

```

1: // Aufgl_Felder002_Zeig.cpp
2:
3: //*****
4: // Programm: Programm ermittelt die minimale Distanz der Elemente eines
5: //           2-dimensionalen Feldes zu einem Vergleichswert
6: //           Loesung auf Zeigerarithmetik basierend
7: // Dateiname: Aufgl_Felder002_Zeig.c
8: // Autor:    L. Eckert
9: // Datum:    02.11.2003
10: //*****
11: #include "stdafx.h"
12: #include <stdio.h>
13: #include <conio.h>
14: #include <float.h>
15:
16:
17: // Konstantendefinition
18: #define MAXZ 4           // Anzahl Zeilenelemente
19: #define MAXS 4           // Anzahl Spaltenelemente
20:
21:
22:
23: // Funktionsdeklaration
24: double Abs(double wert);
25: void MinDist(double Feld[][MAXS], double Vergleichszahl, int *PosZ, int *PosS,
26:             double *MinDistanz);
27:
28:
29:
30: // Hauptprogramm
31: int main(int argc, char* argv[])
32: {
33:     double Feld[4][4];
34:
35:     // für den Test der Funktionen MinDist() und Abs()
36:     //   double Feld[4][4] = {1,2,3,4,
37:     //                       5,6,7,8,
38:     //                       9,10,11,12,
39:     //                       13,14,15,16}; // Deklaration und Initialisierung
40:
41:
42:     double Vergleichszahl;
43:     int i=0, j=0;
44:     int PosZ, PosS;
45:     double MinDistanz;
46:
47:     // Vergleichselement einlesen
48:     printf("\nBitte Vergleichszahl einlesen: ");
49:     scanf("%lf", &Vergleichszahl);
50:     printf("\nVergleichszahl: %lf \n\n", Vergleichszahl);
51:
52:
53:     // Einlesen der Zahlen in das Zahlenfeld feld[100]
54:     printf("\nBitte geben Sie %d Feldelemente ein: ", MAXZ*MAXS);
55:     for (i=0; i<MAXZ; i++)
56:     {
57:         for (j=0; j<MAXS; j++)
58:             scanf("%lf", (*(Feld+i)+j) ); // statt &Feld[i][j]);
59:     }
60:
61:     MinDist(Feld, Vergleichszahl, &PosZ, &PosS, &MinDistanz); // Funktionsaufruf
62:     // oder MinDist(&Feld[0][0], Vergleichszahl, &PosZ, &PosS, &MinDistanz);
63:
64:     // Ergebnisausgabe
65:     printf("Vergleichszahl: %lf \n", Vergleichszahl);
66:     printf("Am naechsten liegt das Feldelement Feld[%d][%d],
67:           mit der Distanz: %lf. \n" , PosZ+1, PosS+1, MinDistanz);
68:     return 1;
69: }
70:
71:
72:
73: // Funktion gibt den Betrag eines Wertes zurück
74: double Abs(double wert)
75: {
76:     if (wert<0) return (-wert);
77:     else return wert;
78: }

```

```

80:
81:
82: // Ermittle das Feldelement mit seinen Indizes, das betragsmaessig den kleinsten
83: // Abstand zum Vergleichselement hat
84: void MinDist(double Feld[][MAXS], double Vergleichszahl, int *PosZ, int *PosS,
85:             double *MinDistanz)
86: {
87:     double aktDistanz;
88:     int i, j;
89:
90:     // Allgemeines
91:     // zum Inhalt
92:     //      **Feld = Feld[0][0]   Zugriff auf das 1. Feldelement der
93:     //                               Matrix Feld
94:     //      *(Feld+i) = Feld[i][0]   Zugriff auf das 1. Feldelement der
95:     //                               i-ten Zeile
96:     //      *( *(Feld+i) + j) = Feld[i][j]   Zugriff auf ein beliebiges
97:     //                               Feldelement [i-ten Zeile][j-ten Spalte]
98:     //
99:     //      *(Feld+i) = &Feld[i][0]   Zugriff auf Adresse des 1.Feldelementes
100:    //                               der i-ten Zeile
101:
102:    // zur Adresse
103:    //      Feld = &Feld[0][0]
104:    //      Feld+i = &Feld[i][0]
105:    //      (Feld+i)+1 = &Feld[i+1][0]   an das j-Element gelangt man nur über
106:    //                               den * Operator
107:    //      *(Feld+i) = &Feld[i][0]
108:    //      *(Feld+i)+j = &Feld[i][j]
109:
110:    // zu Zeigerarithmetik
111:    //      double Feld [MAXZ][MAXS];
112:    //      double (*zeilenPtr)[MAXZ] = Feld;   ein Array von MAXZ Zeigern auf
113:    //                                           jeweils ein double
114:    // dann gilt
115:    //      (*ZeilenPtr++)   führt zur nächsten Zeile
116:    //      (*ZeilenPtr++)[j]   ist das j-te Element der naechsten Zeile
117:
118:
119:    // Ausgabe des Feldes mit Zugriff basierend auf Zeigerarithmetik
120:    for (i=0; i<MAXZ; i++)
121:        {
122:            for (j=0; j<MAXS; j++)
123:                printf(" %lf", *( *(Feld+i) +j) );
124:            printf(" \n");
125:        }
126:    printf(" \n");
127:
128:
129:    // index-basierender Zugriff auf die Feldelemente
130:    //      for (i=0; i<MAXZ; i++)   // Ausgabe des Feldes
131:    //      {
132:    //          for (j=0; j<MAXS; j++)
133:    //              printf(" %lf", Feld[i][j]);
134:    //          printf(" \n");
135:    //      }
136:    //      printf(" \n");
137:
138:    *MinDistanz = DBL_MAX;   // Maximum value of double
139:
140:    for (i=0; i<MAXZ; i++)
141:        {
142:            for (j=0; j<MAXS; j++)
143:                {
144:                    aktDistanz = *( *(Feld+i) +j) - Vergleichszahl;
145:                    aktDistanz = Abs(aktDistanz);
146:
147:                    if (aktDistanz < *MinDistanz)
148:                        {
149:                            *MinDistanz = aktDistanz;
150:                            *PosZ = i;
151:                            *PosS = j;
152:                        }
153:                }
154:        }
155: }

```