Lösungen zu den Übungsaufgaben zu Verbunden (Records)

Aufgabe 1: Bruchrechnung

```
type tbruch = record // Type Bruch als Record vereinbaren
                z, n : integer; // z-Zähler, n-Nenner
              end;
var b1, b2: tbruch; // Variablen vom Typ Tbruch deklarieren

function ggt(a, b: integer): integer; // Verfahren von EUKLID
var
  r: integer;
begin
  a := abs(a);
  b := abs(b);
  while b <> 0 do
  begin
    r := a mod b;
    a := b;
    b := r;
  end;
  result := a;
end;

function kuerzen(b: tbruch): tbruch; // Bruch kürzen
var
  teiler: integer;
begin
  teiler := ggt(b.z, b.n);
  kuerzen.z := b.z div teiler;
  kuerzen.n := b.n div teiler;
end;

function addbruch(b1, b2: tbruch): tbruch; // Brüche addieren
begin
  result.z := (b1.z*b2.n)+(b2.z*b1.n);
  result.n := b1.n*b2.n;
end;

function subbruch(b1, b2: tbruch): tbruch; // Brüche subtrahieren
begin
  result.z := (b1.z*b2.n)-(b2.z*b1.n);
  result.n := (b1.n*b2.n);
end;

function multbruch(b1, b2: tbruch): tbruch; // Brüche multiplizieren
begin
  result.z := b1.z*b2.z;
  result.n := b1.n*b2.n;
end;

function divbruch(b1, b2: tbruch): tbruch; // Brüche dividieren
begin
  result.z := b1.z*b2.n;
  result.n := b1.n*b2.z;
end;
```

```
// Bruch für Ausgabe in String umwandeln
function BruchToString(b: tbruch): string;
begin
  result := IntToStr(b.z) + ' / ' + IntToStr(b.n);
end;

procedure TForm1.Button1Click(Sender: TObject); // Addition
var
  summe: tbruch;
begin
  b1.z := StrToInt(Edit1.Text);
  b1.n := StrToInt(Edit2.Text);
  b2.z := StrToInt(Edit3.Text);
  b2.n := StrToInt(Edit4.Text);
  summe := addbruch(b1, b2);
  summe := kuerzen(summe);
  Edit5.Text := BruchToString(summe);
end;

procedure TForm1.Button2Click(Sender: TObject); // Subtraktion
var
  diff: tbruch;
begin
  b1.z := StrToInt(Edit1.Text);
  b1.n := StrToInt(Edit2.Text);
  b2.z := StrToInt(Edit3.Text);
  b2.n := StrToInt(Edit4.Text);
  diff := subbruch(b1, b2);
  diff := kuerzen(diff);
  Edit5.Text := BruchToString(diff);
end;

procedure TForm1.Button3Click(Sender: TObject); // Multiplikation
var
  prod: tbruch;
begin
  b1.z := StrToInt(Edit1.Text);
  b1.n := StrToInt(Edit2.Text);
  b2.z := StrToInt(Edit3.Text);
  b2.n := StrToInt(Edit4.Text);
  prod := multbruch(b1, b2);
  prod := kuerzen(prod);
  Edit5.Text := BruchToString(prod);
end;

procedure TForm1.Button4Click(Sender: TObject); // Division
var
  quotient: tbruch;
begin
  b1.z := StrToInt(Edit1.Text);
  b1.n := StrToInt(Edit2.Text);
  b2.z := StrToInt(Edit3.Text);
  b2.n := StrToInt(Edit4.Text);
  quotient := divbruch(b1, b2);
  quotient := kuerzen(quotient);
  Edit5.Text := BruchToString(quotient);
end;
```

Aufgabe 2: Arbeiten mit Adressen

```
type TName = record  // Record für Name
               vorname: ShortString;
               nachname: ShortString;
     end;
     TAnschrift = record  // Record für Anschrift
                    plz: ShortString;
                    ort: ShortString;
                    strasse_hausnummer: ShortString;
     end;
     TPerson = record  // Record für Person
                 anrede: ShortString;
                 name: TName;
                 anschrift: TAnschrift;
     end;

var  // Festlegung der globalen Variablen
    feld:array[1..10] of TPerson;  // Feld zum Speichern der Personen
    p:TPerson;

const anz:integer=1;  // globaler Zähler zum Zählen der erfassten Personen

procedure erfassen(var ds:TPerson);  // Erfassen der Daten einer Person
begin
   p.anrede := Form1.Edit1.Text;
   p.name.vorname := Form1.Edit2.Text;
   p.name.nachname := Form1.Edit3.Text;
   p.anschrift.plz := Form1.Edit4.Text;
   p.anschrift.ort := Form1.Edit5.Text;
   p.anschrift.strasse_hausnummer := Form1.Edit6.Text;
end;

procedure saeubern;  // Edit-Komponenten säubern zur weiteren Datenerfassung
begin
with form1 do begin
  Edit1.Text:=' ';
  …
  Edit6.Text:=' ';
  end;
end;

procedure TForm1.Button1Click(Sender: TObject);  // Datensätze erfassen
begin
  if anz > 10 then  // Prüfen, ob Höchstzahl der Datensätze überschritten ist
  begin
    saeubern;
    ShowMessage('Zulässige Anzahl der Datensätze überschritten!');
    Exit;
  end
  else begin
    erfassen(p);
    feld[anz]:=p;  // Einlesen der Daten in das Feld
    saeubern;
    anz:=anz+1;
    Edit1.SetFocus;
  end;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);  // Datensätze ausgeben
var
    i:integer;
begin
   StringGrid1.Cells[0,0]:='Anrede';  // Überschriften des StringGrid schreiben
   StringGrid1.Cells[1,0]:= 'Name';
   StringGrid1.Cells[2,0]:= 'Vorname';
   StringGrid1.Cells[3,0]:=  'Wohnort';
   for i := 1 to anz do  // Datensätze aus dem Feld auslesen und im StringGrid
   begin                 ausgeben
     p := feld[i];
     StringGrid1.Cells[0,i] := p.anrede;
     StringGrid1.Cells[1,i] := p.name.nachname;
     StringGrid1.Cells[2,i] := p.name.vorname;
     StringGrid1.Cells[3,i] := p.anschrift.ort;
   end;
end;

// Daten in einer Datei abspeichern
procedure TForm1.Button4Click(Sender: TObject);
var
  ds: TPerson;
  i: integer;
  Daten: file of TPerson;
begin
  with SaveDialog1 do
  begin
    DefaultExt := 'dat';
    Title := 'Datensätze speichern';
    Filter := 'Datenfile *.dat| *.dat' + '|Alle Dateien *.*| *.*';
    InitialDir := ExtractFilePath(Application.ExeName);
  end;
  if SaveDialog1.Execute then
  begin
    AssignFile(Daten, SaveDialog1.FileName);  // Datei Namen zuweisen
    Rewrite(Daten);  // Datei zum Schreiben öffnen
    for i := 1 to 10 do
    begin
      ds := feld[i];
      Write(Daten, ds);  // Datensatz in Datei schreiben
    end;
    CloseFile(Daten);  // Datei schließen
  end;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject);  // Datei öffnen
var
  ds: TPerson;
  i: integer;
  Daten: file of TPerson;
begin
  with OpenDialog1 do
  begin
    DefaultExt := 'dat';
    Title := 'Datenfile öffnen';
    Filter := 'Datenfile *.dat| *.dat' + '|Alle Dateien *.*| *.*';
    InitialDir := ExtractFilePath(Application.ExeName);
  end;
  if OpenDialog1.Execute then
  begin
    AssignFile(Daten, OpenDialog1.FileName);  // Datei Namen zuweisen
    Reset(Daten);  // Datei lesend öffnen; Zeiger auf Anfang der Datei setzen
    StringGrid1.Cells[0,0]:='Anrede';
    StringGrid1.Cells[1,0]:= 'Name';
    StringGrid1.Cells[2,0]:= 'Vorname';
    StringGrid1.Cells[3,0]:=  'Wohnort';
    for i:= 1 to 10 do
    begin
      Read(Daten, ds);  // Datensatz aus Datei lesen
      StringGrid1.Cells[0,i] := ds.anrede;  // Datensatz in StringGrid schreiben
      StringGrid1.Cells[1,i] := ds.name.nachname;
      StringGrid1.Cells[2,i] := ds.name.vorname;
      StringGrid1.Cells[3,i] := ds.anschrift.ort;
    end;
    CloseFile(Daten);  // Datei schließen
  end;
end;
```