

Wir werden nun eine sehr lehrreiche C++-Aufgabe bearbeiten. Zur Lösung der Aufgabe müssen Sie einen Konstruktor, einen Destruktor, einen Kopier-Konstruktor, konstante Elementfunktionen, Zeiger, Referenzen, Templates und überladene Operatoren einsetzen.

Schreiben Sie eine Klasse **Matrix** für die Erzeugung von $(n \times n)$ -Matrizen. Schreiben Sie einen Operator $*$ für die Multiplikation von zwei $(n \times n)$ -Matrizen. Überladen Sie den Ausgabeoperator $<<$ für die formatierte Ausgabe einer Matrix. Über den Konstruktor der Klasse **Matrix** kann die Dimension n übergeben werden. Für die dynamische Verwaltung eines $(n \times n)$ -Arrays verwenden Sie einen Zeiger ****m**, der auf ein Array von n Zeigern verweist. Jeder dieser Zeiger verweist auf eine Zeile der Matrix. Eine Zeile besteht wieder aus einem Array der Länge n . Alle $n + 1$ Arrays für eine Matrix werden dynamisch mit **new** erzeugt. Das Array aus Zeigern ist vom Typ **T*[dim]**. Sie können dann komfortabel mit der Indexschreibweise (**m[i][j]**) auf die einzelnen Einträge des 2-dimensionalen Arrays zugreifen. Für das Testprogramm

```
Matrix<int> m(2);

m.set(0,0,1);
m.set(0,1,2);
m.set(1,0,3);
m.set(1,1,4);

std::cout << m << "*" << std::endl << m << "=" << std::endl << m*m;
```

soll die Ausgabe wie folgt aussehen:

```
  1   2
  3   4
*
  1   2
  3   4
=
  7  10
 15  22
```

Natürlich soll das Programm auch für andere Dimensionen und Matrizen korrekt funktionieren.