

# Паттерн проектирования Decorator

Или почему new BufferedReader(new InputStreamReader(...)) выглядит именно так

---

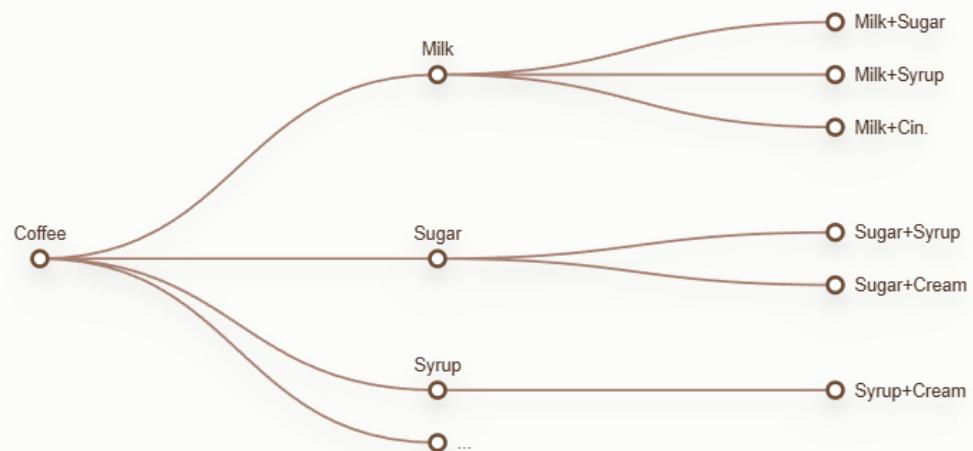
DESIGN PATTERNS SERIES

## Проблема: Взрыв классов

Задача: Расчет стоимости кофе с добавками.

Проблема: Class Explosion («Взрыв» количества классов).

Минус: Сложно поддерживать, статичная структура.



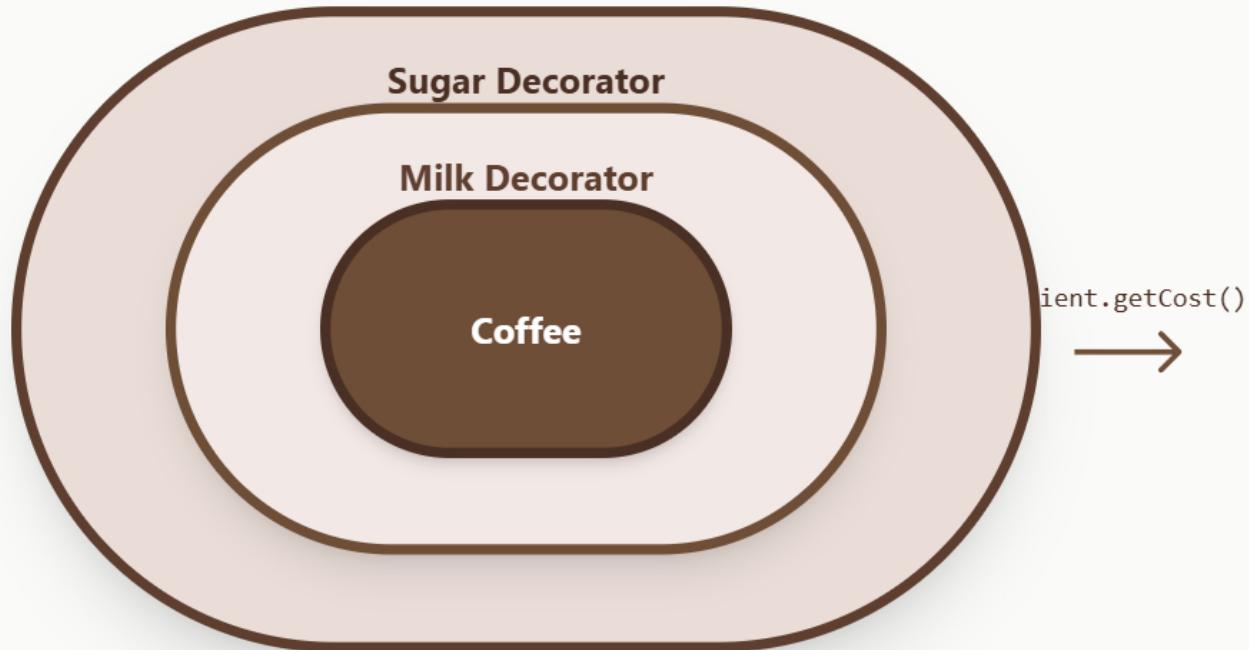
*"Class Explosion" Problem*

## Решение — Обертки (Wrappers)

Принцип: Композиция вместо наследования.

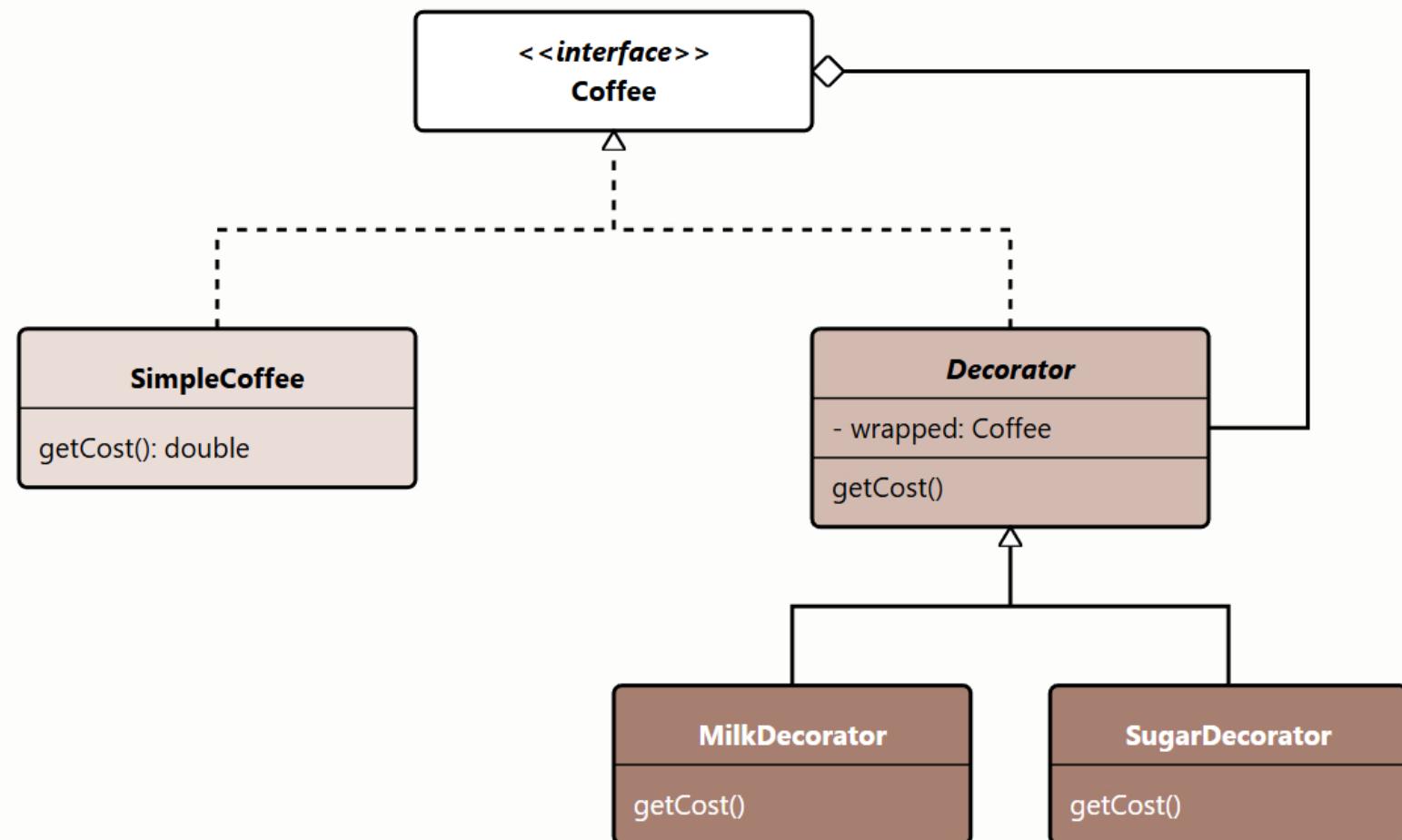
Идея: Объект помещается внутрь другого объекта.

Важно: И «Обертка», и «Оборачиваемый объект» имеют один интерфейс.



Calls pass through wrappers, each adding their behavior.

## Структура (UML)



## Реализация (Код)

● ● ● JAVA

```
// 1. Общий интерфейс
public interface Coffee {
    double getCost();
}

// 2. Базовый компонент
public class SimpleCoffee implements Coffee {
    public double getCost() { return 100; }
}

// 3. Декоратор
public class MilkDecorator implements Coffee {
    private Coffee tempCoffee; // Ссылка на объект внутри

    public MilkDecorator(Coffee newCoffee) {
        this.tempCoffee = newCoffee;
    }
}
```

## Использование (Client Code)

JAVA

```
public static void main(String[] args) {
    // 1. Просто кофе
    Coffee myCoffee = new SimpleCoffee();

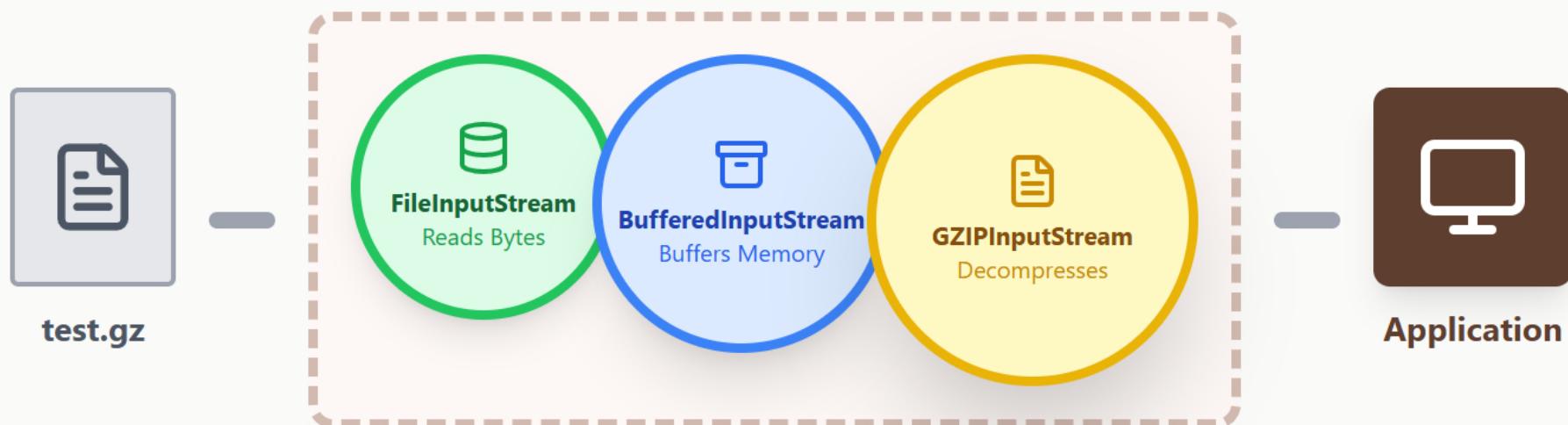
    // 2. Добавили молоко и сахар (матрешка)
    myCoffee = new SugarDecorator(
        new MilkDecorator(myCoffee)
    );

    System.out.println(myCoffee.getCost());
    // Результат: 100 (база) + 30 (молоко) + 10 (сахар) = 140
}
```

## Пример из жизни (Java I/O)

```
new GZIPInputStream(new BufferedInputStream(new FileInputStream("test.gz")))
```

```
new GZIPInputStream( new BufferedInputStream( new FileInputStream(...) ) )
```



## Плюсы и минусы

- + Гибкость выше, чем у наследования.
- + Динамическое добавление функций на лету (runtime).
- + SRP (Принцип единственной ответственности).
- Трудно отлаживать (много мелких объектов).
- Код создания объектов громоздкий (много new).

## Заключение

Декоратор — структурный паттерн.

Расширяет функционал без наследования.

Повсеместно используется в Java (I/O потоки).

**Спасибо за внимание!**