

Noter til "Design Patterns - GoF Observer"

1. Introduktion til GoF Observer

- Forelæser: Claudio Gomes
- Version: 1.0.5

2. Generelt problem med dataopdatering

- **Provider:** Indeholder data, der ændres lejlighedsvis af en ekstern enhed.
- **Consumer:** Bruger data og ønsker at opdatere baseret på ændringer i data.

3. Løsning: GoF Observer

- **Intent:** Definer en en-til-mange afhængighed mellem objekter, så når et objekt ændrer tilstand, bliver alle dets afhængige objekter notificeret og opdateret automatisk.
- **Struktur:**
 - **Subject:** Abstrakt basisklasse for alle datasubjekter.
 - **Observer:** Interface, der skal implementeres af alle klasser, der ønsker at blive informeret om dataændringer.
 - **ConcreteSubject:** Arver fra Subject og er den faktiske klasse, der skal overvåges.
 - **ConcreteObserver:** Implementerer Observer interface og er den faktiske klasse, der skal modtage opdateringer.

7. Håndtering af subjects af forskellige typer

- Brug af generiske typer for at gøre det muligt for en Observer at oprette forbindelse til forskellige typer Subjects.

8. Observer Pattern i C#

- Selvom Observer-mønsteret ikke er en del af standardbiblioteket i C#, kan det let implementeres med generiske interfaces og klasser.

9. Spørgsmål?

- Afsluttende diskussion og spørgsmål.