

# Seminaroppgaver uke 5

## 1.A

Lag et interface «DrikkBar», den skal ha to metoder: `drikkGlass()`; og `fillOpp()`;

## 1.B

Lag deretter en klasse «Drikke» som implementerer grensesnittet DrikkBar. Drikke må holde på hvor mye det er plass (dl) til i flasken og hvor mye drikke det er i flasken nå. Max-størrelsen på flasken settes i konstruktøren og vi kan anta at flasken er full når den opprettes.

Dersom metoden «drikkGlass» kalles skal flasken reduseres med 2 dl. Dersom det er mindre enn 2 dl igjen, skal flasken tømmes (til 0). «fillOpp» skal fylle flasken opp max størrelse.

## 1.C

Utvid så programmet ditt med subclassen øl (Ol). Den skal være lik drikke, men i stede for at flasken reduseres med 2 dl, skal den reduseres med en halv liter når et glass drikkes.

Lag også en `toString()`-metode for både drikke og øl.

*Hint: husk @Override der du mener det er riktig.*

## 1.D

Tegn klassehierarkiet for oppgaven.

## 1.E

Implementer en liten test av programmene dine.

## 2.A

Gitt følgende klasse- og grensesnitt-definisjoner, tegn klassehierarkiet:

interface Z

interface Y extends Z

class A extends B implements Y

class B implements Z

class C extends A

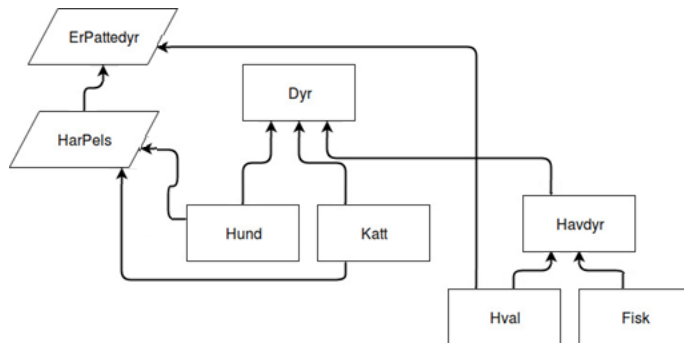
## 2.B

Diskusjon:

Hvilket, om noe, grensesnitt implementeres i C?

## 3.A

Gitt følgende klassehierarki, skriv opp klasse- og grensesnitt-definisjonene (som 2A).



## 4.A

Vi skal lage en enkel modell av en næringskjede. På toppen av næringskjeden finnes rovdyrene og på bunnen finnes planteeterne. Skriv to grensesnitt, et for rovdyr og et for planteetere. Rovdyrene skal kunne jakte, mens planteeterne skal kunne beskytte seg selv.

## 4.B

Skriv klassene Bjorn, Ulv, Elg og Sau som implementerer de riktige grensesnittene over. Opprett objekter av klassene og legg alle rovdyrene i en array av rovdyr, mens du legger alle planteeterne i en array av planteetere. Test programmet ditt.

## 4.C

Tegn opp klassehierarkiet med alle klasser og grensesnitt.