

## **Методичні вказівки до лабораторних робіт з курсу «Інженерія програмного забезпечення»**

### **Лабораторна робота №3**

**Тема:** Структурні шаблони проектування ПЗ. Шаблони Composite. Decorator. Proxy.

**Мета:** Ознайомлення з видами шаблонів проектування ПЗ. Вивчення структурних шаблонів. Отримання базових навичок з застосування шаблонів Composite, Decorator та 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 (ЛР1) 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 розроблюваних класах 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 (№зк mod 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).

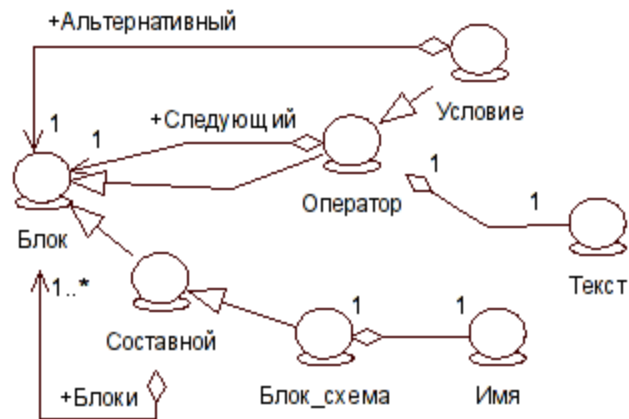
1. Визначити специфікації класів, які подають дерево розбору складного виразу з лапками відповідно до синтаксичних правил:

`<вираз>::=<простий вираз> | <складний вираз>`  
`<простий вираз>::=<константа> | <змінна>`  
`<константа>::=(<число>)`  
`<змінна>::=(<ім'я>)`  
`<складний вираз>::=(<вираз><знак операції><вираз>)`  
`<знак операції>::=+|-|*|/`

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.



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 елементу 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-обгортки.

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 сирцевий код та згенеровану документацію в форматі JavaDoc.

### Матеріали

Підготовка до лабораторної роботи здійснюється за допомогою книги:

Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns: Elements of Reusable Object-Oriented Software. — СПб: «Питер», 2007. — С. 366. — [ISBN 978-5-469-01136-1](#) (таже [ISBN 5-272-00355-1](#))

За необхідності додаткової інформації можливо використання матеріалів з мережі Інтернет, наприклад:

Шаблони проектування програмного забезпечення

- [Шаблони проектування програмного забезпечення](#)
- [Шаблон проектирования](#)
- [Обзор паттернов проектирования](#)
- [Объектно-ориентированное проектирование, паттерны проектирования \(Шаблоны\)](#)
- [David Gallardo. Шаблоны проектирования Java](#)
- [Design pattern \(computer science\)](#)
- [Подготовка к собеседованию по Java/J2EE](#)

Структурні шаблони

- [Структурні шаблони](#)
- [Structural pattern](#)
- [Структурные шаблоны](#)

- [Шаблоны проектирования: структурные паттерны](#)
- [Структурные шаблоны проектирования](#)