

*Denne oppgaven skal løses to og to vha. systemutviklingsmetoden "Parprogrammering". For å få levere må **alle** registrere seg gjennom et enkelt [skjema](#) innen 21. februar. Selve innleveringen skal bestå av Java-program og utfylt skjema fra parprogrammeringen. Krav og praktisk informasjon ligger på filen som ligger [her](#).*

Det er meningen at du skal jobbe med denne obligatoriske oppgaven i fire uker fra 19. februar til 18. mars. Mye av pensumstoffet du trenger i denne oppgaven er allerede gjennomgått. Det siste (om Iterator og Comparable) blir gjennomgått 26. februar.

Du skal i denne oppgaven ikke bruke beholder-klasser fra Java biblioteket (ikke ArrayList, HashMap, etc.), men skrive alle beholderne selv ved hjelp av tabeller (array) eller lister.

Oppgaven skal besvares sekvensielt, dvs. løs først oppgave 1, så oppgave 2, osv. Likevel er det lurt å lese hele oppgaven før du starter å besvare den. Spesielt er det lurt å se på oppgave 7, der du skal lage et program som bruker alle delene som er beskrevet tidligere.

Du skal i denne oppgavene implementere grensesnittet Iterator minst to steder. Hvordan dette gjøres vil bli nøye gjennomgått på gruppene.

Om du ikke er vant til å lage et ordrestyrt program (se oppgave 7), kan du følge en av ekstragruppene, der dette vil bli gjennomgått.

## 1. KLASSEHIERARKIET

### Legemidler

Et legemiddel har et navn, et unikt nummer og en pris. Når nye legemidler registreres gis de et nytt løpende (unikt) nummer som starter på null.

Et legemiddel er enten av type A, narkotisk, eller av type B, vanedannende, eller av type C, vanlige legemidler.

Legemidler av type A har i tillegg et heltall som sier hvor sterkt narkotisk det er.

Legemidler av type B har i tillegg et heltall som sier hvor vanedannende det er.

Legemidler av type C har ingen nye egenskaper (annet enn klassen).

I tillegg til egenskapene beskrevet over kommer legemidler enten som piller, som liniment (salve) eller som injeksjon. Disse egenskapene skal du beskrive vha. grensesnitt.

For enkelhets skyld skal du for piller bare lagre hvor mange piller det er i en eske. For liniment skal det beskrives hvor mange cm<sup>3</sup> det er i en tube. For injeksjoner skal det beskrives hvor mye virkemiddel det er i en dose (i mg).

## Resepter

En resept har et unikt nummer som starter på null med første resept som opprettes. En resept inneholder en peker til et legemiddel, en peker til den legen som har skrevet ut resepten, og nummeret til den personen som eier resepten (se nedenfor om leger og personer).

En resept har et antall ganger som er igjen på resepten (kalles "reit"). Hvis antall ganger igjen er null, er resepten ugyldig.

Noen resepter er blå, andre er hvite. Blå resepter er sterkt subsidiert, og for enkelhets skyld sier vi her at de er gratis.

## Leger

En lege har et unikt navn. Legene skal kunne sorteres alfabetisk etter navn, og man skal kunne finne en lege basert på navn. Klassen Lege skal derfor implementere grensesnittene Comparable (med seg selv) og Lik.

Grensesnittet Lik inneholder en metode kalt "samme" som har som parameter en String og returnerer sann eller usann. Dette grensesnittet kan f.eks. brukes til å finne om et objekt som inneholder et navn (String) har samme navn som parameteren til metoden.

En lege er enten en spesialist eller så er vedkommende ikke spesialist (vanlig lege).

Noen leger har avtaler med kommunen der de jobber (avtaleleger). For en avtalelege finnes det et avtalenummer. Dette er en egenskap både vanlige leger og spesialister kan ha, og skal beskrives ved hjelp av et grensesnitt.

En lege har en beholder som inneholder alle reseptene han eller hun har skrevet ut. Mer om denne beholderen senere i oppgaven.

## Personer

Personer har et navn og et unikt nummer. Når en ny person registreres gis personen et nytt løpende (unikt) nummer som starter på null med første personen som opprettes. En person har en beholder over alle personens resepter. Mer om denne beholderen senere i oppgaven.

Noen personer er kvinner, andre personer er menn.

## Oppgave 1

Tegn opp klassehierarkiene beskrevet over. Ta også med alle grensesnitt. Du skal ikke ta med data og metoder i klassehierarkiet.

## Oppgave 2

Skriv programmene for alle klassene og grensesnittene beskrevet over.

### DATASTRUKTUR

I denne oppgaven skal det være fire beholdere som tar vare på hhv. legemidler, resepter, leger og personer. I tillegg har en lege en beholder over alle reseptene vedkommende har skrevet ut, og en person har en beholder som inneholder alle personenes resepter. Klassene som beskriver disse beholderne defineres lenger nede i oppgaven. Legemidlene skal lagres i et objekt av klassen Tabell, reseptene skal lagres i et objekt av klassen EnkelReseptListe, legene skal lagres i et objekt av SortertEnkelListe og personene skal lagres i et objekt av klassen Tabell. Beholderen som inneholder en leges resepter skal være av klassen EldsteForstReseptListe, mens beholderen som inneholder en persons resepter skal være av klassen YngsteForstReseptListe.

Oppgave 3. Tegn opp denne datastrukturen (uten å detaljere hvordan de enkelte beholderne er implementert). Tegn noen legemiddel-objekter, noen lege-objekter, noen person-objekter og noen resept-objekter. La det komme klart frem at en resept er med i mange beholdere.

Noen av disse klassene bygger på grensesnitt som først må defineres:

#### Oppgave 4. Grensesnitt for beholdere

Skriv programmet for det generiske grensesnittet AbstraktTabell. Det skal ikke være noen restriksjoner på hva slags elementer den abstrakte tabellen skal kunne inneholde.

AbstraktTabell beskriver en beholder og du skal kunne:

- sette et objekt inn i tabellen på en oppgitt plass (indeks). Metoden returnerer sann eller usann avhengig om operasjonen gikk bra eller ikke.
- finn et objekt basert på en indeks.
- returnere en Iterator over listen.

Skriv programmet for det generiske grensesnittet AbstraktSortertEnkelListe. En slik liste skal bare kunne inneholde elementer som implementerer grensesnittene Comparable (med seg selv) og Lik. En slik liste skal kunne:

- sette inn et nytt element (i sortert rekkefølge, minste først).
- finne et element basert på en nøkkel av typen String
- returnere en Iterator over listen, slik at innholdet kan bli listet opp i sortert rekkefølge, minste først.

#### Oppgave 5. Klasser for beholdere

##### a) Generiske klasser

Skriv den generiske klassen Tabell som implementerer AbstraktTabell.

Klassen skal lagre alle elementene i en array, og arrayens lengde skal oppgis som parameter til konstruktøren. På gruppene vil du få hjelp til å lage iteratoren over listen.

Hvis du har lyst og tid: Når du setter noe inn i Tabellen og det ikke er plass, skal du lage en ny array som er lang nok (innenfor rimelighetens grenser), og så kopiere alle elementene over til den nye arrayen.

Skriv den generiske klassen `SortertEnkelListe` som implementerer `AbstraktSortertEnkelListe` som en enveisliste.

b) Ikke generiske klasser

Skriv klassen `EnkelReseptListe`. Klassen `EnkelReseptListe` skal inneholde en envisliste med en peker til første og en peker til siste element i listen. Klassen skal kunne ta vare på resepter, og en resept må kunne være med i flere objekter av denne klassen. Metodene i klassen skal kunne sette inn en resept og finne en resept basert på reseptnummeret. Hvis resepten som det letes etter ikke finnes i listen, skal det kastes et unntak. Skriv også en iterator over listen.

Skriv subclassene `EldsteForstReseptListe` og `YngsteForstReseptListe`. Når du itererer over den første klassen skal du starte med den eldste resepten (den som ble satt inn først) og gå mot mot yngre (de som ble satt inn sist). Når du itererer i den andre klassen, skal du starte med i den yngste enden.

Hint 1: Forskjellen på de to subclassene til klassen `EnkelReseptListe` er bare metoden som setter inn en resept.

Hint 2: I både `SortertEnkelListe` og `EnkelReseptListe` skal du skrive en iterator. Kan du klare å bruke den samme iteratoren i begge klassene?

## FULLSTENDIGE PROGRAMMER

Oppgave 6. Lag enhetstester for alle beholderene.

Svaret på denne oppgaven skal ikke leveres inn, men du bør gjøre den for å overbevise deg om at klassene er riktig programmert.

Skriv et fullstendig testprogram for hver av klassene `Tabell`, `SortertEnkelListe`, `EldsteForstReseptListe` og `YngsteForstReseptListe`. Prøv å lage programmene slik at både vanlige tilfeller og de fleste spesialtilfellene blir testet.

Oppgave 7. Skriv et ordrestyrt program for leger og resepter.

Nå må du se nøye på datastrukturen du tegnet i oppgave 3.

Det ordrestyrte programmet skal kunne:

Opprette og legge inne et nytt legemiddel.  
Opprette og legge inn en ny lege.  
Opprette og legge inn en ny person.  
Opprette og legge inn en ny resept.

Hente legemiddelet på en resept basert på nummeret til personen som skal ha resepten og reseptens nummer. Siden vi i denne oppgaven ikke har noe data om mengden av legemiddel på lager, betyr dette at vi bare teller ned antallet ganger resepten kan brukes (reit). Om antallet blir null, betyr dette at resepten er ugyldig. Prisen som skal betales skrives ut. Skriv også ut legens navn, personens navn og all dataene du har om legemiddelet på resepten (inkludert antall piller i en eske, hvor stort volum en tube har eller hvor mye virkemiddel det er i en dose).

Lese hele datastrukturen fra en fil som har et gitt filformat. Dette filformatet blir beskrevet i et eget dokument som blir tilgjengelig senest 24. februar [her](#).

Programmet skal kunne håndtere mange former for oppslag og statistikk.  
Bruk for-each-løkker for å gå gjennom beholderne.  
I programmet du skriver skal du gjøre følgende:

Skriv ut de data som er tilgjengelig om alle legemidler, leger og personer. Legene skal skrives ut i sortert rekkefølge (på navn). Data om resepter skal ikke skrives ut.

-For en gitt person (med et gitt personnummer):

Gå gjennom alle reseptene og skriv ut hvor mange gyldige blå resepter denne personen har og for disse gyldige blå reseptene hvor mange injeksjonsdoser det er igjen.

- Gå gjennom alle legene og for hver eneste lege med avtale skriv ut legens navn og gå gjennom alle legens resepter og skriv ut hvor mange resepter denne legen har skrevet ut på narkotiske legemidler.

- Gå gjennom alle personene og for hver eneste person skriv ut personens navn og gå gjennom alle personens resepter og skriv ut hvor mange gyldige resepter som er skrevet ut på vanedannende legemidler. Skriv helt til slutt ut hvor mange slike gyldige resepter det er totalt og hvor mange av disse er til kvinner og hvor mange er til menn.

Hvis du har tid og lyst (å skrive ut statistikk blir mye morsommere hvis det er mye data): Skriv hele datastrukturen til fil på det samme formatet som du brukte da du leste inn fra fil. Rekkefølgen på reseptene behøver ikke opprettholdes ved skrivning/lesing til/fra fil.

Slutt obligatorisk oppgave 4.

