

# Observer pattern

<https://github.com/Balen-Nore/Clock4You.git>

## Varför Observer Pattern?

Observer Pattern passar perfekt för Clock4You eftersom klockan uppdateras varje sekund och jag vill att flera komponenter, exempelvis användargränssnittet, automatiskt ska få uppdateringar utan att vara direkt kopplade till tidslogiken.

## Fördelar:

- **Lös koppling:** UI och logik separeras, vilket gör koden mer underhållbar.
- **Automatiska uppdateringar:** UI-komponenten får uppdaterad tid direkt utan att ständigt fråga klockan om den har ändrats.
- **Flexibilitet:** Nya observers kan enkelt läggas till utan att modifiera existerande kod.

## Use Case: Uppdatering av tid i gränssnittet

**Pre-villkor:** Applikationen körs och UI är öppet.

## Huvudflöde:

1. Clock4You's klocka tickar och uppdaterar sin interna tid.
2. Observer Pattern notifierar UI-komponenten om den nya tiden.
3. UI-komponenten uppdaterar sin visning.
4. Användaren ser den aktuella tiden utan att behöva göra något.

**Post-villkor:** UI visar alltid den senaste tiden i realtid.

## User Story

"Som användare vill jag att Clock4You uppdateras automatiskt så att jag alltid ser den aktuella tiden utan att behöva ladda om sidan."

## Acceptanskriterier:

- Tiden ska uppdateras automatiskt varje sekund.
- UI ska reflektera aktuell tid utan att användaren behöver interagera.
- Systemet ska stödja både 12-timmars och 24-timmars format.

### Förklaring av koden:

- Clock4YouClock.cs: Hanterar tiden och notifierar observers.
- IClockObserver.cs: Ett interface för observers.
- Clock4YouDisplay.cs: Implementerar IClockObserver och uppdaterar UI.
- Program.cs: Skapar instanser och kopplar samman objekten.

### Sammanfattning

Genom att använda **Observer Pattern** kan jag hålla UI-komponenten uppdaterad i realtid utan att binda den direkt till tidslogiken. Detta ger en flexibel och modulär lösning där jag enkelt kan lägga till fler observers om jag i framtiden vill ha fler komponenter som behöver uppdaterad tid (exempelvis en loggfunktion eller en historikmodul).