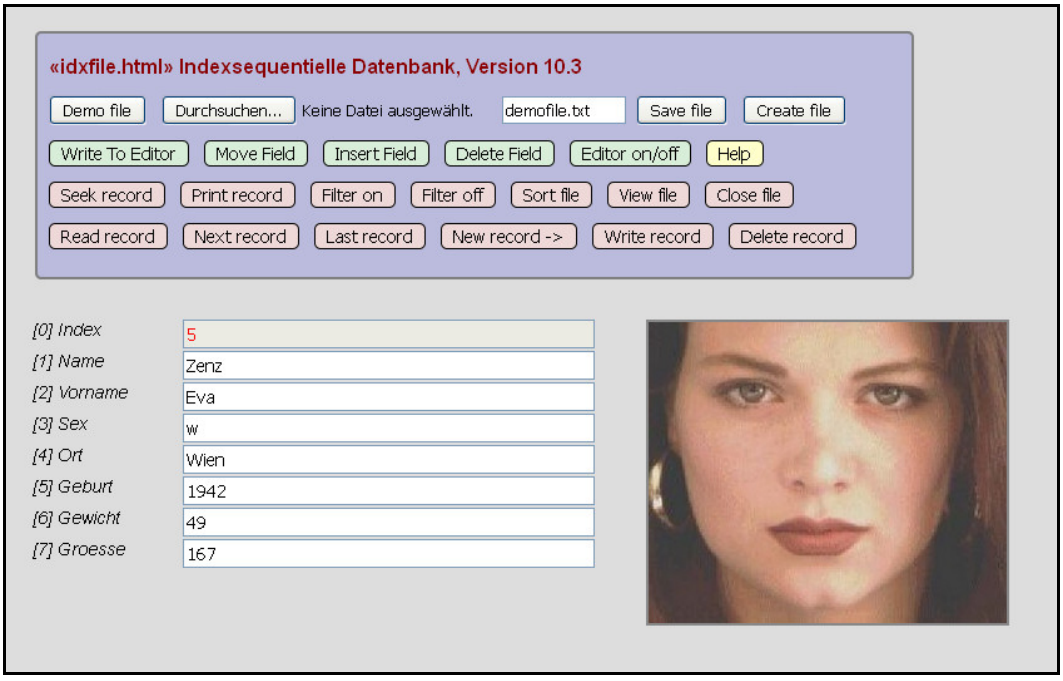
Indexsequentielle Datenbank (idxfile.html)
(programmiert in Javascript von Herbert Paukert)

Hinweis: Das hier beschriebene Programm findet man auf der Homepage des Autors in den JavaScript-Übungsprogrammen, Teil 5.

Eine solche Datenbank (File) ist eine einfache Textdatei, die aus verschiedenen Datensätzen (Records) besteht, welche durch den Separator "#" getrennt sind. Jeder Datensatz enthält gleichviele Datenfelder (Fields), welche durch den Separator ";" getrennt sind. Die Datensätze sind automatisch fortlaufend nummeriert (indiziert). Der erste Datensatz (Header) enthält immer die Namen der Datenfelder. Das erste Datenfeld (0) enthät immer den Index, der nicht verändert werden kann. Das nachfolgende Datenbank-Beispiel "demofile.txt" besteht aus dem Header und 25 Datensätzen.

```
Index;Name;Vorname;Sex;Ort;Geburt;Gewicht;Groesse#
 1;Ronka;Lisa;w;Wien;1945;65;165#
 2;Meier;Herbert;m;Graz;1940;87;190#
 3;Dorfer;Maria;w;Graz;1950;83;160#
 4;Stanka;Rudolf;m;Linz;1943;61;171#
 5;Zenz;Eva;w;Wien;1942;49;167#
 6;Wille;Heinz;m;Linz;1940;82;182#
 7;Rutger;Herbert;m;Wien;1943;74;178#
 8;Hauer;Friedl;m;Linz;1942;76;178#
 9;Mueller;Roland;m;Wien;1941;81;185#
10;Wollner;Christa;w;Linz;1947;67;156#
11;Klaus;Eva;w;Wien;1948;58;160#
12;Holler;Sabine;w;Wien;1945;58;170#
13;Ebner;Harald;m;Graz;1940;78;178#
14;Bauer;Ernst;m;Graz;1945;72;178#
15;Kurz;Werner;m;Wien;1942;73;185#
16;Stadler;Susi;w;Linz;1950;56;169#
17;Wolf;Maria;w;Graz;1949;58;167#
18;Klad;Eva;w;Linz;1943;52;165#
19;Artner;Heinz;m;Wien;1943;61;173#
20;Baier;Helene;w;Wien;1944;65;171#
21;Adam;Franz;m;Wien;1945;90;177#
22;Neuner;Eva;w;Graz;1950;56;170#
23;Troll;Rudolf;m;Linz;1943;69;178#
24;Sarg;Willi;m;Wien;1950;93;182#
25;Toth;Maria;w;Wien;1947;61;157
```



[***Demo file***] lädt eine Demo-Datenbank aus dem Hauptspeicher.
[***Durchsuchen (browse)***] lädt eine Datenbank (z.B. myfile.txt) aus einem Ordner.
[***Save file***] speichert eine Datenbank immer in den Download-Ordner.

Erzeugung einer neuen Datenbank im Hauptspeicher:
Zuerst den internen Texteditor öffnen.
([***Edit on/off***] öffnet und schließt den Editor.)
Dann die Datensätze in den geöffneten Editor schreiben.
Zuletzt den Schalter [***Create file***] anklicken.

Änderung der Datenbank-Struktur im Hauptspeicher:
[***WriteTo Edit***] transferiert die aktuelle Datenbank in den Editor.
Mit [***Move Field***] wird ein Datenfeld verschoben.
Mit [***Insert Field***] wird ein neues Datenfeld eingefügt.
Mit [***Delete Field***] wird ein Datenfeld entfernt.
[***Create file***] erzeugt die geänderte Datenbank.

[***Read record***] zeigt einen wählbaren Datensatz.
[***Next record***] zeigt den nachfolgenden Datensatz.
[***Last record***] zeigt den vorangehenden Datensatz.
[***New record***] legt einen neuen Datensatz an,
dessen Inhalt dann mit [***Write record***] gespeichert wird.
[***Write record***] speichert einen Datensatz ab,
dessen Inhalt vorher geändert oder neu angelegt wurde.
[***Delete record***] löscht den aktuellen Datensatz,
wobei alle Datensätze neu nummeriert (indiziert) werden.
[***Seek record***] sucht in allen Datensätzen
in einem wählbaren Datenfeld nach einem
Suchbegriff (unabhängig von Groß- oder Klein-
Schreibung und immer am Feldanfang beginnend).
[***Print record***] druckt einen Datensatz aus.

[***Filter on***] filtert Datensätzen heraus. Dabei wird
zuerst ein wählbares Datenfeld eingegeben,
dann ein Vergleichsoperator (<, =, >, ?) und
zuletzt ein Suchbegriff. (?) bedeutet, dass der
Suchbegriff im Datenfeld enthalten sein muss.
Auch wiederholte Filterungen sind möglich.
Es erfolgt eine automatische Unterscheidung von
numerischen und nicht numerischen Datenfeldern.
[***Filter off***] löscht alle vorher gesetzten Filter.
Für einige Funktionen ([***New record***], usw.)
müssen alle gesetzten Filter gelöscht werden.

[***View file***] zeigt wählbare Datenfelder der gesamten Datenbank an
und ermöglicht auch deren statistische Auswertung.
[***Sort file***] sortiert die Datenbank nach einem Datenfeld.
Dabei erfolgt eine automatische Unterscheidung von
numerischen und nicht numerischen Datenfeldern.
[***Close file***] schließt die aktuelle Datenbank.

Ein Datenbank-Beispiel mit Header und drei Datensätzen:
Index; Name; Vorname; Sex;Telefon #
1; Zenz; Eva; w; 01/7234219 #
2; Meier; Fritz; m; 01/6157730 #
3; Adam; Franz; m; 01/2315643
Eine solche Textdatei kann mit jedem Texteditor erzeugt werden,
aber auch im internen Editor (mit [***Edit on/off***] und [***Create file***]).

Befinden sich im aktuellen Ordner JPG-Bilder mit den Dateinamen
"Feld1_Feld2.jpg", dann werden die Bilder bei den entsprechenden
Datensätzen automatisch angezeigt (beispielsweise "Zenz_Eva.jpg“).

Jede im Hauptspeicher erzeugte Datenbank kann mit [***Save file***] nur
im Download-Ordner abgespeichert werden (wegen Internet-Sicherheit).
Sie kann dann – falls gewünscht – im jeweiligen Betriebssytem vom
Download-Ordner in jeden anderen Ordner kopiert werden.

Hinweis: In die Datenfelder dürfen keine Steuerzeichen (**, ; #**)
geschrieben werden, d.h. keine Beistriche, Strichpunkte und Rauten,
weil diese Zeichen für die Struktur der Datensätze intern verwendet
werden.

Das Programm "idxfile.html" stellt menügeführt alle wichtigen Verwaltungsroutinen
für Datenbanken zur Verfügung:

- Laden, Speichern und Schließen von Datenbanken (*file management*).
- Eingabe, Ausgabe und Drucken von Datensätzen in **Maskenform**.
- Ändern, Löschen und Anfügen von Datensätzen (*edit data*).
- Ändern der Datenbank-Struktur (*modify structure*)
  durch Verschieben, Einfügen oder Entfernen von Datenfeldern
  im internen Texteditor. In der **Listenform** im Editor können
  Feldinhalte geändert und auch ganze Datensätze gelöscht werden.
- Suchen von Suchbegriffen in allen Datensätzen (*find data*).
- Mehrstufiges Filtern von Datensätzen (*select data)*.
- Sortieren der Datensätze entsprechend einem wählbaren Datenfeld (*sort data*).
- Listendarstellung von wählbaren Datenfeldern (*view data*).
  Zusätzlich kann daraus ein Datenfeld bestimmt werden,
  welches dann statistisch ausgewertet wird.

Die unten stehende Grafik zeigt die zu „demofile1.txt" geänderte Datenbank „demofile.txt",
wobei folgende sechs Routinefunktionen durchgeführt wurden:

(1) Laden der Datenbank „demofile.txt" (mit [Demo file]).

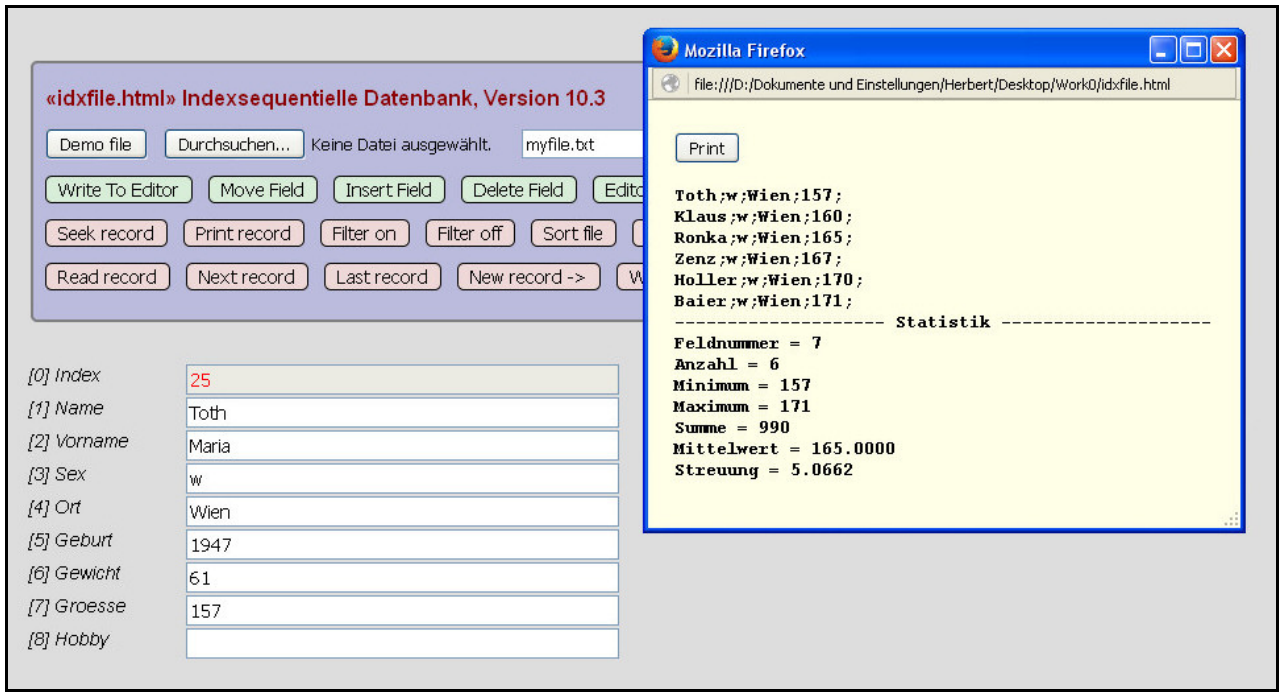
(2) Anhängen eines neuen Datenfeldes „Hobby" (mit [WriteTo Edit], [Insert Field] und [Create file]).

(3) Erster Datenfilter „Ort = Wien" (mit [Filter on]) → filtert 12 Datensätze (von 25).

(4) Zweiter Datenfilter „Sex = w" (mit [Filter on]). → filtert 6 Datensätze (von 12).

(5) Sortieren der Datensätze entsprechend dem Datenfeld „Groesse" (mit [Sort file]).

(6) Listendarstellung der Felder „Name", „Sex", „Ort" und „Groesse" (mit [View file]),
und statistische Auswertung des Datenfeldes „Groesse".



```
<!DOCTYPE html">
<html>
<head>
<title>Indexsequentielle Datenbank</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="Indexsequentielle Datenbank">
<meta name="author" content="Herbert Paukert">

<style>
body {margin: 30px; background-color:#DDDDDD; color:black; font-family: Arial; font-size: 15px; }
.bs  {border: 1px solid black; border-radius:5px; background-color: #EED8D8; font-size: 15px;}
.bs1 {border: 1px solid black; border-radius:5px; background-color: #D8EED8; font-size: 15px;}
.bs2 {border: 1px solid black; border-radius:2px; background-color: #FFFFFF; font-size: 15px;}
.bs3 {border: 1px solid black; border-radius:5px; background-color: #FFFFCC; font-size: 15px;}
img  {position:absolute; left:540px; height:240px; border: 2px solid #888888; visibility:hidden;}
#tbox {border: 2px solid #888888; visibility:hidden; font-size: 14px; }
#tit {font-size:16px; font-weight: bold; color: darkred;}
#fs  {background-color:#BBBBDD; width:720px; border: 2px solid gray; border-radius:5px;}
#btns {line-height:8px; }
</style>

<script>
   var childWindow;
   var child = false;
   window.addEventListener( "unload", function() { closeChild(); } );

   function closeChild() {
   // Hilfetext schließen
     if (!childWindow || childWindow.closed) { return; }
     if (child) { childWindow.document.close(); childWindow.close(); }
   }

  function help_() {
  // Hilfetext anzeigen
    if (child) { closeChild(); child = false; return; }
    child = true;
    childWindow = window.open('','childWindow','top=20, left=720, width=600, height=640, srollbars=yes, menubar=no, toolbar=no');
    childWindow.document.open();
    childWindow.document.write('<html><head></head><body style="background-color:#FFFFE8; font-family:Arial, sans-serif;
                           font-size:13px; margin-left:20px; margin-bottom:40px;">');
    childWindow.document.write('<br><input type="button" value="Print" onclick="print();"><br><br>');
    childWindow.document.write('<b>Indexsequentielle Datenbank "idxfile.html" (c) Herbert Paukert.</b><br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Eine solche Datenbank (file) ist eine einfache Textdatei, die aus<br>');
    childWindow.document.write('verschiedenen Datensätzen (records) besteht, welche durch den<br>');
    childWindow.document.write('Separator "#" getrennt sind. Jeder Datensatz enthält gleichviele<br>');
    childWindow.document.write('Datenfelder (fields), welche durch den Separator ";" getrennt sind.<br>');
    childWindow.document.write('Die Datensätze sind automatisch fortlaufend nummeriert (indiziert).<br>');
    childWindow.document.write('Der erste Datensatz (Header) enthält immer die Namen der Datenfelder.<br>');
    childWindow.document.write('Im ersten Datenfeld (0) ist der Index, der nicht verändert werden kann.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Hinweis: In die Datenfelder dürfen keine Steuerzeichen<b>  ","  ";"  "#"</b><br>');
    childWindow.document.write('geschrieben werden, d.h. keine Beistriche, Strichpunkte und Rauten!<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('<b>[Demo file]</b> lädt eine Demo-Datenbank aus dem Hauptspeicher.<br>');
    childWindow.document.write('<b>[Durchsuchen (browse)]</b> lädt eine Datenbank aus einem Ordner.<br>');
    childWindow.document.write('<b>[Save file]</b> speichert eine Datenbank immer in den Download-Ordner.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Erzeugung einer neuen Datenbank im Hauptspeicher:<br>');
    childWindow.document.write('    Zuerst den internen Texteditor öffnen.<br>');
    childWindow.document.write('    <b>[Editor on/off]</b> öffnet oder schließt den Editor.<br>');
    childWindow.document.write('    Dann die Datensätze in den offenen Editor schreiben.<br>');
    childWindow.document.write('    Zuletzt den Schalter <b>[Create file]</b> anklicken.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Änderung der Datenbank-Struktur im Hauptspeicher:<br>');
    childWindow.document.write('    <b>[WriteTo Editor]</b> transferiert die aktuelle Datenbank in den Editor.<br>');
    childWindow.document.write('    Mit <b>[Move Field]</b> wird ein Datenfeld verschoben.<br>');
    childWindow.document.write('    Mit <b>[Insert Field]</b> wird ein neues Datenfeld eingefügt.<br>');
    childWindow.document.write('    Mit <b>[Delete Field]</b> wird ein Datenfeld entfernt.<br>');
    childWindow.document.write('    <b>[Create file]</b> erzeugt die geänderte Datenbank.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('<b>[Read record]</b> zeigt einen wählbaren Datensatz.<br>');
    childWindow.document.write('<b>[Next record]</b> zeigt den nachfolgenden Datensatz.<br>');
    childWindow.document.write('<b>[Last record]</b> zeigt den vorangehenden Datensatz.<br>');
    childWindow.document.write('<b>[New record]</b> legt einen neuen Datensatz an,<br>');
    childWindow.document.write('    dessen Inhalt dann mit <b>[Write record]</b> gespeichert wird,<br>');
    childWindow.document.write('    wobei alle Datensätze neu nummeriert (indiziert) werden.<br>');
    childWindow.document.write('<b>[Write record]</b> speichert einen Datensatz ab,<br>');
    childWindow.document.write('    dessen Inhalt vorher geändert oder neu angelegt wurde.<br>');
    childWindow.document.write('<b>[Delete record]</b> löscht den aktuellen Datensatz,<br>');
    childWindow.document.write('    wobei alle Datensätze neu nummeriert (indiziert) werden.<br>');
    childWindow.document.write('<b>[Seek record]</b> sucht in allen Datensätzen<br>');
    childWindow.document.write('    in einem wählbaren Datenfeld nach einem<br>');
    childWindow.document.write('    Suchbegriff (unabhängig von Groß- oder Klein-<br>');
    childWindow.document.write('    Schreibung und immer am Feldanfang beginnend).<br>');
    childWindow.document.write('<b>[Print record]</b> druckt einen Datensatz aus.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('<b>[<i>Filter on</i>]</b> filtert Datensätze heraus. Dabei wird<br>');
    childWindow.document.write('    zuerst ein wählbares Datenfeld eingegeben,<br>');
    childWindow.document.write('    dann ein Vergleichsoperator (<, =, >, ?) und<br>');
    childWindow.document.write('    zuletzt ein Suchbegriff. (?) bedeutet, dass der<br>');
    childWindow.document.write('    Suchbegriff im Datenfeld enthalten sein muss.<br>');
    childWindow.document.write('    Auch wiederholte Filterungen sind möglich.<br>');
    childWindow.document.write('    Es erfolgt eine automatische Unterscheidung von<br>');
    childWindow.document.write('    numerischen und nicht numerischen Datenfeldern.<br>');
    childWindow.document.write('<b>[Filter off]</b> löscht alle vorher gesetzten Filter.<br>');
    childWindow.document.write('    Für einige Funktionen (<b>[New record]</b>,usw.) müssen<br>');
    childWindow.document.write('    alle gesetzten Filter gelöscht werden.<br>');
    childWindow.document.write('<br>');
```

```
    childWindow.document.write('<b>[Sort file]</b> sortiert die Datenbank nach einem Datenfeld.<br>');
    childWindow.document.write('    Dabei erfolgt eine automatische Unterscheidung von<br>');
    childWindow.document.write('    numerischen und nicht numerischen Datenfeldern.<br>');
    childWindow.document.write('<b>[View file]</b> zeigt wählbare Felder der gesamten Datenbank an<br>');
    childWindow.document.write('    und ermöglicht deren statistische Auswertung.<br>');
    childWindow.document.write('<b>[Close file]</b> schließt die aktuelle Datenbank.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Datenbank-Beispiel mit Header und drei nachfolgenden Datensätzen:<br>');
    childWindow.document.write('    Index;Name;Vorname;Sex;Telefon#<br>');
    childWindow.document.write('    1;Zenz;Eva;w;01/7234219#<br>');
    childWindow.document.write('    2;Meier;Fritz;m;01/6157730#<br>');
    childWindow.document.write('    3;Adam;Franz;m;01/2315643<br>');
    childWindow.document.write('Eine solche Textdatei kann mit jedem einfachen Editor erzeugt werden,<br>');
    childWindow.document.write('aber auch im internen Editor (mit <b>[Editor on/off]</b> und <b>[Create file]</b>).<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Befinden sich im aktuellen Ordner JPG-Bilder mit den Dateinamen<br>');
    childWindow.document.write('"Feld1_Feld2.jpg", dann werden die Bilder bei den entsprechenden<br>');
    childWindow.document.write('Datensätzen automatisch angezeigt (beispielsweise "Zenz_Eva.jpg").<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('Jede im Hauptspeicher erzeugte Datenbank kann mit <b>[Save file]</b> nur<br>');
    childWindow.document.write('im Download-Ordner abgespeichert werden (wegen Internet-Sicherheit).<br>');
    childWindow.document.write('Sie kann dann - falls gewünscht - im jeweiligen Betriebssystem vom<br>');
    childWindow.document.write('Download-Ordner in jeden anderen Ordner kopiert werden.<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('<b>Ein Anwendungsbeispiel:</b><br>');
    childWindow.document.write('(1) Laden der Demo-Datenbank "demofile.txt" mit [Demo].<br>');
    childWindow.document.write('(2) Anhängen eines neuen Datenfeldes "Hobby" mit [WriteTo Editor]<br>');
    childWindow.document.write('    und [Insert field] und [Create file]. Dann Eingaben in "Hobby".<br>');
    childWindow.document.write('(3) Erster Datenfilter "Ort = Wien" mit [Filter on].<br>');
    childWindow.document.write('(4) Zweiter Datenfilter "Sex = w" mit [Filter on].<br>');
    childWindow.document.write('(5) Sortieren nach dem Datenfeld "Groesse" mit [Sort file].<br>');
    childWindow.document.write('(6) Listenausgabe von "Name", "Sex", "Ort", "Groesse" mit [View file].<br>');
    childWindow.document.write('    und statistische Auswertung des Datenfeldes "Groesse".<br>');
    childWindow.document.write('(7) Datenbank speichern als "demofile1.txt" mit [Save file].<br>');
    childWindow.document.write('<br>');
    childWindow.document.write('</body></html>');
   }

// Variable für die Datenfelder
var header = "";        // String der Datenfeld-Namen
var fsep   = ";";       // Datenfeld-Trennzeichen
var fname  = [];        // Datenfeld-Namen als Array
var fmax   = 0;         // Anzahl der Datenfelder
var fsize  = "15px";    // Datenfeld-Schriftgröße

// Bildkoordinaten der Datenfelder festlegen
var x1 = 30;            // x-Position des ersten Feldnamens
var x2 = x1 + 120;      // x-Position des ersten Datenfeldes
var y1 = 250;           // y-Position des ersten Feldnamens
var y2 = 250;           // y-Position des ersten Datenfeldes
var hy = 26;            // vertikaler Abstand der Datenfelder
var wx = 45;            // horizontale Breite der Datenfelder

// Variable für die Datensätze
var sep = '#';          // Datensatz-Tennzeichen
var file0;              // Datenbank-Datei als String
var allrec = [];        // Datensätze als Array
var orgrec = [];        // erste Kopie von allrec
var coprec = [];        // zweite Kopie von allrec
var rmax   = 0;         // Anzahl der Datensätze
var rnum   = 0;         // Laufende Datensatz-Nummer (Index)
var filt   = false;     // Steuervariable für Datenfilterung
var imageName = '';     // Name einer zum Datensatz passenden Bilddatei
var komma  = ',';       // interne Trennvariable

// Steuervariable
var fileLoad  = false;  // Kennzeichen für geladene Datenbank
var readFirst = false;  // Kennzeichen für eingelesenen Datensatz
var writeNew  = false;  // Kennzeichen für neuen Datensatz
var txtvisi   = false;  // Kennzeichen für die Sichtbarkeit des Texteditors
var transfer  = false;  // Kennzeichen für Datentransfer in den Texteditor

// Erstes und zweites Index-Array zur Datenbank-Sortierung
var index0 = [];        // originales Indexarray
var index1 = [];        // beliebiges Datenarray

// Kindfenster zur Listendarstellung und zum Listenausdruck
var childWindow;

// Minimale Datenbank
var file1 = 'Index;Name;Vorname;Sex;Telefon#' + '\n' +
            '1;Zenz;Eva;w;01/82765409#'+ '\n' +
            '2;Meier;Fritz;m;01/23408641#' + '\n' +
            '3;Adam;Franz;m;01/5376627';
```

```
function createStruc() {
// Feldstruktur erzeugen
  for (i = 0; i < fmax; i++) {
    s1 = '['+i+ ']  ' + fname[i]+' ';
    s2 = "feld"+i;  // Feldnamen-ID
    var f = document.createElement("TEXT");
    var t = document.createTextNode(s1);
    f.appendChild(t);
    f.setAttribute("id",s2);
    document.body.appendChild(f);
    document.getElementById(s2).style.fontStyle = "italic";
    document.getElementById(s2).style.fontSize = fsize;
    setPosition(s2,x1,y1+i*hy);
  }
}

function removeStruc() {
// Feldstruktur entfernen
  for (i = 0; i < fmax; i++) {
    s1 = "feld"+i;  // Feldnamen-ID
    s2 = "data"+i;  // Felddaten-ID
    var node1 = document.getElementById(s1);
    var node2 = document.getElementById(s2);
    node1.parentNode.removeChild(node1);
    node2.parentNode.removeChild(node2);
  }
}

// Entfernt führende und nachfolgende Leerzeichen eines Strings
function myTrim(x) { return x.replace(/^\s+|\s+$/gm,'') }

function setPosition(element,x,y) {
// absolute Positionierung
   var el = document.getElementById(element);
   el.style.position = "absolute";
   el.style.left = x + "px";
   el.style.top =  y + "px";
}

function copy0() {
// allrec -> orgrec: originale Datenbank sichern
   orgrec.length = 0;
   var n = allrec.length;
   for (j = 0; j < n; j++) { orgrec.push(allrec[j]); }
}

function copy1() {
// allrec -> coprec: Datenbank zum Filtern kopieren
   coprec.length = 0;
   var n = allrec.length;
   for (j = 0; j < n; j++) { coprec.push(allrec[j]); }
   allrec.length = 0;
}

function copy2() {
// orgrec -> allrec: gesicherte, originale Datenbank wieder erstellen
   allrec.length = 0;
   var n = orgrec.length;
   for (j = 0; j < n; j++) { allrec.push(orgrec[j]); }
   rmax = n;
}

function filterOff() {
// gesetzte Filter ausschalten
   if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
   copy2();
   rnum = -1;
   nextRec();
   filt = false;
   alert('Datenfilter aufgehoben !');
}

function filterRec() {
// Datensätze mehrstufig filtern, mit Feldwerten <, =, >, ? Suchbegriff
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if ( !filt ) { copy0(); }
  copy1();
  var gesucht = ' x , ? , y ';
  var info = ' Feldnummer x, Operation (<,=,>,?), Suchbegriff y ';
  gesucht = prompt(info,gesucht);
  gesucht = myTrim(gesucht);
  var s = gesucht.split(',');
  var a = myTrim(s[0]);
```

```
  if ( isNaN(a) || (a < 1) || (a > fmax-1) ) {
    alert('Falsche Datenfeldnummer!');
    filterOff();
    return;
  }
  var b = myTrim(s[2]);
  var c = myTrim(s[1]);
  var okay = false;
  if ( (c == '<') || (c == '=') || (c == '>') || (c == '?') ) { okay = true; }
  if ( !okay) {
    alert('Falsche Vergleichsoperation!');
    filterOff();
    return;
  }
  if (c == '?') {
     b = b.toUpperCase();
     var count = 0;
     for (j = 0; j < rmax; j++) {
       var rec = coprec[j].split(fsep);
       s = rec[a].toUpperCase();
       if ( s.includes(b) ) { count++; allrec.push(coprec[j]); }
     }
  }
  if (c == '<') {
     var count = 0;
     for (j = 0; j < rmax; j++) {
       var rec = coprec[j].split(fsep);
       s = rec[a];
       if ( isNaN(s) ) {
          if (s < b) { count++; allrec.push(coprec[j]); }
       }
       else {
          if (1*s < 1*b) { count++; allrec.push(coprec[j]); }
       }
     }
  }
  if (c == '=') {
     var count = 0;
     for (j = 0; j < rmax; j++) {
       var rec = coprec[j].split(fsep);
       s = rec[a];
       if (s == b) { count++; allrec.push(coprec[j]); }
     }
  }
  if (c == '>') {
     var count = 0;
     for (j = 0; j < rmax; j++) {
       var rec = coprec[j].split(fsep);
       s = rec[a];
       if ( isNaN(s) ) {
          if (s > b) { count++; allrec.push(coprec[j]); }
       }
       else {
          if (1*s > 1*b) { count++; allrec.push(coprec[j]); }
       }
     }
  }
  rmax = count;
  alert(count + ' Datensätze gefunden !');
  if (count == 0) { filterOff(); return; }
  rnum = -1;
  nextRec();
  filt = true;
}

function sortFile() {
// Gesamte Datenbank nach einem Datenfeld sortieren
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  var info = 'Datenfeldnummer (x) eingeben';
  var n = prompt(info,'x');
  var a = myTrim(n);
  if ( isNaN(a) || (a < 0) || (a > fmax-1) ) {
    alert('Falsche Datenfeldnummer!');
    return;
  }
  for (j = 0; j < rmax; j++) {
    var rec = allrec[j].split(fsep);
    index0[j] = rec[0];
    index1[j] = rec[a];
  }
```

```
  if ( isNaN(rec[a]) ) {
    for (i = 0; i < rmax; i++) {
     for (j = i+1; j < rmax; j++) {
      if (index1[j] < index1[i]) {
        x = index1[i]; index1[i] = index1[j]; index1[j] = x;
        y = index0[i]; index0[i] = index0[j]; index0[j] = y;
        z = allrec[i]; allrec[i] = allrec[j]; allrec[j] = z;
      }
     }
    }
  }
  if ( !isNaN(rec[a]) ) {
    for (i = 0; i < rmax; i++) {
     for (j = i+1; j < rmax; j++) {
      if (1*index1[j] < 1*index1[i]) {
        x = index1[i]; index1[i] = index1[j]; index1[j] = x;
        y = index0[i]; index0[i] = index0[j]; index0[j] = y;
        z = allrec[i]; allrec[i] = allrec[j]; allrec[j] = z;
      }
     }
    }
  }
  alert('Datenbank sortiert!');
  rnum = - 1;
  nextRec();
}

function readRec() {
// Einen aktuellen Datensatz entsprechend der Indexnummer auswählen
  if ( !fileLoad ) { alert('Zuerst Datenbank erzeugen!'); return; }
  document.getElementById('pic').style.visibility = "hidden";
  var info = '( 1 <= n <= ' + rmax + ' )';
  var num = prompt('Datensatz-Nummer ' + info, 1);
  num = myTrim(num);
  if ( isNaN(num) || (num < 1) || (num > rmax)) { num = 1; }
  rnum = num - 1;
  var rec = allrec[rnum].split(fsep);
  var s1,s2;
  if (!readFirst) {
    for (i = 0; i < fmax; i++) {
      s1 = rec[i];
      s2 = "data"+i;
      var f = document.createElement("INPUT");
      f.setAttribute("id",s2);
      document.body.appendChild(f);
      document.getElementById(s2).style.fontSize = fsize;
      document.getElementById(s2).size = wx;
      document.getElementById(s2).value = s1;
      setPosition(s2,x2,y2+i*hy);
      readFirst = true;
    }
    document.getElementById("data0").readOnly = true;
    document.getElementById("data0").style.backgroundColor = "AntiqueWhite";
    document.getElementById("data0").style.color= "red";
    imageName = rec[1] + '_' + rec[2];
    loadImage(imageName);
  }
  else {
    for (i = 0; i < fmax; i++) {
      s1 = rec[i];
      s2 = "data"+i;
      document.getElementById(s2).value = s1;
    }
    imageName = rec[1] + '_' + rec[2];
    loadImage(imageName);
  }
  document.getElementById("feld1").focus();
  transfer = false;
}

function nextRec() {
// Zum nachfolgenden Datensatz gehen
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if (rnum < rmax - 1) { rnum = rnum + 1 }
  var rec = allrec[rnum].split(fsep);

  var s1,s2;
  for (i = 0; i < fmax; i++) {
    s1 = rec[i];
    s2 = "data"+i;
    document.getElementById(s2).value = s1;
  }
```

```
  imageName = rec[1] + '_' + rec[2];
  loadImage(imageName);
  document.getElementById("feld1").focus();
}

function lastRec() {
// Zum vorangehenden Datensatz gehen
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if (rnum > 0) { rnum = rnum - 1 }
  var rec = allrec[rnum].split(fsep);
  var s1,s2;
  for (i = 0; i < fmax; i++) {
    s1 = rec[i];
    s2 = "data"+i;
    document.getElementById(s2).value = s1;
  }
  imageName = rec[1] + '_' + rec[2];
  loadImage(imageName);
  document.getElementById("feld1").focus();
}

function newRec() {
// Einen neuen Datensatz anlegen
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if ( filt ) { alert('Zuerst den Datenfilter aufheben!'); return; }
  var s1,s2;
  for (i = 0; i < fmax; i++) {
    s1 = '';
    if (i == 0) { s1 = (1)*rmax + 1; }
    s2 = "data"+i;
    document.getElementById(s2).value = s1;
  }
  writeNew = true;
  imageName = '';
  loadImage(imageName);
  document.getElementById("data1").focus();
}

function writeRec() {
// Einen neuen oder geänderten Datensatz speichern
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if ( filt ) { alert('Zuerst den Datenfilter aufheben!'); return; }
  var rec = '';
  var s1,s2,s3;
  for (i = 0; i < fmax; i++) {
    s2 = "data"+i;
    s1 = document.getElementById(s2).value;
    if ((writeNew) && (i == 0))  { s1 = (1)*rmax + 1; s3 = s1; }
    if ((!writeNew) && (i == 0)) { s1 = (1)*rnum + 1; s3 = s1; }
    rec = rec + s1 + ';';
  }
  document.getElementById("data0").value = s3;
  if (writeNew) {
  // Datenbank neu indizieren
    allrec.push(rec);
    rmax = rmax + 1;
    rnum = rmax;
    allrec.splice(rnum,1);
    var s1 = header + sep + '\n';
    for (i = 0; i < rmax; i++) {
      t1 = allrec[i];
      p  = t1.indexOf(fsep);
      t2 = t1.substr(p);
      t1 = (1*i+1) + t2;
      allrec[i] = t1;
      s1 = s1 + allrec[i] + sep;
    }
    file0 = s1.substr(0,s1.length-1);
    alert('Neuer Record gespeichert !');
  }
  else {
    allrec[rnum] = rec;
    rmax = allrec.length;
    s1 = header + sep + '\n';
    for (i = 0; i < rmax; i++) { s1 = s1 + allrec[i] + sep; }
    file0 = s1.substr(0,s1.length-1);
    alert('Aktueller Record geändert !');
  }
  writeNew = false;
}
```

```
function delRec() {
// Den aktuellen Datensatz löschen
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if ( filt ) { alert('Zuerst den Datenfilter aufheben!'); return; }
  var ein = confirm('Aktuellen Datensatz wirklich löschen?');
  if (!ein) { return; }
  if (rmax == 1) { alert('Eine weitere Löschung ist nicht mehr möglich !'); return; }
  // Datenbank neu indizieren
  rmax = rmax - 1;
  allrec.splice(rnum,1);
  var s1 = header + sep + '\n';
  for (i = 0; i < rmax; i++) {
    t1 = allrec[i];
    p  = t1.indexOf(fsep);
    t2 = t1.substr(p);
    t1 = (1*i+1) + t2;
    allrec[i] = t1;
    s1 = s1 + allrec[i] + sep;
  }
  file0 = s1.substr(0,s1.length-1);
  rnum = rnum - 1;
  alert('Datensatz gelöscht und Datenbank neu indiziert!');
  nextRec();
}

function printRec() {
// Den aktuellen Datensatz drucken
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  document.getElementById('btns').style.visibility = "hidden";
  print();
  document.getElementById('btns').style.visibility = "visible";
}

function seekRec() {
// Einen Begriff in einem Datenfeld suchen (caseinsensitiv und nur am Feldanfang)
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  var found = false;
  var count = 0;
  var info = 'Datenfeldnummer (x) und Suchbegriff (y) eingeben';
  var gesucht = ' x , y ';
  gesucht = prompt(info,gesucht);
  gesucht = myTrim(gesucht);

  var s = gesucht.split(',');
  var a = myTrim(s[0]);
  var b = myTrim(s[1]);
  b = b.toUpperCase();
  var len = b.length;
  if ( isNaN(a) || (a < 1) || (a > fmax-1) ) {
    alert('Falsche Datenfeldnummer!');
    return;
  }
  for (j = 0; j < rmax; j++) {
    rec = allrec[j].split(fsep);
    index0[j] = rec[0];
    index1[j] = rec[a];
    c = index1[j].toUpperCase();
    c = c.substring(0,len);
    if (c == b) {
      found = true;
      count++;
      posi = j;
      rec = allrec[posi].split(fsep);
      for (i = 0; i < fmax; i++) {
        s1 = rec[i];
        s2 = "data"+i;
        document.getElementById(s2).value = s1;
      }
      imageName = rec[1] + '_' + rec[2];
      loadImage(imageName);
      document.getElementById("data1").focus();
      result = confirm('Suche fortsetzen ?');
      if ( !result ) { break; }
    }
  }
  if ( !found ) {
    alert('Begriff <' + s[1] + '> am Anfang von Feld <' + s[0] + '> NICHT gefunden!');
  }
  else {
    alert('Begriff <' + s[1] + '> am Anfang von Feld <' + s[0] + '> bisher ' + count +
          '-Mal gefunden!');
    rnum = posi;
  }
}
```

```
function loadFile() {
// Eine neue Datenbank öffnen
  if ( txtvisi ) { alert('Zuerst mit [Editor on/off] den Texteditor schließen!'); return; }
  if ( fileLoad ) { alert('Zuerst Datenbank mit [Close file] schließen!'); return; }
  var fileToLoad = document.getElementById("fileToLoad").files[0];
  textType = /text.*/;
  if ( !fileToLoad.type.match(textType) ) { alert('Falscher Datentyp'); return; }
  var reader = new FileReader();
  reader.readAsText(fileToLoad,"ISO-8859-1");
  reader.onload = function(event) {
     var textFile = event.target.result;
     file0  = textFile;
     allrec.length = 0;
     fname.length = 0;
     allrec = file0.split(sep);
     header = allrec.shift();
     rmax   = allrec.length;
     fname  = header.split(fsep);
     fmax   = fname.length;
     createStruc();
     copy0();
     alert('Zuerst mit [Read record] einen Datensatz einlesen!');
     document.getElementById('fileName').value = 'myfile.txt';
     document.getElementById("btn1").focus();
     transfer = false;
     fileLoad = true;
  }
}

function saveFile() {
// Die aktuelle Datenbank speichern
  if (!fileLoad) { alert('Zuerst Datenbank erzeugen!'); return; }
  if ( filt ) { alert('Zuerst den Datenfilter aufheben!'); return; }
  document.getElementById("fileToLoad").value = '';
  var fileToWrite = file0;
  var textBlob = new Blob([fileToWrite], {type:'text/plain'});
  var name = document.getElementById("fileName").value;
  var link = document.createElement("a");
  link.download = name;
  link.innerHTML = "Download File";
  link.href = window.URL.createObjectURL(textBlob);
  link.style.display = "none";
  document.body.appendChild(link);
  link.click();
  document.body.removeChild(link);
}

function imgExists(url) {
// Prüft die Existenz eines zum "Feld1_Feld2" passenden JPEG-Bildes
  var image = new Image();
  var exists = true;
  image.src = url;
  if (!image.complete) { exists = false; }
  if (image.width === 0) { exists = false; }
  return exists;
}

function loadImage(nam) {
// Zeigt ein zum "Feld1_Feld2" passendes JPEG-Bild
  nam = nam + '.jpg';
  var imgbox = document.getElementById("pic");
  imgbox.src = nam;
  var url = imgbox.src;
  var exists = imgExists(url);
  if (exists) { imgbox.style.visibility = "visible"; }
  else { imgbox.style.visibility = "hidden"; }
}

function demoFile() {
// Demo-Datenbank
  if ( txtvisi ) { alert('Zuerst mit [Editor on/off] den Texteditor schließen!'); return; }
  if ( fileLoad ) { alert('Zuerst Datenbank mit [Close file] schließen!'); return; }
  file0 = "Index;Name;Vorname;Sex;Ort;Geburt;Gewicht;Groesse#" +
          "1;Ronka;Lisa;w;Wien;1945;65;165#" +
          "2;Meier;Herbert;m;Graz;1940;87;190#" +
          "3;Dorfer;Maria;w;Graz;1950;83;160#" +
          "4;Stanka;Rudolf;m;Linz;1943;61;171#" +
          "5;Zenz;Eva;w;Wien;1942;49;167#" +
          "6;Wille;Heinz;m;Linz;1940;82;182#" +
          "7;Rutger;Herbert;m;Wien;1943;74;178#" +
          "8;Hauer;Friedl;m;Linz;1942;76;178#" +
          "9;Mueller;Roland;m;Wien;1941;81;185#" +
          "10;Wollner;Christa;w;Linz;1947;67;156#" +
```

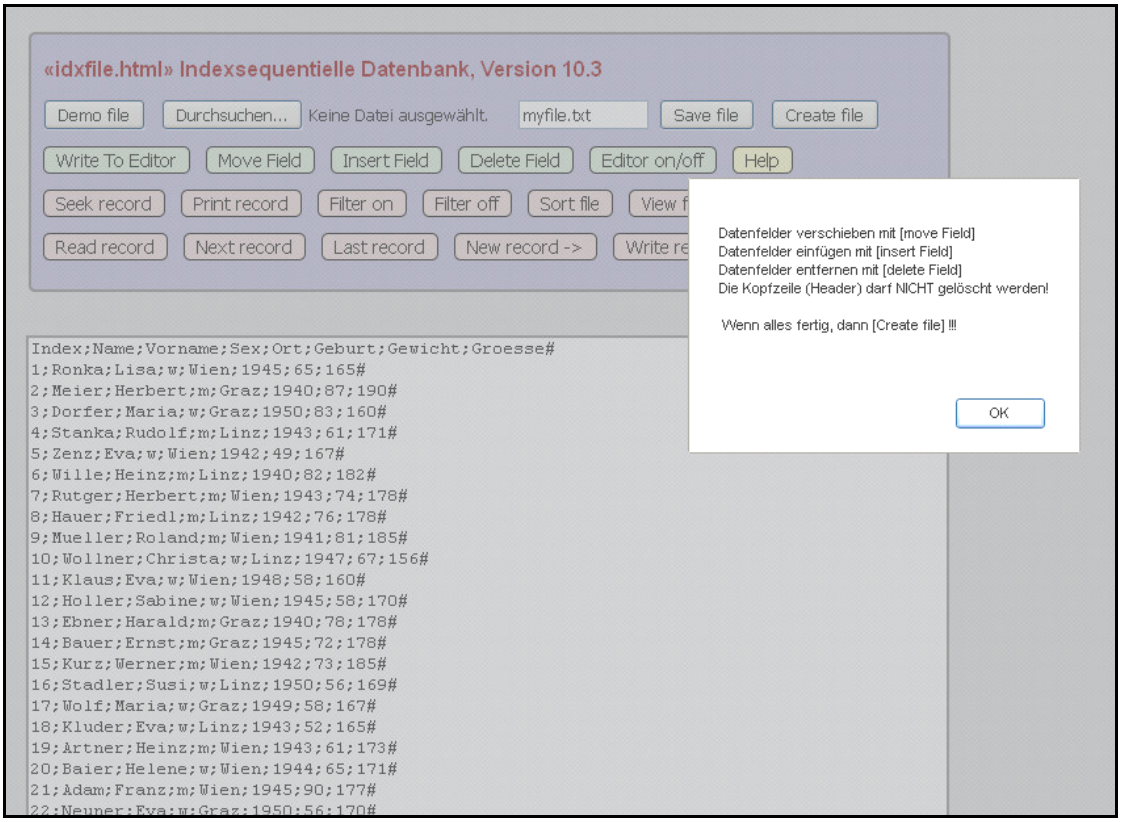
```
          "11;Klaus;Eva;w;Wien;1948;58;160#" +
          "12;Holler;Sabine;w;Wien;1945;58;170#" +
          "13;Ebner;Harald;m;Graz;1940;78;178#" +
          "14;Bauer;Ernst;m;Graz;1945;72;178#" +
          "15;Kurz;Werner;m;Wien;1942;73;185#" +
          "16;Stadler;Susi;w;Linz;1950;56;169#" +
          "17;Wolf;Maria;w;Graz;1949;58;167#" +
          "18;Kluder;Eva;w;Linz;1943;52;165#" +
          "19;Artner;Heinz;m;Wien;1943;61;173#" +
          "20;Baier;Helene;w;Wien;1944;65;171#" +
          "21;Adam;Franz;m;Wien;1945;90;177#" +
          "22;Neuner;Eva;w;Graz;1950;56;170#" +
          "23;Troll;Rudolf;m;Linz;1943;69;178#" +
          "24;Sarg;Willi;m;Wien;1950;93;182#" +
          "25;Toth;Maria;w;Wien;1947;61;157";
  allrec.length = 0;
  fname.length = 0;
  allrec = file0.split(sep);
  header = allrec.shift();
  rmax   = allrec.length;
  fname  = header.split(fsep);
  fmax   = fname.length;
  createStruc();
  copy0();
  document.getElementById('fileToLoad').value = '';
  document.getElementById('fileName').value = 'demofile.txt';
  alert('Zuerst mit [Read record] einen Datensatz einlesen!');
  document.getElementById("btn1").focus();
  ransfer = false;
  fileLoad = true;
}

function createFile() {
// Neue Datenbank erzeugen
  if ( fileLoad && !transfer ) { alert('Zuerst Datenbank mit [Close file] schließen!'); return; }
  if ( !txtvisi ) { alert('Zuerst mit [Editor on/off] den Texteditor öffnen!'); return; }
  tansfer = false;
  var text = document.getElementById('tbox').value;
  text = myTrim(text);
  var okay = true;
  if (text == '') { okay = false; }
  if (text.indexOf(sep) == -1) { okay = false; }
  if (text.indexOf(fsep) == -1) { okay = false; }
  if ( !okay ) { alert('Abbruch wegen falscher Datenbank-Struktur!'); return; }
  document.getElementById('tbox').style.visibility = "hidden";
  txtvisi = false;
  file0 = text;
  allrec.length = 0;
  fname.length = 0;
  allrec = file0.split(sep);
  header = allrec.shift();
  rmax   = allrec.length;
  fname  = header.split(fsep);
  fmax   = fname.length;
  // Datenbank neu indizieren
  var s1 = header + sep + '\n';
  for (i = 0; i < rmax; i++) {
    t1 = allrec[i];
    p  = t1.indexOf(fsep);
    t2 = t1.substr(p);
    t1 = (1*i+1) + t2;
    allrec[i] = t1;
    s1 = s1 + allrec[i] + sep;
  }
  file0 = s1.substr(0,s1.length-1);
  createStruc();
  copy0();
  alert('Zuerst mit [Read record] einen Datensatz einlesen!');
  document.getElementById('fileToLoad').value = '';
  document.getElementById('fileName').value = 'myfile.txt';
  document.getElementById("tbox").value = '';
  document.getElementById("btn1").focus();
  fileLoad = true;
}

function closeFile() {
// Datenbank schließen
  if (!fileLoad) { alert('Zuerst Datenbank erzeugen!'); return; }
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if (!transfer) {
    var ein = confirm('Datenbank wirklich schließen?');
    if (!ein) { return; }
  }
```

```
  document.getElementById('fileToLoad').value = '';
  document.getElementById('fileName').value = 'myfile.txt';
  fileLoad  = false;
  readFirst = false;
  filt = false;
  if (!transfer) {
    document.getElementById('tbox').style.visibility = "hidden";
    if (!transfer) { window.location.reload(); }
  }
}

function viewFile() {
// gesamte Listendarstellung von ausgewählten Datenfeldern
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  if (child) { closeChild(); child = false; return; }
  var info = 'Liste der Feldnummern: 1, 2, ... ' + (fmax-1) + ', ' + '\n' + 'a = Ausgabe von ALLEN
Feldern';
  var wahl = '0,1,7,';
  wahl = prompt(info,wahl);
  if (wahl != 'a') {
     var select = 0;
     select = prompt('Statistik von Feldnummer X (0 = KEINE Statistik)',select);
     if ( isNaN(select) || (wahl.indexOf(select) == -1) ) { select = 0; }
  }
  child = true;
  childWindow = window.open('','childWindow','top=10, left=10, width=750, height=600, srollbars=yes,
menubar=no, toolbar=no');
  childWindow.document.open();
  childWindow.document.write('<html><head></head><body id="all" style="background-color:#FFFFE8;
font-family:Courier New, Arial; font-weight:bold; font-size:14px; text-size-adjust:none; margin-
left:20px; margin-bottom:40px;">');
  childWindow.document.write('<br><input type="button" value="Print" onclick="print();"><br><br>');
  var count = 0, mini = 0, maxi = 0, sum = 0, qsum = 0; mwt = 0, str = 0;
  s = '';
  for (j = 0; j < rmax; j++) {
      rec = allrec[j].split(fsep);
      t = '';
      for (i = 0; i < fmax; i++) {
        c = i + ',';
        if (wahl == 'a') { t = t + rec[i] + ';'; }
        if ((wahl != 'a') && (wahl.indexOf(c) > -1)) {
           t = t + rec[i] + ';';
           if ((select > 0) && (i == select)) {
              count++;
              sum = sum + 1*rec[i];
              qsum = qsum + 1*rec[i]*rec[i];
              if (j == 0) { mini = rec[i]; maxi = rec[i]; }
              if (rec[i] < mini) { mini = rec[i]; }
              if (rec[i] > maxi) { maxi = rec[i]; }
           }
        }
      }
      s = s + t + '<br>';
  }
  mwt  = sum / count;
  zahl = qsum/count - (mwt*mwt);
  str  = Math.sqrt(zahl);
  result = 'Feldnummer = ' + select + '<br>' +
           'Anzahl = ' + count + '<br>' +
           'Minimum = ' + mini + '<br>' +
           'Maximum = ' + maxi + '<br>' +
           'Summe = ' + sum + '<br>' +
           'Mittelwert = ' + mwt.toFixed(4) + '<br>' +
           'Streuung = ' + str.toFixed(4) + '<br>' + ' ';
  childWindow.document.write(s);
  if ((wahl != 'a') && (select > 0)) {
     childWindow.document.write('-------------------- Statistik --------------------' + '<br>');
     childWindow.document.write(result);
  }
  childWindow.document.write('</body></html>');
}

function transferData() {
// transferiert die aktuelle Datenbank in den Texteditor
  txt = file0.replace(/#/gi,'#\n');
  document.getElementById('tbox').value = txt;
}

function moveField() {
// Datenfeld verschieben
  var info = 'Ein Feld X nach Y  verschieben (von 1 bis ' + (fmax-1) + ', 0 = Abbruch)';
  var gesucht = prompt(info,'X,Y');
  var s = gesucht.split(',');
```

```
  num0 = myTrim(s[0]); if (isNaN(num0))  { alert('Abbruch: Falsche Feldnummer !'); return; }
  num1 = myTrim(s[1]); if (isNaN(num1))  { alert('Abbruch: Falsche Feldnummer !'); return; }
  if ( (num0 < 1) || (num0 > (fmax-1)) ) { alert('Abbruch: Falsche Feldnummer !'); return; }
  if ( (num1 < 1) || (num1 > (fmax-1)) ) { alert('Abbruch: Falsche Feldnummer !'); return; }
  rec = header.split(fsep);
  removed = rec.splice(num0,1);
  rec.splice(num1,0,removed);
  txt = rec + sep + '\n';
  for (j = 0; j < rmax; j++) {
    rec = allrec[j].split(fsep);
    removed = rec.splice(num0,1);
    rec.splice(num1,0,removed);
    txt = txt + rec + sep + '\n';
  }
  txt = txt.replace(/,/gi,';');
  txt = txt.replace(/\n\n/gi,'\n');
  txt = txt.substring(0,txt.length-2);
  document.getElementById('tbox').value = txt;
  file0 = txt;
  allrec = file0.split(sep);
  header = allrec.shift();
  rmax   = allrec.length;
  fname  = header.split(fsep);
  fmax   = fname.length;
  fileload = false;
}

function insertField() {
// Datenfeld anfügen
  var info = 'Neues Feld X (1 bis ' + fmax + ') einfügen (0 = Abbruch)';
  var num = prompt(info,'X');
  if ( isNaN(num) || (num < 1) || (num > fmax) ) { alert('Abbruch: Falsche Feldnummer !'); return; }
  var info = 'Neuen Feldnamen eingeben';
  var nname = prompt(info,'Feld');
  rec = header.split(fsep);
  removed = rec.splice(num,0,nname);
  txt = rec + sep + '\n';
  for (j = 0; j < rmax; j++) {
    rec = allrec[j].split(fsep);
    removed = rec.splice(num,0,'');
    txt = txt + rec + sep + '\n';
  }
  txt = txt.replace(/,/gi,';');
  txt = txt.replace(/\n\n/gi,'\n');
  txt = txt.substring(0,txt.length-2);
  document.getElementById('tbox').value = txt;
  file0 = txt;
  allrec = file0.split(sep);
  header = allrec.shift();
  rmax   = allrec.length;
  fname  = header.split(fsep);
  fmax   = fname.length;
  fileload = false;
}

function deleteField() {
// Datenfeld entfernen
  var info = 'Feld X (1 bis ' + (fmax-1) + ') entfernen (0 = Abbruch)';
  var num = prompt(info,'X');
  if ( isNaN(num) || (num < 1) || (num > (fmax-1)) ) { alert('Abbruch: Falsche Feldnummer !');
return; }
  rec = header.split(fsep);
  removed = rec.splice(num,1);
  txt = rec + sep + '\n';
  for (j = 0; j < rmax; j++) {
    rec = allrec[j].split(fsep);
    removed = rec.splice(num,1);
    txt = txt + rec + sep + '\n';
  }
  txt = txt.replace(/,/gi,';');
  txt = txt.replace(/\n\n/gi,'\n');
  txt = txt.substring(0,txt.length-2);
  document.getElementById('tbox').value = txt;
  file0 = txt;
  allrec = file0.split(sep);
  header = allrec.shift();
  rmax   = allrec.length;
  fname  = header.split(fsep);
  fmax   = fname.length;
  fileload = false;
}
```

```
function open_() {
// Editor öffnen oder schließen
  if (txtvisi) { alert('[Create file] anklicken!'); return; }
  var info = '\n' +
             ' Datenfelder verschieben mit [move Field, F4] ' + '\n' +
             ' Datenfelder einfügen mit [insert Field, F8] ' + '\n' +
             ' Datenfelder entfernen mit [delete Field, F9] ' + '\n' +
             ' Die Kopfzeile (Header) darf NICHT gelöscht werden! ' + '\n' + '\n ' +
             ' Wenn alles fertig, dann [Create file] !!!' + '\n' + ' ';
  var OK = false;
  if ( fileLoad ) { alert('Zuerst Datenbank mit [Close file] schließen!'); return; }
  txtvisi = !txtvisi;
  if (txtvisi) { alert('Neue Datenbank editieren und dann [Create file] anklicken!'); }
  document.getElementById('tbox').value = file1;
  if (txtvisi) { document.getElementById('tbox').style.visibility = "visible"; }
  else { document.getElementById('tbox').style.visibility = "hidden"; }
  if (txtvisi && transfer && okay) {
    imageName = '';
    loadImage(imageName);
    alert(info);
  }
}

function write_() {
// Datenbank in den Editor schreiben
  if (transfer) { alert('Mit [Create file] eine neue Datenbank erzeugen !'); return; }
  if (!readFirst) { alert('Zuerst mit [Read record] einen Datensatz einlesen!'); return; }
  var info = '\n' +
             ' Datenfelder verschieben mit [move Field] ' + '\n' +
             ' Datenfelder einfügen mit [insert Field] ' + '\n' +
             ' Datenfelder entfernen mit [delete Field] ' + '\n' +
             ' Die Kopfzeile (Header) darf NICHT gelöscht werden! ' + '\n' + '\n ' +
             ' Wenn alles fertig, dann [Create file] !!!' + '\n' + ' ';
  transferData();
  removeStruc();
  txtvisi = true;
  transfer = true;
  closeFile();
  okay = true;
  if (txtvisi) { document.getElementById('tbox').style.visibility = "visible"; }
  else { document.getElementById('tbox').style.visibility = "hidden"; }
  if (txtvisi && transfer && okay) {
    imageName = '';
    loadImage(imageName);
    alert(info);
  }
}

function move_() {
// Ein Datenfeld verschieben
  if (!transfer) { return; }
  if (txtvisi && transfer) { moveField(); }
  if (txtvisi) { document.getElementById('tbox').style.visibility = "visible"; }
  else { document.getElementById('tbox').style.visibility = "hidden"; }
  if (txtvisi && transfer && okay) {
    imageName = '';
    loadImage(imageName);
  }
}

function insert_() {
// Ein  Datenfeld einfügen
  if (!transfer) { return; }
  if (txtvisi && transfer) { insertField(); }
  if (txtvisi) { document.getElementById('tbox').style.visibility = "visible"; }
  else { document.getElementById('tbox').style.visibility = "hidden"; }
  if (txtvisi && transfer && okay) {
    imageName = '';
    loadImage(imageName);
  }
}

function delete_() {
// Ein  Datenfeld löschen
  if (!transfer) { return; }
  if (txtvisi && transfer) { deleteField(); }
  if (txtvisi) { document.getElementById('tbox').style.visibility = "visible"; }
  else { document.getElementById('tbox').style.visibility = "hidden"; }
  if (txtvisi && transfer && okay) {
    imageName = '';
    loadImage(imageName);
  }
}
```

```
</script>
</head>

<body oncontextmenu="return false">
<fieldset id="fs">
<div id="btns">
<h3 id="tit">«idxfile.html» Indexsequentielle Datenbank, Version 10.5</h3>
<input type="button" value="Demo file" class="bs2" onclick="demoFile()">  
<input type="file" id="fileToLoad" class="bs2" onchange="loadFile()">  
<input type="text" id="fileName" class="bs2" size="12" value="myfile.txt">
<input type="button" value="Save file" class="bs2" onclick="saveFile()">  
<input type="button" value="Create file" class="bs2" onclick="createFile()">  
<br><br>
<input type="button" value="Write To Editor" class="bs1" onclick="write_()">  
<input type="button" value="Move Field" class="bs1" onclick="move_();">  
<input type="button" value="Insert Field" class="bs1" onclick="insert_();">  
<input type="button" value="Delete Field" class="bs1" onclick="delete_();">  
<input type="button" value="Editor on/off" class="bs1" onclick="open_()">  
<input type="button" value="Help" class="bs3" onclick="help_()">
<br><br>
<input type="button" value="Seek record" class="bs" onclick="seekRec()">  
<input type="button" value="Print record" class="bs" onclick="printRec()">  
<input type="button" value="Filter on" class="bs" onclick="filterRec()">  
<input type="button" value="Filter off" class="bs" onclick="filterOff()">  
<input type="button" value="Sort file" class="bs" onclick="sortFile()">  
<input type="button" value="View file" class="bs" onclick="viewFile()">  
<input type="button" value="Close file" class="bs" onclick="closeFile()">  
<br><br>
<input type="button" id="btn1" value="Read record" class="bs" onclick="readRec()">  
<input type="button" value="Next record" class="bs" onclick="nextRec()">  
<input type="button" value="Last record" class="bs" onclick="lastRec()">  
<input type="button" value="New record ->" class="bs" onclick="newRec()">  
<input type="button" value="Write record" class="bs" onclick="writeRec()">  
<input type="button" value="Delete record" class="bs" onclick="delRec()">  
<br><br>
</div>
</fieldset>
<br><br>
<img id="pic" src="" onload="loadImage(imageName)">
<textarea id="tbox" value="" cols="85" rows="26" wrap="off" style="visibility:hidden;">
</textarea>
</body>
</html>
```



Die Grafik zeigt den internen Texteditor zur Änderung der Datenstruktur.