

```
1 // StrukturSchueler.cpp : Definiert den Einstiegspunkt für die Konsolenanwendung.
2 //
3 #include "stdafx.h"
4 #include <iostream>
5 #include <string>
6 using namespace std;
7
8 struct Schueler { // die Struktur Schueler deklarieren
9     int Matrikelnummer;
10    string Vorname;
11    string Nachname;
12    float Note;
13 };
14
15 int main()
16 {
17     // Struktur verwenden mit und ohne Initialisierung in der Definition
18     Schueler ITAler1;
19     Schueler ITAler2 = {1234, "Max", "Mustermann", 3.0};
20     ITAler1.Matrikelnummer = 2345;
21     ITAler1.Vorname = "Peter";
22     ITAler1.Nachname= "Pan";
23     ITAler1.Note = 4.5;
24
25     // Strukturdaten ausgeben
26     cout << "Daten des Schuelers ITAler1: \n";
27     cout << "Der Schueler " << ITAler1.Vorname << " " << ITAler1.Nachname <<
28         " hat die Matrikelnummer " << ITAler1.Matrikelnummer << " und die Note " <<
29         ITAler1.Note << endl << endl;
30
31     cout << "Daten des Schuelers ITAler2: \n";
32     cout << "Der Schueler " << ITAler2.Vorname << " " << ITAler2.Nachname <<
33         " hat die Matrikelnummer " << ITAler2.Matrikelnummer << " und die Note " <<
34         ITAler2.Note << endl << endl;
35
36     system("pause");
37     return 0;
38 }
39
40
```

```
1 // SchuelerArray.cpp : Definiert den Einstiegspunkt für die Konsolenanwendung.
2 //
3 #include "stdafx.h"
4 #include <iostream>
5 #include <string>
6 using namespace std;
7
8 struct Schueler { // die Struktur Schueler deklarieren
9     int Matrikelnummer;
10    string Vorname;
11    string Nachname;
12    float Note;
13 };
14
15 int main()
16 {
17     Schueler ITA[10] = { // Struktur verwenden in einem Array
18         {1234, "Max", "Mustermann", 3.0},
19         {2345, "Peter", "Pan", 4.0},
20         {3456, "Lieschen", "Mueller", 2.0}
21     };
22
23     for (int Nr=0; Nr<3; Nr++) { // Struktur-Datenfeld ausgeben
24         cout << "Daten des Schuelers Nr: " << Nr+1 << "\n";
25         cout << "Der Schueler " << ITA[Nr].Vorname << " " << ITA[Nr].Nachname <<
26             " hat die Matrikelnummer " << ITA[Nr].Matrikelnummer << " und die Note " <<
27         <<
28             ITA[Nr].Note << endl << endl;
29     }
30     system("pause");
31     return 0;
32 }
33
```

```
1 // Aufgabe3.cpp : Definiert den Einstiegspunkt für die Konsolenanwendung.
2 //
3 #include <iostream>
4 #include <string>
5 using namespace std;
6
7 int main() {
8
9     // Anlegen der Struktur "Kunde":
10    struct Kunde {
11        int Kundennummer;
12        string Kundenname;
13    };
14
15    int Eingabe;          // für Aufgabe 4
16    bool gefunden=false; // für Aufgabe 4
17
18    Kunde Kunde1 = {123456, "Max Mustermann"};
19    Kunde Kunde2 = {234567, "Lieschen Mueller"};
20
21    cout << "Kunde 1 hat folgende Stammdaten:\n";
22    cout << "Kundennummer: " << Kunde1.Kundennummer << endl;
23    cout << "Kundenname:  " << Kunde1.Kundenname << endl << endl;
24
25    cout << "Kunde 2 hat folgende Stammdaten:\n";
26    cout << "Kundennummer: " << Kunde2.Kundennummer << endl;
27    cout << "Kundenname:  " << Kunde2.Kundenname << endl << endl;
28
29    // ##### Aufgabe 4: #####
30    // Datenfeld für 5 Kunden anlegen:
31    Kunde MeinKunde[5] = {
32        {123456, "Max Mustermann"},
33        {234567, "Lieschen Mueller"},
34        {345678, "Fernando Alonso"},
35        {456789, "Obi Wan Kenobi"},
36        {567890, "Hans im Glueck"} // beim letzten Datum KEIN Komma setzen!
37        // sonst meint der Compiler, jetzt kommt noch ein
38    Kunde...
39    };
40
41    // So nicht als Aufgabe genannt - aber es übt ungemein:
42    cout << "Kd-Nr.: \tKundenname:\n";
43    for (int x=0; x<5; x++) {
44        cout << MeinKunde[x].Kundennummer << "\t" << MeinKunde[x].Kundenname << endl;
45    }
46    cout << endl;
47
48    // Suchen nach einer Kundennummer:
49    cout << "Welche Kundennummer suchen Sie: ";
50    cin >> Eingabe;
51
52    gefunden = false;
53
54    for (int x=0; x<5; x++) {
55        if (Eingabe==MeinKunde[x].Kundennummer) {
56            cout << "Kundennummer " << Eingabe << " gehoert zu " << MeinKunde
```

```
[x].Kundenname << endl << endl;
56     gefunden = true;
57     }
58     }
59     // Nach dem Durchsuchen per Zählschleife:
60     if (!gefunden) {
61         cout << "Kundennummer " << Eingabe << " leider nicht gefunden..." << endl << ↵
        endl;
62     }
63
64     // Jetzt eine alternative Komfort-Lösung mit Wiederholungsfunktion:
65     do {
66         cout << "Welche Kundennummer suchen Sie: (Ende mit 0) ";
67         cin >> Eingabe;
68
69         if (Eingabe!=0) { // Benutzer hat eine Kundennummer eingegeben (nicht die 0 ↵
            für Ende):
70             // der eigentliche "Kern" ist ebenfalls abgewandelt:
71             // do-while- statt Zählschleife läuft nur solange der Eintrag nicht ↵
            gefunden wurde.
72             // Bei 100.000 Kunden macht sich das durchaus bemerkbar...
73             gefunden = false;
74
75             int x=-1;
76             do {
77                 x++;
78                 if (Eingabe==MeinKunde[x].Kundennummer) {
79                     gefunden = true;
80                 }
81             } while (!gefunden && x<4); // Obacht: dran denken, dass in der Schleife ↵
            sogleich x erhöht wird...
82             // wenn Kundennummer gefunden, steht in x der Index des gefundenen ↵
            Kunden!
83
84             // Nach dem Durchsuchen per do-while-Schleife:
85             if (!gefunden) {
86                 cout << "Kundennummer " << Eingabe << " leider nicht gefunden..." << ↵
                endl << endl;
87             } else {
88                 cout << "Kundennummer " << Eingabe << " gehoert zu " << MeinKunde ↵
                [x].Kundenname << endl << endl;
89             }
90         }
91     } while (Eingabe!=0);
92
93     system("pause");
94     return 0;
95 }
96
```