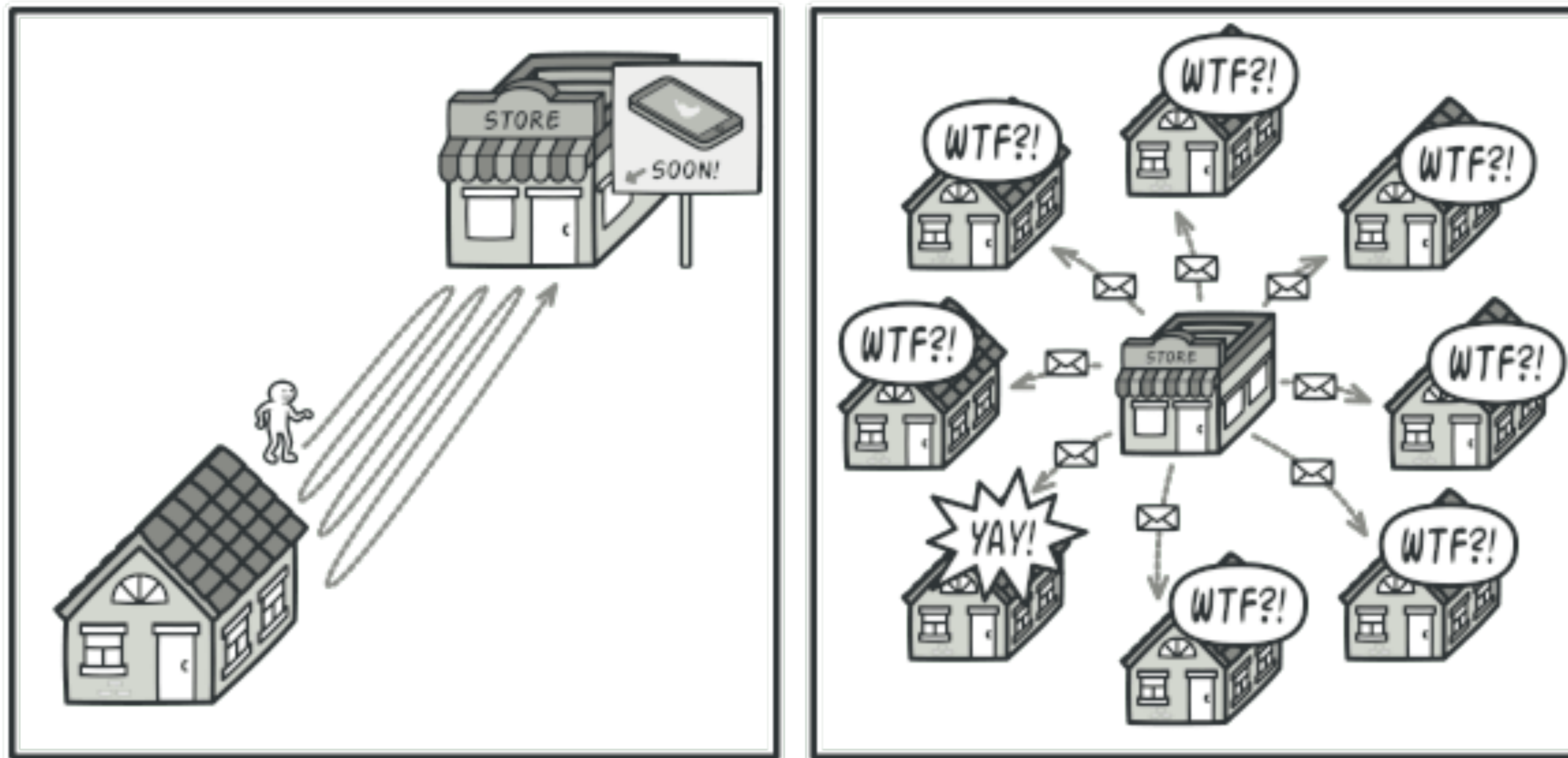


Observer Pattern

Вибрані питання програмної інженерії

Яременко Петро, ФІ БП КН-4, 2024

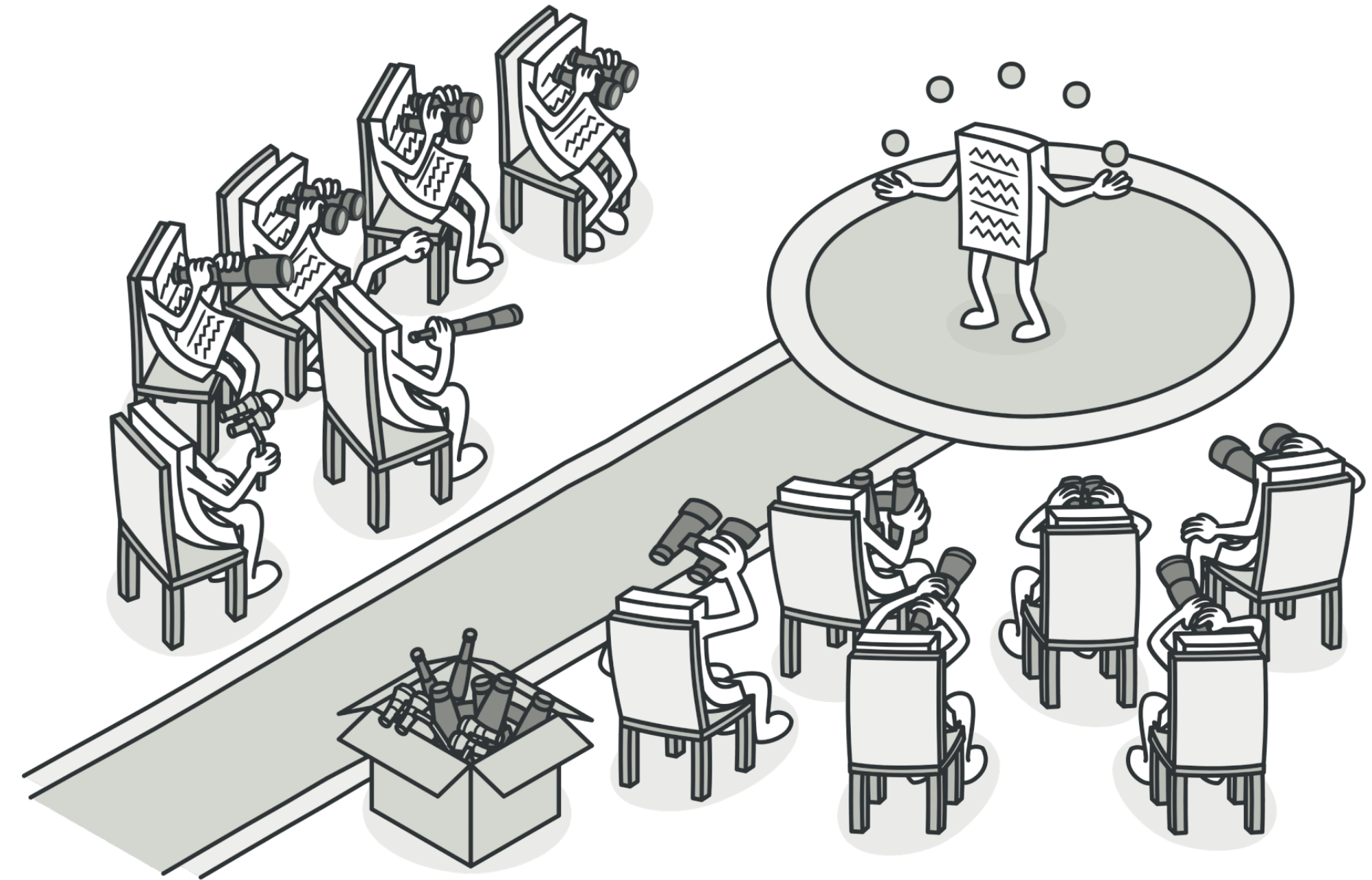
Проблема



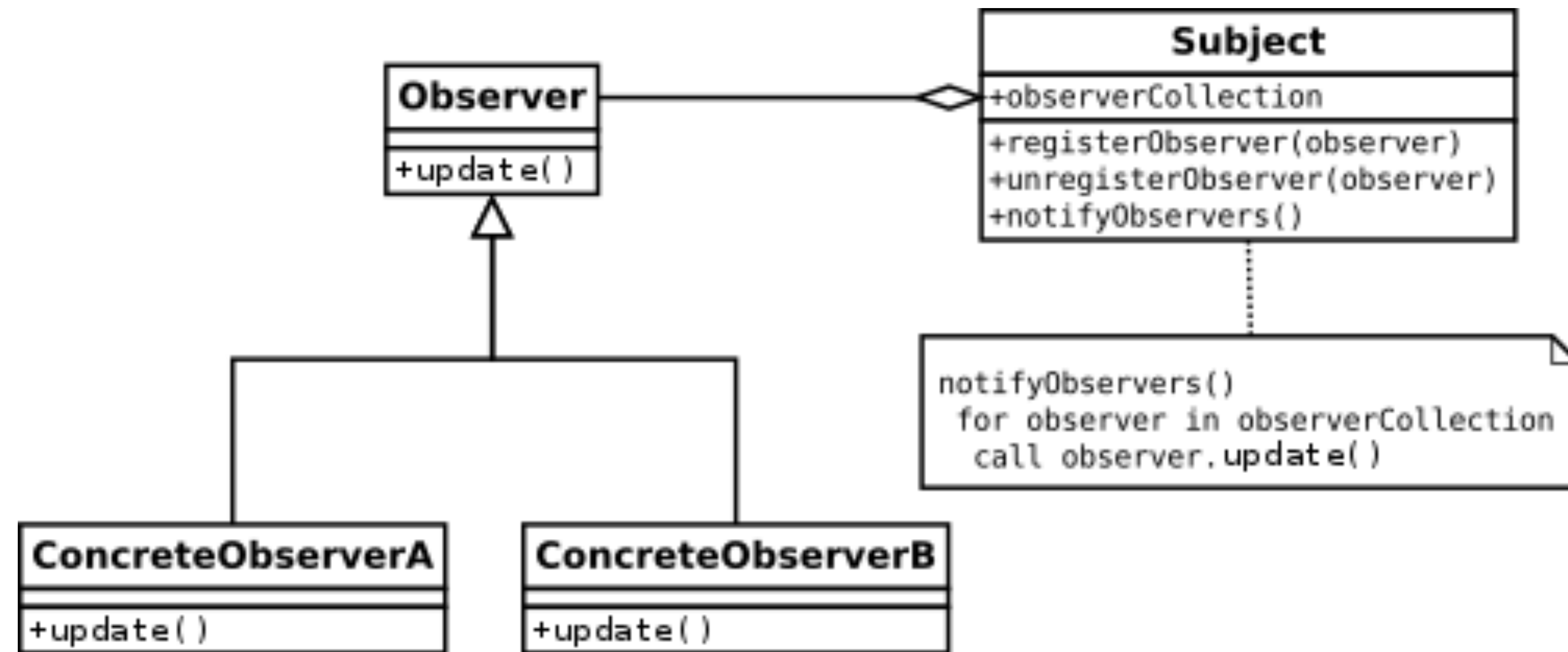
Або клієнт витрачає час на перевірку наявності продукту, або магазин витрачає ресурси на сповіщення неправильних клієнтів.

Рішення

Паттерн програмування **"Observer"** — це паттерн проектування, який використовується для створення механізму subscription, щоб сповіщати об'єкти про будь-які зміни стану у **observable** об'єкті. Цей паттерн входить до категорії поведінкових паттернів, оскільки він визначає спосіб комунікації та взаємодії об'єктів.



Основні ролі



- **Subject (Суб'єкт)**: Це об'єкт, який має стан і може сповіщати своїх спостерігачів про будь-які зміни свого стану.
- **Observer (Спостерігач)**: Це об'єкт, який очікує на зміни у суб'єкта і реагує на них.

Основні юзкейси

- Поширення інформації про зміни стану
- Реакція на зміни у реальному часі
- Розділення відповідальності
- Забезпечення розширюваності
- Підтримка багатоварової архітектури
- Повідомлення про події

Переваги

- Розділення обов'язків
- Гнучкість
- Реактивне програмування
- Модульність
- Підтримка багатошарової архітектуриб. Менша залежність

Недоліки та нюанси

- Можливість витоку пам'яті
- Навантаження на публікацію подій
- Взаємозв'язок між видавцем та підписниками
- Необхідність оновлення всіх підписників
- Синхронізація при багатопотоковості
- Підвищена складність при великій кількості подій та підписників

Порівняння з іншими паттернами

Порівняння з паттерном "Callback"

У паттерні "Callback" функція передається як аргумент іншій функції для виклику у відповідь на певну подію. Паттерн "Observer" використовує об'єкти, які взаємодіють один з одним, замість функцій.

Використання об'єктів може бути більш гнучкою або структурованою, особливо для більш складних взаємодій.

Порівняння з паттерном "Mediator"

У паттерні "Mediator" об'єкт, відомий як посередник, виступає посередником між різними компонентами програми, сприяючи їх взаємодії. У порівнянні з "Observer", де об'єкти підписуються один на одного, у "Mediator" всі взаємодії відбуваються через посередника, що робить систему менш зв'язаною.

Порівняння з паттерном "State" (стан)

У паттерні "State" об'єкт змінює свій стан, і його поведінка змінюється відповідно до цього стану. Паттерн "Observer" може використовуватися для сповіщення про зміну стану об'єкта і оновлення відповідних підписників. Обидва паттерни можуть співпрацювати для організації складних систем з внутрішніми станами та поведінкою.

Висновки

Паттерн "Observer" є потужним і корисним інструментом для організації взаємодії між об'єктами в програмах. Він дозволяє створювати реактивні системи, де підписники автоматично реагують на зміни стану видавця, без необхідності явних запитів або опитувань.

У певних випадках може бути важко визначити, коли і де краще використовувати паттерн "Observer", але він є важливим інструментом при створенні реактивних, модульних та гнучких систем. При правильному використанні паттерн "Observer" допомагає створювати добре організовані та підтримувані програмні системи.

Джерела

- Design Patterns: Elements of Reusable Object-Oriented Software" (The "Gang of Four" book) - Еріх Гамма, Річард Хелм, Ральф Джонсон, Джон Вліссідес.
- "Head First Design Patterns" - Ерік Фрімен, Елізабет Фрімен, Кеті С'єра, Берт Бейтс.
- "Patterns of Enterprise Application Architecture" - Мартін Фаулер.