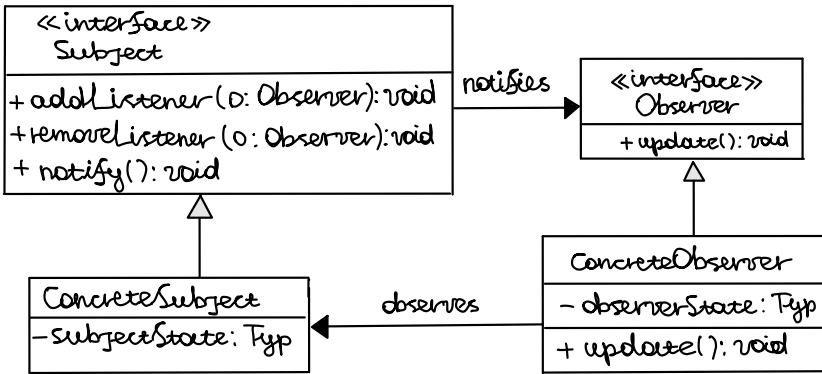


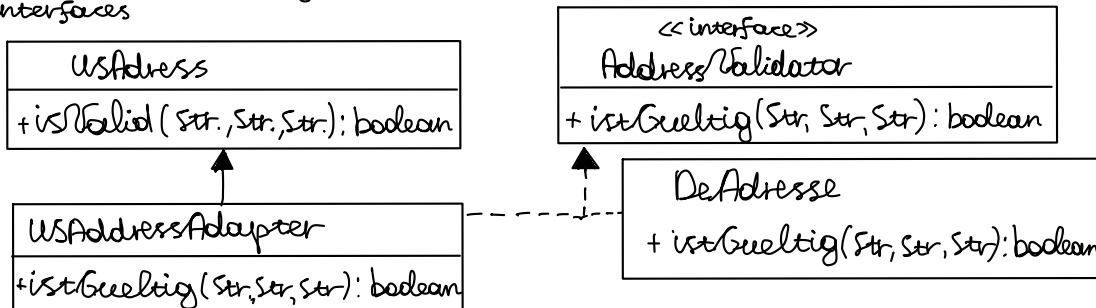
Observer

Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automat.



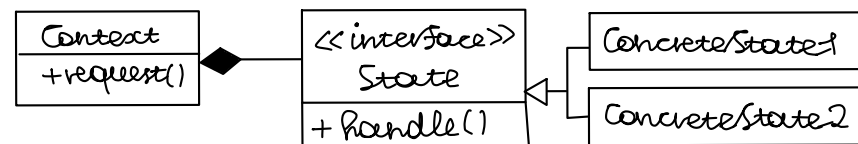
Adapter

Convert the interface of a class into another interface clients expect. Lets classes work together that couldn't otherwise because of incompatible interfaces



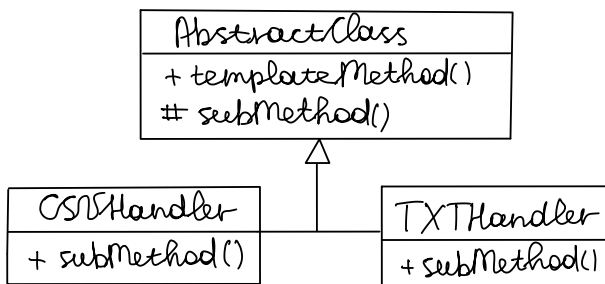
State

Allow an object to alter its behavior when its internal state changes. The object will appear to change its class



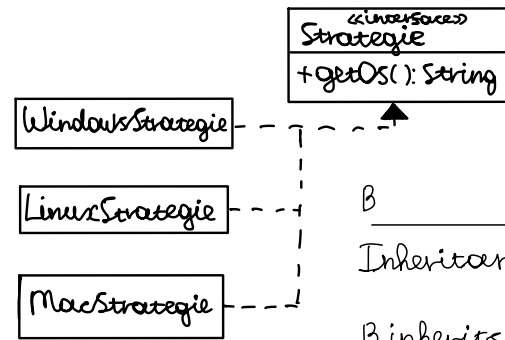
Template

Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure.



Strategy

Define a family of algorithms encapsulate one, and make them interchangeable. Lets the algorithm vary independently from clients that use it.



B A
Inheritance

B inherits from A.
Creates an "is-a" relation-ship. A is a generalization

B A
Implementation
B is a concrete impl. of A

A B
Association
A and B call each other

A B
One way association

A can call B's props/methods, but not vice versa.

A B
1 1...*

Aggregation

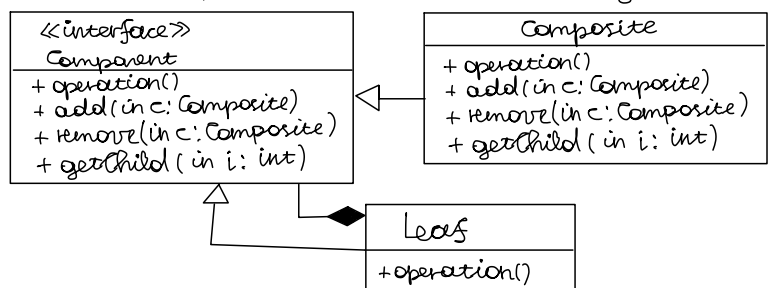
A has 1 or more instances of B. B can survive if A is disposed

A B
1 1

A has 1 or more instances of B. B can survive if A is disposed

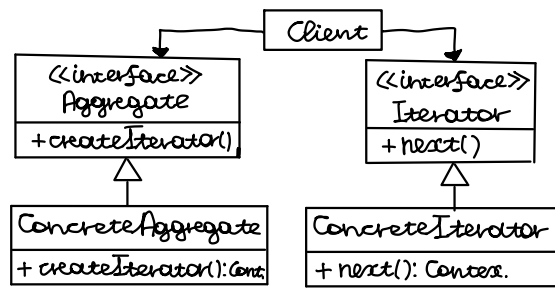
Composite

Compose objects into tree structures to represent part-whole hierarchies. Lets clients treat individual objects and compositions of objects uniformly



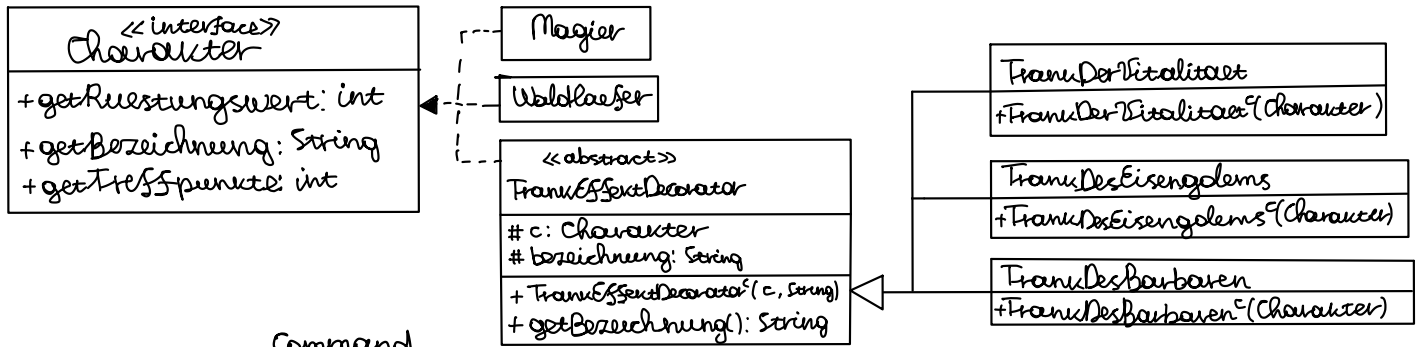
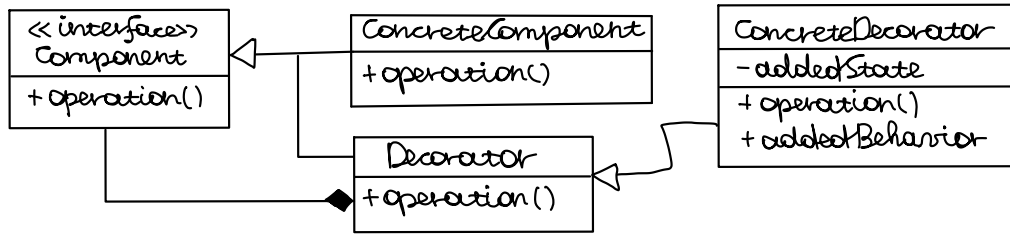
Iterator

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation



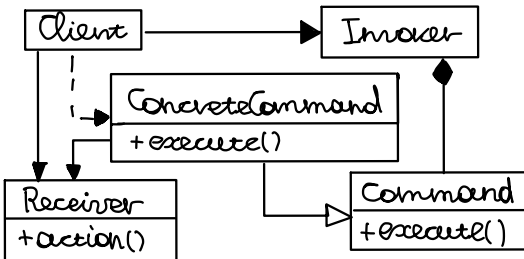
Decorator

Attach additional responsibilities to an object dynamically. Provide a flexible alternative to sub-classing for extending functionality



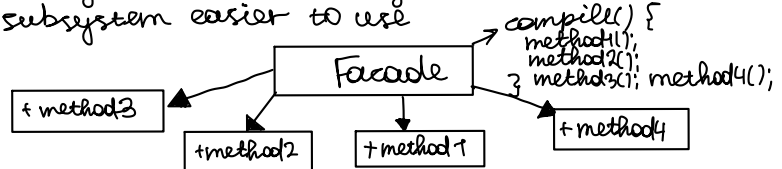
Command

Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations.



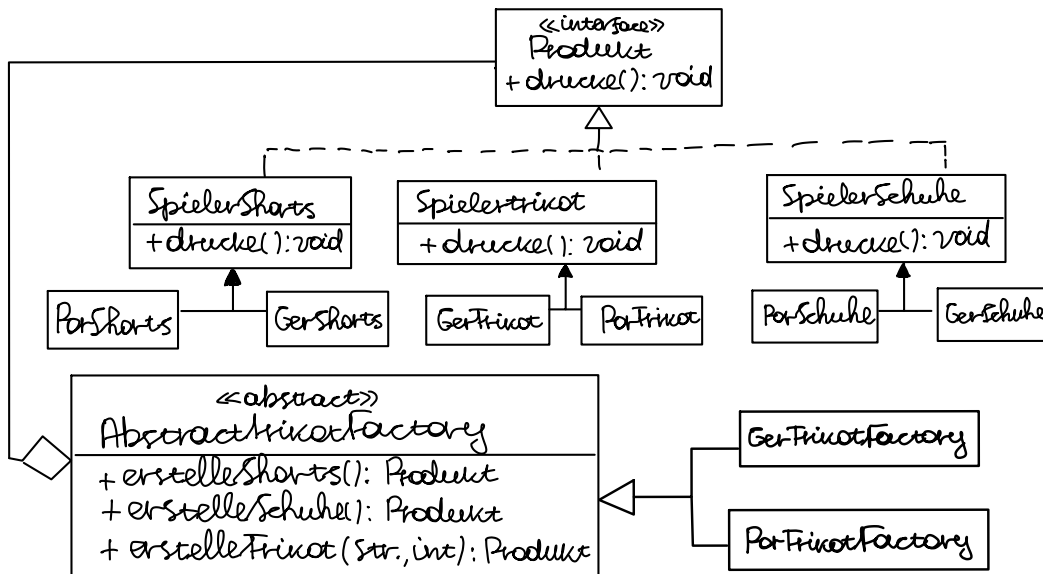
Facade

Provide a unified interface to a set of interfaces in a subsystem. Defines a highlevel interface that makes the subsystem easier to use



Abstract Factory

Provides an interface for creating families of related or dependent objects without specifying their concrete class.



Proxy

Provide a surrogate or placeholder for another object to control access to it.

