityUpdater є фінальним і містить фіксовану послідовність кроків: getEntity validateData [Опціональний Хук: on Патерн проєктування Посередник (Mediator)ValidationFailure] [Хук: preSaveHook] saveEntity [Хук: postSaveHook]. Ця послідовність не може бути змінена підкласами.
2. Абстрактні кроки: Такі методи, як getEntity, validateData та saveEntity, є абстрактними і мають бути обов'язково реалізовані в конкретних класах: ProductUpdater, UserUpdater, OrderUpdater.
3. Хуки (Hooks): Хуки — це опціональні кроки, які дозволяють підкласам "вклинигатися" в процес. Вони були використані для реалізації специфічних вимог:
  - onValidationFailure (Хук 1): Перевизначений у ProductUpdater для надсилання сповіщення адміністратору, коли валідація не пройшла.
  - preSaveHook (Хук 2): Перевизначений у UserUpdater для примусового видалення поля email зі вхідних даних перед збереженням, що обходить умовну логіку валідації.
  - postSaveHook (Хук 3): Перевизначений у OrderUpdater для модифікації стандартної відповіді, додаючи до неї JSON-подання оновленого замовлення.

Такий підхід забезпечив стандартизацію процесу оновлення (що є перевагою патерна), дозволяючи при цьому кожній сутності мати свою унікальну логіку у визначених точках.

Програмний код реалізації на Java розміщено у Додатку А.

## **Висновок**

Під час виконання лабораторної роботи було успішно вивчено та реалізовано поведінковий патерн Шаблонний метод. Головне завдання — уніфікація процесу оновлення сущностей при збереженні можливості реалізації специфічних кроків — було досягнуто шляхом створення абстрактного класу `AbstractEntityUpdater`.

Шаблонний метод `update()` гарантує фіксовану послідовність виконання кроків для всіх сущностей. Використання хуків дозволило реалізувати унікальні вимоги для Товарів, Користувачів та Замовлень, не порушуючи загальну структуру.

Це підтверджує, що патерн Шаблонний метод є ефективним інструментом для проєктування архітектур, де існує спільний алгоритм з різними варіаціями його частин, забезпечуючи високий рівень повторного використання коду та дотримання принципу інверсії залежностей (DIP).

## **Додаток А**



**Lab8.java**