```
MODULE Stars;   (*NW 15.1.2013, 15.11.2013*)
  (* Stars.Open  Stars.Step  Stars.Run Stars.Stop  ORG.Compile @ *)
  IMPORT SYSTEM, Display, Viewers, Texts, Oberon, MenuViewers, TextFrames;

  CONST N = 6;  (*nof stars*)
    w = 16;   (*width of star*)
    interval = 200;  (*millisec*)

  TYPE Frame = POINTER TO FrameDesc;
    Pos = RECORD x, y, dx, dy: INTEGER END ;
    FrameDesc = RECORD (Display.FrameDesc) s: ARRAY N OF Pos END ;
    RestoreMsg = RECORD (Display.FrameMsg) END ;
    StepMsg = RECORD (Display.FrameMsg) END ;

  VAR T: Oberon.Task;
    W: Texts.Writer;

  PROCEDURE Draw(x, y: INTEGER);
  BEGIN Display.CopyPattern(Display.white, Display.star, x, y, Display.invert)
  END Draw;

  PROCEDURE Restore(F: Frame);
    VAR x0, y0: INTEGER;
  BEGIN Oberon.RemoveMarks(F.X, F.Y, F.W, F.H);
    Display.ReplConst(0, F.X+1, F.Y, F.W-1, F.H, 0);
    x0 := F.W DIV 2 + F.X; y0 := F.H DIV 2 + F.Y;
    F.s[0].x := x0; F.s[0].y := y0; F.s[0].dx := 2; F.s[0].dy := 4; Draw(F.s[0].x, F.s[0].y);
    F.s[1].x := x0; F.s[1].y := y0; F.s[1].dx := 3; F.s[1].dy := 9; Draw(F.s[1].x, F.s[1].y);
    F.s[2].x := x0; F.s[2].y := y0; F.s[2].dx := -5; F.s[2].dy := -2; Draw(F.s[2].x, F.s[2].y);
    F.s[3].x := x0; F.s[3].y := y0; F.s[3].dx := -10; F.s[3].dy := 8; Draw(F.s[3].x, F.s[3].y);
    F.s[4].x := x0; F.s[4].y := y0; F.s[4].dx := -7; F.s[4].dy := -4; Draw(F.s[4].x, F.s[4].y);
    F.s[5].x := x0; F.s[5].y := y0; F.s[5].dx := 8; F.s[5].dy := -10; Draw(F.s[5].x, F.s[5].y)
  END Restore;

  PROCEDURE Move(F: Frame; VAR p: Pos);
    VAR X1, Y1: INTEGER;
  BEGIN X1 := F.X + F.W - w; Y1 := F.Y + F.H - w;
    Draw(p.x, p.y); INC(p.x, p.dx); INC(p.y, p.dy);
    IF p.x < F.X THEN p.x := 2*F.X - p.x; p.dx := -p.dx
    ELSIF p.x >= X1 THEN p.x := 2*X1 - p.x; p.dx := -p.dx
    END ;
    IF p.y < F.Y THEN p.y := 2*F.Y - p.y; p.dy := -p.dy
    ELSIF p.y  >= Y1 THEN p.y := 2*Y1 - p.y; p.dy := -p.dy
    END ;
    Draw(p.x, p.y)
  END Move;

  PROCEDURE Steps(F: Frame);
    VAR i: INTEGER;
  BEGIN i := 0;
    WHILE i < N DO Move(F, F.s[i]); INC(i) END
  END Steps;

  PROCEDURE Handle(F: Display.Frame; VAR M: Display.FrameMsg);
    VAR F1: Frame;
  BEGIN
    CASE F OF Frame:
      CASE M OF
        Oberon.InputMsg:
          IF M.id = Oberon.track THEN
            Oberon.DrawMouseArrow(M.X, M.Y)
          END
      | StepMsg: Steps(F)
      | RestoreMsg: Restore(F)
      | Oberon.CopyMsg: Oberon.Remove(T); NEW(F1); F1^ := F^; M.F := F1
      | MenuViewers.ModifyMsg:
        IF (M.Y # F.Y) OR (M.H # F.H) THEN F.Y := M.Y; F.H := M.H; Restore(F) END
      END
    END
  END Handle;
  PROCEDURE Step*;
    VAR k: INTEGER; M: StepMsg;
  BEGIN
    IF Oberon.Par.vwr.dsc = Oberon.Par.frame THEN Steps(Oberon.Par.frame.next(Frame))
    ELSE Viewers.Broadcast(M)
    END
  END Step;

  PROCEDURE Open*;
    VAR F: Frame; V: Viewers.Viewer; X, Y: INTEGER;
  BEGIN NEW(F); F.handle := Handle;
    Oberon.AllocateUserViewer(Oberon.Par.vwr.X, X, Y);
    V := MenuViewers.New(
           TextFrames.NewMenu("Stars",
             "Stars.Close  System.Grow  System.Copy  Stars.Step  Stars.Run  Stars.Stop"),
           F, TextFrames.menuH, X, Y)
  END Open;

  PROCEDURE Run*;
  BEGIN Oberon.Install(T)
  END Run;

  PROCEDURE Stop*;
  BEGIN Oberon.Remove(T)
  END Stop;

  PROCEDURE Close*;
  BEGIN
    IF Oberon.Par.vwr.dsc = Oberon.Par.frame THEN Stop; Viewers.Close(Oberon.Par.vwr) END
  END Close;

  PROCEDURE Step1;
    VAR M: StepMsg;
  BEGIN Viewers.Broadcast(M)
  END Step1;

  PROCEDURE SetPeriod*;
    VAR S: Texts.Scanner;
  BEGIN Texts.OpenScanner(S, Oberon.Par.text, Oberon.Par.pos); Texts.Scan(S);
    IF S.class = Texts.Int THEN T.period := S.i END
  END SetPeriod;

BEGIN Texts.OpenWriter(W); T := Oberon.NewTask(Step1, interval);
END Stars.
```