

Realfagskonferanse 05.05.22

Programmering som et redskap i matematikk-faget

Carina Tidei
Heimdal Vgs/skolelaboratoriet
Kontakt: carti@trondelagfylke.no

Fra læreplan i 1T:

Mål for opplæringa er at eleven skal kunne:

- forklare forskjellen mellom en identitet, ei likning, et algebraisk uttrykk og en funksjon

Fra <<https://www.udir.no/lk20/mat09-01/kompetansemaal-og-vurdering/kv42>>, fritt oversatt til bokmål av Carina Tidei

Fra læreplan i 1P:

Mål for opplæringa er at eleven skal kunne:

- identifisere variable størrelser i ulike situasjoner og bruke det til utforskning og generalisering
- Tolke og bruke formler som gjelder samfunnsliv og arbeidsliv
- Tolke og bruke funksjoner i matematisk modellering og problemløsning,
- planlegge, utføre og presentere selvstendig arbeid knytt til modellering og funksjoner innenfor samfunnsfaglige tema

Fra <<https://www.udir.no/lk20/mat08-01/kompetansemaal-og-vurdering/kv31>>, fritt oversatt til bokmål av Carina Tidei

Funksjon, likning, identitet, uttrykk? Hva er forskjellen?

Oppgave:

Hva er $f(x) = 4$?

- | | | |
|-------------------------------------|---------------|---|
| <input checked="" type="checkbox"/> | En funksjon? | Umulig å avgjøre ved bruk av vanlig notasjon i matematikk |
| <input type="checkbox"/> | En identitet? | |
| <input checked="" type="checkbox"/> | Et uttrykk? | |
| <input checked="" type="checkbox"/> | En likning? | |

Oppgave:

Hva er

```
1 def f(x):  
2     return 4
```

- | | |
|-------------------------------------|---------------|
| <input checked="" type="checkbox"/> | En funksjon? |
| <input type="checkbox"/> | En identitet? |
| <input type="checkbox"/> | Et uttrykk? |
| <input type="checkbox"/> | En likning? |

Hva er

$f(x) == 4$

- | | |
|-------------------------------------|---|
| <input type="checkbox"/> | En funksjon? |
| <input checked="" type="checkbox"/> | En identitet? Eventuelt om $f(x)$ var definert fra før av til å være $f(x) = 4$. |
| <input type="checkbox"/> | Et uttrykk? |
| <input checked="" type="checkbox"/> | En likning? |

Ved å bruke programmering, MÅ det være en tydelig forskjell som tvinger elevene til å tenke.

Funksjon: Bruk av def kommandoen

Uttrykk: et matematisk uttrykk som bruker tall og/eller variabler, gjerne brukt til å definere en ny variabel

Likning: Bruk av ==

Identitet: Likning som er alltid sant

Opgave:

Lag et program som sjekker alle heltallige løsninger ved prøv/feil metode til å finne alle løsningene til likningen $2(x+3)-8=x$ i intervallet $[-5,5]$

Vi starter alltid med en pseudo-kode der vi skriver med ord/setninger det vi må gjøre i riktig rekkefølge.

```
1 '''
2 Vi må definere en variabel x, med startverdi -5
3 Vi må sammenligne 2(x-3)+ 8 og x for alle hele tallene mellom -5 og 5
4 Vi må bruke en løkke så lenge x blir mindre enn 5
5
6
7 Hvis uttrykene er like stor, print til brukeren at det er en løsning
8 Ellers skriver vi at det ikke er en løsning
9 Vi må øke x-verdien med 1 for hver gang løkken kjører
10 '''
11
12 x = -5
13 while x <= 5:
14     if 2*(x-3)+8 == x:
15         print(x, ' er en løsning')
16     else:
17         print(x, ' er ikke en løsning')
18     x = x+1
```

Vi får følgende output:

Running: test.py

```
-5 er ikke en løsning
-4 er ikke en løsning
-3 er ikke en løsning
-2 er en løsning
-1 er ikke en løsning
0 er ikke en løsning
1 er ikke en løsning
2 er ikke en løsning
3 er ikke en løsning
4 er ikke en løsning
5 er ikke en løsning
```

Lag et program som bruker prøv/feil metode til å finne alle heltallige løsninger til likningen $2(x+4)-8 = 2x$ i intervallet $[-5,5]$

```
1 '''
2 Vi må definere en variabel x, med startverdi -5
3 Vi må sammenligne 2(x-3)+ 8 og x for alle hele tallene mellom -5 og 5
4 Vi må bruke en løkke så lenge x blir mindre enn 5
5
6
7 Hvis uttrykene er like stor, print til brukeren at det er en løsning
8 Ellers skriver vi at det ikke er en løsning
9 Vi må øke x-verdien med 1 for hver gang løkken kjører
10 '''
11
12 x = -5
13 while x <= 5:
14     if 2*(x-4)+8 == 2*x:
15         print(x, ' er en løsning')
16     else:
17         print(x, ' er ikke en løsning')
18     x = x+1
```

Vi får da følgende output:

Running: test.py

```
-5 er en løsning
-4 er en løsning
-3 er en løsning
-2 er en løsning
-1 er en løsning
0 er en løsning
1 er en løsning
2 er en løsning
3 er en løsning
4 er en løsning
5 er en løsning
```

Hva blir resultatene her?

Er $2(x+3)-8=x$:

- En likning? Det var bare én løsning
- En identitet?

Er $2(x+4)-8=2x$:

- En likning?
- En identitet? Alle hele tallene var løsninger, litt algebra viser at det er en identitet.

Bruk av programmering/algoritmisk tankegang som vei til å lage en formel.

Opgaven:

~~Lag en formel~~ Lag en algoritme/pseudokode for å finne ut prisen for drosjetur gitt kjøreavstand og kjøretid for drosjetur med trøndertaxi på hverdagen.

Takstopplysning for aktuell dato og tidspunkt

Takstnavn		Tidspunkt		
T1 - Dagtid hverdag		Man-Fre 09-15		

Startpris	Kr pr km	Kr pr min	Minstepris	Jamførpris *
kr 47	kr 12,43	kr 9,83	kr 135	kr 282

<https://www.trondertaxi.no/Side?kategori=priser&side=taksttabell>

```
1 '''
2 Vi må få inn fra brukeren antall km
3 Vi må få inn fra brukeren antall minutter for turen
4 Vi skriver inn startprisen,
5 Vi må skrive inn km prisen,
6 Vi må skrive inn prisen per minutt
7 Vi må skrive inn minsteprisen
8
9 Vi regner prisen for antall kjørte km
10 Vi regner prisen for brukt antall minutter
11 Vi regner ut prisen for hele turen
12
13 Vi må sjekke om den utregnet prisen er høyere enn minsteprisen eller ikke
14     Om ja, Skriver vi til brukeren at prisen for turen er utregnet prisen
15     Om nei, skriver vi til brukeren at prisen for turen er minsteprisen
16
17 '''
18
19 avstand = float(input('Hvor lange var turen i km? '))
20 minutter = float(input('Hvor lenge varte turen i minutter? '))
21
22 startpris = 47
23 kmpris = 12.43
24 minpris = 9.83
25 minstepris = 135
26
27 pris_avstand = avstand*kmpris
28 pris_minutter = minutter*minpris
29 prisen = pris_avstand + pris_minutter + startpris
30
31 if prisen > minstepris:
32     print('Prisen for turen er ', prisen, ' kr.')
33 else:
34     print('Prisen for turen er ', minstepris, ' kr.')
```

Her er løsningsforslaget jeg fikk sammen med elevene mine, men det er verdt å notere at andre elever fikk det til på en annen måte, blant annet:

- Ikke bruk variabler for startpris, pris/km, pris/min og minstepris, men bare bruke tallene
- Regne prisen i ett
- Ikke ta hensyn til minstepris i programmet, men bare tolket svaret i etterkant "for hånd"

Så det er lett å tilpasse oppgaven til nivået elevene har, både matematisk og algoritmisk.

Det er også mulig å lage en funksjon for å regne prisen for forskjellige taxiselskap i det samme programmet.

Funksjonen kan da se sånn ut:

```
def pris(startpris, kmpris, minpris, minstepris):
    pris_avstand = avstand*kmpris
    pris_minutter = minutter*minpris
    prisen = pris_avstand + pris_minutter + startpris
    if prisen < minstepris:
        prisen = minstepris
    return prisen
```

Den tar inn 4 argumenter og returnerer prisen. Da er det mulig å bruke funksjonen for flere taxiselskap i det samme programmet og er en fin utfordring for de som kan litt mer programmering.

Bruk programmering for å finne en modell, lettere inngang til modellering enn ved å finne funksjon med én gang

Oppgaven:

Lag et program til å løse oppgaven nedenfor

Tenk at du har et par kanin på en øy.

Hver 4 måneder, kan de få i snitt 4 unge som kan få barn selv etter 4 måneder. (Tenk at de får 2 hann- og 2 hunn- i snitt per kule)

Hvor mange kanin er det etter 4 år?

Her er det kanskje vanskeligere å se pseudokoden med én gang, så det er lurt å skrive det vi forventer for de første stegene for hånd:

Start: 2 kaniner

Etter 4 måneder: 2 kaniner fra før, 1 par-> 4 unger: 2+4=6 kaniner

Etter 8 måneder: 6 kaniner fra før, 6/2=3 par -> 3*4 unger= 12 unger: 6+12=18 kaniner

Etter 12 måneder: 18 kaniner fra før, 18/2=9 par -> 4*9=36 unger: 18+36=54 kaniner

Pseudokoden:

Vi må ha antall kanin som er der

Vi må finne ut antall par de er

Vi må regne ut antall unger de får

Vi må regne ut antall kaniner til neste runde

De stegene over må gjentas for alle 12 periodene

Vi må gi ut antall kaniner til slutt til brukeren.

```
kanin = 2
par = 1
unge = 0

# Vi legger inn at brukeren velger antall periodene
antall_perioder = int(input('Hvor mange perioder vil du regne for? '))

for i in range(antall_perioder):
    # Vi legger inn en linje slik at det skrives ut periodenr og antall kanin for hver stegene
    # med dette kan vi bruke deretter data for å bruke regresjonsanalyse på det.
    print(i, kanin)
    par = kanin/2
    unge = par*4
    kanin = kanin + unge

print('Til slutt er det', kanin, 'på øya')
```

Vi kan også legge inn i programmet noe som gjør at kaninene reproduseres ikke så fort etter hvert.

For eksempel at antall unge per par blir til 3 i snitt om de er over 10 000

Og så til 2 i snitt over 20 000

Og så til 1 i snitt over 30 000

Og så at det blir null om de er over 40 000 (like mange som fødes som de som dør)

Pseudokoden blir:

Vi må ha antall kanin som er der

Vi må finne ut antall par de er

Om antall kanin er under 10000-> da er antall unge gitt ved antall par*4

Ellers om antall kanin er under 20000-> da er antall unge gitt ved antall par*3

Ellers om antall kanin er under 30000-> da er antall unge gitt ved antall par*2

Ellers om antall kanin er under 40000-> da er antall unge gitt ved antall par*1

Ellers er antall unge er null

Vi må regne ut antall kaniner til neste runde ved å legge sammen antall kanin det var fra før av og antall unge.

Programmet blir da til:

```
kanin = 2
par = 1
unge = 0

# Vi legger inn at brukeren velger antall periodene
antall_perioder = int(input('Hvor mange perioder vil du regne for? '))

for i in range(antall_perioder):
    # Vi legger inn en linje slik at det skrives ut periodenr og antall kanin for hver stegene
    # med dette kan vi bruke deretter data for å bruke regresjonsanalyse på det.
    print(i, kanin)
    par = kanin/2
    # Vi legger inn betingelse på antall unge per par avhengig av populasjon
    if kanin <= 10000:
        unge = par*4
    elif kanin <= 20000:
        unge = par*3
    elif kanin <= 30000:
        unge = par*2
    elif kanin <= 40000:
        unge = par*1
    else:
        unge = 0
    kanin = kanin + unge

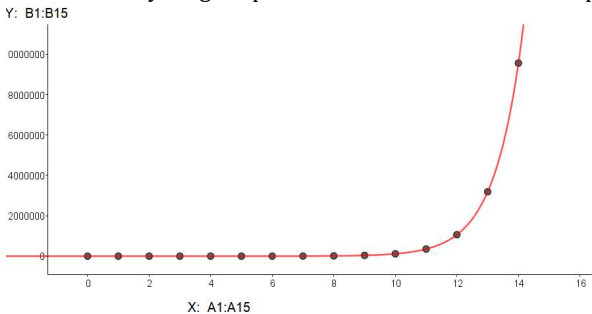
print('Til slutt er det', kanin, 'på øya')
```

For å bruke tallene som er gitt ut er det flere muligheter:

Kopier dem i excel-> Bruk excel til å gjøre om til 2 koloner (data -> tekst til koloner -> Følg steg for steg ved å si at det er mellomrom som er skilletegn

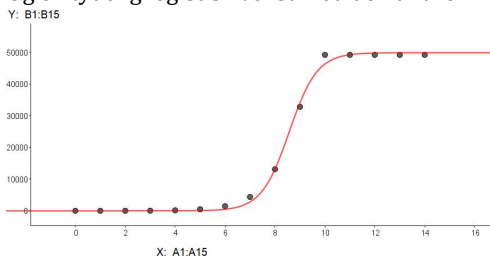
Kopier kolonene over inn i geogebra for å bruke regresjonsanalyse som vanlig.

Her får vi en tydelig eksponentiell vekst med den første programmet



Regresjonsmodell
Eksponentiell $y = 2 \cdot 3^x$

Og en tydelig logistisk vekst med den andre:



Regresjonsmodell
Logistisk $y = \frac{49876.52}{1 + 4732994.04 e^{-1.79x}}$

Opggaven:

Lag et program til å løse oppgaven nedenfor

Tenk at du sparer 100 kr per måned i en sparekonto som har en månedsrente på 0,1%. Hvor mye har du etter ett år?

```
1 '''
2 Vi må skrive inn startbeløpet
3 Vi må skrive inn sparingsbeløpet per måned
4 Vi må skrive inn rente i %
5 Vi må gjenta for de 12 måneder:
6     Regne renter fra forrige beløpet
7     legg sammen sparingsbeløpet med rente med det som var fra før av
8 Vi må skrive ut til brukeren beløpet etter 12 måneder
9 '''
10
11 start = 100
12 sparing = 100
13 rente = 0.1
14
15 for i in range(12):
16     renter = start*rente/100
17     start = start + renter + sparing
18
19 print('Etter ett år, er det: ', round(start, 2), ' kr på konto')
```

Utfordring

Endre programmet ovenfor slik at brukeren kan skrive inn for hver gang programmet kjøres:

- beløpet som skal spares per måned
- startbeløpet
- Renter
- Varighet på sparing gitt i antallet år.

```
1 '''
2 Vi må få inn startbeløpet
3 Vi må få inn sparingsbeløpet per måned
4 Vi må få inn rente i %
5 Vi må få inn antall år
6 Vi må gjenta for de 12*år måneder:
7     Regne renter fra forrige beløpet
8     legg sammen sparingsbeløpet med rente med det som var fra før av
9 Vi må skrive ut til brukeren beløpet etter 12 måneder
10 '''
11
12 start = float(input('Hvor mye legger du inn på sparekonto i starten? '))
13 sparing = float(input('Hvor mye vil du spare per måned? '))
14 rente = float(input('Hvor stor er månedsrente i prosent? '))
15 år = int(input('Hvor mange år skal du bruke for å spare? '))
16
17 for i in range(12*år):
18     renter = start*rente/100
19     start = start + renter + sparing
20
21 print('Etter ', år, ' år, er det: ', round(start, 2), ' kr på konto')
```