МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп’ютерної інженерії та інформаційних систем

Лабораторна робота №8

# з дисципліни «Обєктно-орієнтовані технології програмування»

на тему:

«ІФасад (Facade). Замісник (Проксі). Міст (Bridge).»

Виконав:

студент 1 курсу, групи КІ2м-23-3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Юрко П.П.

(Підпис)

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лисенко С.М.

(Підпис)

Хмельницький - 2024

Третій рівень.

Фасад (Facade)

Паттерн "Фасад" (Facade) у коді (code1.cs) надає єдиний інтерфейс до групи інтерфейсів у підсистемі, що спрощує взаємодію з нею. У нашому випадку ми можемо використовувати паттерн "Фасад" для синхронізації зі складом, де знаходиться товар. Підсистема 1: InventorySystem (Управління запасами): UpdateInventory - оновлює кількість продуктів на складі, CheckInventory - перевіряє кількість продуктів на складі. Підсистема 2: OrderSystem (Управління замовленнями): CreateOrder - створює нове замовлення, CancelOrder - скасовує замовлення. Підсистема 3: ShippingSystem (Управління доставкою): ShipProduct - відправляє продукт.

Клас WarehouseFacade (Фасад для синхронізації зі складом): Інкапсулює взаємодію з трьома підсистемами (InventorySystem, OrderSystem, ShippingSystem). Метод SyncProduct перевіряє наявність продукту на складі, створює замовлення, оновлює кількість продукту на складі та організовує відправку. Клас Program: Створює об'єкт фасаду WarehouseFacade та викликає метод SyncProduct для синхронізації продукту. Цей паттерн спрощує взаємодію з підсистемами управління складом, замовленнями та доставкою, надаючи єдиний інтерфейс для користувачів системи. Це дозволяє уникнути складнощів при роботі з окремими підсистемами та зменшує зв'язок між компонентами системи.

Замісник (Проксі)

Паттерн "Замісник" (Proxy) у коді (code2.cs) забезпечує сурогат або замісник іншого об'єкта для контролю доступу до нього. У нашому випадку ми можемо використовувати паттерн "Замісник" для отримання інформації про те, скільки певного товару є на складі, з можливістю кешування або контролю доступу. Інтерфейс IInventory: Оголошує метод GetStock, який повертає кількість певного товару на складі. Клас RealInventory: Реалізує метод GetStock, який звертається до реального складу і повертає кількість товару; Використовує словник productStock для зберігання інформації про запаси товару. Клас InventoryProxy: Реалізує інтерфейс IInventory і виступає в ролі замісника для RealInventory; Використовує кеш для зберігання запитів про запаси товару, щоб зменшити кількість звернень до реального складу; Якщо інформація про запаси товару є в кеші, вона повертається з кешу. Якщо ні, здійснюється звернення до RealInventory, і результат зберігається в кеші. Клас Program: Створює об'єкт InventoryProxy; Отримує і виводить інформацію про запаси товару, показуючи, як працює кешування. Цей підхід дозволяє зменшити навантаження на реальний склад за допомогою кешування результатів запитів і забезпечує контрольований доступ до даних про запаси товару.

Міст (Bridge)

Паттерн "Міст" (Bridge) у коді (code3.cs) розділяє абстракцію і реалізацію, дозволяючи їм змінюватися незалежно. У випадку з перевіркою наявності товару на складі, ми можемо використовувати "Міст" для розділення логіки перевірки та відображення результатів. Інтерфейс IInventoryImplementation: Оголошує метод GetStock, який повертає кількість певного товару на складі. Клас RealInventoryImplementation: Реалізує інтерфейс IInventoryImplementation і представляє реальний склад; Метод GetStock звертається до реального складу і повертає кількість товару; Використовує словник productStock для зберігання інформації про запаси товару. Абстрактний клас Inventory: Містить захищене поле implementation, яке зберігає конкретну реалізацію інтерфейсу IInventoryImplementation; Оголошує абстрактний метод CheckStock, який буде реалізовано в підкласах. Клас InventoryChecker: Наслідує клас Inventory і реалізує метод CheckStock; Викликає метод GetStock реалізації і перевіряє кількість товару. Якщо товару немає, виводить повідомлення про те, що товар закінчився. Клас Program: Створює об'єкт RealInventoryImplementation і передає його в InventoryChecker; Викликає метод CheckStock для перевірки наявності різних товарів. Цей підхід дозволяє легко змінювати реалізації перевірки наявності товару незалежно від логіки відображення результатів, забезпечуючи гнучкість і розширюваність системи.

Посилання на репозиторій гітхаб: https://github.com/Laypsel/OOP/tree/main