

# C++ Teil 4

## 4.4. Felder

Möchte man mehrere Daten vom gleichen Typ in einer Variablen speichern, verwendet man Felder (engl. Arrays). Die Anzahl der Elemente, die in einem Feld stehen, werden nach dem Variablennamen in eckigen Klammern angegeben (eindimensionales Feld).

Neben eindimensionalen Feldern können auch mehrdimensionale Felder definiert werden. Die weiteren Dimensionen werden dann in zusätzlichen eckigen Klammern angegeben.

Zum besseren Verständnis: Eine Reihe mit ihren acht Quadraten auf einem Schachbrett kann mit einem eindimensionalen Feld verglichen werden; jedes Quadrat kann eine Figur aufnehmen. Mehrdimensional, nämlich zweidimensional ist das ganze Schachbrett; es besteht aus 8x8 Quadraten und kann somit 64 Figuren aufnehmen.

### Beispiele

```
short muenzen[100];    // eindimensionales Feld mit 100 Elementen

const int monat[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
// eindimensionales Feld mit 12 Elementen

SCHACH brett[8][8] =    {{a1,b1,c1,d1,e1,f1,g1,h1},
                        {a2,b2,c2,d2,e2,f2,g2,h2},
                        ...
                        {a7,b7,c7,d7,e7,f7,g7,h7},
                        {a8,b8,c8,d8,e8,f8,g8,h8}};
// zweidimensionales Feld mit 64 Elementen
```

Wie in den Beispielen zu sehen ist, initialisiert man Felder, indem nach der Definition die Initialisierungswerte innerhalb von geschweiften Klammern angegeben werden.

Auf ein bestimmtes Element innerhalb eines Feldes kann über den Index zurückgegriffen werden, der in eckige Klammern angegeben wird. Der Index beginnt in C++ immer mit 0.

### Beispielprogramm

```
// Demonstration eines mehrdimensionalen Felds (Arrays)
#include <iostream>
using namespace std;

int main()
{
    int meinFeld[3][2] = { {0,0} , {1,2} , {3,6} };
    // Definition und Initialisierung
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 2; j++)
```

```

        {
            cout << "meinFeld[" << i << "]"[" << j << "]: ";
            cout << meinFeld[i][j] << endl;
        }
    return 0;
}

```

## Aufgabe

Ein `string` ist eine Reihe von aufeinanderfolgenden Zeichen wie z.B. "Hello World". Mit folgender Definition kann man ein `char`-Feld erzeugen, das einen `string` enthält: `char text[12] = "Hello World";` Mit `cin.get(text, laenge);` kann man einen `string` von der Tastatur einlesen, wobei `laenge` für die Länge des `strings` steht. Die Eingabe wird normalerweise mit `enter` abgeschlossen. Achtung: C++ fängt bei 0 an zu zählen und als letztes Zeichen fügt der Compiler das `NUL`-Zeichen (`'\0'`) hinzu, was das Ende des `strings` kennzeichnet.

Schreibe ein Programm, das ein `char`-Feld verschlüsselt, indem es jeden Buchstaben von A-M um 13 erhöht bzw. von N-Z um 13 erniedrigt. Unterschiede Klein- und Großbuchstaben. Sowohl der unverschlüsselte als auch der verschlüsselte Text sollen auf dem Bildschirm ausgegeben werden.

Um festzustellen, ob ein Zeichen ein Buchstabe, ein Klein- oder Großbuchstabe ist, können die Funktionen `isalpha(c)`, `islower(c)`, `isupper(c)` aus `<cctype>` verwendet werden.