

## Methoden und Operatoren der Containerklassen der STL

Diese Liste ist unvollständig und enthält nur die wichtigsten Methoden. Im Folgenden sei der Name des Containers **behaelter**. Die folgenden Befehle gelten für die Containertypen **vector**, **list** und **deque**.

*Zugriff auf das i'te Element des Containers (i ist natürliche Zahl).*

- **behaelter[i]**  
Zugriff auf das i'te Element des Containers. Sollte das i'te Element nicht angelegt worden sein liefert das Programm *keinen* Fehler! (Operator nicht für Listen!)
- **behaelter.at(i)**  
Zugriff auf das i'te Element des Containers. Sollte das i'te Element nicht angelegt worden sein, bricht das Programm ab. Wegen der zusätzlichen Prüfung, ob das Element existiert, ist **behaelter.at(i)** langsamer als **behaelter[i]**. (Methode nicht für Listen!)
- **behaelter.front()**  
Zugriff auf das erste Element des Containers.
- **behaelter.back()**  
Zugriff auf das letzte Element des Containers.

*Methoden zum Anhängen und Entfernen von 'Randelementen'.*

- **behaelter.push\_back( el )**  
Ein neues Element **el** wird ans Ende des Containers gesetzt.
- **behaelter.pop\_back()**  
Das letzte Element des Containers wird entfernt.
- **behaelter.push\_front( el )**  
Ein neues Element **el** wird an den Anfang des Containers gesetzt. Nicht für Vektoren!
- **behaelter.pop\_front()**  
Das erste Element des Containers wird entfernt. Nicht für Vektoren!

*Allgemeine Methoden.*

- **behaelter.clear()**  
Löscht alle Elemente des Containers.
- **behaelter.size()**  
Liefert die Anzahl der Elemente des Containers.
- **behaelter.max\_size()**  
Liefert das maximale Fassungsvermögen des Containers.

*Methoden zum Einfügen und Löschen von Elementen*  
(ZUGRIFF NUR ÜBER ITERATOREN!).

- `behaelter.insert( it , el )`  
Ein neues Element `el` wird an die Position im Containers gesetzt, auf die der Iterator zeigt. Alle weiteren Elemente verschieben sich um eins nach hinten.
- `behaelter.erase( it )`  
Das Element des Containers, auf das der Iterator `it` zeigt wird gelöscht.
- `behaelter.erase( it1 , it2 )`  
Entfernt alle Elemente des Containers, die sich im Iterator-Bereich `[it1,it2-1]` befinden (`it2` selbst wird also nicht entfernt).  
Insbesondere tut `behaelter.erase(it1,it1)`; gar nichts und es gilt '`behaelter.erase(it1,it1+1) = behaelter.erase(it1)`'.