

Proxy DP

4-Feb-20 11:11 PM

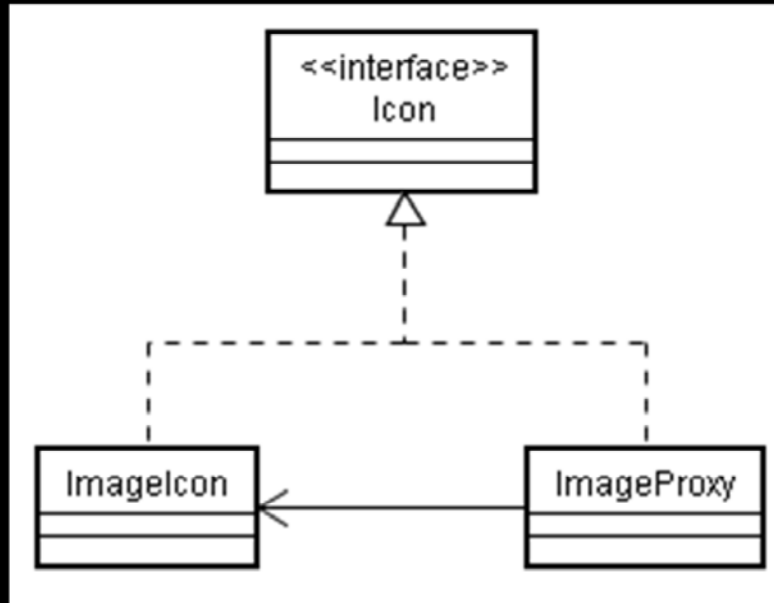
209

Problem

- Design for this task.
 - We display a list of CD titles in a menu. The image for each CD title is retrieved from an online service.
 - We don't want to waste time in getting the image of all the CD titles.
 - The first time a CD title is requested, we start a background thread to get the image. We display a temporary general image for the CD title till the time we retrieve the actual image.
 - The second time the CD title is requested, we need not fetch the image again because it is cached.



Solution



4-Feb-20 11:11 PM

211

Proxy is applicable whenever there is a need for a more versatile reference to an object than a simple pointer.

=====

We need to support resource-hungry objects.

We do not want to instantiate such objects unless and until they are actually requested by the client.

Design a surrogate or proxy object that

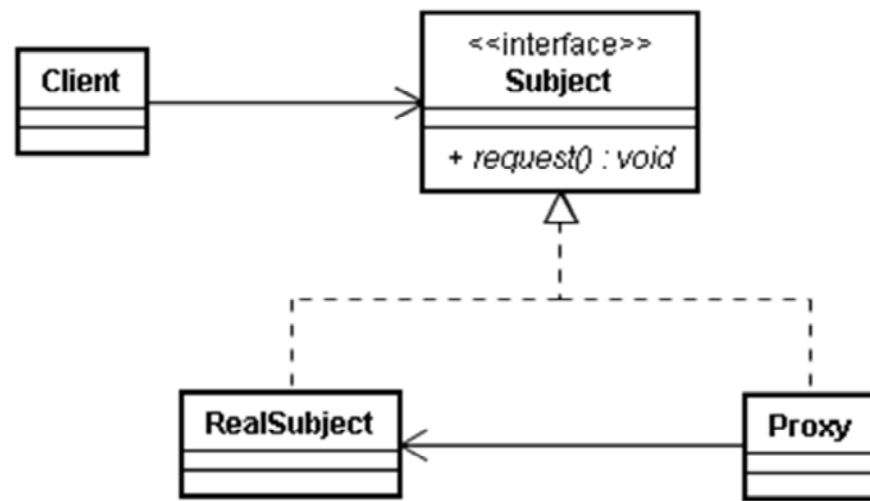
instantiates the real object the first time the client makes a request of the proxy

remembers the identity of this real object

forwards the instigating request to this real object.

Then all subsequent requests are simply forwarded directly to the encapsulated real object.

Diagram for Proxy



4-Feb-20 11:11 PM

212

The Subject provides same interface for RealSubject and the Proxy.

By implementing this interface, the Proxy can be substituted for the RealSubject anywhere it occurs.

The RealSubject is the object that does the real work.

The Proxy holds a reference to the RealSubject.

Clients interact with the RealSubject via Proxy

Non Software Analog



4-Feb-20 11:11 PM

213

Movie double replaces a real hero during light setup

Another examples: Garden hose with and without attachment.

I use a hand-held hose to water my garden. A few times a year, I add a container between the hose fitting and the watering head, which puts plant food into the water as it passes through.

I take no different action when watering in this way, and yet there is additional behavior; feeding the plants. Also, various different plant foods could be introduced in this way (although only one at any given time), and again I would take no different action at watering time.

Improve this Pseudo-code

- Let's say we are setting in the Graphical User Interface (GUI) screen
- The end user gets to decide whether to see all the titles, or just the titles from the current decade.

```
get the list of titles
if(GUI filter box is checked)
    filter the list down
        to current decade
endif
return the list of titles
```

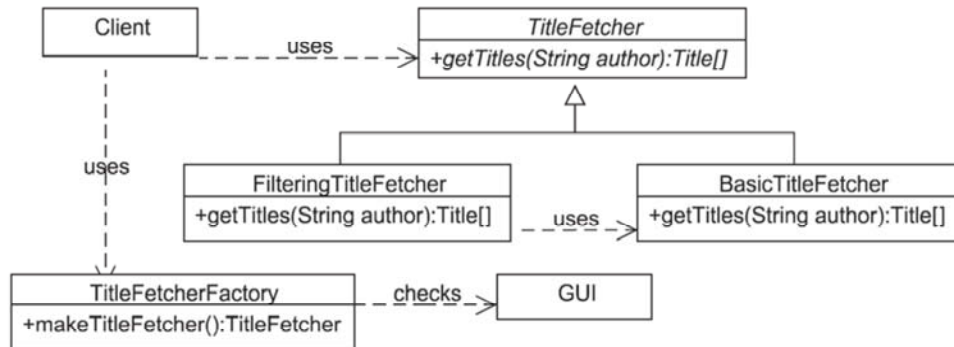


4-Feb-20 11:11 PM

214

It is pretty clear that I have coupled the client to the GUI: the fact that there is a GUI, that it works a certain way, that you can get the state of the check box a certain way, all that. If the GUI changes in any significant way, I'll have to go back into this code and change it. Now, imagine that there are many different objects that need this kind of behavior: If the GUI changed, I would have to remember to change them all.

Solution



4-Feb-20 11:11 PM

215

TitleFetcher is an abstract class or an interface.

Usage of Proxy DP

- Logging Proxy
- Protection Proxy
- Remote Proxy e.g. Stub in RMI / CORBA
- Virtual Proxy.
- Cache Proxy

4-Feb-20 11:11 PM

216

Virtual Proxy: When the actual object is heavyweight

A remote proxy provides a local representative for an object that resides in a different address space.

This is what the "stub" code in RPC and CORBA provides.

A protective proxy controls access to a sensitive master object. The "surrogate" object checks that the caller has the access permissions required prior to forwarding the request.

A smart proxy interposes additional actions when an object is accessed.

Typical uses include:

Counting the number of references to the real object so that it can be freed automatically when there are no more references (like a smart pointer),

Loading a persistent object into memory when it's first referenced,

Checking that the real object is locked before it is accessed to ensure that no other object can change it.

=====

There are a number of different proxies, named for the behavior they add to class being proxied. Only some of them are possible with the class proxy form, but all of them are possible with the form that uses delegation.

Some examples:

Logging Proxy: Logs all calls to the method of the original class, either before or after the base behavior, or both.

Protection Proxy: Block access to one or more methods in the original class. When those methods are called, the Protection Proxy may return a null, or a default value, or throw an exception, etc...

Remote Proxy: The Proxy resides in the same process space as the client object, but the original class does not. Hence, the Proxy contains the networking, piping, or other logic required to access the original object across the barrier. This cannot be accomplished with a class proxy.

Virtual Proxy: The Proxy delays the instantiation of the original object until its behavior is called for. If its behavior is never called for, then the original class is never instantiated. This is useful when the original class is heavyweight and/or there are a large number of instances needed. The Virtual Proxy is very lightweight. This cannot be accomplished with a class proxy.

Cache Proxy: The Proxy adds caching behavior to an object that represents a data source.

Differentiate between

- Adapter
- Proxy
- Decorator



4-Feb-20 11:11 PM

217

Adapter vs. Proxy vs. Decorator.

Adapter provides a different interface to its subject.

Proxy controls access to the subject by providing same interface or a subset of that interface.

Decorator provides an enhanced interface.

Decorator and Proxy have different purposes but similar (not same) structures.

Decorator usually does not create the actual object itself, while a proxy usually does that.

OOAD slides: Integration Issues, Causes of Application Failure, A Bank Interface, Avoid out of memory errors,

Usage of Proxy

- Spring @Transactional

4-Feb-20 11:11 PM

218



Demo of how Aspects (proxies) are used.

Java Demo In project DesignPatternsParticipants springAspectDemo and proxy.circuitBreaker

C# Demo in DesignPatternsParticipants.proxy

Unit Testing proxy example: mocking

proxy.unitTesting in DesignPatternsParticipants project for Java

proxyUnitTesting in DesignPatternsParticipatns project for C++