

prosedural / funksjonsorientert programmering

- sentrert rundt prosedyrer / funksjoner
- ulempe / utfordring: etter som programmet blir større og mere komplekst (og flere programmerere jobber med det samme prosjektet / programmet), interne bindinger / sammenhenger gjennom koden økes

objektorientert programmering

- sentrert rundt objekter (består både av metoder (≈ funksjoner / prosedyrer) og data)
- gir færre bindinger i programkoden
- ulempe / utfordring: vanskeligheter med objektorientert programmering mot relasjonsdatabase

pp: variable og funksjoner

oop: klasse, objekt (pakkeinst / instanse), metode, attributter

Klasse - mal/beskrivelse

Objekt - instanse/førekomst ("bruk av klassen")

En klasse må defineres og et objekt må instansieres

- klassenavn konvensjon i Python: Upper Camel Case, stor forboelse

- konstruktør: en metode som gjør det mulig å skape nye objekter basert på klassen som mal, i Python:
-- init -- som første metode i en klasse

- metoder, definisjon svært lik funksjoner. Hovedforskjell er at metodene har self som første parameter

```
class Mynt:
    def __init__(self):
        self.sideopp = 'Krone'
    ...

def main():
    min_mynt = Mynt()
    ...
```

Definerer klassen

self parameteren

første metode i klassen, instansierer objektet

instanse variabel

opprettet et myntobjekt (som heter min_mynt), førekomst/instanse

"Hva er metode-kode / logikk og hva er program logikk?"

"Hva kan vi gjøre med mynten / 5'eren?"

- "se hvilken side som ligger opp" startverdiene
- "kaste/flippe"
- "se hvilken side som ble opp etter flipping"

○ Dette blir metoder til myntobjektet / klassen

"Hvor mange ganger vi skal flippe og oppstilling av resultatet etter hvert kast er ikke noe vi kan gjøre med mynten".

└─ det blir program logikk

└─ så da skal vi ikke ha en
kast-tellopp / mynt-tellopp metode i klassen eller
oppstilling i kast-metoden

```
1  #PRG1100-2022-opptelling myntkast pp
2
3  #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
4  #funksjonsorientert/prosedural tilnærming
5  import random
6
7  antall_kron=0
8  antall_mynt=0
9
10 antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
11
12 for antall_ganger in range(1,antall_kast+1,1):
13     if random.randint(0,1)==0:
14         sideopp='Krone'
15         antall_kron=antall_kron+1
16     else:
17         sideopp='Mynt'
18         antall_mynt=antall_mynt+1
19     print('Resultatet av kast nr',antall_ganger,'ble',sideopp)
20 print('Resultatet av forsøksrekke ble',antall_kron,'Krone
    og',antall_mynt,'Mynt')
21
```

```
1 #PRG1100-2022-opptelling myntkast oo
2
3 #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
4 #objekt-orientert tilnærming, hvor klassens metoder håndterer egenskaper ved
  mynten
5 #og programlogikken/spillet ligger i main()
6 import random
7
8 class Mynt:
9     def __init__(self):
10         self.sideopp='Krone'
11
12     def kast(self):
13         if random.randint(0,1)==0:
14             self.sideopp='Krone'
15         else:
16             self.sideopp='Mynt'
17
18     def hent_sideopp(self):
19         return self.sideopp
20
21 def main():
22     antall_kron=0
23     antall_mynt=0
24
25     min_mynt=Mynt()
26
27     print('Før første kast er denne siden opp:',min_mynt.hent_sideopp())
28
29     antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
30
31     for antall_ganger in range(1,antall_kast+1,1):
32         min_mynt.kast()
33         print('Resultatet av kast
34         nr',antall_ganger,'ble',min_mynt.hent_sideopp())
35         if min_mynt.hent_sideopp()=='Krone':
36             antall_kron=antall_kron+1
37         else:
38             antall_mynt=antall_mynt+1
39
40     print('Resultatet av forsøksrekka ble',antall_kron,'Krone
41     og',antall_mynt,'Mynt')
42
43 main()
```

```

1  #PRG1100-2022-instansiering parameteroverføring
2
3  #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
4  #objekt-orientert tilnærming, hvor klassens metoder håndterer egenskaper ved
   mynten
5  #og programlogikken/spillet ligger i main()
6  import random
7
8  class Mynt:
9      def __init__(self,sideopp):
10         self.sideopp=sideopp
11
12     def kast(self):
13         if random.randint(0,1)==0:
14             self.sideopp='Krone'
15         else:
16             self.sideopp='Mynt'
17
18     def hent_sideopp(self):
19         return self.sideopp
20
21 def main():
22     antall_kron=0
23     antall_mynt=0
24
25     sideopp=input('Hvilken side på mynten er opp før første kast? ')
26
27     min_mynt=Mynt(sideopp)
28
29     print('Før første kast er denne siden opp:',min_mynt.hent_sideopp())
30
31     antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
32
33     for antall_ganger in range(1,antall_kast+1,1):
34         min_mynt.kast()
35         print('Resultatet av kast
36         nr',antall_ganger,'ble',min_mynt.hent_sideopp())
37         if min_mynt.hent_sideopp()=='Krone':
38             antall_kron=antall_kron+1
39         else:
40             antall_mynt=antall_mynt+1
41
42     print('Resultatet av forsøksrekke ble',antall_kron,'Krone
43     og',antall_mynt,'Mynt')
44
45     main()

```

```
1 #PRG1100-2022-instansiering input
2
3 #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
4 #objekt-orientert tilnærming, hvor klassens metoder håndterer egenskaper ved
  mynten
5 #og programlogikken/spillet ligger i main()
6 import random
7
8 class Mynt:
9     def __init__(self):
10         self.sideopp=input('Hvilken side på mynten er opp før første kast? ')
11
12     def kast(self):
13         if random.randint(0,1)==0:
14             self.sideopp='Krone'
15         else:
16             self.sideopp='Mynt'
17
18     def hent_sideopp(self):
19         return self.sideopp
20
21 def main():
22     antall_kron=0
23     antall_mynt=0
24
25     min_mynt=Mynt()
26
27     print('Før første kast er denne siden opp:',min_mynt.hent_sideopp())
28
29     antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
30
31     for antall_ganger in range(1,antall_kast+1,1):
32         min_mynt.kast()
33
34         print('Resultatet av kast
35 nr',antall_ganger,'ble',min_mynt.hent_sideopp())
36
37         if min_mynt.hent_sideopp()=='Krone':
38             antall_kron=antall_kron+1
39         else:
40             antall_mynt=antall_mynt+1
41
42     print('Resultatet av forsøksrekke ble',antall_kron,'Krone
43 og',antall_mynt,'Mynt')
44
45 main()
```

```

1  #PRG1100-2022-juksekode
2  #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
3  #objekt-orientert tilnærming, hvor klassens metoder håndterer egenskaper ved
   mynten
4  #og programlogikken/spillet ligger i main()
5
6  #Introduksjon til private/skjulte attributter
7  #Vi bør sikre oss at det bare er metodene til objektet som kan endre/aksessere
   attributtene/instansevariablene
8  #Slik det er nå kan kode i main endre verdier på attributter til objektet/dvs
   "jukse i spillet"
9
10 import random
11
12 #Mynt-klassen simulerer en mynt og hva en kan gjøre med den
13 class Mynt:
14     #__init__ metoden initierer objektet/forekomsten/instansen
15     #og tilordner sideopp-attributtet (self.sideopp) startverdi via en input i
       __init__
16     #dvs setter en startverdi som ikke skal telles med
17     def __init__(self):
18         #Oppgi "myntside" opp før første kast
19         self.sideopp=input('Hvilken side på mynten er opp før første kast? ')
20
21     #kast metoden simulerer ett kast med mynten
22     #og gir sideopp-attributtet ny verdi
23     def kast(self):
24         if random.randint(0,1)==0:
25             self.sideopp='Krone'
26         else:
27             self.sideopp='Mynt'
28
29     #hent_sideopp metoden returnerer til enhver tid
30     #verdien/("siste verdi") på mynten, dvs sideopp-attributtet
31     def hent_sideopp(self):
32         return self.sideopp
33
34 def main():
35     antall_kron=0
36     antall_mynt=0
37
38     #Oppretter et mynt-objekt, en forekomst/instanse
39     min_mynt=Mynt()
40
41     print('Før første kast er denne siden opp:',min_mynt.hent_sideopp())
42
43     antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
44
45     for antall_ganger in range(1,antall_kast+1,1):
46         #Mynten kastes
47         min_mynt.kast()
48
49         #Resultatet av kastet skrives ut
50         print('Resultatet av kast
           nr',antall_ganger,'ble',min_mynt.hent_sideopp())
51
52         #Her kommer den nye "jukse-koden"
53         #Uansett hva som blir resultatet av kast-metoden overstyrer vi
           resultatet til å bli f eks Krone
54         min_mynt.sideopp='Krone'
55         print('Resultatet av kast nr',antall_ganger,'ble manipulert
           til',min_mynt.hent_sideopp())
56         print()
57
58         #Opptelling gjennomføres
59         if min_mynt.hent_sideopp()=='Krone':
60             antall_kron=antall_kron+1
61         else:
62             antall_mynt=antall_mynt+1
63

```



```
64     print('Resultatet av forsøksrekke ble',antall_kron,'Krone  
65         og',antall_mynt,'Mynt')  
66 main()  
67  
68
```

```

1  #PRG1100-2022-private attributter
2
3  #Program for brukerbestemt antall myntkast og opptelling av antall Krone og Mynt
4  #objekt-orientert tilnærming, hvor klassens metoder håndterer egenskaper ved
   mynten
5  #og programlogikken/spillet ligger i main()
6
7  #For å sikre oss at annen kode ikke kan endre/korrupte verdier på objektets
   data-attributter/instansevariabler
8  #gjør vi attributtene private, dvs at det bare er objektets metoder som har
   direkte aksess/tilgang til data-attributtene
9  #I Python gjøres det ved å starte attributt-navnet med 2 __, dvs __
10 import random
11
12 class Mynt:
13     def __init__(self):
14         self.__sideopp=input('Hvilken side på mynten er opp før første kast? ')
15
16     def kast(self):
17         if random.randint(0,1)==0:
18             self.__sideopp='Krone'
19         else:
20             self.__sideopp='Mynt'
21
22     def hent_sideopp(self):
23         return self.__sideopp
24
25 def main():
26     antall_kron=0
27     antall_mynt=0
28
29     min_mynt=Mynt()
30
31     print('Før første kast er denne siden opp:',min_mynt.hent_sideopp())
32
33     antall_kast=int(input('Hvor mange ganger skal mynten kastes? '))
34
35     for antall_ganger in range(1,antall_kast+1,1):
36         min_mynt.kast()
37
38         print('Resultatet av kast
   nr',antall_ganger,'ble',min_mynt.hent_sideopp())
39
40         min_mynt.__sideopp='Krone'
41         print('Resultatet av kast nr',antall_ganger,'ble forsøkt manipulert
   men hindret',min_mynt.hent_sideopp())
42         print()
43
44         if min_mynt.hent_sideopp()=='Krone':
45             antall_kron=antall_kron+1
46         else:
47             antall_mynt=antall_mynt+1
48
49     print('Resultatet av forsøksrekka ble',antall_kron,'Krone
   og',antall_mynt,'Mynt')
50
51 main()
52
53

```

```

1 #PRG1100-2022-flere instanser
2
3 #Flere objekter/instanser av samme klasse, her: "kontoene til flere personer,
  Kari og Knut"
4
5 class BankKonto:
6     def __init__(self,saldo):
7         self.__saldo=saldo
8
9     def innskudd(self,belop):
10        self.__saldo=self.__saldo+belop
11
12    def uttak(self,belop):
13        if self.__saldo>=belop:
14            self.__saldo=self.__saldo-belop
15        else:
16            print('Feil: ikke nok på konto')
17
18    def hent_saldo(self):
19        return self.__saldo
20
21 def main():
22     saldo=float(input('Hva er saldoen på konto til Kari? '))
23
24     karis_konto=BankKonto(saldo)
25
26     saldo=float(input('Hva er saldoen på konto til Knut? '))
27
28     knuts_konto=BankKonto(saldo)
29
30     belop=float(input('hvor mye skal Kari sette inn på konto? '))
31     karis_konto.innskudd(belop)
32
33     print('Saldoen på kontoen til Kari er:',karis_konto.hent_saldo())
34
35     belop=float(input('hvor mye skal Knut sette inn på konto? '))
36     knuts_konto.innskudd(belop)
37
38     print('Saldoen på kontoen til Knut er:',knuts_konto.hent_saldo())
39
40     belop=float(input('hvor mye skal Kari ta ut fra konto? '))
41     karis_konto.uttak(belop)
42
43     print('Saldoen på kontoen til Kari er:',karis_konto.hent_saldo())
44
45     belop=float(input('hvor mye skal Knut ta ut fra konto? '))
46     knuts_konto.uttak(belop)
47
48     print('Saldoen på kontoen til Knut er:',knuts_konto.hent_saldo())
49
50 main()
51

```