

Obligatorisk oppgave 3 INF1010 2013

Versjon 1.4 — 6.2.13 Forandret SkrivUtMegOgVenner og presisert blirVennMed i personklassen. Lagt til en skisse av mulig datastruktur. Fra nå blir kun feil rettet.

Nytt i denne obligen er lenkelister. Dette kan du lese om i notatet Lenkelister og beholdere av lenkelister. For listene i denne obligen, er figuren på side 14 spesielt relevant.

Du skal i denne oppgaven lage lenkelister av personobjekter. Hvert personobjekt skal ha en beholder med venner (også personobjekter). Datastrukturen i beholderen skal være en lenkeliste av personer. Du kan ikke bruke Javas LinkedList e.l. for å lage beholderen.

Programmet har altså to forskjellige personbeholdere. En personbeholder som inneholder *alle* personene i programmet. Denne er det bare en av. Så har hvert personobjekt sin egen vennebeholder (*mineVenner*).

Beholderen, som ikke trenger å være generisk fordi den bare brukes for å oppbevare personobjekter, skal ha 4 metoder som kan kalles utenfra:

```
public int antall()
// returnerer antall personer i beholderen

public void settInnPerson (Person p)
// setter p inn i beholderen hvis p ikke er der allerede

public boolean erIbeholder (String s)
// true hvis det finnes en person med navn s i beholderen

public void skrivAlle(String prefiks)
// skriver ut navnene til alle personene i beholderen under hverandre med
teksten prefiks foran hvert navn

    Personklassen må ha en peker til vennebeholderen. En person har (som
i tidligere obliger) et unikt navn (en String). Du kan fritt legge til andre
variable og metoder. Klassen skal skal ha disse variablene og metodene:

private PersonBeholder mineVenner;
private String navn;

public String hentNavn() // som før

public int antallVenner() // returnerer antall venner

public void blirVennMed(Person p)
// legger p inn i vennebeholderen hvis p er i personlista

public boolean erVennMed(String s)
// returnerer true hvis en av personene i vennebeholderen har navnet s

public void skrivUtMegOgVenner(String prefiks) {
    System.out.println(prefiks + navn);
    if (prefiks.equals(""))
        mineVenner.skrivAlle("_>_");
}
```

Alle personobjektene i programmet ligger i en lenkeliste. Du kan bruke beholderen du har laget til dette. (Hvis du bruker en egen klasse, skal klassen inneholde de samme metodene som beholderklassen pluss de du lager selv.)

For å teste programmet skal du lage en egen testklasse. Her opprettes nye personer. Personlista opprettes ved suksessive kall på `settInnPerson`. Vennskapsbeholderne fylles opp ved kall på `blirVennMed`. F.eks. slik (NB! Dette er bare et eksempel. Dette kan gjerne gjøres annerledes, særlig kan signaturer til metoder og konstruktør bli forskjellig):

```
...
Person p = new Person("Emil");
pl.settInnPerson(p); // pl peker på personlista
Person a = new Person("Ane");
pl.settInnPerson(a);
p.blirVennMed(a);
a = new Person("Ask");
p.blirVennMed(a);
...
```

For å bevare tilstandspåstand 2 nedenfor, må et personobjekt enten «vite om» personlista eller inneholde informasjon om objektet er satt inn eller ikke. Dette for at `blirVennMed` skal kunne sjekke om personen som skal bli venn ligger i personlista. (Siste kall på `blirVennMed` i eksemplet ovenfor bør gi feilmelding, siden «Ask», i motsetning til «Emil» og «Ane» ikke er lagt inn i personlista.)

Invariante tilstandspåstander:

1. ingen person er venn med seg selv
2. ingen har venner som ikke ligger i personlista
3. det finnes ikke to personer i personlista med samme navn
4. en persons venneliste har ikke to pekere til samme person

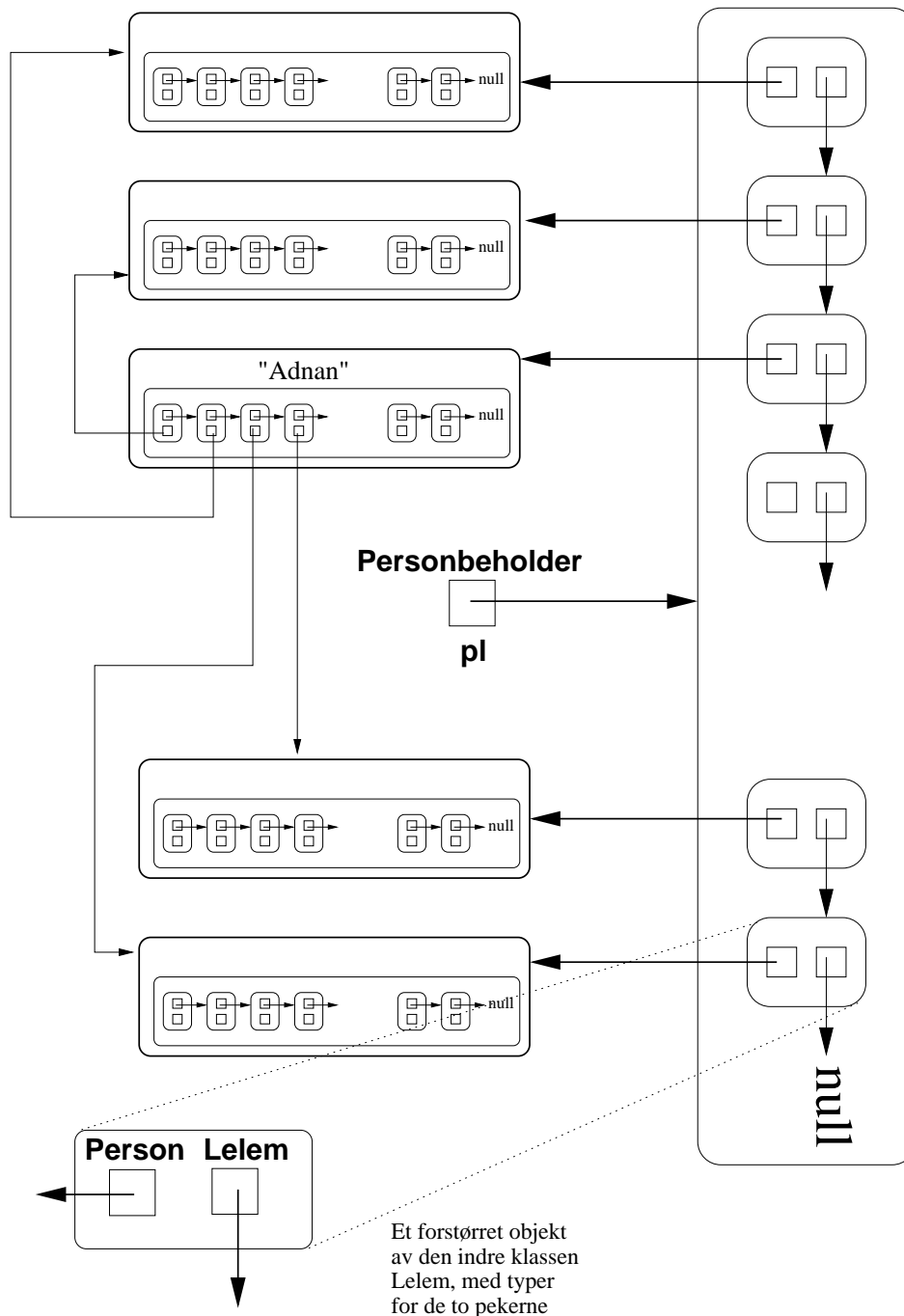
Testklassen skal teste at de fire invariantene bevares ved å forsøke å bryte dem. Programmet skal da gi en feilmelding. Testprogrammet skal opprette minst 7 personer. Minst tre personer skal ha mer enn tre venner. Minst en skal ikke ha venner.

Vennskapsforhold kan være gjensidige og ensidige. Du må gjerne lage flere tilstandspåstander så lenge de ikke bryter med de fire påstandene ovenfor.

Bortsett fra hovedprogrammet (med `main`) og klassen som svarer til liste-elementklassen i beholderen, skal programmet ha minst 3 klassedefinisjoner. En for personbeholderen, en for personklassen, og minst en testklasse.

Anbefalt framgangsmåte: Skriv (og test (kompiler og kjør!)) hver klasse for seg. Skriv først personbeholderen og test metodene med personobjekter uten vennebeholder. Deretter skrives og testes personklassen med vennebeholderen. (Hvis du velger å lage en egen personliste for alle personene, skriver du denne til slutt). Testklassen skrives om for hver «runde».

Nedenfor en grov skisse av datastrukturen:



Tegningen av datastrukturen viser at personlista **pl** her (til høyre) er en personbeholder. Personbeholderen har en lenkeliste av listeelementobjekter. Alle nestepekere i beholderne peker på et annet listeelementobjekt (eller null). I tillegg er det tegnet 5 personobjekter med hver sin personbeholder med venner (pekere til personobjekter). I personobjektet med navn «Adnan» (nr tre ovenfra) er det tegnet inn pekere til 4 av Adnans venner. Bortsett fra **pl** er alle pekere (piler i tegningen) av typen **Person** eller **Lelem**. Merk at mange metoder og variable ikke er tegnet inn for å få fram strukturen av lenkelistene.

Obligatorisk oppgave 3 slutt.

Ønsker du flere utfordringer, kan du legge til metoder i klassene som fjerner personer og vennskap. Du kan også utvide testklassen slik at programmet mottar kommandoer fra bruker (ordreløkke). Eller la beholderklassen være generisk slik at den kan brukes til andre objekter enn personer.