

## **Om obligatorisk oppgave 5, 6 og 7 i INF1010, våren 2015: "Leger og resepter"**

### **Versjon 1,0**

#### **Innledning.**

Du skal jobbe med en problemstilling omkring leger og resepter i 4 uker fra 19. februar til 25. mars og du kan til sammen få maksimum 8 poeng. Den første delen av oppgaven (oblig 5), som du skal jobbe med de første to ukene (maks 2 poeng), skal leveres senest 4. mars kl 10:00. Den 26. februar frigis den andre del av oppgaven (oblig 6). Denne skal leveres senest 18. mars kl 10:00 og gir maks 3 poeng. Den siste delen (oblig 7) frigjøres 5. mars. Den skal leveres senest 25. mars kl 10:00, og har også maks 3 poeng.

Siden disse tre oppgavene bygger på hverandre er det bare mulig å hoppe av underveis. Det er ikke mulig å hoppe på, dvs. at du må ha godkjent 5 for å få godkjent 6, og du må ha godkjent 6 for å få godkjent 7. Du har godkjent en oppgave om du får ett eller flere poeng.

Obligene 6 og 7 skal løses vha. systemutviklingsmetoden "Parprogrammering". Du MÅ melde deg på oblig 6 og 7 innen 25. februar. Informasjon om dette finner du under "oblig 6 og 7" på kursets hjemmeside.

I hovedsak består oblig 5 av å lage et klassehierarki, oblig 6 av å lage en del beholdere, og oblig 7 av å sette det hele sammen til slutt til et ordrestyrt program. I oblig 6 og oblig 7 må du kanskje gå tilbake til klassehierarkiet i oblig 5 og gjøre små forandringer.

**Obligatorisk oppgave 5, INF1010, våren 2015, maks 2 poeng.**  
**Innlevering senest onsdag 4. mars kl. 10:00**

#### **KLASSEHIERARKIENE til Leger og Resepter m.m.**

Hensikten med denne oppgaven er at du skal lære deg å skrive klassehierarkier med subklasser og grensesnitt (interfaces). Det er ingen generiske klasser eller generiske interfacer i oblig 5. Ikke gjør oppgaven for stor eller for vanskelig. Som beskrevet over må du kanskje utvide noen av klassene du skriver i oblig 5 når du ser hva obligene 6 og 7 trenger. Du skal derfor foreløpig bare ta med de variable og metodene som det eksplisitt blir bedt om her. Hvis du synes noe er uklart, gjør dine egne fornuftige presiseringer.

#### **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 0.

Et legemiddel er enten av type A, narkotisk, eller av type B, vanedannende, eller av type C, vanlige legemiddel. For alle legemidler må vi kunne vite hvor mye virkestoff (mg) det inneholder totalt.

Det er stor forskjell på legemidler av disse tre typene, men i denne oppgaven skal vi bare ta hensyn til:

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

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

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

I tillegg til egenskapene beskrevet over kommer legemidler enten som piller eller som mikstur (flytende). Disse egenskapene skal du beskrive vha. grensesnitt (interface).

For enkelhets skyld skal vi for piller bare vite hvor mange piller det er i en eske og hvor mye virkestoff det er i hver pille. For mikstur skal vi vite hvor mange cm<sup>3</sup> det er i en flaske og hvor mye virkestoff det er i hver cm<sup>3</sup>.

## **Resepter**

En resept har et unikt nummer som starter på 0 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. Vi skal ta høyde for at det er stor forskjell på hvite og blå resepter, men igjen skal vi gjøre en forenkling og si at bare prisen er forskjellig: Blå resepter er sterkt subsidiert, og for enkelhets skyld sier vi her at de er gratis.

## **Leger**

En lege har et unikt navn.

I oblig 6 skal vi kunne finne en lege basert på navn. Klassen Lege skal derfor implementere grensesnittet (interfacet) Lik der sammenlikningen (likheten) skal skje med legens navn.

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.

Noen leger har avtaler med kommunen der de jobber (fastleger). Dette å ha en avtale med kommunen er noe ikke bare leger kan ha. Denne egenskapen skal derfor beskrives med et grensesnitt (interface). For alle som har en avtale med kommunen skal vi kunne få tak i et avtalenummer.

## **Personer**

Personer har et navn og et fødselsnummer (11 siffer) og en adresse. Adressen skal være et veinavn og nummer (som til sammen lagres i en String) og et firesifret postnummer. Når en ny person registreres gis personen i tillegg et nytt løpende (unikt) nummer som starter på 0 med første personen som programmet oppretter.

### **Oppgave 5.1**

Tegn opp klassehierarkiene beskrevet over. Ta også med alle grensesnitt (interface). I denne tegningen skal du ikke ta med data og metoder.

### **Oppgave 5.2**

Skriv programmene for alle klassene og grensesnittene (interface) beskrevet over. Skriv konstruktører som initialiserer alle variable og konstanter i alle klasser.

Du skal lever inn tegningen fra oppgave 5.1 og programmene (kildekoden) til alle klassene og grensesnittene (interface) fra oppgave 5.2. Du skal følgelig ikke levere noe fullstendig, kjørbart program i denne oppgaven.

Slutt obligatorisk oppgave 5.