Pola Desain Perangkat Lunak

[Week] 10 – Structural Pattern, Decorator

Prepared by: Tifanny Nabarian

Design Patterns Category

Creational Patterns	Creational patterns prescribe the way that objects are created.
Structural Patterns	• Structural patterns are concerned with how classes and objects are composed to form larger structures
Behavioral Patterns	• Behavioral patterns are concerned with algorithms and the assignment of responsibilities between objects.
Concurrency Patterns	• Concurrency patterns prescribe the way access to shared resources is coordinated or sequenced

Design Patterns Scope

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	• Factory method	• Adapter	• Interpreter • Template method
	Object	• Abstract factory • Builder • Prototype • Singleton	• Adapter • Bridge • Composite • Decorator • Fasad • Flyweight • Proxy	• Chain of responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

Structural Pattern

Decorator

Decorator Concept

Definisi GoF

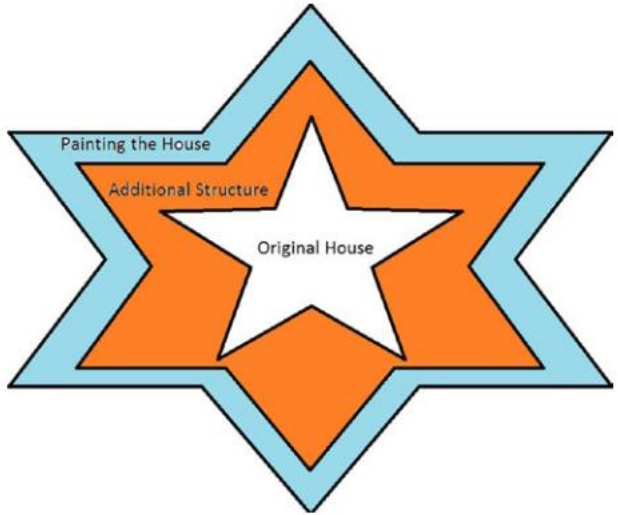
- Attach **additional responsibilities** to an object dynamically. Decorators provide a **flexible alternative to subclassing** for extending functionality.
- This pattern says that the class must **be closed for modification** but **open for extension**; That is, a new functionality can be added without disturbing existing functionalities.

Decorator Concept

Real World Example

- Suppose you already own a house.
- Now you have decided to build an additional floor on top of it.
- You **may not want to change the architecture of the ground floor** (or existing floors),
- but you may want to change the design of **the architecture for the newly added floor** without affecting the existing architecture.

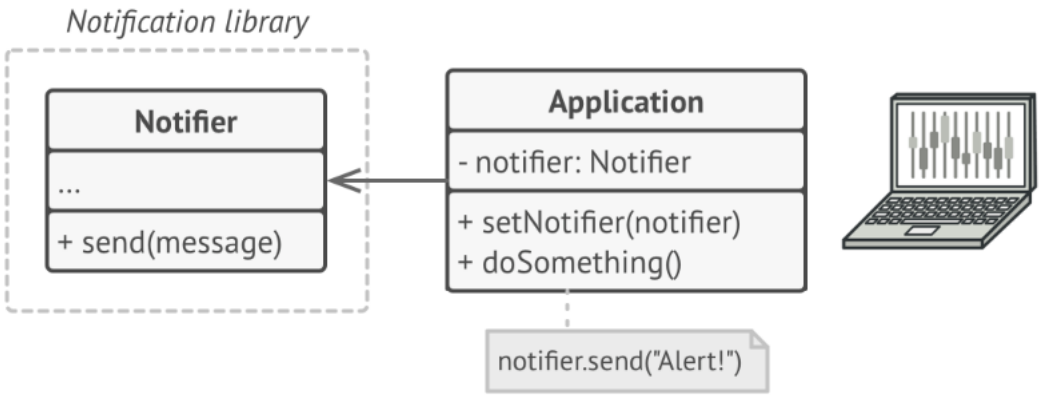
Decorator Concept



Computer World Example

https://refactoring.guru/design-patterns/decorator

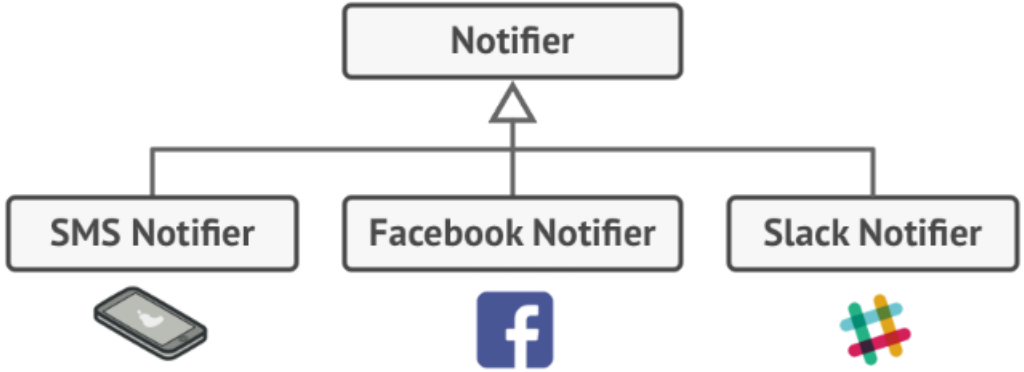
Imagine that you're working on a notification library which lets other programs notify their users about important events.



Computer World Example

https://refactoring.guru/design-patterns/decorator

At some point, you realize that users of the library expect more than just email notifications

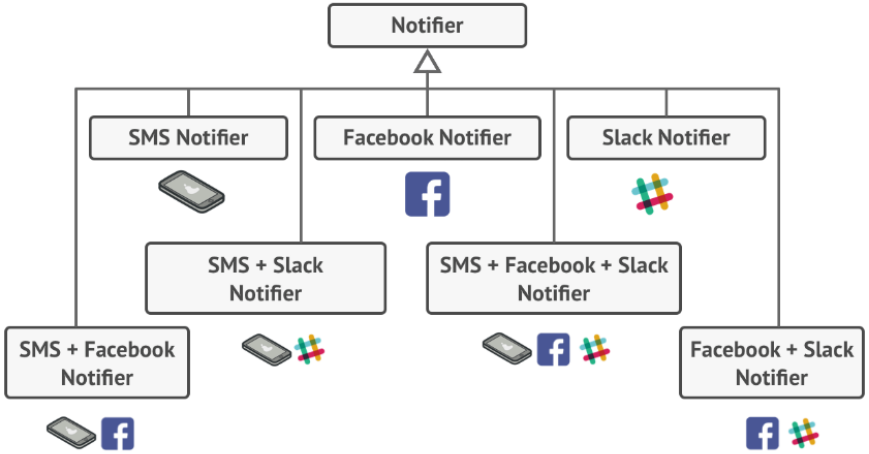


Each notification type is implemented as a notifier's subclass.

Computer World Example

https://refactoring.guru/design-patterns/decorator

But then someone reasonably asked you, “Why can’t you use several notification types at once? If your house is on fire, you’d probably want to be informed through every channel.”

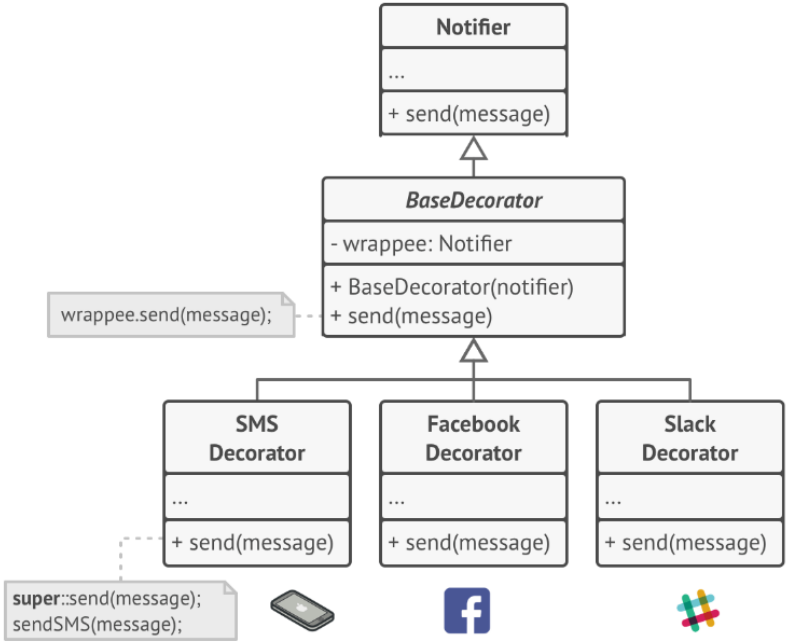


Combinatorial explosion of subclasses.

Computer World Example

https://refactoring.guru/design-patterns/decorator

Solution



Various notification methods become decorators.

Decorator Example

Note You can notice the use of the decorator pattern in the I/O streams implementations in both .NET Framework and Java. For example, the java.io.BufferedOutputStream class can decorate any java.io.OutputStream object.

Let's Practice!

Decorator Example - Illustration

Illustration

Go through the following example. Here we never tried to modify the core `makeHouse()` method. We have created two additional decorators: ConcreteDecoratorEx1 and ConcreteDecoratorEx2 to serve our needs but we kept the original structure intact.

Class Diagram

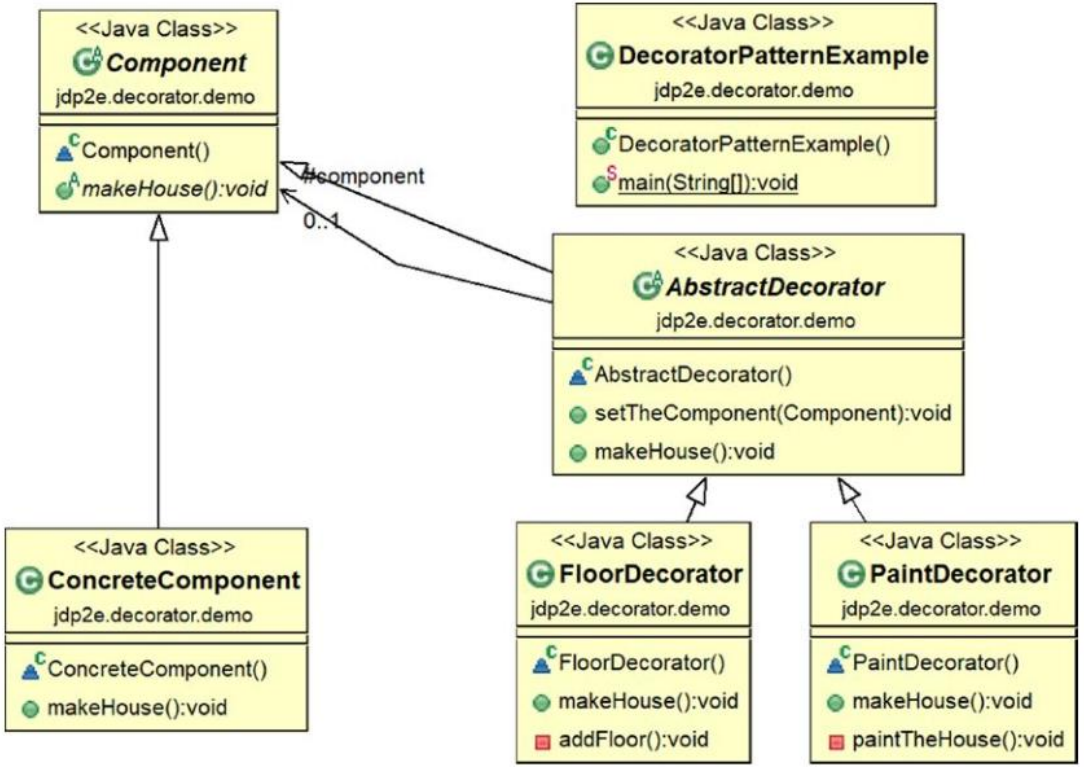
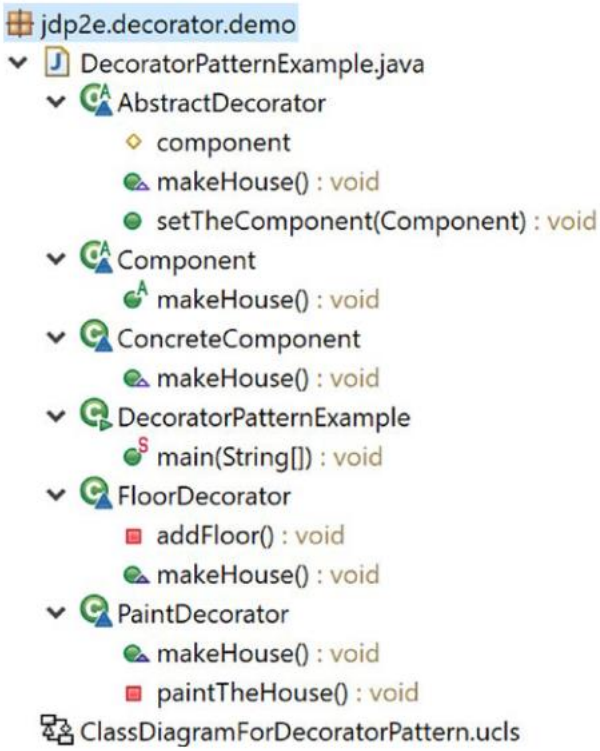


Figure 7-4. *Class diagram*

Decorator Example (Package Explorer)



Silahkan Kerjakan Tugas Praktikkum..