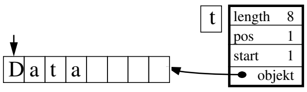
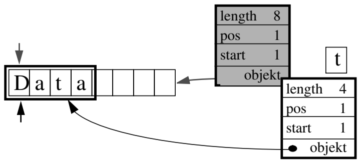
Tekstoperasjoner

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

Notext

er en tekst med en tom tekst-verdi
"tilsvarer **none** for klasser" (tekst == none er feiltypet)

blanks(10) – en tekst med lengde 10 fylt med blanker

NB! **notext** er forskjellig fra blanks(10)
[**notext** == blanks(10) => **false**]

blanks(10) . strip – deltekst av blanks(10)

notext er lik delstrengen blanks(10) . strip
[**notext** == blanks(10) . strip => **true**]

notext == blanks(0) => **true**

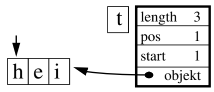
Et par eksempler

ord :– blanks(6) ;	! ord = "______" ;
del :– ord . sub(2, 3) ;	
ord := "abc" ;	! ord = "abc___" , ord.pos=4 ;
	! del = "bc_" , del.pos=1 ;
ord . setpos(1) ;	! ord.pos=1 ;
ord . putint(156) ;	! ord = "___156" , ord.pos=7 ;
	! del = "__1" ;
i := ord . sub(2,4) . getint ;	! i = 15 ;
ord . sub(1,2) . putint(10) ;	! ord = "10_156" ;
	! del = "0_1" ;
i := del . getint ;	! i = 0 , del.pos = 2 ;
i := del . getint ;	! i = 0 , del.pos = 2 ;
i := del . sub(2,2) . getint ;	! i = 1 , del.pos = 2 ;
dato :– blanks(8) ;	
dag :– dato . sub(1,2) ;	
mnd :– dato . sub(4,2) ;	
aar :– dato . sub(7,2) ;	
dato . sub(3,1) . putchar('/') ;	
dato . sub(6,1) . putchar('/') ;	
dag . putint(9) ;	
mnd . putint(10);	
aar . putin(94);	! dato = "_9/10/94" ;

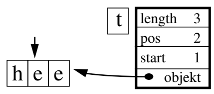
Tekst-tilordning

text t, p;

t :– copy("hei") ;



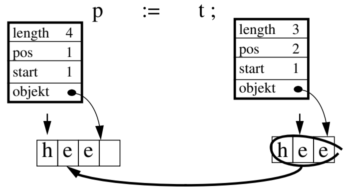
t := "hee";
t . getchar ;



~~p := t~~ ;

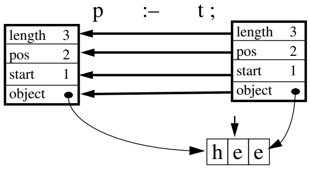
p :– blanks(4) ;

p := t ;



verdi-tilordning

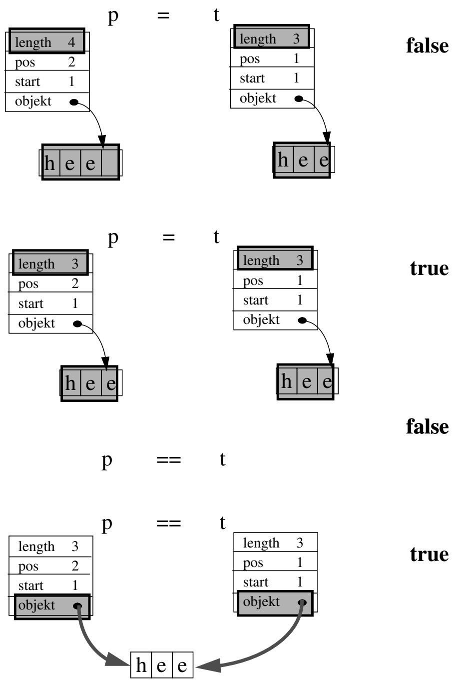
p :– t ;



referanse-tilordning

Tekstsammenligning

av verdi:



av referanse:

! p == t impliserer alltid p=t ;

NB ! Ingen innebygget sammenligning av *innhold*.

Texttools verktøykassen

```
class texttools;                ! Filen $i110/lib/texttools.sim ;
begin

  text procedure frontstrip(t); text t;
   if t == notext or else t.strip == notext then frontstrip :– notext
   else begin character c;
       t.setpos(1);
       c := t.getchar;
       while c = ‘ ‘ do c := t.getchar ;
       frontstrip :– t . sub (t . pos-1, t . length - t . pos + 2);
   end;

  text procedure int_as_text(int); integer int;
   begin text t;
       t :– blanks(12); t . putint(int); int_as_text :– frontstrip(t) ;
   end;

  text procedure char_as_text(char); character char;
   begin text t;
       t :– blanks(1); t . putchar(char); char_as_text :– t ;
   end;

  text procedure real_as_text(r, dig); real r; integer int;
   begin text t;
       t :– blanks(12+dig); t . putfix(r, dig); real_as_text :– frontstrip(t) ;
   end;

  text procedure next_word_in(t); text t;
   if t == notext or else not t . more then next_word_in :– notext
   else begin character c; text word;
       c := t.getchar;
       while c = ‘ ‘ and t . more do c := t.getchar ;
       while c <> ‘ ‘ do
         begin
           word :– word & char_as_text(c) ;
           c := if t . more then t.getchar else ‘ ‘;
         end ;
       next_word_in :– word ;
   end;

end texttools ;
```


Entydig navngiving

Gitt:

class ting(type,navn); **character** type; **text** navn; ;

Generer n ting av type ‘A’ og n ting av type ‘B’ slik at alle har entydig navn.

```
begin
  external class texttools;
  integer n;
  class ting(type,navn); character type; text navn; ;
  n := inint ;
  . . .
  texttools begin
   ref(ting) array A_ting, B_ting (1 : n);
   integer j;
   for j:=1 step 1 until n do
    begin
      A_ting(j) :– new ting(‘A’, “A” & int_as_text(j) ) ;
      B_ting(j) :– new ting(‘B’, “B” & int_as_text(j) ) ;
    end;
   . . .
  end textools;
end ;
```


Bruk av flere verktøykasser

```
begin
  external class texttools;
  texttools begin
    integer i;
    i := inint ;
    outint(i,12); outimage;
    outtext(int_as_text(i)); outimage ;
    . . .
  end texttools;
end ;
```

```
begin
  external class texttools, promptools;
  texttools begin
    promptools begin
      . . .
      outtext ( int_as_text ( be_om_int(“Tallet som skal konverteres til tekst:”) ));
      . . .
    end promptools ;
  end texttools;
end ;
```

```
inspect new texttools do
begin
  inspect new promptools do
  begin
```

```
begin
  external class texttools, promptools;
  ref(texttools) tt;
  tt :– new texttools ;
  promptools begin
    . . .
    outtext( tt . int_as_text ( be_om_int(“Tallet som skal konverteres til tekst:”) ));
    . . .
  end prompttools;
end ;
```

Tekst parametre

text t1, t2, t3;
procedure vt (v, r, n); **value** v; **name** n; **text** v, r, n;
begin
  v:= "aa"; r:= "bb"; n:= "cc";
end ;
procedure rt (v, r, n); **value** v; **name** n; **text** v, r, n;
begin
  v:– "aa"; r:– "bb"; n:– "cc";
end ;

t1 :– copy("tst1") ;
t2 :– copy("tst2") ;
t3 :– copy("tst3") ;

vt(t1, t2, t3);

! t1= "tst1", t2= "bb", t3= "cc";

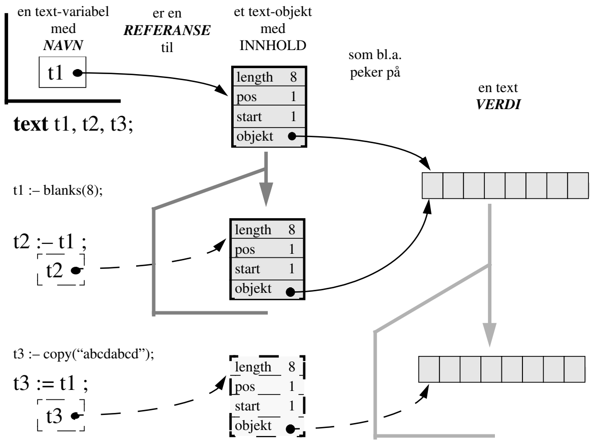
begin
  text v, r ;
  v := t1 ;
  r :– t2 ;
  v := "aa" ;
  r := "bb" ;
  ~~n~~ t3 := "cc" ;
end;

! t1==t1', t2==t2', t3==t3' ;
rt(t1, t2, t3) ;

! t1 == t1', t1 = "tst1" ;
! t2 == t2', t2 = "bb" ;
! t3 =/= t3', t3 = "cc" ;

begin
  text v, r ;
  v := t1 ;
  r :– t2 ;
  v :– "aa" ;
  r :– "bb" ;
  ~~n~~ t3 :– "cc" ;
end;

Tekst-tilordning og -parametre



procedure vt (v, r, n);
  value v; **name** n; **text** v, r, n;
begin
  v:= "aa"; r:= "bb"; n:= "cc";
end ;
t1 :– copy("tst1");
t2 :– copy("tst2");
t3 :– copy("tst3");
vt (t1, t2, t3);
! t1 = "tst1", t2 = "bb", t3 = "cc";

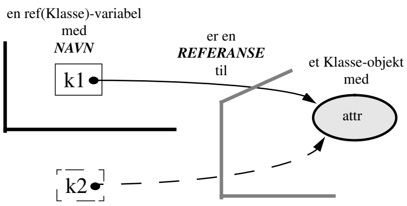
begin
  text v, r ;
  v := t1 ;
  r :– t2 ;
  v := "aa" ;
  r := "bb" ;
  ~~n~~ t3 := "cc" ;
end;

Det samme for ref(Klasse)-parametre nesten . . .

class Klasse ; ;

ref(Klasse) k1, k2;

k1 :– new Klasse ;



k2 :– k1 ;

! istedenfor k2 := k1, har vi ;
k2 . attr := k1 . attr ;

procedure rt (r, n); **name** n; **ref**(Klasse) r, n;
begin
  r . attr := B1; r :– **new** Klasse ;
  n . attr := B2; n :– **new** Klasse ;
  ! Alt.: r :– **new** Klasse   r . attr := B1;
  !   n :– **new** Klasse n . attr := B2;
end ;
k1 :– new Klasse; ! k1', A1;
k2 :– new Klasse; ! k2', A2;

rt (k1, k2);

! k1 == k1', B1   k1==k1', A1
 k2 =/= k2', ??   k2=/=k2', B2 ;

begin
  ref(Klasse) r ;
  r :– k1;
  r . attr := B1;
  r :– **new** Klasse;
  ~~n~~ k2 . attr := B2;
  ~~n~~ k2 :– **new** Klasse;
end;