МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Факультет комп’ютерних наук

**Контрольна робота**

Варіант №18(6): «Proxy»

Виконали:

студентка групи КС-31

Свиридова Д.М

ХАРКІВ – 2022

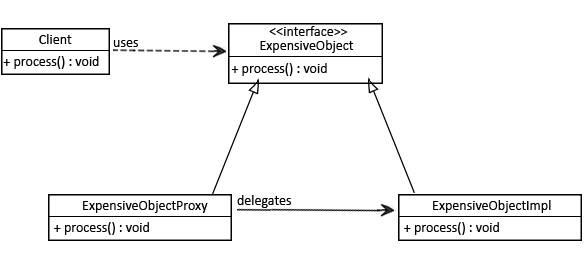
**Загальний огляд**

Шаблон Proxy дозволяє нам створити посередника, який діє як інтерфейс до іншого ресурсу, приховуючи базову складність компонента.

**Приклад використання**

Розглянемо важкий об’єкт Java (наприклад, з’єднання JDBC або SessionFactory), який потребує початкової конфігурації.

Ми хочемо, щоб такі об’єкти ініціалізувалися лише на вимогу, і як тільки це буде зроблено, ми хочемо повторно використовувати їх для всіх викликів:



Давайте тепер створимо простий інтерфейс і конфігурацію для цього об’єкта:

public interface ExpensiveObject {

void process();

}

І реалізація цього інтерфейсу з великою початковою конфігурацією:

public class ExpensiveObjectImpl implements ExpensiveObject {

public ExpensiveObjectImpl() {

heavyInitialConfiguration();

}

@Override

public void process() {

LOG.info("processing complete.");

}

private void heavyInitialConfiguration() {

LOG.info("Loading initial configuration...");

}

}

Тепер ми використаємо шаблон проксі та ініціалізуємо наш об’єкт на вимогу:

public class ExpensiveObjectProxy implements ExpensiveObject {

private static ExpensiveObject object;

@Override

public void process() {

if (object == null) {

object = new ExpensiveObjectImpl();

}

object.process();

}

}

Щоразу, коли наш клієнт викликає метод process(), він просто зможе побачити обробку, а початкова конфігурація завжди залишатиметься прихованою:

public static void main(String[] args) {

ExpensiveObject object = new ExpensiveObjectProxy();

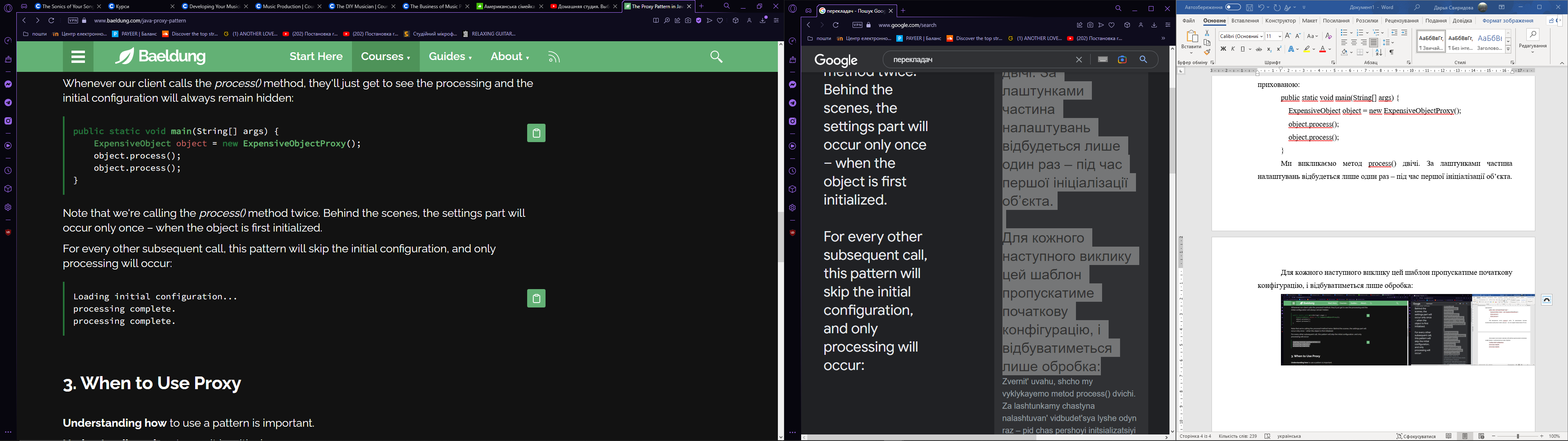
object.process();

object.process();

}

Ми викликаємо метод process() двічі. За лаштунками частина налаштувань відбудеться лише один раз – під час першої ініціалізації об’єкта.

Для кожного наступного виклику цей шаблон пропускатиме початкову конфігурацію, і відбуватиметься лише обробка:



**Висновки**

У цій роботі ми розглянули шаблон проектування проксі. Це хороший вибір у таких випадках:

* Коли ми хочемо мати спрощену версію об’єкта або отримати більш безпечний доступ до об’єкта
* Коли нам потрібна локальна версія віддаленого об’єкта