

ЛАБОРАТОРНА РОБОТА №3. СТРУКТУРНІ ШАБЛОНИ ПРОЕКТУВАННЯ. ШАБЛОНИ COMPOSITE, DECORATOR, PROXY

Мета: Ознайомлення з видами шаблонів проектування ПЗ. Вивчення структурних шаблонів. Отримання базових навичок з застосування шаблонів Composite, Decorator та Proxy.

Довідка

Composite

Проблема. Як обробляти атомарний об'єкт та композицію об'єктів однаково?

Рішення. Визначити класи для композитних (Composite) і атомарних (Leaf) об'єктів таким чином, щоб вони реалізовували один і той же інтерфейс (Component).

Decorator

Проблема. Покласти додаткові обов'язки (прозорі для клієнтів) на окремий об'єкт, а не на клас в цілому.

Рішення. Динамічно додати об'єкту нові обов'язки не вдаючись при цьому до породження підкласів (наслідування). "Component" визначає інтерфейс для об'єктів, на які можуть бути динамічно покладені додаткові обов'язки, "Concrete Component" визначає об'єкт, на який покладаються додаткові обов'язки, "Decorator" – зберігає посилання на об'єкт "Component" і визначає інтерфейс, відповідний інтерфейсу "Component". "Concrete Decorator" покладає додаткові обов'язки на компонент. "Decorator" переадресує запити об'єкту "Component".

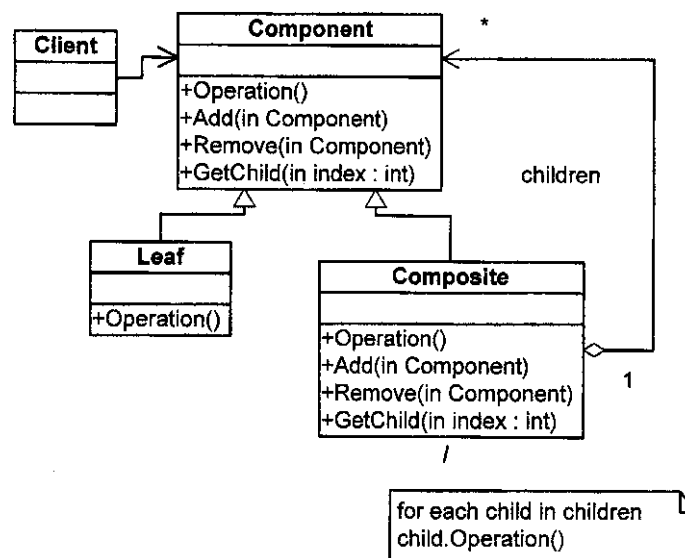


Рис.1. Структура шаблону Composite

Застосування декількох "Decorator" до одного "Component" дозволяє довільним чином поєднувати обов'язки, наприклад, одну властивість можна додати двічі.

Більша гнучкість, ніж у статичного наслідування: можна додавати і видаляти обов'язки під час виконання програми в той час як при використанні наслідування треба було б створювати новий клас для кожного додаткового обов'язку. Даний шаблон дозволяє уникнути перевантажених методами класів на верхніх рівнях ієрархії - нові обов'язки можна додавати по мірі необхідності.

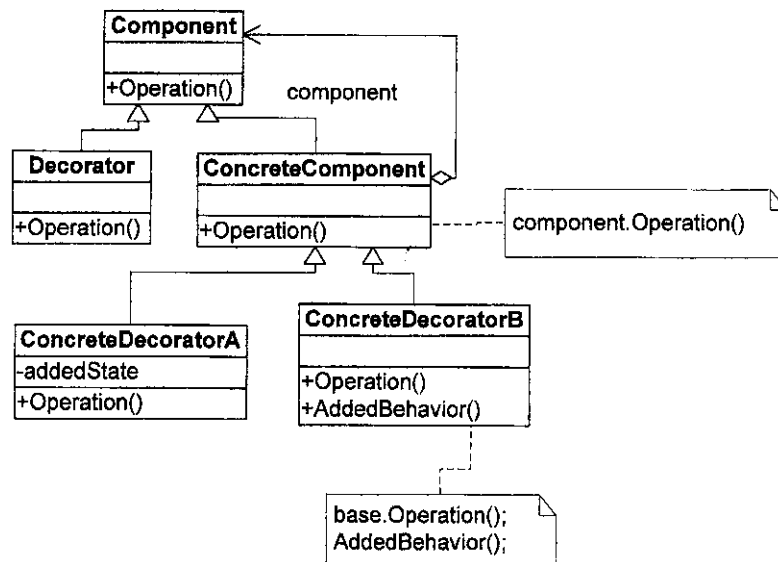


Рис.2. Структура шаблону Decorator

"Decorator" і його "Component" не ідентичні, і, крім того, виходить, що система складається з великої кількості дрібних об'єктів, які схожі один на одного і розрізняються тільки способом взаємозв'язку, а не класом і не значеннями своїх внутрішніх змінних - така система складна в вивченні та налагодженні.

Proxy

Проблема. Необхідно управляти доступом до об'єкта для таких цілей:

- створювати громіздкі об'єкти "на вимогу" (Virtual Proxy);
- кешувати дані (SmartLink Proxy);
- динамічно обмежувати доступ (Security Proxy);
- забезпечувати доступ до об'єкта в іншому адресному просторі (Remote Proxy).

Рішення. Створити сурогат громіздкого об'єкта. "Proxy" зберігає посилання, яке дозволяє заступникові звернутися до реального суб'єкта. "Proxy" контролює доступ до "Real Subject", відповідаючи за створення або видалення об'єкту "Real Subject". "Subject" визначає загальний для "Real Subject" і "Proxy" інтерфейс, так, що "Proxy" може бути використаний скрізь, де очікується "Real Subject". При необхідності запити можуть бути переадресовані "Proxy" "Real Subject".

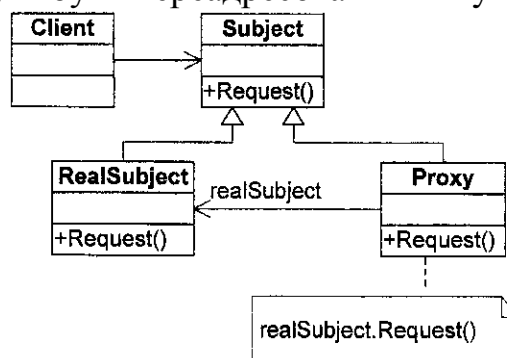


Рис.3. Структура шаблону Proxy

Завдання

1. Ознайомитись з призначенням та видами шаблонів проектування ПЗ. Вивчити класифікацію шаблонів проектування ПЗ. Знати назви шаблонів, що відносяться до певного класу.
2. Вивчити структурні шаблони проектування ПЗ. Знати загальну характеристику структурних шаблонів та призначення кожного з них.
3. Детально вивчити структурні шаблони проектування Composite, Decorator та Proxy. Для кожного з них:
 - вивчити Шаблон, його призначення, альтернативні назви, мотивацію, випадки, коли його застосування є доцільним та результати такого застосування;
 - знати особливості реалізації Шаблону, споріднені шаблони, відомі випадки його застосування в програмних додатках;
 - вільно володіти структурою Шаблону, призначенням його класів та відносинами між ними;
 - вміти розпізнавати Шаблон в UML діаграмі класів та будувати сирцеві коди Java-класів, що реалізують шаблон.
4. В підготованому проєкті (JP1) створити програмний пакет com.lab111.labwork3. В пакеті розробити інтерфейси і класи, що реалізують завдання (згідно варіанту) з застосуванням одного чи декількох шаблонів (п.3). В класах, що розробляються, повністю реалізувати методи, пов'язані з функціонуванням Шаблону. Методи, що реалізують бізнес-логіку, закрити заглушками з виводом на консоль інформації про викликаний метод та його аргументи. Приклад реалізації бізнес-методу:

```
void draw(int x, int y){  
    System.out.println("Метод draw з параметрами x="+x+" y="+y);  
}
```
5. За допомогою автоматизованих засобів виконати повне документування розроблених класів (також методів і полів), при цьому документація має в достатній мірі висвітлювати роль певного класу в загальній структурі Шаблону та особливості конкретної реалізації.

Варіанти завдання

Номер варіанту завдання обчислюється як залишок від ділення номеру залікової книжки на 12.

0. Визначити специфікації класів, які подають графічні примітиви та їх композиції у редакторі векторної графіки. Кожний примітив має атрибути розміщення – позиція (координати x та y) і розмір (ширина та висота). Реалізувати бізнес-метод відображення таких атрибутів розміщення для примітивів (задаються в конструкторі) і композицій (динамічно обчислюються).
1. Визначити специфікації класів, які подають дерево розбору складного виразу з лапками відповідно до синтаксичних правил:

$$\begin{aligned} \langle \text{вираз} \rangle &::= \langle \text{простий вираз} \rangle \mid \langle \text{складний вираз} \rangle \\ \langle \text{простий вираз} \rangle &::= \langle \text{константа} \rangle \mid \langle \text{змінна} \rangle \\ \langle \text{константа} \rangle &::= (\langle \text{число} \rangle) \\ \langle \text{змінна} \rangle &::= (\langle \text{ім'я} \rangle) \\ \langle \text{складний вираз} \rangle &::= (\langle \text{вираз} \rangle \langle \text{знак операції} \rangle \langle \text{вираз} \rangle) \\ \langle \text{знак операції} \rangle &::= +|-|*|/ \end{aligned}$$

Реалізувати бізнес-метод відображення наповнення елемента у вигляді виразу.

2. Визначити специфікації класів для подання ігрового простору з багаторівневою ієрархічною структурою. Реалізувати бізнес-метод обчислення площі, що займає елемент в умовних одиницях.
3. Визначити специфікації класів для подання блок-схем алгоритмів з блоковою організацією відповідно до семантичної діаграми.

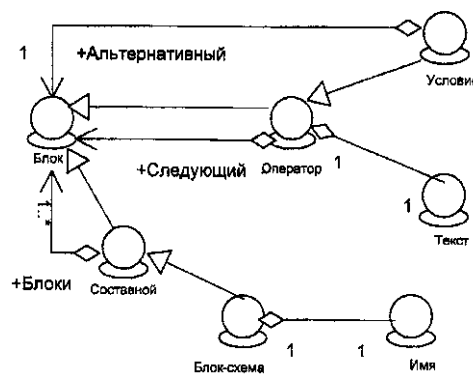


Рис.4. Семантична діаграма блок-схем алгоритмів

Реалізувати бізнес-метод ідентифікації елемента та його зв'язків з іншими елементами блок-схеми.

4. Визначити специфікації класів для подання файлової системи у вигляді дерева об'єктів (файл - листовий об'єкт, каталог - вузловий). Кожний об'єкт має атрибут розміру (для файлу задається в конструкторі, для каталогів обчислюється). Реалізувати бізнес-метод отримання розміру для класу каталогу.
5. Визначити специфікації класів та реалізацію методів для подання вибраного графічного елемента у редакторі векторної графіки. Забезпечити можливість динамічної зміни відображення елементу.

6. Визначити специфікації класів додаткових графічних зображень для графічних елементів у редакторі векторної графіки. Навести приклади використання розроблених класів-обгорток.
7. Визначити специфікації класів для подання графічних маніпуляторів геометричних властивостей(положення, розмір) у редакторі векторної графіки.
8. Визначити специфікації класів та реалізацію методів для елементів в текстовому редакторі. Розробити класи для динамічної зміни відображення елементу (приведення до верхнього регістру, приведення до нижнього регістру, додавання в кінці символу нової строки тощо).
9. Визначити специфікації класів та реалізацію методів для маніпулювання зображеннями великого розміру з можливістю прозорого кешування. Реалізувати бізнес-метод для визначення кольору точки за його координатами.
10. Визначити специфікації класів та реалізацію методів для маніпулювання зображеннями з можливістю їх "пізнього завантаження". Реалізувати бізнес-метод для визначення кольору точки за його координатами.
11. Визначити специфікації класів та реалізацію методів для маніпулювання зображеннями з можливістю контролювання доступу до об'єкта — доступ відкритий лише до точок чиї координати (x,y) лежать в межах $x_1 < x < x_2$ и $y_1 < y < y_2$ (значення x_1, x_2, y_1, y_2 задаються в конструкторі). Реалізувати бізнес-метод для визначення кольору точки за його координатами.

Протокол

Протокол має містити титульну сторінку (з номером залікової книжки), завдання, роздруківку діаграми класів, розроблений програмний код та згенеровану документацію в форматі JavaDoc.