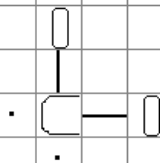
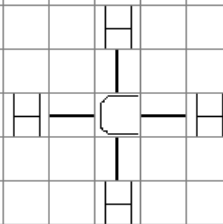


# Formeleditor

Marc Willem, Florian Singer  
Informatik

- Funktionen
- Aufbau
- Wichtige Prozeduren / Funktionen
- Umsetzung

## **Inhaltsverzeichnis**



## Werkzeuge

## Glyphen

## Elemente

C  
H  
N  
O

## Bindungen

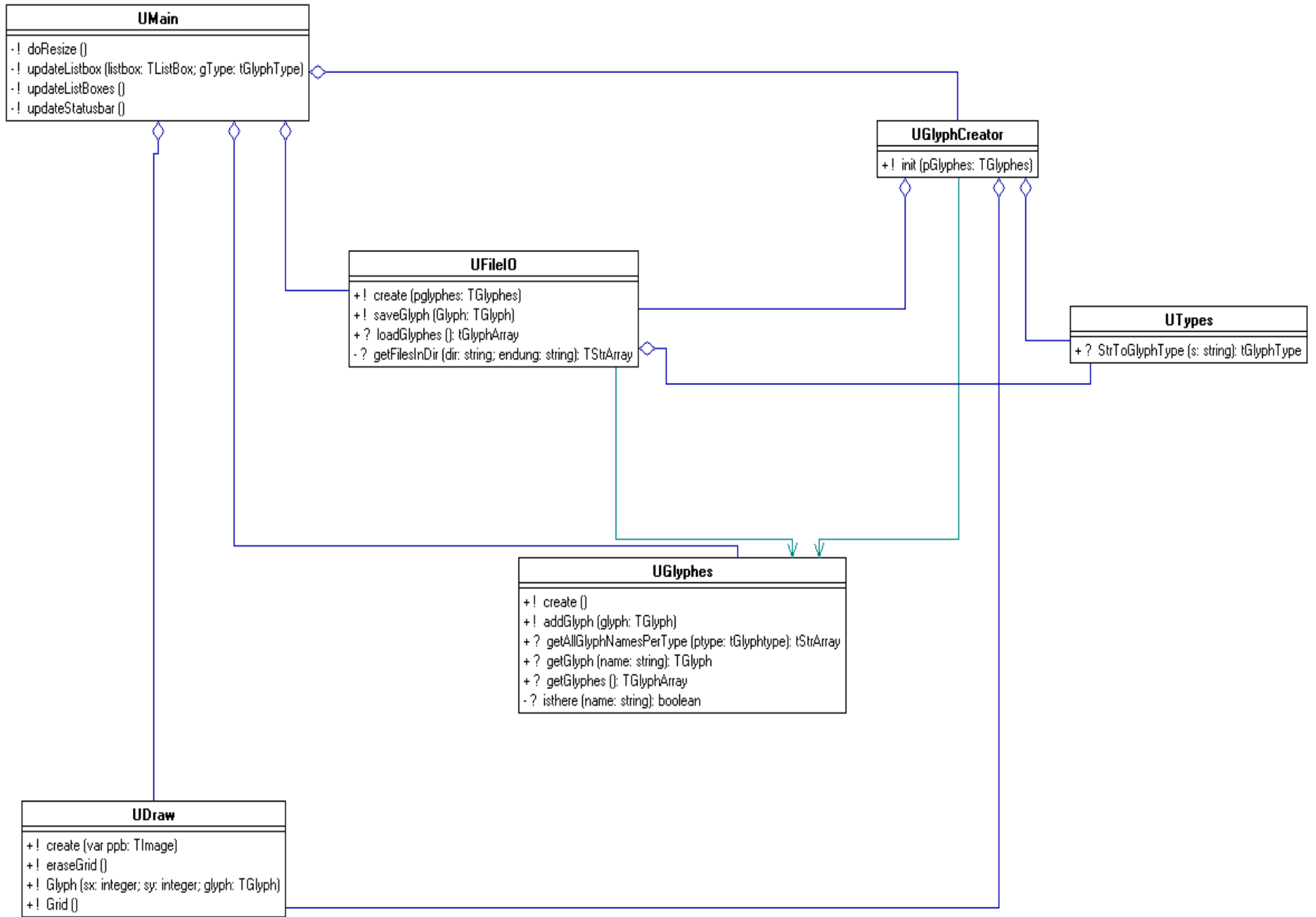
## Operatoren / Pfeile

Punkt  
strich(senkr.)  
strich(waagr.)

## Sonstiges

- Erstellen und Speichern von benutzererstellten Glyphen
- Jede chemische Verbindung lässt sich darstellen
- Abspeichern des Elements, welches mithilfe von Glyphen gezeichnet wurde
- Problemloses Entfernen eines falsch angebrachten Glyphs
- Durch viele kleine „Rechtecke“ lässt sich praktisch alles problemlos erstellen(Glyphenerstellung)

## Funktionen



## UMain:

- Hauptunit
- Einstellen der Menüleiste
- Einstellen der Benutzeroberfläche
- Laden notwendiger Funktionen bei Programmstart (FormCreate)

**Aufbau**

## UGlyphes: Verwaltung/ Speicherung der Glyphen

- Glyph hinzufügen
- Einzelner Glyph mit Namen erhalten
- Alle vorhandenen Glyphen erhalten
- Erhalten von Glyphen gleichen Typs

**Aufbau**

## UGlyphCreator: Erzeugung von Glyphen

- Oberfläche für die Glyphen
- Weiterleitung an UGlyph, zur Speicherung in der Datenbank

## UDraw: Zeichnen der Glyphen

- Einzelne Rechtecke markieren → Glyph
- Gitternetz zeichnen / löschen

**Aufbau**



## UFileIO: Verarbeitung von Daten

- Glyph abspeichern
- Alle Glyphen laden
- Auflistung aller Dateien mit bestimmter Endung in einem Ordner

## UTypes: Umwandlung der Typen

- String → GlyphType

**Aufbau**

```
procedure Tfmain.Speichern1Click(Sender: TObject); // Wenn Speichern geklickt
begin
  if SaveDialog1.Execute() then begin           // Wenn SaveDialog fertig
    draw.eraseGrid();                          // Gitter löschen
    idraw.Picture.SaveToFile(SaveDialog1.FileName); // In BMP speichern
    draw.Grid();                                // Gitter erstellen
  end;
end;
```

```
procedure Tfmain.Oeffnen1Click(Sender: TObject); // Wenn Öffnen geklickt
begin
  if OpenFileDialog1.Execute() then begin       // Wenn OpenFileDialog fertig
    idraw.Picture.LoadFromFile(OpenDialog1.FileName); // Aus BMP laden
    draw.Grid();                               // Gitternetz erstellen
  end;
end;
```

## Wichtige Prozeduren / Funktionen

```

procedure TDraw.Glyph(sx, sy: integer; Glyph: TGlyph); // Glyph
wird gezeichnet
var i,j:integer; // Zählvariablen
begin
  with pb.canvas do begin
    pb.canvas.pen.color:=clblack; //Schwarzer Stift
    for i:=1 to glyphen-1 do begin //Spalten
      for j:=1 to glyphen-1 do begin //Zeilen
        if Glyph.pixels[i,j]=true then pen.color:=clblack // Wenn Pixel
gesetzt
          else pen.color:=clwhite;// Wenn Pixel nicht gesetzt
        pb.Canvas.MoveTo(sx*glyphen+i,sy*glyphen+j);
//Anfangspunkt

pb.canvas.LineTo(sx*glyphen+i+1,sy*glyphen+j+1);//Endpunkt
      end;
    end;
  end;
end;

```

**Wichtige Prozeduren / Funktionen**

```
procedure TfGlyphCreator.idrawMouseDown(Sender: TObject; // Wenn Maus
geclicked
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var bx, by:integer; // Variablen für Kästchen
begin
  bx:=(X div 10); // Welches Kästchen wurde ausgewählt?
  by:=(Y div 10);
  if button = mleft then begin //Linksklick -> Kästchen setzen
    idraw.canvas.brush.color:=clblack; // Schwarze Füllfarbe
    myGlyph.Pixels[bx,by]:=true; // in myGlyph Pixel setzen
  end
  else begin //Rechtsklick -> Kästchen entfernen
    idraw.canvas.brush.color:=clwhite; // Weiße Füllfarbe
    myGlyph.Pixels[bx,by]:=false; // Pixel nicht setzen
  end;
  idraw.Canvas.Rectangle(bx*10,by*10,bx*10+10,by*10+10); // Kästchen malen
  draw.Glyph(0,0,myGlyph); //Vorschau malen
end;
```

**Wichtige Prozeduren / Funktionen**

Probleme:

- Stackoverflow
- Pixel setzen (Mousedown)

**Umsetzung**