

Gruppetime uke 8

IN1000 - gruppe 7, høst 19

Ukens temaer

- Mer om OO-programmering:
 - 'self'
 - 'None'
 - Referanser
 - Datastruktur
 - Kontrollflyt

Repetisjon: 'self'

- 'self' brukes for at objektet skal referere til seg selv
- **Instansmetoder og konstruktør:** self må være første parameter
- **Instansvariabler:** `self._variabelnavn` signaliserer at det er private variabler som hører til hvert enkelt objekt. Skal ikke aksesseres utenfor klassen.
- For å kalle på en instansmetode i en annen instansmetode inne i selve klassen må metoden kalles på self:
 - `self.skrivUt()`
- Dette er en særtning for Python!

Gjennomgang av klassen Person

- Liveprogrammering: gjennomgang av eksempelet til egenoppgaven i Oblig 6:
 - person.py
 - testPerson.py

‘None’

- ‘None’ brukes for å representere *fraværet av en verdi*
- Eksempel: hvis du vil ha en variabel som skal brukes til et objekt, men ikke vil gi den noe verdi, kan variabelen settes lik None:
 - objektetMitt = None
- None er ‘default’-verdien til enhver metode, om noe ikke annet er spesifisert ved **return**
 - Lager vi en metode uten returverdi, så ser det ut som den ikke returnerer noe, men i virkeligheten returnerer den None

'None'

- Eksempel: en prosedyre (metode uten returverdi) returnerer alltid None:

```
def prosedyre():  
    print("Hei!")  
    print("Dette er en prosedyre")  
  
returverdi = prosedyre()  
print(returverdi)
```



```
Hei!  
Dette er en prosedyre  
None
```

'None' - fortsettelse Person-programmene

- None kan være nyttig å bruke som "startverdi" når vi leter etter et gitt objekt
- Liveprogrammering: finne en person (**objektet, ikke navnet**) i en liste:
 - testPerson.py

Oppgave

Oppgave 1

Skriv en klasse sirkel. En sirkel har en radius, lag en konstruktør som setter radius.

Lag tre metoder til: en som returnerer diameteren til sirkelen, en som returnerer omkretsen til sirkelen og en som returnerer arealet til sirkelen.

*Hint: omkrets av en sirkel er diameter * pi, arealet av en sirkel er radius² * pi. Eksponenter skrives som <base>**<eksponent>, f.eks. 2⁸ skrives 2**8.*

Lag deretter 2 sirkler med ulik radius.

Skriv ut den ene sirkelens omkrets og areal, og den andre sirkelens diameter.

Pause

Referanser og datastrukturer

- Referanser er en måte å få tak i objekter på
- Tegning: klassen Dyr fra forrige uke
 - Egenskaper: art, vekt, kjønn
 - hund = Dyr("hund", "5", "hannkjønn")
 - katt = Dyr("katt", "3", "hunkjønn")
 - 'katt' og 'hund' er **referanser** til objektene
 - Hva skjer hvis vi skriver dette?:
 - katt = hund
- Relasjonen mellom objekt og referanse: *minnegaten.pdf*

Større OOP-eksempel

- Ikke gjennomgått forrige time
 - bygning.py
 - testBygning.py

Hvis tid: oppgaver

Oppgave 2

Skriv en klasse *Leilighet*. Konstruktøren skal definere instansvariabler som representerer nummer (leilighetsnummer) og etasje. Skriv også metoder for å hente disse verdiene.

Oppgave 3

1. Skriv en klasse *Bygaard*. Konstruktøren til klassen skal definere en tom liste for leiligheter.
2. Skriv en metode som legger til en leilighet i listen med leiligheter.
3. Skriv en metode som tar i mot et leilighetsnummer og forsøker å finne og returnere leiligheten med dette nummeret. Hvis det ikke eksisterer en slik leilighet så skal metoden returnere *None*.

Til info

- Repetisjonstime mandager kl 12.15
 - Vanligvis på rom Prolog, men førstkommande mandag er det flyttet til rom Caml
- Oblig 7
 - Større OOP-oppgave
 - **Frist: 21. Oktober kl 23:59**
 - Utsettelse: forespørsler sendes til meg på mail (marialjo@ifi.uio.no). Det gis 3 kalenderdager utsettelse
 - Nytt forsøk: om innleveringen ikke blir godkjent, men kun har noen få mangler, gis det et nytt forsøk med en ny frist
 - Oblig 7 og 8 er **godkjent/ikke godkjent**, ikke poenggivende. Begge må bestås for å kvalifisere til eksamen
- Poengscore på oblig 1-6 publiseres i løpet av neste uke i Devilry
 - Minst 19 av 29 poeng kreves for å kunne levere oblig 7