

INF1000 - Obligatorisk innlevering 7

Høsten 2016, IFI UiO

Frist: 6. November 2016 kl 22:00

Tema denne uka: *Et større objektorientert program.*

Administrasjon av eierskap og utlån av DVD-er

I denne oppgaven skal du designe og skrive et program som holder orden på dine egne og andres DVD-er. Programmet skal støtte utlån av DVD-er. Det vil si at det både skal være mulig å låne ut DVD-er en person eier, og låne DVD-er fra andre. Programmet skal holde styr på hvilke DVD-er en person eier, hvilke DVD-er personen eier som er utlånt til andre og hvilke DVD-er personen låner fra andre. Du kan anta at personer har unike navn.

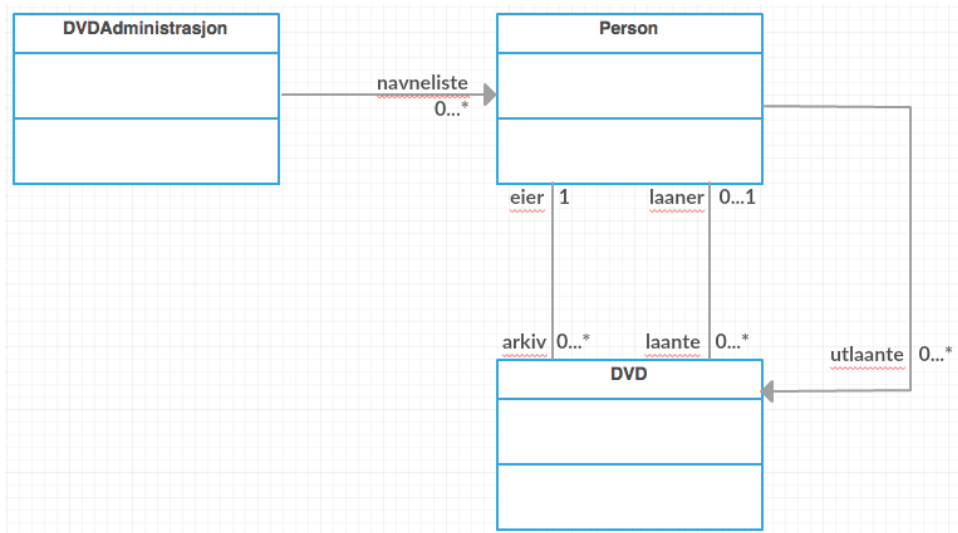
I denne oppgaven skal du bruke HashMap i samtlige lister i programmet ditt (og ikke ArrayList/array).

Det er lurt å lese gjennom hele oppgaveteksten først for å forstå bedre hva programmet skal gjøre.

1 Design av programmet

Det første steget i å lage et større objektorientert Java-program er å planlegge hvordan du skal løse det - lage et design til programmet ditt. En slik slagplan vil gjøre implementasjon enklere, ved at du har tenkt nøye gjennom oppbygningen før du begynner kodingen. Som et ledd i planleggingen av programmet ditt skal du bruke UML-diagrammet (klassediagram)

i figur 1 som utgangspunkt, og lage ditt eget diagram. Diagrammet viser hvilke klasser du minimum skal bruke for å løse oppgaven, samt relasjonene mellom disse. Din første oppgave er å bestemme hvilke klasser du skal lage og hva relasjonene mellom disse skal være. Du står fritt til å legge til klasser/relasjoner du mener er nyttig for løsning av oppgaven.



Figur 1: UML-diagram (forslag til hva som minst må med)

Legg merke til at det her er tre relasjoner mellom Person og DVD: 1) er for eierskapet, 2) DVD-er du selv eier og låner bort til andre og 3) DVD-er du låner fra andre og som andre eier.

Relasjon 1 innebærer blant annet at alle DVD-instanser må ha en referanse til en Person-instans (kalt "eier").

Utfylling av UML-diagrammet

Vær oppmerksom på at ikke alt i UML-diagrammet er eksplisitt angitt, deriblant metoder og variabler. Fyll derfor ut UML-digrammet ditt med de klassene du tilføyer, hvilke variabler og metoder som trengs. Husk type for variabler, returverdier og parametere.

Diagrammet du lager skal leveres som en bildefil (JPEG, PNG, PDF). Det enkleste er nok å gjøre tegning og utfylling med penn og papir og deretter scanne det eller ta bilde av det med telefonen. Du står også fritt til å bruke tegneprogrammer om du ønsker det.

2 Opprette datastruktur og teste programmet

Opprett deretter datastrukturen du har beskrevet i diagrammet ditt fra oppgave 1. Dette betyr at du skal lage én fil for hver av klassene i diagrammet ditt, og implementere nødvendige metoder, instansvariablene og lignende. Hver klasse skal ha en kommentar på 1-3 linjer som beskriver klassens formål og ansvar. Generelt skal programmene dine være kommenterte med korte, konsise kommentarer. Underveis er det viktig at du tester at programmet ditt gjør som forventet. Opprett derfor objekter av klassene og test de hver for seg. Husk også å teste at objektene fungerer godt sammen. Av testprogrammene du bruker skal du velge ut 3 testtilfeller som viser at programmet ditt virker for de mest alminnelige tilfellene. *Hvert av disse 3 testtilfellene skal ligge i et separat javaprogram.* Du må lage så mange testprogrammer som du ønsker, men du skal bare levere de 3 utvalgte nevnt ovenfor. Hvert testprogram skal starte med en kommentar, som beskriver testen du gjør på 1-5 linjer.

Testprogrammene dine skal **ikke** være med i UML klassesdiagrammet.

3 Brukerdialog med kommandoløkke

Neste del av programmet ditt er å lage en kommandoløkke som printer ut en meny og tar imot kommandoer fra bruker. Et eksempel på kjøring av programmet finner du på side 7 og utover.

Kommandoløkken skal tilby følgende kommandoer:

- 1: Ny person
Oppretter en ny person. Programmet skal be om personens navn.
- 2: Kjøp
Lager en ny DVD. Programmet skal be om

1. hvilken person som har kjøpt DVD-en
2. hva tittelen på DVD-en er

Husk å sjekke at personen ikke kjøper flere eksemplarer av samme DVD.

3: Laan

Brukes når en person vil låne en DVD av en annen. Programmet skal be om

1. hvilken person som vil låne
2. hvilken person det skal lånes fra
3. tittelen på den aktuelle DVD-en

En person skal ikke kunne ha fremleie av lånte DVDer.

Husk å sjekke at eieren virkelig eier den angitte DVD-en og at den ikke allerede er utlånt. Husk også at en person ikke kan låne av seg selv.

4: Vis person

Viser en oversikt over alle DVD-ene en spesifikk person eier og låner for øyeblikket. Programmet skal be om hvilken person det skal vises en oversikt for. Hvis navnet angis som '*' (bare stjernen), skal oversikten omfatte alle personene i systemet.

Oversikten skal for hver person vise

1. tittelen på alle DVD-ene som han eller hun eier
2. hvilke av disse som for tiden lånes ut og hvem som låner dem
3. hvilke DVD-er personen selv har lånt fra andre og hvem de er lånt fra

5: Vis oversikt

Gir en oversikt over alle personene i arkivet. For hver person skal det angis

1. antallet DVD-er han eller hun eier
2. antallet han/hun har lånt ut for tiden
3. antallet han/hun selv har lånt

6: Retur

Skal benyttes når en lånt DVD skal leveres tilbake til dens rettmessige

eier. Programmet skal be om hvilken person som har lånt DVD-en og tittelen til DVD-en.

Husk å sjekke at DVD-en har vært utlånt til den angitte personen.

7: Avslutt

Avslutter kommandoløkken.

4 Innlesing fra fil

I programmet ditt skal du også støtte innlesing fra fil. Når programmet starter, skal det lese filen `dvdarkiv.txt`. Filen inneholder informasjon om hvilke personer som skal være med i systemet, hvilke DVD-er de eier, om de lånes ut og eventuelt hvem de lånes ut til. Se neste avsnitt for hvordan denne filen skal se ut.

Filen `dvdarkiv.txt` skal ha følgende struktur:

- Alle personene som er inkludert i programmet skal listes opp først. Hvert navn skal stå på sin egen linje. Etter at alle navnene er listet opp, skal det stå en `-(bindestrek)` på sin egen linje.
- Deretter skal en og en person med DVD-er listes opp. For hver person vil det først stå navnet på en linje. På hver følgende linje under navnet kommer tittel på DVD-er personen eier (ingen hvis en person ikke eier noen). Hvis en tittel er leid ut, vil tittelen begynne med en `*(stjerne)`. Da vil den som låner DVD-en stå på neste linje. Om den ikke er lånt ut, vil da tittelen stå uten `*(stjerne)`.
- Etter hver persons opplisting av sine eide DVD-er vil det stå en `-(bindestrek)`.

Filen som ble skrevet fra eksemplet bakerst(se side 7) vil se slik ut:

```
Per
Kari
-
Per
The hobbit 1
*The music man
Kari
-
```

Kari

Denne eksempelfilen vil ved innlesing gi en datastruktur der vi har personene Per og Kari, hvor Per eier DVD-ene 'The hobbit 1' og 'The music man' og Kari har lånt sistnevnte.

Om du ønsker å teste med en større fil kan du laste ned dvdarkiv2.txt.

Legg vekt på oversiktlig og lettest kode. Det forventes at kommentarer, variabel- og metodenavn er gode og forklarende til hva de representerer.

Tilleggsoppgave (frivillig): Når bruker har avsluttet programmet, skal den oppdaterte datastrukturen skrives tilbake til filen dvdarkiv.txt. Dette er lurt å trene på, men ikke noe krav for å bestå oppgaven. Klassen PrintWriter kan være nyttig for skriving til fil.

Fremgangsmåte for innleveringer i INF1000

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
 - (a) Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
 - (b) Hvor lang tid (ca) brukte du på innleveringen?
 - (c) Var det noen deler du ikke fikk til? Hvis ja:
 - Hvilken del av oppgaven er det som ikke fungerer?
 - Hvorfor tror du at det ikke fungerer?
 - Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
 - (d) En liste med de tre testprogrammer, som du har spesielt valgt ut i avsnitt 2.
2. Logg inn på Devilry.
3. Lever alle de aktuelle .java-filene, UML-diagrammet og README.txt *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

Samarbeid

Husk at samarbeid om ideer, strukturer etc er helt OK, men du skal aldri kopiere Java-kode fra andre. Reglene for dette finner du på <http://www.uio.no/studier/admin/obligatoriske-aktiviteter/mn-ifi-oblig.html>.

Kjøreeksempler

Dette er et eksempel på en interaksjon med programmet; i dette tilfellet er arkivet tomt i starten.

```
> java DVDArkiv
MENY FOR DVD-ADMINISTRASJON
1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 1
```

```
Hva heter den nye personen?
> Per
```

```
MENY FOR DVD-ADMINISTRASJON
1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 2
```

```
Hvem har kjøpt DVD-en?
> Per
```

```
Hva er tittelen paa DVD-en?
> The hobbit 1
```

```
MENY FOR DVD-ADMINISTRASJON
1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
```


5: Oversikt.
6: Retur.
7: Avslutt.
> 2

Hvem har kjøpt DVD-en?
> Per

Hva tittelen på DVD-en?
> The music man

MENY FOR DVD-ADMINISTRASJON

1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 1

Hva heter den nye personen?
> Kari

MENY FOR DVD-ADMINISTRASJON

1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 3

Hvem vil laane DVD-en?
> Kari

Hvem skal DVD-en laanes fra?
> Per

Hva er tittelen paa DVD-en?

> The music man

MENY FOR DVD-ADMINISTRASJON

1: Ny person.

2: Kjop.

3: Laan.

4: Vis.

5: Oversikt.

6: Retur.

7: Avslutt.

> 5

Person: Per

Eier: 2

Laant: 0

Utlaant: 1

Person: Kari

Eier: 0

Laant: 1

Utlaant: 0

MENY FOR DVD-ADMINISTRASJON

1: Ny person.

2: Kjop.

3: Laan.

4: Vis.

5: Oversikt.

6: Retur.

7: Avslutt.

> 3

Hvem vil laane DVD-en?

> Kari

Hvem skal DVD-en laanes fra?

> Johannes

Det er ingen personer som heter Johannes.

MENY FOR DVD-ADMINISTRASJON

1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 4

Hvilken person vil du se? (* for alle)
> Kari

Person: Kari
Kari eier ingen DVD-er.
Kari låner ikke bort noen DVD-er.
DVD-er Kari låner:
The music man er lånt fra Per.

MENY FOR DVD-ADMINISTRASJON

1: Ny person.
2: Kjøp.
3: Laan.
4: Vis.
5: Oversikt.
6: Retur.
7: Avslutt.
> 7

Programmet avsluttes.

Legg merke til at brukeren i ett tilfelle oppga gale data; da kom det en feilmelding og arkivet ble ikke endret.