Windowtools

Verktøy for bruk av vinduer fra Simula
Del I

Bjørn Kirkerud

Versjon 2.1 (Oppdatert 19. oktober 1992)

1 Innledning

Windowtools er et sett av verktøy – ialt vesentlig klasser med prosedyrer – som kan benyttes i Simula-programmer for å opprette og bruke vinduer.

Tre viktige Windowtools-klasser heter MainWindow, Button og PromptWindow. I avsnitt 2 vises – ved hjelp av enkle eksempler – hvordan disse klassene og noen av deres prosedyrer kan brukes.

I avsnitt 3 beskrives de viktigste prosedyrene i MainWindow, og også i SubWindow, som er en klasse som inneholder nøyaktig de samme prosedyrene som MainWindow.

I avsnitt 4 og 5 beskrives henholdsvis Button og PromptWindow.

I avsnitt 6 beskrives et program som er litt mer komplisert enn programmene i avsnitt 2.

Det finnes en rekke Windowtools-klasser i tillegg til MainWindow, SubWindow, Button og PromptWindow. Disse klassene vil bli beskrevet i et notat *Windowtools, Del II*.

2 Noen enkle eksempler

Dette avsnittet inneholder noen enkle eksempler som viser hvordan klassene MainWindow, Button og PromptWindow kan brukes.

2.1 Et meget enkelt eksempel

Følgende program (en kopi befinner seg på filen `$i110/filer/forelesning/heihei1.sim`) lager et lite vindu som plasseres øverst til venstre på skjermen, og skriver teksten `"Hei-Hei Beate"` midt i dette vinduet. Deretter venter programmet inntil brukeren finner det for godt å trykke CR i sitt kommando-vindu.

```
external class Windowtool2;
begin
   ref (MainWindow) vnd;
   vnd :- new MainWindow("Hei-Hei");
   vnd.PlaceAt(20, 50);
   vnd.SetSize(600, 200);
   vnd.Show;
   vnd.DrawCenterText("Hei-hei Beate!");
   inimage;
end
```

De Windowtools-verktøyene som er brukt i dette eksemplet virker som følger:

- Virkningen av "**external class Windowtool2**" er at alt Windowtools-verktøyet blir tilgjengelig til bruk i resten av programmet.

  Årsaken til at det står "Windowtool2" (og ikke bare Windowtools) er at dette er versjon 2 av Windowtools. Versjon 1 (som ble brukt i IN105 i Oslo våren 1992) er ikke tilgjengelig.

- MainWindow er en ganske stor klasse som inneholder diverse vindus-verktøy (ialt vesentlig prosedyrer).

- Virkningen av "vnd :− **new MainWindow(**`"Hei-Hei"`**);**" er at det opprettes et nytt vindu, som får navn `"Hei-Hei"`.

- Virkningen av "vnd . PlaceAt(50, 50)" er at vinduet referert til av vnd plasseres med venstre kant 20 pixeler til høyre for skjermens venstre kant, og toppen 50 pixeler nedenfor skjermens øvre kant.

- Virkningen av "vnd . SetSize(600, 200)" er at vinduet vnd blir 600 pixeler bredt og 200 pixeler høyt.

- Virkningen av "vnd . Show" er at vinduet vnd blir synlig på skjermen (inntil Show utføres, er vinduet usynlig).

- Virkningen av "vnd . DrawCenterText(`"Hei-hei Beate!"`)" er at teksten `"Hei-hei Beate!"` skrives midt i vinduet vnd.

Dersom programmet ovenfor plasseres i en fil med navn heihei1 . sim, kan det kompileres ved å gi kommandoen

```
sc  -wtools  heihei1
```

Legg merke til at det står wtools, og ikke bare tools.

2.2 Et ganske enkelt eksempel

Nedenfor følger en litt mer avansert versjon av heihei-programmet:

```
external class Windowtool2;
begin
  ref(MainWindow) vnd;
  ref(Button) stopp_knapp;
    ! Button er en Windowtools-klasse som definerer "trykk-knapper";
  vnd :- new MainWindow("Hei-Hei")
                   . PlaceAt(50, 50)
                   . SetSize(600, 250)
                   . Show;
    ! Hver av prosedyrene PlaceAt, SetSize og Show er en funksjons-prosedyre
    ! som har som verdi en referanse til det objektet prosedyren blir kalt i.
    ! Dermed kan de invokeres slik som gjort her;
  stopp_knapp :- new Button(vnd, "Klikk her!")
                   . PlaceAt(220, 200)
                   . Show;
    ! Det opprettes en ny Button som hører til vnd og som plasseres
    ! med sitt øvre venstre hjørne 220 pixeler fra venstre kant
    ! av vnd og sin øvre kant 200 pixeler fra øvre kant av vnd;
  vnd . DrawCenterText("Hei-hei Beate!");
  stopp_knapp . Wait;
```

```
    ! Programmet "henger" her inntil brukeren flytter muse-markøren
    ! inni stopp_knapp-en og klikker en muse-knapp;
end
```

2.3 Et temmelig enkelt eksempel

En tredje versjon av heihei-programmet:

```
external class windowtool2;
begin
  ref (MainWindow) vnd;
  ref (PromptWindow) be_om_vnd;
    ! PromptWindow er en klasse i Windowtools
    ! som inneholder diverse dialog-verktøy;
  ref (Button) stopp_knapp;
  text bruker_navn;
  vnd :- new MainWindow("Hei-Hei")
                .PlaceAt(50, 50)
                .SetSize(600, 250)
                .Show;
  be_om_vnd :- new PromptWindow(vnd)
                .PlaceAt(10, 10)
    ! Det opprettes et PromptWindow som eies av vnd,
    ! og som plasseres nesten helt opp i vnd's øvre venstre hjørne
    ! Det er (foreløpig) usynlig ( Show er ikke utført);
  bruker_navn :- be_om_vnd.ask_for_text("Hva heter du", 15);
    ! PromptWindow-et blir synlig. Det inneholder ledeteksten
    ! "Hva heter du", og et felt med plass til 15 tegn. Brukeren
    ! kan flytte markøren inn i dette feltet, skrive navnet sitt, og
    ! avslutte enten med CR eller med å klikke i en egen OK-boks som
    ! befinner seg inni PromptWindow-et;
  vnd.DrawCenterText("Hei-hei " & bruker_navn & "!");
  stopp_knapp :- new Button(vnd,"Klikk her " & bruker_navn & "!")
                .PlaceAt(220, 200)
                .Show;
  stopp_knapp.Wait;
end
```

2.4 Et siste enkelt eksempel

En fjerde (og siste) versjon av heihei-programmet (en kopi finnes på filen $i110/filer/forelesning/heihei4.sim):

```
external class Windowtool2;
begin
  ref (MainWindow) vnd;
  ref (PromptWindow) be_om_vnd;
  ref (Button) start_knapp, stopp_knapp;
```

```
    text bruker_navn;
    integer x1, y2, x2, y2, froe;
    vnd :- new MainWindow("Hei-Hei")
                    .PlaceAt(50, 50)
                    .SetSize(600, 250)
                    .Show;
    be_om_vnd :- new PromptWindow(vnd)
                    .PlaceAt(10, 10)
    bruker_navn :- be_om_vnd.ask_for_text("Hva heter du", 15);
    vnd.DrawCenterText("Hei-hei " & bruker_navn & "!");
    start_knapp :- new Button(vnd,"Klikk her " & bruker_navn & "!")
                    .PlaceAt(220, 200)
                    .Show;
    start_knapp.Wait;
    start_knapp.Unshow;
      ! Knappen blir usynlig;
    stopp_knapp :- new Button(vnd, "Klikk her når du vil stoppe")
                    .PlaceAt(220, 200)
                    .Show;
    stopp_knapp.StartWaiting;
      ! StartWaiting setter stopp_knapp-en igang med å vente på at brukeren
      ! skal klikke knappen. Klassen Button inneholder en Boole'sk prosedyre
      ! IsWaiting som er sann (d.v.s. verdi true) inntil brukeren klikker knappen.
      ! Men så snart brukeren har gjort det, får IsWaiting verdien false;
    froe := 1237865;
    while stopp_knapp.IsWaiting do
      begin
        x2 := randint(0, vnd.width, froe);
          ! vnd.width er bredden (i pixeler) til vinduet vnd;
        y2 := randint(0, vnd.height, froe);
          ! vnd.height er høyden til vnd;
        vnd.DrawLine(x1, y1, x2, y2);
          ! Det tegnes en linje fra (x1, y1) til (x2, y2);
        x1 := x2; y1 := y2;
      end;
  end
```

3 Verktøy for utskrift og tegning

Windowtools-vinduer er av to hovedslag: *hovedvinduer* og *delvinduer*. Et program kan bare opprette ett hovedvindu, men gjerne mange delvinduer. Ethvert delvindu må høre til et annet vindu, som kan være et hovedvindu eller et delvindu.

Windowtools inneholder et antall klasser, de to viktigste er MainWindow – som definerer hovedvinduer – og SubWindow – som definerer delvinduer. Disse to klassene inneholder nøyaktig de samme prosedyrene, men har forskjellige parametre. MainWindow-objekter og SubWindow-objekter opprettes derfor på ulik vis.

I dette avsnittet beskrives først hvordan hovedvinduer (d.v.s. MainWindow-objekter) og delvinduer (SubWindow-objekter) opprettes. Deretter beskrives de viktigste prosedyrene som befinner seg i de to klassene.

3.1 Hvordan opprette vinduer

Klassen MainWindow har én parameter, en tekst, som benyttes som navn på vinduet. Dette navnet vil bli skrevet i et eget navne-felt som blir plassert over selve vinduet. Selve opprettelsen av et hovedvindu utføres av et eget program som kalles en *window-manager*. Det fins mange ulike slike "vindus-direktører" eller "vindus-styrere", og de oppretter vinduer hver på sin måte. Mange vindus-styrere gir brukeren anledning til å bestemme eller påvirke (ved bruk av mus) størrelse og plassering av hovedvinduer.

I et program som bruker Windowtools opprettes et hovedvindu slik:

new MainWindow(<*vindus-navn*>**);**

Vanligvis vil man ta vare på en referanse til et slikt vindus-objekt. Det kan gjøres slik:

<*hovedvindu-ref*>:– **new MainWindow(**<*vindus-navn*>**);**

Her må *hovedvindu-ref* være deklarert som en **ref(MainWindow)** variabel.

Klassen SubWindow har også bare en parameter, nemlig en referanse til et vindu (hovedvindu eller delvindu) som blir det nye vinduets eier. Et delvindu opprettes slik:

<*delvindu-ref*>:– **new SubWindow(**<*eiervindu-ref*>**);**

Her må *delvindu-ref* være deklarert som en **ref(SubWindow)** variabel, og *eiervindu-ref* ha som verdi en referanse til enten et SubWindow eller et MainWindow.

3.2 Hvordan gi størrelse til vinduer

MainWindow og SubWindow inneholder følgende tre prosedyrer som kan brukes til å gi størrelse til vinduer:

ref(XWindow) procedure SetSize(w, h); integer w, h;
<*Vinduet blir* w *pixeler bredt, og* h *pixeler høyt*>;

ref(XWindow) procedure SetWidth(w); integer w;
<*Vinduet blir* w *pixeler bredt*>;

ref(XWindow) procedure SetHeight(h); integer h;
<*Vinduet blir* h *pixeler høyt*>;

XWindow er en klasse som inneholder alt som er felles for de to klassene MainWindow og SubWindow (disse to klassene er deklarert som såkalte *subklasser* til XWindow). De tre prosedyrene ovenfor vil ha som verdi en peker til det vinduet som invokerer prosedyrene. Eksempler på hvordan dette kan brukes til såkalt *"fortsatt dotting"*, finnes bl.a. i de tre siste heihei-programmene i avsnitt 2.

3.3 Hvordan plassere vinduer

Det fins en rekke prosedyrer i XWindow (og dermed i både MainWindow og SubWindow) som kan brukes til å plassere vinduer. To av disse er PlaceAt og PlaceCentered:

ref(XWindow) procedure PlaceAt(x, y); integer x, y;
<*Vinduet plasseres med sin venstre kant* x *pixeler til høyre for eiers venstre kant og sin topp* y *pixeler nedenfor eiers topp*>;

ref(XWindow) procedure PlaceCentered(x, y); integer x, y;
<*Vinduet plasseres med sitt midtpunkt i* (x, y)>;

3.4 Hvordan gjøre vinduer synlige og usynlige

Prosedyrene Show og Unshow i MainWindow og SubWindow kan brukes til synlig-gjøring og usynlig-gjøring av vinduer:

ref(XWindow) procedure Show;
<*Vinduet blir synlig*>;

ref(XWindow) procedure Unshow;
<*Vinduet blir usynlig*>;

3.5 Hvordan få tak i størrelsen på vinduer

Vinduer har diverse attributter som kan brukes til å få tak i størrelse og plassering. De viktigste er disse:

integer width, height;
! width *er vinduets bredde (i pixeler),* height *er høyden*;

integer min_x, min_y;
! min_x *er avstand fra eiers venstre kant til vinduets venstre kant*
! min_y *er avstand fra eiers topp til vinduets topp*;

integer procedure max_x;
max_x := <*avstand fra eiers venstre kant til vinduets høyre kant*>;
integer procedure max_y;
max_x := <*avstand fra eiers topp til vinduets bunn*>;

3.6 Hvordan tegne linjer, rektangler og sirkler

MainWindow og SubWindow inneholder en lang rekke prosedyrer som kan brukes til å tegne linjer, punkter, sirkler og lignende. De viktigste av disse prosedyrene:

procedure DrawLine(x1, y1, x2, y2); integer x1, y1, x2, y2;
<*Tegn en linje fra* (x1, y1) *til* (x2, y2)>;

procedure DrawRectangle(x, y, width, height); integer x, y, width, height;
<*Tegn et rektangel med øvre venstre hjørne i* (x, y),
bredde width *og høyde* height >;

procedure DrawPoint(x, y); integer x, y;
<*Tegn et punkt i* (x, y)>;

procedure DrawPolygon(x, y, n); integer array x, y; integer n;
<*Tegn et polygon med hjørner i* (x(0), y(0)), (x(1), y(1)), . . . , (x(n−1), y(n−1))>;

procedure DrawCircle(center_x, center_y, radius); integer center_x, center_y, radius;
<*Tegn en sirkel med sentrum i* (center_x, center_y)>;

procedure DrawInvertLine(x1, y1, x2, y2); integer x1, y1, x2, y2;
<*Linjen fra* (x1, y1) *til* (x2, y2) *inverteres*>;

3.7 Hvordan bestemme "blekk" og linje-tykkelse

Følgende prosedyrer kan brukes til å bestemme hva slags "blekk" som brukes ved tegning i et vindu av linjer og annet, og til å bestemme hvor tykk strek som benyttes ved tegning;

ref(XWindow) procedure SetWhiteonBlack;
<*Heretter tegnes med hvitt "blekk"*>;

ref(XWindow) procedure SetBlackonWhite;
<*Heretter tegnes med sort "blekk"*>;

ref(XWindow) procedure SetLineWidth(linewidth); integer linewidth;
<*Linjetykkelse settes til* linewidth *pixeler*>;

Dersom disse prosedyrene ikke brukes i et vindu, tegnes med sort blekk og med linjetykkelse lik 1.

3.8 Hvordan skrive tekst

MainWindow og SubWindow inneholder følgende prosedyrer som kan brukes til å skrive tekst i et vindu:

procedure DrawText(x, y, t); integer x, y; text t;
<*Teksten* t *skrives med venstre kant i* x *og grunnlinje i* y>;

procedure DrawTextCenter(x, y, t); integer x, y; text t;
<*Teksten* t *skrives med sitt midtpunkt i* (x, y)>;

procedure DrawCenterText(t); text t;
<*Teksten* t *skrives midt i vinduet*>;

procedure DrawTextPos(x, y, t, hpos, vpos); integer x, y; text t; integer hpos, vpos;
<*Teksten* t *skrives. Posisjonen i forhold til* (x, y) *bestemmes av* hpos, vpos:
hpos = LeftPos: x *er venstre kant av teksten*
hpos = CenterPos: x *er midt på teksten*
hpos = RightPos: x *er høyre kant av teksten*
vpos = TopPos: y *er tekstens overkant*
vpos = CenterPos: y *er midt på teksten*
vpos = BottomPos: y *er tekstens grunnlinje.*
Her er CenterPos, RightPos, LeftPos, BottomPos *og* TopPos
heltallskonstanter som er deklarert i MainWindow *og* SubWindow>;

3.9 Hvordan bestemme font

Tekst kan skrives i vinduer med forskjellig *skriftsnitt* eller *font*. Det finnes et stort antall fonter (for tiden godt over 1000) som kan brukes i Windowtools-programmer. Disse fontene varierer på mange slags vis, blant annet med hensyn til størrelse og fethet, men også etter hvorvidt bokstavene skrives med eller uten seriffer, eller etter hvorvidt de skrives skråstilt eller loddrett.

Prosedyren SetFont kan brukes til å bestemme hvilken font som skal brukes ved utskrift av tekst i et vindu:

ref(XWindow) procedure SetFont(font_name); text font_name;
<*Font* font_name *brukes ved utskrift av tekst*>;

For å bruke denne prosedyren må man kjenne til navnene på en del aktuelle fonter. De fleste fonter har ganske lange og kompliserte navn, og det er ikke så helt enkelt å få en god oversikt over mulighetene. Den fonten som brukes i Windowtools-vinduer dersom ikke SetFont benyttes har for eksempel navn

```
"-adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso646-60"
```

Det finnes imidlertid en del fonter med korte og greie navn, f.eks. `"gallant.r.19"` og `"9x15"`.

Det finnes ferdiglagete programmer som kan brukes til å skaffe seg oversikt over tilgjengelige fonter og fontnavn. To av disse programmene heter xfontsel og xfbrows.

3.10 Hvordan finne størrelsen til tekst

Følgende funksjons-prosedyrer i MainWindow og SubWindow kan brukes til å finne ut horisontal og vertikal størrelse til tekst i den font som er i bruk i et vindu:

integer procedure width_of_text(t); text t;
  width_of_text := <*bredden på teksten* t *i den font som er i bruk*>;

integer procedure font_height;
  font_height := <*maksimal høyde på tegn i den font som er i bruk*>;

3.11 Hvordan tømme, invertere og fylle vinduer

Det finnes en del prosedyrer i XWindow (og dermed også i MainWindow og SubWindow) som kan brukes til å tømme, invertere og fylle vinduer og områder, og til å flytte områder. Tre av disse prosedyrene:

ref(XWindow) procedure Clear;
  <*Alt som er skrevet i vinduet fjernes*>;

ref(XWindow) procedure Invert;
  <*Vinduet inverteres (sort blir til hvitt og hvitt til sort)*>;

ref(XWindow) procedure Fill;
  <*Bunnen i vinduet fylles*>;

Normalt fylles et vindu med sort når Fill invokeres. Dersom man ønsker noe annet "fyll-blekk", man man bruke følgende prosedyre:

ref(XWindow) procedure SetFill(fillname); text fillname;
  <*Heretter brukes fyll-verdi definert av* fillname>;

Her kan teksten fillname være en av tekstene `"black"`, `"white"`, `"gray25"`, `"gray50"` eller `"gray75"`; eller navn på en fil som inneholder en såkalt *bitmap* (hva det er, blir ikke forklart i dette notatet. Den spesielt interesserte får finne ut av det på egen hånd). Virkningen av å utføre vnd.SetFill med et de fem fyll-navnene ovenfor burde være opplagt. SetFill(`"gray25"`) har for eksempel som virkning at all fylling i vinduet vnd foregår med en grå-tone som er 25% av veien fra hvitt til sort, altså lysegrå.

3.12 Hvordan tømme, fylle, invertere og flytte områder

Følgende prosedyrer i XWindow kan brukes til å tømme, fylle og invertere områder, og til å flytte områder:

procedure ClearRectangle(x, y, width, height); integer x, y, width, height;
  <*Tøm rektangelet med øvre venstre hjørne i* (x, y),
    bredde = width, *høyde* = height>;

procedure FillRectangle(x, y, width, height); integer x, y, width, height;
  <*Fyll rektangelet med øvre venstre hjørne i* (x, y),
    bredde = width, *høyde* = height>;

procedure InvertRectangle(x, y, width, height); integer x, y, width, height;
  <*Inverter rektangelet med øvre venstre hjørne i* (x, y),
    bredde = width, *høyde* = height>;

procedure CopyRectangle(srcx, srcy, width, height, destx, desty);
  integer srcx, srcy, width, height, destx, desty;
  <*Kopier rektangelet med øvre venstre hjørne i* (srcx, srcy),
    bredde = width, *høyde* = height *til* (destx, desty)>;

procedure FillPolygon(x, y, n); integer array x, y; integer n;
  <*Fyll polygonet med hjørner i* (x(0), y(0)), (x(1), y(1)), ..., (x(n−1), y(n−1))>;

procedure FillCircle(center_x, center_y, radius); integer center_x, center_y, radius;
  <*Fyll sirkelen med sentrum i* (center_x, center_y)>;

4 Vente-verktøy

Klassen Button, og også klassen FancyButton, inneholder de samme prosedyrene som SubWindow (Button og FancyButton er deklarert som subklasser av SubWindow), og dessuten følgende 6 spesielle prosedyrer.

procedure Wait;
<*Denne* Button-*en blir synlig. Deretter venter programmet inntil brukeren flytter kursøren inn i knappen og klikker en muse-knapp*>;

procedure StartWaiting;
<*Starter venting. Inntil brukeren klikker i denne* Button-*en eller prosedyren* StopWaiting *utføres, har prosedyren* StillWaiting **true** *som verdi*>;

Boolean procedure IsWaiting;
<*Er sann inntil brukeren klikker denne* Button *eller* StopWaiting *utføres*>;

procedure StopWaiting;
<*Avslutter venting*>;

procedure SetActive;
<*Det markeres visuelt at denne knappen er aktiv*>;

procedure SetInactive;
<*Det markeres at denne knappen ikke er aktiv*>;

Klassene Button og FancyButton har *to* parametere: først en referanse til et vindu (eier-vinduet), dernest en tekst. Dermed opprettes en "knapp" som følger:

<*knappe-ref*>:– **new Button(**<*eiervindu-ref*>, <*knappe-tekst*>**);**

eller

<*knappe-ref*>:– **new FancyButton(**<*eiervindu-ref*>, <*knappe-tekst*>**);**

Her må *knappe-ref* være deklarert som en **ref(Button)** variabel (evt. **ref(FancyButton)**) og *eier-vindu-ref* ha som verdi en referanse til enten et SubWindow eller et MainWindow. Teksten <*knappe-tekst*> vil bli skrevet midt på knappen når knappen blir synlig på skjermen.

Forskjellen på Button og FancyButton er visuell: En FancyButton ser tredimensjonal ut, mens en vanlig Button er flat.

5 Prompt-verktøy

Klassen PromptWindow er en subklasse av SubWindow, og inneholder derfor de samme attributtene som SubWindow. I tillegg inneholder den prosedyrer som kan brukes til å be om og lese tekster, tall, logiske verdier og enkelt-tegn:

text procedure ask_for_text(prompt, max_length);
text prompt; integer max_length;
<*Dette* PromptWindow-*et blir synlig. Det inneholder fire felter:*
–: *Et hvor ledeteksten* prompt *er skrevet,*
–: *Et "svar-felt", med plass til en tekst med lengde* max_length,
–: *En* Button *med teksten* OK,
–: *En* Button *med teksten* Cancel
Brukeren kan flytte mus-markøren inni svar-feltet og skrive en tekst.
Når brukeren er fornøyd, flytter brukeren markøren til OK-*knappen*
og trykker en muse-knapp. En referanse til et tekst-objekt som inneholder
teksten som står i svar-feltet blir verdi til ask_for_text.
Dersom brukeren klikker i Cancel-*knappen, brukes* **notext** *som verdi*>;

integer procedure ask_for_int(prompt); text prompt;
< *Som* ask_for_text, *men svaret må være et heltall* >;

real procedure ask_for_real(prompt); text prompt;
< *Som* ask_for_int, *men svaret må være et reelt tall* >;

Boolean procedure ask_for_Bool(prompt); text prompt;
< *Som* ask_for_text, *men svaret må være en logisk verdi* >;

character procedure ask_for_char(prompt); text prompt;
< *Som* ask_for_text, *men svaret må være et tegn* >;

6 Et *litt* mer komplisert eksempel

En kopi av følgende program befinner seg på filen `$i110/filer/forelesning/slide0.sim`

```
begin
  external class Windowtool2;

  ref(MainWindow)   kontroll_vnd;
  ref(SubWindow)    blokk_vnd;
  ref(PromptWindow) be_vnd;
  ref(Button)       storrelse_knapp,
                    omstart_knapp,
                    stopp_knapp;

  integer blokkw = 35, blokkh = 35,
          maxx, maxy, minx, miny,
          xsteg, ysteg, xpos, ypos;

  procedure InverterBlokk;
    begin
      blokk_vnd . InvertRectangle(xpos, ypos, blokkw, blokkh);
      if xpos >= maxx then xsteg := -1 else
      if xpos <= minx then xsteg := +1;
      if ypos >= maxy then ysteg := -1 else
      if ypos <= miny then ysteg := +1;
      xpos := xpos + xsteg;
      ypos := ypos + ysteg;
    end;

  procedure Bestemm_Storrelse;
    begin
      blokk_vnd . SetSize(be_vnd . ask_for_int("Bredde?"),
                          be_vnd . ask_for_int("Høyde?"));
      maxx := blokk_vnd . width - blokkw; minx := 0;
      maxy := blokk_vnd . height - blokkh; miny := 0;
      storrelse_knapp . StartWaiting;
      Omstart;
    end;

  procedure Omstart;
    begin
      blokk_vnd . Clear;
      xpos := 0; ypos := 0;
      xsteg := 1; ysteg := 1;
      omstart_knapp . StartWaiting;
```

```
    end;

  kontroll_vnd     :- new MainWindow("Slide")
                          . PlaceAt(50, 50)
                          . SetSize(600, 500);
  storrelse_knapp :- new Button(kontroll_vnd, "Ny størrelse")
                          . PlaceAt(10, 10)
                          . Show;
  omstart_knapp   :- new Button(kontroll_vnd, "Omstart")
                          . PlaceAt(storrelse_knapp . max_x + 10, 10)
                          . Show;
  stopp_knapp     :- new Button(kontroll_vnd, "Stopp")
                          . PlaceAt(omstart_knapp . max_x + 10, 10)
                          . Show;
  blokk_vnd       :- new SubWindow(kontroll_vnd)
                          . PlaceAt(10, storrelse_knapp . max_y + 10)
                          . Show;
  be_vnd          :- new PromptWindow(kontroll_vnd)
                          . PlaceAt(10, storrelse_knapp . max_y + 10);
  kontroll_vnd . Show;
  Bestemm_Storrelse;
  stopp_knapp . StartWaiting;
  whilestopp_knapp . IsWaiting do
    begin
      if not storrelse_knapp . IsWaiting then Bestemm_Storrelse else
      if not omstart_knapp . IsWaiting then Omstart
      else InverterBlokk;
    end;

end
```