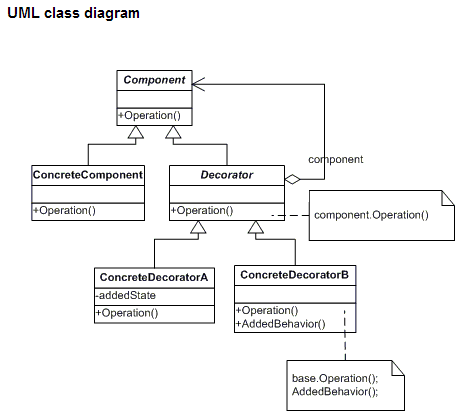
*Decorator Design Pattern*



Decorator е structural design pattern, който служи за добавяне на допълнителна функционалност към клас/обект статично или динамично (runtime) без да има ефект върху други инстанции на обекти от същия клас. По този начин се спазва open/closed principle – даден клас да е отворен за добавяне на функционалност (екстендване) и затворен за модификация.

Имаме базов клас, които си има наследници, абстрактен декоратор клас, който приема в конструктора си инстанция на оригиналния клас, като по този начин я обвива и се държи като нея. Абстрактния декоратор бива наследен от контретни декоратори, в които вече се добавя небходимата допълнителна функционалност.

Decorator е алтернатива на наследяването, като използването му предотвратява т.нар. class explosion – експоненициално нарастване на класове, съответно трудна поддръжка, повтаряне на код и т.н.

*Източници:*

[*http://www.dofactory.com/Patterns/PatternDecorator.aspx#\_self1*](http://www.dofactory.com/Patterns/PatternDecorator.aspx#_self1)

[*http://en.wikipedia.org/wiki/Decorator\_pattern*](http://en.wikipedia.org/wiki/Decorator_pattern)

[*http://www.codeproject.com/Articles/479635/UnderstandingplusandplusImplementingplusDecoratorp*](http://www.codeproject.com/Articles/479635/UnderstandingplusandplusImplementingplusDecoratorp)