Design Patterns
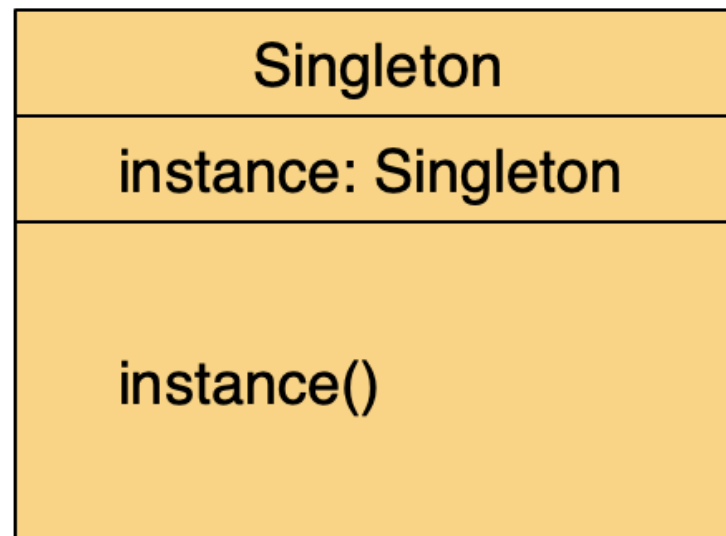
A workshop with examples in Ruby
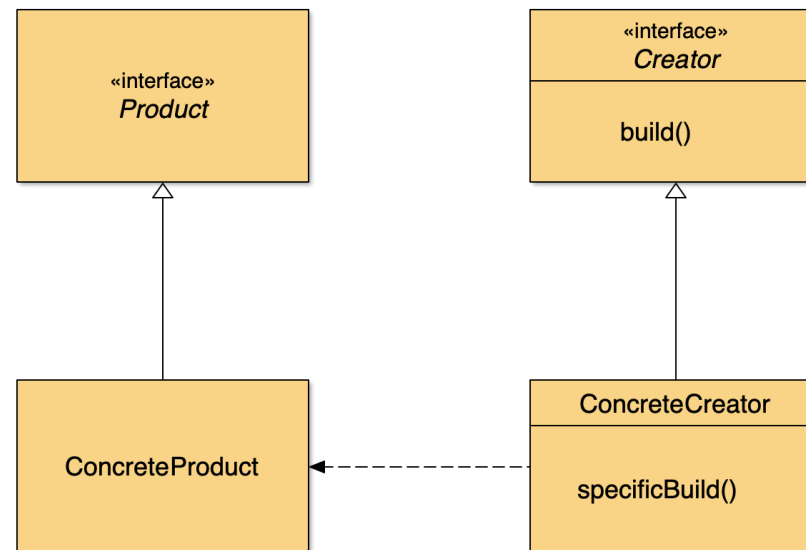
1. Singleton *(Creational)*

Restricts object creation for a class to only one instance
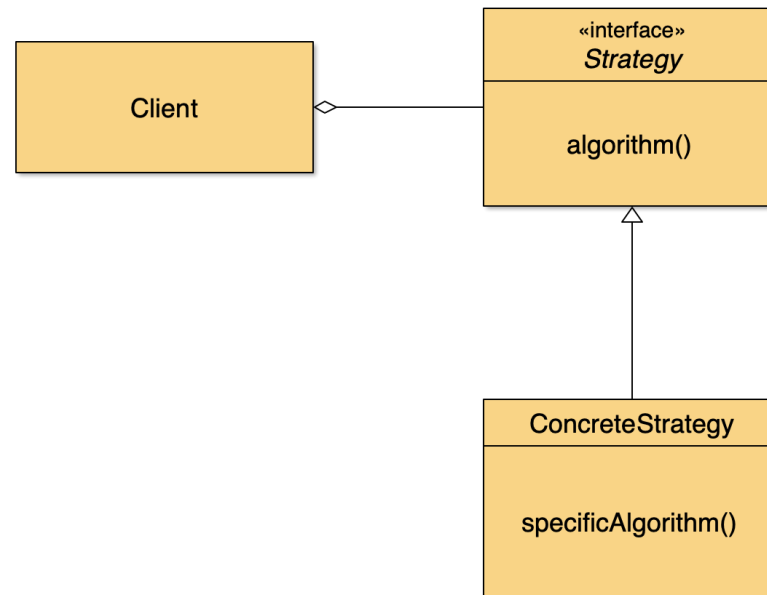
Singleton
instance: Singleton
instance()

2. Factory Method *(Creational)*

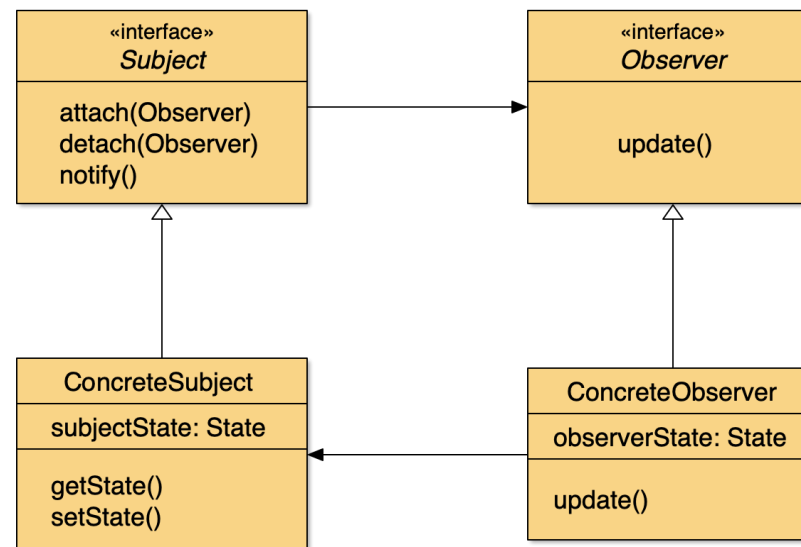> Creates objects without specifying the exact class to create

3. Strategy *(Behavioural)*

> Allows one of a family of algorithms to be selected on-the-fly at runtime
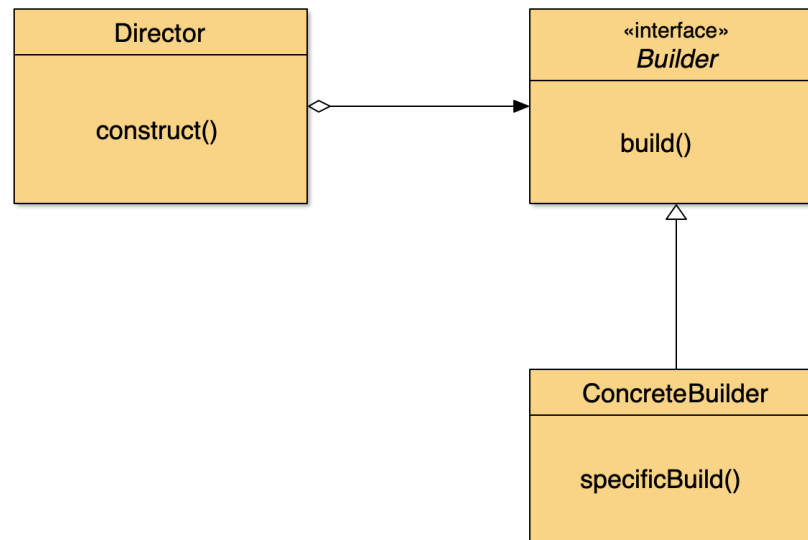
4. Observer *(Behavioural)*

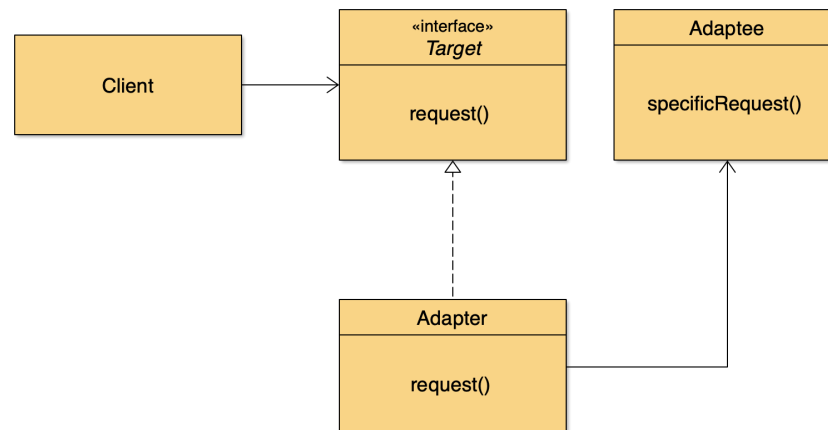A publish/subscribe pattern which allows a number of observer objects to see an event

5. Builder *(Creational)*

Constructs complex objects by separating construction and representation

6. Adapter *(Structural)*

Allows classes with incompatible interfaces to work together by wrapping its own interface around that of an already existing class

7. State *(Behavioural)*

Allows an object to alter its behaviour when its internal state changes