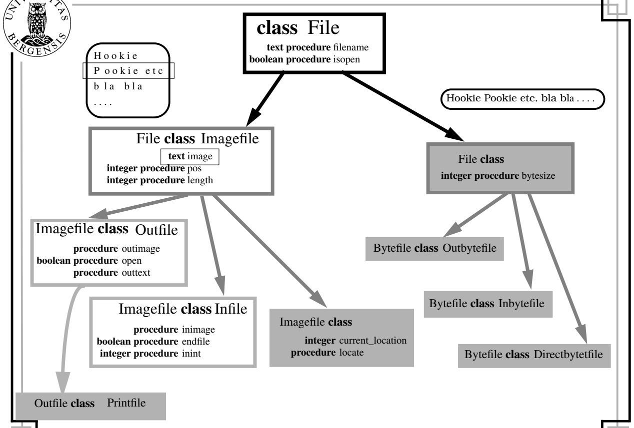


forelesning 16

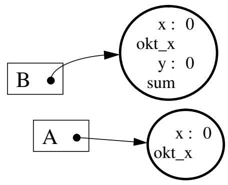
Subklasser

begin
  class Super;
   begin
     integer x;
     procedure okt_x; x:= x+1;
   end;

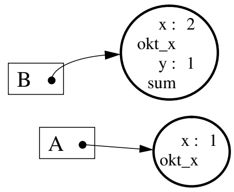
  Super **class** Sub;
   begin
     integer y;
     procedure sum; y:= y + x;
   end;

Sub **arver** alle attributtene deklarert i *Super*.
Hvert objekt av *Sub*-klassen vil ha **alle** attributtene deklarert i *Super*-klassen i tillegg til sine egne.

  ref(super) A;
  ref(sub) B;
  B :– **new** Sub ;
  A :– **new** Super ;

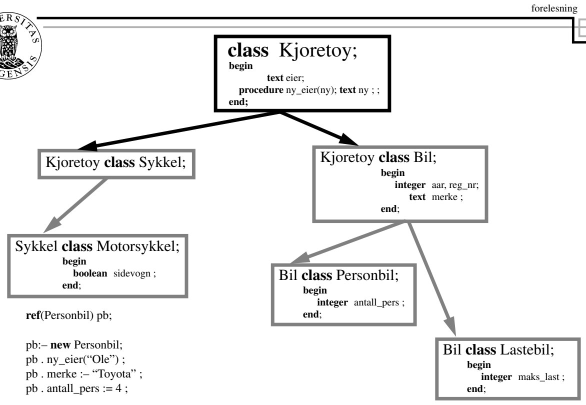


  B . okt_x;
  ! B . x = 1;
  B . sum ;
  ! B . y = 1;
  B . okt_x ;
  ! B . x = 2;
  A . okt_x ;
  ! A . x = 1 ;
  . . .



end ;

forelesning 16



ref(Personbil) pb;

pb:– **new** Personbil;
pb . ny_eier("Ole") ;
pb . merke :– "Toyota" ;
pb . antall_pers := 4 ;

pb . sidevogn := false ;
pb . maks_last := 500 ;

forelesning 16

Filer i SIMULA

```
class File(protected_name); ... ; text protected_name;
protected protected_name, file_is_open ;
begin
   text procedure filename; filename :– protected_name ;
   Boolean file_is_open;
   Boolean procedure isopen; isopen := file_is_open ;
   procedure setaccess(access_code); text access_code;
   if protected_name == notext then <Error, Abort>;
end File;
```

```
File class Imagefile;
begin
   text image ;
   procedure setpos(i); integer i; image . setpos(i);
   integer procedure pos; pos := image . pos;
   integer rocedure length; length := image . length;
   Boolean procedure more; more := image . more;
end File;
```

```
Imagefile class Outfile;
begin
   Boolean procedure open(img); text img;
   Boolean procedure close;
   procedure outimage;       procedure breakoutimage;
   procedure outrecord;
   procedure outchar(c); character c;
   procedure outtext(t); text t;
   procedure outtint(i,w); integer i,w;
   procedure outfix(r,d,w); real r; integer d,w;
   procedure outreal(r,d,w);  real r; integer d,w;
end Outfile;
```

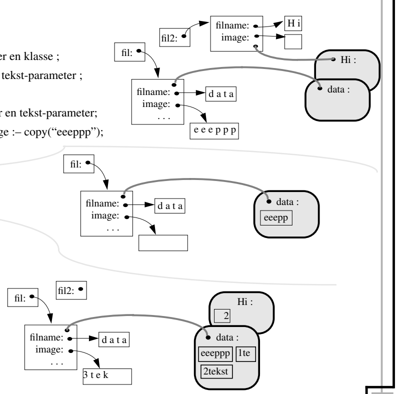
```
Imagefile class Infile;
begin
   Boolean procedure open(img); text img;
     . . . ; image :– img;
     image := notext; . . .
   Boolean end_of_file_met ;
   Boolean procedure endfile; endfile := end_of_file_met;
   Boolean procedure lastitem;
   Boolean procedure close;
   procedure inimage;
   procedure inrecord;
   character procedure inchar;
   text procedure intext(w); integer w;
   integer procedure inint;
   real procedure inreal;
   real procedure infrac;
end Infile;
```

Imagefile **class** Outfile

```
File class Imagefile;
begin
   ! attributtene som er arvet fra File : filename, file_is_open, isopen, setaccess;
   text image ;
   procedure setpos(i); integer i; image . setpos(i);
   integer procedure pos; pos := image . pos;
   integer procedure length; length := image . length;
   Boolean procedure more; more := image . more;
end File;

Image class Outfile;
begin
   ! attributtene som er arvet fra Imagefile ;
   Boolean procedure open(img); text img;
     begin
        ! ber OS finne filen filname (ev. finne plass for en ny);
        if (not isopen) and then <access OK> then
          begin
            image :– img;
            setpos(1);
            file_is_open:=true; open := true;
        end else open:=false;
     end open;
   Boolean procedure close;
     . . . ; if pos>1 then outimage;  image :– notext;  file_is_open := false; . . .
   procedure outimage;
     begin
        ! flytt innholdet av image til den eksterne filen;
        image := notext; setpos(1);
     end;
   procedure outchar(c); character c;
     begin   if not more then outimage;   image . putchar(c); end;
   procedure outtext(t); text t;
     begin
        if pos > 1 and then pos + t.length - 1 > length then outimage;
        t . setpos(1);
        while t . more do outchar(t . getchar);
     end outtext ;
   . . . outint . . . outreal . . . outfix . .
end Outfile;
```


For eksempel

```
begin
   ref (outfile) fil, fil2;              ! outfile er en klasse ;
   fil :– new outfile("data");           ! med en tekst-parameter ;
   fil2 :– new outfile("Hi");
   fil . open(copy("eeeppp"));           !open har en tekst-parameter;
                                         !fil . image :– copy("eeeppp");
   fil2 . open(blanks(2)) ;

   fil . outimage ;

   fil . outtext("1te");
   fil . outimage ;
   fil . image := "2tekst" ;
   fil . outimage ;
   fil2 . outtint(2,2);
   fil2 . close ;
   fil . outtext("3tek") ;
   . . .
```

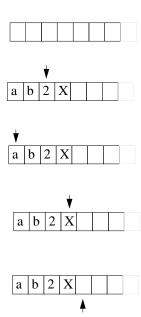


Imagefile **class** Infile

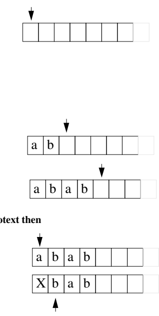
```
begin
   ! attributtene som er arvet fra Imagefile ;
   Boolean procedure open(img); text img;
     begin
        ! ber OS finne filen filename
        if (not isopen) and then <access OK> then
          begin
            image :– img; file_is_open:=true;  open := true;
            image := notext;   setpos(length+1); end_of_file_met := false;
        end else open:=false;
     end open;
   Boolean procedure close;
   . . . ; image :– notext; file_is_open := false; end_of_file_met := true;
   Boolean end_of_file_met ;
   Boolean procedure endfile; endfile := end_of_file_met;
   Boolean procedure lastitem;
     begin c := ' ';
        while (not endfile) and then c:=' ' do c:=inchar;
        lastitem:=endfile;     if c<>' ' then setpos(pos-1);
     end;
   Boolean procedure more; more := image . more;
   procedure inimage;
     if endfile then <Error> else
     if <ingen flere image i den eksterne filen> then
        begin end_of_file_met := true;  image := "!25!" end
     else if length < <lengden av den eksterne image> then <Error>
     else begin <hent den eksterne image>
        if length > <lengden av den eksterne> then <fyll med blank>;
        setpos(1);
     end inimage;
   procedure inrecord;  ! inimage – ekstern>intern: ingen feil & ekstern<intern: ingen blank;
   character procedure inchar;
     begin if not more then inimage; inchar := image . getchar; end;
   text procedure inttext(w); integer w;
     begin text t;
        t :– blanks(w);
        while t . more do t . putchar(inchar);
        intext :– t;
     end intext ;
   . . . inint . . . inreal . . . infrac . .
end Outfile;
```


Sysin / Sysout

ab2X

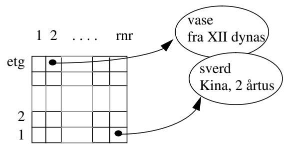



```
begin
  character c;
  integer i, j, k, l ;
  c := inchar ;
  while not digit(c) do
     begin k:=k+1; c:=inchar; end;
  Sysin . setpos (Sysin . pos – 1) ;
  i := inint ;
  for j:= 1 step 1 until i do
     begin
        Sysin . setpos(1) ;
        for l:= 1 step 1 until k do outchar(inchar) ;
     end ;
  inint ;
  if Sysin . image . sub(Sysin.pos, 81–Sysin.pos) . strip =/= notext then
     begin
        Sysout . setpos(1) ;
        outchar(inchar) ;
     end ;
end ;
```



Datastruktur og filformat

```
class ting(navn, beskr); text navn, beskr ; ;

class Museum(etg, rnr); integer etg, rnr;
  begin
    ref(ting) array ling(1:etg, 1:rnr);
  end;

ref(Museum) sam ;
```



Data lagret på en fil skal

- inneholde **all informasjon** samlet i datastrukturen
  - "avspeile" datastrukturen
- gjøre det mulig å **gjenoppbygge** datastrukturen
  - de samme prosedyrene som leser inn data fra skjermen bør kunne lese dem fra en fil
  - filformatet vil "avspeile" rekkefølgen av innlesning

1	6	11	21 80
1	15	sverd	Kina, 2 årtusen f.K., . . .
5	2	vase	fra XII dynastiet, utført i . . .

. . . .

Skriv og Les

```
procedure skriv_en_ting(x, y ,ut); integer x,y;  ref(outfile) ut;
begin ref(outfile) ut;
  ut :– new outfile(intext(30).strip);
  ut . setaccess("Append");
  ut . open (blanks(80));
  ut . outint(x,5); ut . outint(y,5);
  ut . outtext(sam.ling(x,y).navn);
  ut . setpos(21); ut . outtext(sam.ling(x,y).beskr);
  ut . close;
end skriv_en_ting;

procedure les_en_ting  (inn) ; ref(infile) inn;
begin ref(infile) inn; inn :- new infile(intext(30).strip);
  inn . open (blanks(80));
  sam.ling( inn . inint, inn . inint ) :–
                   new ting( inn . intext(10).strip,  inn . intext(60).strip);
  inn . close ;
end les_en_ting;

procedure les_Museum;
begin ref(infile) inf;
  inf :- new infile(intext(30).strip);
  inf . open (blanks(80));
  sam:–
      new Museum(inf.inint, inf.inint);
  inf . inimage ;
  while not inf . endfile do
   begin
     les_en_ting(inf);
     inf . inimage;
   end;
  inf . close ;
end les_Museum;
```

```
procedure skriv_Museum;
begin ref(outfile) utf;
  integer i, j;
  utf :- new infile(intext(30).strip);
  utf . open (blanks(80));
  utf . outint(sam.etg,5);
  utf . outint(sam.rnr,5);
  utf . outimage ;
  for i:=1 step 1 until sam.etg do
   for j:=1 step 1 until sam.rnr do
     if sam.ling(i,j) =/= none then
        begin  skriv_en_ting(i, j, utf);
               utf . outimage; end ;
  utf . close ;
end skriv_Museum;
```


Fil / Skjerm

```
procedure les_Museum  (inf);
                ref(infile) inf;
begin ref(infile) inf;
  inf :- new infile(intext(30).strip);
  inf . open (blanks(80));
  sam:–
     new Museum(inf.inint, inf.inint);
  inf . inimage ;
  while not inf . endfile do
   begin
     les_en_ting(inf);
     inf . inimage;
   end;
  inf . close ;
end les_Museum ;
```

```
procedure skriv_Museum  (utf);
                   ref(outfile) utf;
begin ref(outfile) utf;
  integer i, j;
  utf :- new infile(intext(30).strip);
  utf . open (blanks(80));
  utf . outint(sam.etg,5);
  utf . outint(sam.rnr,5);
  utf . outimage ;
  for i:=1 step 1 until sam.etg do
   for j:=1 step 1 until sam.rnr do
     if sam.ling(i,j) =/= none then
        begin  skriv_en_ting(i, j, utf);
               utf . outimage; end ;
  utf . close ;
end skriv_Museum;
```

```
procedure Les;
begin ref(infile) inf;
  text t;
  t :– intext(30).strip;
  if t=/= notext then
   begin
     inf :- new infile(t);
     inf . open (blanks(80));
     les_Museum(inf);
     inf . close ;
   end else
     les_en_ting(Sysin);
end ;
```

```
procedure Skriv;
begin ref(outfile) utf;
  text t;
  t :– intext(30).strip;
  if t=/= notext then
   begin
     utf :- new outfile(t);
     utf . open (blanks(80));
     skriv_Museum(utf);
     utf . close ;
   end else
     skriv_Museum(Sysout);
end les_Museum ;
```


Hele programmet

```
begin                                   class ting...; class Museum....;
  procedure les_en_ting(inn);  ref(infile) inn;
   sam.ling(inn.inint, inn.inint) :– new ting (inn.intext(10).strip, inn.intext(60).strip);

  procedure les_Museum(inf);  ref(infile) inf;
  begin
   sam:– new Museum(inf.inint, inf.inint); inf . inimage ;
   while not inf . endfile do
      begin les_en_ting(inf); inf . inimage; end;
  end les_Museum ;

  procedure Les;
  begin text t;
   t :– intext(30).strip;
   if t=/=notext then
    begin ref(infile) inf;
       inf :- new infile(t); inf . open (blanks(80)); les_Museum(inf); inf . close ;
    end else les_en_ting(Sysin);
  end Les ;

  procedure skriv_en_ting(x,y,ut); integer x,y; ref(outfile) ut;
  begin
   ut.outint(x,5); ut.outint(y,5); ut.outtext(sam.ling(x,y).navn);
   ut.setpos(21); ut.outtext(sam.ling(x,y).beskr);
  end skriv_en_ting;

  procedure skriv_Museum(utf); ref(outfile) utf;
  begin integer i, j;
   utf . outint(sam.etg,5); utf . outint(sam.rnr,5);utf . outimage ;
   for i:=1 step 1 until sam.etg do
    for j:=1 step 1 until sam.rnr do
       if sam.ling(i,j)=/= none then begin  skriv_en_ting(i, j, utf); utf . outimage; end ;
  end skriv_Museum;

  procedure Skriv;
  begin ref(outfile) utf; text t;
   t :– intext(30).strip;
   if t=/=notext then
    begin
       utf :- new outfile(t); utf . open (blanks(80)); skriv_Museum(utf); utf . close ;
    end else skriv_Museum(Sysout);
  end Skriv ;

  ref(Museum) sam; character c;

  sam :– new Museum(inint,inint); c:=inchar;
  while c<>'q' do
   begin
     if c='l' then Les
     else if c='s' then Skriv
     else if c='p' then skriv_en_ting(inint,inint,Sysout);
     c:=inchar;
   end;
end ;
```

Tekstskille

```
procedure les_en_ting(inn);  ref(infile) inn;
  sam.ling(inn.inint, inn.inint) :–
                     new ting (inn.intext(10).strip, inn.intext(60).strip);

procedure skriv_en_ting(x,y,ut); integer x,y; ref(outfile) ut;
begin
  ut.outint(x,5); ut.outint(y,5); ut.outtext(sam.ling(x,y).navn);
  ut.setpos(21); ut.outtext(sam.ling(x,y).beskr);
end skriv_en_ting;
```

```
> l
> 2 3 ljaa meget verdifull, fra indre Telemark
> p 2 3
    2    3  ljaa meget verdifull, fra indre Telemark
> s ljaa.dta
```

1	6	11	21	80
2	3	ljaa mege	t verdifull, fra indre Telemark	

Hvordan skiller vi mellom tekster på en fil ?

1. ved å bruke **fast lengde**

2. ved å skrive **tekstlengden foran** selve teksten
   (ubrukelig for Sysin)

3. ved å markere et skille med et spesielt **skilletegn**

Tekstlengden

1	6	11	14	21	24
2	3	7	en ljaa	34	meget verdifull, fra indre Telemark

```
procedure skriv_en_ting(x,y,ut); integer x,y; ref(outfile) ut;
begin
  ut . outint(x,5);   ut . outint(y,5);
  ut . outint(sam.ling(x,y).navn.length,3);  ut . outtext(sam.ling(x,y).navn);
  ut . outint(sam.ling(x,y).beskr.length,3); ut . outtext(sam.ling(x,y).beskr);
end skriv_en_ting;
```

```
procedure les_en_ting(inn);  ref(infile) inn;
begin
  integer x, y; text n, b;
  x := inn . inint; y := inn . inint;
  n :– inn . intext( inint );
  b :– inn . intext( inint );
  sam.ling(x,y) :– new ting ( n , b );
end les_en_ting;
```

1	6	11	14	20	23
2	3	6	12 ljaa	34	meget verdifull, fra indre Telemark

. . . n :– inn . intext(612)

Skilletegn

1	6	11	19		
2	3	en ljaa	$	meget verdifull, fra indre Telemark	$

```
character skill = '$' ;  ! må være sikker at det ikke forekommer i selve teksten ;

procedure skriv_en_ting(x,y,ut,skill); integer x,y; ref(outfile) ut;
                                        character skill;
begin
  ut . outint(x,5);   ut . outint(y,5);
  ut . outtext(sam.ling(x,y).navn);   ut . outchar(skill) ;
  ut . outtext(sam.ling(x,y).beskr);  ut . outchar(skill) ;
end skriv_en_ting;

text procedure til_skill(inn,skill); ref(infile) inn; character skill;
begin
  character c; integer start;
  start := inn . pos;
  c := inn . inchar;
  while inn . more and then c <> skill do c := inn . inchar;
  til_skill :– copy( inn . image . sub (start, inn . pos - start - 1) );
end;

procedure les_en_ting(inn,skill);  ref(infile) inn; character skill;
begin
  integer x, y; text n, b;
  x := inn . inint;  y := inn . inint;
  n :–  til_skill(inn,skill);
  b :–  til_skill(inn,skill);
  sam.ling(x,y) :– new ting ( n , b );
end les_en_ting;
```


Innlesning/utskrift
(med et skilletegn)

```
procedure les_en_ting(inn, skill);  ref(infile) inn;  character skill;

procedure les_Museum(inf, skill);  ref(infile) inf; character skill;

procedure Les;
begin text t;
  t :– intext(30).strip;
  if t=/=notext then
   begin ref(infile) inf;
      inf :- new infile(t);
      inf . open (blanks(80));
      les_Museum(inf, '$' );
      inf . close ;
   end else les_en_ting(Sysin, '$' );
 end Les ;

procedure skriv_en_ting(x,y,ut, skill);
                       integer x,y; ref(outfile) ut; character skill;

procedure skriv_Museum(utf, skill); ref(outfile) utf;  character skill;

procedure Skriv;
begin ref(outfile) utf; text t;
  t :– intext(30).strip;
  if t=/=notext then
   begin
      utf :- new outfile(t);
      utf . open (blanks(80));
      skriv_Museum(utf, '$' );
      utf . close ;
   end else skriv_Museum(Sysout, ' ' );
end Skriv ;
```