

# Opgaver torsdag den 17. februar

## Opgave 1

### Gruppearbejde Head First Design Patterns, Kap. 1

#### 1) Side 1-7

Den oprindelige løsning med nedarvning og interfaces

- Diskuter de beskrevne løsninger: den med nedarvning(side 3) og den anden med interfaces(side) (For at sikre at I alle i gruppen har forstået de to løsninger)
- Hvorfor er disse løsninger ikke gode løsninger i eksemplet Duck? (I mange andre situationer er nedarvning og/eller interfaces en god løsning)

#### 2) Side 8-21

Diskuter isolering af det der ændrer sig fra det der er konstant

Den nye løsning med delegation

- Diskuter den nye løsning
- Hvilke designprincipper anvendes? Formuler disse med egne ord.
- Udpeg de vigtigste dele i den færdige kode.
- Hvordan kan opførslen for "the ducks" ændres på kørselstidspunktet?

#### 3) Side 22-24

Strategy Pattern

- Indsæt IS-A, HAS-A og Implements på pilene I diagrammet s. 22
- Kommentér Design Principle "Favor composition over inheritance" s. 23. Formuler dette med egne ord.
- Forklar Strategy Pattern side 24

#### 4) Side 25-32

Sammendrag af kapitlet

- Løs Design Puzzle på side 25
- Formuler med egne ord en opsummering af siderne 26-28.

## **Opgave 2**

Diskuter "Opgaver mandag den 22. november " opgave 3.c i relation til Strategy pattern (Opgaven med Ordre - Discount)

## **Opgave 3**

Diskuter "Opgaver onsdag den 24. november" opgave 5 i relation til Strategy pattern (Opgaven med Moms)

Modificer løsningen til opgave 5, så Strategy pattern nu bruges på moms beregningen. Sammenlign de to løsninger.

## **Opgave 4**

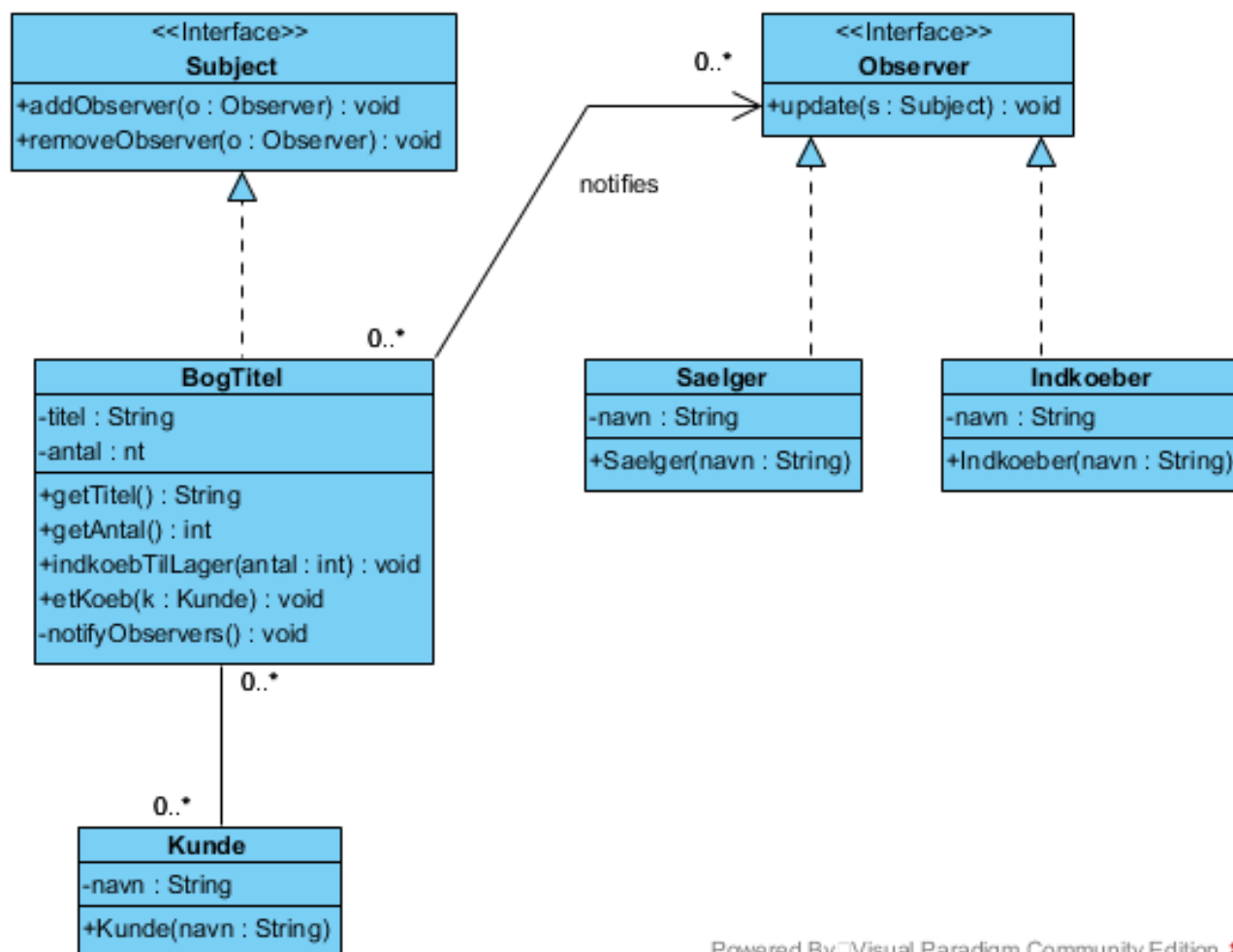
Lav en Kunde klasse med attributterne navn og nummer (lav constructor, getter og setter samt toString() - metoder)

Kunde klassen skal implementerer Comparable<Kunde> interfacet. Anvend Strategy Pattern sammen med Comparator<Kunde> interfacet til at implementere compareTo() metoden, så kunder kan sammenlignes på enten nummer eller navn.

Afprøv Kunde klassen i en main() metode.

## Opgave 5

Denne opgave omhandler et system til håndtering af salg af bøger. Systemet skal anvendes til såvel at holde lageret af bøger opdateret samt til at understøtte sælgernes mulighed for at sælge flere bøger til kunderne. Systemet skal programmeres i henhold til nedenstående klassediagram, der bl.a. indeholder den variation af Observer pattern – at en observer kan observere på flere subjects.



Powered By Visual Paradigm Community Edition

Der er to meget væsentlige funktionelle krav til systemet. For det første skal systemet sørge for at give besked til indkøberen, når der skal bestilles bøger ved forlaget. Dette skal ske, når der er færre end 6 bøger på lager af en bestemt titel. For det andet skal systemet ved salg af en bog kunne give information til sælgeren om, hvilke andre bøger der er solgt til kunder, der også har købt den netop solgte bog.

Dvs.

- at metoden `etKøb(k: Kunde)`: void på `BogTitel` har følgende specifikation:  
Associeringen mellem den aktuelle bogtitel og kunden `k` er opdateret, bogens antal er reduceret med en og alle observers skal have information om, at bogen er solgt.
- at metoden `update(Subject s)`: void på `Indkøber` har følgende specifikation:  
Hvis der er mindre end 6 bøger tilbage af bogtitelen `s`, udskrives på skærmen, at der skal bestilles 10 bøger med den pågældende titel. Endvidere registreres med det samme at der er købt 10 bøger til lageret.
- at metoden `update(Subject s)`: void på `Sælger` har følgende specifikation:  
Der er udskrevet en liste på skærmen med titlen på de bøger, der er købt af andre kunder, der også har købt den netop solgte bog `s`. Listen må ikke indeholde den samme titel flere gange. Endvidere skal listen ikke indeholde titlen på den aktuelle bog.

a) Lav ud fra klasseerklæringerne en programstump der kan simulere følgende:

*En sælger med navn Hansen.  
En indkøber med navn Jensen  
En bog med titel Anders And i 6 eksemplarer  
En bog med titel Java i 8 eksemplarer  
En kunde med navn Læsehest1  
En kunde med navn Læsehest2  
En kunde med navn Læsehest3*

*Hansen er interesseret i at vide, når et eksemplar af bogen Anders And er solgt og få at vide hvilke andre bøger der er solgt til kunder, der også har købt Anders And.  
Indkøber Jensen skal genbestille Anders And og Java, når der er færre end 6 eksemplarer*

*Bogen Java købes af Læsehest1  
Bogen Java købes af Læsehest2  
Bogen Java købes af Læsehest3  
Bogen Anders And købes af Læsehest1  
Bogen Anders And købes af Læsehest2  
Bogen Anders And købes af Læsehest3*

- b) Tilføj samtidig de link-attributter og de metoder, som er nødvendig på klasserne `BogTitel` og `Kunde` for at kunne vedligeholde og anvende associeringen mellem `BogTitel` og `Kunde`.
- c) Programmer klasserne `BogTitel`, `Kunde`, `Indkøber` og `Sælger`.
- d) Afprøv programmet fra a)