

Kapitel 9 Beispiel 2

Zunächst die Header-Dateien:

```
1  #ifndef EINGABEMESSWERTE_H
2  #define EINGABEMESSWERTE_H
3  void eingabeMesswerte(float[], float[], int*);
4  #endif

1  #ifndef AUSGABEFUNKTION_H
2  #define AUSGABEFUNKTION_H
3  void ausgabeFunktion(float[], float[], int, float*, float*);
4  #endif

1  #ifndef GEGENUEBERSTELLUNGWERTE_H
2  #define GEGENUEBERSTELLUNGWERTE_H
3  void gegenueberstellungWerte(float[], float[], int, float, float);
4  #endif

1  #ifndef GEGENUEBERSTELLUNGABWEICHUNG_H
2  #define GEGENUEBERSTELLUNGABWEICHUNG_H
3  void gegenueberstellungAbweichung(float[], float[], int, float, float);
4  #endif
```

```

1 // Programm 9.2.cpp: Hauptprojektdatei.
2 // Programm zur Berechnung von Ausgleichsgeraden
3 // nach der Fehlerquadrat-Methode mit Menuesteuerung
4 // Autor: Heiderich / Meyer
5 // -----
6 #include "stdafx.h"
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <conio.h>
10 #include "eingabeMesswerte.h"
11 #include "ausgabeFunktion.h"
12 #include "gegenueberstellungWerte.h"
13 #include "gegenueberstellungAbweichung.h"
14 using namespace System;
15 void main()
16 {
17     char auswahl;        // Steuerungsgröße, Auswahl aus Menue
18     int wieder = 0;      // Steuerungsgröße zum Löschen des Monitors
19     float mess_x[100];   // x-Koordinaten der Messwerte
20     float mess_y[100];   // y-Koordinaten der Messwerte
21     int anzahl;          // Anzahl der Messwerte (Eingabe)
22     float m;             // Steigung der Ausgleichsgeraden
23     float b;            // y-Achsen-Schnittpunkt Ausgleichsgerade
24     bool ok1 = false;    // Menue-Punkt (1) abgearbeitet
25     bool ok2 = false;    // Menue-Punkt (2) abgearbeitet
26     // Begrüßung
27     do
28     {
29         // Ausgabe des Hauptmenues
30         // ggfls. Bildschirm löschen
31         if (wieder == 0)
32         {
33             printf("\n\n\tAusgleichsrechnung f%cr eine  
linearer Funktion\n",char(129));
34             printf("\t-----\n");
35             wieder = 1;
36         }
37         else system("cls");
38         printf("\n\tM e n u e\n\n");
39         printf("\tEingabe der Messwerte ..... (1)\n");
40         printf("\tAusgabe der linearen Funktion ..... (2)\n");
41         printf("\tGegenüberstellung\n",char(129));
42         printf("\tFunktionswerte - Messwerte ..... (3)\n");
43         printf("\tAbweichungen ..... (4)\n");
44         printf("\n\tProgrammende ..... (5)\n");
45         printf("\n\tBitte wählen Sie: ",char(132));
46         fflush(stdin);
47         scanf("%c",&auswahl);
48         // Steuerung über auswahl
49         switch(auswahl)
50         {
51             case '1': eingabeMesswerte(mess_x, mess_y, &anzahl);
52                       // sind neue Messwerte eingegeben, so muss (2)
53                       // (2) erneut aufgerufen werden vor (3) oder (4)
54                       ok1 = true;
55                       ok2 = false;
56                       break;
57             case '2': if (ok1)
58                     {
59                         ausgabeFunktion(mess_x,mess_y,anzahl,&m,&b);
60                         ok2 = true;
61                     }
62                     else
63                     {
64                         printf("\n\n\tzunächst Messwerte  
eingegeben!!",char(132));
65                         getch();
66                     }
67                     break;
68             case '3': if (ok2)
69                     {
70                         gegenueberstellungWerte(mess_x,mess_y,anzahl,  
m, b);
71                     }
72                     else
73                     {
74                         if (ok1) printf("\n\n\tzunächst lineare  
Funktionen ausgeben!",char(132));
75                         else printf("\n\n\tzunächst  
Messwerte eingeben!!",char(132));

```

```

76         getch();
77     }
78     break;
79     case '4': if (ok2)
80     {
81         gegenueberstellungAbweichung(mess_x,mess_y,
                                         anzahl, m, b);
82     }
83     else
84     {
85         if (ok1) printf("\n\n\tzun%cchst lineare
                        Funktion ausgeben!!",char(132));
86         else     printf("\n\n\tzun%cchst
                        Messwerte eingeben!!",char(132));
87         getch();
88     }
89     break;
90     case '5': break;
91     default: printf("\n\n\tfehlerhafte Eingabe (%c)\n",
                        auswahl);
92     printf("\tbitte korrigieren!");
93     getch();
94 }
95 }while(auswahl!='5');
96 }

```

```

1 // Funktion zum Einlesen von Messwerten
2 // Autor: Heiderich / Meyer
3 // -----
4 #include "stdafx.h"
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include <conio.h>
8 void eingabeMesswerte(float x[], float y[], int* anz)
9 {
10     int anzahl, i;
11     printf("\n\n\n\tEingabe der Messwerte\n");
12     printf("\t-----\n\n");
13     do
14     {
15         printf("\tBitte geben Sie die Anzahl ( < 100 !) der
16             Messwerte an: ");
17         fflush(stdin);
18         scanf("%i",&anzahl);
19     }while (anzahl >= 100);
20     *anz = anzahl;
21     for (i = 0; i < anzahl; i++)
22     {
23         printf("\n\tx[%2i] = ",i+1);
24         fflush(stdin);
25         scanf("%f",&x[i]);
26         printf("\ty[%2i] = ",i+1);
27         fflush(stdin);
28         scanf("%f",&y[i]);
29     }

```

```

1 // Funktion zur Berechnung und
2 // zur Ausgabe der linearen Funktion
3 // Autor: Heiderich / Meyer
4 // -----
5 #include "stdafx.h"
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <conio.h>
9 void ausgabeFunktion(float x[],float y[],int anz,float* m,float* b)
10 {
11     float sxy = 0.0; // Summe xi*yi
12     float sx2 = 0.0; // Summe xi*xi
13     float sx = 0.0; // Summe über xi
14     float sy = 0.0; // Summe über yi
15     float sxsy = 0.0; // Produkt der Summen über x und y
16     int i;
17     for (i = 0; i < anz; i++)
18     {
19         sxy = sxy + x[i] * y[i];
20         sx2 = sx2 + x[i] * x[i];
21         sx = sx + x[i];
22         sy = sy + y[i];
23     }
24     *m = (anz * sxy - sx * sy) / (anz * sx2 - sx * sx);
25     *b = (sx2 * sy - sx * sxy) / (anz * sx2 - sx * sx);
26     // Ausgabe der linearen Funktion
27     printf("\n\ndie lineare Funktion zu den %2i
        Koordinatenpaaren lautet:",anz);
28     printf("\n\n\t y = %f * x ",*m);
29     if (*b >= 0.0) printf("+ %f",*b);
30     else printf("%f",*b);
31     getch();
32 }

```

```

1 // Funktion zur Gegenüberstellung
2 // Messwerte / Funktionswerte
3 // Autor: Heiderich / Meyer
4 // -----
5 #include "stdafx.h"
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <conio.h>
9 void gegenueberstellungWerte(float x[],float y[],int anz,float m,
                             float b)
10 {
11     int i;
12     float yre; // rechnerischer Funktionswert
13     printf("\n\n\t+-----+-----+\n");
14     printf("\t| Wertepaar (x|y) | y - errechnet |\n");
15     printf("\t+-----+-----+\n");
16     printf("\t| x | y | |\n");
17     printf("\t+-----+-----+\n");
18     for (i = 0; i < anz; i++)
19     {
20         yre = m * x[i] + b;
21         printf("\t| %9.3f | %9.3f | %9.3f |\n",x[i],y[i],yre);
22     }
23     printf("\t+-----+-----+-----+\n");
24     getch();
25 }

```

```

1 // Funktion zur Ausgabe der Abweichungen
2 // Autor: Heiderich / Meyer
3 // -----
4 #include "stdafx.h"
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include <conio.h>
8 #include <math.h>
9 void gegenueberstellungAbweichung(float x[],float y[],int anz,
                                   float m, float b)
10 {
11     float sx2 = 0.0; // Summe xi*xi
12     float sx = 0.0; // Summe über xi
13     float sv2 = 0.0; // Summe über vi*vi
14     float delta_m; // mittlerer Fehler von m
15     float delta_b; // mittlerer Fehler von b
16     int i;
17     for (i = 0; i < anz; i++)
18     {
19         sx2 = sx2 + x[i] * x[i];
20         sx = sx + x[i];
21         sv2 = sv2 + pow((y[i] - (m * x[i] + b)),2);
22     }
23     delta_b = sqrt( (sx2*sv2)/((anz-2)*anz*sx2-(anz-2)*sx*sx) );
24     delta_m = sqrt( anz * sv2 / (anz * sx2 - sx * sx) );
25     // Ausgabe der mittleren Fehler:
26     printf("\n\ndie Abweichungen betragen zu den %2i
           Koordinatenpaaren betragen:",anz);
27     printf("\n\tdelta_m = %f\n",delta_m);
28     printf("\tdelta_b = %f\n",delta_b);
29     printf("\n\tFehlerquadratsumme = %f",sv2);
30     getch();
31 }

```