CISC 323 (Week 10)
Design Patterns

Jeremy Bradbury
Teaching Assistant
March 22, 2005
bradbury@cs.queensu.ca

Midterm

- Questions?????

Assignment #4

- Due Date: Thursday, Mar. 24, 2005 (3:00 pm)
- Second of 3 project assignments:
  - Assignment #3: object-oriented analysis
  - **Assignment #4: detailed project design**
  - Assignment #5: implementation.

Design Patterns

- Creational: creation of objects
- Structural: how classes and objects are put together to form larger structures
  - Adapter
  - Composite
  - Facade
- Behavioral: algorithms and communication between objects
  - Iterator

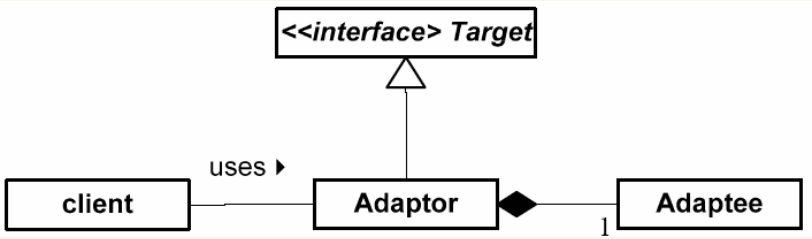
Past Midterm/Exam Questions

- **Question 1, Midterm 2002 (5 pts):** What is the difference between a design pattern and an architectural style?

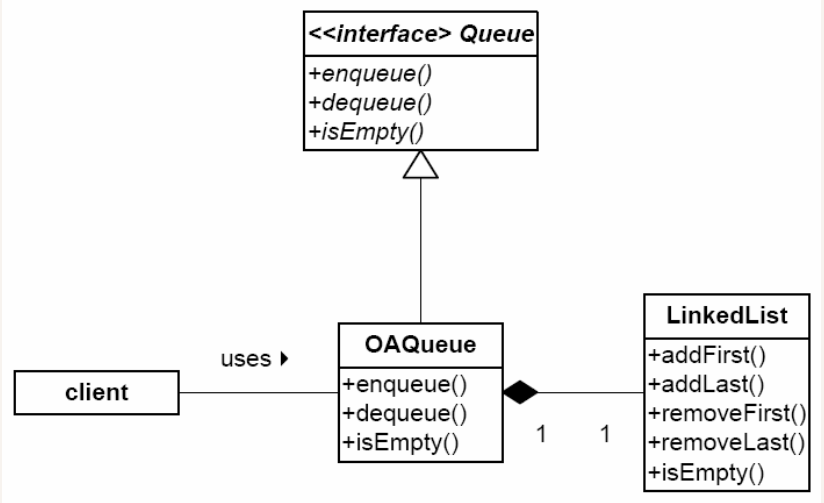
Past Midterm/Exam Questions

- **Answer:** The difference is the scale or level at which they are applied.
  - A design pattern describes a relationship between individual classes to solve a small problem inside a larger system. Design patterns are also called micro-architectures.
  - An architectural style is a way of breaking up a large system into subsystems or components, each of which may contain multiple classes.
  - Each component in a system may use one or more design patterns to structure the relationships between the classes in the component.

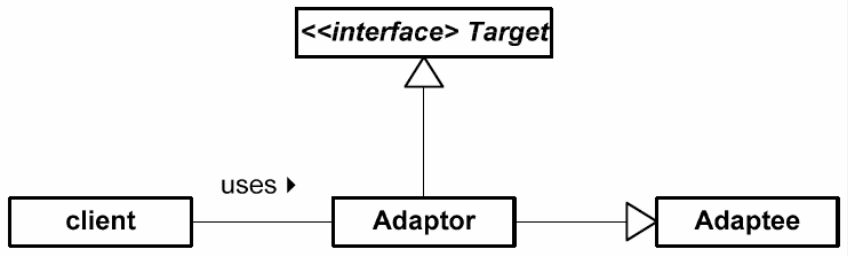
Object Adapter Design Pattern



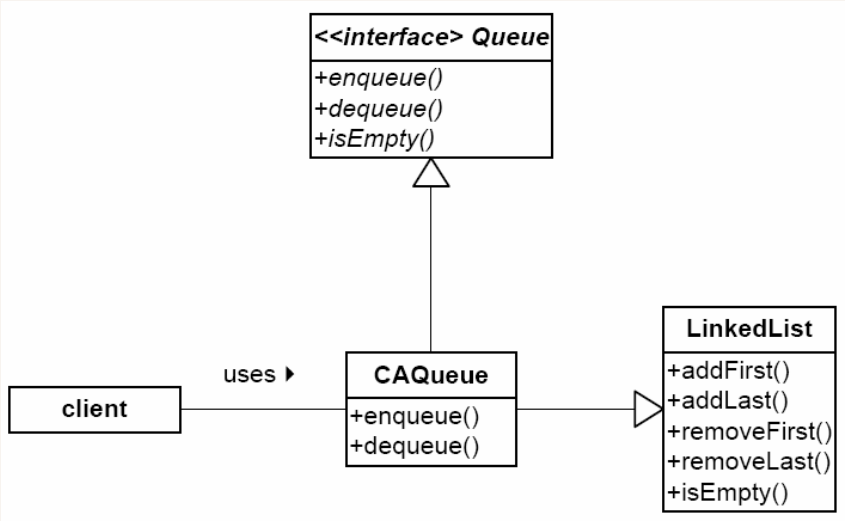
Object Adapter Design Pattern



Class Adapter Design Pattern



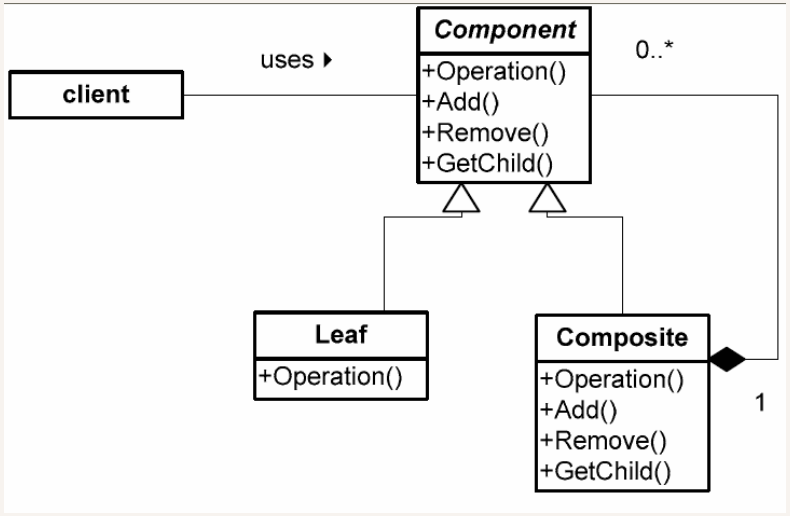
Class Adapter Design Pattern



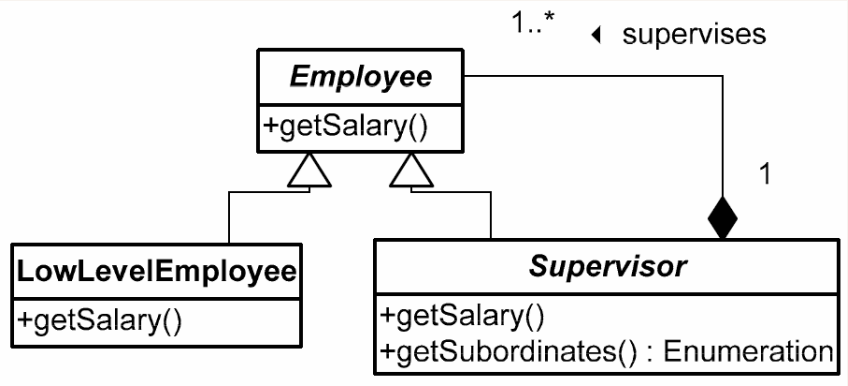
Object vs. Class Adapter

- ***Object* Adapter:**
  - Puts an *object* of the adaptee (i.e. LinkedList) inside each *object* of the adaptor (i.e. OAQueue)
  - Sometimes called a "wrapper"
- ***Class* Adapter:**
  - Adaptor is derived from the adaptee
  - Uses *class* extension capability of OO languages

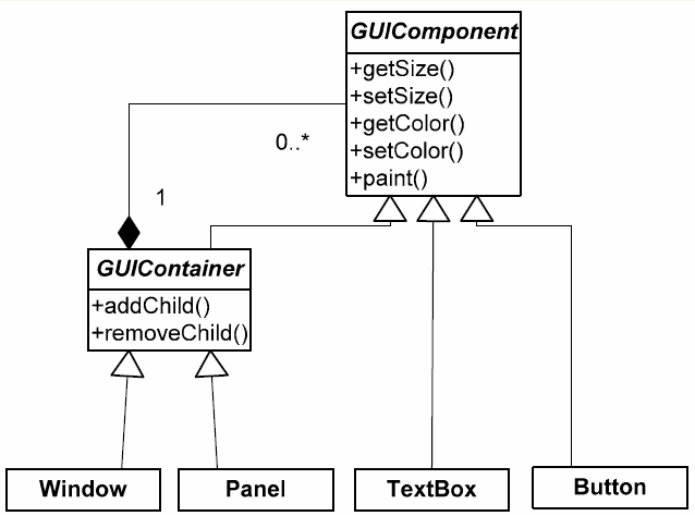
Composite Design Pattern



Composite Design Pattern

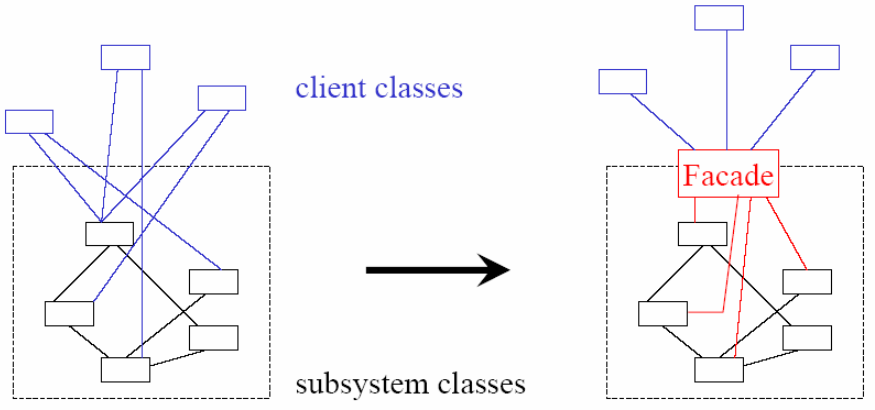


Composite Design Pattern



Façade Design Pattern

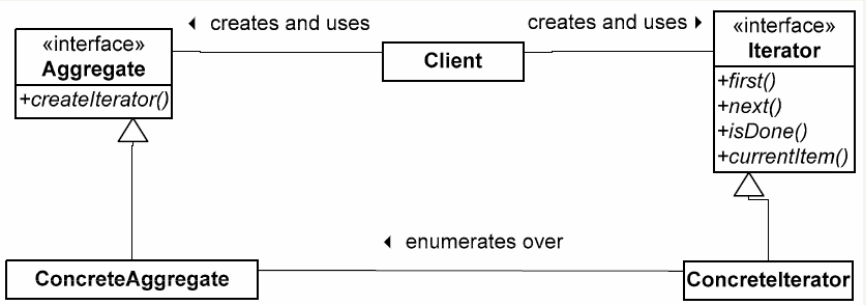
- Allows classes in a subsystem to be hidden



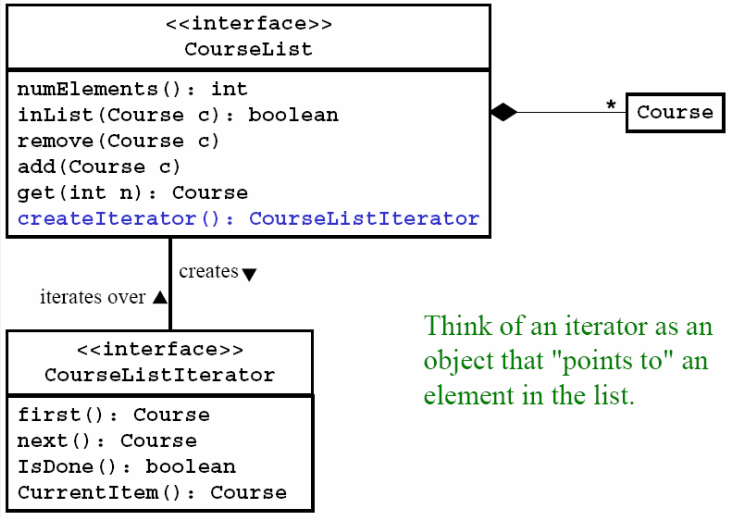
Façade Design Pattern

- **Examples:**
  - Compiler provides simple interface for scanner, parser, optimizer, etc.
  - JOptionPane provides a simple façade for creating simple dialogs in Java.

Iterator Design Pattern



Iterator Design Pattern



Questions

- **Question:** What is the difference between an adapter and a façade?

Questions

- **Answer:** both are structural patterns with the goal of providing a "better" interface to some existing functionality
- differences?
  - adapter is providing an alternate interface to *one class*; façade is providing an interface to a *collection of classes*
  - Both provide "better" interfaces in a slightly different sense
    - façade: better means simpler
    - adapter: better means more suited to a particular situation (i.e. using a linked list as a queue)

Past Midterm/Exam Questions

- **Question 3 Final 2003 (3 parts, 5 points each).** Determine which design pattern (if any) can be used to solve the following coding problem:

  [Description of problem here]
- If a suitable pattern exists,
  - write down the name of the pattern,
  - draw a class diagram that explains how the pattern is applied to the problem, and
  - write a few (2 or 3) sentences that explain how the pattern solves the problem.
- If no suitable pattern exists, explain why none of the patterns we studied in this course is appropriate. You do *not* need to write any code for this question.

Past Midterm/Exam Questions

- Your company sells computer parts. You are building an inheritance hierarchy of different kinds of parts. The hierarchy should contain the superclass Part and the three subclasses CPU, HardDisk, and CDROM. However, your supplier Franz already has classes for two of these parts: namely, FranzCPU and FranzHardDisk. How can you build a class hierarchy for your system and reuse Franz' classes?

Past Midterm/Exam Questions

- **Answer:** Use the Adapter design pattern so that FranzHarddisk and FranzCPU can be reused.

Past Midterm/Exam Questions

- You are implementing a simple, tree-like file system (similar to Windows or UNIX) for an operating system. You are making the following design decisions:
  - The entire file system is represented by a file system object which contains a reference to the root folder.
  - A folder contains two kinds of items: files and folders.
- How can you implement the file system?

Past Midterm/Exam Questions

- **Answer:** Use the Composite design pattern so that you can structure your classes such that a folder can consist of other folders and files.

Past Midterm/Exam Questions

- You have created a subsystem with 10 classes c1, ..., c10. To maximize reuse, each class is very general, that is, the methods in the class can be used in a variety of different ways. You know that most other subsystems will typically only use the classes c1, c2, and c3. Moreover, they will use the methods in these classes only in a certain, specialized way. How can you simplify the view that the other subsystems have of your subsystem?

Past Midterm/Exam Questions

- **Answer:** Use the Façade design pattern to provide an interface to the c1, c2, c3 classes for other subsystems.