Reuse at Design Level: Design Patterns – II

CS 617- Lecture 18
Mon., 24 March 2008
3:30-5:00 pm

Rushikesh K. Joshi
Department of Computer Sc. & Engg.
Indian Institute of Technology, Bombay
Mumbai - 400 076

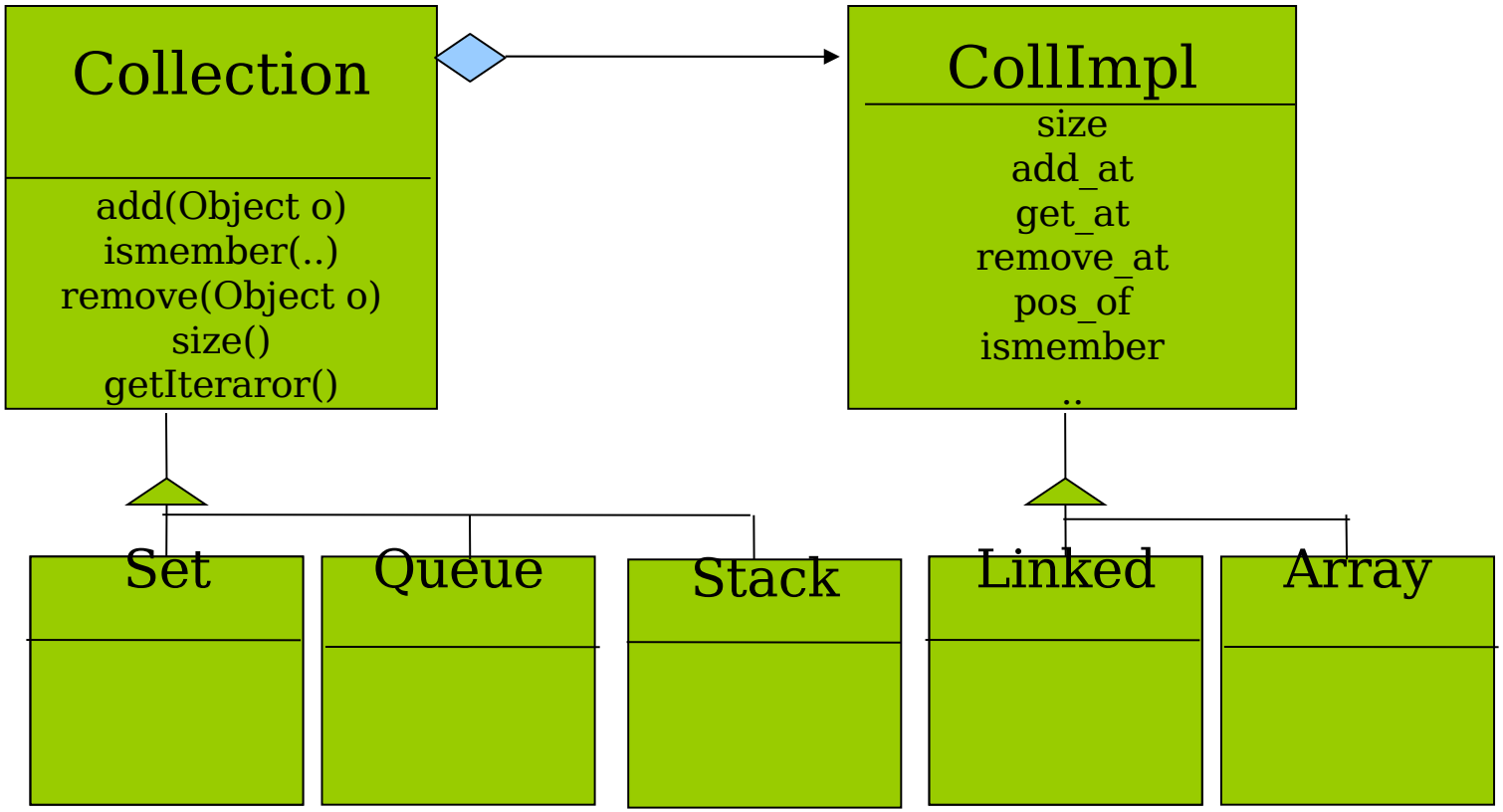
A Structural Pattern

- Bridge
  - Implementation and Abstraction Hierarchies can vary independently

A problem: Varying Implementations and Abstractions

- In a collection hierarchy, one one hand, the abstractions vary
  - collection – set, queue, array
- On the other hand, their implementations may also vary
  - array based, linked list based

A structural pattern: Bridge



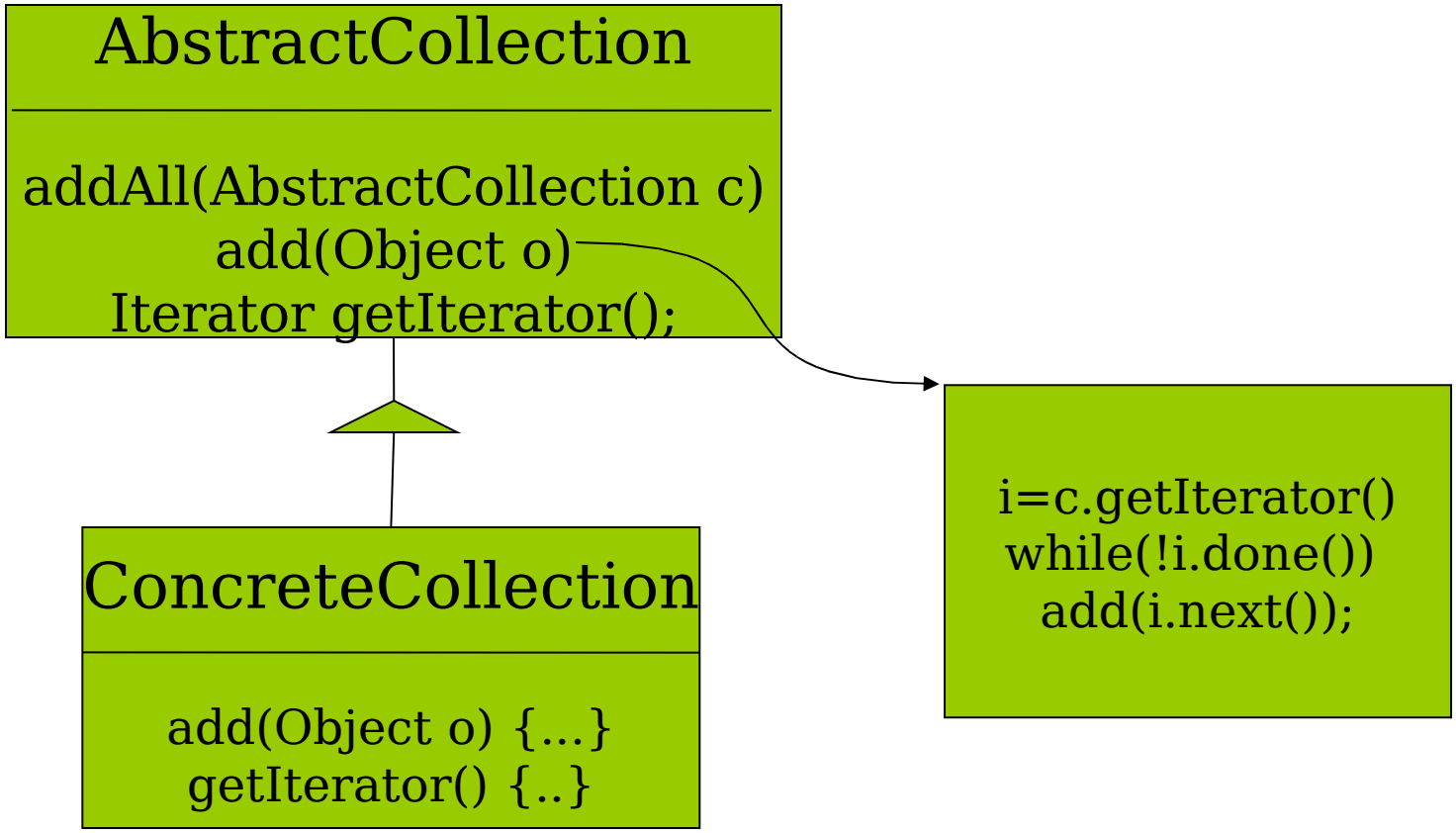
Behavioral Patterns

- Template Method
  - Let certain steps in a superclass be defined by the subclass
- Strategy
  - encapsulate a family of algorithms and make them interchangeable
- Iterator
  - provide accessors for iterating over the elements of an aggregation

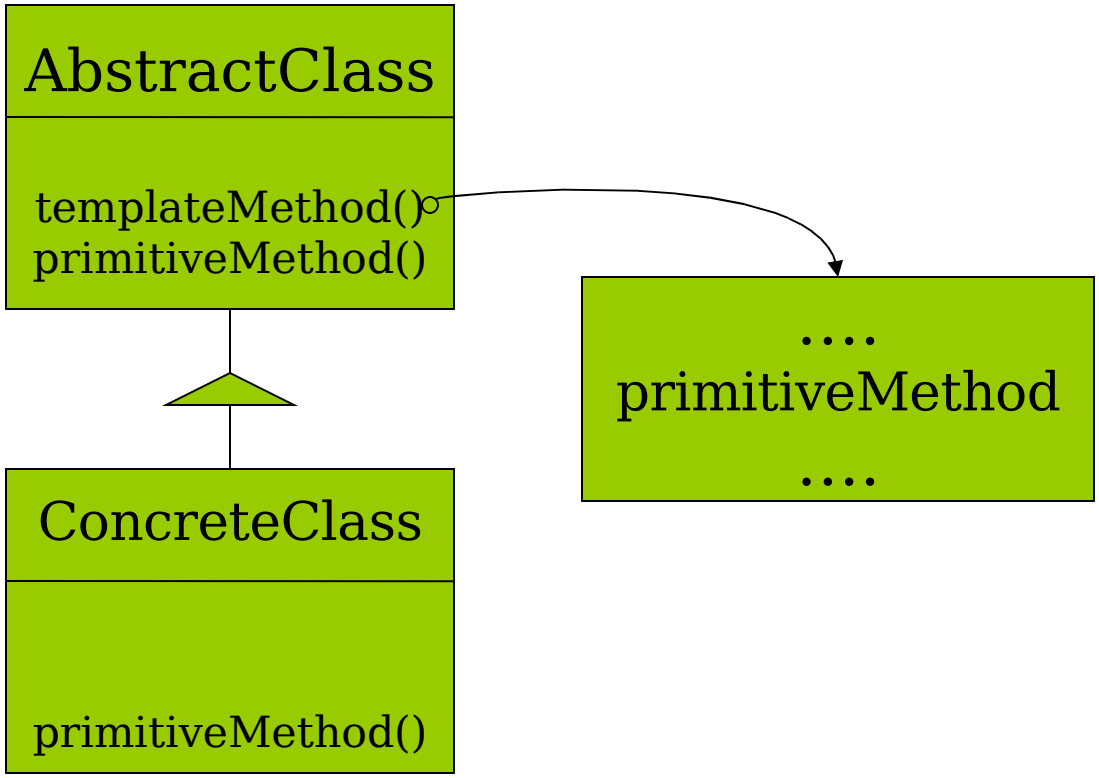
Next problem: Some steps in an implementation can vary

- In a hierarchy of classes, a behavior is quite common for all classes, but only that some steps are dependent on the the nature of these classes
- How to avoid redundancy in defining such behavior
- Where should such a method be located?

In a Hierarchy: Template Method



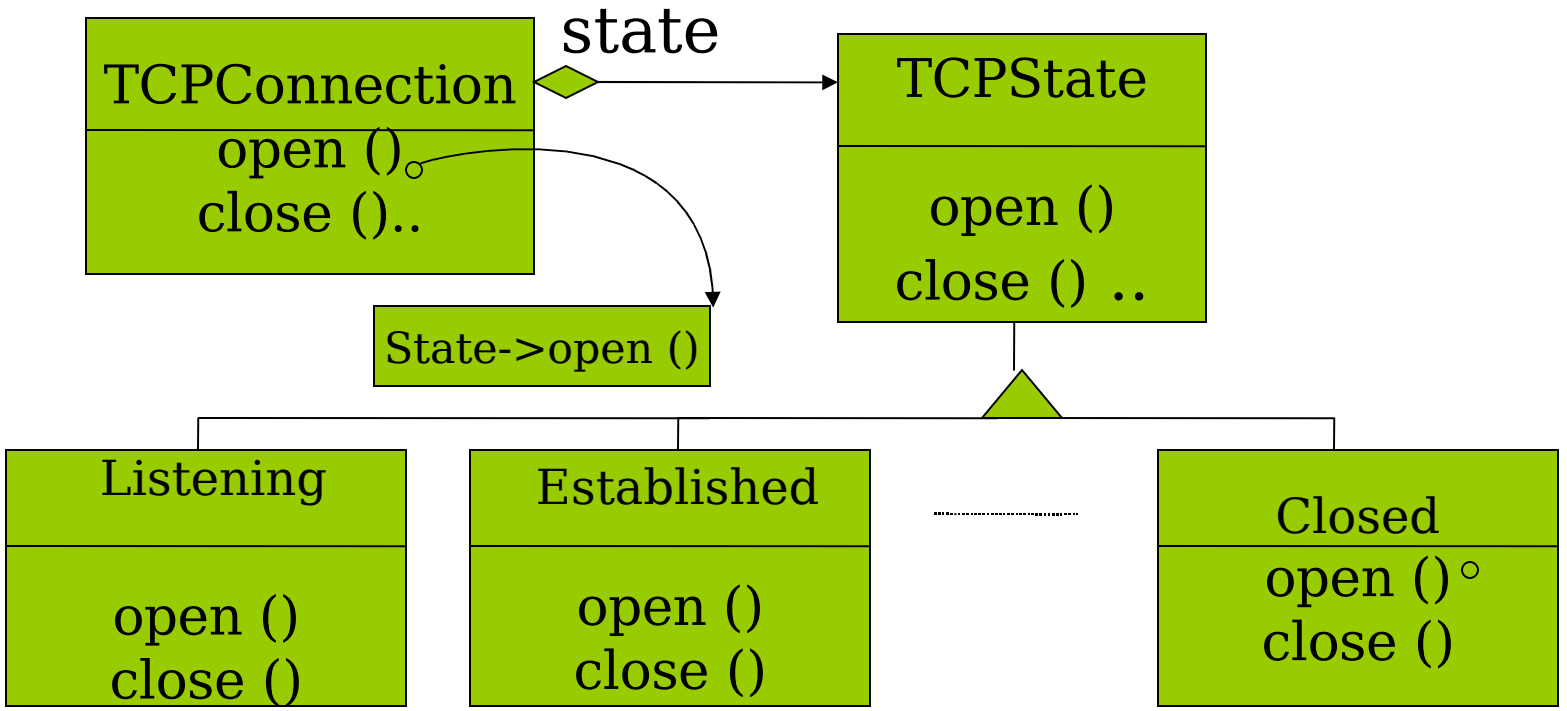
The Template Method Pattern



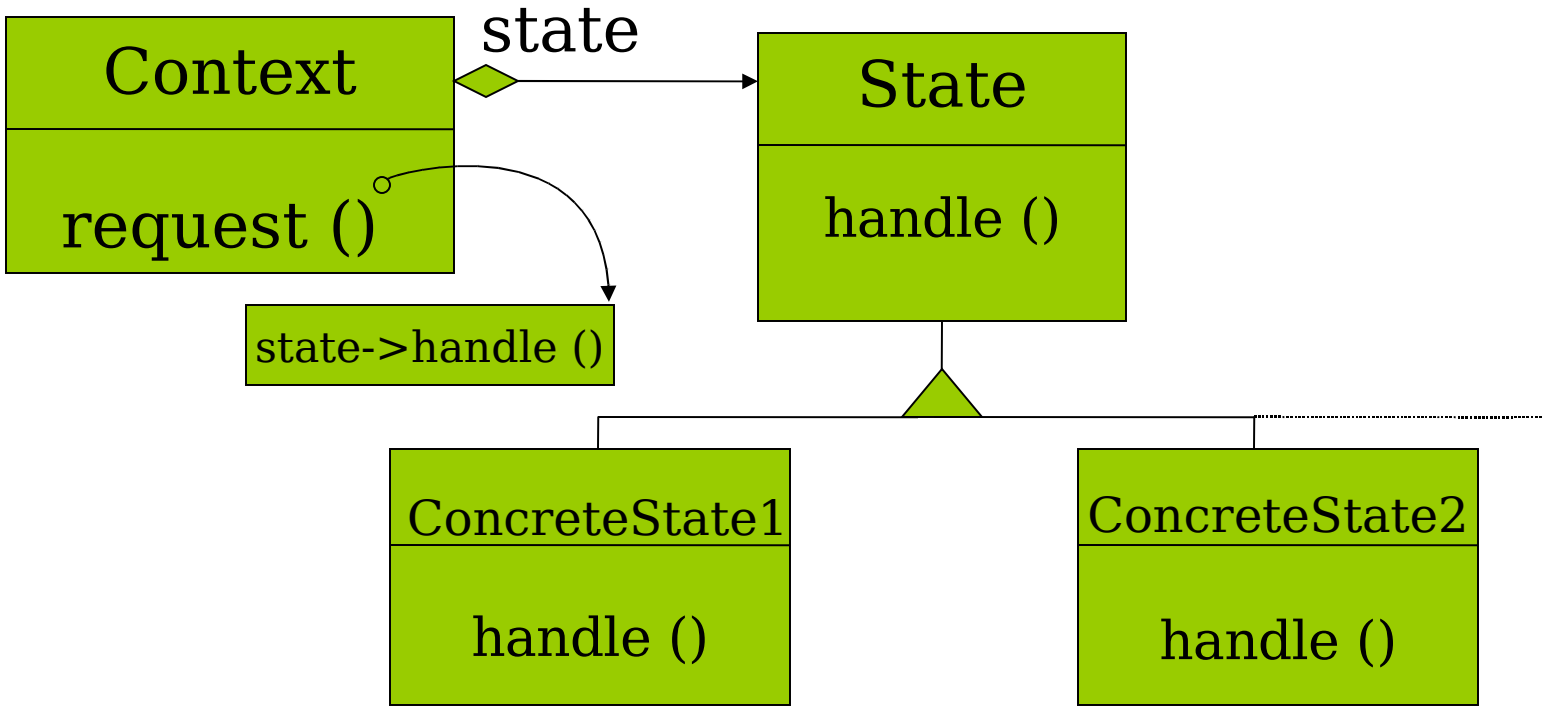
Next problem: An object alters its behavior as it changes its state

Example: A TCP connection object provides methods such as open(), close(), send().. The connection object changes the behavior of these methods as it changes its state from *disconnected* to *listening* to *established* to *closed*

TCP States through the State Pattern



The State Pattern



Next Problem: Use different algorithms for different situations in a given problem context: Strategy Pattern

Example: A document is composed of text. Various line breaking algorithms can be used in formatting the document before printing

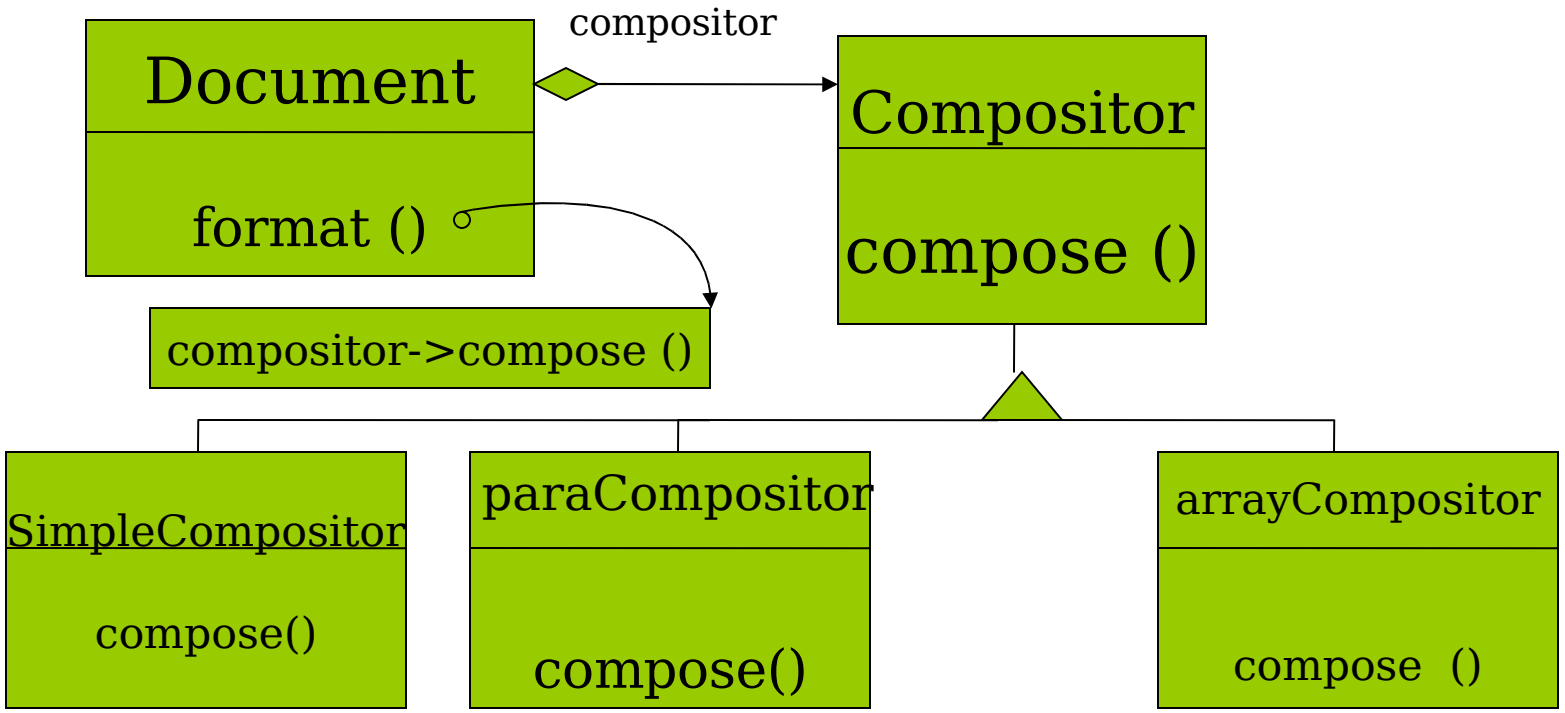
Consider the following strategies:

simple compose: determine line breaks, one line at a time

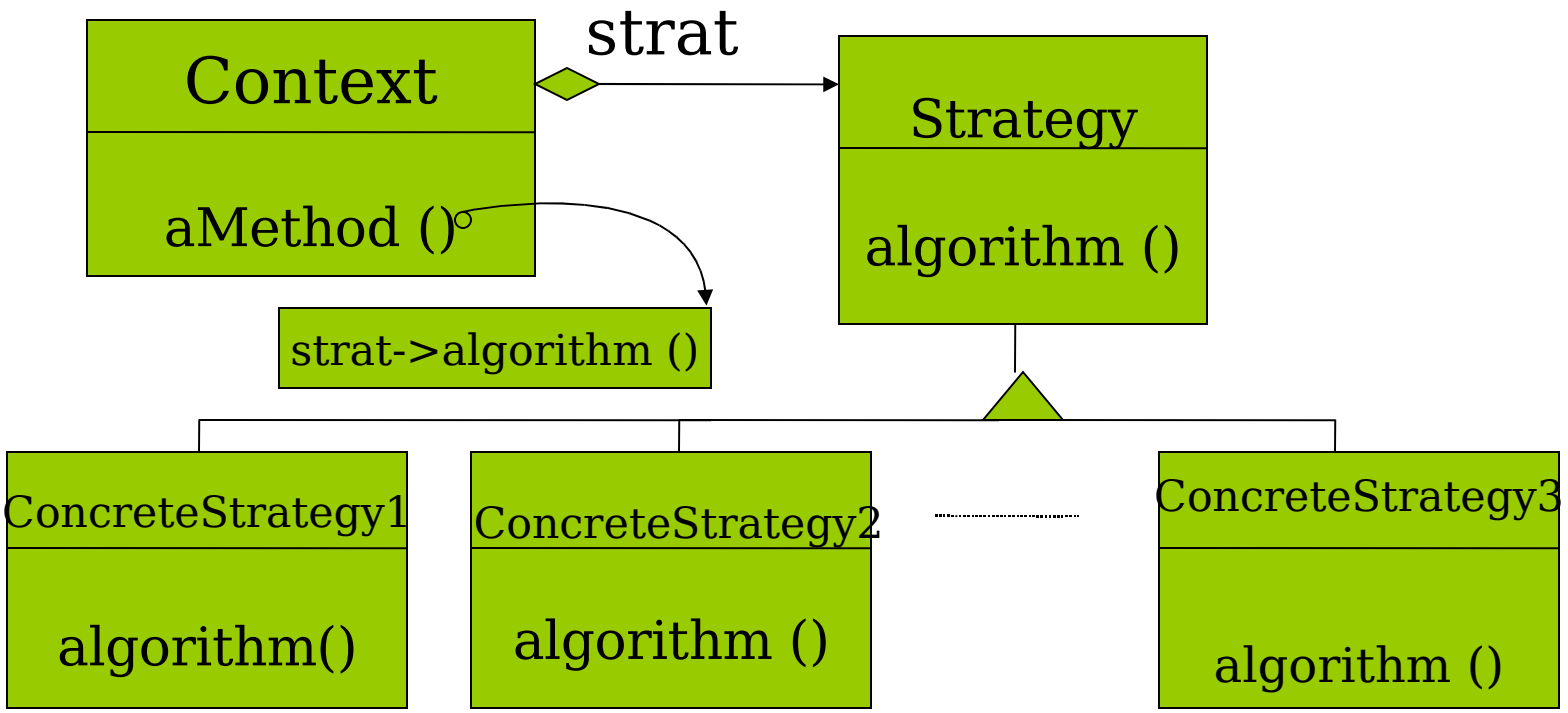
para compose: consider lines in an entire paragraph

array compose: each row has a fixed number of letters

The Solution



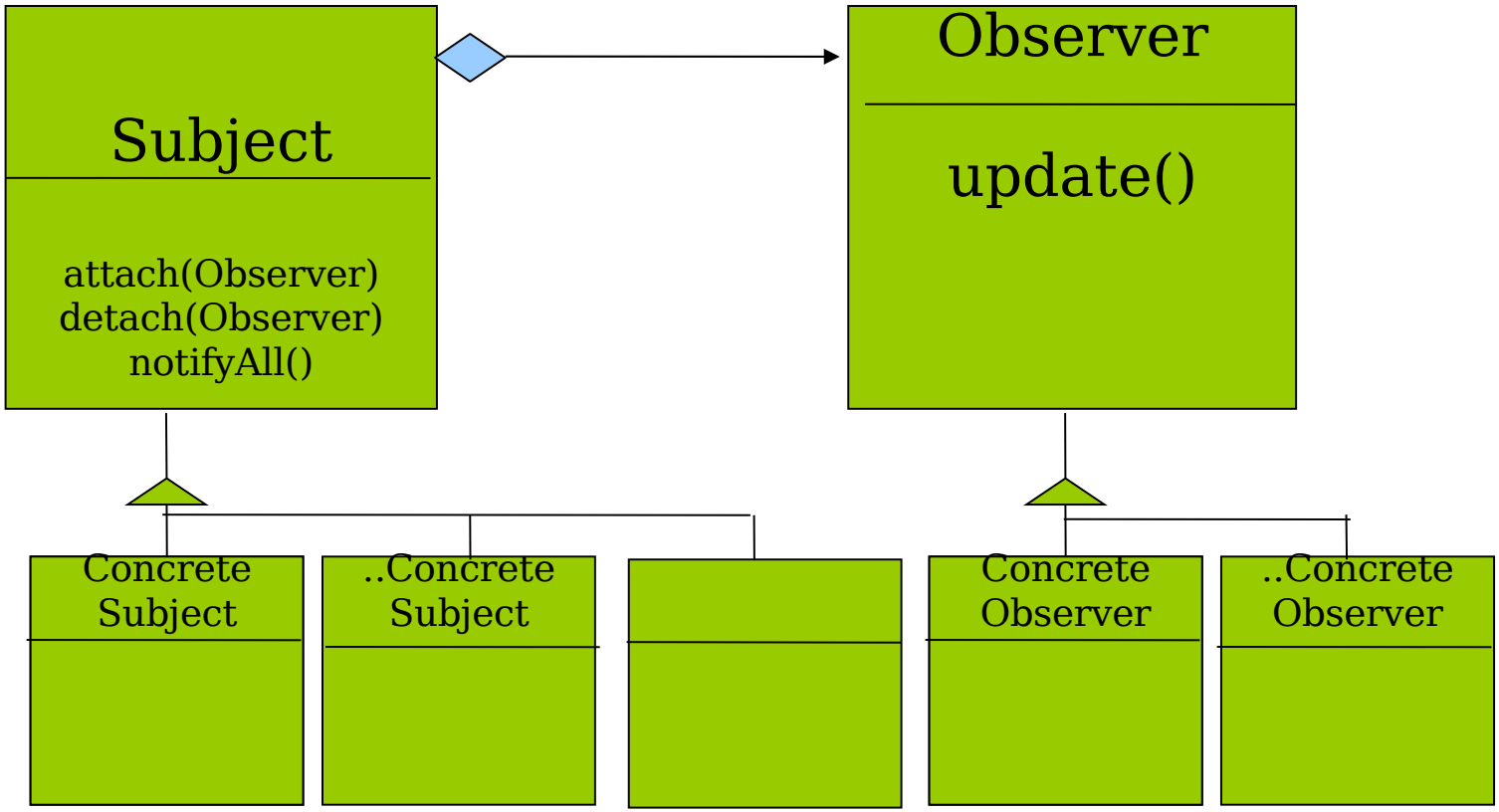
The Strategy Pattern



Problem: When an object changes its state, its dependents are updated

- 1-* dependency between observed and observers
- One observed, Many observers

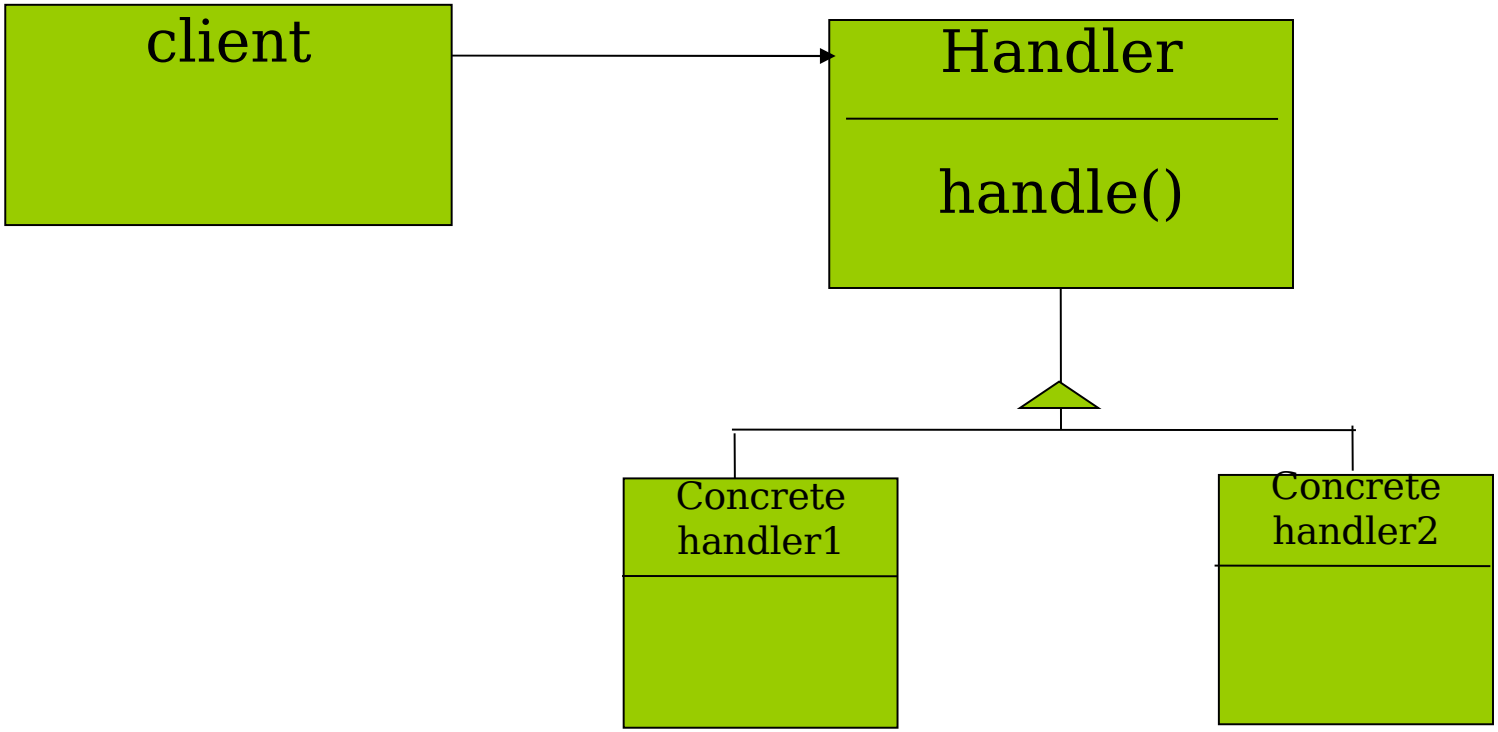
The Observer Pattern



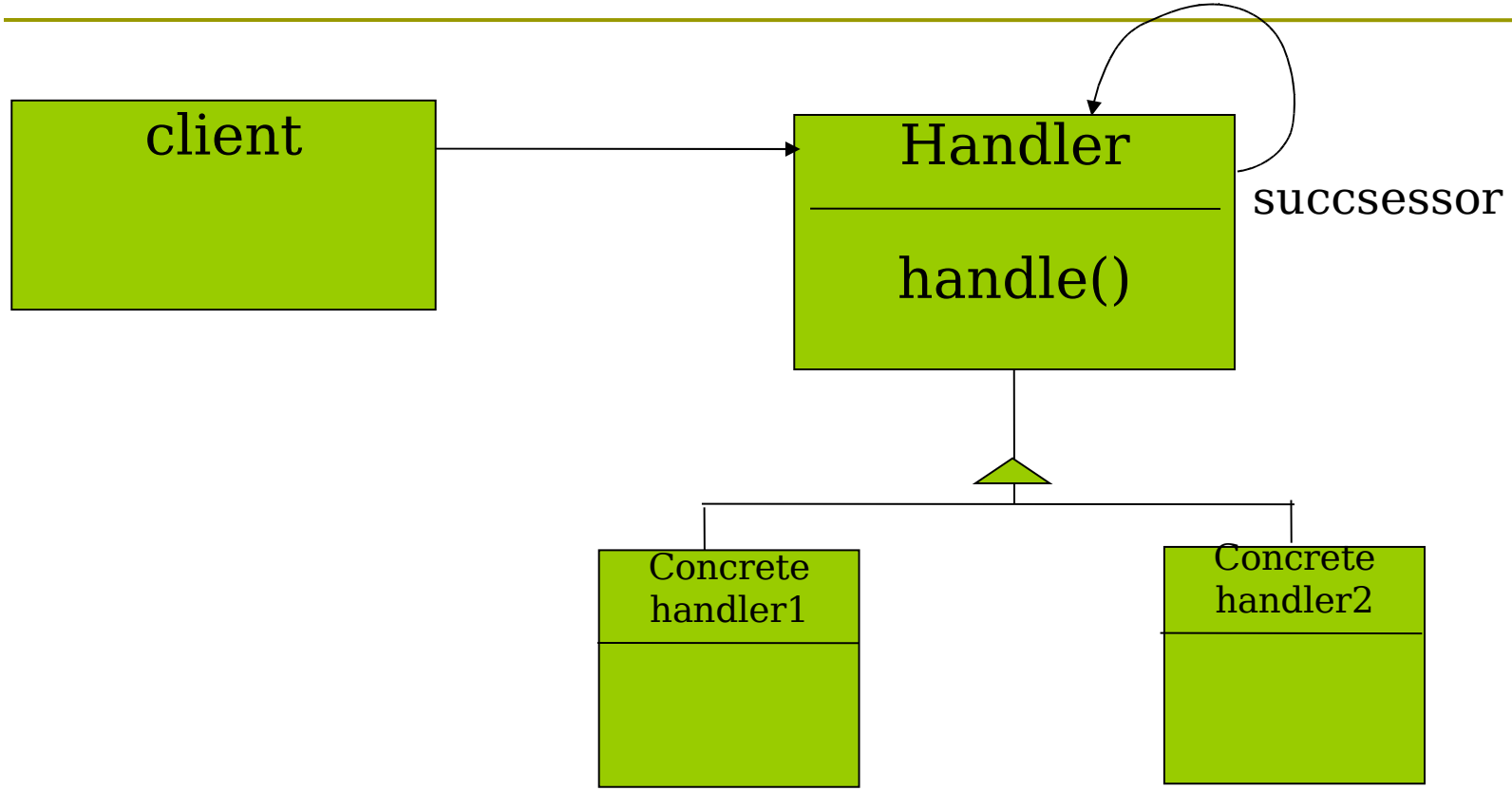
A Problem: one of many request handlers

- A request may get handled by one of many objects
- The sender may not exactly which one can handle the request
- One way is to chain the handlers..

Example: Chain of Responsibility?



Example: Chain of Responsibility



References

[1] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns, Addison-Wesley, 1995

[2] W. Pree, Design Patterns for Object-Oriented Development, Addison-Wesley, 1995

[3] Linda Rising, The Patterns Handbook, Cambridge University Press, 1998