

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 №10

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

На тему:

«Патерн проєктування Посередник (Mediator)»

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 Посередник (Mediator), а також набуття навичок його застосування для зменшення складності взаємодії між набором класів, замінюючи прямі зв'язки об'єктів на зв'язок через єдиний об'єкт-посередник.

Завдання

У межах виконання лабораторної роботи необхідно виконати такі завдання:

1. Ознайомитися з теоретичними засадами та основними ідеями патерна проєктування Посередник.
2. Створити структуру класів, що моделює сторінку оформлення замовлення квітів.
3. Реалізувати взаємодію елементів форми за допомогою патерна Посередник (FlowerOrderMediator), який керує логікою:
 - Залежність доступних проміжків часу від обраної дати доставки.
 - Встановлення обов'язковості полів Ім'я/Телефон залежно від чекбоксу "Отримувач інша особа".
 - Деактивація всіх полів доставки при виборі самовивозу.
4. Створити структуру класів та методів, що вирішує описане завдання, та приклад клієнтського коду.

Теоретичні відомості та Опис Проєкту

Патерн Посередник (Mediator) — це поведінковий патерн проєктування, який допомагає зменшити зв'язність класів, інкапсулюючи складну взаємодію між набором об'єктів (Колег). Замість того, щоб Колеги посилалися один на одного, вони спілкуються виключно через єдиний об'єкт-посередник. Це централізує логіку взаємодії та спрощує підтримку системи.

У межах цієї лабораторної роботи було реалізовано динамічну сторінку оформлення замовлення квітів. Центральним елементом є клас `FlowerOrderMediator` (Конкретний Посередник), який керує всіма елементами форми (Колегами), такими як `DateSelector`, `SelfPickupCheckbox`, `RecipientCheckbox` тощо.

Ключова логіка взаємодії, вирішена за допомогою Посередника:

- Дата Час: При зміні стану `DateSelector` (подія `DateSelected`), він сповіщає Посередника. Посередник перевіряє логіку та викликає метод `setAvailableTimeSlots()` на `TimeSelector`.
- "Інша особа" Ім'я/Телефон: При зміні стану `RecipientCheckbox` (подія `RecipientChanged`), Посередник, керуючись станом цього чекбокса, встановлює видимість та обов'язковість полів `RecipientNameField` та `RecipientPhoneField`.
- Самовивіз Форма Доставки: При активації `SelfPickupCheckbox` (подія `PickupChanged`), Посередник деактивує або приховує всі компоненти, що стосуються доставки (`AddressField`, `CourierNotesField`, `DateSelector`, `TimeSelector`), демонструючи централізований контроль над великою кількістю взаємопов'язаних елементів.

Кожен елемент форми (Колега) містить лише посилання на Посередника і має простий метод `changed(String event)`, яким він сповіщає про свої зміни. Таким чином, Колеги не знають ні про існування один одного, ні про складну логіку, яка їх пов'язує, що є головною перевагою патерна Посередник.

Програмний код реалізації на Java розміщено у Додатку А.

Висновок

Під час виконання лабораторної роботи було успішно вивчено та реалізовано поведінковий патерн Посередник (Mediator). Завдання інкапсуляції складної логіки взаємодії елементів форми було вирішено шляхом створення єдиного координатора — FlowerOrderMediator.

Ця реалізація дозволила значно зменшити зв'язність між компонентами форми. Додавання нових елементів або зміна логіки взаємодії вимагатиме модифікації лише класу FlowerOrderMediator, не торкаючись при цьому кодів окремих компонентів. Це підтверджує ефективність патерна Посередник як засобу для керування складними системами, де багато об'єктів взаємодіють один з одним.

Додаток А



Lab10.java