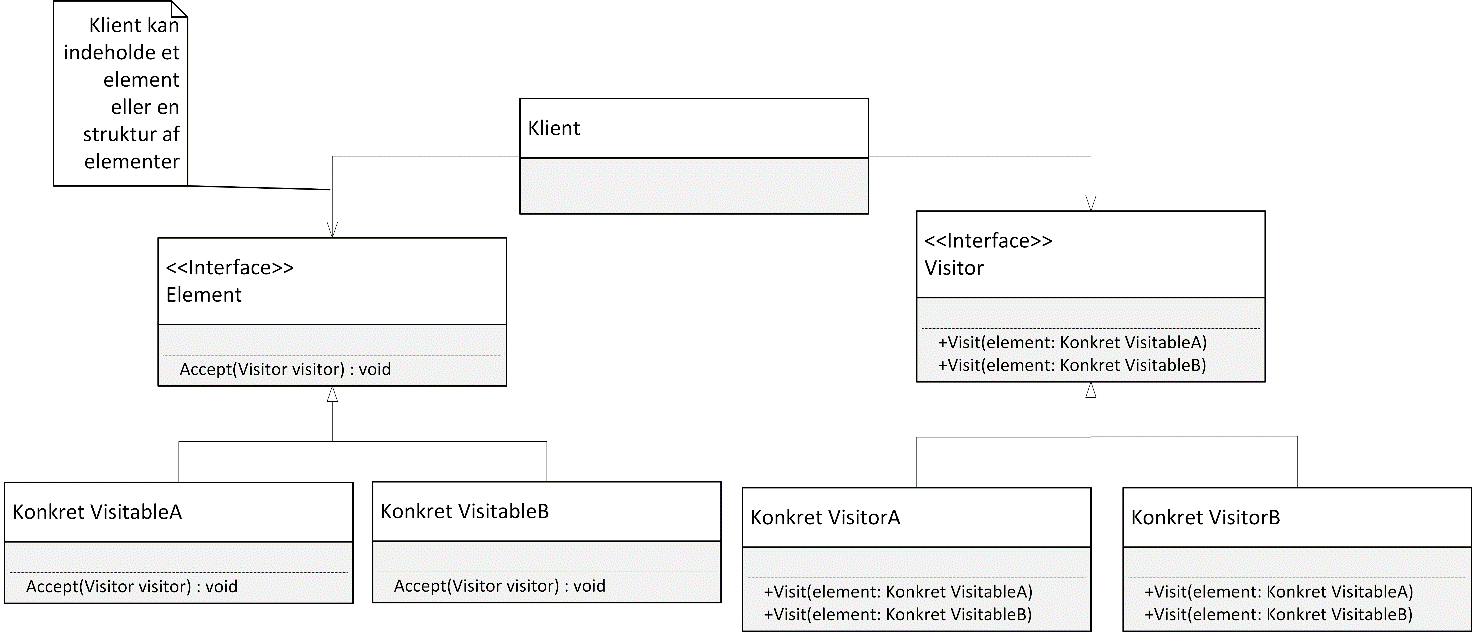
13. Obligatorisk opgave

# Redegør for, hvad et software design pattern er.

Et software design pattern er en general genbrugelig løsning til problemer der tit opstår i en given kontekst i software design. Det er ikke et færdigt design der kan laves direkte til kilde kode. Det er en beskrivelse eller skabelon for hvordan et problem kan løses i mange forskellige situationer. Det er formaliserede bedste praksisser som en programmør kan bruge til at løse problemer med.

# Hvad er en Visitor?

Målet med visitor er at kunne besøg elementer der ligner hinanden og udføre operationer på disse.



### Klient

Er bindeledet mellem Visitors og dem der er visitable. Vil enten have en datastruktur med elementer der kan traverses eller selv stå for at traversere over en række elementer.

### Konkrete Visitors (Visitors)

Dette kan for eksempel være en årsrapport, der udskriver hvor meget løn, hver medarbejder har modtaget.

Indeholder Visit funktioner der kan udføres på elementerne i data strukturen. For at programmere en Visitor bruger vi et interface der indeholder alle de forskellige overloadede visit funktioner.

### Konkrete Visitable (elementer)

Dette kan være en medarbejder, leder eller studentermedhjælper f.eks.

Der alle implementerer Accept funktionen, der modtager en Visitor. Accept funktionens formål er så at kalde Visitorens Visit funktion med sig selv som parameter. Derved bliver den korrekte Visit funktion kaldt, baseret på elementets type.

En Visitable bliver programmeret ved at den implementere et interface med Accept metoden

### Datastrukturen (Ikke nødvendig for pattern)

Vil ofte være bygget via Composite Pattern, da denne kan indeholde forskellige typer elementer (kompleks data), og kan traverseres uden at kende til disse. Dog kan det også blot være en kompleks struktur.

## Godt pattern?

Nej der er meget der ikke er overholdt i forhold til solid.