kontor.sim

```
ref(M_ottak) M;

ref(B_ehandling) B;

class M_ottak ;
 begin
    procedure hent_skjema(skj, knd); ref(S_kjema) skj; ref(K_unde) knd ;
     if knd.antall < knd.maks then
        send_videre(skj, knd)
     else knd . motta_svar(skj, false);
    procedure send_videre(skj, knd); ref(S_kjema) skj; ref(K_unde) knd ;
        B . hent_skjema(skj, knd) ;
 end;

class B_ehandling ;
 begin
    procedure hent_skjema(skj, knd); ref(S_kjema) skj; ref(K_unde) knd ;
        gi_svar(skj, knd) ;
    boolean procedure resultat(skj); ref(S_kjema) skj ;
        resultat := false;
    procedure gi_svar(skj, knd); ref(S_kjema) skj; ref(K_unde) knd ;
        knd . motta_svar(skj, resultat(skj)) ;
 end;

class K_unde(maks); integer maks;
 begin
    integer antall;     ref(S_oknad) array mine(1:maks) ;
    procedure motta_svar(skj, res); ref(S_kjema) skj; boolean res;
     begin ref(S_oknad) soknad;
        soknad :- finn(skj);
        if soknad =/= none then soknad . godkjent := res;
     end ;
    ref(S_oknad) procedure finn(skj); ref(S_kjema) skj ; ;
    procedure lever(soknad); ref(S_oknad) soknad;
     begin
        if antall < maks then
          begin antall:=antall+1; mine(antall):-soknad; end ;
        M . hent_skjema(soknad.skjema, ??? );
     end ;
 end;
```

```
begin  integer i; boolean ok;
  while i<=maks and not ok do
    if mine(i)=/= none and then
       mine(i).skjema == skjema
    then ok:= true
    else i:=i+1;
  if ok then finn :- mine(i)
       else finn :- none;
end ;
```


Verdi-overføring – input parametre

```
begin
   integer x;

   procedure nullstill (y); integer y;
      y := 0;

   procedure femstill; x := 5;

   . . .
   x := 100;
         ! x = 100 ;
   femstill;
         ! x = 5 ;
   nullstill(x);
         ! x = 5;
end
```

```
begin
   integer y;
   y := x;
   y := 0;
end;
```

verdien til den ***aktuelle*** parameteren
tilordnes den ***formelle*** parameteren
– “en lokal variabel”

Parametre av enkle typer **– integer, real, character, boolean –** overføres “by default” som verdier !!!!

Navn-overføring – output parametre

```
begin
   integer x;

   procedure nullstill (y); integer y;
      y := 0;

   procedure nullstill_n (y); name y; integer y;
      y := 0;

   . . .
   x := 100;
         ! x = 100 ;
   nullstill(x);               ! nullstill(100) ;
         ! x = 100;
   nullstill_n(x);
         ! x = 0;
   nullstill(x+2);             ! nullstill(2) ;
         ! x = 0;
   nullstill_n(100);           ! Feil ;
   nullstill_n(x+2);           ! Feil ;
end
```


Verdi- vs. navn-overføring

verdi

```
integer x;

procedure en (y); integer y;
   y := 1;

procedure verdi (y); integer y;
   while y<100 do x:=x*2;

x := 2;

en ( x );          begin integer y;
                     y := x ;
                     y := 1;
                   end;

verdi(4*x);        begin integer y;
                     y := 4*x;    !4*2;
                     while y<100 do
                       x := x*2;
                   end;
```

tekst

```
integer x;

procedure en (y); name y; integer y;
   y := 1;

procedure navn (y); name y; integer y;
   while y<100 do x:=x*2;

x := 2;

en ( x );                    x := 1;

navn( 4*x );       while 4*x <100 do x := x*2;
```

Referanse-overføring av ref-parametre

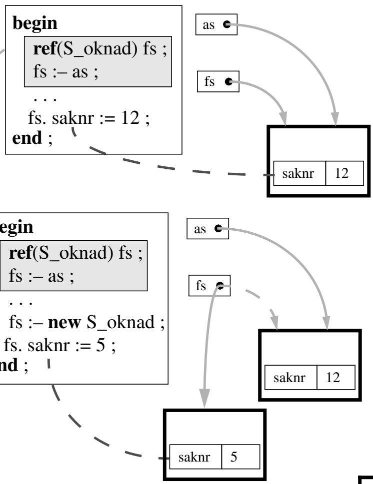
procedure lever(soknad); **ref**(S_oknad) soknad ;
  begin
    . . .
    soknad . saknr := 12 ;
  end ;

procedure lever_ny(soknad); **ref**(S_oknad) soknad ;
  begin
    . . .
    soknad :– **new** S_oknad ;
    soknad . saknr := 5 ;
  end ;

ref(S_oknad) as ;
as :– **new** S_oknad ;

lever(as) ;
! as . saknr = 12 ;

lever_ny(as) ;
! as . saknr = 12 ;



Navn-overføring av ref-parametre

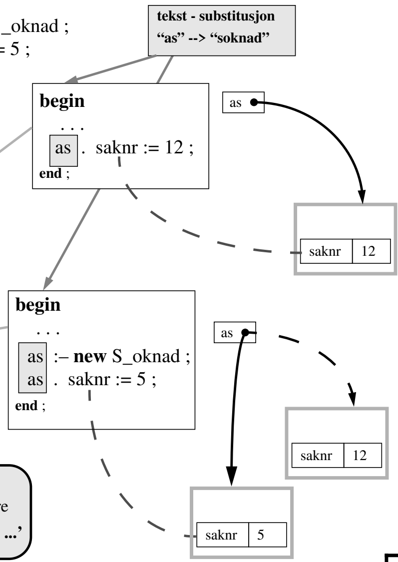
procedure lever(soknad); **name** soknad ; **ref**(S_oknad) soknad ;
  begin
    . . .
    soknad . saknr := 12 ;
  end ;

procedure lever_ny(soknad); **name** soknad ; **ref**(S_oknad) soknad ;
  begin
    . . .
    soknad :– **new** S_oknad ;
    soknad . saknr := 5 ;
  end ;

ref(S_oknad) as ;
as :– **new** S_oknad ;

lever(as) ;
! as . saknr = 12 ;

lever_ny(as) ;
! as . saknr = 5 ;



Brukes kun når prosedyren skal utføre **'formell parameter :– ...'**

F. eks.

class Elev ;
  begin . . . end;

ref(Elev) Ole, Marit, Jan ;
ref(Elev) beste, verste ;

~~**procedure** set_beste(ny_beste); **ref**(Elev) ny_beste ;~~
~~beste :– ny_beste ;~~

~~**procedure** set_verste(ny_verste); **ref**(Elev) ny_verste ;~~
~~verste :– ny_verste ;~~

procedure set(gam, ny); **name** gam; **ref**(Elev) gam, ny;
  gam :– ny ;

Ole :– **new** Elev; Marit :– **new** Elev; Jan :– **new** Elev;

set_beste(Ole) ;
! beste == Ole ;

set_verste(Marit) ;
! verste == Marit;

set(beste, Marit) ;
~~*! beste == Ole ;*~~ *! beste == Marit ;*

set(verste, Jan) ;
! verste == Jan ;

Typen tekst

begin

text navn, hei ; *! "referanse til et tekstobjekt" ;*

inspect new outfile("Stud.dta") **do**
  . . .

navn :– intext(20) ; *! "et tekstobjekt må lages"*
(navn "peker på" et nytt tekstobjekt)

inspect new intfile(navn) **do**
  . . .

hei :– "Hei" ; *! et konstant tekstobjekt (med lengde 3) ;*

end;

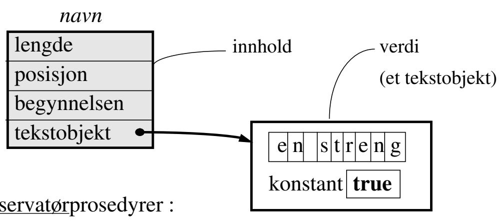
Hva skjer når man “lager” en tekst ?

tekst_variabel **:–** tekst_generator

text navn ;

navn **:–** “en streng” ;
:– copy(“en streng”) ;
:– intext(10) ;
:– blanks(20) ;
:– **notext** ;

Text er ikke en klasse, men:
text-variabler er sammensatte objekter
(ulikt **integer**, **character** ...).



med observatørprosedyrer :

navn . ***length*** = 9
navn . ***pos*** = 1
navn . ***start*** = 1
navn . ***constant*** = **true**

Hva kan man gjøre med *én* tekst ?

text t, p;
character c; integer i;
t :– intext(8) ;

c := t . getchar ;
! c = ‘a’ ;

t . putchar(‘x’);
! if **not** t . constant ;

i := t . getint; ! FEIL ;

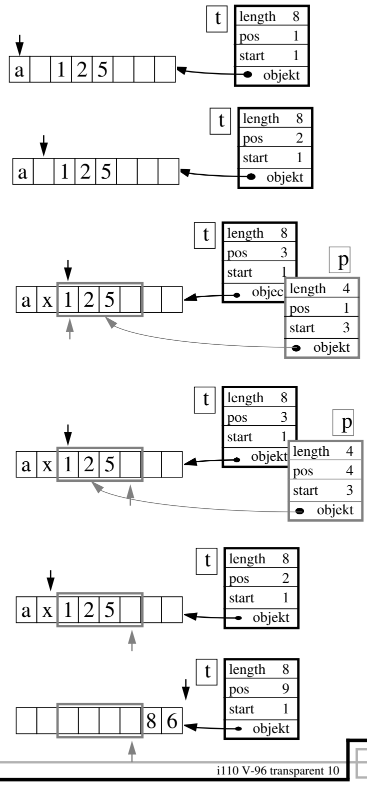
p :– t . sub(3,4) ;

i := p . getint;
! i=125 ;
i := t . sub(3,4) . getint;

p . main . setpos(2);

t . putint(86);

! t . more = false ;
! more := pos <= length ;

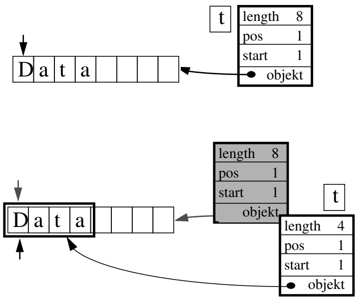


. . . og mer

text t ;

t :– intext(8) ;

t :– intext(8) . strip ;



t :– **genererer**
“en streng”
copy(tekst)
intext(x)
blanks(x)
notext

tekst . sub(x,y)
tekst . main
tekst . strip
tekst & tekst
produserer

tekst .
constant
pos
length
more
observerer

forandrer
setpos(x)
getchar
getint / getreal
konverterer hele tekstobjektet

putchar
putint / putfix / putreal
forandrer hele tekstobjektet

Pensum hittil

Kap. 1 - 6
(Kap. 7)
Kap. 8
Kap. 9.1 - 9.10
(9.11 - 9.12 : senere)
(9.13 - 9.14 : går ut)
9.15
9.16 (+ kompendiet “Om eksterne prosedyrer og Klasser”)
(Kap. 10 : går ut)
Kap. 11