CSCI 253

Object Oriented Design:
Structural Patterns
George Blankenship

Overview

Creational Patterns
- Singleton
- Abstract factory
- Factory Method
- Prototype
- Builder

Structural Patterns
- Composite
- Façade
- Proxy
- Flyweight
- Adapter
- Bridge
- Decorator

Behavioral Patterns
- Chain of Respons.
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

The Elements of a Design Pattern

- A pattern name
- The problem that the pattern solves
  - Including conditions for the pattern to be applicable
- The solution to the problem brought by the pattern
  - The elements (classes-objects) involved, their roles, responsibilities, relationships and collaborations
  - Not a particular concrete design or implementation
- The consequences of applying the pattern
  - Time and space trade off
  - Language and implementation issues
  - Effects on flexibility, extensibility, portability

Structural Patterns

- The **Adapter** pattern, used to change the interface of one class to that of another one.
- The **Bridge** pattern, intended to keep the interface to your client program constant while allowing you to change the actual kind of class you display or use. You can then change the interface and the underlying class separately.
- The **Composite** pattern, a collection of objects, any one of which may be either itself a Composite, or just a primitive object.
- The **Decorator** pattern, a class that surrounds a given class, adds new capabilities to it, and passes all the unchanged methods to the underlying class.
- The **Façade** pattern, which groups a complex object hierarchy and provides a new, simpler interface to access those data.
- The **Flyweight** pattern, which provides a way to limit the proliferation of small, similar class instances by moving some of the class data outside the class and passing it in during various execution methods.
- The **Proxy** pattern, which provides a simple place-holder class for a more complex class which is expensive to instantiate.

Structural Patterns George Blankenship 4

Problems

- Download Cooper's examples (http://www.patterndepot.com/put/8/JavaPatterns.ZIP)
- **Select one of the patterns**
  - **Adapter** pattern, **Bridge** pattern, **Composite** pattern, **Decorator** pattern, **Façade** pattern, **Flyweight** pattern, or **Proxy pattern**
  - **Explain the example supplied by Cooper**

Structural Patterns George Blankenship 5