5. Patterns 3: GoF Factory Method/Abstract Factory

# Redegør for, hvad et software design pattern er.

Design Pattern kan beskriver et problem der opstår igen og igen og derefter beskriver kernen i en løsning til det problem. Det der gør det til et mønster er så at man kan bruge løsningen om og om igen, uden at gøre samme.

Når vi anvender designregler skal det nemlig altid tilpasses. Et design pattern er derfor aldrig en færdig løsning på et problem, men en skabelon til løsning af et problem.

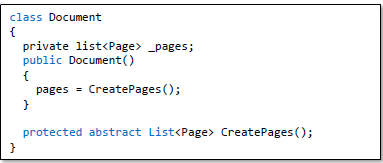
# Redegør for opbygningen af GoF Factory Method og GoF Abstract Factory.

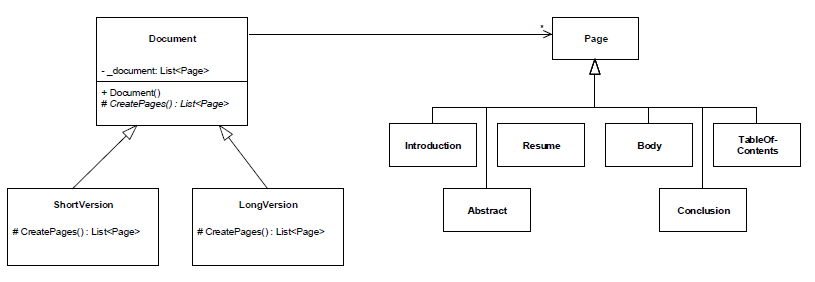
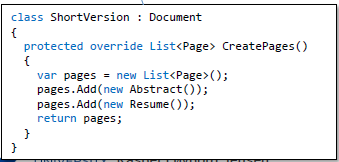
Den gennerelle ide med factories er at separerer oprettelsen af et object fra dens brug (fra kontrolleren). Der er forskellige former for factories, men her ses der på Factory Method og Abstract Factory.

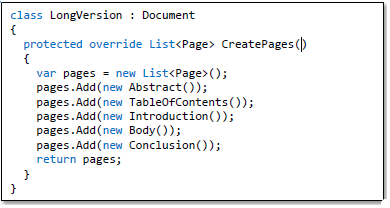
## Factory Method

## Factory Method: Definere et interface til at oprette en eller flere objekter, men lader klasserne der implementere interfacet afgøre hvilke objekter der skal instantieres.

Dette gør den ved at enkapsulere objekt oprettelse ved at lade Abstract Creator have en Factory Method som subklasser nedarver. Det er så op til subklasserne at implementere factory Method. Factory bruger således arv til at skabe et nyt objekt ved at subklasser implementerer Factory Method.



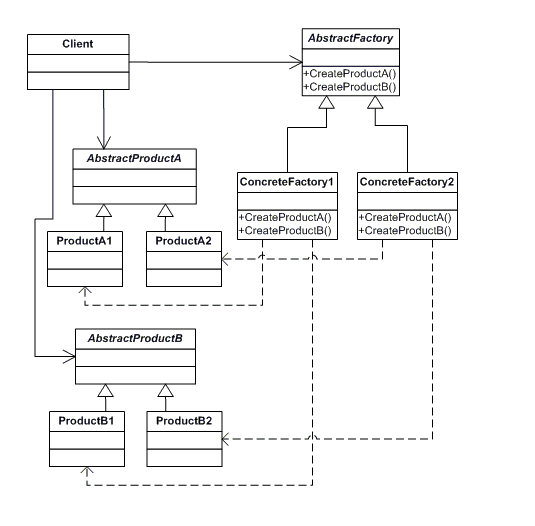




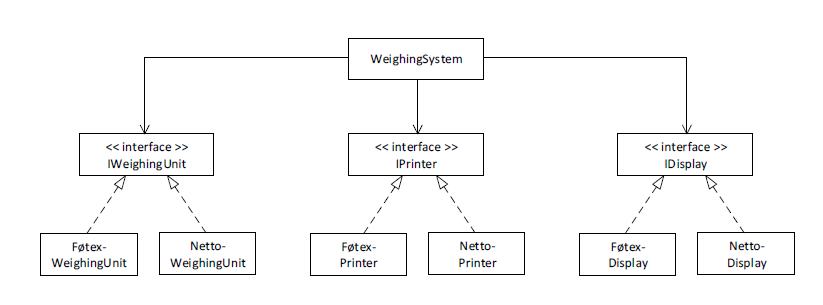
Abstract Factory  
Abstract factory pattern er et creational pattern, som står for oprettelsen af objekter og lige som i factory method ønsker vi at oprettelsen af et objekt fra brugen af et objekt.

Formålet med abstract factory pattern er at lave et interface til oprettelse af familier af objekter, uden at specificere deres konkrete klasser.

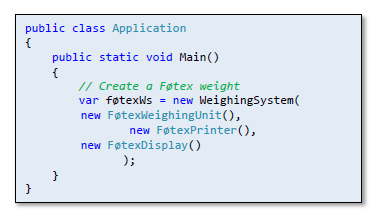
I stedet for at en klasse selv skal kende til hvilke afhængigheder den skal oprette, kan den få en abstract factory med i dens konstruktor, som så skaber en sammensætning af afhængigheder.



# Giv et designeksempel på anvendelsen af GoF Abstract Factory.



**Uden Factory**



**Med Factory**

