IN1000 Repetisjonskurs

OO-design, klasser, objekter, referanser

Læringsmål - Objektorientert programmering

- Kjenne til motivasjon og bakgrunn for objektorientert programmering
- **Programmering: Kunne definere en klasse, opprette objekter og kalle metoder på objektene**
- Kunne bruke og forklare begrepene klasse, objekt, klassedefinisjon, konstruktør, metode og instansvariabel
- Forstå sentrale begreper som grensesnitt og innkapsling
- Kjenne til utviklingsprosessen for en klasse gjennom design, implementasjon og testing

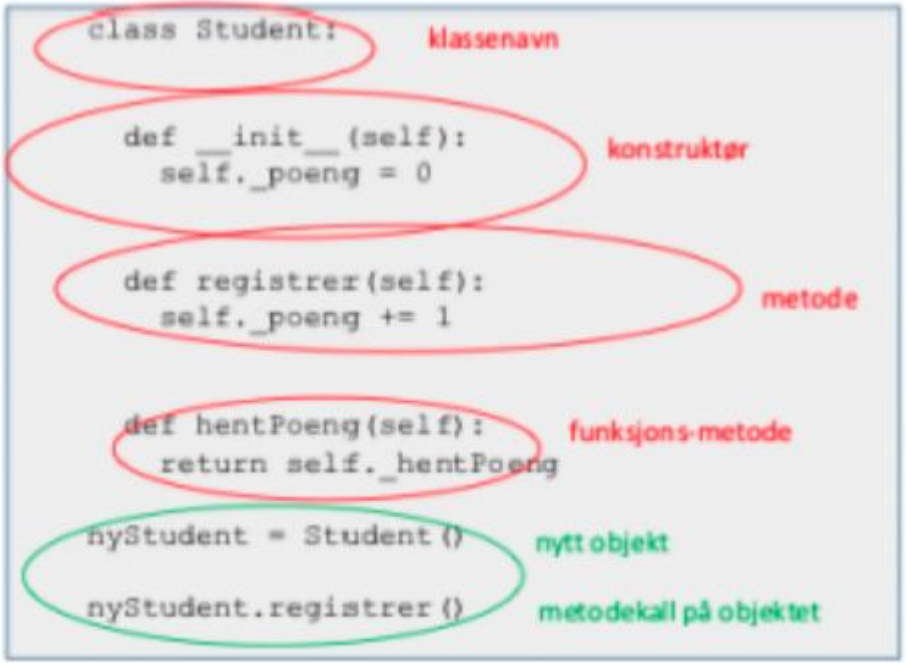
Hvorfor objektorientert programmering?

- Modellere virkeligheten
- Strukturere data og programmer
- Dele opp større programmer i mindre deloppgaver

Noen begreper

- **Objekt:** presenterer en verdi, instanser av en bestemt klasse
- **Klasse**: et mønster/oppskrift for objekter av samme type
- **Klassedefinisjon**: beskriver hvilke data hvert objekt skal ha og de metodene som skal er tilgjengelig for et objekt av klassen
- **Instansvariabel**: variablene til et ovjekt av en klasse
- **Metode**: prosedyre/funksjon som tilhører en klasse - brukes på objekter
- **Grensesnitt**: Et sett med tjenester i form av metoder (lese av / endre et objekt)
- **Innkapsling**: Hvordan et objekt viser og manipulerer sine data er skjult - utenfor en klasse vet vi hva objektet kan gjøre - men hvordan er skjult utenfor klassedefinisjonen
  → for å beskytte data mot uønskede endringer

Klasse



Alt det "røde" er **grensesnittet** til klassen Student

Eksempel 1

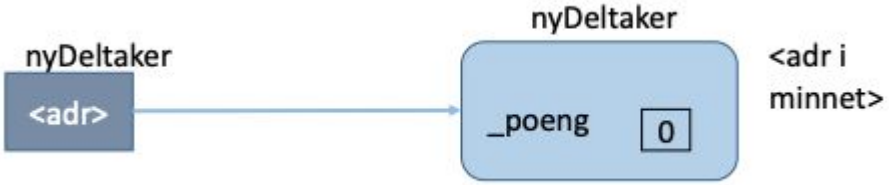
- **Student.py**
- **Hovedprogram.py**

Læringsmål - referanser

- Forstå (mer av) hva som skjer bak kulissene når vi oppretter og bruker objekter
- Kjenne til forskjellen på å endre en referansevariabel og å endre objektet den refererer til
- Kunne skrive programmer med samlinger av (referanser til) objekter
- Kunne sette seg inn i enkle programmer med flere klasser og objekter som refererer til andre objekter

Referanser

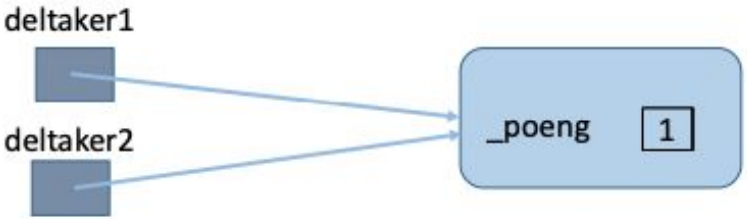
- En variabel kan sees på som en “boks” som inneholder en verdi
- Objekter lagres ikke direkte i variabler - variablene inneholder isteden bare *referansen* til objektet



- Variabler som holder rede på objekter kalles *referansevariabler*

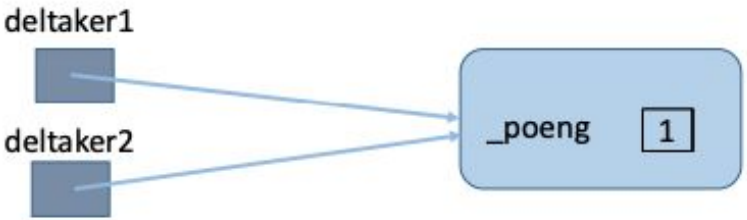
Referanser til objekter

```
deltaker1 = Deltaker()
deltaker2 = deltaker1
deltaker2.registrer()
print (deltaker1.hentPoeng())
```



Referanser til objekter

```
deltaker1 = Deltaker()
deltaker2 = deltaker1
deltaker2.registrer()
print (deltaker1.hentPoeng())
```



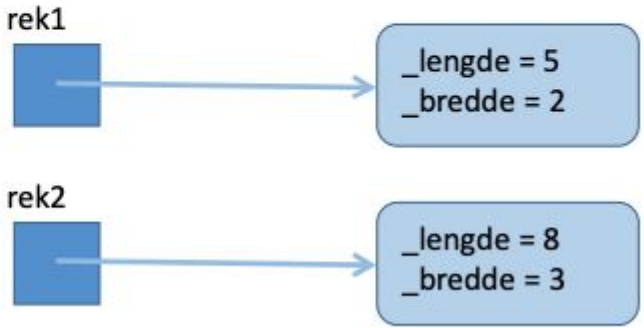
- *deltaker1* og *deltaker2* **refererer** til samme objekt
- Begge referansevariablene kan endre og skrive ut verdier fra samme objekt

Opprette objekter

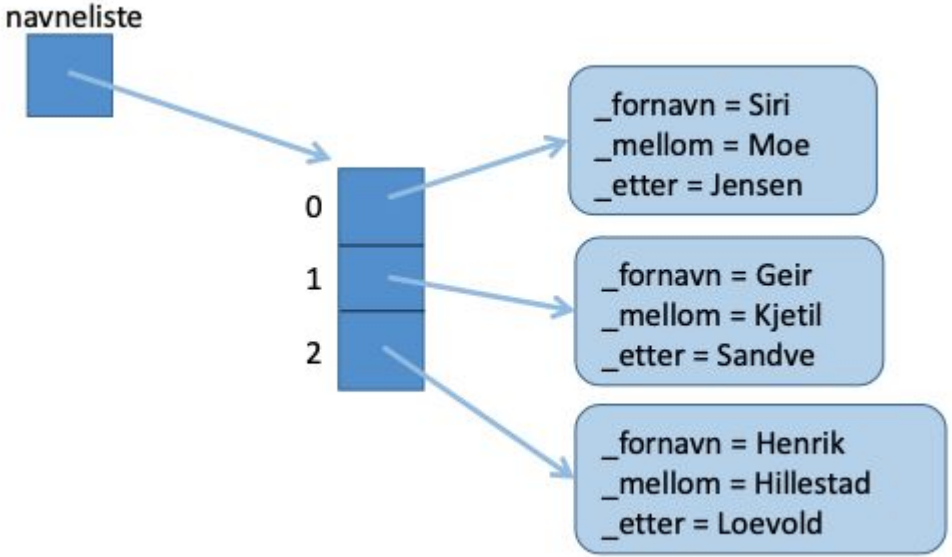
```
class Rektangel :
  def __init__(self, lengde, bredde):
    self._lengde = lengde
    self._bredde = bredde
```

```
rek1 = Rektangel(5,2)

rek2 = Rektangel (8,3)
```



Flere objekter av samme klasse



Eksempel 2

- Utgangspunkt i eksempel 1
- **emne.py**
- Utvide **hovedprogram.py**

Trix-oppgaver

OO, klasser, objekter

- Tag “klasse” - eller “uke7” og utover

Referanser

- Tag “referanse” - eller “uke 8” og utover

Større oppgaver / utfordringer

- Tag “klasse” + “utfordring”