

Uke 9

Oppgave 1 – objekter og referansevariabler

Gitt klassen under

```
class Sparegris:

    def __init__(self, startBeloe):
        self._beloe = startBeloe

    def seBeloe(self):
        return self._beloe

    def taUt(self, beloeUt):
        self._beloe -= beloeUt

    def settInn(self, beloeInn):
        self._beloe += beloeInn
```

a) Hva vil skrives ut her?

```
from sparegris import Sparegris

persSG = Sparegris(0)
paalsSG = persSG

print("Per setter inn 50 kr")
persSG.settInn(50)
print("Paals spargris inneholder", paalsSG.seBeloe(), "kr")
print("Paal tar ut 10 kr")
paalsSG.taUt(10)
print("Pers spargris inneholder", persSG.seBeloe(), "kr")
print("Paals spargris inneholder", paalsSG.seBeloe(), "kr")
```

b) Hva skrives ut her?

```
print("persSG is paalsSG:", persSG is paalsSG)
print("persSG==paalsSG:", persSG==paalsSG)
```

c) Hvor mange objekter av Sparegris-klassen har vi? Hvor mange referansevariabler?

```
persSG = Sparegris(0)
paalsSG = persSG
karisSG = Sparegris(50)
mettesSG = Sparegris(40)
hannesSG = mettesSG
meretesSG = karisSG
ingersSG = Sparegris(100)
heidisSG = Sparegris(0)
```

Oppgave 2

- a) Utvide Student-klassen med en metode tarFag(self, fagnavn) som returnerer True dersom studenten tar faget og ellers False
- b) Lag metoden __str__(self) som returnerer en tekststreng med informasjon om studenten
- c) Lag metoden __eq__(self, annenStudent) hvor annenStudent er en referanse til en annen student. Metoden skal returnere True dersom den andre studenten har samme navn og brukernavn.
- d) Kjør testStudent.py og sørg for at metodene er riktige

Eksempel på studentklassen:

```
class Student:

    def __init__(self, navn, brnavn, program, fag):
        self._navn = navn
        self._program = program
        self._brukernavn = brnavn
        self._fagListe = fag
        self._studiepoeng = 0

    def hentNavn(self):
        return self._navn

    def hentBrukernavn(self):
        return self._brukernavn

    def hentProgram(self):
        return self._program

    def hentFagListe(self):
        return self._fagListe

    def hentStudiepoeng(self):
        return self._studiepoeng

    def byttStudieprogram(self, nyttProgram):
        self._program = nyttProgram

    def leggTilFag(self, nyttFag):
        self._fagListe.append(fag)

    def fjernFag(self, faget):
        if faget in self._fagListe:
            self._fagListe.remove(faget)

    def ferdigMedFag(self, faget):
        if faget in self._fagListe:
            self.fjernFag(faget)
            self._studiepoeng += 10

    def tarFag(self, fagnavn):
        #fyll inn

    def __str__(self):
        # fyll inn

    def __eq__(self, annenStudent):
        # fyll inn
```

Eksempel på testStudent.py

```
from student import Student

stud1 = Student("Geir Geirson", "ggeir", "digøk",
               ["IN1000", "IN1020", "INEC1800"])
stud2 = Student("Hanne Hårfagre", "hhaar", "digøk",
               ["IN1000", "IN1020", "INEC1800"])
stud3 = Student("Inger Iversen", "iiver", "robotikk",
               ["IN1000", "IN1020", "MAT1100"])
stud4 = Student("Per Persson", "ppers", "prosa",
               ["IN1000", "IN1020", "EXPHIL03"])
stud5 = stud1
stud6 = Student("Hanne Hårfagre", "hhaar", "robotikk", [])

print("\nstud1:", stud1)
print("stud2:", stud2)
print("stud3:", stud3)
print("stud4:", stud4)
print("stud5:", stud5, "\n")

assert stud1.tarFag("IN1000")
assert stud4.tarFag("EXPHIL03")
assert not stud2.tarFag("MAT1100")
# alt: assert stud2.tarFag("MAT1100") == False

assert not stud1 == stud2
#alt: assert (stud1 == stud2) == False
assert not stud2 == stud3
assert not stud1 == stud3
assert not stud3 == stud4
assert stud1 == stud5
assert stud2 == stud6
```