Datumsoperationen Turbo Pascal 5.5 / 6.0

ältere Version
neue s. DTCNV1

```
unit dtcnv;

interface

type
 mtype = array[0..12] of integer;
 WdN = array[1..7] of string[2];

 year = object
      day, month, yr, wday : integer;
      cday : LongInt;
      wdayName : string[2];
      msg : boolean;  { flag für Meldung wenn yr = 0 }
      constructor Init;
      destructor Done;
      procedure SetDatum(j,m,t : integer);
      procedure SetCday(cd : LongInt);
      procedure DayOfWeek;
     end;

 hjyear = object(year)
              ep, sj : integer;
              constructor Init;
              procedure CalcCday; virtual;
              procedure CalcDatum; virtual;
              procedure ChangeSetting( e,s : integer);
              end;

 adyear = object(year)
              procedure CalcCday; virtual;
              procedure CalcDatum; virtual;
              end;

implementation

 const Gregein = 577735;
       Sisal = 10631;
       Hegire = 227014;
       MaxInt = 32767;
       WdayNms : WdN = ('SO','MO','DI','MI','DO','FR','SA');
       ma : mtype = (0,31,59,90,120,151,181,212,243,273,304,334,365);
       mb : mtype = (710,1773,2482,3545,4608,5317,6380,7443,8506,9215,10278
       mh : mtype = (2,5,7,10,13,0,18,21,24,26,29,MaxInt,0);



function floor(a,b : LongInt): LongInt;
 begin
   if (a < 0) and (b > 0) or (a > 0) and (b < 0)
      then floor := a div b - 1
      else floor := a div b
 end; {floor}

constructor year.Init;
```

```
  begin
    day := 1; month := 1; yr := 1; wday := 7;
    cday := -1; wdayName :=  'SA'; msg := FALSE;
  end;

destructor year.Done;
     begin end;

procedure year.SetDatum;
  begin
    yr := j;
    month := m;
    day := t;
  end;

procedure year.SetCday;
  begin
    cday := cd;
  end;

procedure year.DayOfWeek;
  begin
    if cday >= 0 then wday := cday mod 7+1
                 else wday := 8-(abs(cday) mod 7);
    wdayName := WdayNms[wday];
  end;

constructor hjyear.Init;
  begin
    year.Init;
    ep := 15;
    sj := 15;
  end;

procedure hjyear.ChangeSetting;
  begin
    ep := e;
    sj := s;
  end;

procedure hjyear.CalcCday;
 var mhig : mtype;
     dh,tage,l,s,j,k,m,t,u,v,cyc : LongInt;

 procedure accumulate; { Monate in Tage,
                         Anzahl Tage alterniert zw. 30 u. 29 }
  var x,q : integer;
    begin
      q := 1;              { Startwert Anzahl verg. Monate}
      for x := 1 to 6 do
          begin
            u := u + 30;
            q := q + 1;
            if m < q then exit
                     else begin
```

```
                              u := u + 29;
                              q := q + 1;
                              if m < q then exit
                            end;
          end
    end; { accumulate }

 begin
   mhig := mh;
   j := yr;
   m := month - 1;
   t := day;
   if ep = 16 then dh := Hegire else dh := Hegire-1;
   if j = 0 then begin
                   msg := TRUE;
                   j := 1;
                   yr := 1;
                 end
            else msg := FALSE;
    if j < 0 then j := j + 1;
    k := j - 1;
    cyc := floor(k,30); { Anzahl 30jahreszyklen }
    tage := Sisal*cyc;
    k := k-cyc*30;      { Jahre im aktuellen Zyklus }
    mhig[5] := sj;      { Einsetzen Schaltjahr }
    v := -1;            { Startwert }
    repeat
      v := v+1;         { Schalttage }
    until k < mhig[v];
    u := 0;             { Vergangene Monate in Tagen }
    if m >= 1 then accumulate;
    cday := tage + k*354 + v + u + t + dh;
 end;

procedure hjyear.CalcDatum;
  var mbd : mtype;
      dh,tage,y,j,k,m,t,cyc,s,v : LongInt;
      lr,yrl,vr : real;

  procedure countmonths(var m : LongInt; s: LongInt);
   var x,q : integer;
    begin
      q := 60;
      for x := 1 to 5 do
         begin
           m := m+1;
           if s < q then exit;
           q := q+30;
           m := m+1;
           if s < q then exit;
           q := q+29
         end;
         m := m+1
    end; { countmonths }
```

```
    begin
     mbd := mb;
     tage := cday;
     if ep = 16 then dh := Hegire else dh := Hegire - 1;
     y := tage-dh;
     cyc := floor(y,Sisal);          { 30jahreszyklen }
     y := y-Sisal*cyc;               { Tage im aktuellen Zyklus }
     if y = 0 then begin             { letzter Tag im Zyklus }
                     cyc := cyc-1;
                     y := Sisal
                   end;

     k := 30*cyc;                          { Anzahl Jahre der verg. Zyklen }
     t := -1;                              { Startwert }
     if sj = 16 then mbd[5] := 5671  { Schaltjahr wird berücksichtigt }
                else mbd[5] := 5317;
     repeat
        t := t+1                           { Schalttage }
     until y < mbd[t];
     if y < 1418
         then lr := 0.9985
         else if y < 2482
                 then lr := 0.9986
                 else if (sj = 15) and (y = 5316) or (y < 4253)
                         then lr := 0.9988
                         else lr := 0.9989;
     yrl := y;                              { Umrechnung von Tagen in Jahre }
     vr := lr*yrl/354.0;
     v := trunc(vr);
     j := k+v+1;
     s := y-354*v-t;                        { Verbleibende Tage im Jahr }
     m := 1;                                { Startwert Monate }
     if s >= 31 then countmonths(m,s);
     t := s-trunc((m-1)*29.5+0.5);
     if j = 0 then msg := TRUE else msg := FALSE;
     if j <= 0 then j := j-1;
     yr := j;
     month := m;
     day := t;
    end;

procedure adyear.CalcCday;
 var mad : mtype;
     k,l,tage,q,v,y,w,j,m,t : longint;

begin
     mad := ma;
     j := yr;
     m := month - 1;
     t := day;
     if j = 0 then begin
                     msg := TRUE;
                     j := 1;
                     yr := 1
                   end
```

```
            else msg := FALSE;
    if j < 0 then j := j+1;
    k := j-1;
    v := floor(k,400);
    y := 400*v;
    tage := 365*y+97*v;
    k := k-y;
    v := floor(k,100);
    y := 100*v;
    tage := tage+365*y+24*v;
    k := k-y;
    v := floor(k,4);
    y := 4*v;
    tage := tage+365*y+v;
    if ((((j mod 4) = 0) and ((j mod 100) <> 0) or ((j mod 400 = 0)))
       and (m > 1))  then w := t+1
                     else w := t;
    l := mad[m];
    tage := tage+(k-y)*365+l+w;  {gregorianisch}
    if tage <= Gregein then begin
                    if ((j mod 4) = 0) and (m > 1)
                       then w:= t+1
                       else w := t;
                    k := j-1;
                    tage := k*365+floor(k,4)+l+w-2;
                   end;
    cday := tage;
end;

procedure adyear.CalcDatum;
  var mad : mtype;
      k,l,s,v,y,tage,j,m,t : LongInt;

  procedure vierhundert;
   begin
    l := 146097;
    y := floor(tage,l);
    j := 400*y;
    k := tage-y*l;
   end; {vierhundert}

  procedure hundert;
   begin
     l := 36524;
     y := floor(k,l);
     j := j+100*y;
     k := k-y*l;
     y := floor(k,1461);
     j := j+4*y;
     k := k-y*1461
   end; {hundert}

  procedure ersterjanuar;
   begin
     y := floor(k,365);
```

```
    j := j+y+1;
    k := k-y*365;
    if k = 0 then begin j := j-1; k := 365 end
  end; {ersterjanuar}

procedure vhschalttag;
  begin
    if (k = 0) and ((j mod 400) = 0) then ersterjanuar;
    if (k = 0) and ((j mod 100) > 0) then k := 366
                                     else ersterjanuar

  end; {vhschalttag}

procedure vorgregor;
  begin
    y := floor(tage,1461);
    j := 4*y;
    k := tage-y*1461+2;
    if k = 1461 then begin k := 366; j := j+4 end
                else begin
                       if k = 1462 then k := 1461;
                       ersterjanuar
                     end
  end;   {vorgregor}

begin
  mad := ma;
  tage := cday;
  if tage <= Gregein then vorgregor
             else begin
                    vierhundert;
                    if k = 0 then k := 366
                             else begin
                                    hundert;
                                    vhschalttag
                                  end
                  end;
  s := 0;
  k := k-1;
  m := 1;
  v := 0;
  if k >= 31 then begin
                    if (j mod 4) <= 0 then begin
                                             s := 1;
                                             if ((j mod 100) = 0)
                                                and ((j mod 400) > 0)
                                                and (tage > Gregein)
                                                then s := 0
                                           end;
                    repeat
                     m := m+1;
                     if m = 2 then v := 31-s
                              else v := mad[m-1]
                    until k < (mad[m]+s)
                  end;
  t := k-v-s+1;
```

```
    if j = 0 then msg := TRUE else msg := FALSE;
    if j <= 0 then j := j-1;
    yr := j;
    month := m;
    day := t;
  end;

end.
```