

The background of the slide is a blue-tinted image of architectural blueprints. A yellow ruler is visible in the top left corner, and a yellow pencil is in the bottom right corner. The blueprints contain various technical drawings, lines, and text, including the word 'FIRST' in large letters. A blue rectangular box is centered on the page, containing the title text in white.

Software Design & Architecture

Design Patterns/ Behavioral Design Patterns

Pengyu Nie

Design Patterns Categories

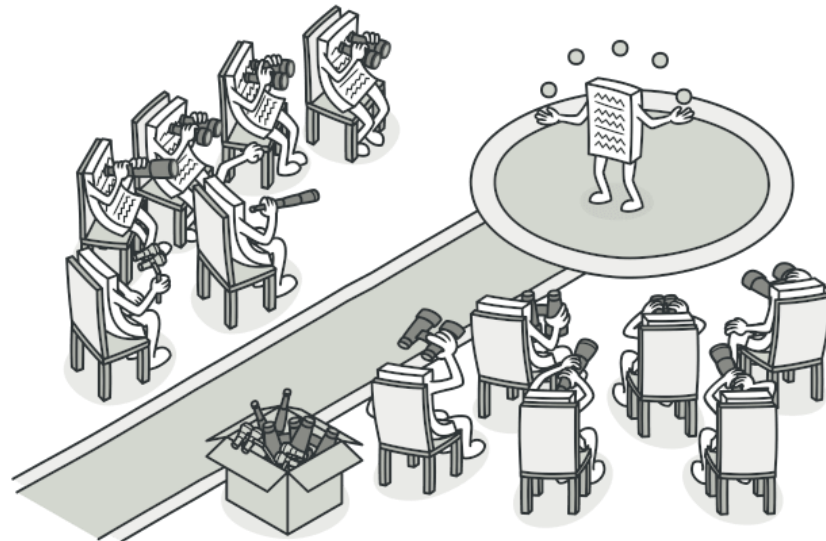
- **Creational**: concern the process of object creation
 - Singleton, Factory Method, Abstract Factory, Builder, Prototype, Object Pool *design patterns 1*
- **Structural**: concern the process of assembling objects and classes
 - Adapter, Composite, Decorator, Façade, Bridge, Flyweight, Proxy *design patterns 2*
- **Behavioral**: concern the interaction between classes or objects
 - Observer, Strategy, Template Method, Iterator, State, Chain of Responsibility, Command, Mediator, Memento *today*
design patterns 4 – your pick from the remaining ones

The image features a background of architectural blueprints on a light blue surface. A large, semi-transparent blue rectangle is overlaid on the left side of the image. The word "Observer" is written in white, sans-serif font within this rectangle. The blueprints in the background show various technical drawings, including lines, dimensions, and text such as "FIRE SHUTTERS", "SPEC.", "FLOOR DRAIN", and "UPPER". A yellow ruler is visible in the top left corner, and a yellow pencil is in the bottom right corner. The overall scene suggests a professional or technical environment.

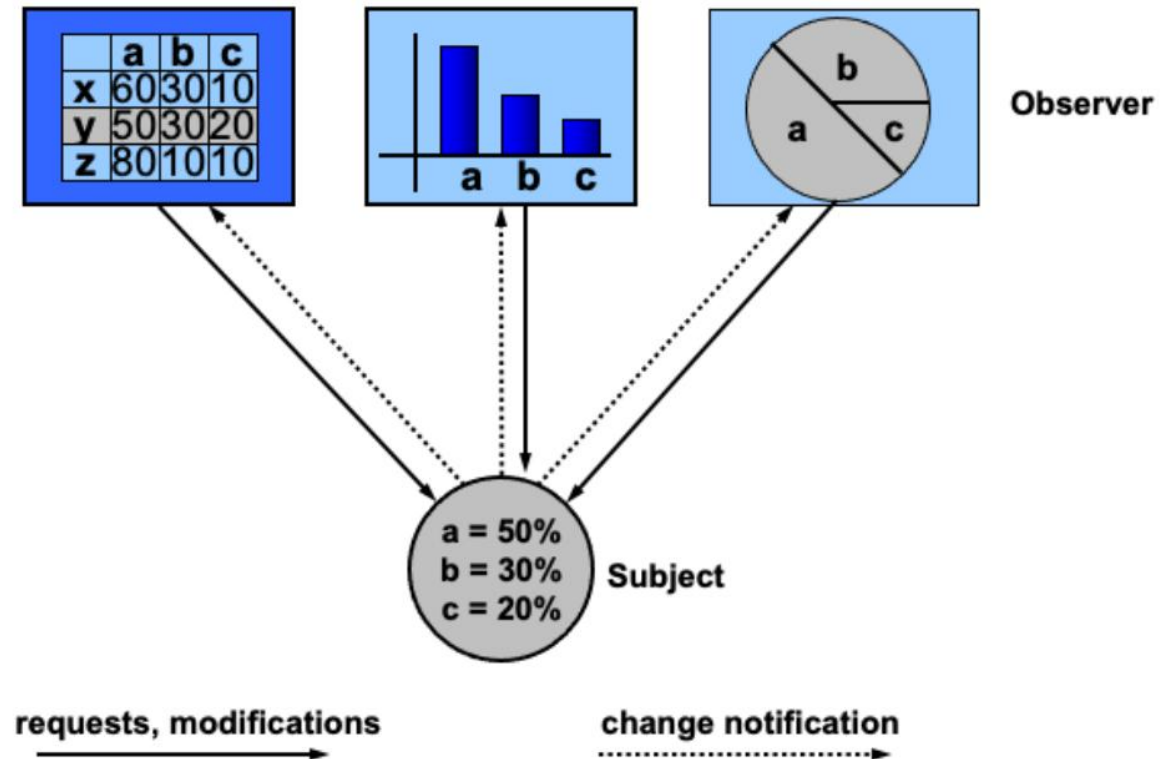
Observer

Observer: Motivation and Intent

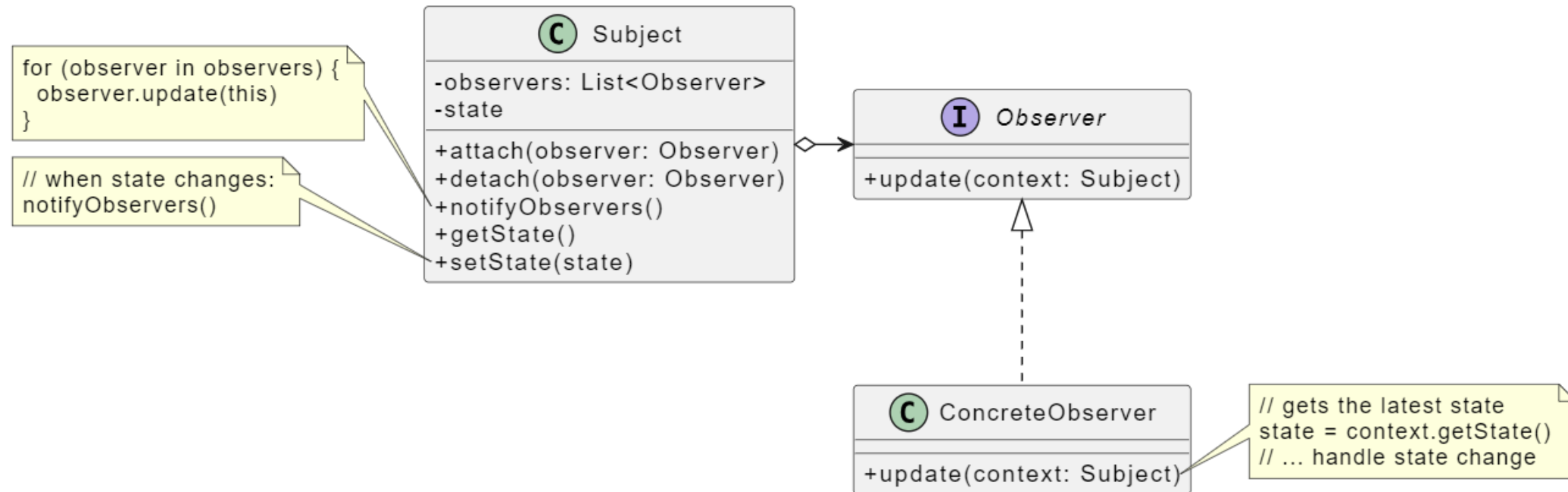
- **Motivation:**
 - a common side-effect of partitioning a system into a collection of cooperating classes is the need to maintain consistency between related objects
- **Intent:**
 - define a one-to-many dependency between objects
 - when one object changes state, all its dependents are notified and updated automatically



Observer: Example



Observer: Solution



The image shows a close-up of architectural blueprints on a light blue background. A large, semi-transparent blue rectangle is overlaid on the center of the page. The word "Strategy" is written in white, sans-serif font within this rectangle. The blueprints in the background contain various technical drawings, including floor plans, dimensions, and labels such as "FIRST", "UPPER", "FLOOR DRAIN", and "SLOPE". A yellow ruler is visible in the top left corner, and a yellow pencil is in the bottom right corner.

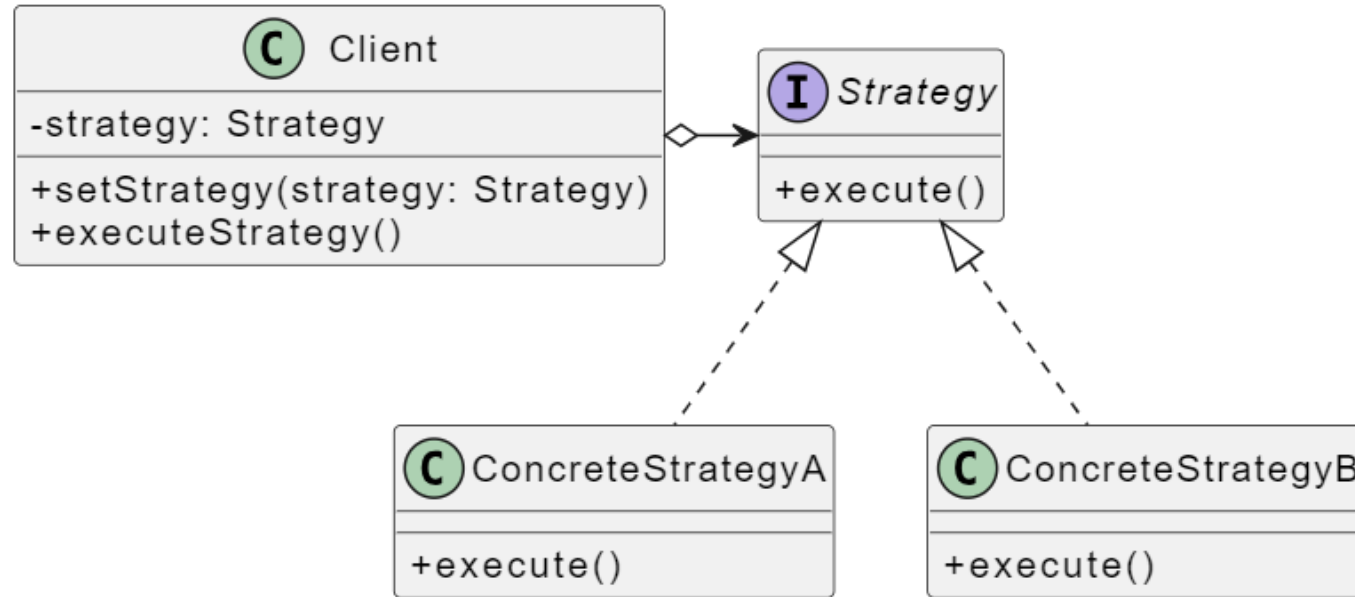
Strategy

Strategy: Motivation and Intent

- **Motivation:**
 - a problem with multiple well-defined solutions that conform to a common interface
 - client can vary the implementation according to specific needs
- **Intent:** define a family of related algorithms behind a common interface



Strategy: Solution

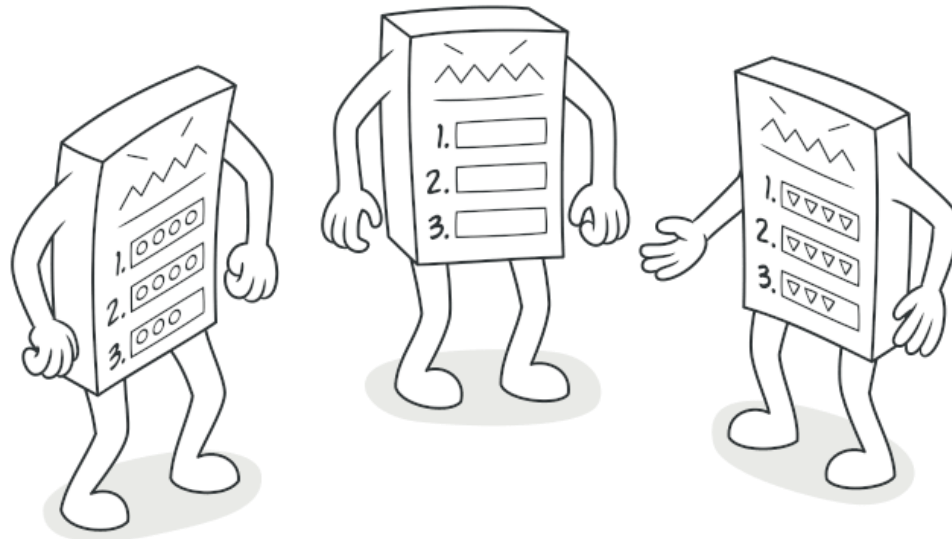


The image shows a close-up of architectural blueprints on a light blue background. A large, semi-transparent blue rectangle is overlaid on the center of the page, containing the text 'Template Method' in white. The blueprints in the background feature various technical drawings, including lines, dimensions, and text labels such as 'FIRST', 'UPPER', 'FLOOR DRAIN', and 'SLOPE'. A yellow pencil is visible in the upper left corner, and a white eraser is in the lower left corner. The overall scene is a professional architectural workspace.

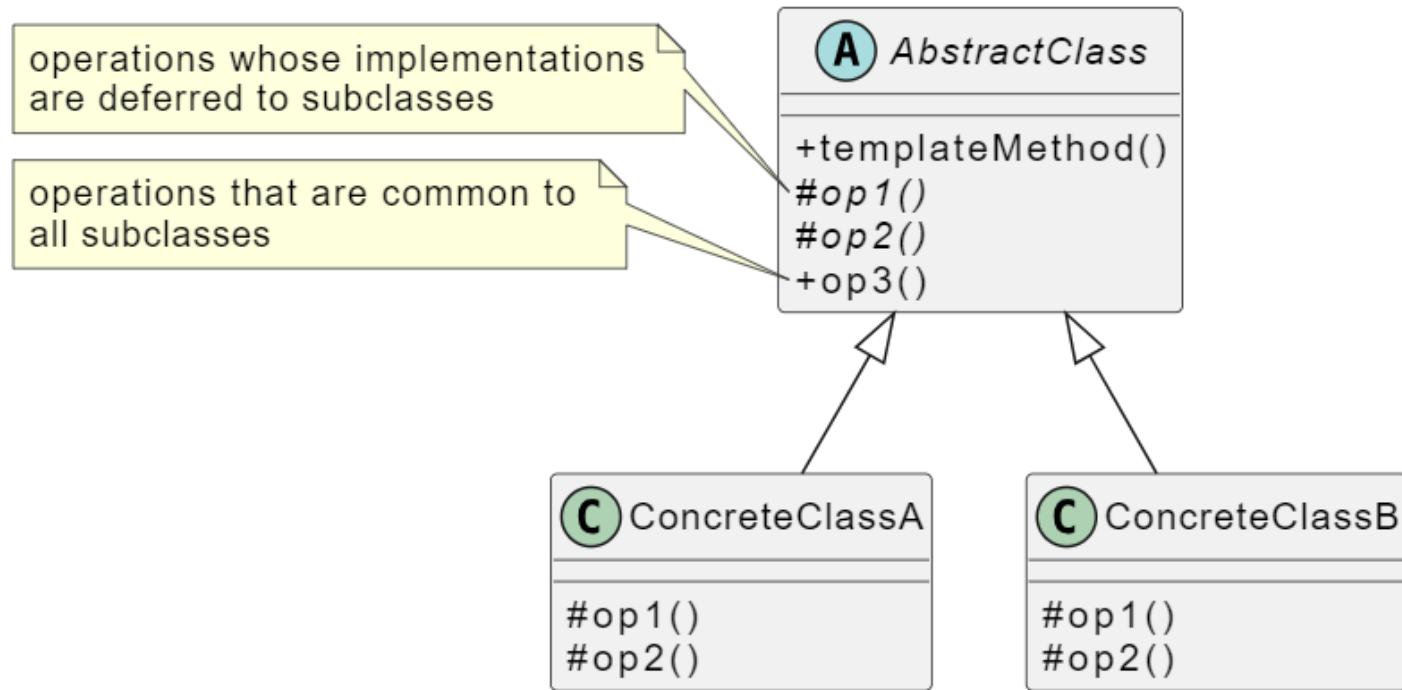
Template Method

Template Method: Motivation and Intent

- **Motivation:**
 - the ordering of steps in an algorithm is fixed
 - the implementations of each step can vary to suite specific needs (e.g., reading and processing files in different formats)
- **Intent:** define the skeleton of the algorithm in the superclass, defer some steps to subclasses



Template Method: Solution



Agenda (recap)

- Behavioral design patterns
 - Observer
 - Strategy
 - Template method

- P4: Iteration 2 Demo this Wednesday!