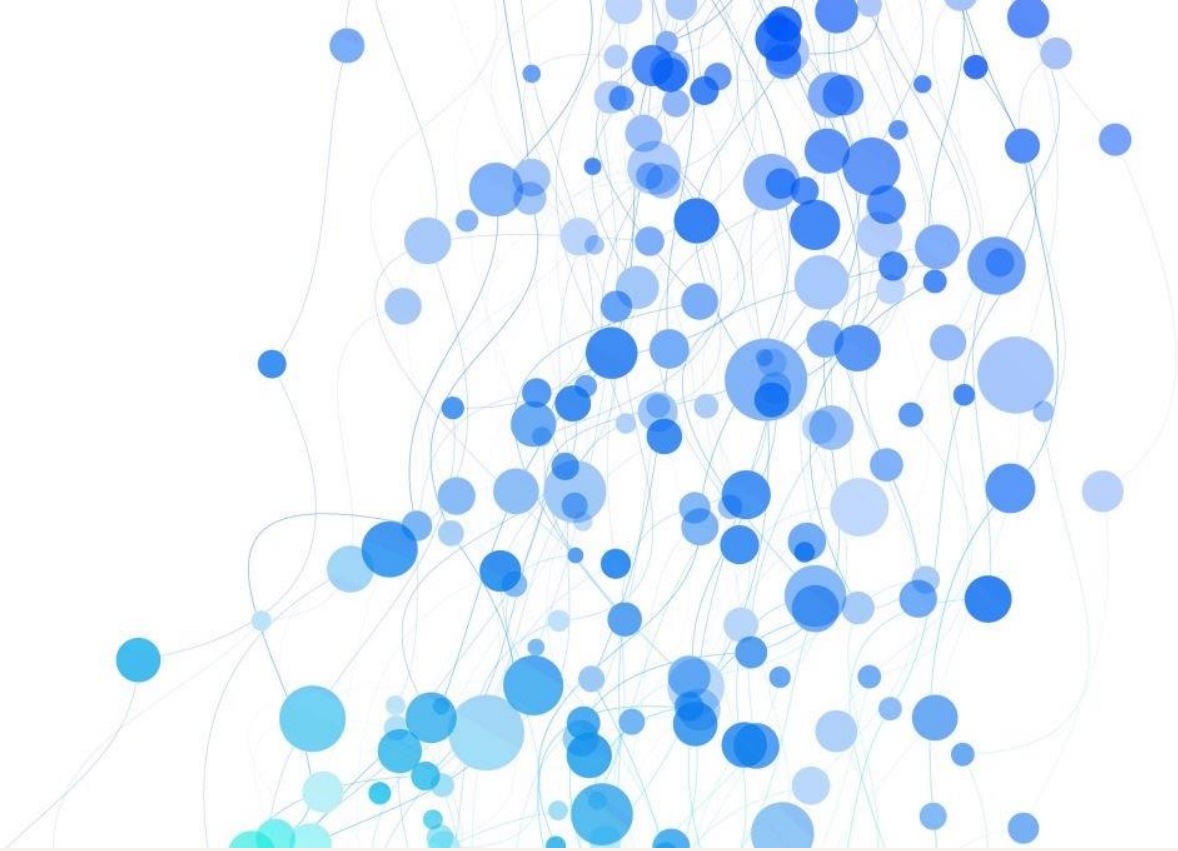


IN1000 Uke 7

GRUPPE 16

Agenda

Repetisjon av Objektorientering og klasser

Felles oppgaver

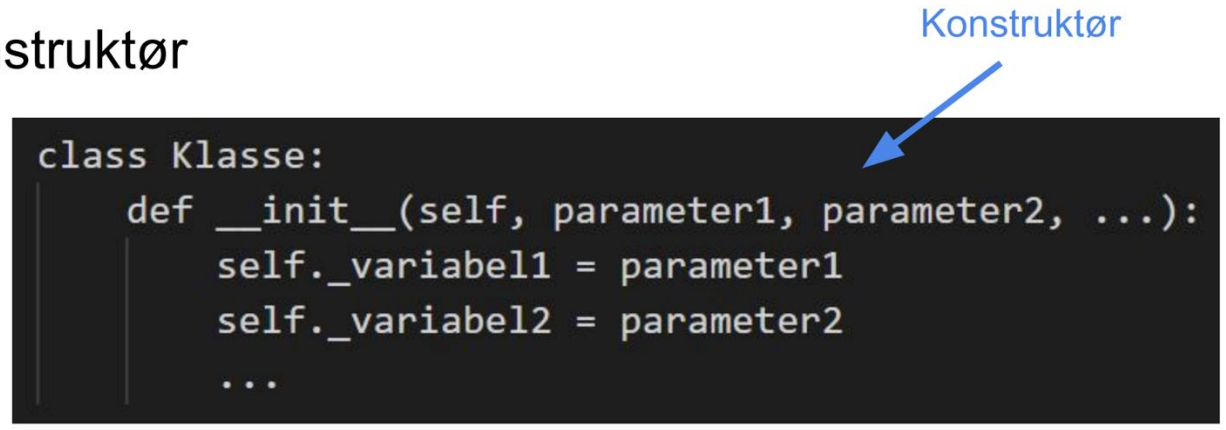
Quiz på slutten

Objekter og klasser – hva er det?

- **Objektorientert programmering** handler om å "modellere" verden
- **Objekter** representerer ting, og har
  - egenskaper (*instansvariabler*)
  - oppførsel (*metoder*)
- En **klasse** beskriver et sett med objekter med lik oppførsel (mal for objektet)
- Objekter er **instanser** av klasser

Instansvariabler og konstruktør

Konstruktør



Konstruktør - bestemmer hva som skal være med ved opprettelsen av et objekt. Parametre til konstruktøren angir hvilke *instansvariabler* vi kan velge verdi for når vi oppretter et nytt objekt.

- Instansvariabler er egenskapene til en klasse, **dvs. variabler som hører til** klassen
- **De vil få egne verdier for** hver instans av klassen (hvert objekt)
- Husk dobbel understrek! __init__(self):

Metoder

```
class Klasse:
    ...

    def metode_navn(self, parameter1, ...):
        # Kode inni metoden
```

- I tillegg til egenskaper (*instansvariabler*), vil vi at klassen også skal tilby tjenester (via *metode*r)
- Metoder er som prosedyrer/funksjoner, bare at de tilhører en klasse (og har derfor tilgang ti alle instansvariablene til klassen)

Grensesnitt og innkapsling

- Public interface (offentlig grensesnitt) – alle metodene man kan bruke
  - Grensesnittet til en klasse er alt som kan gjøres utenfor skopet til klassen
  - Altså alt man har tilgang til "utenifra"
- Innkapsling – instansvariabler og metoder som ikke skal være i grensesnittet
  - Disse skal altså ikke aksesseres utenifra
  - Får navn som begynner med _ for å indikere at de er private

Laste inn klasser

- I IN1000 vil vi at alle klasser skal skrives i egen fil (gjelder også for Java, som dere har neste semester)
- Vi vil altså ha en fil til hver av klassene våre og en fil til hovedprogrammet
- Det gjør at vi må laste inn klassen til hovedfilen (hovedprogrammet)
- Merk: Lettest hvis man har alle filene I samme mappe!

from <filnavn> import <Klasse>

Oppsummering

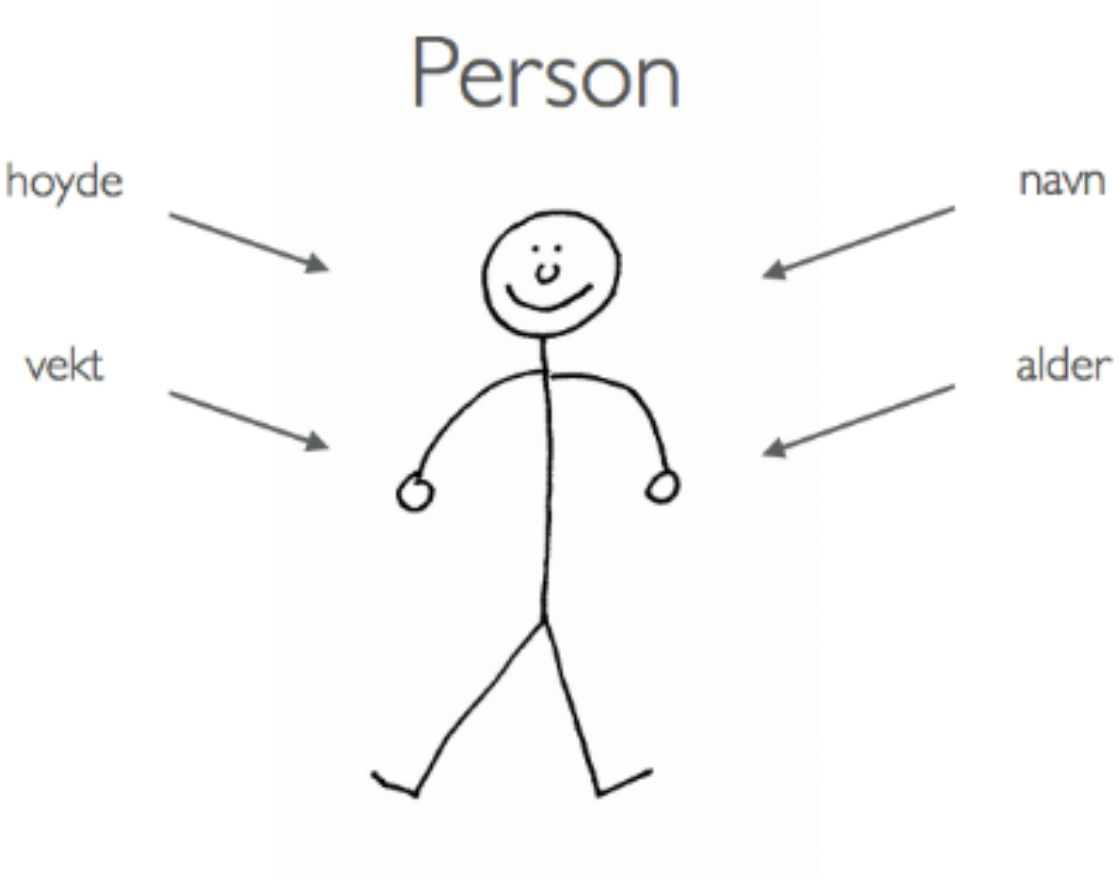
- Klasser defineres med class, starter på stor bokstav, eks: class Person
- Klassens konstruktør skal ha dobbel understrek: __init__(self):
- Klassens parametre knyttes til et objekt gjennom self: self.navn = navn
- Klassens metoder gir oss tilgang til objektets verdier utenfor klassen:

def hent_navn(self): return self.navn

- Klassens variabler som ikke skal aksesseres utenfra (direkte) angis med Klassens variab understrek: _navn

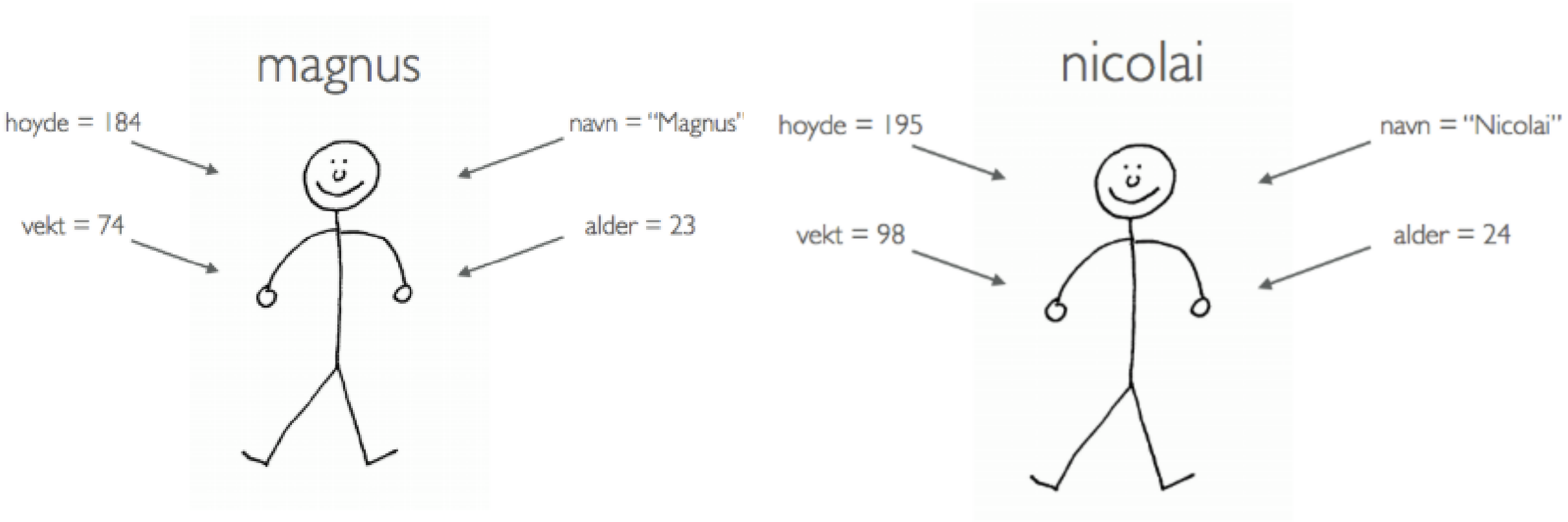
Oppgave 1: Lag en klasse Person

En person skal registreres med
navn, høyde, vekt, alder
Prøv selv, så går vi gjennom felles etterpå



Oppgave 2: Opprett 2 objekter av Person

Eks:



Oppgave 3: Utvide klassen

Vi skal nå utvide Person-klassen vår med følgende metoder:
endre_navn(self, nytt_navn) → endre navnet til nytt navn.

skriv_ut_hilsen(self): → skriv ut en hilsen fra personen, på formen "Hei, jeg heter <navn> og jeg er <alder> år gammel".

feire_bursdag(self) →
Øk alderen til personen med 1.
Skriv ut en bursdagshilsen med den nye alderen.
Hvis personen fremdeles er under 18 år, skal også høyden økes med 2.