



Mehrdimensionale Datentypen



Deklaration und Initialisierung

*Objekte einer
Container-Klasse erzeugen*

Deklaration reserviert Speicherplatz für **alle** Elemente

```
// deklariert Array von 12 Ganzzahlen und reserviert damit 48 Byte:
int    TageProMonat [12];
      // Größe über eine Konstante festgelegt (grundsätzlich so machen !):
const int MonateProJahr = 12;
int    TageProMonat [MonateProJahr];
double MonatlicherUmsatz [MonateProJahr];
```

hier Compile-Time-Ausdruck möglich

```
// Initialisierung eines Arrays mit Konstanten:
int TageProMonat [MonateProJahr] = {31,28,31,30,31,30,31,31,30,31,30,31};
int TageProMonat [] = {31,28,31,30,31,30,31,31,30,31,30,31};
int AlleAufNull [Groesse] = {0}; // nicht initialisierte Elemente werden auf 0
                                // gesetzt
```

Trick

Operationen

§ Indizierung

n Elemente deklariert: Index 0 ... n-1

```
TageImMai = TageProMonat [4];
```

§ Bestimmung der Größe in Byte (nicht Elementen)

```
sizeof (TageProMonat)  ergibt 48
```

```
sizeof (int)           ergibt 4
```

§ Übergabe als Parameter an eine Funktion

∅ wird später im Zusammenhang mit Zeigern und Funktionen betrachtet

§ keine Zuweisung oder Parameterübergabe von ganzen Arrays !

		⋮	⋮	⋮	⋮	
abc		2	5	3	8	9
11		3	1			
10		3	0			
9		3	1			
8		3	0			
7		3	1			
6		3	1			
5		3	0			
4		3	1			
3		3	0			
2		3	1			
1		2	8			
TageProMonat	0	3	1			
xyz		8	8	8		

Mehrdimensionale Arrays

```
const int Quartale = 4;
const int Regionen = 3;
int Umsatz [Quartale][Regionen] = {
    {00, 01, 02}, {10, 11, 12},
    {20, 21, 22}, {30, 31, 32}
};
```

```
HessensUmsatz = Umsatz [3][1];
```

Ä ergibt 31

Umsatz [3,1] **ist falsch !**

		abc	25389
Umsatz	Quartale	2	32
		3 1	31
		0	30
		2	22
		2 1	21
		0	20
		2	12
		1 1	11
		0	10
	Regionen	2	02
		0 1	01
		0	00
	xyz	888	