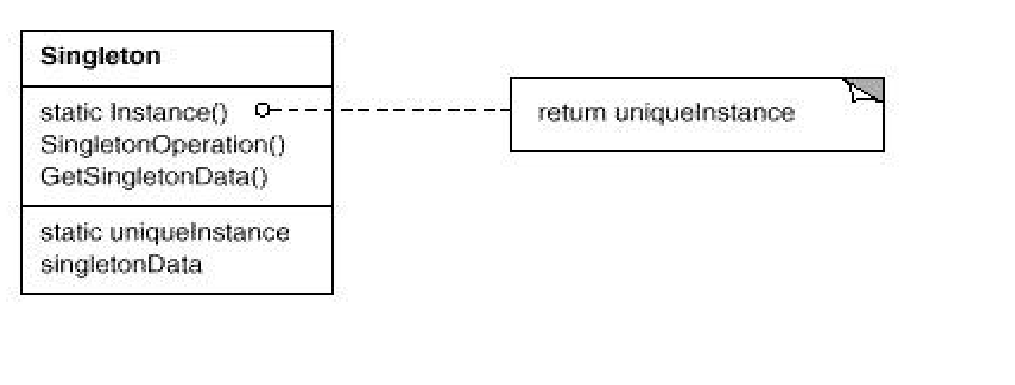
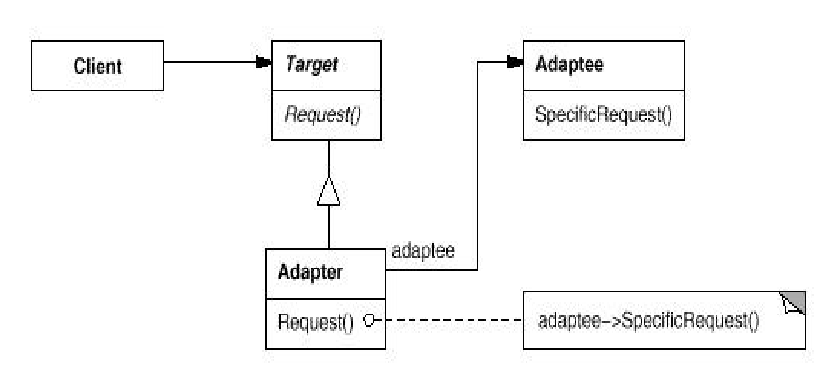
**Singleton Pattern**

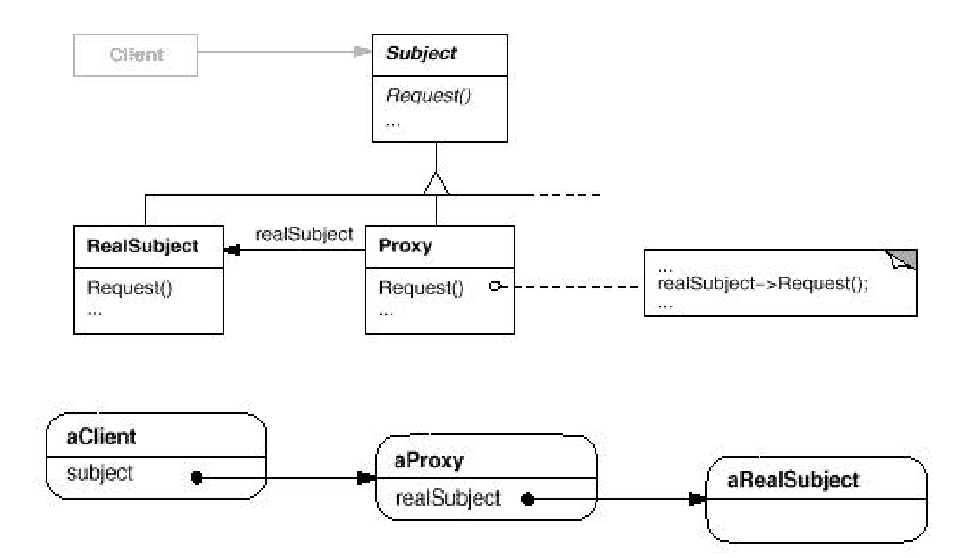
**Intent:** Ensure that a class have only one instance created and provide a global point to access to it.

**Adapter Pattern**

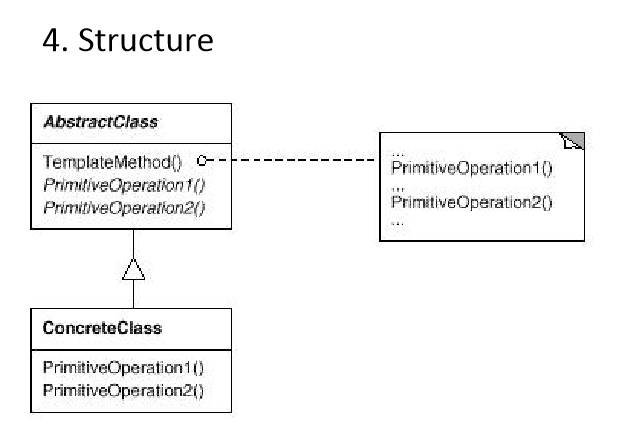
**Intent:** Convert the interface of a class in to another interface client expected and it let incompatible interfaces work togethers.

****

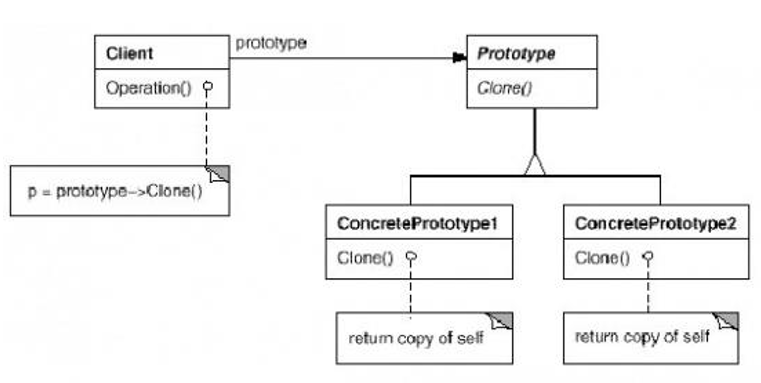
**Proxy Pattern**

**Intent:** Provide surrogate or place holder for another object to control access to it.

**Template Pattern**

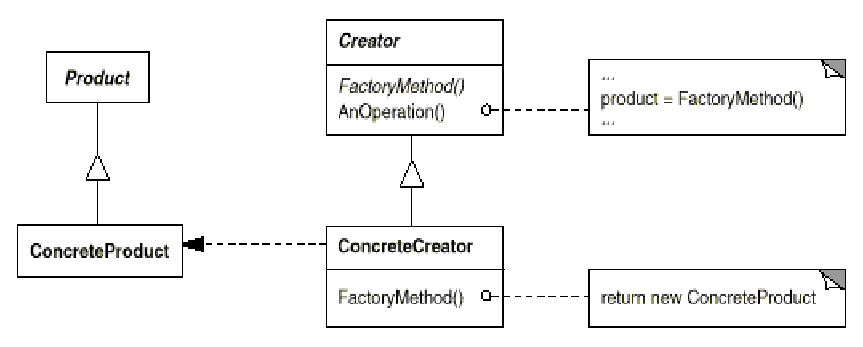
**Intent:** Define the skeleton of the algorithms in an operation and deferring some steps to subclass without changing the algorithms structure.

**Prototype Pattern**

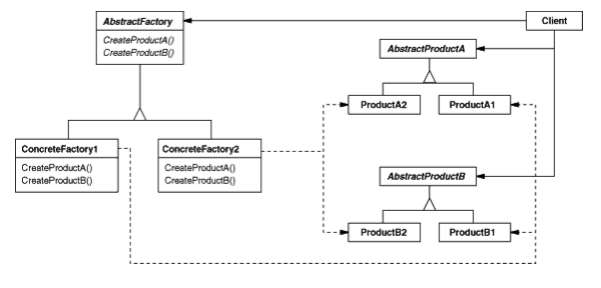
**Intent:** Specifies the kind of object from prototypical instance, and create new object by copying from this prototype.

**Factory Method Pattern**

**Intent:** Define an interface for creating object but let subclass decide which class to instantiate.



**Abstract Factory Pattern**

**Intent:** Provide an interface for creating families of related or dependent without specifies their concrete class.

**Chain of Responsibility Pattern**

**Intent:** Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it.

