

Clicker

Bitte den Link

<https://vc2.sonia.de/b/har-zq1-o0p-dhs>

für WS 2024/25 nutzen.

Die verschiedenen Programmierparadigmen von C++

Vorles	Termin	Vorlesung	Übungen und Feedback
Woche		Freitag ; Block 1+2	
10	11 06. Dez Fr	lineare und assoziative Container; STL Algorithmen	Unterstützung/Vorabnahme der Übungen; z.B 11.12 für Abgabe am 18.12 (Zusatztermin in diesem Zeitraum)
11	12 13. Dez Fr	Algorithmus oder Methode Klasse unique_ptr,, shared_ptr, Lambda-Ausdrücke	finale Abgabe der Übungen; Di 17.12
12	13 20. Dez Fr	Verschiebeoperatoren , Vorbereitung Klausur	

Klausur: Do, 16.1.25, 14-15:30 Uhr; Raum 152

Klausureinsicht: Fr. 24.1.2025 10:00- Uhr

Rückblick, 13.12.2024

- Fortsetzung des Komponenten/Lieferanten Beispiels: Algorithmus versus Methode
- Intelligente Zeiger (`unique_ptr`)
- Verschiebeoperatoren angedeutet (`operator=(X&& org);`)
- Ihr Stand bei der Rufzeichen-Extraktion (**eher übersichtlich**)
- Intelligente Zeiger (`shared_ptr`)

Clicker: Raten Sie mal

```
list<int> cont;
for (int i=1; i<6; ++i) { cont.push_back(i); }
    // Ausgabe wäre nun 1 2 3 4 5
//copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));
// 3 mit remove entfernen
remove(cont.begin(), cont.end(), 3);
copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));
```

- Ausgabe?
1. 1 2 4 5 5
 2. 1 2 4 5
 3. 1 2 3 4 5
 4. 5 4 2 1

Clicker

```
list<int> cont;  
for (int i=1; i< 6; ++i) {  
    cont.push_back( i );  
}
```

2. Ausgabe?
- 1. 1 2 3 4 5
 - 2.
 - 3. 1 2 4 5
 - 4. 1 2 4 5 3

```
// Ausgabe ist 1 2 3 4 5  
copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));  
remove(cont.begin(), cont.end(), 3 );  
cont.erase(cont.begin(), cont.end());  
copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));
```

Clicker

```
list<int> cont;  
for (int i=1; i< 6; ++i) {  
    cont.push_back( i );  
}
```

2. Ausgabe?
- 1. 1 2 3 4 5
 - 2. 1 2 4 5 5
 - 3. 1 2 4 5
 - 4. 1 2 4 5 3

```
// Ausgabe ist 1 2 3 4 5  
copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));  
cont.erase(remove(cont.begin(), cont.end(), 3), cont.end() );  
copy(cont.begin(), cont.end(), ostream_iterator<int>(cout, " "));
```