Oppgavesett for uke 9 time 1

Oppgave 1

```
1  class Person:
2      def __init__(self, fødselsnummer, navn):
3          self._fødselsnummer = fødselsnummer
4          self._navn = navn
5
6      def __str__(self):
7          return self._fødselsnummer + " (" + self._navn + ")"
8
9  print(Person("01010112345", "Indigo Matheus"))
```

a) Hva blir skrevet ut til terminalen ved kjøring av programmet over? Kom frem til et svar før dere kjører programmet.
b) Hva skjer hvis vi bruker `print(self._fødselsnummer + " (" + self._navn + ")")` i stedet for `return self._fødselsnummer + " (" + self._navn + ")"` ?
c) Hva blir skrevet ut til terminalen ved fjerning av `__str__`-metoden over? Kom frem til et svar før dere kjører programmet.

Oppgave 2

```
1   class Adresse:
2       def __init__(self, gatenavn, bolignummer, postnummer):
3           self._gatenavn = gatenavn
4           self._bolignummer = bolignummer
5           self._postnummer = postnummer
6
7
8       def __eq__(self, annen):
9           return (self._gatenavn == annen._gatenavn and
10          self._bolignummer == annen._bolignummer and
11          self._postnummer == annen._postnummer)
12
13
14  adr1 = Adresse("Pythongata", 1, 0000)
15  adr2 = Adresse("Pythongata", 1, 0000)
16  print(adr1 == adr2)
17  print(adr1 is adr2)
```

a) Hva skrives ut til terminalen med `print(adr1 == adr2)` i programmet over? Kom frem til et svar før dere kjører programmet.
b) Hva skrives ut til terminalen med `print(adr1 is adr2)` i programmet over? Kom frem til et svar før dere kjører programmet.
c) Sammenlign svarene i a) og b). Hva oppdager dere, og hvorfor er det slik?
d) Hvorfor trenger vi ikke egne metoder for å hente ut `_gatenavn`, `_bolignummer` og `_postnummer` fra `annen`-objektet i `__eq__`-metoden?

Oppgave 3

```
class Konsulentselskap:
    def __init__(self, organisasjonsnr, selskapsnavn, beskrivelse):
        self._organisasjonsr = organisasjonsnr
        self._selfskapsnavn = selskapsnavn
        self._beskrivelse = beskrivelse
```

a) Gitt klassen ovenfor, lag en `__str__`-metode med informasjon om konsulentselskap.
b) Lag en `__eq__`-metode som returnerer `True` hvis organisasjonsnr er likt, ellers `False`.

Oppgave 4

```
class Matbutikk:
    def __init__(self):
        self._varer = []


    def legg_til_vare(self, vare):
        self._varer.append(vare)


    def hent_varer(self):
        return self._varer


class Vare:
    def __init__(self, vareid, varenavn):
        self._vareid = vareid
        self._varenavn = varenavn

```

```
20  def hovedprogram():
21      skinkebutikk = Matbutikk()
22      for i in range(5):
23          ny_skinke = Vare(i, "Skinke-" + str(i))
24          skinkebutikk.legg_til_vare(ny_skinke)
25
26
27  hovedprogram()
```

a) Utvid programmet ovenfor med en `__str__`-metode i `Vare`-klassen. Metoden skal inkludere både vareid og varenavn.

b) Fortsett å utvide programmet, slik at det ved hjelp av en annen for-løkke skriver ut vareid og varenavn til terminalen for varene i lista `_varer` refererer til. Dere skal dra nytte av `__str__`-metoden fra a) i løsningen (men aldri kalle på denne direkte).

c) Det er viktig at en vare med samme vareid og varenavn som en som allerede ligger i lista `_varer` refererer til, ikke kan legges til flere ganger. Utvid programmet med slik funksjonalitet ved å inkludere en `__eq__`-metode som del av løsningen. Hint: dere kan bruke `str(self)` og `str(annen)` i `__eq__`-metoden når to varer skal sammenlignes.

d) Hvorfor kunne dere ikke brukt `str(self)` og `str(annen)` i c) hvis `Vare`-klassen ikke hadde en `__str__`-metode?