



Seminartime uke 7 – Gr. 10

IN1000 – HØST 2021

Hvor langt har vi kommet?

NB: Endringer kan komme	Legges ut ca.	Frist
1.innlevering (3 poeng)	25. august kl 13.00	31. august kl 23.59
2. innlevering (5 poeng):	1. september kl 13.00	7. september kl 23.59
3.innlevering (5 poeng):	08. september kl 13.00	14. september kl 23.59
4.innlevering (5 poeng):	15. september kl 13.00	21. september kl 23.59
5.innlevering (5 poeng):	22. september kl 13.00	5. oktober kl 23.59
6. innlevering (6 poeng):	6. oktober kl 13.00	12. oktober kl 23.59

Uke	Tema
6	Repetisjon/pust i bakken
7	Objektorientert programmering <ul style="list-style-type: none">- Objekter- Klasser
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">- Samling av objekter- Magiske metoder- Komplekse strukturer
10	?
11	?
12	Problemløsning + prøveeksamen?
13+	Repetisjon og øve til eksamen ?

Plan for i dag

- Introduksjon til objektorientert programmering (oop)
- Noen begreper
- Syntaks: klasser og objekter
- Grensesnitt og implementasjon
- Større oppgave i par





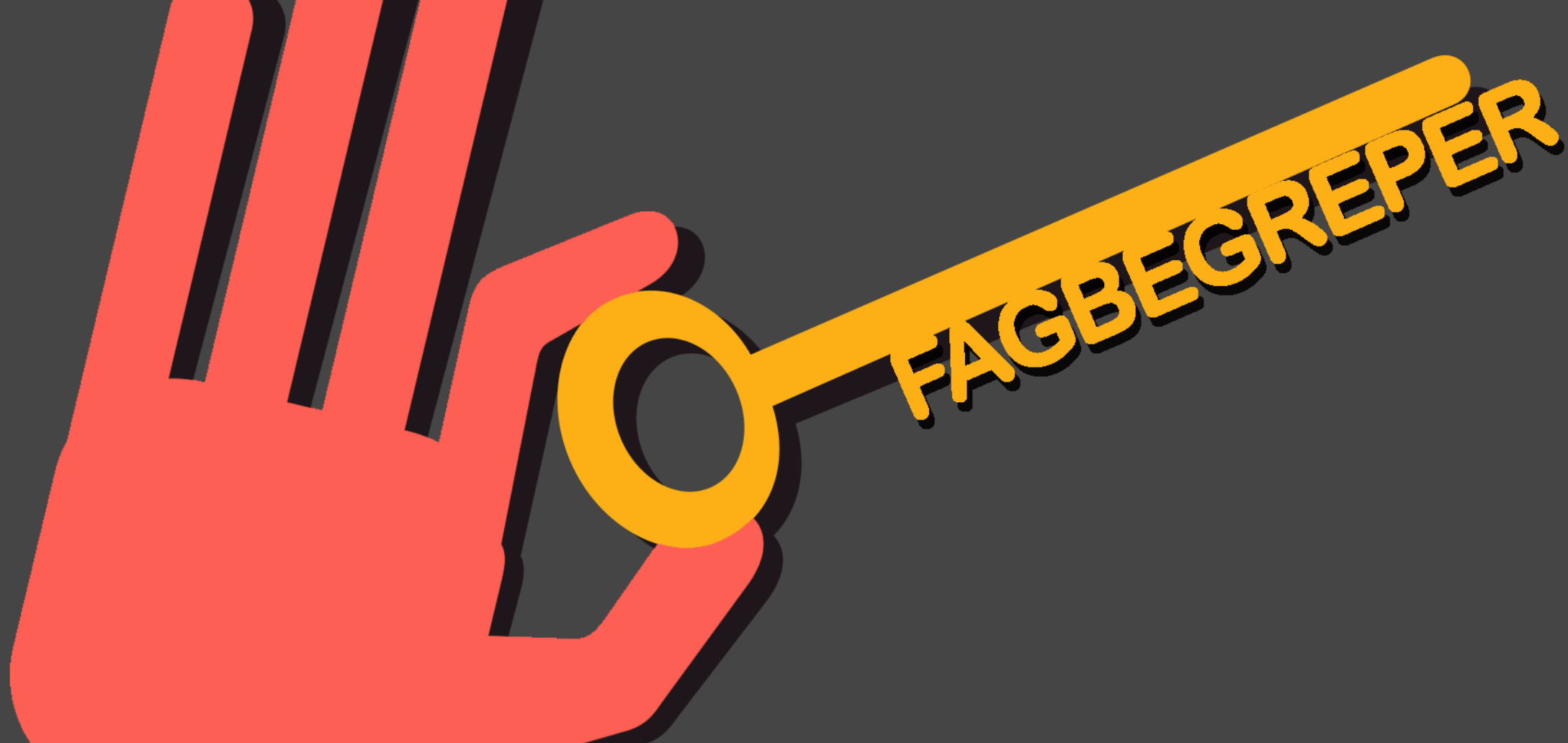
Objektorientert programming (OOP)

Hva er objektorientert progging?

- Programmeringsmetode eller et paradigme innen programmering
- Konsept: representerer «virkeligheten» ifa klasser og objekter (modellering)
- I virkeligheten finner vi mange mønstre og kjennetegn på ting, tar dette med videre inn i programmeringen
- Innkapsler verdier og variabler, ingen «fri flyt» eller «ukontrollert» bruk

Illustrerende eksempel: hunder

I virkeligheten:	I OOP:
 <p>Vekt</p> <p>Alder</p> <p>Rase</p> <p>Farge</p>	 <p>6 kg</p> <p>1 år</p> <p>Border collie</p> <p>Rødbrun</p> <hr/> <p>Klassenavn: hund</p> <p>Egenskaper/attributter: vekt alder rase farge</p> <hr/> <p>Instans: klasse: hund vekt: 6 alder: 1 rase: Border collie farge: Rødbrun</p>



BEGREPER DERE MÅ LÆRE

Begreper dere må kunne (til nå)

Funksjoner	Klasse
Prosedyre	Objekt
Parameter/argumenter	Instans/instansvariabel
Retur-verdi	Konstruktør (self)
Iterere	Grensesnitt/interface
Kodeflyt/scope	Metode/instansmetode
Innkapsling	Metodekall
Evaluerer (kode)	Representasjon
Implementere/implementasjon	Dokumentasjon
Syntaks	Indeks
Modellering	Attributt
Tilordning/verditilordning	Global vs lokal

Coding

Lorem ipsum dolor sit amet, onsec.
Nulla vitae tincidunt dul. Phasellus elit
sit suspendise ac nis. Pelentesque hab
itant malesuda fames turis egestas.
Duis tincidunt celerisque, sevelitus
vitae tincidunt dul.

SYNTAKS: klasser og objekter

Nærmere om klasser og objekter(instanser)

KLASSER

- Kategorier med attributter (egenskaper) og metoder
- Eksempler: dyr, møbel, frukt, person, bygning osv...
- Klasser dere kjenner til fra før:
 - ✓ String
 - ✓ Int
 - ✓ Float
 - ✓ Boolean
- Metoder dere kjenner til:
 - ✓ `split()` → liste

OBJEKTER/INSTANSER

- Ting som tilhører kategorier. Et «eksemplar» av en klasse
- Eksempler: katt, stol, eple, Donald Trump, hotell osv...
- Instanser av kjente klasser:
 - ✓ Navn = «Donald Trump»
 - ✓ Alder = 3
 - ✓ Pi = 3.14
 - ✓ Regn = True
- Metodekall på instansene:
 - ✓ `navn.split()` → ["Donald", "Trump"]

Syntaks: klasser

```
person.py
1  class Person:
2
3      def __init__(self, navn, alder):
4          self._navn = navn
5          self._alder = alder
6          self._kjonn = "Kvinne"
7          self._barn = []
8
9
10     def burdag(self):
11         self._alder += 1
12
13
14     def hentNavn(self):
15         return self._navn
16
17     def faaBarn(self, barn):
18         self._barn.append(barn)
19
```

Forhåndsdefinerte

- Konstruktøren: `__init__(self)` → her ligger alle attributtene
- «self» er med i alt
- Endrer på instansvariablene (attributtene til objektet)
- Returnerer instansvariablene (attributtene til objektet)
- Klassenavnet har STOR forbokstav
- Klassen er definert (laget) i en egen fil

Syntaks: objekter og metodekall

```
hovedprogram.py
1 from person import Person
2
3 def hovedprog():
4     #Oppretter en instans av Person
5     marie = Person("Marie", 42)
6     #Kaller på metoden burdag() som endrer på hennes instansvariabel
7     marie.burdag()
8     #Kaller på metode som legger "Elias" til i listen hennes
9     marie.faaBarn("Elias")
10
11 hovedprog()
```

```
person.py
1 class Person:
2     def __init__(self, navn, alder):
3         self._navn = navn
4         self._alder = alder
5         self._kjonn = "Kvinne"
6         self._barn = []
7
8     def burdag(self):
9         self._alder += 1
10
11     def faaBarn(self, barn):
12         self._barn.append(barn)
```

- Objektene opprettes i en *annen* fil enn der klassen er definert
- Sender inn nødvendige parametre (som er definert i konstruktøren til klassen)
- Kaller på metodene som er definert i klassen (slik som vi gjorde med `navn.split()`)
- Instansvariablene til objektet endrer seg



Større oppgave i par

Steg 1 (lett)

Bestem dere for hvem som skal være person A og person B

Person A	Person B
<ul style="list-style-type: none">- lag en klasse HUND- Attributter:<ul style="list-style-type: none">• Navn• Alder• Vekt• Rase• Favorittmat[]- Metoder:<ul style="list-style-type: none">• bjeff() → skriv ut «Bjeff!»• hentAlder() → returnerer alder• hentNavn() → returnerer navn• hentDyreart() → returnerer «Hund»• leggTilMat() → legg til i «favorittmat»• skrivUtFavoritter() → skriver ut lista	<ul style="list-style-type: none">- lag en klasse KATT- Attributter:<ul style="list-style-type: none">• Navn• Alder• Vekt• Farge• Favorittsted[]- Metoder:<ul style="list-style-type: none">• mjau() → skriv ut «Mjau»• hentAlder() → returnerer alder• hentNavn() → returnerer navn• hentDyreart() → returnerer «Hund»• leggTilSted() → legg til i «favorittsted»• skrivUtFavoritter() → skriver ut lista

Steg 2 (lett)

1. Send koden av klassen deres til den andre personen.
2. Lag en ny (tom) fil der dere importerer klassen til både hund og katt

```
from <filnavn> import <Klasse>  
from hund import Hund  
from katt import Katt
```

3. Dere skal nå bruke klassen til den andre personen for å gjøre følgende:
 - Opprett 3 instanser/objekter av den klassen
 - Skriv ut navn og alder (i en fin string) til alle 3 objektene
 - Legg til 2 favorittsted/-mat til objekt nr 2 og objekt 3
 - Skriv ut favorittene til objekt nr 2

Steg 3 (medium)

Denne delen skal løses sammen (både person A og B).

- I en ny fil: lag en ny klasse «shelter» som skal ha en tom liste som attributt
- Metodene denne klassen skal ha er:
 - leggTilDyr() → legger til et dyreobjekt (hund eller katt)
 - fjernDyrForst() → fjerner det første dyret som ble lagt til
 - fjernYngstSiste() → fjerner siste dyret som ble lagt til
 - returnListe() → returnerer lista til objektet (attributtet)

Steg 4 (vanskelig!)

Denne delen skal løses sammen (både person A og B).

- I fila fra steg 2: importer klassen Shelter
- Opprett en instans av klassen Shelter (husk å lagre i variabel)
- Lag en for-løkke som kjører et tilfeldig antall ganger mellom 1 og 15
- Når telleren er oddetall, opprett en hunde-instans og legg til i shelter-instansen
- Når telleren er partall, lag og legg til en katte-instans
- Skriv så ut følgende (print):
 - Shelter-instansen (hvorfør ser utskriften slik ut?)
 - Lista til shelter-instansen (hvilke metode må dere bruke?)
 - Hvert objekt i listen til shelterinstansen (er utskriften som forventet?)
 - Dyrearten til hvert objekt i listen til shelterinstansen (får du «hund»/ «katt»?)