Uke 5

Interface

- Kan kun arve fra 1 en klasse, mens interface kan man arve så mange man vil
  - implements
- Peker kan være av interface type
  - Beholder kan holde interface type som peker
- ArrayList og HashMap implementerer interface List og Map

Oppgave 1

a) Lag et interface, Drikkbar. Interfacet skal ha to metoder: drikkGlass(); og fyllOpp();

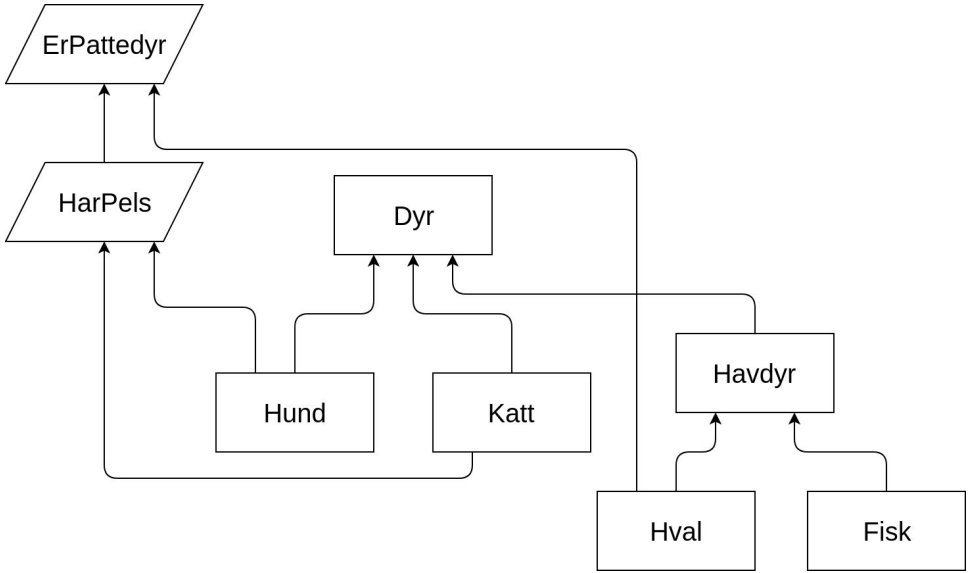
b) Lag deretter en klasse «Drikke» som implementerer grensesnittet Drikkbar. Drikke må holde på hvor mye drikke det er plass til i flasken (dl) og hvor mye drikke det er i flasken nå (dl). Maksstørrelsen på flasken settes i konstruktøren og vi kan anta at flasken er full når den opprettes.

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

d) Utvid programmet ditt med subklassen Ol (øl). Subklassen skal være lik drikke, men i stedet 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.*

e) Implementer en liten test av programmene dine.

Eksempel på struktur:



Oppgave 2

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

b) Skriv klassene Bjorn, Ulv, Elg og Sau som implementerer de riktige grensesnittene over. Metodene som må implementeres trenger kun å printe ut en beskjed om hva dyret gjør. 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. *Diskusjon: hva gjør man med Bjørn; er den et rovdyr eller en planteeter, eller kan vi gjøre den til begge deler?*

c) (Ekstra) Tegn opp klassehierarkiet med alle klasser og grensesnitt.

Menti