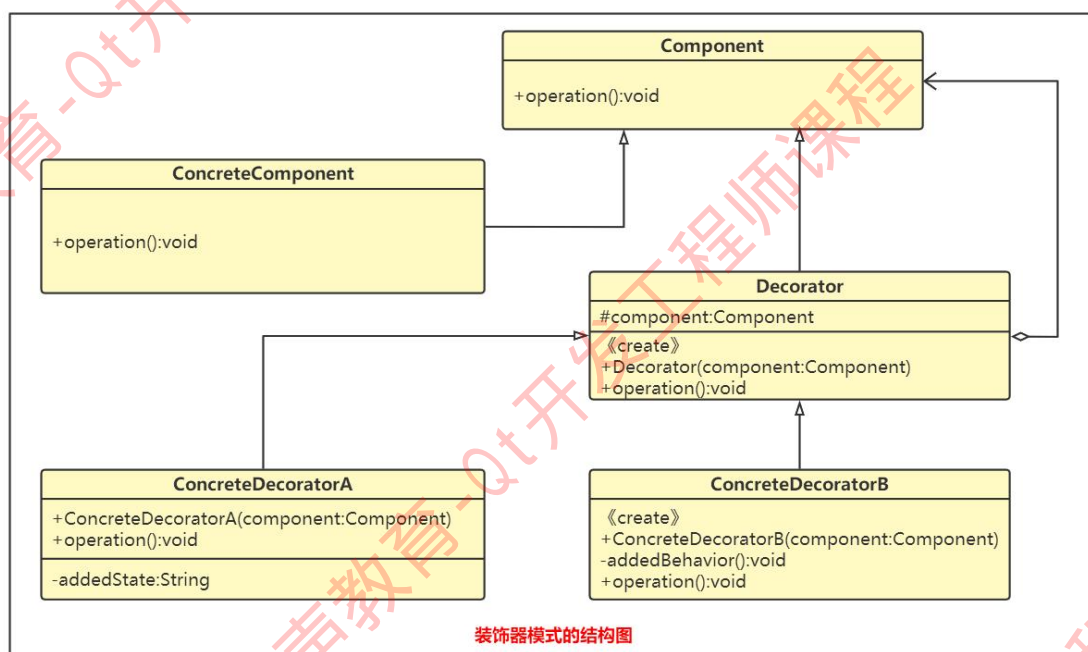


1045--23 种设计模式之《装饰器模式》--零声教育

一、装饰器模式基础知识

装饰器模式定义：装饰器模式（Decorator Pattern）也称为包装模式（Wrapper Pattern），动态地给一个对象添加一些额外的职责。就增加功能来说，装饰器模式相比生成子类更为灵活。装饰器模式的结构图如下：



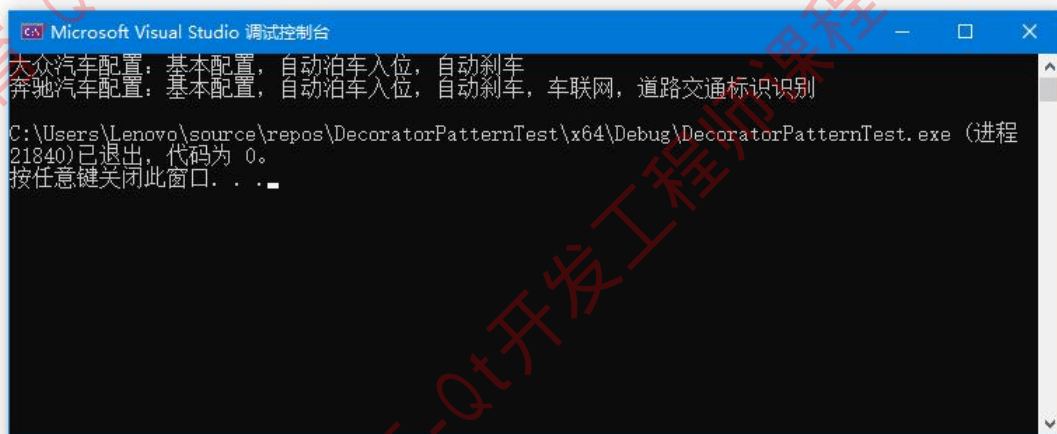
- **Component:** 组件对象的接口，可以给这些对象动态地添加职责。
- **ConcreteComponent:** 具体的组件对象，实现组件对象接口，通常就是被装饰器装饰的原始对象，也就是可以给这个对象添加职责。
- **Decorator:** 所有装饰器的抽象父类，需要定义一个与组件接口一致的接口，并持有一个 **Component** 对象，其实就是持有一个被装饰的对象。

- **ConcreteDecorator**: 实际的装饰器对象，实现具体要向被装饰对象添加的功能。

装饰器模式优点：比继承更加灵活；简化高层定义。

装饰器模式缺点：会产生很多粒度对象。

二、装饰器模式实战应用



```
Microsoft Visual Studio 调试控制台
大众汽车配置：基本配置，自动泊车入位，自动刹车
奔驰汽车配置：基本配置，自动泊车入位，自动刹车，车联网，道路交通标识识别
C:\Users\Lenovo\source\repos\DecoratorPatternTest\x64\Debug\DecoratorPatternTest.exe (进程
21840)已退出，代码为 0。
按任意键关闭此窗口。 . . .
```