

Patrons

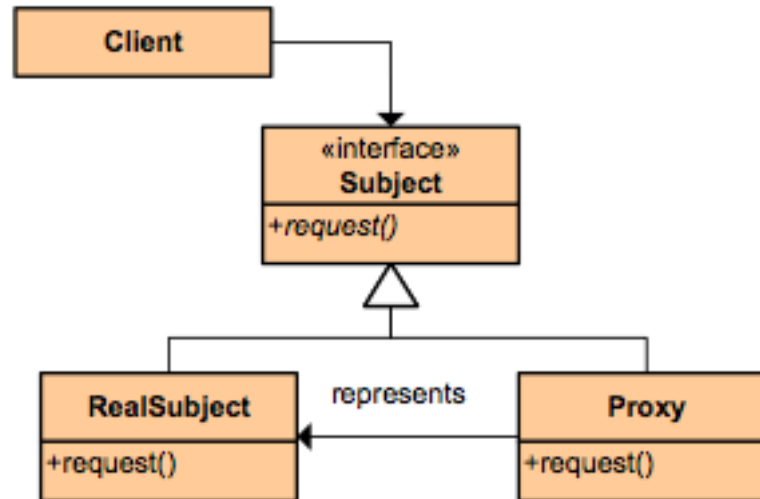
Proxy

Patró proxy

Objectiu

- Proporciona un objecte que controla l'accés a un altre. Amb aquest patró podem controlar quan s'inicialitza i crea un objecte, permeten que només es faci quan és necessari.

Diagrama



Patró proxy

- **Subject:** Defineix la interfície comú entre el RealSubject i el Proxy. D'aquesta manera és pot utilitzar el Proxy quan s'espera un RealSubject.
- **RealSubject:** Defineix l'objecte sobre el que treballa el Proxy.
- **Proxy:** Manté una referència al objecte real. Proporciona una interfície igual a la del Subject per poder substituir-la ho. És el responsable de la creació, eliminació del objecte i de l'accés a ell.

Tres tipus de Proxy

- Proxy remot: Són els responsables de enviar les peticions al objecte.
- Proxy virtual: Guarden informació addicional del objecte.
- Proxy de protecció: Comprova que l'objecte que crida al Subject té permís de fer-ho.

Patró proxy

Aplicabilitat

Es pot fer servir el patró Proxy quan es necessita:

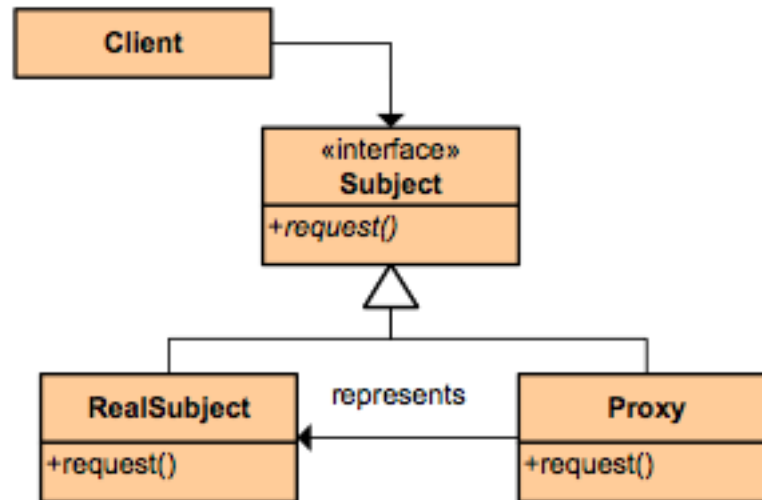
1. un representant d'un determinat objecte (Proxy remot).
2. controlar l'accés a un objecte (Proxy de protecció).
3. un substitut d'un objecte que a més a més proporcioni funcionalitats noves o addicionals (Proxy virtual).

Avantatges i inconvenients

- Un Proxy pot optimitzar la gestió dels objectes.
- Permeten realitzar tasques de manteniment.
- A vegades es necessari fer còpies d'objectes que poden ser costoses.
Mitjançant el Proxy podem realitzar les còpies amb l'avantatge de que només ho farem quan l'objecte canviï.

Patró proxy

El patró Proxy proporciona un substitut d'un objecte per a controlar-ne l'accés (es tracta d'una mena de mitjancer). S'utilitza quan es vol interactuar amb un objecte sense tenir-ne una referència directa ja sigui perquè no es vol o simplement perquè no es te. Disposa de la mateixa interfície que l'objecte al que controla l'accés; d'aquesta manera es pot tractar el Proxy com si es tractes de l'objecte en si. Normalment s'encarrega de la instanciació de l'objecte.



Patró proxy

Aquest patró pot servir per a solucionar problemes diversos. Per aquest motiu, se sol separar en tres:

- **Virtual Proxy**: controla la creació d'un objecte molt pesat pel que fa a memòria. L'objecte no es crea fins que no es necessita de veritat. Un bon exemple d'aquesta variant podria ser un programa que utilitza imatges molt grans. Es podria definir un Proxy per les imatges i com que disposen de la mateixa interfície, el client no sabria que existeix una altra classe pel mig. A l'inici del programa es podrien crear totes les imatges (que serien realment objectes Proxy molt petits) i quan es necessitessin de veritat (mostrar-les per pantalla . . .) el Proxy crearia l'objecte de veritat.
- **Protection Proxy (rewall)**: controla l'accés a un recurs basat en permisos d'accés. Es crida un objecte protegit pel Proxy i aquest decideix si permet accedir-hi o no.
- **Remote Proxy**: controla l'accés a un objecte remot. Utilitzat quan es fa servir RMI (Remote Method Invocation) en Java, per exemple.