Oppgaver uke 9 - konsept

Oppgave 1

```
class Person:
    def __init__(self, fødselsnummer, navn):
        self._fødselsnummer = fødselsnummer
        self._navn = navn

    def __str__(self):
        return self._fødselsnummer + " (" + self._navn + ")"

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

```
class Adresse:
    def __init__(self, gatenavn, bolignummer, postnummer):
        self._gatenavn = gatenavn
        self._bolignummer = bolignummer
        self._postnummer = postnummer

    def __eq__(self, annen):
        return (self._gatenavn == annen._gatenavn and
        self._bolignummer == annen._bolignummer and
        self._postnummer == annen._postnummer)

adr1 = Adresse("Pythongata", 1, 0000)
adr2 = Adresse("Pythongata", 1, 0000)
print(adr1 == adr2)
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
```

```
def hovedprogram():
    skinkebutikk = Matbutikk()
    for i in range(5):
        ny_skinke = Vare(i, "Skinke-" + str(i))
        skinkebutikk.legg_til_vare(ny_skinke)

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?

Oppgave 5

```
class Telefonnummer:
    def __init__(self):
        self._landskode = None
        self._nummer = None

    def sett_landskode(self, landskode):
        self._landskode = landskode

    def sett_nummer(self, nummer):
        self._nummer = nummer

    def __str__(self):
        return "+" + str(self._landskode) + " " + str(self._nummer)
```

```
def hovedprogram():
    telefonbok = {
                    "Iris Holter" : Telefonnummer(),
                    "Jørgen Olaisen" : Telefonnummer(),
                    "Laila Ringsaker" : Telefonnummer()
                }

hovedprogram()
```

a) Hva skjer hvis dere skriver ut Telefonnummer-objektet til Jørgen Olaisen? Forklar hvorfor det er slik.

b) Det skal kun være mulig å sette landskode hvis det ikke finnes fra før, og det skal kun være mulig å sette nummer hvis det ikke finnes fra før. Gjør nødvendige endringer i Telefonnummer-klassen for å møte disse kravene. Hvis landskode eller nummer (eller begge) finnes fra før, skal brukeren få beskjed om dette via utskrift i terminalen.

c) Utvid hovedprogrammet, slik at Iris Holter, Jørgen Olaisen og Laila Ringsaker alle får et telefonnummer bestående av landskode og nummer. Landskode skal være 47 (for Norge), men nummer velger dere selv.

d) Hovedprogrammet trenger også funksjonalitet for å skrive ut navn og telefonnummer for alle i telefonboka. Skriv kode for dette.

e) Prøv å sette landskode og nummer for Laila Ringsaker, selv om de finnes fra før.

f) Hvis en person i telefonboka skulle byttet telefonnummer (etter at de har fått både landskode og nummer), hvordan skulle man fått til dette?