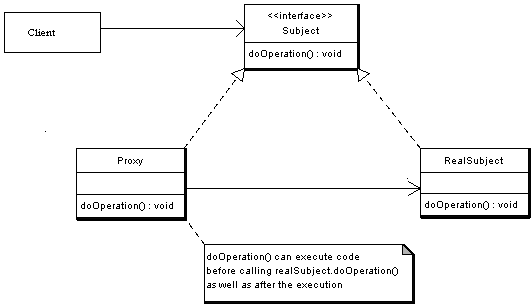
Proxy

**Definition**: The Proxy Pattern will provide an object a surrogate or placeholder for another object to control access to it. It is used when you need to represent a complex object with a simpler one. If the creation of an object is expensive, it can be postponed until the very need arises and meanwhile a simpler object can serve as a placeholder. This placeholder object is called the “Proxy” for the complex object.

**Problem**: You need to support resource-hungry objects, and you do not want to instantiate such objects unless and until they are actually requested by the client.

**Solution**:



**Participants:**

* **Subject** — Interface implemented by the RealSubject and representing its services. The interface must be implemented by the Proxy as well so that the Proxy can be used in any location where the RealSubject can be used.
* **Proxy**1.Maintains a reference that allows the Proxy to access the RealSubject.  
  2. Implements the same interface implemented by the RealSubject so that the Proxy can be substituted for the RealSubject.  
  3. Controls access to the RealSubject and may be responsible for its creation and deletion.  
  4. Other responsibilities depend on the kind of proxy.
* **RealSubject** — the real object that the proxy represents.