



Seminartime uke 8 – Gr. 10

IN1000 – HØST 2021

Hvor langt har vi kommet?

	Legges ut ca.	Frist
7. Innlevering OBLIGATORISK	13. oktober	26. oktober kl 23.59
8. Innlevering OBLIGATORISK	27. oktober	9. november kl 23.59

Uke	Tema
8	Mer objektorientert programmering <ul style="list-style-type: none">- (Repetisjon objekter og klasser)- Referanser og objekter- Uforanderlige typer (immutable data types)
9	Mer komplekse strukturer <ul style="list-style-type: none">- Objekter av flere klasser- Magiske metoder- Referanser mellom objekter
10	Mer komplekse strukturer <ul style="list-style-type: none">- Egne beholdere for objekter- Mer komplekse datastrukturer med objekter
11	Problemløsning med klasser <ul style="list-style-type: none">- Større eksempler- Løse problemer ved bruk av flere klasser
12	Problemløsning + prøveeksamen?
13+	Repetisjon og øve til eksamen ?

Plan for i dag

Del 1

- Repetisjon av klasser og objekter (set- og get-metoder)
- Formuleringer i oppgaver
- Referanser (minnegaten ?)
- Verdien None
- Datastrukturer – tegninger

Del 2

- Egen jobbing/oppgaver



Repetisjon - klasser

Rep: klasser

- **En klasse** = en beskrivelse eller oppskrift på hvordan noe skal være (f.eks hund, møbel, bygning...)
- **Et objekt** av en klasse = et eksemplar av en klasse, har de samme egenskapene som er definert i den klassen
- **En instans** av en klasse = samme som et objekt av en klasse
- En klasse **består av**: konstruktør, 0 eller flere instansvariabler, 0 eller flere metoder
- **Innkapslingen** av en klasse gjør det sikkert å bruke en klasse uten å ødelegge den.
- Får tak i/modifiserer instansvariabler gjennom **get- og set-metoder**

HUSK! Bruk set- og get-metoder

```
class Hund:

    def __init__(self, navn, alder, vekt, rase):
        self._navn = navn
        self._alder = alder
        self._vekt = vekt
        self._rase = rase
        self._favorittmat = []

    def bjeff(self):
        print("Bjeff!")

    def hentAlder(self):
        return self._alder

    def leggTilMat(self, mat):
        self._favorittmat.append(mat)
```

Get-metode

Set-metode

- Innkapsling, et konsept dere vil møte flere ganger, inkludert i funksjonell programmering og web-design (selv når OOP ikke brukes)
- Get- og set-metoder brukes for å unngå direkte aksessering av instansvariabler → not safe!
- Get-metoder er metoder som *returnerer* instansvariabler → å «få tak i noe», to «get something»
- Set-metoder er metoder som *endrer* på instansvariabler → å «sette noe», to «set something»

Formuleringer i oppgaver

- Spør brukeren = input
- Tar inn = tar inn som parameter/argumenter
- Leser inn = tar inn som parameter/argumenter fra f.eks fil
- Sender inn = sender inn som parameter/argument
- Gir ut = returnere
- Gir tilbake = returnere
- Skriver ut = print



References

Referanser

Referanser i programmering

- Hva mener vi med referanser (ikke jobb-referanser)?
- Tidligere: en variabel er som en boks som inneholder verdien
- Nå: en variabel inneholder *adressen* til der verdien ligger

Referanser i programmering

Et illustrerende eksempel



```
class Hund:  
    def __init__(self):  
        self._favorittmat = []
```



Fido!



Fodi!



#Variabelen fido peker på en instans klassen Hund

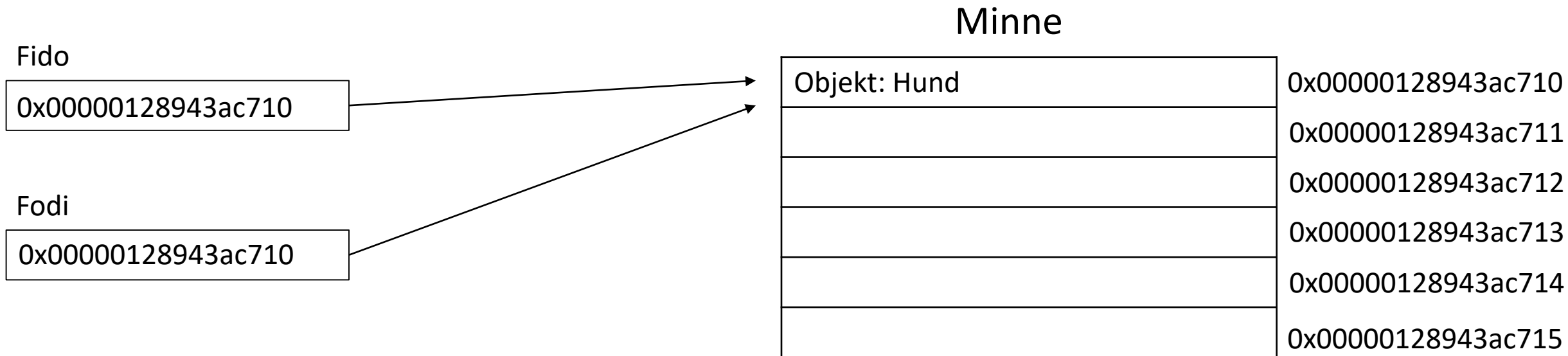
```
fido = Hund()
```

#Lager ny variabel som peker på samme Hund-objekt som fido

```
fodi = fido
```

Referanser i programmering

- Hva mener vi med referanser (ikke jobb-referanser)?
- Tidligere: en variabel er som en boks som inneholder verdien
- Nå: en variabel inneholder *adressen* til der verdien ligger





Minnegata

Kjapt om «None»

- «None» brukes til å representere fraværet av verdi
- En variabel som peker på «None» peker ikke på noe
- Kan brukes når vi vil ha en variabel som ikke trenger å peke på noe før senere
- Kan brukes i if-sjekker for å sjekke om vi faktisk har en verdi eller ikke
→ sjekke om variabelen fortsatt holder på et objekt!
- Bruk og fallgruver med None og instansmetoder





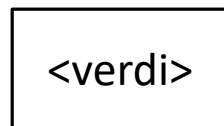
STRUCTURES

Strukturtegninger

Strukturtegninger i IN1000

Variabler med de «vanlige» typene (string, int, boolean....)

<variabelnavn>

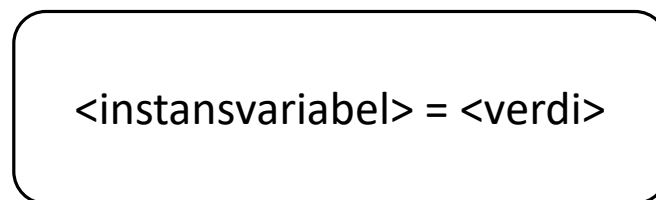


alder

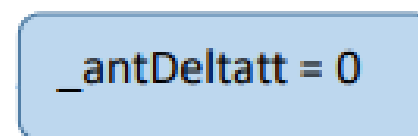


Objekter av (egne) mutable klasser

:<klassenavn>



: Student

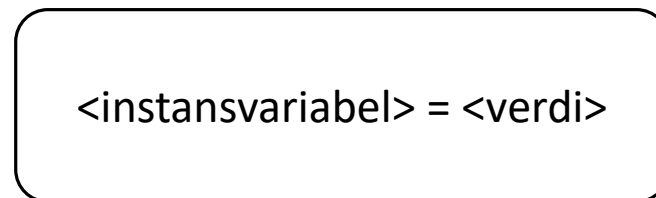


Variabler som refererer til objekter

<variabelnavn>



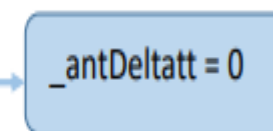
:<klassenavn>



nyDeltaker



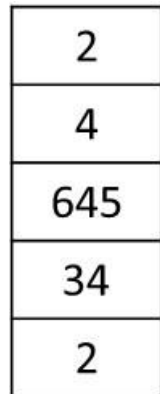
: Student



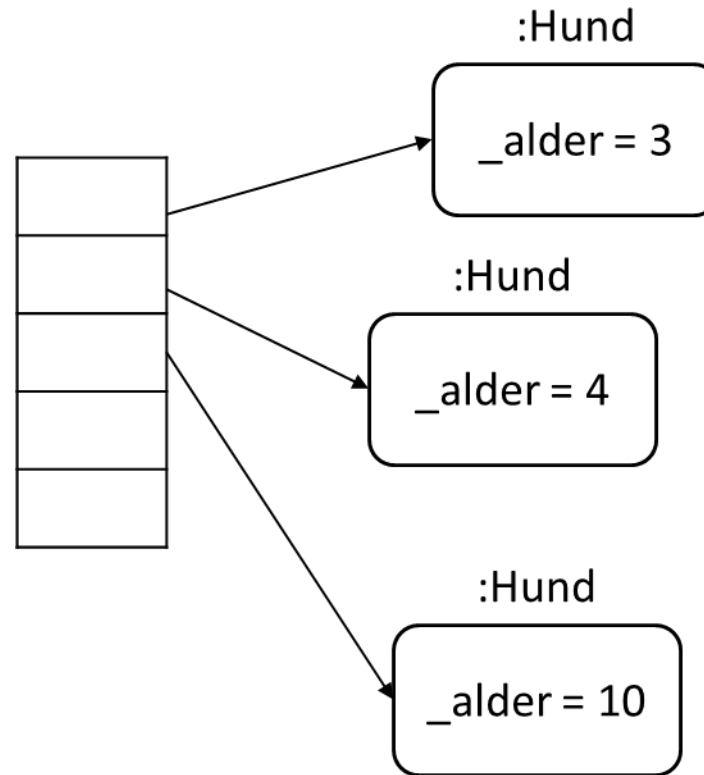
Strukturtegninger i IN1000 forts.

Lister med «vanlige»
datatyper (string, int,
boolean....)

ny_liste



Lister med objekter



Variabel som peker på lister med objekter

ny_liste

