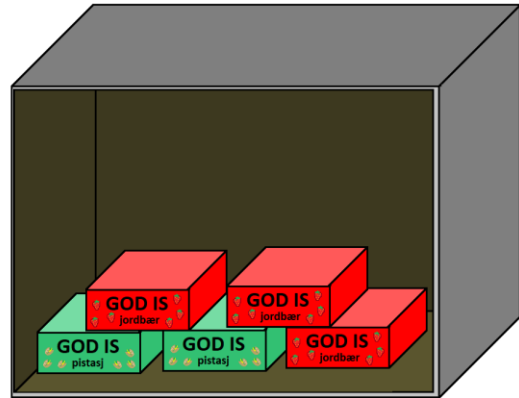


Hvor mye is er det i fryseren?

Et veldig kort eksempel for å se på hvordan vi bruker metoder for å hente verdiene til instansvariabler i en klasse som brukes inne i en annen klasse.

Vi gjør det ved å lage to klasser – én isboksklasse, og én fryserklasse.



Isboks-klassen

Vi lager klassen veldig enkel for dette formålet. Det eneste vi krever er at vi må spesifisere hvor mange liter is det er i en boks, når vi oppretter et objekt av denne klassen. I tillegg trenger vi en hente-metode for å hente ut instansvariabelen. Husk at dette er en del av innkapslingen. Når vi skriver objektorientert kan vi ikke hente verdien direkte (selv om Python tillater det), vi må lage en metode for å hente den.

```
class Isboks:
    def __init__(self, liter):
        self._liter = liter

    def hentLiter(self):
        return self._liter
```

Fryser-klassen

Denne klassen er heller ikke komplisert. Vi krever ikke at vi må oppgi noen argumenter når vi oppretter et objekt av klassen, men vi gjør det sånn at når et objekt opprettes, så har det en tom liste, som vi kan ha referanser til Isboks-objekter i.

For at objekter av Fryser-klassen skal kunne få noen isbokser, må vi lage en metode som kan legge dem inn. Denne metoden tar inn et Isboks-objekt, og legger det inn i lista. Merk at vi ikke sjekker om objektet finnes fra før av, eller noe sånt.

Til slutt lager vi en metode som teller antall liter som finnes i fryseren totalt. Vi vet at lista vi går gjennom inneholder Isboks-objekter, fordi det er sånn vi tenker oss at koden skal fungere. Siden vi vet at det er Isboks-objekter, vet vi også at objekter av denne klassen har en metode som heter hentLiter(), som returnerer verdien vi leter etter. Så er det bare å legge sammen alle disse verdiene i en egen variabel (antLiter), og returnere den til slutt. Merk at metoden vi lagde tidligere, leggTills() ikke vil si ifra om vi legger til

noe annet enn Isboks-objekter, men vi ville fått feil i hentIsMengde() hvis det er andre objekttyper i lista.

```
class Fryser:
    def __init__(self):
        self._isBokser = []

    def leggTilIs(self, isboks):
        self._isBokser.append(isboks)

    def hentIsMengde(self):
        antLiter = 0
        for isboks in self._isBokser:
            antLiter += isboks.hentLiter()
        return antLiter
```

Test

Til slutt kan vi teste klassene og metodene ved å lage en testklasse. Vi oppretter 3 isbokser på 2 liter hver (jordbærisen i bildet over?), og 2 isbokser på 3 liter hver (pistasjisen i bildet over?). Merk at vi oppretter objektene av Isboks-klassen "anonymt", vi legger dem til uten å mellomlagre referansen til dem i en variabel.

Vi skriver ut ismengden til Fryser-objektet tre ganger, så vi kan se endringene.

```
def hovedprogram():
    frysa = Fryser()
    print(frysa.hentIsMengde())

    for i in range(3):
        frysa.leggTilIs(Isboks(2))

    print(frysa.hentIsMengde())

    for i in range(2):
        frysa.leggTilIs(Isboks(3))
    print(frysa.hentIsMengde())
```

```
hovedprogram()
```

Utskriften blir som forventet:

```
0
6
12
>>>
```

Det er et enkelt eksempel, men jeg håper det kan være nyttig å se gjennom det!