

# Übersicht

Lehrinhalt: Programmierung in C

- Überblick über Programmiersprachen, Allgemeines
- C-Basisdatentypen, Zahlendarstellung, Variablen, Konstanten
- Operatoren und Ausdrücke
- Anweisungen
- Kontrollstrukturen
- Funktionen
- **Zeiger und Felder**
- Zeichenketten (Strings)
- Benutzerdefinierte Datentypen
- Dynamischer Speicher
- Dateiarbeit
- Funktionspointer, Rekursion
- Preprozessor

# Felder in C

**Felder** stellen eine **Reihung von Elementen gleichen Typs** dar. Man spricht auch von Vektoren oder Arrays. Durch die Reihung (hintereinander Speichern) wird eine Ordnungsrelation über die Nummer (Position) des Elements festgelegt. Über diese Nummer (Index) wird der Zugriff zum betreffenden Element vorgenommen. In C hat das erste Element die Nummer 0.

Element 0	Element 1	Element 2	...	Element n-1
-----------	-----------	-----------	-----	-------------

In C werden Felder durch die Verwendung der Indexklammern [ ] gekennzeichnet:

Beispiel: `int vektor[10];`

# Deklaration von Feldern in C

Felder können wie Variable deklariert werden. Die Deklaration legt den Typ der Elemente, den Namen des Feldes und die Anzahl der Elemente fest.

Beispiel:

```
double a[10];
```

`double` ist der Typ der Elemente

`a` ist der Variablenname des Feldes

`10` ist die Anzahl der Elemente ( $a_0 a_1 a_2 \dots a_9$ )

# Deklaration von Feldern in C

```
double a[10];
```

Man beachte, dass die Anzahl der Elemente eine Konstante sein muss (gilt für Microsoft C-Compiler).

Ist diese Anzahl bei der Programmierung (vor dem Ausführen des Programms) nicht bekannt, so muss trotzdem das Feld mit einer konstanten, maximal notwendigen Größe deklariert werden.

Innerhalb dieser maximalen Größe kann jeweils mit unterschiedlich vielen Elementen gearbeitet werden. Der Speicherplatz ist für die maximale Anzahl reserviert, aber es werden aktuell ggf. weniger Elemente belegt.

# Initialisierung von Feldelementen / Zugriff auf Feldelemente

Feldelemente können bei der Deklaration mit Werten belegt werden.

Beispiel:

```
double a[10]={-3.22, 0.0, 1, -7.234, 55.5, 6.6,  
             -0.77, 8, 0.09, 3};
```

Der Zugriff auf die Elemente eines Feldes erfolgt über den Index (Positionsnummer) des Elementes. Dieser Index wird in der Indexklammer angegeben.

Beispiel: es soll auf das 7.Element von a zugegriffen werden:

```
x = a[6];
```

(da das 1.Element den Index 0 hat, ist 6 der Index des 7.Elementes)

Bei Zugriffen auf Feldelemente ist stets sicherzustellen, dass der Index in den Deklarationsgrenzen des Feldes liegt.

# Beispiel mit Feld

```
#include <stdio.h>
//arithmetisches Mittel
// einer Messreihe

int main()
{ int i,n;
  float a[50],s;
  printf("Eingabe Anzahl=");
  scanf("%d",&n);    // n<=50
  printf("Eingabe des Vektors:\n");
  for (i=0; i<n; i=i+1)

  {   printf("a[%2d]=",i);
      scanf("%f", &a[i] );
  }
```

```
// Fortsetzung
s = 0;

for (i=0;i<n; i=i+1)
    s = s + a[i];

s = s / n;

printf("ar. Mittel=%f \n",s);
return 0;
}
```

Zählschleifen (for) sind die  
Typischen Steuerfluss-Konstrukte  
zur Verarbeitung der Felder

# Mehrdimensionale Felder

Mehrdimensionale Felder (z.B. Matrizen als 2-dimensionale Felder in der Mathematik) werden analog zum eindimensionalen Fall deklariert.

Eine zusätzliche Dimension wird durch eine weitere Indexklammer angegeben.

Beispiel: `float matrix[3][4];`

dies ist eine Matrix mit 3 Zeilen und 4 Spalten. Diese Matrix hat die Struktur:

$$\begin{array}{cccc} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{array}$$

## Regel:

Bei der Deklaration wird zuerst die Zeilenanzahl, danach die Spaltenanzahl geschrieben.

# Initialisierung mehrdimensionaler Felder

Mehrdimensionale Felder können ebenfalls bei der Deklaration initialisiert werden. Bei zweidimensionalen Feldern werden nacheinander die Inhalte der Zeilen festgelegt.

Beispiel:

```
float matrix[3][4]={ {1,2,3,4},  
                     {5,6,7,8},  
                     {9,10,11,12}  
};
```

Größenangabe [3][4]: zuerst Zeilenanzahl, danach Spaltenanzahl

Initialisierung:

Äußere Reihung sind 3 Zeilen, Innerer Reihung 4 Spalten



# Mehrdimensionale Felder / Zugriff auf Elemente

```
float a = matrix[2][3];
```

Da die Indizierung mit Null beginnt, selektiert die Angabe *matrix[2][3]* die dritte Zeile, und darin das vierte Element, also 12.

**Regel:** Der Zugriff auf Elemente erfolgt analog zur Deklaration durch Angabe der Indexklammern und Indexwerte in der Reihenfolge Zeilenindex, Spaltenindex.

# Mehrdimensionale Felder - Beispiel

Beispiel Matrix-Transposition:

```
float matrix[10][10];
float t;
int zeile, spalte;
// Eingabe Matrix
// ...

for(zeile=0; zeile<10; zeile++)
{
    for(spalte=zeile+1; spalte<10; spalte++)
    {
        // Tausch matrix[zeile][spalte] mit matrix[spalte][zeile]
        t = matrix[zeile][spalte];
        matrix[zeile][spalte] = matrix[spalte][zeile];
        matrix[spalte][zeile] = t;
    }
}
```