



Uke 8 - Forståelse av objekter

Gruppe9 IN1000
jonasbny@ifi.uio.no



Agenda



en referanse til objektet i seg selv: *self*

referansevariabler og objekter

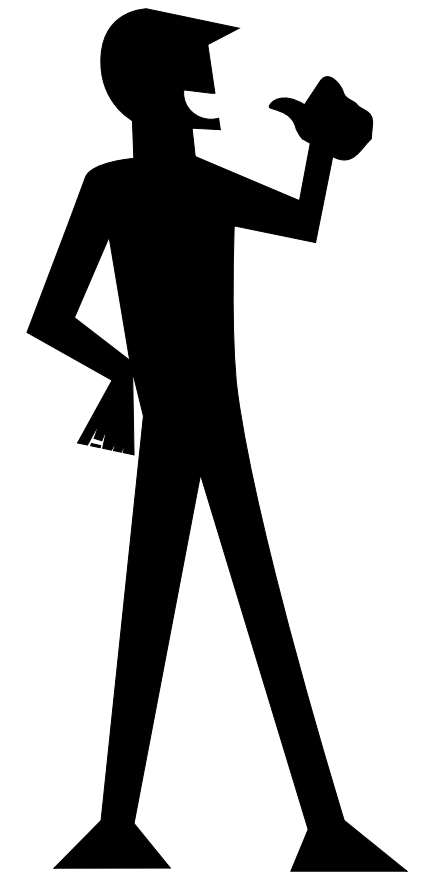
objekter som refererer til andre objekter

endring av referanse vs endring av objekt

ingen gruppetime på fredag :(

referanse til objektet

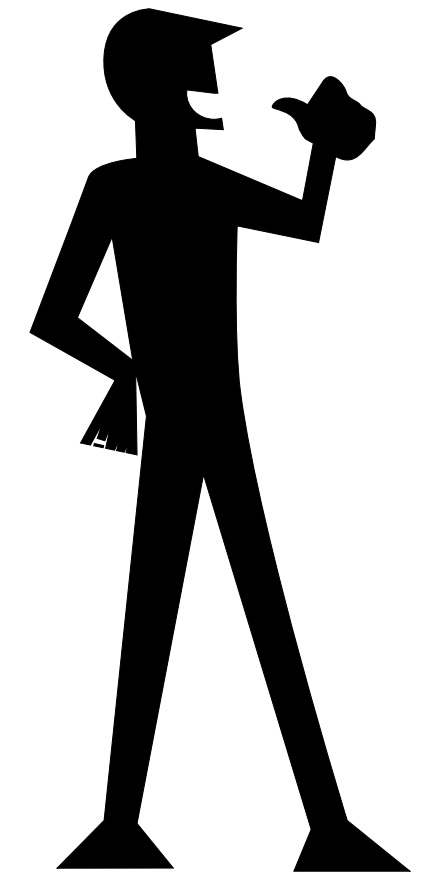
en referanse til objektet i seg selv: *self*



referanse til objektet

en referanse til objektet i seg selv: *self*

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

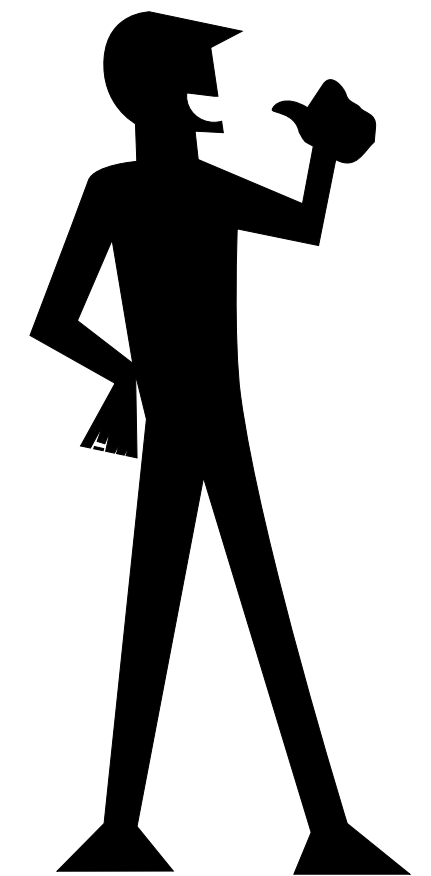


referanse til objektet

en referanse til objektet i seg selv: *self*

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på



referanse til objektet

```
def mjaug(self):  
    print(f"{self.navn} sier: Mjau!")
```

en referanse til objektet i seg selv: *self*

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

```
katt = Katt("Pus", 3)  
katt.mjaug
```

Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på

referanse til objektet

en referanse til objektet i seg selv: *self*

```
def mjau(self):  
    print(f"{self.navn} sier: Mjau!")
```

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

```
katt = Katt("Pus", 3)  
katt.mjau
```

Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på

referanse til objektet

en referanse til objektet i seg selv: *self*

```
def legg_til_eier(self, eier):  
    self.eier = eier
```

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

```
katt.legg_til_eier("Jonas")
```

Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på

referanse til objektet

en referanse til objektet i seg selv: *self*

```
def legg_til_eier(self, eier):  
    self.eier = eier
```

Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv


```
katt.legg_til_eier("Jonas")
```

Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på

referanse til objektet

en referanse til objektet i seg selv: *self*

```
def legg_til_eier(self, eier):  
    self.eier = eier
```



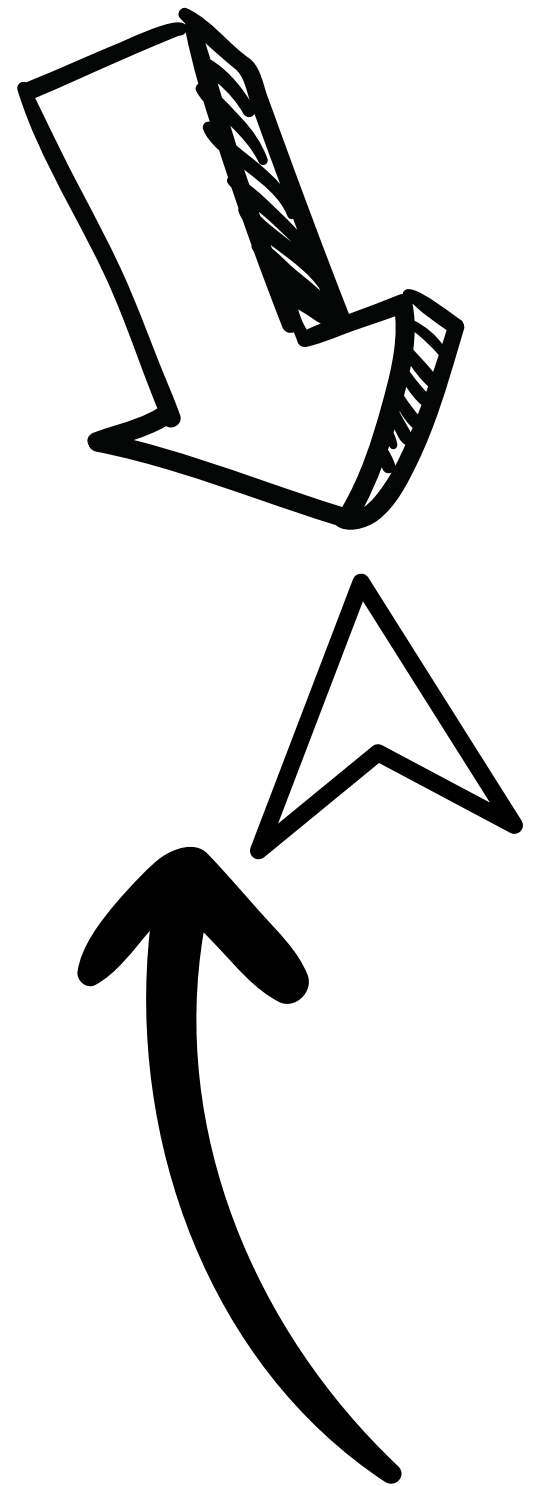
Når vi skal endre egenskapene til et objekt er det greit om den kan referere til seg selv

```
katt.legg_til_eier("Jonas")
```



Hvis ikke den har en *self* referanse, har vi ingen måte å referere til objektet vi vil endre på

referansevariabler og objekter



referansevariabler og objekter

referansevariabler er **ikke** selve objektet

referansevariabler og objekter

referansevariabler er **ikke** selve objektet

```
eier1 = Eier("Viktora")  
eier2 = Eier("Jonas")
```

referansevariabler og objekter

referansevariabler er **ikke** selve objektet

```
eier1 = Eier("Viktora")  
eier2 = Eier("Jonas")
```

```
eier1 -> [Eier-objekt med navn="Viktora"]  
eier2 -> [Eier-objekt med navn="Jonas"]
```

referansevariabler og objekter

referansevariabler er **ikke** selve objektet

Hva skjer egentlig om vi printer ut
eier1 og eier2?

```
eier1 = Eier("Viktora")  
eier2 = Eier("Jonas")
```

```
eier1 -> [Eier-objekt med navn="Viktora"]  
eier2 -> [Eier-objekt med navn="Jonas"]
```

referansevariabler og objekter

```
eier1 = Eier("Viktora")  
eier2 = Eier("Jonas")
```

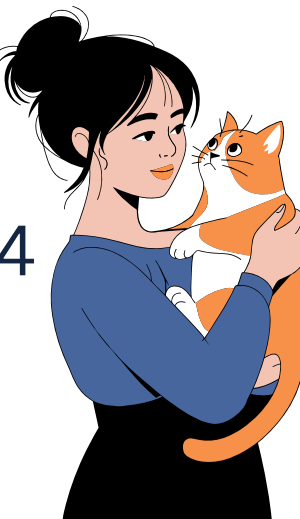
referansevariabler er **ikke** selve objektet

Hva skjer egentlig om vi printer ut
eier1 og eier2?

(eier1)
0x7ffe5367e044



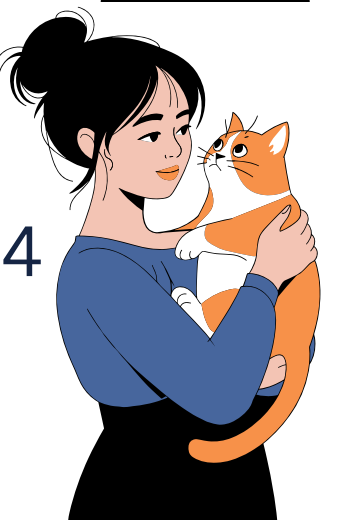
0x4ffe5463e044



(eier2)
0x3ffe5367e044



0x9ffe5463e044



=

objekter som refererer til andre objekter



objekter som refererer til andre objekter

Det er veldig vanlig at flere av klassene vi lager har referanser til andre objekter vi oppretter



objekter som refererer til andre objekter

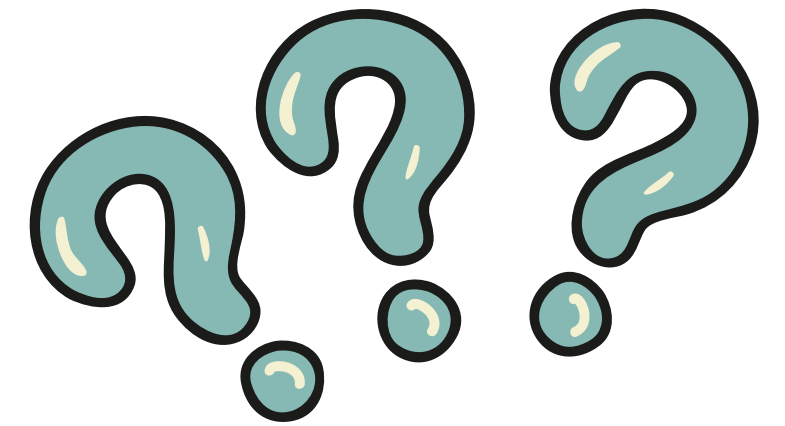
Det er veldig vanlig at flere av klassene vi lager har referanser til andre objekter vi oppretter

Progge et eksempel, Katt og Eier



endring av referanse vs endring av objekt

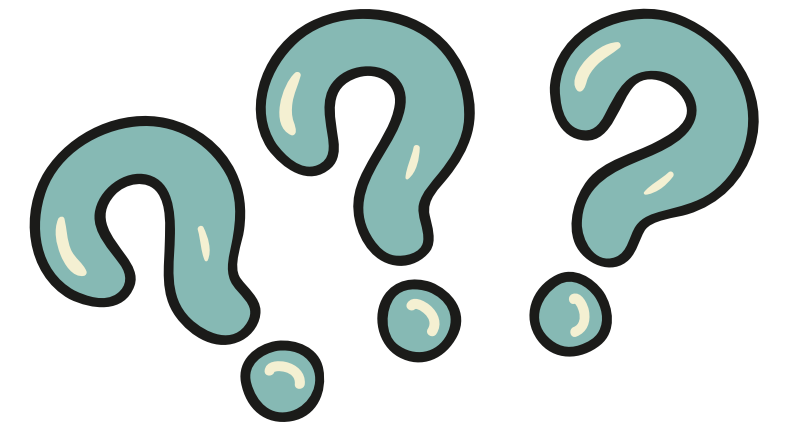
Det kan være forvirrende å vite hva som faktisk endres på når vi jobber med referansevariabler



endring av referanse vs endring av objekt

Det kan være forvirrende å vite hva som faktisk endres på når vi jobber med referansevariabler

Vi kan ha flere variabler som peker på **samme** objekt

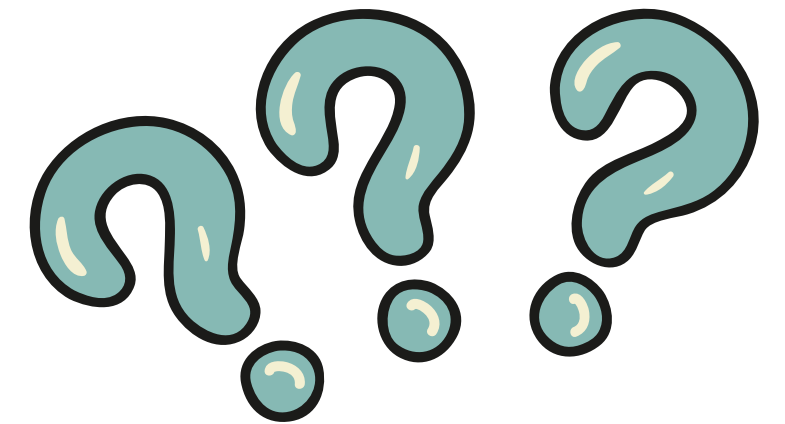


endring av referanse vs endring av objekt

Det kan være forvirrende å vite hva som faktisk endres på når vi jobber med referansevariabler

Vi kan ha flere variabler som peker på **samme** objekt

Hva skjer da når vi endrer på **det** objektet?



Liten oppklaring med begreper

objekt/instans = faktiske “tingen” (fks: Katt(“pus”))

klasse = oppskriften/blueprinten for å lage objekter

variabel = et navn du bruker for å peke på noe. kan tenke på det som en post-it lapp med et navn på, som du fester på en boks i minnet

referanse = selve koblingen mellom variabel og objekt

Selvstendig arbeid 