

Если жизнь без JAVA? Паттерны проектирования




Урок 1

Знакомство с паттернами проектирования





Что будет на уроке сегодня

-  Рассуждения о том, что все, что здесь изучили применимо для других языков программирования
-  Паттерны проектирования, что это такое и зачем они нужны
-  Задания по модернизации кода согласно SOLID и паттернов



Если жизнь без JAVA?

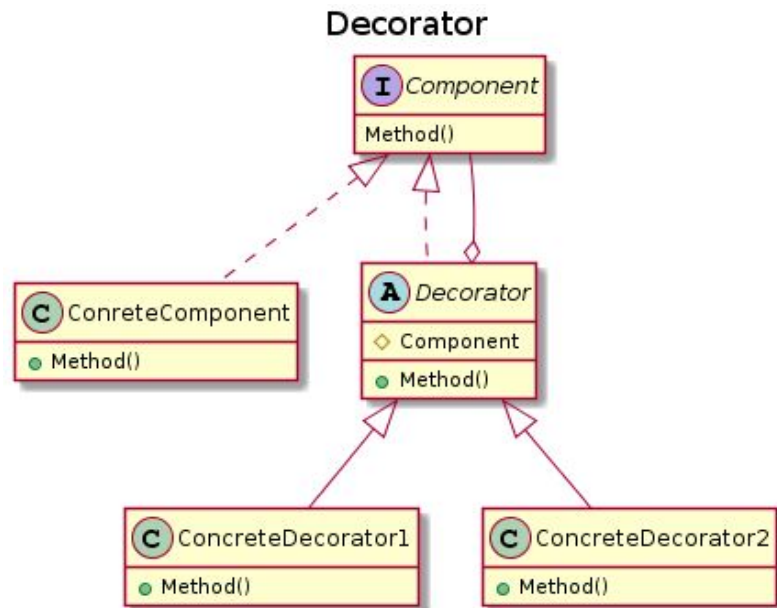


Что такое паттерны
проектирования?



Декоратор

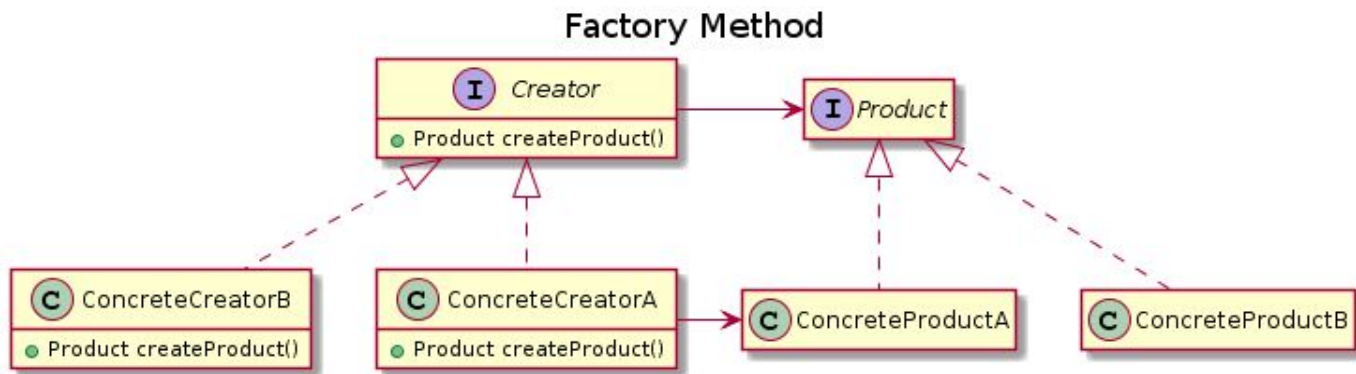
Структурный паттерн, расширяет объект новой функциональностью не используя наследование





Фабричный метод

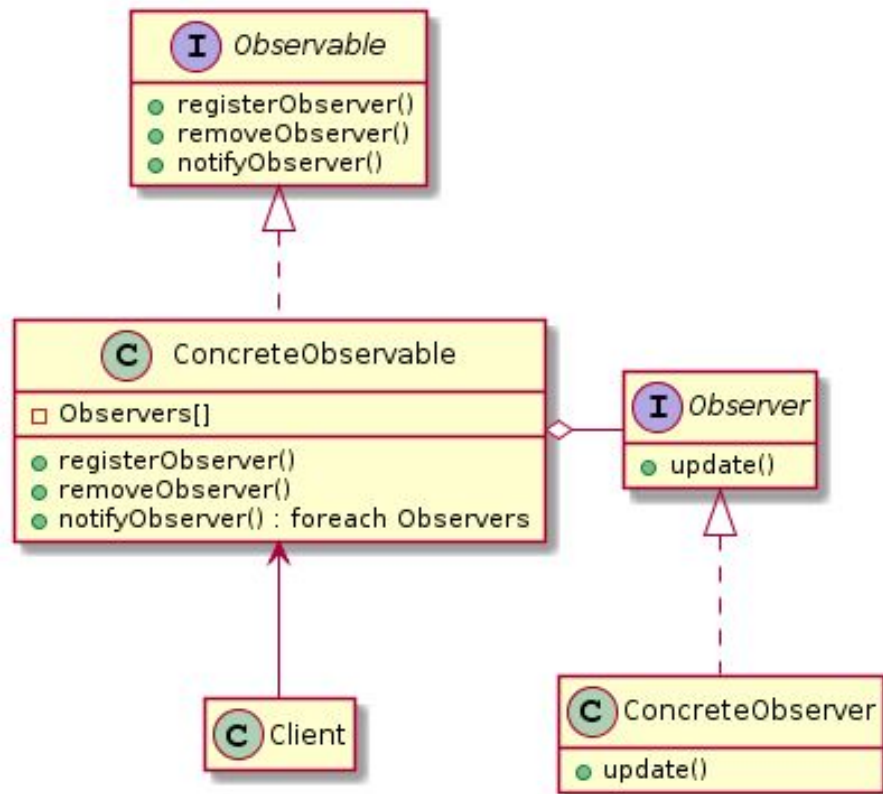
Порождающий паттерн, динамически создающий необходимый класс по интерфейсу из семейства классов





Наблюдатель

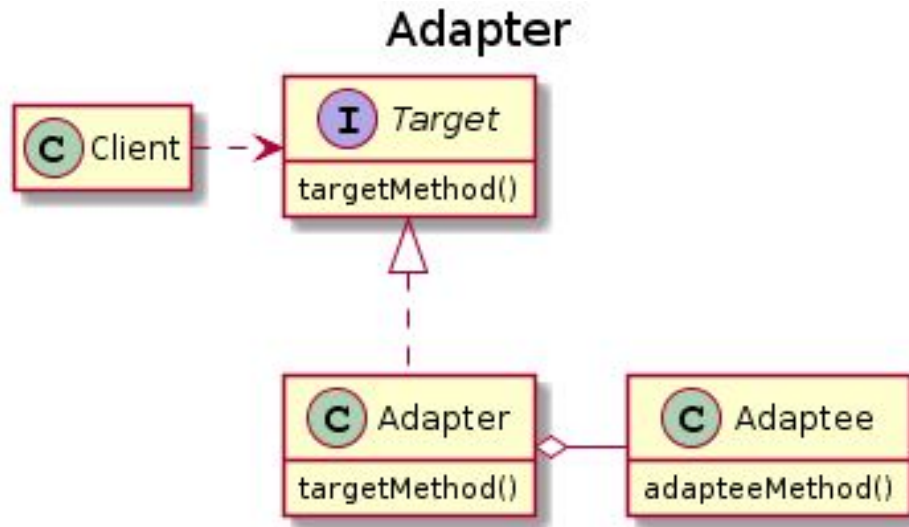
Поведенческий паттерн, позволяет наблюдать за изменениями объектов, при этом наблюдаемый объект и наблюдатель не имеют прямой связи





Адаптер

Структурный паттерн, адаптирует один интерфейс под другой





Задание 1

Проект DecoratorFactory: Создать логирование операций, для класса Calculator. Класс Calculator не должен измениться.

Можно использовать ссылку на код:

https://github.com/DwarfPorter/OOP_Lesson7_Patterns/tree/master/DecoratorFactory



Перерыв?

Голосуйте в чате



Задание 2

Проект Observer: Добавить нового наблюдателя (сейчас два наблюдателя Master и Student), организовать наблюдение за объектом Company.

Можно использовать ссылку на код:

https://github.com/DwarfPorter/OOP_Lesson7_Patterns/tree/master/Observer



Задание 3

Проект Adapter: Включить датчик от другой компании, интерфейс которого отличается от предыдущих, в систему. Первоначальный интерфейс `MeteoSensor`, адаптируем `SensorTemperature`.

Можно использовать ссылку на код:

https://github.com/DwarfPorter/OOP_Lesson7_Patterns/tree/master/Adapter



Домашнее задание



Итоговое домашнее задание по курсу

Создать проект калькулятора комплексных чисел (достаточно сделать сложение, умножение и деление). Применить при создании программы архитектурные паттерны, добавить логирование калькулятора. Соблюдать принципы SOLID, паттерны проектирования. Можно выбрать другой язык программирования, например C# или Python, если выбран язык, отличный от JAVA, то необходимо написать документ, каким образом можно запустить приложение (что необходимо установить, каким образом запускать и т.п.)



Критерии оценивания домашнего задания

Зачтено (отлично) ставится, если приложение написано, сдано через github, либо другую платформу на основе git, в runtime работает без ошибок, принципы ООП, SOLID соблюдены.

Зачтено (хорошо) ставится, если приложение написано, сдано через github, либо другую платформу на основе git, в runtime возможны незначительные ошибка (не стопперы), но при этом принципы ООП соблюдены и возможны отклонения по SOLID.

Зачтено (удовлетворительно) ставится, если приложение написано, сдано через github, либо другую платформу на основе git, в runtime присутствуют ошибки стопперы или принципы ООП и SOLID не соблюдаются.

Не зачтено ставится, если приложение сдано не через github, либо другую платформу на основе git, сдано что-то другое, сдана пустышка и т.п.



Вопросы?

Вопросы?



Вопросы?





Был ли вам полезен курс?
Что было нового и интересного?





Спасибо за работу!