Et språk = syntaks + semantikk

Naturlig språk | *Programmeringsspråk*

SEMANTIKK — SYNTAKS | SYNTAKS — SEMANTIKK

Hele den store verden

vokabular:

"lovlige" ord
lege, en, jeg, er, som, han ...

identifikatorer, literaler, spesielle tegn
end, 0, sum, '0', "tast inn"...

fraser:

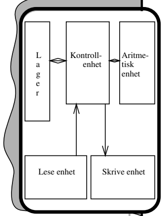
sammensatte fraser
en lege, som kom i går ...

uttrykk
sum+1, sum/antall, inreal ...

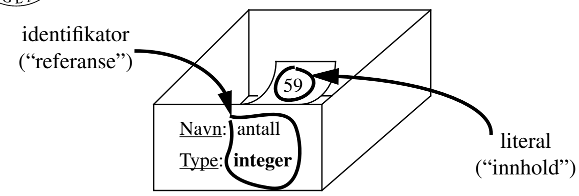
setninger:

kategorier: *substantiv, verb ...*
substantiv + verb
Jeg er, En lege går
Huset flyr
setningstyper:
påstand
ordre

kategorier: *variabel, uttrykk ...*
variabel := uttrykk
sum := sum+1
tall := inreal
imperativer:
tilordning (lager)
input/output
kontroll



Identifikatorer og literaler



Identifikator	*Literal*
et ord som begynner med en bokstav	**betegner en konstant verdi**

definerte: antall, sum, a1, a_b
– deklarerte i programmet

nøkkelord: **real**, **end**, **if, true**
– reserverte ord med fast betydning

predefinerte: **inimage**, **outtext**
– prosedyrer og funksjoner med en gitt betydning som kan forandres i programmet

heltall: 59 0 1993

reeltall: 3.14 .5 0.5 5&-1 $= 5*10^{-1}$

tegn: 'a' '0' '+' '5'

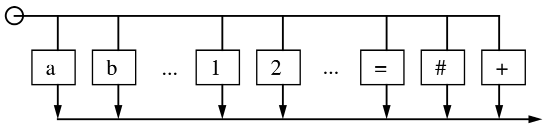
boolsk: true false

tekst: "a piece of text" "if" "a" "0" "3.14"

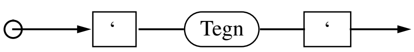
I tillegg har man også *spesielle tegn*, f.eks., ; ( ) & !
og *operatorer*, f.eks., + * - <> := <= =

Syntaks diagrammer

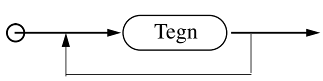
Tegn



Tegn Literal

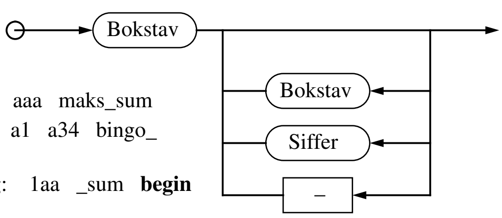


Tekst



Tekst Literal

Identifikator (Navn)

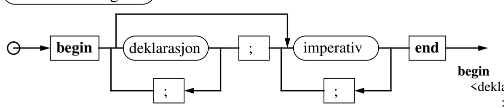


lovlig: aaa maks_sum
a1 a34 bingo_

ulovlig: 1aa _sum **begin**

Deklarasjoner og Imperativer – SIMULA0

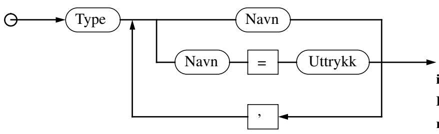
Simula0 - Program



begin
<deklarasjon> ;
:
<deklarasjon> ;

<imperativ> ;
:
<imperativ>
end

Deklarasjon



integer antall, sum
Boolean ferdig
real pi=3.14, noe,
to_pi = 2*pi

Uttrykk

(Verdi) type	Uttrykk (har en verdi) av typen
T	• variabel, (konstant,) literal • funksjonskall • typespesifike uttrykk
Aritmetiske uttrykk: **integer**	antall, 2, 97 **inint**, **rem**(8, 3), **rank**(‘a’) 8+antall, (2*4) // 3
Aritmetiske uttrykk: **real**	sum, 3.14, 0.5, 2.035, 2.667 **inreal**, **cos**(0), **sqrt**(3.14+1) 8+sum-antall, (2*4) / 3
boolean	ferdig, OK, **true**, **false** **letter**(‘c’), **digit**(‘c’), **not**(ferdig) <u>sammenligning</u>: antall = 3 55 >= 210 <u>operasjon</u>: ferdig **and** OK (3>sum) **or** antall=1
character	tegn_variabel, ‘z’, ‘+’ **inchar**, **char**(97)
text	“en tegnstreng”

Evaluering av uttrykk

Aritmetiske uttrykk	uttrykk	evaluering	verdi
fra venstre til høyre	3+4-2	(3+4)-2	5
innebygd rekkefølge			
1: **	3+4*2	(4*2)+3	11
2: *, /, //	3+4*2*3	((4*2)*3)+3	27
3: +, -	3+4*2**3	((2**3)*4)+3	35
innenfra og ut	3+(4*2)**3	((4*2)**3)+3	515

BRUK PARANTESER !!!

<u>Boolske operasjoner</u>

x \ y **and**	true	false
true	true	false
false	false	false

x \ y **and then** 2	true	false
true	true	false
false	false	false

x \ y **imp** 4	true	false
true	true	false
false	true	true

n>0 **and** m/n=10   n>0 **and then** m/n=10

x \ y **or**	true	false
true	true	true
false	true	false

x \ y **or else** 3	true	false
true	true	true
false	true	false

x \ y **eqv** 5	true	false
true	true	false
false	false	true

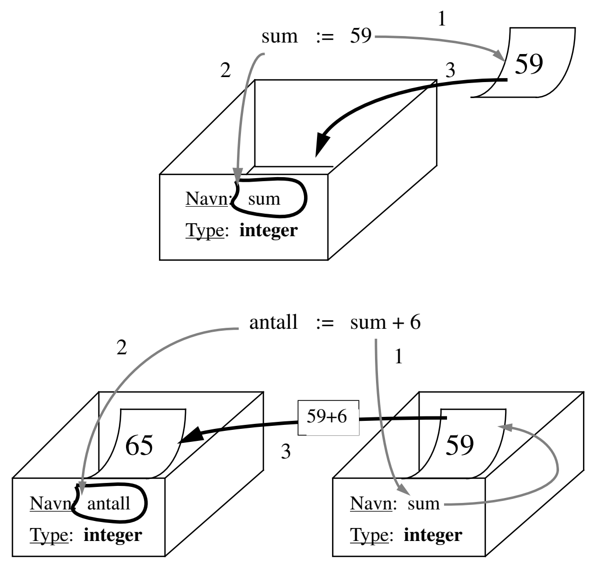
n=0 **or** m/n=10   n=0 **or else** m/n=10

Imperativer

tilordning	navn **:=** uttrykk plasser verdien til uttrykk i boksen navn
lager/minne	
input	**inimage** **inint**, **inreal**, **inchar**
output	**outimage**, **breakoutimage** **outtext**, **outint**, **outfix**, **outreal**, **outchar**
I / O	
betinget	**if** <boolsk uttrykk> **then** <imperativ-1> (**else** <imperativ-2>) **hvis** boolsk-uttrykk er sant **gjør** imperativ-1 **ellers** gjør imperativ-2
løkke	**while** <boolsk uttrykk> **do** <imperativ> **så lenge** boolsk-uttrykk er sant **gjør** imperativ
sammensatt	**begin** <imperativ>; ... ; <imperativ> **end** sett sammen imerativene til ett imperativ

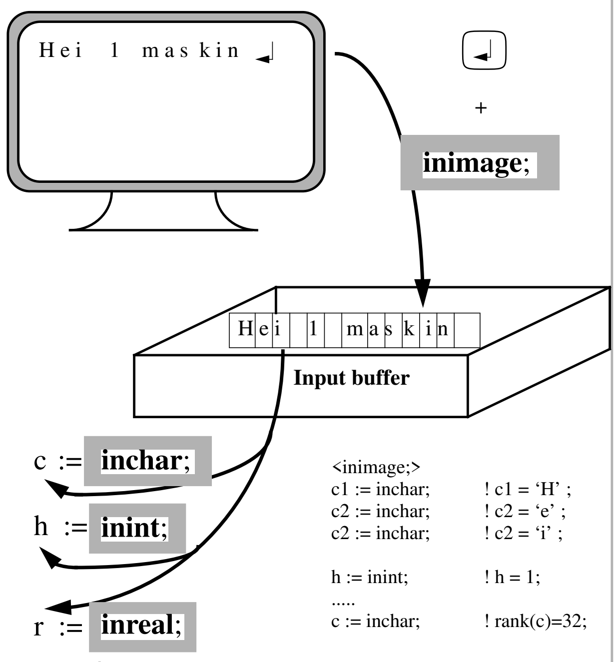
Tilordning

navn **:=** uttrykk



Navn må være (deklarert som) en variabel.
Navn og *uttrykk* må være av samme type !!

Input imperativer

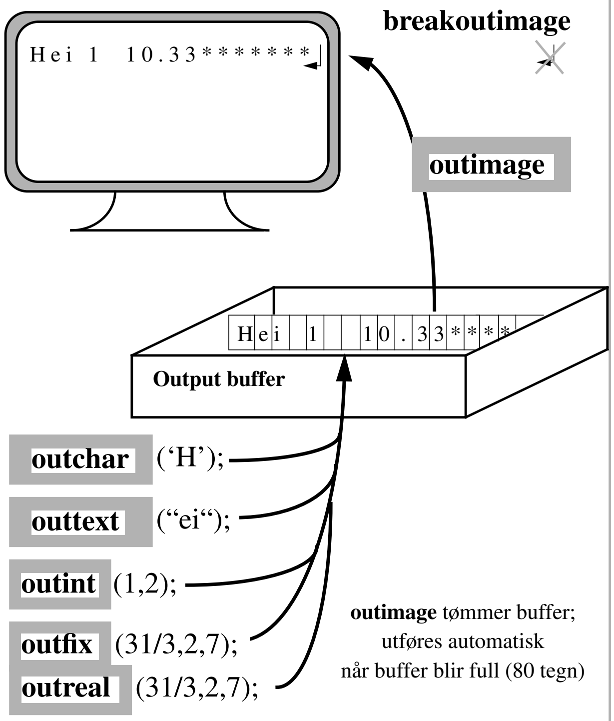



```
<inimage;>
c1 := inchar;      ! c1 = 'H' ;
c2 := inchar;      ! c2 = 'e' ;
c2 := inchar;      ! c2 = 'i' ;

h := inint;        ! h = 1;
.....
c := inchar;       ! rank(c)=32;
```

flytter posisjonspeker etter det innleste symbolet,
utfører **inimage** automatisk når buffer er tom.

Output imperativer



outimage tømmer buffer;
utføres automatisk
når buffer blir full (80 tegn)

Betinget imperativ

if <boolsk uttrykk> **then** <imperativ>

if <boolsk uttrykk> **then** <imperativ>
else <imperativ>

---

if antall=0 **then** outtext("Ingen observasjoner ?")

```
if (antall <= 0) and (siste_tegn = '?') then
   outtext("Takk")
else
   if antall <= 0 then        ! siste_tegn <> '?';
      outtext("Mer !!")
   else                       ! antall > 0 ;
      outint(antall,3)
```


Løkke-imperativ

while <boolsk uttrykk> **do** <imperativ>

---

while antall <> 0 **do** antall := inint

! Henter et heltall fra bufferen inntil tallet er 0 ;

! Dersom 0 ikke finnes i bufferen utfører **inimage** ;

Sammensatt imperativ

```
begin
   <imperativ-1>;
   <imperativ-2>;
        . . . .
   <imperativ-n>
end
```

```
if antall <= 0 then
   begin
      outtext("Mer !!"); outimage;
      inimage; antall := inint;
   end else
   begin
      outtext("Antallet er ");
      outint(antall,3); outimage;
   end
```

```
while   antall <> 0   do
   begin
      inimage;
      antall := inint
   end
```

!Leser inn kun det første tallet fra hver linje skrevet på skjermen inntil brukeren skriver 0 ;

Programmet skal også leses av mennesker !!!

Layout

```
begin integer antall; real tall,sum; antall:=0; sum:=0;
outtext("Tast inn tallene. ");outtext("Avslutt med 0. ");
outimage;inimage;tall:=inreal;while tall<>0 do begin
sum:=sum+tall;antall:=antall+1;inimage;tall:=inreal;end;
if antall=0 then begin outtext("Du tastet ingen tall !");
outimage;end else begin outtext("Gjennomsnittet av
tallene var "); outfix(sum/antall,3,10);outimage;end;end;
```

Kommentarer

....
! *Detter er en kommentar* ;
.....
! *Og så er*
dette ;
.....
end *og nok en kommentar* ; <**else**, **when**, **otherwise**>

Navn

integer x1, x2, x3, y;

integer antall, tall, sum, max_tall, x;