

3. GoF Strategy + GoF Template Method

1 Redegør for hvad et software design pattern er

Et design pattern er kernen i en løsning til et problem, som opstår gentagende gange. Årsagen til, at det er et mønster, er, at problemet og løsningen af problemet kan anvendes igen og igen uden, at man gør det samme 2 gange.

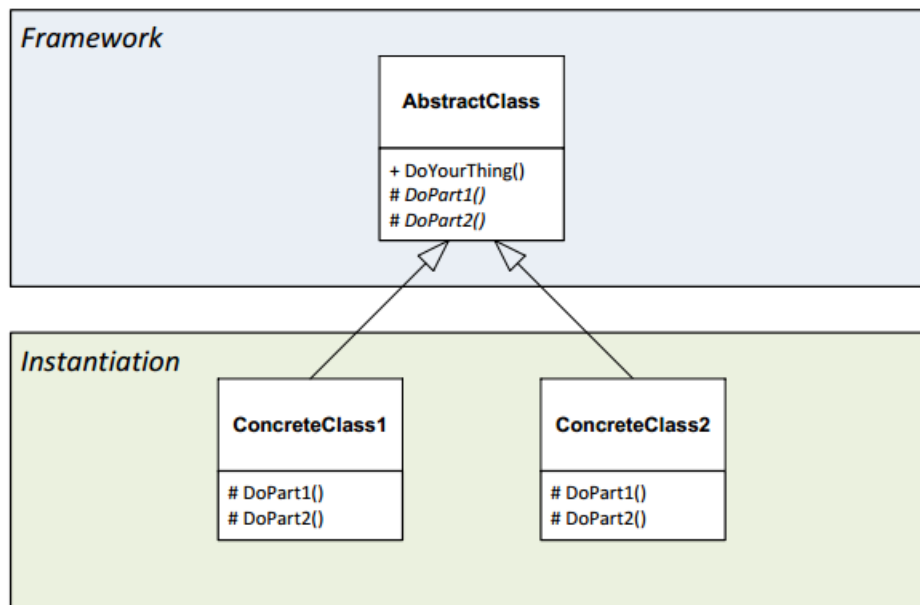
Da design patterns altid skal tilpasses vores problem i vores kontekst. Et design pattern er derfor aldrig en færdig løsning på et problem, men en skabelon til løsning af et problem.

2 Sammenlign de to design patterns GoF Strategy og GoF Template Method

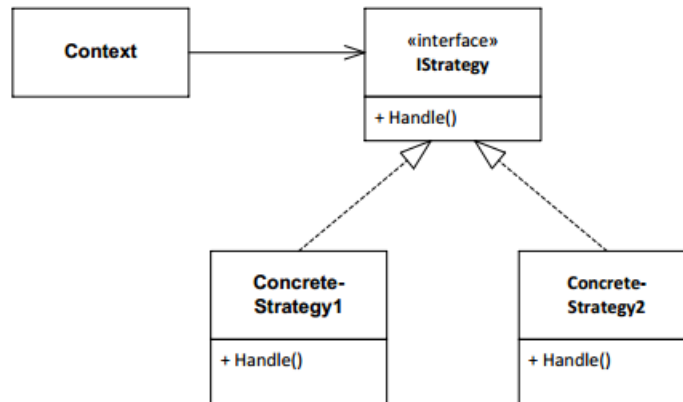
2.1 Template

GoF template har til formål at strukturere generel adfærd i software. Bruges de samme strategier eller laver man fx mange turbaseret spil, kan man lave en template klasse som håndterer den mest generelle adfærd.

Et turbaseret spil indholder bestemte elementer som er generelle for den kategori og disse kan beskrives med en template. (initGame(), GameOver(), TakeTurn(Player p), Disse metoder laves så abstrakte så de nedarvede klasser tvinges til at implementere deres udgave.



2.2 Strategy



Formålet med Strategy er at definere en samling af algoritmer, indkapsle dem hver for sig og gøre det muligt at skifte imellem dem.

Strategy pattern gør det muligt at ændre kontekstens behaviour i runtime ved at uddelegere behaviour til et andet objekt. Vi faktorerer altså kontekstens behaviour ud i et interface.

2.3 hvornår vil du anvende hvilket, og hvorfor?

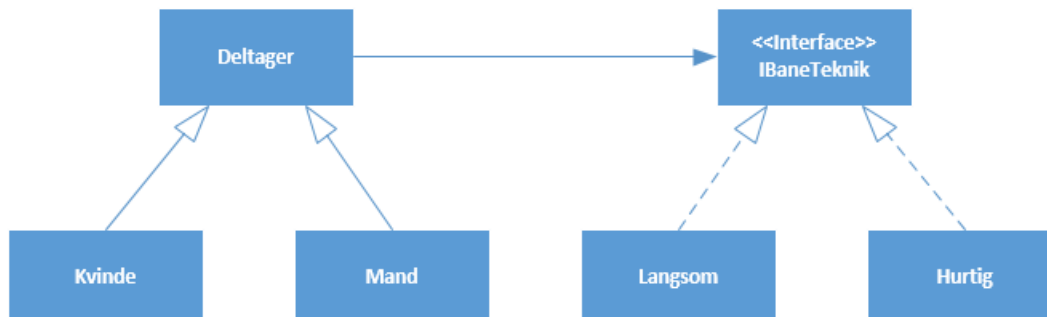
Man skal anvende Strategy Pattern, når det skal være muligt at ændre kontekstens behavioural runtime. Strategy pattern er derfor mere genanvendeligt.

Mens hvis kontekstens behavioural ikke skal ændres under runtime, altså bestemt under compile time.

3 Vis et designeksempel på anvendelsen af GoF Strategy

I eksemplet SKYDEBANE er konteksten, at der er 2 typer af DELTAGERE: KVINDE og MAND. De har hver deres måde at gennemgå skydebanen på. Den MAND er HURTIG og den KVINDE er LANGSOM. KVINDER kan blive HURTIG og MÆND kan blive LANGSOM, derfor skal dette kunne ændres dynamisk (RUNTIME).

Eksempel Skydebane



4 Redegør for, hvordan anvendelsen af GoF Template fremmer godt software design

GoF Template sikrer genanvendelse af kode der hvor det er muligt. det giver os en stærk struktur da det beskriver generelle strukturer og adfærd. Det er naturligt i et arvehierark at få generaliseret så meget adfærd for at få mindst muligt kode. Dette er også en strategi som er god når vi har mange nedarvede klasser som kun adskiller sig meget lidt i adfærd.

5 Redegør for, hvilke(t) SOLID-princip(per) du mener anvendelsen af GoF Strategy understøtter

- Open-Close (OPC)
- Singel Responsibility (SRP)
- Dependency Inversion (DIP)