Design Patterns: The Sequel!

Krysta Yousoufian
CSE 331

With material from Hal Perkins, David Notkin, Michael Ernst, Marty Stepp, and Joshua Bloch (Effective Java)

A Note

We didn't cover these examples in class (except singleton briefly in the 8:30 section) and you are not held responsible for their content. They are purely optional content for your intellectual curiosity or to help you understand the lecture concepts better.

File Server & Logger

File Server

- Allows clients to read files across a network
  - (For simplicity, ignore writes)
- Server…
  - …**accepts** a request from a client
  - …**loads** requested file
  - …**sends** contents back to client
  - …**logs** activity via logger
  - …**repeat**

File Server

- Allows clients to read files across a network
  - (For simplicity, ignore writes)
- Server…
  - …**accepts** a request from a client
  - …**loads** requested file
  - …**sends** contents back to client
  - …**logs** activity via logger
  - …**repeat**

File Server

- Requests handled via `ClientHandler`
- Several `ClientHandlers` run simultaneously in separate threads
- Problem: every `ClientHandler` has its own logger object for writing to the log
  - (Almost) simultaneous writes → overwrite each other's changes
- How do we solve this?

Singleton

- One shared instance of a class
- Use for:
  - Global state; coordinating among objects or threads
  - Often lower-level tasks (e.g. hardware interaction)
- Don't use for:
  - Managing state/data specific to each use (instance fields)
- Examples: logger, window manager

Implementing Singleton

- Always make constructor private
- Several options(*Effective Java* pp. 18+):
  1. Private static instance accessed with getInstance()
  2. Public static instance accessed directly
  3. Enum

Singleton with Logger

Word

• • •

(Text example, revisited)

Word

- Last time: Book, Chapter, Paragraph
- Want to break down further: Sentence, Word
- Creating an object for every word in the book takes up too much space
- What do we do?

Interning

- Cache existing objects
- Don't allow client to create objects directly
  - Private constructor
- When the client requests a value:
  - If an object with that value exists, return it
  - Else, create it, add it to the existing objects, and return it
- Only works for immutable data (why?)

Interning with Word

Word II

- Need Word to point to previous and next Words in sentence
- What problems does this cause?
- How can we fix it?

Flyweight

- Use when:
  - Objects are almost the same… but not quite
  - Most state is shared and can be interned
  - But some state are mutable or too specific to be shared among many objects
- **Caution:**
  - Flyweight makes your code messy and harder to use
  - Only use if memory usage is a demonstrated problem for your program

Implementing flyweight

- Remove extrinsic (non-shared) state from objects
- Client keeps track of the extrinsic state
  - ("Client" = another ADT, a main program, …)
- Accept extrinsic fields as method arguments where needed
- Then intern the objects as usual

Flyweight with Word