IN1000 – Uke 8

Plan for timen

- Instansvariabler vs lokale variabler vs parametre
- Variabler, referanser og objekter
- Kort om None og self
- Datastrukturtegning (merk: del av obligen)
- Individuell jobbing

Forskjellen på instansvariabler, lokale variabler og parametre

- **Instansvariabler** – kan aksesseres i hele klassen (på tvers av metoder)
- **Lokale variabler** – kan bare aksesseres i egen metoden, oppstår og "dør" hver gang metoden kalles (finnes kun mens metoden utføres)
- **Parametre** – oppfører seg som lokale variabler men initialiseres med argumentene i metodekallet (får verdi "utenifra")

```python
class Bankkonto:
    def __init__(self):
        self._saldo = 0

    def betale_leie(self, strom):
        leie = 8000 + strom
        self._saldo -= leie
```

Hva i betale_leie() er instansvariabel, lokal variabel og parameter?

```
class Bankkonto:
    def __init__(self):
        self._saldo = 0

    def betale_leie(self, strom):
        leie = 8000 + strom
        self._saldo -= leie
```

instansvariabel lokal variabel parameter

Forskjellen på variabel, referanse og objekt

- *Variabler* inneholder *referanser* til *objekter*

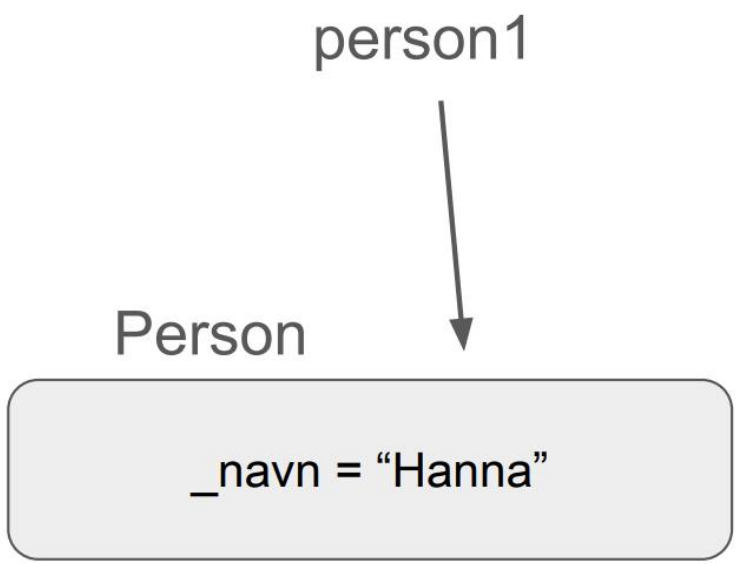
- **Variabel** – merkelapp vi setter på en verdi
- **Referanse** – hvor verdien ligger i minnet
- **Objekt** – selve verdien

Når endrer vi variabel og når endrer vi objekt?

```
person1 = Person("Hanna")
person2 = Person("Erik")

person1.endre_navn("Hanne") # Endrer objektet
person1 = person2           # Endrer variabelen
```

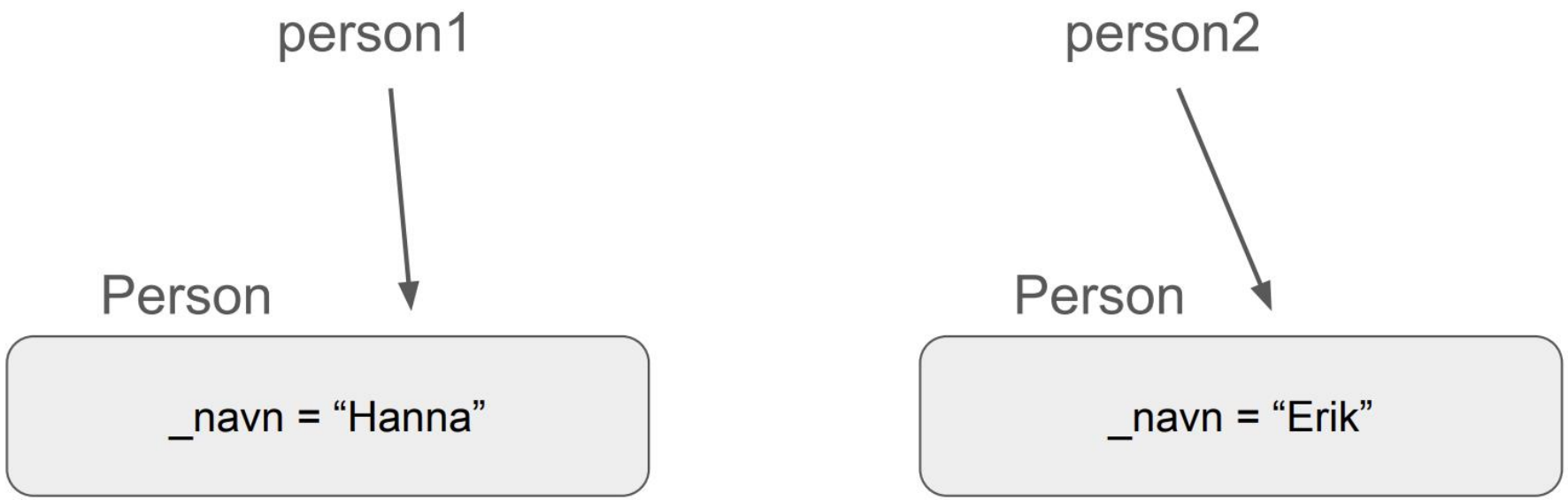

person1 = Person("Hanna")
person2 = Person("Erik")
person1.endre_navn("Hanne") # Endrer objektet
person1 = person2 # Endrer variabelen


person1
Person
_navn = “Hanna”



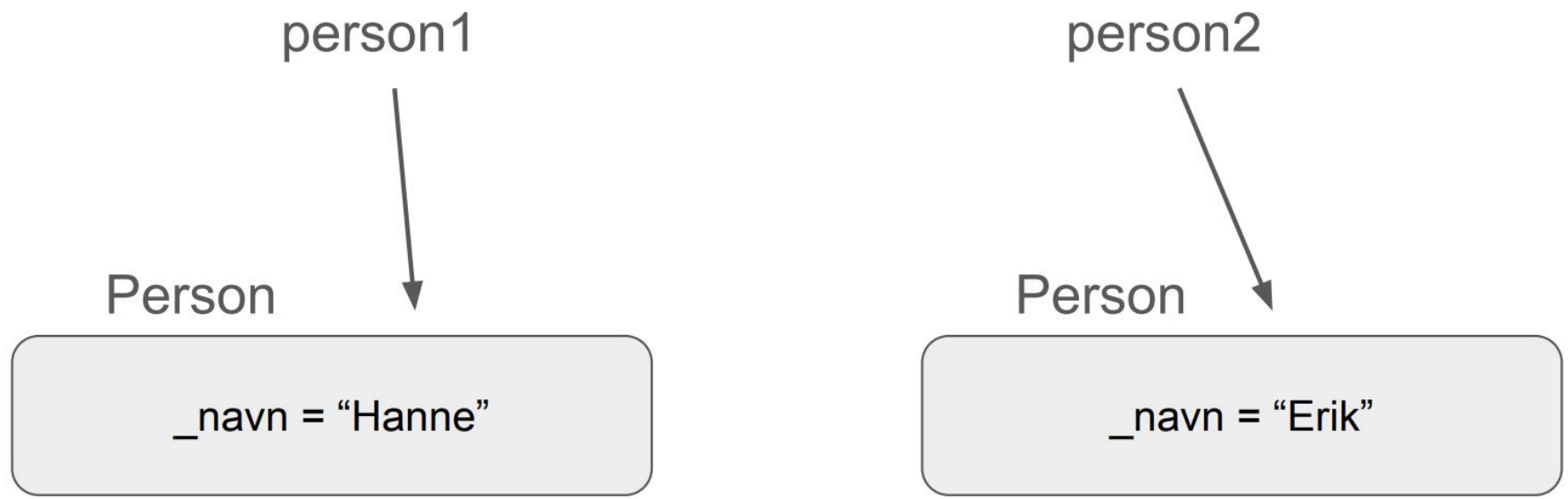
```
person1 = Person("Hanna")
person2 = Person("Erik")

person1.endre_navn("Hanne") # Endrer objektet
person1 = person2           # Endrer variabelen
```



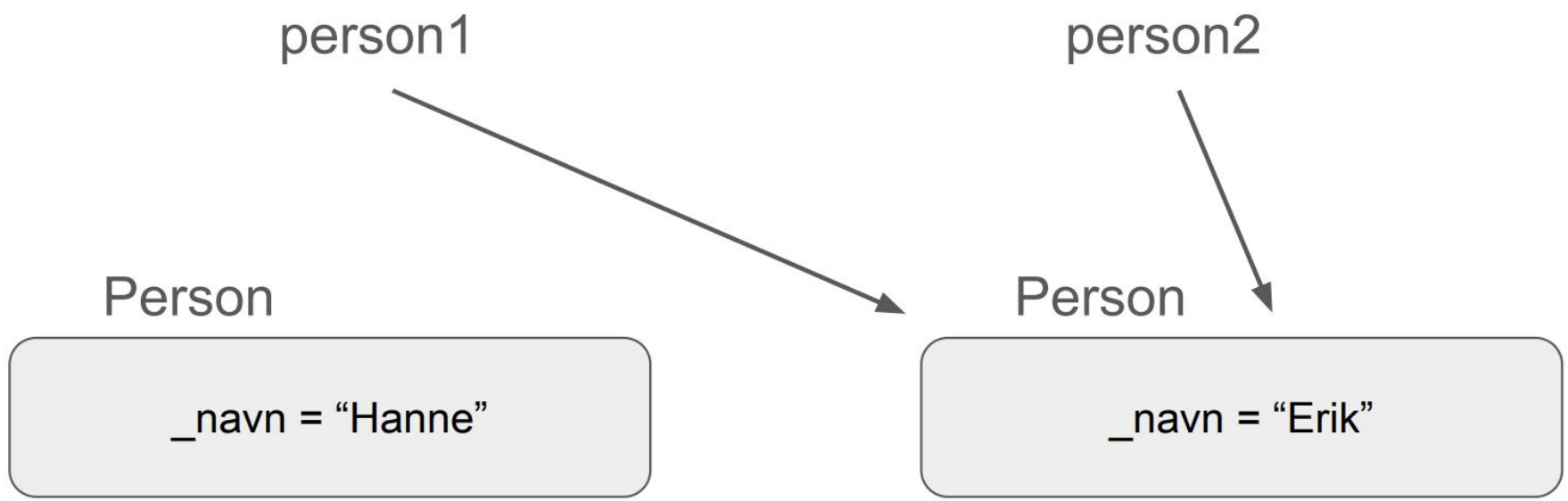
```python
person1 = Person("Hanna")
person2 = Person("Erik")

person1.endre_navn("Hanne") # Endrer objektet
person1 = person2           # Endrer variabelen
```

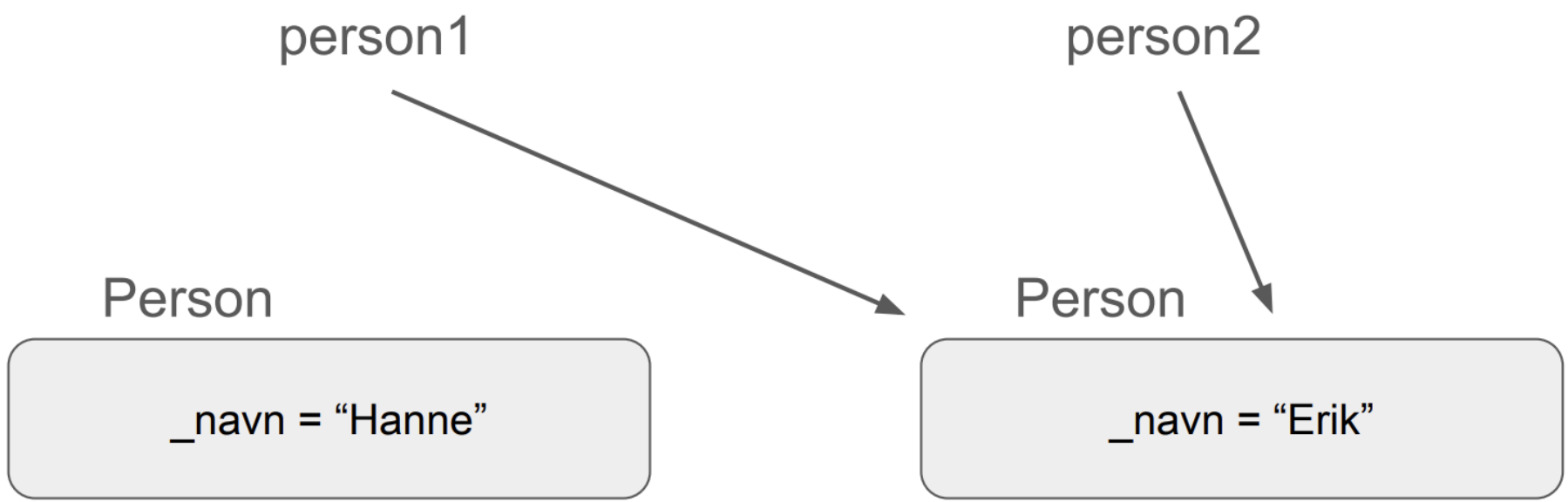


```python
person1 = Person("Hanna")
person2 = Person("Erik")

person1.endre_navn("Hanne") # Endrer objektet
person1 = person2           # Endrer variabelen
```



Merk: vi får en kopi av *referansen*, ikke *objektet*! person1 og person2 peker på *samme* objekt, ikke to kopier. Hvis vi endrer objektet person1 refererer til vil endringen også gjelde person2 (ettersom det er samme objekt, bare to variabler til det samme)



- **None** er en verdi som sier at variabelen ikke holder på noe objekt for øyeblikket
- **self** brukes for å peke på det nåværende objektet. Vi trenger ikke å sende det inn som argument når vi kaller på metoder fordi det automatisk blir sendt inn når vi kaller metoden på gjeldende objekt

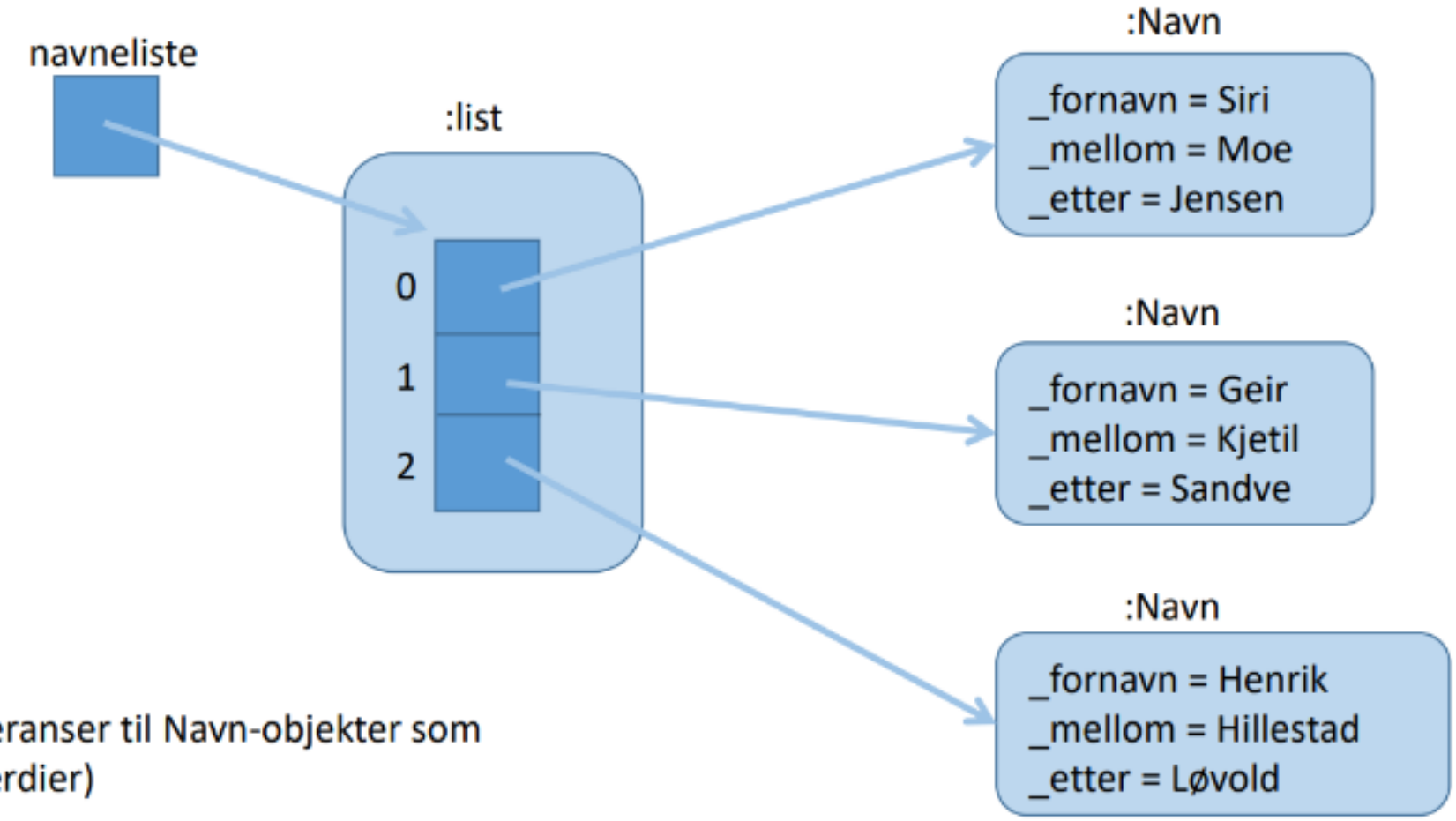
- Also: Objekter kan ha instansvariabler med referanse til andre objekter!

Datastrukturtegning

- **Variabler** – rektangel med verdi inni og variabelnavn over
- **Objekter** – rektangel med runde kanter, med instansvariabler inni og klassen (typen) over med kolon foran
- **Instansvariabler** – inni objektet, enten boks/rektangel eller <_instansvariabel> = <verdi>
- **Referanser** – pil fra variabel til objekt

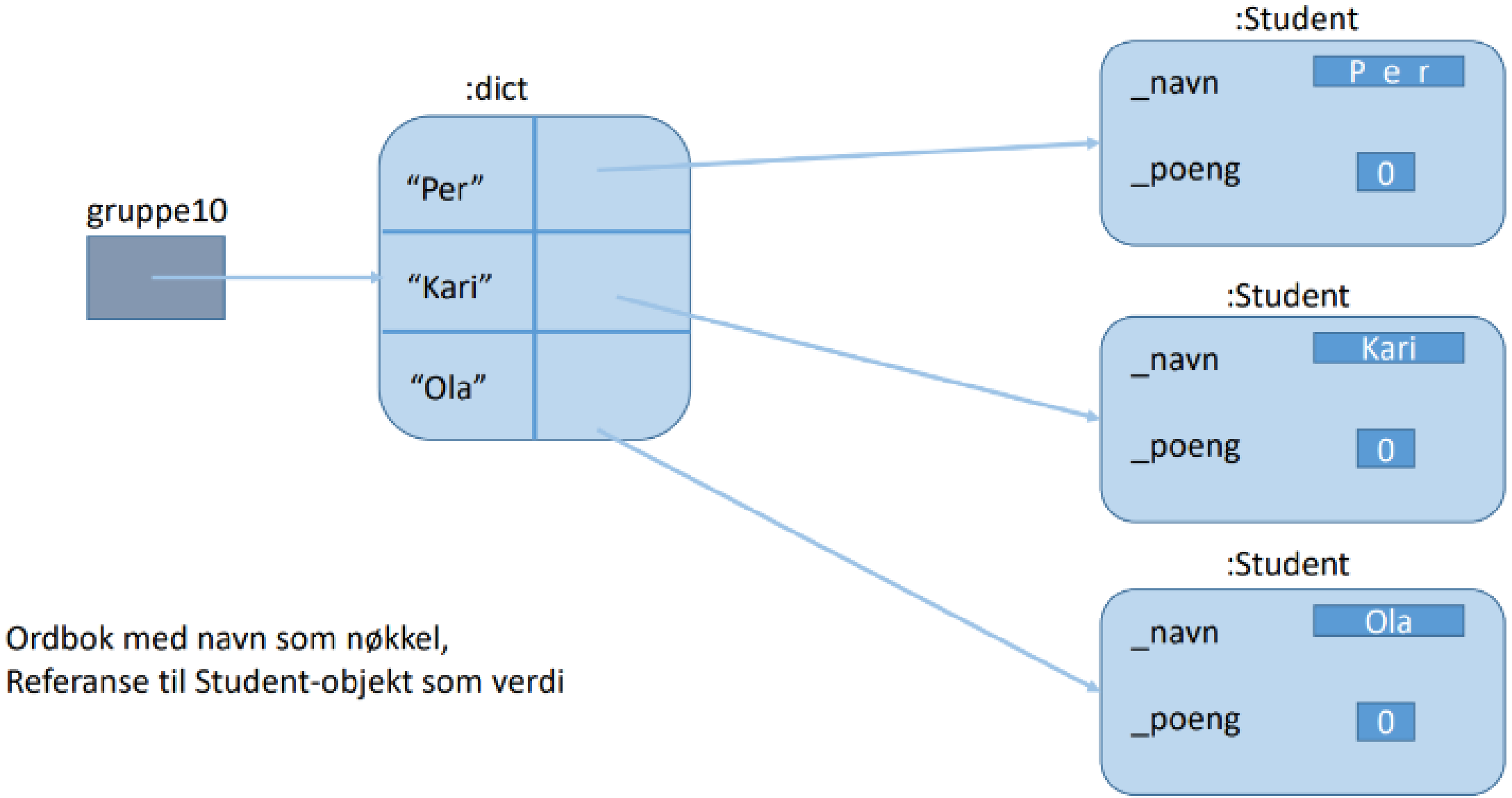
Datastrukturtegninger

Visualisering - liste



Liste med referanser til Navn-objekter som elementer (verdier)

Visualisering - ordbok



Ordbok med navn som nøkkel,
Referanse til Student-objekt som verdi

Visualisering av komplekse datastrukturer

