Design Defects & Restructuring

Week 4: 24 Sep 22

Rahim Hasnani

Agenda

- Any questions from last week!
- Midterm
- First Chapter of Head First Design Patterns
- GOF Design Patterns – Introduction
- Architecture Patterns
- MVC
- SoC, Repository and Dependency Injection

New Homework

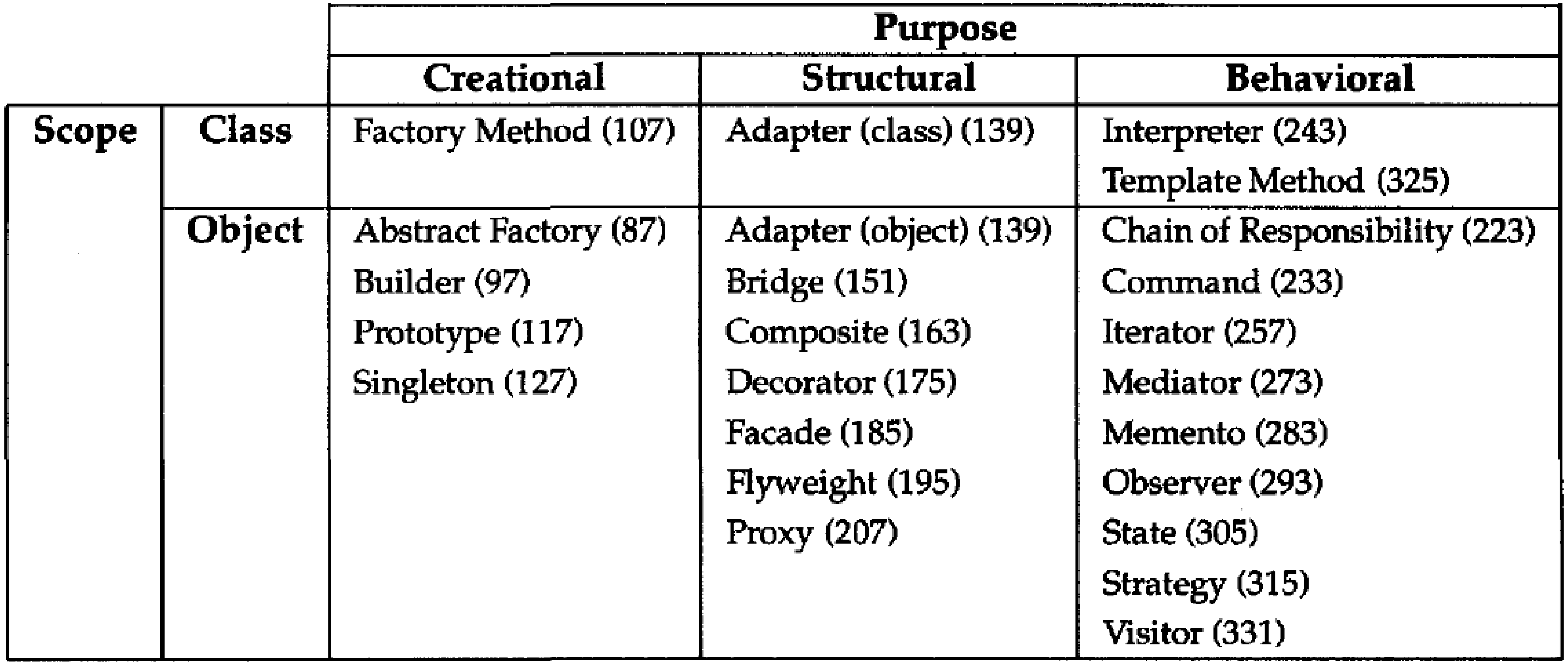
- Subclass vs Subtype
- Their relationship with public vs private inheritance
- How subtyping is implemented/enforced/achieved in C++, Java, C#? What about Python?

GOF Patterns

- Walk-through of First chapter of head first design patterns book

GOF Design Patterns - Introduction

- Basic Idea
  - Flexibility
  - Re-use
  - Extensibility
- Pattern Categorization by Purpose
  - Creational: object creation
  - Behavioral: Ways in which classes or objects interact and distribute responsibility
  - Structural: Composition of classes or objects
- Pattern Categorization by Scope
  - Class
  - Object



		Purpose		
		Creational	Structural	Behavioral
Scope	Class	Factory Method (107)	Adapter (class) (139)	Interpreter (243) Template Method (325)
	Object	Abstract Factory (87) Builder (97) Prototype (117) Singleton (127)	Adapter (object) (139) Bridge (151) Composite (163) Decorator (175) Facade (185) Flyweight (195) Proxy (207)	Chain of Responsibility (223) Command (233) Iterator (257) Mediator (273) Memento (283) Observer (293) State (305) Strategy (315) Visitor (331)

Pattern Structure

- Name
  - Increases your design vocabulary
  - Design at a higher level of abstraction
- Problem
  - When to apply the pattern
  - (sometimes,) list of conditions that must be met
- Solution
  - Elements that make up the design
  - Template, not concrete
- Consequences
  - Results & trade-offs of applying the pattern

Rules of Object Oriented Design

- Program to an interface, not an implementation
  - Don’t declare variables to be instances of particular concrete class
  - Abstract the process of object creation
- Favor object composition over class inheritance
  - Reusing by inheritance is white-box re-use
  - Reusing by composition is black-box re-use
- Others
  - Inheritance vs parameterized types (templates/generics)

Architectural Patterns

- Database driven
- N-tier
- Client Server
- Master Slave
- MVC
- Micro-Services??

MVC Pattern

- Introduction
- SoC
- What goes into controller, model and views