Systemudvikling modul 3 Domænemodel

# **Opgaver, domænemodel**

## **Opgave 1: Find kandidater til klasser i et ordrestyringssystem**

I denne opgave skal I finde kandidater til klasser dels ud fra workflow (fra IT forundersøgelsen) og dels ud fra *flow of events* i fully dressed beskrivelsen af: *Registrer ordre*



Figur : Forretningsdiagram for Ordrestyring

**Use case: Registrer ordre**

**Flow of events i fully dressed beskrivelse**

1. Use-casen starter med at en kunde henvender sig for bestille varer
2. Ekspedienten opretter en ny ordre
3. Systemet tilføjer ordre nr, dato samt status (= modtaget).
4. Ekspedienten angiver vareid og antal (hvis flere end 1)
5. Systemet returnerer varebeskrivelse , pris, deltotal samt løbende total
6. Når alle varer er registreret beder ekspedienten kunden om leveringsoplysninger
7. Ekspedienten angiver kunde navn, adresse samt tlf nr
8. Systemet validerer oplysningerne
9. Ekspedienten afslutter orden
10. Systemet gemmer ordren
11. Ekspedienten beder om en ordreseddel og faktura
12. Systemet udskriver en bekræftelse
13. Lav en liste over potentielle klasser ved at kigge på navneord (brug nedenstående skema)
14. Gå listen igennem og udvælg klasser efter om der kan svares *JA* til følgende:

* Er der attributter eller hændelser på klassen?
* Er der flere objekter af klassen?
* Er klassen fra problemdomænet – eller er det f.eks. en *udskrift* eller *aflæsning*?

1. Kontroller om der mangler klasser

* Kan alle hændelser registreres på de valgte klasser jf. ovenstående diagram over workflow. F.eks. at det kan registres at ordren er pakket og hvornår det er sket?

| **Kandidat til klasse** | **Vurdering** | **Medtages (Ja/nej)** |
| --- | --- | --- |
| **Kunde** |  |  |
| **……** |  |  |
|  |  |  |

## **Opgave 2: Domænemodel for ordrestyring, boghandel**

I nedenstående figur vises et første udkast til en domænemodel og et objektdiagram. Start med at gøre nedenstående **domænemodel** færdig:

1. Tilføj en vareklasse ved brug af en associering som navngives
2. Tilføj relevante attributter og multiciplicitet/kardinalitet



1. Kontroller om nedenstående kan registreres ved at gøre objektdiagrammet færdigt. Objektdiagrammet udledes ud fra jeres domænemodel. Finder I ud af der mangler klasser/attributter påføres de domænemodellen.

* Hans har købt Satzinger ”OOA&D” bog samt Larmans ”UML” bog på ordre 101 den 15/7. Bøgerne er pakket den 16/7 og sendt den 19/7. Han har betalt den 21/7
* På ordre 110 har Hans den 26/7 købt en java bog. Bogen er ikke pakket endnu
* Grethe købte ”OOA&D” bogen på ordre 102 den 25/7. Bogen er pakket samme dag og sendt 26/7. Betaling er ikke modtaget endnu.
* Knud er registreret som kunde den 20/7, men har ikke handlet endnu

## **Opgave 3: Associeringsklasse, boghandel**

Tag udgangspunkt i jeres domænemodel fra sidste opgave. Der er en mange til mange associering mellem Order og Book. Det viser sig, at der på ordren kan der være et forskelligt antal af de varer der købes, f.eks. har Hans købt 2 Java bøger og 1 UML bog på samme ordre. Java bogen er desuden solgt i 3 eksemplarer til Grethe.

1. Hvordan kan ”antal solgte” registreres i domænemodellen ved tilføjelse af en associeringsklasse, som I f.eks. kalder for delordre/ordrelinie (PartOrder/SalesLine)?
2. Juster domænemodellen fra opgave 2.

## **Opgave 4: Aggregering og beskrivelses-klasse mønstret, bogregistrering**

I skal nu vurdere alternative domænemodeller for en bogregistrering. Der er to muligheder:

Alternativ 1: En bogklasse med attributterne: ISBN, titel, forfatter, pris, eksemplar nr.

Alternativ 2: En bogklasse med attributterne: ISBN, titel, forfatter og pris som associeres til en eksemplarklasse med attributten eksemplar nr.

1. Lav en domænemodel for hvert alternativ
2. For at vælge løsning skal I lave objektdiagrammer af de to forslag til domænemodeller med følgende instanser:

Bogen med *ISBN=0-13-148906-2, titel = Applying UML and Patterns, forfatter = Larman og pris = 400 kr.* findes i 3 eksemplarer med numrene 1,2 og 3

* Hvad er ulemperne ved forslag 1?
* Hvis bogens pris ændres, hvor mange steder skal man ændre i forslag 1 i forhold til forslag 2?
* Hvad kan der være af problemer ved at skulle opdatere samme oplysning flere steder?
* Hvis alle (eller sidste) eksemplar slettes, hvad er så forskellen mellem forslag 1 og forslag 2?

Forslag 2 er et hierarkisk mønster som hedder *beskrivelses-klasse mønstret.* Da der er tale om et meget tæt forhold mellem beskrivelsen og registreringen af den fysiske genstand (eller eksemplar), anvendes en **aggregeringsstruktur** til at signalere dette

## **Opgave 5: Beskrivelses-klasse i domænemodellen for ordrestyring, hvidevarefirma**

Et hvidevarefirma ønsker et ordrestyringssystem.

Det viser sig at domænemodellen fra opgave 4 også kan bruges til at registrere afgivne ordre i hvidevarefirmaet

På grundlag af de afgivne ordre, bestiller hvidevarefirmaet varer hjem og ved ankomsten registreres serienr i systemet. Når kunden afhenter varerne (eller de leveres hos kunden) registreres hvilke fysiske varer, der er leveret hos kunden, ved at påføres serienr til ordren (af garantihensyn):

1. Udvid domænemodellen fra ordrestyringssystemet således, at eksemplarerne kan registreres med de påførte serienumre, efterhånden som de kommer hjem.
2. Endvidere skal der kunne registreres hvilken kunde, der har fået eksemplarer leveret.
3. Endelig skal I genoverveje associeringen mellem Order og PartOrder. Kan der argumenteres for, at det er en aggregering i stedet?

NB. Bestillingen af varer hos leverandøren skal ikke modelleres!

## **Opgave 6: Generalisering, studenterregister**

Som det fremgår af nedenstående klassediagram, er der en del fælles information mellem studerende og lærer

1. Indfør en superklasse til at modellere fælles attributter
2. Juster domænemodellen



## **Opgave 7: Rolle og generalisering, ordrestyring, hvidevarer**

I hvidevarefirmaet ønskes nu også registreret, hvilken medarbejder der har stået for ordren

1. Hvis medarbejderen og kunden har samme attributter hvordan kan dette så modelleres?
2. Kunden registreres ved id, navn, adresse og kontokort og ekspedienten registreres ved id, navn, adresse og lønkonto. Hvordan kan dette modelleres ved at indføre en superklasse (generalisering)?

## **Opgave 8: Domænemodel for Kajs Biler**

Med udgangspunkt i de opgaver I har løst tidligere vedr. IT forundersøgelse, use cases og use case beskrivelser for Kajs Biler, bedes I:

1. Finde og udvælge klasser til Kajs Biler
2. Lave en domænemodel