INF1010 2013 — Obligatorisk oppgave 2

Versjon 1.0 - Oppgaveteksten er helt ny og vi regner med det er ting som vil bli rettet. Både forklaringer og skrivefeil. Har du spørsmål om oppgaven, eller vil påpeke feil, bruk INF1010-bloggen. Her vil du etterhvert finne nyttig tillegsinfo om oppgaven, og ikke minst se spørsmål og svar.

Viktig for å løse denne oppgaven: Samspill mellom objekter ved hjelp av metoder, signatur og returverdier (særlig av pekertyper) for metoder, innkapsling (private/public), klasser og konstruktører. Nytt i denne obligen er tekstsammenligning. Klasser med (generisk) parameter for den som vil. Les Stein Gjessings notat Enkle generiske klasser i Java*.*

Vi skal bruke personklassen fra oblig 1, men vi skal utvide den slik at personobjektene inneholder informasjon om hva personene er interessert i (samler) av bøker og/eller plater.

Oppgave a. Skriv klassene Bok og Plate

Objekter av klassen Bok har tre egenskaper, to tekststrenger som er bokas forfatter og tittel og et årstall som er utgivelsesår.

Objekter av klassen Plate har tre egenskaper, to tekststrenger som inneholder navn på artist og tittel, samt et heltall som er antall spor på plata.

Alle egenskaper (attributter) i klasser skal være skjult, dvs. de skal deklareres private. Alle egenskapene får verdier i konstruktøren. Hvis de skal være synlige eller kunne forandres utenfra (leses/skrives) etter opprettelsen, må det skje ved hjelp av metoder som deklareres public.

Oppgave b. Utvid Person-klassen

Klassen Person fra oblig 1 skal utvides ved å gi den variable og metoder som gjør det mulig å angi og å finne ut hva en person er interessert i å samle på og om personen har særinteresser. Klassen skal ha en peker til en array som kan peke på objekter av klassen Bok, og en array som kan inneholde objekter av klassen Plate. Disse arrayene skal brukes til å ta vare på bøker eller plater personen er samler av.

Person-klassen skal også ha tre nye metoder:

```
public void samlerAv(String smlp, int ant)
// Det opprettes en ny array i personobjektet for å samle på
// - bøker hvis parameteren smlp er bøker
// - plater hvis parameteren smlp er plater
// ant er antall bøker/CD-er som personen kan samle på
// ingenting gjøres hvis smlp peker på noe annet

public void megetInteressertI(String artist)
// merker personobjektet som særlig interessert
// i plater av artisten med dette navnet hvis personen samler på plater

public void megetInteressertI(int eldreEnn)
// merker personobjektet som særlig interessert
// i bøker som er eldre enn årstallet parameteren angir hvis personen
// er samler av bøker
```

Du står selvfølgelig fritt til å utvide med mer enn dette, for at oppgavene nedenfor kan løses enklere. Husk at alle variable, også dine egne, skal være skjult utenfor klassen.

Oppgave c. Testklasse som oppretter personobjektene

Skriv en testklasse med en Person-array med alle personene i programmet. Du står fritt i å lage så mange personer du vil, og hva de skal hete, hva de samler på og er særlig interessert i, men her er minimumskrav:

Minst 7 personer. 6 av personene er samlere. 3 samler både på bøker og plater. En av disse er meget interessert i plater med bandet Queen (artist) og i bøker utgitt før 1946, en annen er opptatt av artisten Silya Nymoen, mens den tredje samler særlig på bøker eldre enn 1900. En samler på plater og er meget interessert i artisten Bob Dylan, en samler bare på plater og en er bare samler av bøker. En person er ikke interessert i hverken bøker eller plater. Alle arrayer for bøker og plater skal ha lengde 5.

Oppgave d. Utvid personklassen med vilDuHaGaven-metodene

En person får en gave (bok eller plate) ved at metoden vilDuHaGaven(...) kalles i personobjektene. Det er en del av oppgaven at du skal bestemme resten av signaturen. Metoden vurderer gaven som parameteren peker på og legger den til bok- eller platearrayen hvis personen er interessert. Hvis gaven beholdes skal vilDuHaGaven returnere null. Hvis ikke, returnerer den bare boka eller plata som ble tilbudt.

Her er reglene for om en person beholder gaven eller ikke:

Ingen beholder gaver man ikke samler på. Ingen beholder noe man har fra før. Ingen beholder en bok eller plate man samler på hvis arrayen er mer enn halvfull av bøker/plater fra før, med mindre personen er særlig interessert i boka eller plata (som markert av megetInteressertI-metodene). Hvis arrayen er full, vil man bytte ut en plate/bok en ikke er så interessert i med en man er meget interessert i. Da vil vilDuHaGaven returnere et annet objekt enn det som var i parameteren da den ble kalt.

Lag først den for plater og test at den virker. Deretter lager du den for bøker. Eller omvendt, om du heller vil det.

Oppgave e. Avslutt testklassen.

Tilslutt skal vi opprette bok- og plateobjekter og tilby disse til personene.

Utvid tilslutt testklassen ved å opprette en mengde (array, hashMap e.l.) objekter som representerer bøker og plater. Husk å få med plater med artister som noen er meget interessert i og bøker med årstall som noen er særlig interessert i. Testprogrammet forsøker å gi alle gavene ved å tilby dem én etter én til alle personene i personarrayen etter tur. Vi er kvitt gaven når vilDuHaGaven-metoden returnerer null. Hvis den ikke er null skal gaven tilbys neste person i personarrayen. Dette gjentas så lenge ingen beholder gaven. Blir man ikke kvitt en gave etter å ha spurt alle, kan man kaste den og gi en melding om det til terminalen, eller lage en egen beholder for upopulære gaver.

Obligatorisk oppgave slutt

Noen råd om utviklingen av programmet.

Ikke skriv alt før du begynner å teste. Selv om testing underveis tar tid, sier erfaringen at summen av tida man bruker faktisk blir mindre hvis man programmerer og tester i flere etapper. Det er mange måter å gjøre dette på en god måte. Her er et forslag:

Start med å lage en ny personstruktur og test denne. Skriv deretter klassen Bok, og utvid personklassen med det som har med samling og interesse for bøker å gjøre. Lag en enkel vilDuHaGaven som bare legger boka i arrayen i det personobjektet som metoden kalles i. Test at dette virker. Lag så den delen av testprogrammet som merker noen av personobjektene med interesse for bøker. Test om du får gitt bøker til alle som er interessert, og ingen andre, vha. en forenklet vilDuHaGaven. Testing gjøres enklest ved å utvide skrivUt-metodene med ny info og kalle på den. Går dette bra, kan du fullføre vilDuHaGaven for bøker. Test at den virker som den skal. Opprett nå flere bokobjekter og test igjen. Da er vi snart i mål og må bare fullføre programmet ved å legge til det som har med plater å gjøre.

Det er også mange andre rekkefølger man kan velge. Legg merke til at dette forslaget avviker en god del fra rekkefølgen av oppgavene a-e. Det viktige er at du begynner med det du synes virker lett før du går i gang med det som virker mer komplisert og tester etter hvert. Når obligen er ferdig spiller det ingen rolle hvilken rekkefølge du valgte, men det kan ha stor betydning for hvor lang tid du bruker. Lykke til!

Frivillige ekstraoppgaver for de som vil

Bruk generiske klasseparametre for å lage beholdere som kan inneholde hva som helst, også bøker og plater. Jf. Stein Gjessings notat. Bruk gjerne grensesnitt til å la bok- og plateklassene gaveegenskaper ved å la disse klassene implementere et grensesnitt. Da kan man lagre gavene i én beholder og man slipper å lage flere versjoner av metodene som har med gaver å gjøre.

La personene være «snillere»: En person som er sammen med eller forelsket i en annen vil alltid tilby gaven til kjæresten (sammenmed eller forelsketi) før hun/han vurderer å beholde gaven selv. Hvis hun/han ikke er interessert i å beholde gaven selv tilbys gaven til de andre vennene i tilfelle noen av dem er interessert. (Pass på at du ikke får uendelige metodekall av vilDuHaGaven-metoden!)

For å få til dette må det opprettes kjennskaps- og vennskapsforbindelser mellom de 7 personene. Dette gjøres lettest ved å kalle på metoder som ble skrevet i oblig 1. Du kan f.eks. opprette en kjennskapsstruktur slik at følgende oppfylles:

Alle personer kjenner minst tre andre. En person har ingen uvenner. Alle andre har minst en de ikke liker. Ingen misliker alle de kjenner. En person er forelsket i en som hun/han ikke er sammen med. Forelskelsen er ikke gjensidig. Det er to par som er sammen. Disse kan ikke være uvenn med den de er sammen med. For å angi hvem som er sammen (bruk sammenmed-variabelen fra oblig 1), må du lage en ny metode som gir sammenmed riktig pekerverdi.