

代理模式



段維瀚 老師





Session

- 代理模式
 - 靜態代理
 - 動態代理



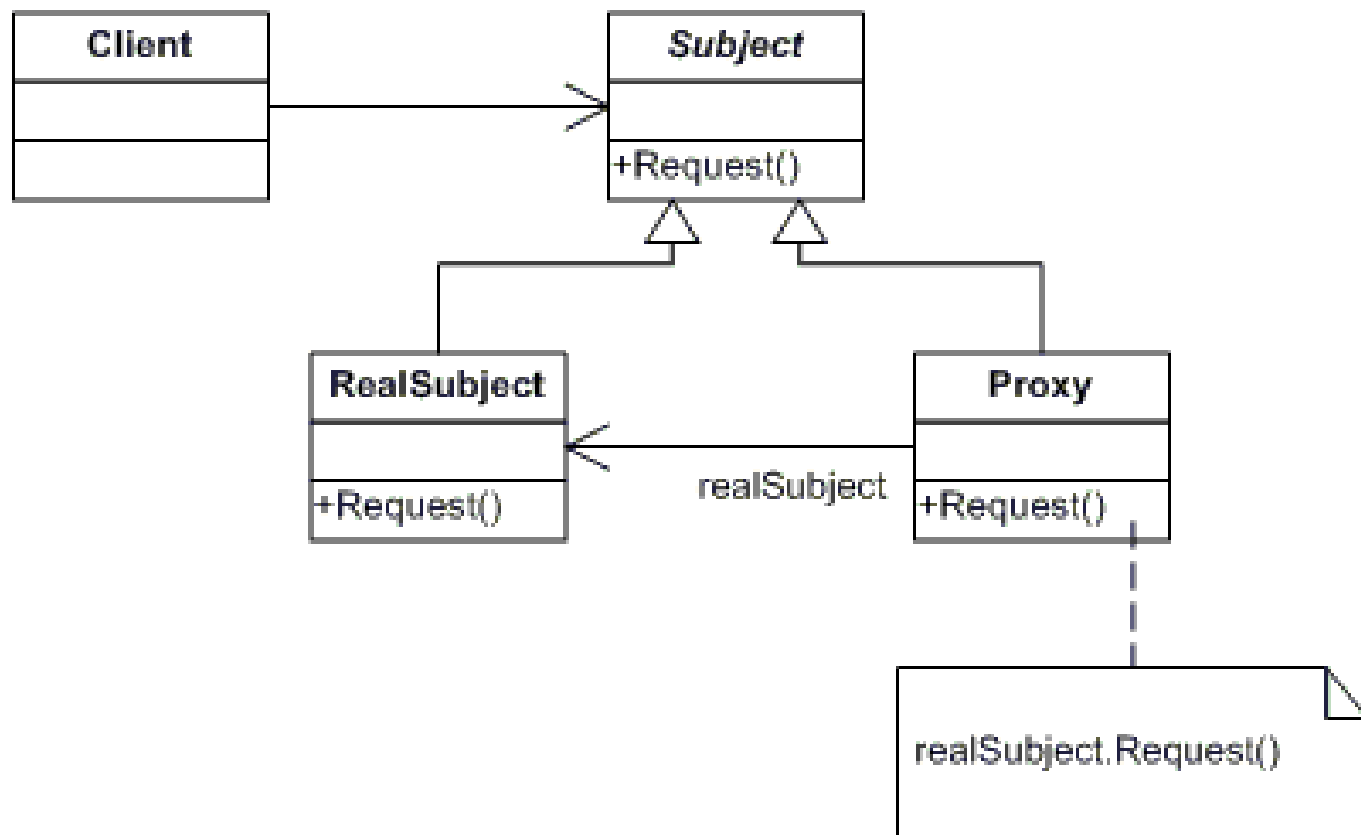


代理模式

- 給某個物件提供一個代理物件，並由代理物件控制對於原物件的訪問，即客戶不直接操控原物件，而是通過代理物件間接地操控原物件。
 - 目的
 - 讓業務方法與公用方法分開
 - 二種代理
 - 靜態代理
 - 動態代理



靜態代理





靜態代理

- 在靜態代理的實現中，代理物件與被代理的物件都必須實現同一個介面，在代理物件中可以實現記錄等相關服務，並在需要的時候再呼叫被代理的物件，如此被代理物件當中就可以僅保留商務相關職責。





靜態代理

- 優點

- 直覺快速，多用在少量類別
- 也可統一處理例外（若有必要的話）

- 缺點

- 如果大量使用靜態代理可能就會每次實現一個代理就得去寫多一個類別 Orz...





動態代理

- 在JDK 1.3之後加入了可協助開發動態代理功能的API等相關類別，您不必為特定物件與方法撰寫特定的代理物件，使用動態代理，可以使得一個處理者（Handler）服務於各個物件。
 - 處理者的類別設計必須實作 `java.lang.reflect.InvocationHandler` 介面





動態代理

- JDK 動態代理

- 僅支援 interface 代理

- 因為 Java 的單一繼承特性（每個代理類都繼承了 Proxy 類），只能針對介面建立代理類，不能針對類建立代理類

- cglib 動態代理

- 相對於 JDK 動態代理而言，cglib 動態代理讓委託類別也可以是非介面的實現類。

