Optimised replacements for DecodeDate and EncodeDate

Here are two replacements for Delphi's *DecodeDate* and *EncodeDate* routines that are optimised for speed.

```
procedure OptDecodeDate(const DateTime: TDateTime;
  out Year, Month, Day: Integer);
var
  J: Integer;
begin
  J := Pred((Trunc(DateTime) + 693900) shl 2);
  Year := J div 146097;
  Day := (J - 146097 * Year) shr 2;
  J := (Day shl 2 + 3) div 1461;
  Day := (Day shl 2 + 7 - 1461 * J) shr 2;
  Month := (5 * Day - 3) div 153;
  Day := (5 * Day + 2 - 153 * Month) div 5;
  Year := 100 * Year + J;
  if Month < 10 then
    Inc(Month, 3)
  else
  begin
    Dec(Month, 9);
    Inc(Year);
  end;
end;
```

```
function OptEncodeDate(Year, Month, Day: Integer): TDateTime;
var
  DayTable: PDayTable;
begin
  DayTable := @MonthDays[IsLeapYear(Year)];
  if (Day <= DayTable^[Month]) and (Year > 0) and (Year < 10000) and
    (Month < 13) and (Day > 0) then
  begin
    if Month > 2 then
      Dec(Month, 3)
    else if Month > 0 then
    begin
      Inc(Month, 9);
      Dec(Year);
    end
    else // Month <= 0
      raise Exception.Create('Invalid Date');
    Result := (146097 * (Year div 100)) shr 2 +
      (1461 * (Year - 100 * (Year div 100))) shr 2 +
      (153 * Month + 2) div 5 + Day - 693900;
  end
  else
    raise Exception.Create('Invalid Date' + IntToStr(Year) + '-'
      + IntToStr(Month) + '-' + IntToStr(Day));
end;
```

The routines require the *SysUtils* unit.

Author:	Unknown
Contributor:	Bill Miller
Added:	2014-01-22
Last updated:	2014-01-22