

IN1000 - Seminaroppgaver til uke 8

Oppgave 1

Skriv en klasse sirkel. En sirkel har en radius, lag en konstruktør som setter radius.

Lag tre metoder til: en som returnerer diameteren til sirkelen, en som returnerer omkretsen til sirkelen og en som returnerer arealet til sirkelen.

*Hint: omkrets av en sirkel er diameter * pi, arealet av en sirkel er radius² * pi. Eksponenter skrives som <base>**<eksponent>, feks. 2⁸ skrives 2**8.*

Lag deretter 2 sirkler med ulik radius.

Skriv ut den ene sirkelens omkrets og areal, og den andre sirkelens diameter.

Løsning

Oppgave 2

Gitt følgende kode:

```
class Bil:
    def __init__(self, regnr, merke):
        self._regnr = regnr
        self._merke = merke

    def hent_regnr(self):
        return self._regnr

    def hent_merke(self):
        return self._merke

    def __eq__(self, annen):
        return self._regnr == annen._regnr

bil1 = Bil("AB23547", "Toyota")
```

```
bil2 = Bil("CD45679", "Mazda")
bil3 = Bil("AB23547", "Ford")
bil4 = bil3
```

Hva blir resultatene av følgende sannhetsuttrykk?

1. `bil1 == bil2`
2. `bil1 == bil3`
3. `bil3 == bil4`
4. `bil3 is bil4`
5. `bil1 is bil3`
6. `bil4 is bil3`

Oppgave 3

Skriv en klasse *Leilighet*. Konstruktøren skal definere instansvariabler som representerer nummer (leilighetsnummer) og etasje. Skriv også metoder for å hente disse verdiene.

Oppgave 4

1. Skriv en klasse *Bygaard*. Konstruktøren til klassen skal definere en tom liste for leiligheter.
2. Skriv en metode som legger til en leilighet i listen med leiligheter.
3. Skriv en metode som tar i mot et leilighetsnummer og forsøker å finne og returnere leiligheten med dette nummeret. Hvis det ikke eksisterer en slik leilighet så skal metoden returnere *None*.

u

Oppgave 5

Gitt klassen under, lag et hovedprogram som skal bestå av deloppgavene under:

```
class Student:
    def __init__(self, n):
        self._navn = n

    def hentNavn(self):
        return self._navn
```

5.1

Lag en liste for studenter. Opprett 5 studenter, en av studentene skal ha ditt navn, og legg alle studentene i listen.

5.2

Sjekk om du finnes i listen (en student med ditt navn).

5.3

Fjern deg selv fra listen.

5.4

Sjekk hvor mange studenter som er i listen.

5.5

Endre oppgave 5.2 til å ligge i en egen funksjon, som ser slik ut:

```
def sjekk(liste, navn):
```

Denne skal returnere Studenten dersom det finnes en student med det oppgitte navnet, ellers skal den returnere None.

5.6

Endre oppgave 5.3 til å ligge i en funksjon som ser slik ut:

```
def fjern(liste, navn):
```

Denne skal returnere True dersom den fikk til å fjerne en student med det oppgitte navnet, og False dersom det ikke gikk (ikke fant noen med det navnet i listen).

