Design Patterns II Summary

- Chain of Responsibility Pattern
- Composite Pattern

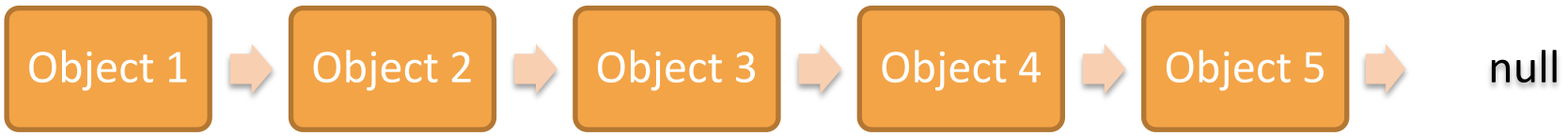
You will also be able to describe a chain of responsibility design pattern and a composite design pattern.

Chain of Responsibility

- Basic idea: a request is passed to the first object in a chain. The first object either handles the request, or passes it to the next object. If it's passed, the second object either handles it or passes it to the third…

- If no object handles the request, an error may occur.

Chain of Responsibility

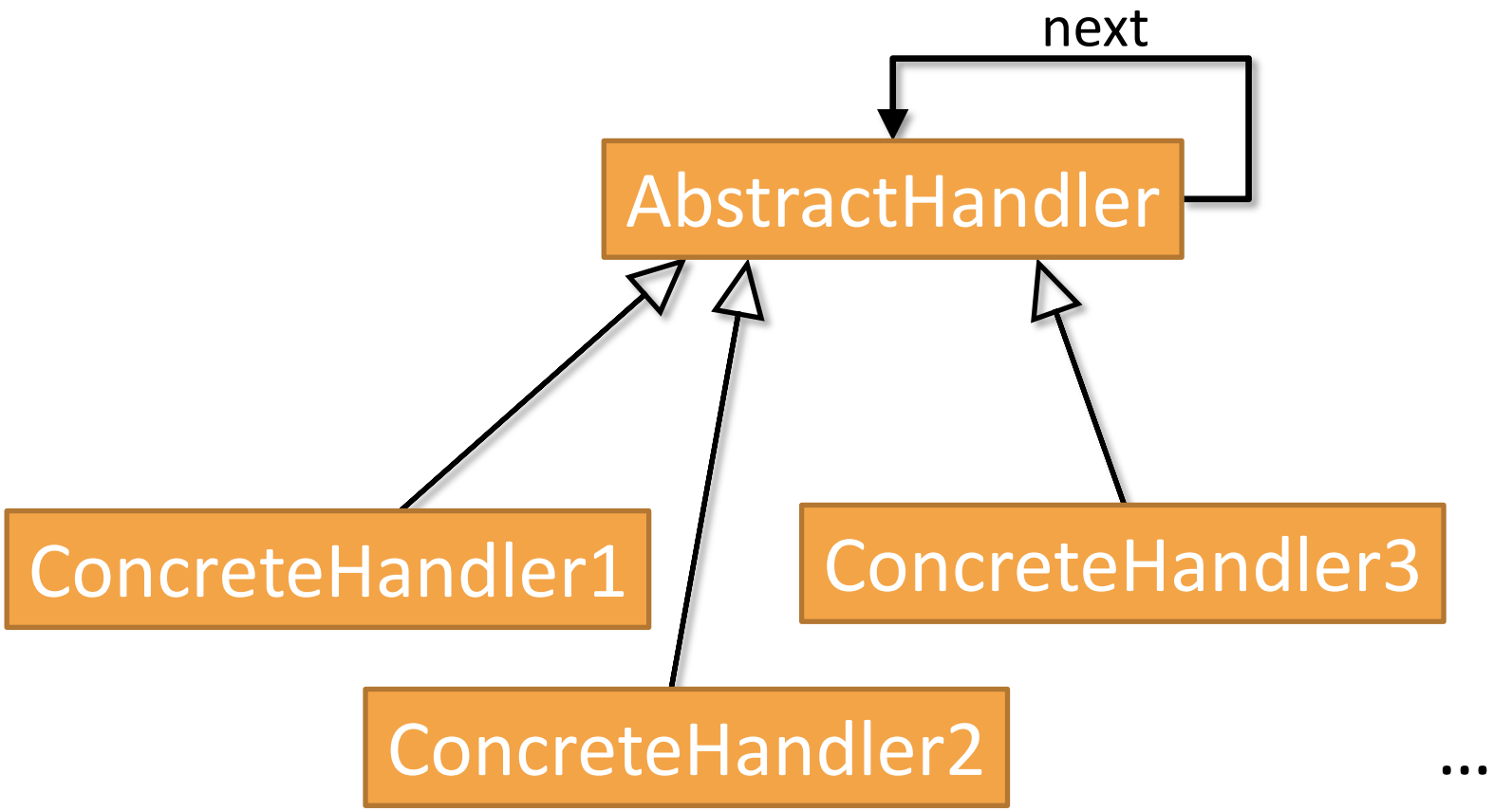


What kind of collection would you use for this set of objects?

What do all of the objects have in common?

How could you generalize this commonality?

Class Model


next
AbstractHandler
ConcreteHandler1
ConcreteHandler2
ConcreteHandler3
...

Is this design pattern creational, behavioural, or structural?

Exercise: Work in groups and identify the chain of responsibility within the Display class for assignment #4.

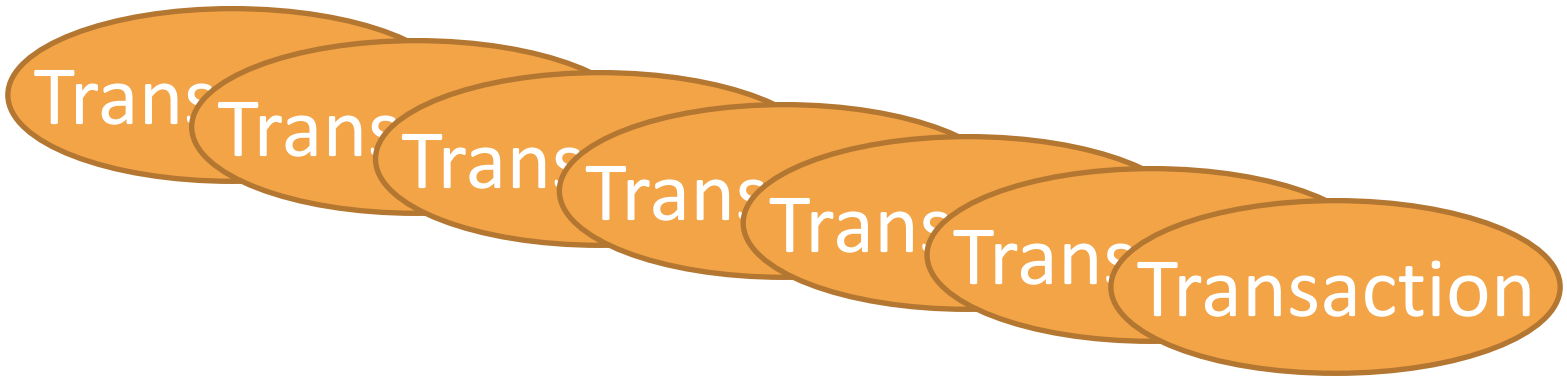
Composite Design Pattern

- Basic idea: groups of objects can be treated in the same way as individual objects.

- Imposes a hierarchy on the object model

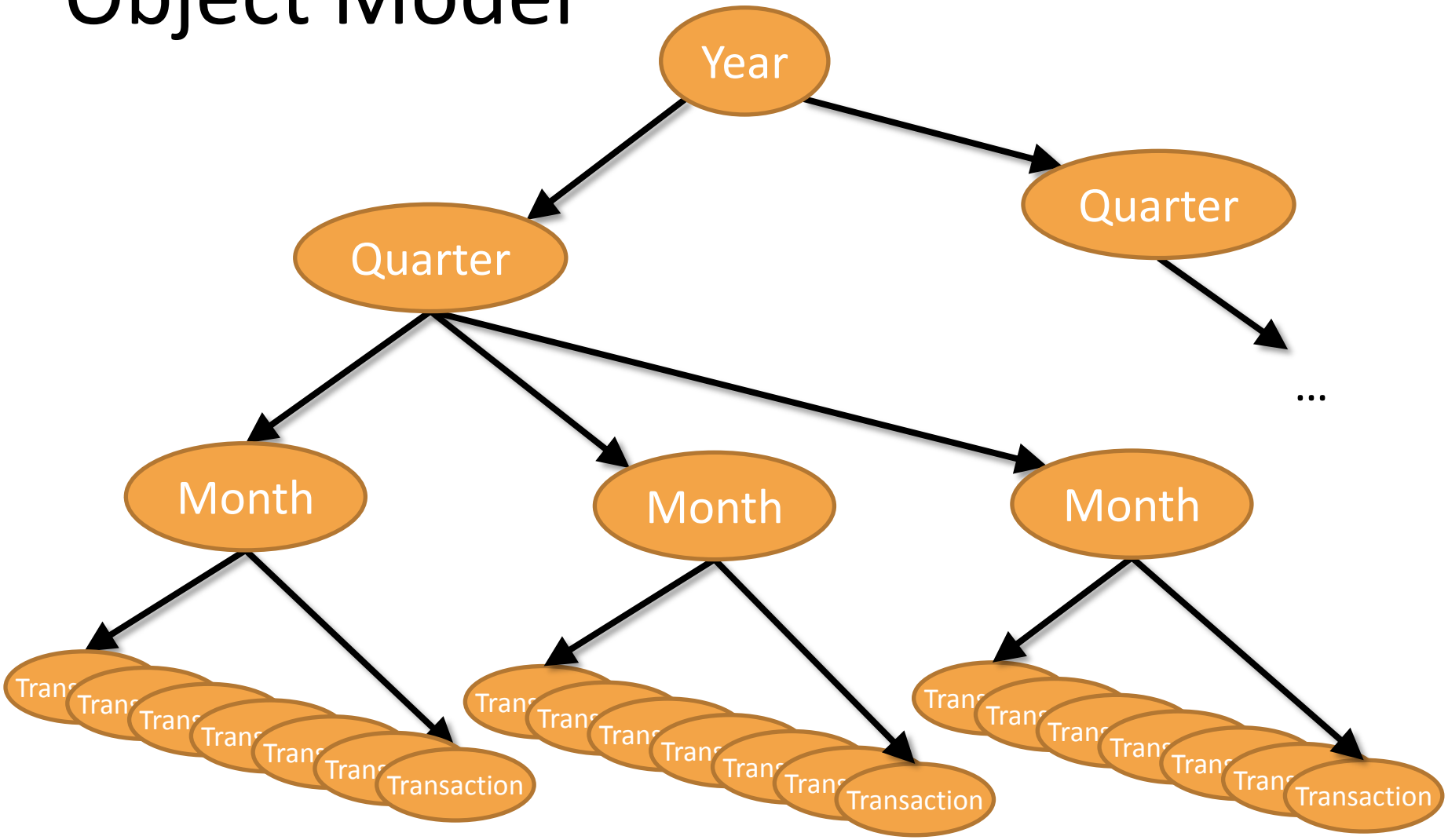
Has anyone ever used the “group” or “ungroup” functionality in a drawing program, or even PowerPoint/Word?

Example: Transactions per Day



How would you compute the revenue for that day?

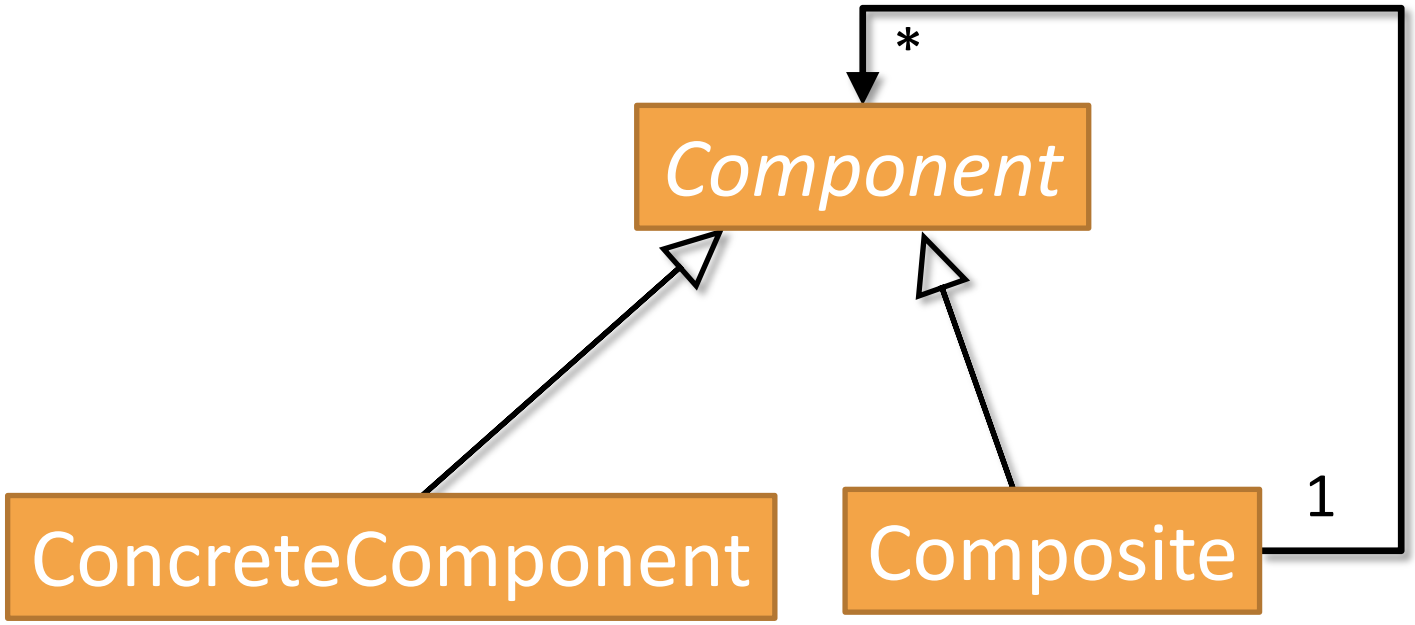
Object Model



How would you calculate the revenue for a month, quarter, or year?

What can we generalize from the month, day, and year classes?

Class Model



Exercise: In groups, use the composite design pattern to create a program that can calculate the amount of energy being used by a specific electronic device, a specific home, a specific building, and/or the city of Calgary.

Part 1: draw the object model

Part 2: create the Java classes for your program (don't implement each method)

Design Patterns II Summary

- Chain of Responsibility Pattern
- Composite Pattern

Assignment #4 Discussion

Display class

- Important Methods:
  - Display constructor
  - pause
  - drawReactor

- You will have to write code that continually performs simulation steps on each reactor component.

Next Class

- Multi-Threading