

Design Patterns II Summary

- Chain of Responsibility Pattern
- Composite Pattern

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

1

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

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

2

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.

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

3

Chain of Responsibility



April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

4

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

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

5

What do all of the objects have in common?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

6

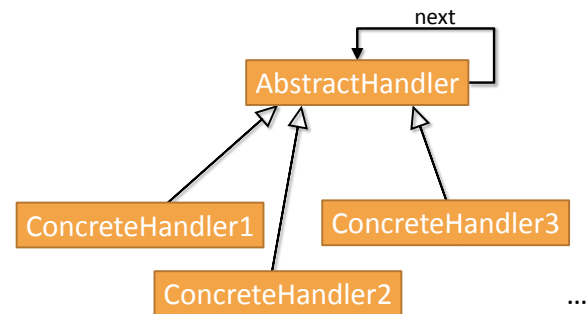
How could you generalize this commonality?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

7

Class Model



April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

8

Is this design pattern **creational**, **behavioural**, or **structural**?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

9

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

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

10

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

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

11

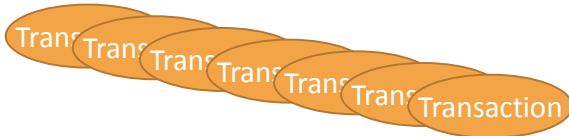
Has anyone ever used the “group” or “ungroup” functionality in a drawing program, or even PowerPoint/Word?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

12

Example: Transactions per Day



April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

13

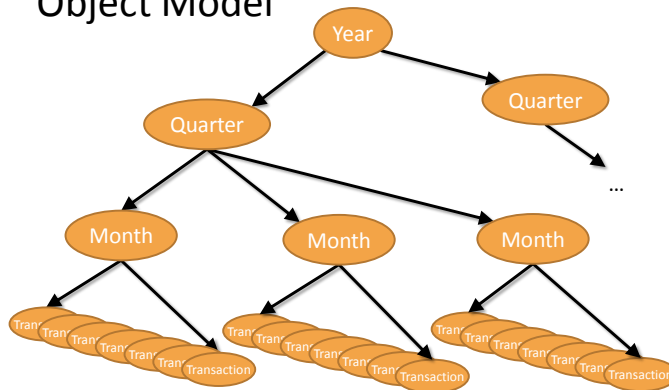
How would you compute the revenue for that day?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

14

Object Model



April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

15

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

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

16

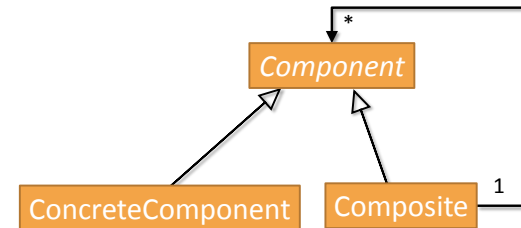
What can we generalize from the month, day, and year classes?

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

17

Class Model



April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

18

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)

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

19

Design Patterns II Summary

- Chain of Responsibility Pattern
- Composite Pattern

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

20

Assignment #4 Discussion

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

21

Display class

- Important Methods:
 - Display constructor
 - pause
 - drawReactor
- You will have to write code that continually performs simulation steps on each reactor component.

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

22

Next Class

- Multi-Threading

April 4, 2009

Slides by Mark Hancock
(adapted from notes by Craig Schock)

23