

Schablonen Klasse `vector<T>`

Der Container *Vektor* (*vector*)

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> cont;

// Elemente 1 bis 6 jeweils
// hinten anfüegen */
for (int i=1; i<7; ++i) {
    cont.push_back(i);
}

for (int j=0;j<cont.size();++j) {
    cout << cont [j] << ' ';
}
cout << endl;
```

Die Elemente werden in einem dynamischen Array verwaltet, d.h. wahlfreier Zugriff ist über einen Index-Operator ([]) möglich.

Das Einfügen und Löschen von Elementen am Ende geht optimal schnell.

Einfügen und Löschen am Anfang und in der Mitte ist dagegen langsam.

Der Container Vektor (2)

```
vector<int> cont;           // Es wird ein leerer Container von Typ vector für  
                           // int-Elemente angelegt.  
  
cont.push_back(i);       // push_back fügt das Argument am Ende des  
                           // Containers ein.  
  
int anzahl = cont.size(); // Die Methode size() gibt die Anzahl der  
                           // Elemente im Container zurück.  
  
cont [i];                // Der Index-Operator [] wird zum wahlfreien  
                           // Zugriff (lesend und schreibend) auf  
                           // das Element an der Position i verwendet.
```

Der Container Vektor (3)

at (i);

*// wie Index-Operator [], allerdings erfolgt
// Prüfung, Wert > 0 und kleiner size ist.
// Wenn nicht, wird Exception geworfen.*

resize(anz);

*// Array wird auf anz Element vergrößert bzw.
// verkleinert.*

Code mit ganz „normalem“ C-Vektor

```
#include <iostream>
#include <vector>
using namespace std;

int cont[10];

// Elemente 1 bis 6 jeweils
// hinten anfüegen */
for (int i=1; i<7; ++i) {
    cont[i-1] = i;
}

for (int j=0; j < 6; ++j) {
    cout << cont [j] << ' ';
}
cout << endl;
```

Die Arraygröße N wird am Anfang unveränderlich festgelegt.

Wahlfreier Zugriff zwischen 0 und N-1 ist möglich.

Mit „normaler“ C-Vektor bzw. Schablonenklasse vector

```
#include <iostream>

int cont[10];

// Elemente 1 bis 6 jeweils
// hinten anfüegen */
for (int i=1; i<7; ++i) {
    cont[i-1] = i;
}

for (int j=0; j < 6; ++j) {
    cout << cont [j] << ' ';
}
cout << endl;
```

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> cont(10);

// Elemente 1 bis 6 jeweils
// hinten anfüegen */
for (int i=1; i<7; ++i) {
    cont[i-1] = i;
}

for (int j=0; j < 6; ++j) {
    cout << cont [j] << ' ';
}
cout << endl;
```

Der Container array (seit C++ 11 im Standard)

```
#include <stdexcept>
#include <iostream>
#include <array>
using namespace std;

int main() {
    array<int, 7> a;

    try {
        a.at(8) = 44;
    }
    catch (const out_of_range& e) {
        cerr << "out of range: " << e.what() << endl;
    }
    //a[8] = 22; // Speicherschutzverletzung
}
```