INF1010 2014 — Obligatorisk oppgave 3

Versjon 1.0—Skrivefeil blir rettet uten at versjonsnummeret endres. Hvis det oppdages feil i prekoden blir dette rettet med en kommentar øverst i den aktuelle filen. Lenke til prekodefilene finnes på bloggen. Denne obligen kan man ikke levere sammen.

Oppgaven bygger på oblig 1 og 2 samt listeeksemplet som lå til grunn for forelesningen om lenkelister 29. januar. En rettet versjon er lagt ut som prekode (ListeAvPersoner.java).

Programmet som skal leveres skal bygge på arbeidet du har gjort i de tidligere obligene, særlig oblig 1. Kort sagt går oppgaven ut på å gjøre forbedringer i listeklassen, opprette en instans av denne, legge personobjektene inn i lenkelista, skrive noen nye metoder i Person-klassen, og gjøre forandringer i andre metoder, særlig de som har med gaver å gjøre. Til slutt skal du hente gaver fra en ferdiglaget gavebeholder og «drysse» disse over personene. Du trenger ikke fjerne metoder fra oblig 1 og 2 (evt. andre attributter) som ikke blir brukt, men det er opp til deg.

Du bør lage tester selv for å teste at metodene i Person blir som forventet, men det blir publisert ferdig programkode (ofte kalt prekode) som obligretterne bruker for å teste programmet ditt for godkjenning. Koden som oppretter personer, relasjonene mellom dem og gaveeinteresser finnes som prekode i filen Personer.java.

For å representere gaver skal dere nå bruke et grensesnitt. Dere skal ikke skrive det selv, da hele grensesnittet er gitt her (finnes i prekoden i filen Gave.java):

```
public interface Gave {
  String kategori();
    // Returnerer en tekststreng som gjør det mulig å vite hva slags gave
    // dette er, f.eks. «bok», «plate», «vin», «sko», «sjokolade», «bil»...

  String gaveId();
    // Returnerer en tekststreng som identifiserer gaven. To gaver med
    // lik kategori og gaveId er samme gave (gjenstand).
}
```

Vi vet ikke mer om gaver enn at de er objekter av klasser som implementerer dette grensesnittet, men det er nok for å skrive metoden som avgjør om en person beholder gaven eller ikke. Denne metoden må skrives om for å tilpasses den nye typen (Gave). Reglene for om en person beholder gaven eller ikke forandres også noe. Detaljer følger lenger ned.

De konkrete gavene (som må være objekter av klasser som implementerer Gave) får man tak i ved å innlemme følgende kode i obligen (testklassen):

```
GaveLager gavelager = new Gavelager();
```

Da har vi en peker til et objekt med et lager av gaver. Vi får tak i gaver via en metode hentGave i GaveLager:

```
Gave nyGave = gavelager.hentGave();
```

Denne metoden returnerer gaver så lenge det er gaver igjen i lageret. Når det er slutt, returnerer den null. Dette gavelageret blir en del av prekoden. Vi

anbefaler at du lager en enkel klasse Gave med de samme metodene og din egen hentGave-metode for å teste før prekoden innlemmes.

Nedenfor følger detaljerte arbeidsoppgaver. For noen av oppgavene har rekkefølgen betydning, men mange kan gjøres i en annen rekkefølge. Først oppgaver knyttet til klassen Person:

- Alle personobjekter må ha en nestepeker. Den er helt essentiell for å få lenket sammen objektene. Må hete det samme som i ListeAvPersoner-klassen.
- Person skal ha en ny metode blirSammenMed med opplagt virkning.
- Gjør endringer i metodene i Person slik at personnettverket (relasjonene mellom personene som modelleres ved hjelp av pekere) som manipuleres av metodene oppfyller følgende regler:
  - Hvis Lisa er sammen med Khizar, må Khizar være sammen med Lisa.
  - En person trenger ikke være forelsket i personen hun er sammen med
  - En person kan være sammen med en han ikke liker
  - En person kan ikke være forelsket i en person hun ikke liker
  - Ingen kan være sammen med, forelsket i eller kjent med seg selv.
- kjenner-arrayen skal ha lengde 100. likerikke lengden 10.
- Lag en peker til en gavearray i Person. Denne skal hete mineGaver og skal brukes til å oppbevare gavene til personen. Lengden på denne settes av samlerAv.
- skrivUtAltOmMeg() utvides slik at den skriver ut hvem personen evt. er sammen med og til slutt skriver ut informasjon om hvilke gaver personen har. Bruk en linje per gave.
- Person skal ha en tekststreng som inneholder informasjon om hva slags ting/gaver en person er interessert i. Denne får verdi gjennom kall på samlerAv fra oblig 2. Denne teksten sammenlignes med gavens kategori (se nedenfor) for å avgjøre om en person er interessert i en gave eller ikke.
- Skriv om/lag metoder som bestemmer om en gave skal beholdes av personen eller ikke. Reglene som skal gjelde finnes nedenfor under overskriften *Regler for gavemottak.*

Endringer—og en ny metode—i listeklassen ListeAvPersoner:

- Skriv en ny metode som sjekker om et personobjekt (parameter) ligger i lista eller ikke (returnerer boolean).
- Tre metoder setter inn nye personer i lista. Lag en sjekk i alle tre, slik at personer som allerede ligger i lista ikke kan legges inn. (Bruk metoden du laget i forrige punkt).
- Skriv om settInnEtter slik at det sjekkes at parameteren denne virkelig peker på et objekt i lenkelista. Sjekk også at denne og inn ikke peker på samme objekt. (Hva skjer da?)

Testklassen vil stort sett være prekode. Men dette må du gjøre selv, lage:

- kode for å hente og legge personene inn i lenkelista. Dette gjøres på liknende vis som for gaver. Opprett et objekt av klassen Personer. Dette objektet har en metode hentPerson() som henter et og et personobjekt inntill det ikke er fler igjen. Da returner den null. Personene skal legges inn sist i personlista i den rekkefølgen de kommer fra hentPerson().
- kode som henter gaver og gir dem til alle personene. Du har (i prekoden) tilgang til en metode hentPersonnavn() som inneholder navn på alle personene. Gavene skal tilbys til alle personene i denne arrayen etter tur (bruk finnPerson) helt til gaven ikke kommer i retur eller det ikke er flere navn igjen.
- kode som skriver ut info om alle personene i lista (med relasjoner og gaver). Dette skal gjøres helt til slutt. (Lurt å teste med færre personer og gaver først...)

Regler for gavemottak

En person vil beholde gaven hvis gavens kategori er lik tekststrengen som bestemmer dette. Hvis personen er interessert *og har plass til den*, beholdes den og metoden returnerer null. Vi gjør personobjektene litt mer sofistikerte ved at en gave man ikke vil/kan beholde først tilbys til den man er sammen med, deretter til den man er forelsket i og til slutt til alle venner etter tur før den evt. returneres. Pass på å lage mekanismer slik at venner og kjærester ikke fortsetter å gi gaven videre! Dette kan løses på flere måter og er kanskje det vanskeligste punktet i hele obligen. Diskuter på labgruppene og med andre hvordan dette best kan gjøres.

Om prekode

Det blir lagt ut fire filer med prekode som du skal bruke:

1. listeklassen som skal endres litt og brukes for å lage lenkelista av personobjekter.
2. programkode for gavelageret
3. kode som oppretter en String-array med navn på personene. Du må selv skrive koden som oppretter personobjektene og legger alle personene inn i personlista.
4. kode som oppretter relasjoner mellom personene (etter at de er lagt inn i lenkelista) ved å kalle på metoder i personlista for å få tak i dem, og i personobjektene for å bestemme realsjonene mellom dem.

Om testing

Det er viktig og nødvendig at du tester programmet mens du utvikler det. *Før* du innlemmer prekoden! For testing av personobjektene kan du f.eks. bruke nettverket fra oblig 1. Test også at listemetodene fungerer korrekt før du henter inn personene fra prekoden. Når du tilslutt mener at alt er riktig, tester du med prekoden. Egne tester skal slettes før levering.