

Design Defects and Restructuring

Lecture 11

Sat, Dec 4, 2021

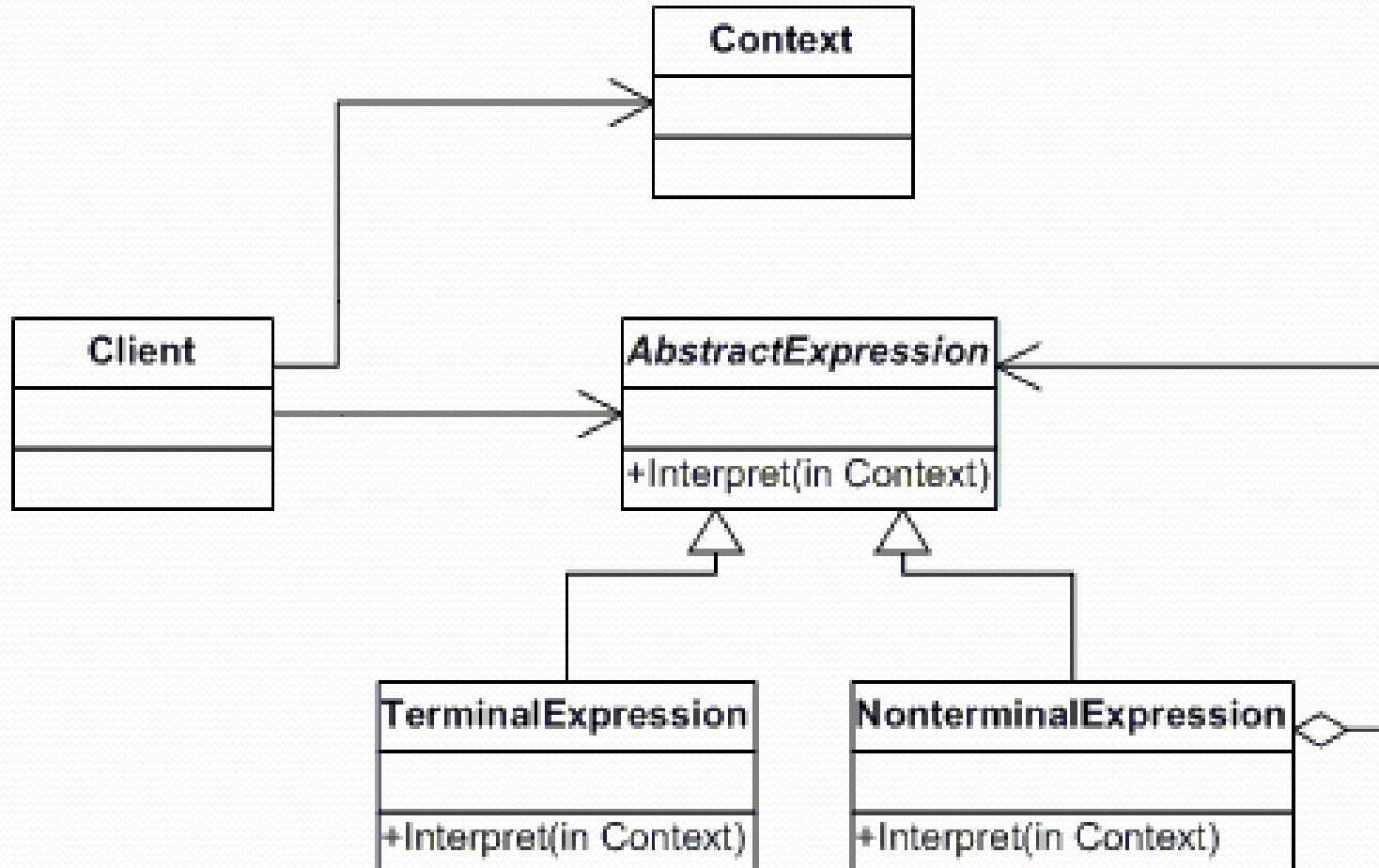
Behavioral Patterns

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

Interpreter

- Intent
 - Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language
- Applicability
 - The grammar is simple
 - Efficiency is not a critical concern

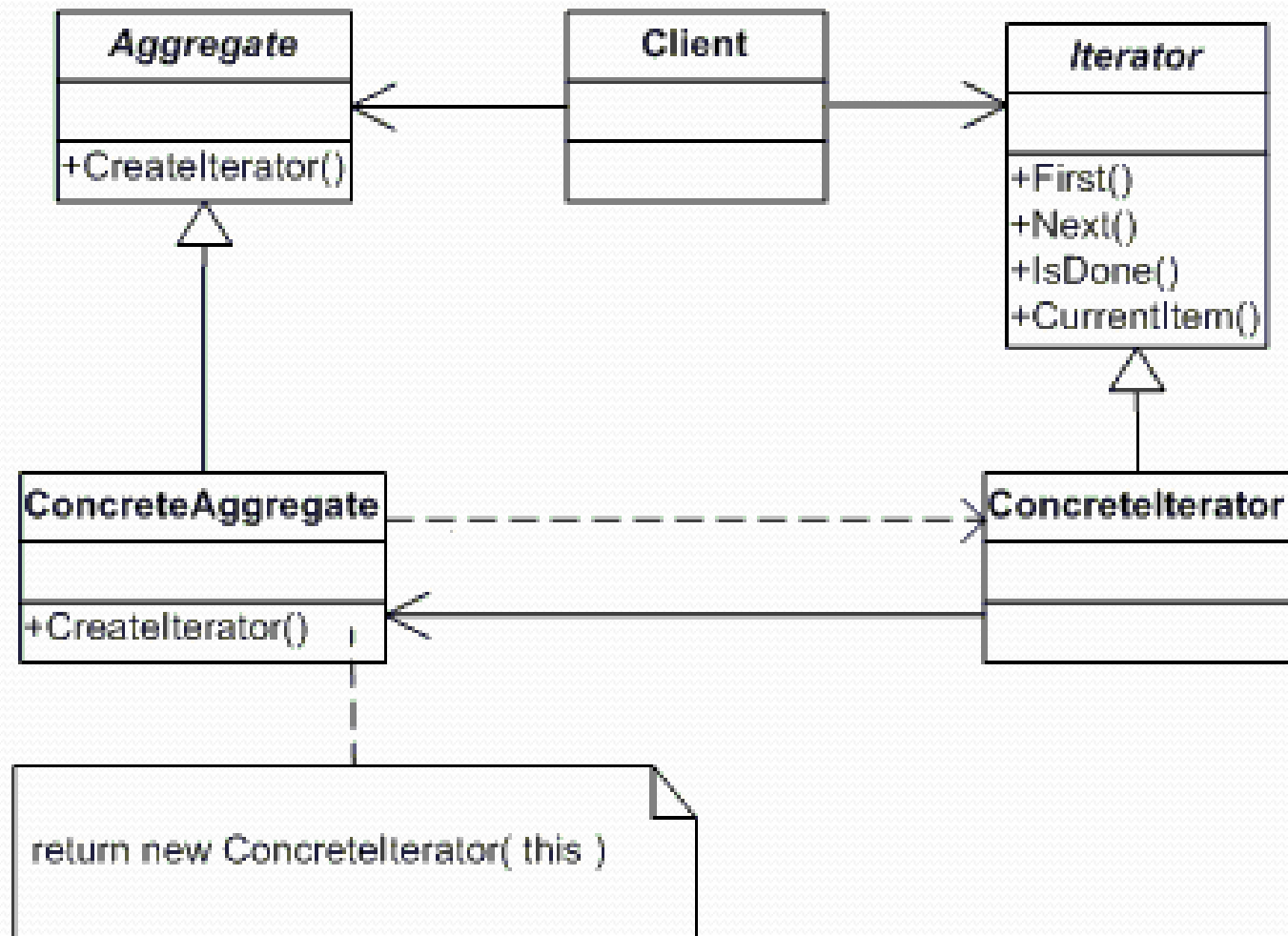
Interpreter



Iterator

- Intent
 - Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation
- Applicability
 - To access an aggregate object's contents without exposing its internal representation
 - To support multiple traversals of aggregate objects
 - To provide a uniform interface for traversing different aggregate structures (to support polymorphic iteration)

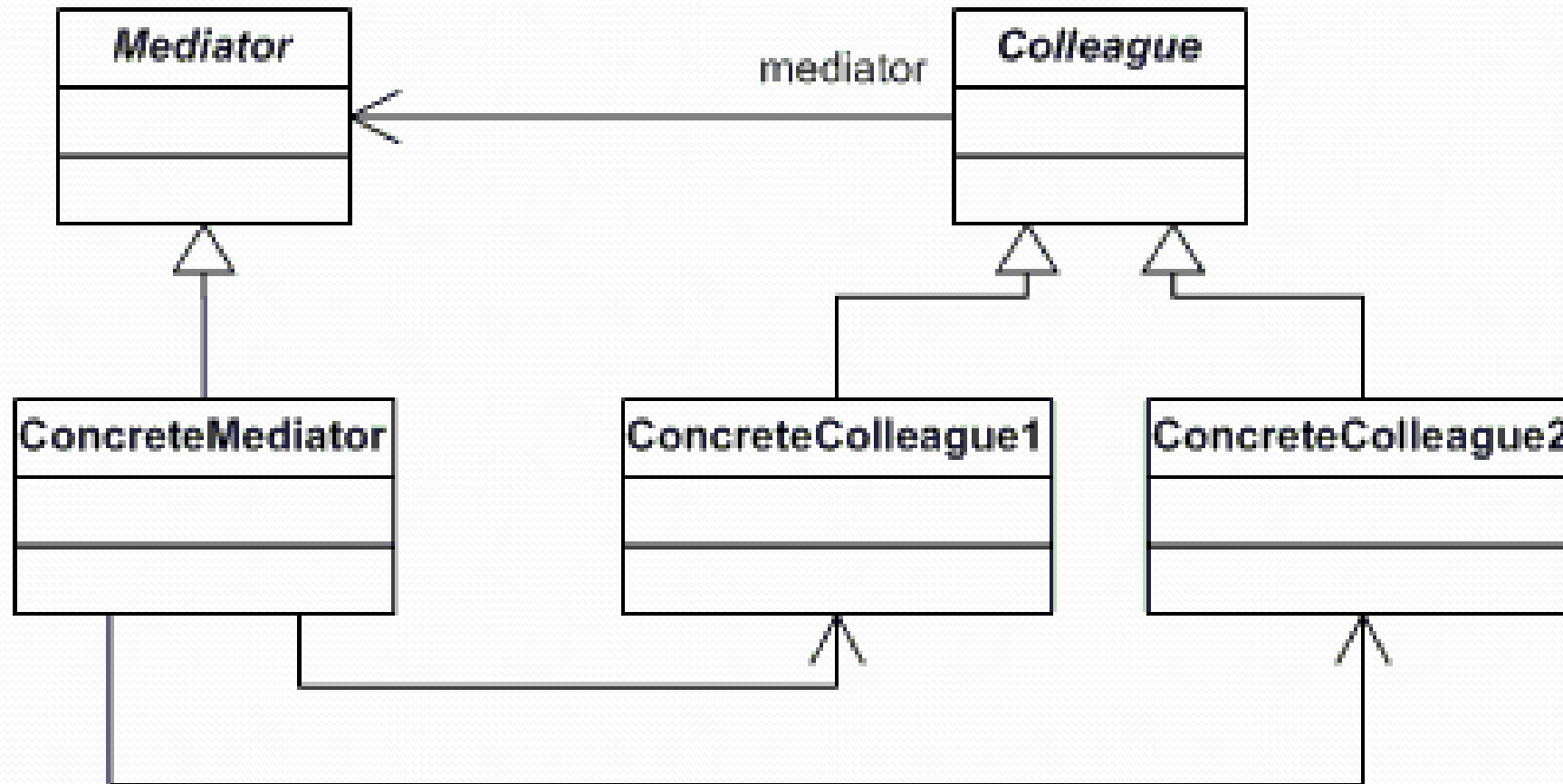
Iterator



Mediator

- Intent
 - Define an object that encapsulates how a set of objects interact
 - Mediator promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently
- Applicability
 - A set of objects communicate in well-defined but complex ways
 - Reusing an object is difficult because it refers to and communicates with many other objects
 - A behavior that is distributed between several classes should be customizable without a lot of subclassing

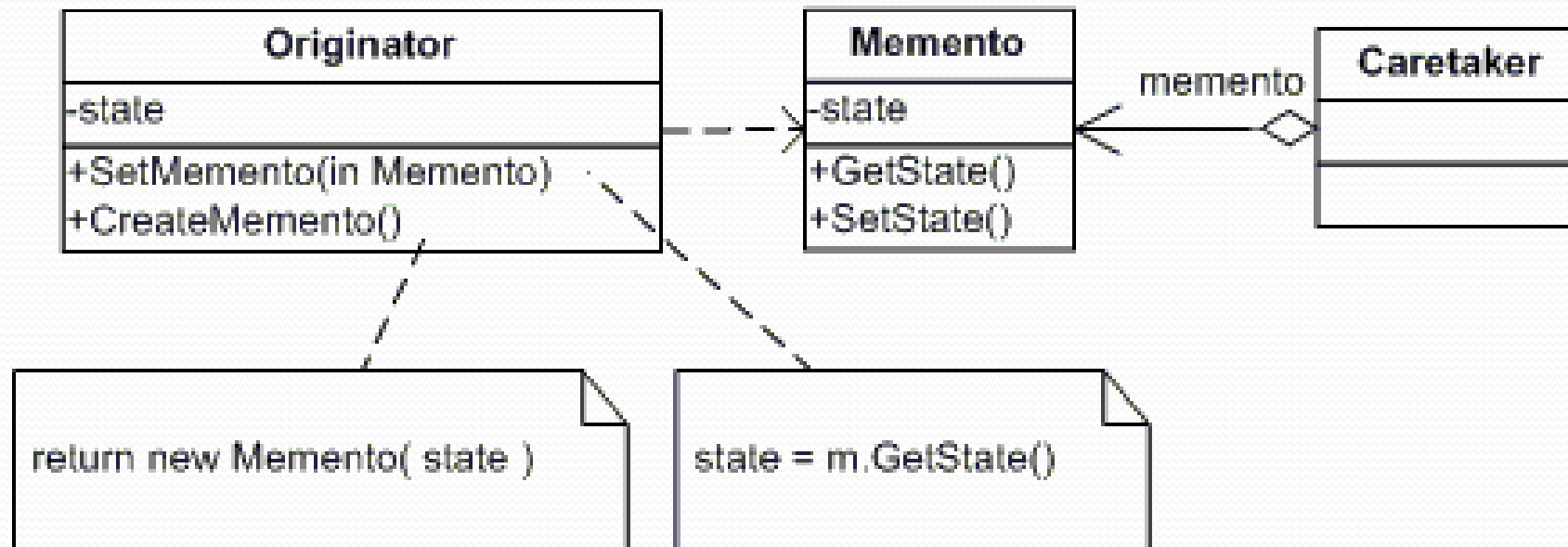
Mediator



Memento

- Intent
 - Without violating encapsulation, capture and externalize an object's internal state so that the object can be restored to this state later
- Applicability
 - A snapshot of (some portion of) an object's state must be saved so that it can be restored to that state later
 - A direct interface to obtaining the state would expose implementation details and break the object's encapsulation

Memento



Observer

- Intent
 - Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically
- Applicability
 - When an abstraction has two aspects, one dependent on the other
 - When a change to one object requires changing others, and you don't know how many objects need to be changed
 - When an object should be able to notify other objects without making assumptions about who these objects are

Observer

