

CISC 3120

C21: Some Design Patterns and Architecture Styles

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- More of a review and recap
 - Concept of design patterns
 - A few creational patterns
 - Static factory, Factory, Builder
 - A behavioral pattern
 - Observer
 - A few architecture styles
 - MVC, client/server, layered.
- Further readings
 - See the class website

Design Patterns

- Design solutions to recurring problems
 - Problem: what recurring problems to solve?
 - Solution: what is the solution?
 - Context: under what conditions that the solution is intended to solve the recurring problem?

Static Factory

- A static method returns instances of the class
 - Static method belongs to class
 - There is no "this" instance of the class
 - Examples in Java API
 - `javafx.scene.paint.Color`
 - `static Color rgb(int red, int green, int blue)`

Problem, Solution, and Context

- There is a need to construct the instance of the class in many different ways
 - Wait! Would constructors solve the problem?
 - How well does overloading solve the problem?
- Static factory method
 - Example: `java.text.Collator`
 - `static Collator getInstance(Locale desiredLocale)`
 - Others you saw?
- When do we create static factory method for our own classes?

Factory

- There are many subclasses that share the same set of methods, and a generic class can be written to take advantage of this.
 - Wait! Would constructors solve the problem?
 - You have to rely on concrete subclasses, how well do their constructors help you solve the problem?
- Factory method
 - Examples:
 - `java.util.ResourceBundle's getBundle()`
 - `java.text.NumberFormat's getInstance()`
 - `java.nio.charset.Charset's forName()`
 - When do we create factory method for our own classes?

Builder

- Objects of a classes can be constructed in extremely many different ways
 - Wait! Can we use constructors, static factory method, factory of abstract class?
- Solution
 - Builder builds an object step by step, with initialization parameter
 - Example:
 - (In CmdLineArgsDemo app) `org.apache.commons.cli.Options`
 - `Options addOption(...)`
 - (In LinkNetInterfaceExplorer) `java.lang.StringBuilder`
 - `StringBuilder append(...)`
- When do we create builder class for own our projects?

Observer

- One depends on many other classes
 - In MVC, when model changes, view also needs to change.
- Solution
 - An observable object can have one or more observers, and observers can be notified the changes the observable object
 - Commonly used in the User Interface design
- When do we create factory interface or abstract factory method for our own classes?

Observer Pattern: Examples

- `java.util.Observable` and `java.util.Observer`
- JavaFX properties and bindings

Dependency Injections

- One class or object may depends on another
- Common patters
 - Via constructor
 - Via setter
 - Via setter interface

MVC

- Model
- View
- Controller

Layered

- Examples:
 - How are networking functionalities structured in a computer system?
 - OSI model & TCP/IP model

Client and Server

- Different granularity
 - One object provides a service to another (calling a method)
 - One thread provides a service to another
 - One process/program provides a service to another
- Message passing

Questions?

- Discussed a few design patterns and architecture styles
 - Static factory
 - Abstract factory
 - Builder
 - Observer
 - Dependency injections
 - MVC
 - Layered
 - Client/server
- How do we architecture an application?
- Further readings: class website, and what would you recommend?