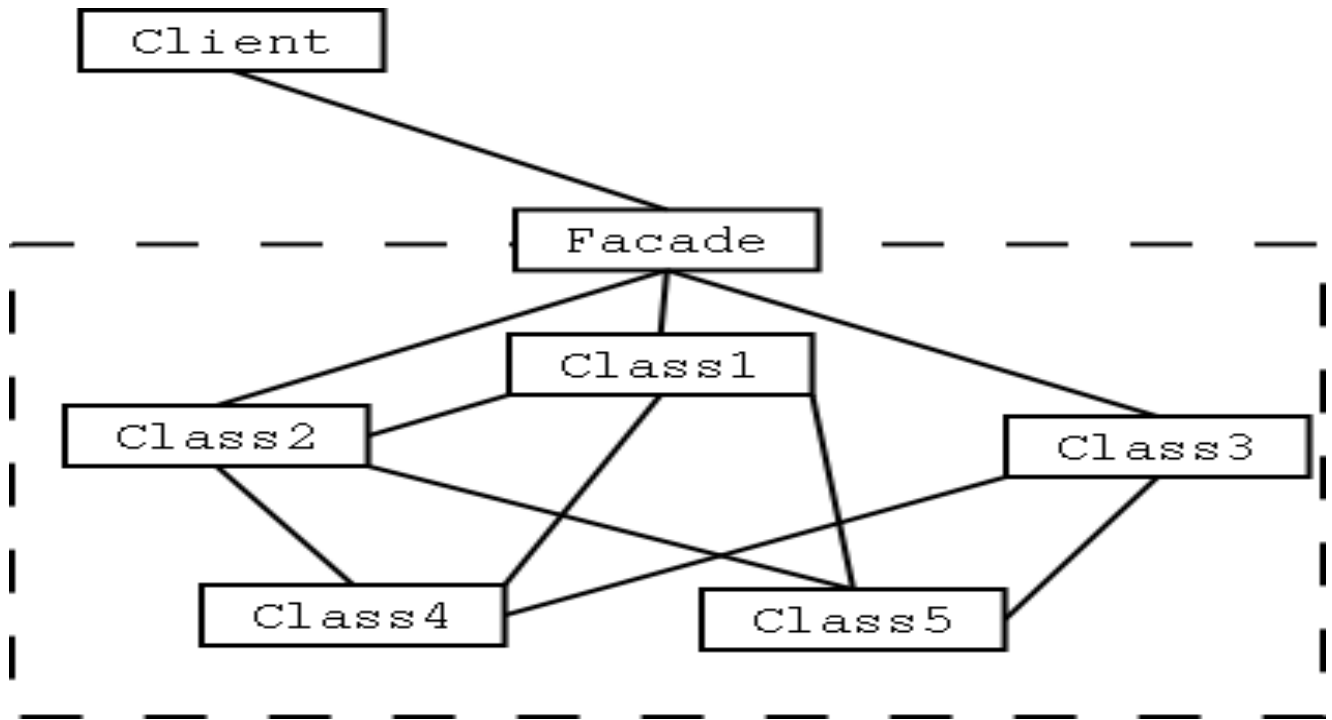


# Façade Pattern

Фасадата улеснява работата с дадена система, като прави конкретни сценарии от нейната функционалност лесни за употреба. Така ненужната за клиентите информация остава “скрита”, но без да лишава от възможността на пълен достъп, тези които трябва да използват сложните сценарии. Прави нещо сложно да “изглежда” просто.



## Основни характеристики:

- Façade има инстанции на нужните за изпълнението класове.
- Класовете на подсистемата не знаят кой създава техни инстанции.
- Клиентите, които искат да работят със пълната функционалност на системата, имат възможността да пренебрегнат съществуването на фасадата и да се обърнат директно към класовете на системата.
- Безсмислено е да имплементираме фасада при положение че ни трябва цялата функционалност на системата.
- Благодарение на фасадата се спазва един основен принцип в ООП – неговият смисъл е всеки обект да бъде обвързан (да инстанцира) колкото се може по малко други обекти от системата, т.е. да познава по малко класове. По този начин тази слаба обвързаност способства за по-добра поддръжка при настъпване на евентуална промяна в архитектурата на системата.

- Можем с различни фасади да предоставяме различни функционалности на клиента, като зад фасадата стои една и съща система.

## Реални имплементации:

- Web Service - много широко употребяван пример за фасада. Сам по себе си сервиза скрива дадена система и предоставя различна функционалност на външни клиенти.
- Logger
- Session Façade
- Track System Usage

## Предимства:

- Предоставя се по-лесен за употреба интерфейс.
- Редуцират се броя обекти, с които клиента работи.
- Намалява се обвързаността.
- Разсложява се системата.
- По-лесна поддръжка.