

INFO132 Laboppgaver 13

Del 1 - funksjonell programmering

1. Flervalgsoppgaver

Hva blir utskriften?

a)

```
l=[1,2,3,4,5,6,7,8,9]
r=list(map(str,l))
print(r)
```

- i. '123456789'
- ii. '[1,2,3,4,5,6,7,8,9]'
- iii. ['1', '2', '3', '4', '5', '6', '7', '8', '9']
- iv. []

b)

```
l=[1,2,3,4,5,6,7,8,9]
r=list(filter(lambda x:x>5, l))
print(r)
```

- i. feilmelding
- ii. [1, 2, 3, 4]
- iii. [5]
- iv. [6, 7, 8, 9]

c)

```
l=[1,2,3,4,5,6,7,8,9]
r=list(map(lambda x:x-1-len(l),l))
print(r)
```

- i. [-9, -8, -7, -6, -5, -4, -3, -2, -1]
- ii. [-8, -7, -6, -5, -4, -3, -2, -1]
- iii. [-9, -8, -7, -6, -5, -4, -3, -2]
- iv. [1, 2, 3, 4, 5, 6, 7, 8, 9]

d)

```
l=[1,2,3,4,5,6,7,8,9]
r=list(filter(lambda x: x%2==0, map(lambda x:x+3, l)))
print(r)
```

- i. [4, 6, 8, 10, 12]
- ii. [1, 3, 5, 7, 9]
- iii. [5, 7, 9, 11]
- iv. [4, 5, 6, 7, 8, 9, 10, 11, 12]

```
e)
r=[x+1 for x in range(0,9) if 7>x>3]
print(r)
```

```
i.    [3, 4, 5, 6, 7]
ii.   [4, 5, 6]
iii.  [5, 6, 7]
iv.   [6, 4, 5]
```

```
f)
l=[1,2,3,4,5,6,7,8,9]
r=[x+x for x in map(str, l) if x>'5']
print(r)
```

```
i.    [6, 8, 10, 12, 14, 16, 18]
ii.   ['66', '77', '88', '99']
iii.  ['12', '14', '16', '18']
iv.   TypeError: unorderable types: int() > str()
```

2.

I det følgende er `l=[1,2,3,4,5,6,7,8,9]`

Hva er

```
i.    [x for x in l if x%2==0]
ii.   [(lambda v:v+v)(x) for x in map(str,l)]
iii.  [(lambda v:v+v)(x) for x in map(str,l) if x>55]
iv.   [(lambda v:v+v)(x) for x in map(str,l) if x>'55']
v.    [(lambda x:x+x)(x) for x in map(str,l) if str(x)>'55']
vi.   [x==y for (x,y) in [(1,2),(9,8),(3,3),(7,6),(7,7)]]
```

(svar 'feilmelding' dersom du mener uttrykket har feil som avbryter eksekveringen)

3. Lag et uttrykk vha list-comprehension som tilsvarer

```
list(filter(partall, map(doble,L)))
```

(vi forutsetter at funksjonene `partall` og `doble` er definert)

4. oversett map/filter-uttrykkene i oppgave 1 a)-d) til list-comprehension

5.

Major Maren og General Gerd sender hverandre hemmelige meldinger flere ganger pr dag. De er redde for at meldingene kan bli fanget opp av fienden så de vil kryptere meldingene vha såkalt 'Cæsar skift kryptering'. Det går ut på at de er enige om en krypteringsnøkkel, i dette tilfellet et tall, og de har bestemt seg for at de alltid bruker dagens dato som nøkkel.

Med nøkkelverdi n vil 'Cæsar skift' kode en gitt bokstav om til bokstaven som ligger n plasser til høyre i alfabetet (Et utvidet alfabet med komma, mellomrom og andre tegn vi trenger for å skrive meldinger). Vi teller plasser syklisk slik at 'a' kommer etter 'å'.

Lag en funksjon `kryptering(nøkkel)` som gitt en nøkkelverdi returnerer et par bestående av en krypteringsfunksjon og en dekrypteringsfunksjon.

Maren og Gerd vil hver morgen kalle denne funksjonen med dagens dato som nøkkel og så kan de kryptere og dekryptere hele dagen vha de returnerte funksjonene.

Eksempel:

```
>>> (krypt,dekrypt)=kryptering(21)
>>> krypt('python')
'gpkåfe'
>>> dekrypt('gpkåfe')
'python'
>>>
```

Ekstra oppgaver

6.

Lag en funksjon `treSmåMusikanter(v)` som skriver ut verset Tre små Musikanter der vokalene er byttet ut med bokstaven `v`. Skriv verset normalt dersom det ikke oppgis noen bokstav.

(Tips: `map` fungerer like godt på tekststrenger som på lister)

```
>>> treSmåMusikanter()
tre små musikanter på høybro plass
sto og spilte på en kontrabass
så kom en konstabel, spurte hva var hendt
tre små musikanter på høybro plass
>>>
>>> treSmåMusikanter('i')
tri smi misikinter pi hiibri pliss
sti ig spilti pi in kintribiss
si kim in kinstibil, spirti hvi vir hindt
tri smi misikinter pi hiibri pliss
>>>
```

7. Lag en funksjon `evaluer(tekst)` som kan tolke talloperatorene `+`, `-`, `*` og `/` i dagligtale.

Eksempler:

```
>>> evaluer('5 pluss 7')
12
>>> evaluer('3 ganger 2')
6
>>> evaluer('3 multiplisert med 2')
6
>>> evaluer('9 delt på 2')
4.5
>>>
```

Del 2 rekursjon

1. Lag en rekursiv funksjon `snu` (*tekst*) som returnerer teksten baklengs.

```
>>> snu('hei')  
'ieh'
```

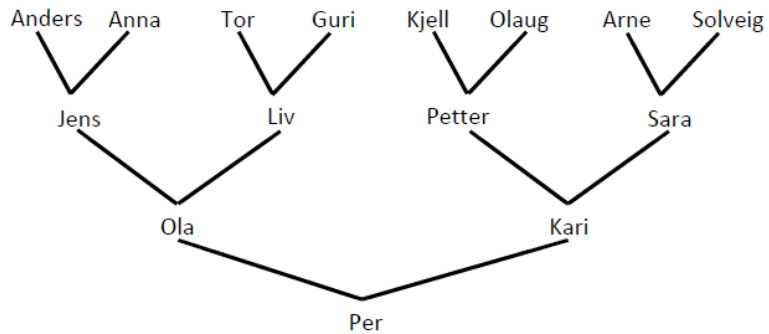
2. Gjensyn med fibonacci-rekken 1, 1, 2, 3, 5, 8, 13,... der hvert tall (bortsett fra de to første) er summen av de to foregående.

Lag en rekursiv-funksjon `fib` (*n*) som returnerer det *n*'te fibonacci-tallet
(Fibonacci-tallene nummereres fra 0 og oppover dvs `fib(0) == 1`, `fib(1) == 1` ... osv)

```
>>> for i in range(0,10):print(fib(i))  
  
1  
1  
2  
3  
5  
8  
13  
21  
34  
55
```

Ekstra oppgaver

3. På forelesningen så vi et eksempel på representasjon og utskrift av stamtrær. Gjenta dette, men bruk lister, i stedet for klasser, til å representere stamtrærne

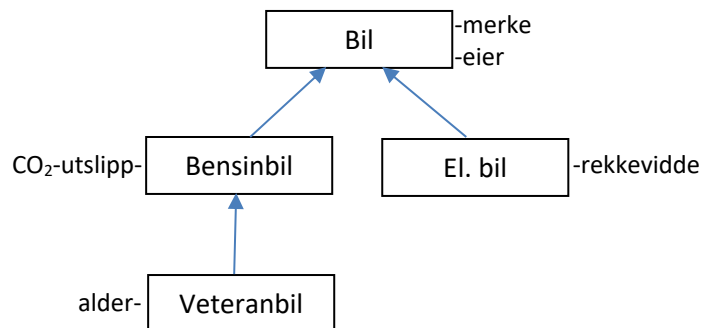


```
>>> skrivStamTre(per)
Per
-- Kari
---- Sara
----- Solveig
----- Arne
---- Petter
----- Olaug
----- Kjell
-- Ola
---- Liv
----- Guri
----- Tor
---- Jens
----- Anna
----- Anders
```

Del 3 Arv

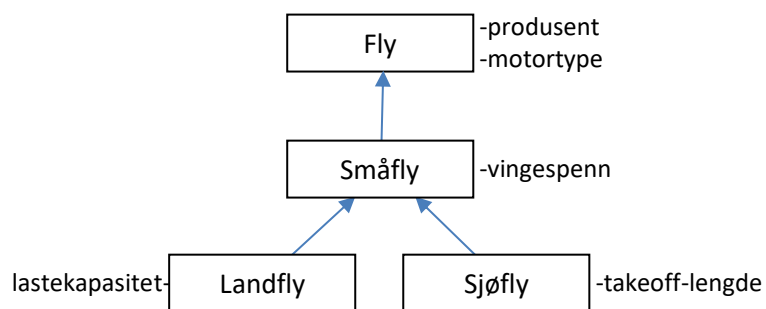
1. Lag klassehierarkier i for det følgende.

a)



Biler skal skrives ut med merke, eier og eventuelt alder eller rekkevidde.

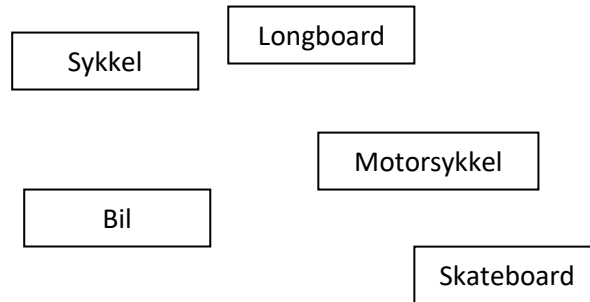
b)



Utskrift skal ha med produsent og eventuelt vingspenn, lastekapasitet eller takeoff-lengde

2.

Her er noen typer kjøretøy.



Lag et klassehierarki i Python for disse, basert på antallet hjul og hvorvidt kjøretøyet er motorisert. Topp-klassen skal altså være 'kjøretøy' og de øvrige skal organiseres i et hierarki under denne. Innfør ekstra klasser om det trengs

Klassifiser primært mhp hvorvidt kjøretøyet er motorisert og sekundært mhp antall hjul

Ta med passende attributter som markerer skiller mellom de ulike klassene. For eksempel

- Motoriserte kjøretøy har et antall hestekrefter, til forskjell fra ikke-motoriserte kjøretøy.
- Longboard er skateboard, forskjellen er de har lengre avstand mellom akslene.

Lag også metoder for å skrive ut ulike og for å hente spesifikke verdier. For eksempel antallet hestekrefter til en motorsykkel.

Ekstra oppgaver

3. Gjensyn med 'FrueMafiaen' fra Lab3:

FrueMafiaen er en kriminell organisasjon der kun kvinner kan være medlemmer. Mafiaen er strengt hierarkisk med *Presidenten* som øverste leder. På neste nivå finner vi *Kommandanter*, deretter *Kapteiner* så *Sersjanter* og nederst finner vi *Soldatene*. De som ikke er kvalifiserte til opptak, kan melde seg som *Støttemedlem*. Man kan stige i gradene, etter søknad, basert på ansiennitet og meritter. Vedtektene spesifiserer kvalifikasjonskravene til de ulike gradene:

- Minstekrav for alle grader, inkludert støttemedlem, er å være Kvinne, minst 20 år gammel og ha middels dårligandel.
- Soldater må være under 30 år og ikke ha barn
- Det samme gjelder for Sersjanter, som i tillegg må ha minst 4 års ansiennitet og ikke rent rulleblad.
- Kapteiner må være yngre enn 40 år, ha minst 6 års ansiennitet og ha tiltale eller straffedom på rullebladet.
- Kommandanter må også være yngre enn 40 år med minst 6 års ansiennitet, men må ha en straffedom på rullebladet samt ha temmelig dårligandel.
- Presidenten må oppfylle samme krav som kommandanter, men i tillegg ha minst 8 års ansiennitet. Dessuten kan ikke presidenten ha et fornavn som kan forveksles med et guttenavn. Dvs at kjønnsnøytrale navn som Kristen, Kim, Janne, og Tony må unngås.

Definer et klassehierarki for gradene. Ved instansiering skal kvalifikasjonskravene sjekkes og graden settes til 'ikke kvalifisert' dersom de ikke er oppfylt. Eksempler:

```
>>> liv=kaptein('Liv', 23, 4, 2, 'tiltalt', 8)
>>> kari=sersjant('Kari', 32, 4, 0, 'straffet',5)
>>> olga=president('Olga', 39, 2, 2, 'straffet',9)
>>> mari=kommandør('Mari', 36, 1,3, 'rent', 8)
>>>
>>> for p in [liv,kari,olga,mari]: p.skriv()

Liv er kaptein
Kari er ikke kvalifisert
Olga er president
Mari er ikke kvalifisert
>>>
```