

IN1000 – seminarartime uke 8

Gruppe 14 – Seminarrom Logo

Div. info

- Er du syk eller har luftveissymptomer? Hold deg hjemme! Finnes en haug med digitale tilbud på IFI
- Kontaktinfo: ahmedws@uio.no, eller huk tak i meg utenom timene
- Spørsmål til pensum, eller problemer dere har møtt på?

Agenda

- Objektorientert programmering
- Kort om oblig 6
- None og litt om self
- Datastrukturer og kontrollflyt – tegninger
- Referanser – gjennomgående tema for dagens gruppetime

Oblig 6 – motorsykkel-oppgaven

- Diskuter i 2-3 min: hvilke data ligger hvor? Hvordan programflyten er under kjøring – tegn gjerne opp et objekt og hvordan variabelen refererer til objektet.
- Hva skjer om man har flere motorsykler, eller at flere deler motorsykkel? Eks. en som tyvlåner en motorsykkel (og kmstand oppdateres automatisk)

Diskuter forskjellene mellom de to kodesnuttene

```
def finn_stud(self, navn) :  
    student = None  
    for s in self.studentliste :  
        if str(s) == navn :  
            student = s  
    return student
```

... kan også skrives slik ...

```
def finn_stud(self, navn) :  
    for s in self.studentliste :  
        if str(s) == navn :  
            return s
```

None (og litt self)

- Kan opprette objekter der man angir argumenter som None
- Default-verdien som returneres fra enhver metode, med mindre noe annet er definert
 - Med andre ord: lager vi en metode uten en returverdi, så ser det ut som den ikke returnerer noe, men i virkeligheten returnerer den None
- Self: brukes til å referere til seg selv, altså «denne» instansen av. Gir oss tilgang til instansvariabler og metoder i objektet

```
student = finn_stud(navn)
if student is None:
    print("Fant ikke studenten")
else:
    <gjøre andre ting, feks kalle på metoder fra student-objektet>
```

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.

Gitt følgende klasse og programkode – tegn opp datastrukturen og referansene

```
class Dyr:  
    def __init__(self, art, kjonn, vekt):  
        self._art = art  
        self._kjonn = kjonn  
        self._vekt = vekt  
  
    def hentArt(self):  
        return self._art  
  
    def hentKjonn(self):  
        return self._kjonn  
  
    def hentVekt(self):  
        return self._vekt
```

```
hund = Dyr("hund", "hann", 10.2)  
katt = Dyr("katt", "hunn", 5.7)  
sau = Dyr("sau", "hann", 30.6)
```

hund = katt
hund.hentArt()

Oppgave 3

Lag en klasse Blomst. En blomst har en art, en høyde, en instansvariabel som sier om blomsten har det bra eller ikke (denne verdien er en boolean), og en verdi som forteller hvor lenge siden blomsten ble vannet sist. Konstruktøren setter alle disse verdiene, når en blomst blir opprettet vil blomsten alltid ha det bra og det vil være 0 dager siden den ble vannet sist.

En blomst har en metode hentStatus som returnerer verdien som forteller om blomsten har det bra eller ikke.

I tillegg har den metoden skrivUtInfo, som skriver ut en info streng om blomsten

Videre har en blomst en metode nesteDag, som øker antall dager siden den ble vannet med en. Hvis det er mer enn tre dager siden blomsten ble vannet sist vil statusen til blomsten være at den har det dårlig. Hvis statusen til blomsten er at den har det bra vokser blomsten 1 cm.

Den siste metoden en blomst har er metoden vann. Hvis det er mindre enn 3 dager siden blomsten ble vannet vil statusen til blomsten bli dårlig (over vanning), ellers vil statusen være bra.