 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 1 von 7

Bei den bisherigen Programmen wurde für jede verwendete Variable (oder für jedes Objekt) ein eigener Typ und Name vergeben. Die Initialisierung, d.h. die Belegung mit einem konkreten Wert, erfolgte entweder bei der Deklaration oder durch Tastatureingabe.

Beispiel:

```
int i=1;           //Integerzahl i anlegen und mit 1 initialisieren
float wert;       //Floatzahl deklarieren
scanf(„%f“,&wert); //Wert über Tastatur eingeben
```

Stellen Sie sich vor, in einer Messwertreihe sollen die Tageswerte der Temperatur eines ganzen Monats gespeichert und verarbeitet werden. Bei der bisherigen Methode müssten dazu 31 verschiedene Namen vergeben und einzeln initialisiert werden. Die Berechnung der Durchschnittstemperatur (Summe aller Zahlen geteilt durch die Anzahl) würde eine endlos lange Formel ergeben. Für noch größere Datenmengen (z.B.: Adressdateien, Telefonverzeichnisse, allgemeine Datenbanken,...) wäre dieses Verfahren völlig unbrauchbar. Zudem ist die Anzahl der Werte nicht immer genau vorhersehbar. Die Lösung für dieses Problem bilden die Datenfelder (Arrays). Kennzeichen eines Datenfeldes ist, dass alle Werte vom gleichen Typ sind und den gleichen Bezeichner haben. Die Unterscheidung der einzelnen Werte erfolgt durch einen Index (ganze Zahl). Bei der Deklaration eines Datenfeldes werden der Typ, der Name und die Anzahl der Werte festgelegt.

#### Allgemeine Deklaration:

```
typ name[n];
```

**typ** ist der Datentyp eines einzelnen Wertes (z.B.: **int**, **float**, **char**,..., benutzerdefinierte Datentypen). **name** ist ein beliebig wählbarer Bezeichner, mit welchem auf die einzelnen Werte des Feldes zugegriffen werden kann. Das **n** in ‚[]‘-Klammern ist die Anzahl der Werte (Integer-Wert). Steht die Anzahl der Werte nicht genau fest, muss die maximale Größe des Datenfeldes abgeschätzt werden. Eine Variable für n ist nicht zulässig. Die Datenfeldgröße lässt sich zur Laufzeit des Programms nicht verändern!


Beispiel:

```
int zahl[5]; //deklariert ein Datenfeld mit 5 Zahlen vom Typ Integer
```

Im diesem Beispiel wird Platz für 5 Zahlen vom Typ Integer reserviert. Da ein Integerwert 4 Byte belegt, ist der gesamte reservierte Speicherbereich 5 mal 4 Byte = 20 Byte. Eine Deklaration der Form

```
double wert[100]
```

würde demnach 100 mal 8 Byte = 800 Byte reservieren.

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 2 von 7

### Initialisierung des Datenfeldes:

Die Inhalte des Datenfeldes sind nach der Deklaration undefiniert. Der Zugriff auf einzelne Werte erfolgt durch die Angabe des Namens mit nachfolgendem Index in []-Klammern.

Beispiel:

```
zahl[0] = 4;    // weist dem ersten Wert des Zahlenfeldes die Zahl 4 zu
zahl[1] = 6;    //..dem zweiten Wert die Zahl 6
zahl[2] = 8;
zahl[3] = 10;
zahl[4] = 12;   //..dem fünften Wert die Zahl 12
```

Beachten Sie, dass der Index immer mit dem Wert 0 beginnt. Der höchste Index ist um 1 kleiner als die Anzahl der reservierten Werte! Ein Wert **zahl[5]** existiert nicht!

### Merke:

**In Datenfeldern ist der erste Index immer die Null!**

Die Initialisierung kann auch direkt bei der Deklaration vorgenommen werden. Dazu wird eine Initialisierungsliste verwendet, welche die einzelnen Werte in {}-Klammern enthält und durch Kommata getrennt sind. Die Reihenfolge in der Initialisierungsliste entspricht der Ablage im Datenfeld.

Das folgende Beispiel initialisiert das Datenfeld genauso wie in obigem Beispiel:

```
int zahl[5] = {4,6,8,10,12};
```

Für die Ausgabe der Werte mit der **printf()**-Funktion wird ebenfalls der Name mit zugehörigem Index verwendet.

Die Anweisung


```
int i=0;
printf(„%2i.Zahl: %2i“,i+1,zahl[i]);
```

führt zu der Ausgabe:

```
1.Zahl: 4
```

Beachten Sie, dass die 1.Zahl (i+1) den Index 0 hat!

Will man alle Werte des Datenfeldes ausgeben, so bietet sich für die Ausgabe eine Schleife an, in welcher der Index fortlaufend von 0 bis 4(!) hoch gezählt wird.

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 3 von 7

Beispiel:

```
#include<stdio.h>
#include<conio.h>
main()
{
    int zahl[5]={4,6,8,10,12};
    int i;
    for (i=0;i<5;i++)
    {
        printf("\n%2i.Zahl: %2i",i+1,zahl[i]);
    }
    getch();
    return(0);
}
```


Das Programm liefert folgende Ausgabe:

```
1.Zahl:  4
2.Zahl:  6
3.Zahl:  8
4.Zahl: 10
5.Zahl: 12
```

Frage:

Zu welchem Ergebnis führt folgende Schleife, wenn man sie in das obige Programm nach der Ausgabe einfügt?

```
...
int s=0;
for (i=0;i<5;i++)
{
    s+=zahl[i];
}
printf("\n%i",s);
...
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 4 von 7

### Initialisierung über Tastatureingabe:

Die `scanf_s()`-Funktion benötigt wie immer neben der Formatangabe die **Adresse** des einzugebenden Objektes. Die Adresse eines Objektes erhält man indem der Adressoperator `&` vor den Bezeichner gestellt wird.

Durch die Anweisungen


```
int i=0;
printf("%2i.Zahl: ",i+1);
scanf_s(,"%i",&zahl[i]);
```

wird als erstes die Meldung `1.Zahl:` ausgegeben und erwartet dann als Eingabe den ersten Wert des Datenfeldes (mit dem Index 0!).

Will man alle Werte des Datenfeldes eingeben bietet sich wieder eine Schleife an, welche nach jeder Eingabe den Index inkrementiert.

Das folgende Beispiel gestattet die Eingabe von 5 Werten, gibt die Werte zur Kontrolle aus und berechnet anschließend den Mittelwert.

```
#include<stdio.h>
#include<conio.h>
main()
{
    int zahl[5];
    int i;
    float s=0;        //Der Mittelwert ist vom Typ float
    for (i=0;i<5;i++)
    {
        printf("\n%2i.Zahl: ",i+1);
        scanf_s("%i",&zahl[i]);
    }
    for (i=0;i<5;i++)
    {
        printf("\n%2i.Zahl: %2i",i+1,zahl[i]);
    }
    for (i=0;i<5;i++)
    {
        s+=zahl[i];
    }
    printf("\nMittelwert: %8.3f",s/5);
    _getch();
    return(0);
}
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 5 von 7

### 1. Übung:

Zu welchem Ergebnis führt das folgende Programm? Analysieren Sie das Programm bevor Sie es testen!

```
#include<stdio.h>
#include<conio.h>
main()
{
    int zahl1[3]={2,5,4};
    int zahl2[3]={3,2,2};
    int i=0,s=0;
    do
    {
        s+=zahl1[i]*zahl2[2-i];
        i++;
    }
    while(i<3);
    printf("\nErgebnis: %3i",s);
    getch();
    return(0);
}
```

### 2. Übung:

In den Rechner sollen 10 unterschiedliche Integer-Zahlen im Bereich von 0 bis 100 eingegeben und in einem Datenfeld abgelegt werden. Werden Werte außerhalb des Bereiches eingegeben oder ist ein Wert bereits vorhanden soll eine entsprechende Fehlermeldung „Wert außerhalb des Bereiches“ oder „Wert schon vorhanden“ ausgegeben werden und die Eingabe dieser Zahl wiederholt werden.

Anschließend sollen die Zahlen zur Kontrolle ausgegeben werden, sowie der Minimal- und der Maximalwert berechnet und ausgegeben werden.

### 3. Übung:

Entwickeln Sie ein Programm, welches sechs unterschiedliche Zufallszahlen im Bereich von 1 bis 49 erzeugt und ausgibt (Lottozahlengenerator).

Bei Betätigung einer beliebigen Taste sollen sechs neue Zahlen ausgegeben werden. Das Programm soll mit Betätigung der ‚ESC‘-Taste beendet werden.

Informieren Sie sich über die rand()-Funktion zur Erzeugung einer Zufallszahl!

Zusatzaufgabe:

Die Ausgabe der Lottozahlen soll aufsteigend sortiert erfolgen.

## Arrays und Zeiger

Der Bezeichner eines Arrays (ohne []-Klammern) ist die Adresse, auf welcher der erste Eintrag steht. Der Bezeichner ist also ein Zeiger auf den Beginn des Arrays. Entsprechend kann dann auf die einzelnen Elemente mit der Adresse zugegriffen werden.

### Beispiel:


Mit der Deklaration

```
int zahl[4]={1,2,3,4};
```

wird ein Datenfeld von 4 Zahlen (vom Typ int) angelegt. Jedes Element belegt vier Speicherplätze. Der Bezeichner 'zahl' ist die Adresse des ersten Eintrages identisch mit &zahl[0]. Die Adresse des nächsten Eintrages ist (zahl+1), usw. . Das Inkrementieren des Adresszeigers ist also physikalisch eine Erhöhung um vier Byte entsprechend der Größe des Datentyps.

Die folgende Tabelle verdeutlicht diesen Zusammenhang. Die (Start-)Adresse 2000 ist willkürlich vergeben. Sie wird vom Compiler erzeugt und kann nicht beeinflusst werden.

<i>Adress-Bezeichner</i>	<i>Adresse</i>	<i>Inhalt</i>	<i>Bezeichner</i>
zahl+3	2015	4	*(zahl+3)
	2014		
	2013		
	2012		
zahl+2	2011	3	*(zahl+2)
	2010		
	2009		
	2008		
zahl+1	2007	2	*(zahl+1)
	2006		
	2005		
	2004		
zahl+0	2003	1	*(zahl+0)
	2002		
	2001		
	2000		

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Datenfelder (Array)	Seite 7 von 7

### Programmbeispiel:

Das folgende Programm gestattet die Eingabe von 4 Zahlen. Anschließend werden die Adressen und die Inhalte der einzelnen Elemente angezeigt.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i=0;
    int zahl[4];
    for(i=0;i<4;i++)
    {
        printf("%i. Zahl: ",i+1);
        scanf_s("%d",zahl+i);
    }

    for(i=0;i<4;i++)
    {
        printf("Adresse: %u\tInhalt: %i\n",zahl+i,*(zahl+i));
    }
    _getch();
}
```