

Aufgabenstellung:

Schreibt ein Programm, welches aus einem beliebigen Delphi Quelltext alle Kommentare, überflüssige Leerzeichen und Leerzeilen entfernt. Dies hat nach folgenden Regeln zu geschehen:

1. Entfernen der Kommentare

Unter Delphi gibt es grundsätzlich drei Möglichkeiten mit der ein Programmierer einen Kommentar einleiten kann:

- (a) Einzeiliger Kommentar: `// Dies ist ein Kommentar`
 Dieser Kommentar endet mit dem Zeilenende und benötigt kein abschließendes Zeichen. Um diesen Kommentar zu entfernen muß man lediglich nach `//` suchen und alles bis zum Zeilenende löschen.
- (b) Mehrzeiliger Kommentar: `{Dies ist ein Kommentar}`
 oder `(* Dies ist auch ein Kommentar *)`
 Die mehrzeiligen Kommentare unterscheiden sich von (a) um die zusätzliche Endemarkierung, woraus folgt, daß diese sich über mehrere Zeilen erstrecken können. Eine Schachtelung von `{ }` und `(* *)` ist möglich, wobei Delphi aber nur die äußersten Kommentarzeichen berücksichtigt. D.h. folgendes Konstrukt ist syntaktisch korrekt:
`{ Kommentar (* (* } bzw.`
`(* Noch ein Kommentar { { { (* *)`
 Die äußersten Kommentarzeichen müssen vom selben Kommentartyp sein, d.h. folgendes Konstrukt kann in Delphi nicht auftreten:
`(* (* { Kommentar }`
 Tritt nun z.B. in einer Quelltextzeile ein öffnender Kommentar auf, so sind alle Zeichen inkl. der Kommentarzeichen (auch in folgenden Zeilen) zu entfernen, bis das schließende Kommentarzeichen auftritt.

2. Entfernen der Leerzeichen

Es können nicht alle redundanten Leerzeichen entfernt werden, da man hierfür einen Parser für Delphi schreiben müßte. Um dies zu vermeiden, sollten folgende Regeln verwendet werden:

- (a) Treten mehrere Leerzeichen in einer Zeile direkt hintereinander auf, so ist deren Länge auf 1 Leerzeichen zu reduzieren.
- (b) Befinden sich mehrere Leerzeichen innerhalb einer Stringkonstante, also z.B.

```
var s:string;
begin
...
s := `Hallo  Welt!`;
...
end.
```

 so sind diese unverändert zu belassen.
- (c) Alle Leerzeichen am Anfang einer Zeile sind zu entfernen.

Ein Leerzeichen kann entweder ein echtes Leerzeichen (ANSI-Code 20h) oder ein Tabulatorzeichen (ANSI-Code 0Bh) sein.

3. Entfernen der Leerzeilen

Leerzeilen werden einfach nicht in die Zielfile kopiert. Leerzeilen sind Zeilen die nur Leerzeichen (Space, Tab) oder gar nichts enthalten. Zu beachten ist, daß Leerzeilen auch nach der Entfernung eines Kommentars auftreten können.

4. Statistik

Erweitert Euer Programm um die Möglichkeit, folgende Statistik in eine dritte Textdatei und auf das Formular zu schreiben:

- Anzahl der aus der Quelldatei gelesenen Zeilen
- Anzahl der gelöschten Leerzeilen
- Anzahl der gelöschten Leerzeichen
- Anzahl der gelöschten Kommentare in alle drei Typen aufgeschlüsselt.

Das Programm soll die Möglichkeit besitzen, zur Laufzeit einen Delphi Quelltext zu spezifizieren, der dann in eine zweite spezifizierbare Datei nach obigen Regeln transformiert wird.

Lagert alle Prozeduren/Funktionen, die nicht zur Formularein-/ausgabe gehören in **eine eigene Unit** aus.

Hinweise und Tips:

- Für die Eingabe der Dateinamen können einfache Editboxen verwendet werden. Ihr dürft selbstverständlich auch die Standarddateidialoge von Windows (*TOpenDialog* und *TSaveDialog*) verwenden.
- Das zeilenweise Einlesen aus der Quelldatei, Bearbeiten und Schreiben in die Zieldatei hat sich als äußerst praktisch erwiesen.
- Beachtet, daß Kommentare und Schlüsselwörter innerhalb einer Stringkonstante **keine** "echten" Kommentare bzw. Schlüsselwörter sind. (Gleiches gilt für Schlüsselwörter innerhalb von Kommentaren.)
- Achtung, die Zeichenfolge { \$ leitet eine Compiler-Anweisung ein, die durch } wieder beendet wird! Es handelt sich hierbei also **nicht** um einen Kommentar (P **nicht** entfernen!)
- Einige nützliche ANSI Codes:
(Space=20h, Tab=0Bh, Hochkomma=27h)

Beispiel Quelltext vor und nach der Bearbeitung mit der Musterlösung für diese

Übung:

```
{*****
* Unitname      : devinfo.pas
* *****}
unit Devinfo;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, DrvHnd, Grids, StdCtrls, Sgrid;

type
  TFormDevInfo = class(TForm)
    buttonClose: TButton;
    sgDriverInformation: TStringGridHinted;
    procedure buttonCloseClick(Sender: TObject);
  private
    { Private declarations }
    { This procedure initializes the driver information }
    procedure InitDriverInfo;
  public
    { Public declarations }
  end;

implementation
{$R *.DFM}

procedure TFormDevInfo.InitDriverInfo;
var Metrics:TTextMetric;
    iIndex:integer;
    iChannelIndex:integer;
begin
  {Calculate the needed row height in dependence of the current font}
  GetTextMetrics(sgDriverInformation.Canvas.Handle, Metrics);
  sgDriverInformation.DefaultRowHeight := Metrics.tmHeight+Metrics.tmExternalLeading;

  {Calculate the column width of the stringgrid}
  sgDriverInformation.ColWidths[0] := sgDriverInformation.Width div 3;

  with sgDriverInformation, DriverHandler.DeviceInformation do begin
```

```
    Cells[0,0] := 'Driverversion';    Cells[1,0] := drvVersion;
    iIndex := 7;
    for iChannelIndex := 1 to devChannelCount do
      Cells[0,iIndex+iChannelIndex-1] := 'Channelname #' +IntToStr(iChannelIndex);
    end;
  end;
end;

{=====}
procedure TFormDevInfo.buttonCloseClick(Sender: TObject);
begin
  close
end;
end.

-----

unit Devinfo;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls, Forms, Dialogs, DrvHnd, Grids, StdCtrls, Sgrid;
type
  TFormDevInfo = class(TForm)
    buttonClose: TButton;
    sgDriverInformation: TStringGridHinted;
    procedure buttonCloseClick(Sender: TObject);
  private
    procedure InitDriverInfo;
  public
  end;
  implementation
  {$R *.DFM}
  procedure TFormDevInfo.InitDriverInfo;
  var Metrics:TTextMetric;
      iIndex:integer;
      iChannelIndex:integer;
  begin
    GetTextMetrics(sgDriverInformation.Canvas.Handle, Metrics);
    sgDriverInformation.DefaultRowHeight := Metrics.tmHeight+Metrics.tmExternalLeading;
    sgDriverInformation.ColWidths[0] := sgDriverInformation.Width div 3;
    with sgDriverInformation, DriverHandler.DeviceInformation do begin
      Cells[0,0] := 'Driverversion'; Cells[1,0] := drvVersion;
      iIndex := 7;
      for iChannelIndex := 1 to devChannelCount do
        Cells[0,iIndex+iChannelIndex-1] := 'Channelname #' +IntToStr(iChannelIndex);
      end;
    end;
  end;
  procedure TFormDevInfo.buttonCloseClick(Sender: TObject);
  begin
    close
  end;
end.
end.
```