

## 9 Übung zu Informatik zum 14.1.2010 Blatt 9

### 9.1

```
program Fachbuecher;
Type verlagstyp = Record
  Verlagsname: String;
  Verlagsort: String; end; {Typ-Verbund Verlagsname und ort}
Fachbuch = Record
  Autor: Array[1..10] of String; {maximal 10 Autoren}
  Titel: String;
  Erschienen: Integer;
  Verlag: verlagstyp;
  Klasse: (Informatik, Mathematik, Physik, Geographie, Didaktik); {5
    Klassen zur Auswahl}
  Seiten: Integer;
  Gewicht: Integer;
  Hoehe: Integer;
  Breite: Integer;
  Kommentar: String;
  Benutzt: Integer;
  Schwerpunkte: String;
  Taschenbuch: Boolean; {Einige Eigenschaften der Bücher, die ausgegeben
    werden}
end;
var Buecher: Array[1..1000] of Fachbuch; Buchanzahl: Integer=0;
procedure rein; {Voreingegebene Beispielbücher}
begin
  Buecher[1].Titel:= 'Taschenbuch der Mathematik'; {Eigenschaften von 2
    Beispielbüchern}
  Buecher[1].Autor[1]:= 'Bronstein'; {werden Beispielwerte
    zugeordnet}
  Buecher[1].Autor[2]:= 'Semendjajew';
  Buecher[1].Autor[3]:= 'Musiol';
  Buecher[1].Autor[4]:= 'Muehling';
  Buecher[1].Erschienen:=2008;
  Buecher[1].Verlag.Verlagsname:= 'Verlag Harri Deutsch';
  Buecher[1].Verlag.Verlagsort:= 'Graefstraße 47, Frankfurt am Main';
  Buecher[1].Klasse:=Mathematik;
  Buecher[1].Seiten:=1221;
  Buecher[1].Gewicht:=2500;
  Buecher[1].Hoehe:=20;
  Buecher[1].Breite:=15;
  Buecher[1].Benutzt:=23;
  Buecher[1].Kommentar:= 'Formelsammlung Mathe/Physik';
  Buecher[1].Schwerpunkte:= 'Themenerklärungen, Formeldetails';
  Buecher[1].Taschenbuch:=True;

  Buecher[2].Titel:= 'Formelsammlung bis zum Abitur';
  Buecher[2].Autor[1]:= 'Becker';
  Buecher[2].Autor[2]:= 'Bossek';
  Buecher[2].Autor[3]:= 'Engelmann';
  Buecher[2].Autor[4]:= 'Ernst';
  Buecher[2].Autor[5]:= 'Liesenberg';
```

```

Buecher[2].Autor[6]:= 'Meyer';
Buecher[2].Autor[7]:= 'Pews-Hocke';
Buecher[2].Autor[8]:= 'Seidel';
Buecher[2].Autor[9]:= 'Simon';
Buecher[2].Autor[10]:= 'Wheser';
Buecher[2].Erschienen:=2009;
Buecher[2].Verlag.Verlagsname:= 'Duden-Paetec';
Buecher[2].Verlag.Verlagsort:= 'Berlin';
Buecher[2].Klasse:=Physik;
Buecher[2].Seiten:=167;
Buecher[2].Gewicht:=700;
Buecher[2].Hoehe:=25;
Buecher[2].Breite:=17;
Buecher[2].Benutzt:=105;
Buecher[2].Kommentar:= 'Formelsammlung_Mathe/Physik/Biologie/Chemie/
    Astronomie/Informatik';
Buecher[2].Schwerpunkte:= 'Formelueberblick ,_Datensammlung';
Buecher[2].Taschenbuch:=False;

Buchanzahl:=2; {Anzahl der von den 1000 möglichen Speicherstellen
    belegten}
end;
function autoraus(index:Integer):String; {gibt Autoren schön formatiert
    aus}
var i:Integer; ausgabe:String;
begin
    ausgabe:= Buecher[index].autor[1]; {erster Autor (auch leer) immer
        vorhanden, da kein Trennzeichen}
    for i:=2 to 10 do
        begin
            if not(Buecher[index].autor[i]='') then ausgabe := ausgabe + ',_' +
                Buecher[index].autor[i] {Autoren mit Komma trennen}
        end;
    autoraus:=ausgabe
end;
procedure raus; {zeigt dem Benutzer die Bücher auf der Oberfläche}
var i:Integer;
begin
    for i:=1 to Buchanzahl do
        begin
            writeln(); {freizeile am Anfang}
            writeln('Titel:.....',Buecher[i].Titel); {Titel: Wert mit Werten
                passend eingezogen}
            if Buecher[i].Autor[2]='' then write('Autor:.....') else write('
                Autoren:.....');
            writeln(autoraus(i)); {bei einem Autor, Autor; bei mehreren, Autoren}
            writeln('Verlag:.....',Buecher[i].verlag.Verlagsname);
            writeln('Verlagsort:...',Buecher[i].verlag.Verlagsort);
            writeln('Erschienen:...',Buecher[i].Erschienen);
            writeln('Seitenanzahl:_',Buecher[i].Seiten);
            writeln('Gewicht:.....',Buecher[i].Gewicht,'g');
            writeln('Hoehe:.....',Buecher[i].Hoehe,'cm');
            writeln('Breite:.....',Buecher[i].Breite,'cm');
        end;
    end;
end;

```

```

writeln( 'Benutzungen:__',Buecher[i].Benutzt, '_mal' );
writeln( 'Kommentar:____',Buecher[i].Kommentar );
writeln( 'Schwerpunkte:_',Buecher[i].Schwerpunkte );
if Buecher[i].Taschenbuch then writeln( 'Taschenbuch:__Ja' ) else writeln
    ( 'Taschenbuch:__Nein' ) ;
write( 'Klasse:_____' );
case Buecher[i].Klasse of { Übersetzung, Bezeichner->String }
  Mathematik: writeln( 'Mathematik' );
  Physik: writeln( 'Physik' );
  Geographie: writeln( 'Geographie' );
  Didaktik: writeln( 'Didaktik' );
  Informatik: writeln( 'Informatik' )
end;
writeln;
if i < Buchanzahl then writeln( '###-----*****-----###' ); {
    auffälliges Trennzeichen zwischen Büchern }
writeln
end;
end;
begin
writeln( 'Hier sehen Sie eine Liste meiner Fachbuecher!' ); { Begrüßung }
rein;
raus;
writeln( 'Druecken Sie Enter zum Beenden!' ); { Abschied }
readln
end.

```

```

program Wortsuche;
var suchwort:String;text:array[1..15] of string;
const
  s1 = 'GAMDUSRJIBUCHENIFORP'; {zu durchsuchender Text}
  s2 = 'OBRGRUNDLAGENIANTKRM';
  s3 = 'EWIQBHLFVXRGJZTFCOER';
  s4 = 'DHEDWQASCRPRUDTOGXKY';
  s5 = 'EIHIOFMOGJCPAUVRYLUE';
  s6 = 'LLWUNQEDLSAEZYAMBARX';
  s7 = 'FEBPJHSZDOTVIMMASWSC';
  s8 = 'GASSEMBLERGPMIXTRQIO';
  s9 = 'UIJOLRAKRUEIVBMIEKOS';
  s10 = 'GBHIETFIDDEFKEEKPRNP';
  s11 = 'EKCCWNJESROCKSMTEAZK';
  s12 = 'ZDOUAPSVUOWUZGQCAFIT';
  s13 = 'PRAHYUUNTILBBAELTYFY';
  s14 = 'DJXLCBGVJTTVIKFLKMCT';
  s15 = 'XPGJKSOBSMGOSPGHJSKA';
function suchehoriz: Boolean; {horizontale Suche}
var i: integer=0;x: integer=0;gefunden: boolean=False;
begin
for i := 1 to 15 do {Durchsuche jede Zeile}
  begin
    x:=pos(suchwort,text[i]); {x ist index des Suchwortes in aktueller Zeile}
  end
  if x>0 then {Wenn gefunden, gefunden=True und zeige an Oberfläche an}
    begin
      gefunden:=True;
      writeln( 'Gefunden, _horizontal_X: ',x, '_Y: ',i)
    end;
  end;
  suchehoriz:=gefunden {Wenn Wort nicht gefunden, bleibt False, sonst auf True}
end;
function suchevert: Boolean; {Vertikale Suche}
var i: Integer=1;j: Integer=1;y: Integer=1;gefunden: Boolean=False; nText:
  array[1..20] of string;
begin
for i:=1 to 20 do nText[i]:= ''; {initialisiere Array}
for i:=1 to 15 do {Wandle das Zeilenarray in ein Spaltenarray um}
  begin
    for j:=1 to 20 do nText[j]:=nText[j]+Text[i][j]; {übereinanderstehende
      Zeichen werden in ein Stringelement geschrieben}
  end;
for i :=1 to 20 do {durchsuche jede Spalte}
  begin
    y:=pos(suchwort,nText[i]); {y ist index des Suchwortes in Spalte}
    if y>0 then {Wenn gefunden, dann ausgabe}
      begin
        writeln( 'Gefunden, _vertikal_X: ', i, '_Y: ', y); {da Zeilenarray->
          Spaltenarray: y:suchindex in spalte, x:Schleifendurchlauf}
        gefunden:=true
      end
    end
  end

```

```

end;
end;
suchever:=gefunden {Wenn gefunden, gebe True zurück}
end;
function suchediagltor:Boolean; {Suche Diagonal links oben nach rechts
    unten}
var i,j,x,y:Integer;nText:Array[1..34] of String;gefunden:Boolean=False;
begin
    for i:=1 to 34 do nText[i]:=''; {Initialisiere Array}
    for i:=1 to 34 do
        begin {wandle Zeilenarray in Diagonalarray um}
        for j:=1 to 20 do if (0 < (j-20+i)) and (16 > (j-20+i)) then nText[i]:=
            nText[i]+Text[j-20+i][j];
        end;
        for i:=1 to 20 do {Durchsuche obere Diagonale Hälfte (Verfahren anders)}
        begin
            y:=pos(suchwort,nText[i]); {y:Index des Suchwortes in Diagonale=index
                in Zeile}
            x:=20-i+y; {da Zeilenarray->Diagonalarray: x:Index in Spalte abhängig
                von y}
            if y>0 then {Wenn gefunden, Ausgabe}
            begin
                writeln('Gefunden, \Diagonal \nach \rechts \unten \X:', x, '\Y:',y);
                gefunden:=True
            end;
        end;
        for i:=21 to 34 do {Durchsuche untere Diagonale Hälfte (Verfahren anders)}
        begin
            x:=pos(suchwort,nText[i]); {x:Index des Suchwortes in Diagonale=index
                in Spalte}
            y:=-20+i+x; {da Zeilenarray->Diagonalarray: y:Index in Zeile abhängig
                von x}
            if x>0 then {Wenn gefunden, Ausgabe}
            begin
                writeln('Gefunden, \Diagonal \nach \rechts \unten \X:', x, '\Y:',y);
                gefunden:=True
            end;
        end;
        suchediagltor:=gefunden {Wenn gefunden, gebe True zurück}
    end;
end;
function suchediagrtol:Boolean; {Suche Diagonal rechts oben nach links
    unten}
var i,j,x,y:Integer;nText:Array[1..34] of String;gefunden:Boolean=False;
begin
    for i:=1 to 34 do nText[i]:=''; {Initialisiere Array}
    for i:=1 to 34 do
        begin
            for j := 1 to 15 do
                begin {Wandle Zeilenarray in Diagonalarray um}
                if (0<(i+1-j))and (21>(i+1-j)) then nText[i]:=nText[i]+Text[j][i+1-j]
                end;
            end;
        end;
        for i:=1 to 20 do {obere Hälfte, da Verfahren anders}

```

```

begin
y:=pos(suchwort,nText[i]); {y:Index in Zeile=Index des Suchwortes in
    Diagonale}
x:=i-y+1; {Da Zeilenarray->Diagonalarray: x (Spaltenindex) abhängig von y
    }
if y>0 then {Wenn gefunden, Ausgabe}
begin
writeln('Gefunden, \Diagonal\_nach\_links\_unten\_X:', x, '\_Y:',y);
gefunden:=True
end;
end;
for i:=21 to 34 do {untere Hälfte, da Verfahren anders}
begin
x:=pos(suchwort,nText[i]); {x:Indes in Spalte=Index des Suchwortes in
    Diagonale}
y:=-i+y+1;{Da Zeilenarray->Diagonalarray:y (Zeilenindex) abhängig von x}
if x>0 then {Wenn gefunden, dann Ausgabe}
begin
writeln('Gefunden, \Diagonal\_nach\_links\_unten\_X:', x, '\_Y:',y);
gefunden:=True
end;
end;
suchediagrtol:=gefunden; {Wenn gefunden, gebe True zurück}
end;
procedure eingabe; {Suchworteingabe}
var gefunden:Boolean=False;
begin {Begrüßung}
    writeln('Bitte\_Geben\_Sie\_ein\_Suchwort\_ein, \dessen\_Koordinaten\_im\_Text\_
        ausgegeben\_werden\_sollen.\_”quit”\_beendet\_das\_Programm');
    write('Suchwort: ');
    readln(suchwort); {Suchwort eingeben}
    if not(suchwort='quit') then {Bei quit beenden}
    begin
        if suchehoriz then gefunden :=True;
        if suchevert then gefunden := True;
        if suchediagltor then gefunden := True;
        if suchediagrtol then gefunden := True;
        if not gefunden then writeln('Text\_nicht\_gefunden!'); {Wenn keine
            Suche erfolgreich}
        writeln(); {Leerzeile (Lesbarkeit)}
        eingabe
    end;
end;
begin {Hauptprozedur}
text[1]:=s1;{wandle Konstanten in ein String-Array um, für
    Schleifendurchläufe}
text[2]:=s2;
text[3]:=s3;
text[4]:=s4;
text[5]:=s5;
text[6]:=s6;
text[7]:=s7;
text[8]:=s8;

```

```
text [9] := s9;  
text [10] := s10;  
text [11] := s11;  
text [12] := s12;  
text [13] := s13;  
text [14] := s14;  
text [15] := s15;  
  eingabe {Suchworteingabe}  
end.
```

### 9.3

```
program Frohe_Weihnachten;  
begin { Hauptprozedur }  
writeln( 'Frohe_Weihnachten ... ' ); { Textausgabe }  
readln  
end.
```



## 9.4

Die Vasallen teilen sich in 2 Gruppen mit je 15 Personen auf.

Diese 2 Gruppen stellen sich nun an den beiden Waagschalen der Waage an. Dann legen die beiden Vasallen, die direkt an der Waage stehen 1 Goldstück (oder 30, für ein klareres Ergebnis) auf die Waagschale.

Wenn beide Vasallen ihr Gold auf die Waagschalen gelegt haben, kommen die nächsten 2 Vasallen. Ist dabei nach 2 Vasallen die Waage nicht im Gleichgewicht, hat der Vasall mit der leichteren Waagschale den König betrogen.