Шаблон Bridge

Шаблонът Bridge разграничава абстракцията от нейното приложение, като дава възможност те да се променят независимо. Този шаблон е много полезен когато има нова версия на софтуера и той трябва да замени вече съществуващата, като старата версия трябва да продължи да работи. Кода на клиента не трябва да се променя, тъй като съответства на дадената абстракция, но клиента трябва да посочи коя версия ще използва неговия код.

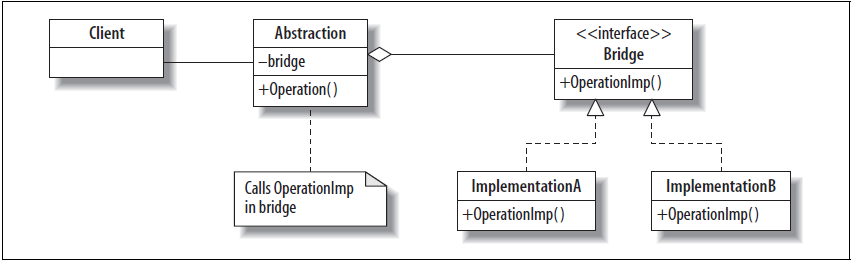
Пример:

Нека разгледаме новата версия на .NET Framework който се използва за компилиране стартиране не C# програми. Обикновено може да има няколко версии на този фреймуърк заредени на нашия компютър по всяко време и може да се избира коя от тях да се използва чрез външно задаване на пътя\* в операционната система Windows. Задаването на пътя е bridge между това, което искат приложенията като фреймуърк и действителната версия която ще получат.

\* Да припомним че пътя при Windows 7 се задава чрез -> десен бутон на My computer -> Properties -> Advanced system properties -> Advanced -> Environment Variables.

Проектиране

Наследяването е често срещан начин за създаване на различни имплементации на една абстракция. Въпреки това, имплементациите са обвързани здраво към абстракцията и е трудно да се променят независимо. Шаблонът Bridge осигурява алтернатива за наследяване когато има повече от една версия на дадена абстракция. Може да разгледаме UML диаграмата по долу. Двете реализации, А и В имплементират интерфейса наречен Bridge. Класът Abstraction съдържа поле от типа Bridge, но от друга страна не е във връзка с имплементациите.



От тази диаграма може да посочим основните роли при този шаблон.

*Abstraction* – интерфейсът който потребителя вижда

*Operation* – метода, извикван от клиента

*Bridge* – интерфейсът който дефинира тази част от Abstraction, така че да може да се променя към *ImplementationA* и *ImplementationB*.

*ImplementationA* и *ImplementationB* – имплементации на Bridge

*OperationImp* – метода в Bridge, който се извиква от Operation в Abstraction

Както при Декоратор и тук може да има няколко имплементации за една абстракция. Наследниците не трябва да имплементират Bridge , ако те ще се инстанцират по стария начин. Тази употреба е смислена само ако те трябва да се използват за взаимозаменяемост с нови реализации.

**Пример**:

От по горе с версиите на .NET Framework:

*Abstraction* - NET Framework

*Operation* - C# компилатор

*Bridge* – пътя за EnvironmentVariables

*ImplementationA* - 4.0 версия на фрейуърка

*ImplementationB* - 3.5 версия на фрейуърка

A’s *OperationImp* - 4.0 компилатор

B’s *OperationImp*- 3.5 компилатор