Design Patten

Observer patten

Observer Patten Generalt

- Kaldes også for subscriber patten
- En måde hvor vi kan sende notifikation eller info på et specifikt tidspunkt.
- Vi kan præciserer hvornår det skal sendes.
- Filtrering af data.
  - Bruger baserede data.

Anvendelse

- **Brugergrænseflader:** I grafiske brugergrænseflader, hvor ændringer i data (f.eks. i en database) skal reflekteres i UI-elementer.
- **Hændelsesbaserede systemer:** I systemer, der kræver realtids opdateringer, som f.eks. chatapplikationer eller finansielle applikationer, hvor der er behov for at modtage opdateringer om priser eller nyheder.
- **Spiludvikling:** Hvor forskellige objekter i et spil (f.eks. spillere, fjender) skal reagere på hændelser i spillet.

- I skal udvikle et Java baserede Student Management System, der benytter forskellige designmønstre til at organisere og håndtere systemets funktionalitet. Programmet skal bruge de vedlagte fiktive studerende, som I gemmer i enten en XML-, JSON- eller CSV-fil og indlæser i jeres program.
- **Krav til funktionalitet:**

1. **Visning af studerende:**
   En funktion der viser en liste over alle studerende med deres fornavn, efternavn og klasse.
2. **Søgning på fag/hold:**
   En måde, hvor man søger på et specifikt fag (fx Matematik A), og de studerende, der er tilmeldt dette fag, vises.
3. **Søgning på studerende:**
   Mulighed hvor man søger på en studerendes navn, og programmet viser hvilke fag og klasse den pågældende elev er tilmeldt.

   Som ekstra implementer en måde hvor det er kun aktive hold er vises.
4. **Skift af fagstatus:**
   Implementér en måde, der kan ændre status på et fag (fx fra 'Aktiv' til 'Afsluttet' eller omvendt).
5. **Tilføjelse af nye data:**
   Mulighed for at tilføje nye elever, fag og klasser i systemet.
6. **Avanceret søgning: (Extra)**

   En søgefunktion, der viser de afsluttede fag for en given elev.

StatDesign Mønster

1. **State Patten**
   Brug State Pattern til at håndtere forskellige status af en studerendes fagstatus (fx aktiv eller afsluttet). Hver status skal repræsenteres som en separat tilstand, og programmet skal kunne ændre mellem disse tilstande.
2. **Command Pattern: (Extra)**
   Anvend Pattern til at implementere kommandoer som fx "Vis studerende", "Søg fag", "Søg studerende", og "Skift fagstatus".
3. **Factory Pattern (Extra):**
   Brug Factory Pattern til at skabe nye instanser af studerende, fag og klasser, når nye data skal tilføjes systemet.