

STL §11

Standard Template Library: Bsp. aus Listing 11.1 des Skriptes.

- Container
 - `map` mit `[key]` und `.at(key)` und `.count(key)`
 - `multimap` mit `.count(key)`
- Algorithmen
 - `copy` : Speicher für Output-Container preallokieren!
Iterator als Returnparameter \implies zur Überprüfung
 - `copy_if` : Speicher für Output-Container preallokieren!
unäre Boolean-Fkt. wird benötigt.
Iterator als Returnparameter \implies zur Überprüfung und zur Verkürzung des Zielcontainers via der Methode `.erase()`
- Iteratoren
- Container und Algorithmen wissen nichts voneinander \implies Iteratoren als Bindeglied.

Analog zu §11.2 des Skriptes

- Algorithmen und welche Funktionen von den Containererelementen bereitgestellt werden müssen.
- Algorithmen mit λ -Funktion als binäre/unäre Funktion/Operator.
- unäre Boolean-Fkt: `bool bool_op(T const &a)`
- binäre Boolean-Fkt: `bool bool_op(T const &a, T const &b)`
- unäre arithm. Operation: `T arith_op(T const &a)`
- unäre arithm. Operation: `S arith_op(S const &a, T const &b)`

Example *v_9a/* analog zu §11.3 demonstrieren.

- zuerst mit `vector<double>`
- dann mit `vector<Komplex>`
- und `list<Komplex>`

Anwendungen:

- Beispiel *sort_index/*
- *graph_1/* : Klasse einfacher Graph mit durchgehender Nummerierung
- *graph_2/* : Klasse Graph ohne durchgehende Nummerierung (`set` für Vertexindizes)
- *graph_3/* : Wie 3, aber eigene Datenstruktur `Edge` statt `array<int,2>`
Mehr Methoden: Delete, Append, max_vertex, ...

Literatur

- [Haase22] Gundolf Haase: Einführung in die Programmierung mit C++ (2022), *www*¹.
- [Stroustrup10] Bjarne Stroustrup: Einführung in die Programmierung mit C++. Pearson Studium, München (2010).

¹http://imsc.uni-graz.at/haasegu/Lectures/Kurs-C/Script/html/script_programmieren.pdf