Ein- und mehrdimensionale Arrays

Eindimensionale Arrays

Möchte man (wie in den einführenden Bemerkungen zu Arrays) mehrere Werte eines Datentyps speichern, so lässt sich das mit einem (eindimensionalen) Array realisieren.

Statt:

```
int Wert1;
int Wert2;
:
:
int Wert10;
```

} `int Werte[10];`

In den eckigen Klammern wird die Anzahl (Tiefe) des Feldes angegeben. Damit stehen zehn Werte vom Datentyp int zur Verfügung.

Beispiel:

Das Array *Werte* soll mit Zahlen gefüllt werden.

```
int Werte[10];
int i;

for ( i = 0; i < 10 ; i++ )  Werte[i] = i*10;
```

Die Elemente des Arrays werden mit dem so genannten **Indexoperator [ ]** angesprochen.

Wenn man ein spezielles Element ansprechen will, so geschieht das mithilfe des Indexoperators und der Angabe des Index.

Nach der Schleife ist das Array *Werte* so gefüllt:

Index i	0	1	2	3	4	5	6	7	8	9
Inhalt: Werte[i]	0	10	20	30	40	50	60	70	80	90

ACHTUNG:

In C++ wird zwar ein Array mit zehn Elementen erstellt, aber der Index läuft von 0 bis 9.

Das hat damit zu tun, dass in C++ die Arrays und die Zeiger bzw. die Zeigerarithmetik ein Konzept bilden. Im Kapitel Zeiger wird diese Problematik noch einmal aufgegriffen.

Die Problematik aus den einführenden Bemerkungen zu Arrays, dass ein Benutzer zehn Zahlen eingibt und das Programm den kleinsten Wert (Minimum) berechnet, kann nun mit den Arrays deutlich eleganter gelöst werden.

Beispiel:

```
#include <iostream>
using namespace std;

int main()
{
const int MAX = 10;
int Werte[MAX];
int Minimum;
int i;
for ( i = 0 ; i < MAX ; i++ )
{
   cout << "Integerwert Nr. " << i+1 << " eingeben:" ;
   cin >> Werte[i];
}
```