

C++17 STL Usage Recipes (Seminar Fortg. Programmierkonzepte) SS 18

Prof.Dr. Hans-Jürgen Buhl

Praktische Informatik, Fachgruppe Mathematik und Informatik
Fakultät für Mathematik und Naturwissenschaften
Bergische Universität Wuppertal



12. März 2018

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

1 Überblick

2 Themenschwerpunkte

3 Organisatorisches

- Zum Seminar: Termin, Zielgruppe, ...
- Terminplanung
- zu erwerbende Nachweise
- Voraussetzungen für die erfolgreiche Teilnahme

4 Themenvorschläge

Überblick

Kochrezepte für in der Programmentwicklung immer wieder auftretende Problemlagen sind das A und O guten Programmierstils. Das Buch C++17 STL Cookbook [1] von J. Galowicz führt mittels vieler kurzer Fallbeispiele in die professionelle Benutzung der modernen C++-Standard-Bibliothek (C++17) ein.

Sein Inhalt (C++17-Neuerungen, bewährte Praktiken bei der Benutzung der STL-Container, Iteratoren und Ranges, Lambda-Ausdrücke, STL-Algorithmen, ...) ist Gegenstand der Vorträge dieses Seminars.

In den Seminarvorträgen sollen die Vortragenden die Rezepte des Buchs vorstellen und auch anhand eigener Programmbeispiele illustrieren. Dazu sind während der Vortragsvorbereitung von den Teilnehmenden u.a. auch eigene Programme am Rechner zu erstellen.

- 1 Überblick
- 2 Themenschwerpunkte**
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

Themenschwerpunkte

Standard C++17 [3]:

- structured bindings (11.5)
- if/switch init-statement (9.4.1/9.4.2)
- braced init lists (11.6)
- `common_type_t`, parameter packs, helper functions (23.15)
- `if constexpr ()` (9.4.1)
- inline variables (10.1.6)
- fold expressions (8.1.6), ...

- Containers (26)
- Iterators/Ranges (27)
- Lambda Ausdrücke (8.1.5)
- Algorithmen (Grundlegendes) (28)
- Algorithmen (Fortgeschrittenes) (28)
- Strings/Streams (24/30)
- Utilities (23)

- 1 Überblick
- 2 Themenschwerpunkte
- 3 **Organisatorisches**
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

Organisatorisches: Zum Seminar

- **Termin:** Mo 12-14, Raum D.13.15
- **Zielgruppe:** Bachelor-Studierende
- **Voraussetzungen:** Kenntnisse der Objektorientierten Programmierung, modernes C++ 11/14/17 (OOP-Vorlesung)
- **Teilnahme:** Verbindliche Anmeldung bei Vorbesprechung
- **Zu erbringende Leistung:** Jede(r) Teilnehmer(in) arbeitet sich in das gewählte Thema ein, hält einen (Beamer-)Seminarvortrag mit Illustrationen durch Programmbeispiele zum gewählten Themenbereich.
- **Workload:** 90 Stunden! (laut Modulbeschreibungen)

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - **Terminplanung**
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

Organisatorisches: Terminplanung

- **Vorbesprechung mit Themenvergabe:**
Montag, den 16. April 2018, 12 Uhr c.t. in Raum D.13.15
- danach Zeitraum zur Einarbeitung ins Thema, zur Literaturrecherche, zur Vorbereitung des Vortrags und Rücksprache mit dem Betreuer, für Probevorträge im Bekanntenkreis, für Korrekturlesen, ...
- **mögliche Vortragstermine:**
28. Mai, 4. Juni, 11. Juni, 18. Juni, 25. Juni,
2. Juli, 9. Juli, 16. Juli

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - **zu erwerbende Nachweise**
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

Je nach Studiengang bzw. Fachrichtung können folgende **Nachweise** erworben werden:

- **Bachelor-IT (PO 2006/2009):**
BIT 15 (Pflichtmodul Ergänzende Wissenschaften, 1 SWS, 3 LP)
- **Bachelor-IT (PO 2011):**
FBE0071 Erg. Wissenschaften, Techniken des wiss. Arbeitens, 2 SWS, 3 LP)
- **Bachelor Applied Science (PO 2007):**
Seminar zur Informatik (2 SWS, 3 LP)
- **Komb. Bachelor of Arts, Informatik:**
Seminar zur Informatik (2 SWS, 3 LP)
- **Alle Studiengänge (freiwillige Teilnahme):**
allg. Leistungsnachweis „Seminar“ (Schein)

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - **Voraussetzungen für die erfolgreiche Teilnahme**
- 4 Themenvorschläge

Organisatorisches: erfolgreiche Teilnahme

- Auswahl Thema, Einarbeitung und Beschäftigung mit diesem Thema
- Konkret: Selbständige Literaturrecherche, didaktische Aufbereitung des Themas, Implementierung eigener und vorgegebener Beispiele
- Ausarbeitung Vortrag (max. 40 Min. + 5 Min. Diskussion)
- Schriftliche Ausarbeitung des Themas (6-8 Seiten) (mit Titelseite und vollständigen Quellenangaben, eventuell Index)
- Mindestens ein Termin beim Betreuer vor dem Vortrag (Deadline: Gliederung+Aufbau 2 Wochen vorher; Vortragsfolien oder -Präsentation 1 Woche vorher)

Organisatorisches: erfolgreiche Teilnahme (Forts.)

- Bereitstellung von Ausarbeitung und Beispielprogrammen auf Web-Server der Fachgruppe Mathematik
- Rechtzeitig vor Beginn des Vortrags erscheinen (mind. 15 Min vorher), rechtzeitiger Test/Kontrolle der techn. Ausstattung (Rechner, Beamer, Laserpointer, Tafel, Kreide, ...)
- Anwesenheitspflicht bei allen Vorträgen
- Aktive Mitarbeit wird erwartet (Fragen, Diskussion, ...)

(siehe auch: [Hinweise für eine erfolgreiche Seminarteilnahme](#))

- 1 Überblick
- 2 Themenschwerpunkte
- 3 Organisatorisches
 - Zum Seminar: Termin, Zielgruppe, ...
 - Terminplanung
 - zu erwerbende Nachweise
 - Voraussetzungen für die erfolgreiche Teilnahme
- 4 Themenvorschläge

Themenvorschläge

- 1 The New C++17 Features: vgl. Seite 7
- 2 STL Containers: `remove()/erase()`, `try_emplace()`, ...
- 3 Iterators: `iterator_category`, `back_insert_iterator`, ...
- 4 Lambda Expressions: `auto counter([count = 0]() mutable { return ++count; });, ...`
- 5 STL Algorithm Basics: `partition()`, `partial_sort()`, ...
- 6 Advanced Use of STL Algorithms: `split()`, `gather_sort()`, ...
- 7 Strings/Stream Classes/Regular Expressions: `char_traits`, ...
- 8 Utility Classes: `std::any`, `std::variant`, `std::optional`, ...

- [1] Jacek Galowicz. C++17 STL Cookbook: Over 90 recipes that leverage the powerful features of the standard library in C++17. Packt Publishing Ltd., 2017
- [2] <http://www.digibib.net/permalink/468/EDS/edsebk:1543832>
- [3] <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/n4659.pdf> (21.03.2017)
- [4] <https://en.wikipedia.org/wiki/C++17> (12. März 2018)

Index

algorithm, 8
braced init list, 7
C++17, 21
common_type_t, 7
container, 8
fold expression, 7
if constexpr(), 7
if init-statement, 7
inline variables, 7
iterator, 8
Kochrezepte, 4

lambda, 8
modern C++, 4
parameter pack, 7
Programmierstil, 4
range, 8
structured bindings, 7
standard library, 4
stream, 8
string, 8
switch init-statement, 7
utility, 8