CSE 331
Software Design & Implementation

Winter 2021

Section 10 – Review and Design Patterns

Administrivia

- HW9 due tonight, 11 pm (+ late day if you have & need one)
  - Any last-minute questions?
- Final quiz out on Sunday
- Demos next week
  - Sign-up sheet to pick a time that works for you
  - 15-minute slot to demo your app and chat with a TA about it
- Do you have a great TA? (Of course you do!) Please consider nominating them for the Allen School Bandes TA award.
- Questions?

Design Patterns

- Creational patterns: get around Java constructor inflexibility
  - Sharing: singleton, interning
  - Telescoping constructor fix: builder
  - Returning a subtype: factories

- Structural patterns: translate between interfaces
  - Adapter: same functionality, different interface
  - Decorator: different functionality, same interface
  - Proxy: same functionality, same interface, restrict access
  - All of these types are wrappers

Design Patterns

- Interpreter pattern:
  - Collects code for similar objects, spreads apart code for operations (classes for objects with operations as methods in each class)
  - Easy to add objects, harder to add operations to all objects
  - Instance of Composite pattern
- Visitor ("Procedural") pattern:
  - Collects code for similar operations, spreads apart code for objects (classes for operations, method for each operand type)
  - Easier to add operations, harder to add objects

Design Patterns

Adapter, Builder, Composite, Decorator, Factory, Iterator, Intern, Interpreter, Model-View-Controller (MVC), Observer, Procedural, Prototype, Proxy, Singleton, Visitor, Wrapper

- What pattern would you use to…
  - Add a scroll bar to an existing window object on the screen.
  - We have an existing object that controls a communications channel. We would like to provide the same interface to clients but transmit and receive encrypted data over the existing channel.
  - When the user clicks the "find path" button in the Campus Maps application (hw9), the path appears on the screen.

Design Patterns

- What pattern would you use to…
  - Add a scroll bar to an existing window object on the screen
    - **Decorator**
  - We have an existing object that controls a communications channel. We would like to provide the same interface to clients but transmit and receive encrypted data over the existing channel.
    - **Proxy**
  - When the user clicks the "find path" button in the Campus Maps application (hw9), the path appears on the screen.
    - **Observer** (used to implement MVC design)

Strategy Design Pattern

- **Problem:** Some parts of our algorithm need to be different depending on which client is making the request

- **Solution:** Have clients pass in an object representing the specific strategy for the missing parts
  - it will invoke that object's methods to perform the strategy

- Example: `HashMap` needs help computing the hash code

- What is another name we already have for this?
  - It is a callback

Command Design Pattern

- **Problem:** Part of the code needs to identify user requests but does not know how to process them. It needs a way to communicate this to the rest of the code.

- **Solution:** Put the request details into an object that can be passed to the code that does the processing.

- Example: the `Request` object in Spark Java

- Another example from the libraries we used?
  - Button clicks

Your turn!

- Use your knowledge of design patterns to solve the problems on the worksheet.