

```
1 #bankkonto.py
2
3 class BankKonto:
4     def __init__(self,saldo):
5         self.__saldo=saldo
6
7     def innskudd(self,belop):
8         self.__saldo=self.__saldo+belop
9
10    def uttak(self,belop):
11        if self.__saldo>=belop:
12            self.__saldo=self.__saldo-belop
13        else:
14            print('Feil: ikke nok på konto')
15
16    def hent_saldo(self):
17        return self.__saldo
18
```

```
1 #PRG1100-2022-moduler
2
3 #Klassen BankKonto lagres i en modul/egen py-fil, bankkonto.py,
4 #vil da være tilgjengelig for mange programmer via import
5 #Det er fila som importeres og gjør klassen tilgjengelig for programmet
6 #Ved oppretting av objekter må en da sette <modulnavn.> foran klassenavnet
7 import bankkonto
8
9 def main():
10     saldo=float(input('Hva er saldoen på konto til Kari? '))
11
12     karis_konto=bankkonto.BankKonto(saldo)
13
14     saldo=float(input('Hva er saldoen på konto til Knut? '))
15
16     knuts_konto=bankkonto.BankKonto(saldo)
17
18     belop=float(input('hvor mye skal Kari sette inn på konto? '))
19     karis_konto.innskudd(belop)
20
21     print('Saldoen på kontoen til Kari er:',karis_konto.hent_saldo())
22
23     belop=float(input('hvor mye skal Knut sette inn på konto? '))
24     knuts_konto.innskudd(belop)
25
26     print('Saldoen på kontoen til Knut er:',knuts_konto.hent_saldo())
27
28     belop=float(input('hvor mye skal Kari ta ut fra konto? '))
29     karis_konto.uttak(belop)
30
31     print('Saldoen på kontoen til Kari er:',karis_konto.hent_saldo())
32
33     belop=float(input('hvor mye skal Knut ta ut fra konto? '))
34     knuts_konto.uttak(belop)
35
36     print('Saldoen på kontoen til Knut er:',knuts_konto.hent_saldo())
37
38 main()
39
```

```

1 #PRG1100-2022-objekt flere dataattributter
2
3 #Objekter med flere dataattributter
4
5 class Student:
6     def __init__(self, studentnr, fornavn, etternavn, epost, studium):
7         self.__studentnr=studentnr
8         self.__fornavn=fornavn
9         self.__etternavn=etternavn
10        self.__epost=epost
11        self.__studium=studium
12
13        # __str__ metoden, holder orden på "An object's state", dvs verdiene til
14        # attributtene
15        def __str__(self):
16            return 'Objektets attributter er: ' + self.__studentnr + '\n' +
17                self.__fornavn + '\n' + self.__etternavn + '\n' + self.__epost + '\n'
18                + self.__studium + '\n'
19
20 studentnr=input('Oppgi studentnr: ')
21 fornavn=input('Oppgi fornavn: ')
22 etternavn=input('Oppgi etternavn: ')
23 epost=input('Oppgi e-post: ')
24 studium=input('Oppgi studium: ')
25
26 ny_student=Student(studentnr, fornavn, etternavn, epost, studium)
27
28 #Skriver ut verdiene til attributtene til objektet ny_student
29 # __str__ kalles ved å sende et objekt som argument til print-funksjonen
30 print(ny_student)

```

Generelt er det være vanlig å ha en metode som endrer verdien på ett dataattributt for hvert dataattributt til objektet

- mutator-metoder/namngis som set'ere

Tilsvarende en metode som viser verdien på ett dataattributt for hvert dataattributt til objektet

- accessor-metoder/namngis som get'ere

```

1 #PRG1100-2022-set og get metoder
2
3 #set'ere og get'ere som metoder for å endre og skrive ut verdier på
  enkelt-attributter
4
5 class Student:
6     # __init__ metoden instansierer objektet med verdier via variable og
      parameteroverføring
7     def __init__(self, studentnr, fornavn, etternavn, epost, studium):
8         self.__studentnr=studentnr
9         self.__fornavn=fornavn
10        self.__etternavn=etternavn
11        self.__epost=epost
12        self.__studium=studium
13
14    def set_studentnr(self, studentnr):
15        self.__studentnr=studentnr
16
17    def set_fornavn(self, fornavn):
18        self.__fornavn=fornavn
19
20    def set_etternavn(self, etternavn):
21        self.__etternavn=etternavn
22
23    def set_epost(self, epost):
24        self.__epost=epost
25
26    def set_studium(self, studium):
27        self.__studium=studium
28
29    def get_studentnr(self):
30        return self.__studentnr
31
32    def get_fornavn(self):
33        return self.__fornavn
34
35    def get_etternavn(self):
36        return self.__etternavn
37
38    def get_epost(self):
39        return self.__epost
40
41    def get_studium(self):
42        return self.__studium
43
44    def __str__(self):
45        return 'Objektets attributter er: ' + self.__studentnr + '\n' +
          self.__fornavn + '\n' + self.__etternavn + '\n' + self.__epost + '\n'
          + self.__studium + '\n'
46
47
48 studentnr=input('Oppgi studentnr: ')
49 fornavn=input('Oppgi fornavn: ')
50 etternavn=input('Oppgi etternavn: ')
51 epost=input('Oppgi e-post: ')
52 studium=input('Oppgi studium: ')
53
54 ny_student=Student(studentnr, fornavn, etternavn, epost, studium)
55
56 print(ny_student)
57
58 #Skriver ut epost og studium
59 print(ny_student.get_epost())
60 print(ny_student.get_studium())
61 epost=input('Oppgi ny e-post: ')
62 ny_student.set_epost(epost)
63 studium=input('Oppgi nytt studium: ')
64 ny_student.set_studium(studium)
65 print()
66 print(ny_student)

```

```

1  #PRG1100-2022-binærlagring av objekt
2
3  #Importerer pickle-modulen for å serialisere/konvertere et objekt til en
   bit-strøm
4  #som kan lagres til fil for senere henting/bruk
5  import pickle
6
7  class Student:
8      def __init__(self, studentnr, fornavn, etternavn, epost, studium):
9          self.__studentnr=studentnr
10         self.__fornavn=fornavn
11         self.__etternavn=etternavn
12         self.__epost=epost
13         self.__studium=studium
14
15     def set_studentnr(self, studentnr):
16         self.__studentnr=studentnr
17
18     def set_fornavn(self, fornavn):
19         self.__fornavn=fornavn
20
21     def set_etternavn(self, etternavn):
22         self.__etternavn=etternavn
23
24     def set_epost(self, epost):
25         self.__epost=epost
26
27     def set_studium(self, studium):
28         self.__studium=studium
29
30     def get_studentnr(self):
31         return self.__studentnr
32
33     def get_fornavn(self):
34         return self.__fornavn
35
36     def get_etternavn(self):
37         return self.__etternavn
38
39     def get_epost(self):
40         return self.__epost
41
42     def get_studium(self):
43         return self.__studium
44
45     def __str__(self):
46         return 'Objektets attributter er: ' + self.__studentnr + '\n' +
47             self.__fornavn + '\n' + self.__etternavn + '\n' + self.__epost + '\n'
48             + self.__studium + '\n'
49
50 studentnr=input('Oppgi studentnr: ')
51 fornavn=input('Oppgi fornavn: ')
52 etternavn=input('Oppgi etternavn: ')
53 epost=input('Oppgi e-post: ')
54 studium=input('Oppgi studium: ')
55
56 ny_student=Student(studentnr, fornavn, etternavn, epost, studium)
57
58 print(ny_student)
59
60 print(ny_student.get_epost())
61 print(ny_student.get_studium())
62 epost=input('Oppgi ny e-post: ')
63 ny_student.set_epost(epost)
64 studium=input('Oppgi nytt studium: ')
65 ny_student.set_studium(studium)
66 print()
67 print(ny_student)

```

```
68 #Nytt, serialisering og skriving av objektet til fil
69 #Åpner en fil for binær skriving
70 #uten encoding - binary mode doesn't take an encoding argument
71 studentfil=open('student.dat','wb')
72
73 #Pickler objektet og lagrer på fil
74 pickle.dump(ny_student,studentfil)
75
76 #Lukker fila
77 studentfil.close()
78
79
80
81
82
83
```

```
1 #PRG1100-2022-fra tekstfil til binærfil via objekt
2
3 #Instansiere objektet med innlesing av data fra fil
4 #Serialiserer og lagrer objektet
5 import pickle
6
7 class Ansatt:
8     def __init__(self, fornavn, etternavn, epost):
9         self.__fornavn=fornavn
10        self.__etternavn=etternavn
11        self.__epost=epost
12
13    def __str__(self):
14        return 'Objektets attributter er: ' + self.__fornavn + '\n' +
15        self.__etternavn + '\n' + self.__epost + '\n'
16
17 ansatt_txt_fil=open('Laerer.txt','r',encoding='utf-8')
18 ansatt_dat_fil=open('Laerer.dat','wb')
19
20 fornavn=ansatt_txt_fil.readline()
21 while fornavn != '':
22     fornavn=fornavn.rstrip('\n')
23     etternavn=ansatt_txt_fil.readline().rstrip('\n')
24     epost=ansatt_txt_fil.readline().rstrip('\n')
25
26     #Legger variablene i attributtene til objektet
27     ny_ansatt=Ansatt(fornavn,etternavn,epost)
28
29     print(ny_ansatt)
30
31     #Serialisering og skriving av objektet til fil
32     pickle.dump(ny_ansatt,ansatt_dat_fil)
33
34     fornavn=ansatt_txt_fil.readline()
35
36 ansatt_dat_fil.close()
37 ansatt_txt_fil.close()
38
```



Klassediagram for klassen Student.

| Student  |
|--|
| <u>__studentnr</u><br><u>__fornavn</u><br><u>__etternavn</u><br><u>__epost</u><br><u>__studium</u>   |
| <u>__init__</u> (studentnr,fornavn,etternavn,epost,studium)<br>set_studentnr(studentnr)<br>set_fornavn(fornavn)<br>set_etternavn(etternavn)<br>set_epost(epost)<br>set_studium(studium)<br>get_studentnr()<br>get_fornavn()<br>get_etternavn()<br>get_epost()<br>get_studium() |