

Лабораторна робота №4

Тема: «Введення в структурні патерни. Adapter, Bridge.»

Перш за все для того щоб зрозуміти про що йде мова нам потрібно визначити, що таке структурні шаблони. Отже, **структурні шаблони** – це шаблони проектування, в яких розглядається питання про те, як з класів і об'єктів створюються більш серйозніші структури.

Основним завданням структурних шаблонів є формування найбільш підходящої структури та взаємодії між класами для виконання певних завдань.

Структурні шаблони рівня класа використовують наслідування для співставлення композицій із інтерфейсів і реалізацій. Простий приклад – використання багаторазового наслідування для об'єднання декількох класів в один. В результаті виходить клас, який володіє властивостями всіх своїх батьків. Особливо корисний цей шаблон, коли потрібно організувати спільну роботу декількох незалежних бібліотек.

До структурних шаблонів належать:

- Адаптер (Adapter)
- Міст (Bridge)
- Компонувальник (Composite)
- Декоратор (Decorator)
- Фасад (Facade)
- Легковаговик (Flyweight)
- Проксі (Proxy)

В цій роботі ми розглянемо перші два шаблони, а саме: Адаптер (Adapter) та Міст (Bridge). Розпочнемо з шаблону Адаптер.

Adapter (Адаптер)

Адаптер, Adapter або Wrapper/Обгортка - структурний шаблон проектування, призначений для організації використання функцій об'єкта, недоступного для модифікації, через спеціально створений інтерфейс. Адаптер надає можливість користуватися об'єктом, який не є прийнятним у нашій системі і який не можна змінити. Ми адаптуємо його функціональність через інший, відомий нашій системі, інтерфейс.

Основне завдання

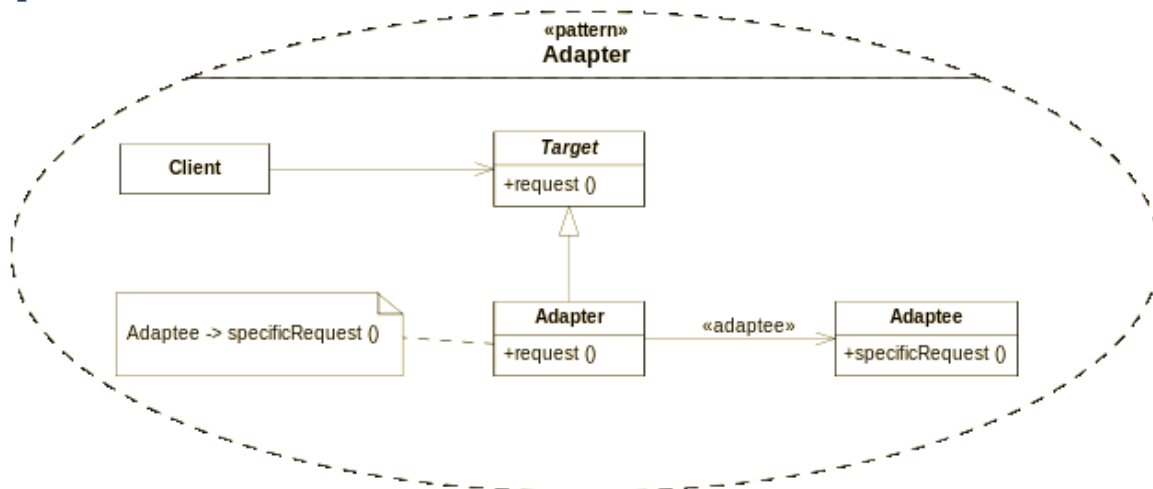
Система підтримує необхідні дані і поведінку, але має невідповідний інтерфейс. Частіше всього шаблон Адаптер застосовується, якщо необхідно

створити клас, похідний від знов обумовленого або уже існуючого абстрактного класу.

Спосіб вирішення

Адаптер передбачає створення класа-оболонки з потрібним інтерфейсом

UML діаграма:

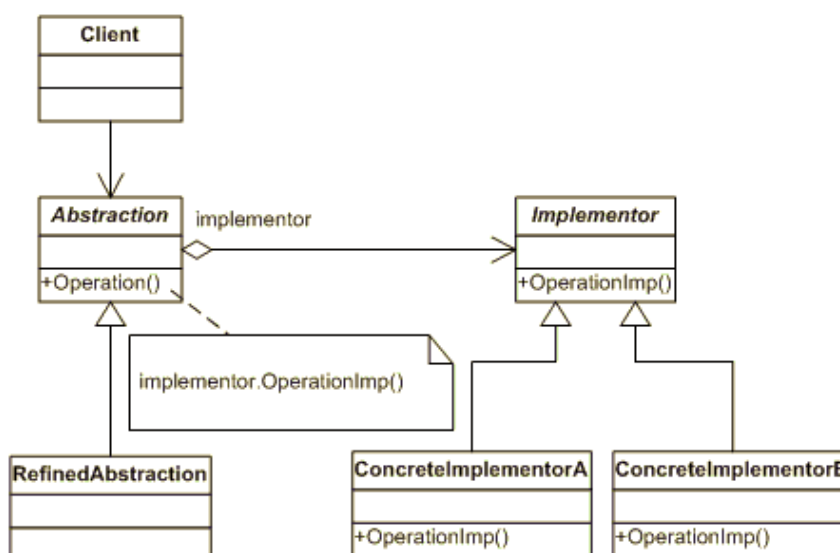


Реалізація

Включення уже існуючого класу в інший клас. Інтерфейс класу який включається приводиться уже з новими вимогами, а виклики його методів перетворюються в виклики методів включеного класу.

Bridge (Міст)

UML діаграма:



Bridge, Міст – шаблон проектування, який використовується в проектуванні програмного забезпечення, щоб розділяти абстракцію і реалізацію так, щоб вони могли змінюватися незалежно один від одного.

Індивідуальні завдання:

1. Написати програму БУДИЛЬНИК з використанням шаблону BRIDGE. Нехай інтерфейс будильника AlarmC(інтерфейс абстракції), вміє запускатися (start),зупинятися (stop), та будити (toWake). Інтерфейс реалізації (AlarmCImpl)описує два методи, а саме ring тобто дзвонити та notify-повідомити. Воочевидь метод toWake буде по черзі викликати ці методи.
2. Написати програму КАЛЬКУЛЯТОР для розрахунку цін заказу з використанням шаблону BRIDGE. Для опису товару в корзині покупця створимо клас ItemInBOX. Він вміщає код товару,кількість та ціну(Id, Quantity, Price). Інтерфейс абстракції буде визначатися потребами клієнта: додавати позиції товарів та отримувати ціну всього замовлення. В реалізацію буде винесено основні функції калькулятора: запит ціни одиниці товару та ціна доставки.
3. Написати програму ПРОКАТ СНОУБОРДІВ з використанням шаблону BRIDGE. Для опису наданих послуг створимо клас SkiMuneris. Він вміщає фірму сноубордів в нашому випадку може бути і код (Id), розмір черевик(shoeSSize) та напрямок розташування кріплень тобто на праву сторону чи на ліву(Direction). Інтерфейс абстракції буде визначатися потребами клієнта: додавати кількість сноубордів та отримувати ціну всього замовлення. В реалізацію буде винесено основні функції прокату: запит ціни одиниці товару та ціна налаштування спорядження під користувача.
4. Написати програму БУГХАЛТЕРІЯ з використанням шаблону BRIDGE. На підприємстві працюють люди різних професій, у кожного з них є свої характеристики : імя, посада тощо. За допомогою шаблону проектування створити структуру класів, яка б дозволяла видавати заробітню плату працівникам двома способами: за кількість відроблених годин та за зроблену роботу(відрядна форма).
5. Написати програму SKYPE з використанням шаблону ADAPTER. Маємо інтерфейс Phone, у якому об'явлений метод Make Call (Зробити Виклик). Зробити можливим використання відео камери (клас VideoCamera) при відео виклику.
6. Написати програму TRAIN з використанням шаблону ADAPTER. Отож, у нас є ноутбук. Припустимо вас кафедра ІТ НЛТУ України відправила у відрядження на міжнародну олімпіаду у м.Київ. Ви сідаєте у поїзд. І тут у вас виникає бажання повторити ООАП(Шаблони проектування) і включивши ноутбук у вас сідає батарея. На скільки ви знаєте Укрзалізниця оновлює свій парк вагонів, але це робиться дуже повільно. В одному з

вагонів у нас все сучасне, тобто новий вагон, тому NewVagonSystem має метод MatchSocket, яким ми просто можемо скористатися(заряджаєм ноутбук). В іншому вагоні у нас проблемки(вагон старий і старі радянські розетки є вузькі, тому штекер від ноутбука не підходить), тобто OldVagonSytem має тільки метод ThinSocket. Нажаль замінити розетки у вагоні ми не можемо. Натомість, ми можемо взяти адаптер, який надає можливість користуватися тією ж функціональністю споживання електричного струму, але із старої системи.