

IN1010 V17, Obligatorisk oppgave 2

Innleveringsfrist: Tirsdag 27.02 kl 10.00

Versjon 1.1. Sist endret av Silje Merethe Dahl (siljemda)

Introduksjon

I de obligatoriske oppgavene fremover skal du lage et system som holder styr på leger, pasienter, resepter og legemidler. I denne første oppgaven skal du opprette objekter og skrive klasser for legemidler, resepter og leger og gjøre relevante tester av disse.

Det forventes at du skriver ryddig og lesbar kode, og det anbefales at du skriver fornuftige kommentarer underveis slik at du selv har oversikt over hvordan de forskjellige delene av programmet fungerer.

Notat om ID

Noen av objektene i denne obligen skal kunne identifiseres ved en ID. Med ID menes her et unikt, ikke-negativt heltall.

Det første objektet som opprettes av en bestemt klasse skal ha id = 0, det andre id = 1, det tredje id = 2, osv. Merk at ID-ene ikke er unike på tvers av alle klassene, men hvert objekt av en bestemt type skal ha en id som ingen andre objekter av den samme klassen har. Det vil si at ingen legemidler har samme ID, og ingen resepter har samme ID, men en resept og et legemiddel kan ha samme ID.

Del A: Legemiddel

Et legemiddel har et navn, en ID og en pris. I tillegg må vi for alle legemidler kunne vite hvor mye virkestoff (mg) det inneholder totalt. Prisen og virkestoffet skal lagres som flyttall.

Et legemiddel **skal** være en av subtypene A, narkotisk, eller B, vanedannende, eller C, vanlig legemiddel. Det er stor forskjell på legemidler av disse tre typene, men i denne oppgaven skal vi bare ta hensyn til følgende krav for de forskjellige typene legemidler:

- *Type A* har et heltall som sier hvor sterkt narkotisk det er.
- *Type B* har et heltall som sier hvor vanedannende det er.
- *Type C* har ingen tilleggsegenskaper (annet enn klassens navn).

A1: Tegn klassehierarkiet beskrevet ovenfor. Du trenger bare å ha med navn på klasser og sammenhengen mellom disse. Lever tegningen som en bildefil eller som PDF.

A2: Skriv klassene *Legemiddel*, *LegemiddelA*, *LegemiddelB* og *LegemiddelC*. De tre sistnevnte klassene arver den førstnevnte. Konstruktøren til *Legemiddel* (og dermed også *LegemiddelC*) skal ta inn String navn, double pris og double virkestoff (i den rekkefølgen). Konstruktørene til *LegemiddelA* og *LegemiddelB* skal i tillegg ta inn int styrke.

Legemiddel skal ha metodene *hentId*, *hentNavn*, *hentPris* og *hentVirkestoff* som returnerer de relevante verdiene. I tillegg skal klassen ha metoden *settNyPris*.

LegemiddelA har i tillegg metoden *hentNarkotiskStyrke*, mens *LegemiddelB* har metoden *hentVanedannendeStyrke*.
bagimlilik siddeti)

A3: Skriv et testprogram *TestLegemiddel*. I denne klassen skal du opprette ett objekt av hver av subklassene du har skrevet. Deretter skal du gjøre enkle enhetstester der du tester alle egenskapene til en instans før du går videre og gjør det samme for neste instans.

Tips 1: For hver test bør du beskrive en forventning til resultatet av testen, slik at du gir utskrift til brukeren avhengig av om du fikk forventet resultat eller ikke.

Tips 2: Det kan være lurt å skille ut testene i statiske metoder inne i *TestLegemiddel*-klassen.

Relevante Trix-oppgaver: [3.02](#), [3.03](#), [3.04](#) & [4.10](#)

Del B: Resepter

En Resept har en ID. I tillegg skal en resept ha en referanse til et legemiddel, en referanse til den legen som har skrevet ut resepten, og ID-en til den pasienten som eier resepten. En resept har et antall ganger som er igjen på resepten (kalles reit). Hvis antall ganger igjen er 0, er resepten ugyldig.

I denne oppgaven skal vi forholde oss til ulike typer resepter. De to hovedkategoriene er **hvite** og **blå** resepter.

Hvite resepter

Hvite resepter har i seg selv ingen nye egenskaper (utover et annet klassenavn). Derimot finnes det to subtyper av hvit resept: Militærresepter og P-resepter.

- **Militærresepter** utgis til vernepliktige i tjeneste. Som en forenkling sier vi at militærresepter alltid gir en 100% rabatt på prisen til et legemiddel.
- **P-resepter** gir unge en rabatt på prevensjonsmidler. Denne rabatten er statisk og gjør at brukeren betaler 116 kroner mindre for legemiddelet. Merk at brukeren aldri kan betale mindre enn 0 kroner. I tillegg til rabatten har P-resepter den egenskapen at de alltid utskrives med 3 reit.

Blå resepter

Det er stor forskjell på vanlige (hvite) og blå resepter (blant annet er utstedelsen av en blå resept forbundet med en del kontroller), men igjen skal vi gjøre en forenkling og si at bare

prisen som betales er forskjellig: Blå resepter er alltid sterkt subsidiert, og for enkelhets skyld sier vi her at de har 75% rabatt slik at pasienten må betale 25% av prisen på legemidlet.

B1: Tegn klassehierarkiet beskrevet ovenfor. Du trenger bare å ha med navn på klasser og sammenhengen mellom disse. Lever tegningen som en bildefil eller som PDF.

B2: Skriv klassen *Resept* og dens subklasser. Konstruktøren i *Resept* skal ta inn et *Legemiddel* legemiddel, en *Lege* utskrivende *Lege*, en *int* *pasientId* og *int* *reit* (i den rekkefølgen). **Merk: Vi skal ikke kunne opprette en instans av selve klassen *Resept***, kun av subtypene.

Klassen *Resept* skal ha følgende metoder som henter relevant data: *hentId*, *hentLegemiddel* (henter tilhørende *Legemiddel*), *hentLege* (henter utskrivende *Lege*), *hentPasientId* og *hentReit*.

I tillegg skal klassen ha følgende metoder:

- *public boolean bruk*: Forsøker å bruke resepten én gang. Returner *false* om resepten alt er oppbrukt, ellers returnerer den *true*.
- *abstract public String farge*: Returnerer reseptens farge. Enten "hvit" eller "blaa".
- *abstract public double prisAaBetale*: Returnerer prisen pasienten må betale.

B3: Skriv et program *TestResepter* der du oppretter instanser av de forskjellige klassene. Du vil også trenge å opprette noen objekter av klassen *Legemiddel* for å gjøre dette.

Tips: Du vil også ha behov for en instans av klassen *Lege*. Selv om du ikke har skrevet denne klassen enda kan du opprette en forenklet (tom) versjon av *Lege* foreløpig.

På samme måte som i Del A skal du nå gjøre enhetstester av instanser av de forskjellige klassene. For hver instans holder det at du tester egenskapene som er implementert forskjellig fra reseptens foreldreklasse.

Relevante Trix-oppgaver: [4.01](#), [4.02](#), [4.03](#) & [4.11](#).

Del C: Leger

Merk: Relevante temaer for å løse del C undervises først tirsdag 13. februar.

Konstruktøren i *Lege* tar kun inn en *String* med legens navn. *Lege* skal ha en metode for å hente ut et navn.

Enkelte leger har en avtale med kommunen de jobber, og disse legene kalles for fastleger. Å ha en kommuneavtale er noe som kan gjelde andre enn leger (men i denne oppgaven blir det bare leger). Denne egenskapen skal derfor beskrives med et grensesnitt (interface). For alle som har en avtale med kommunen, skal det være mulig å hente ut et avtalenummer.

Fastlege skal arve fra Lege og skal i tillegg implementere følgende grensesnitt (dette grensesnittet må du legge ved som en fil i innleveringen din):

```
interface Kommuneavtale {  
    public int hentAvtalenummer();  
}
```

C1: Tegn klassehierarkiet beskrevet ovenfor (inkludert grensesnittet). Du trenger ikke å ta med metoder eller annet innhold i klassene/grensesnittet. Lever tegningen som en bildefil eller som PDF.

C2: Skriv klassene Lege og Fastlege som beskrevet over. Konstruktøren i Fastlege skal i tillegg til navn også ta imot en int avtalenummer.

Relevante Trix-oppgaver: [5.03 & 5.04](#).

Del D: Integrasjonstest

I denne delen skal du lage et hovedprogram som gjør det vi vil kalle en minimal integrasjonstest - vi skal altså teste hvordan de forskjellige delene av systemet fungerer sammen. Hovedprogrammet skal minimum gjøre følgende:

- Opprette minimum en instans av hver eneste klasse og la disse inneholde nødvendige referanser til andre objekter.
- Skrive ut all tilgjengelig informasjon om hvert enkelt objekt.

I tillegg skal du lage en datastrukturtegning som illustrerer objektene og deres tilstand når du har kjørt dette testprogrammet. Ved behov, se [notatet om datastrukturer](#) eller repeter [Trix-oppgaver fra obliq 1](#).

Oppsummering

Du skal levere følgende:

- Klassehierarki-tegninger som beskrevet i deloppgavene (Legemidler, Resepter, Leger).
- Klasser og grensesnitt for legemidler, resepter og leger samt enhetstestene av disse.
- Testprogram og datastrukturtegning fra Del D.

Alle delene av programmet må kompilere og kjøre for å kunne få oppgaven godkjent. Ikke levér zip-filer! Det går an å laste opp flere filer samtidig i Devilry.