

Materialpaket 00_README – Überblick

C/C++, Autor: Prof. Dr.-Ing. Carsten Link

Version 1.3.1 March 3, 2019

Contents

1	Kompetenzen und Lernergebnisse	1
2	Konzepte	2
2.1	Didaktisches Konzept	2
2.2	Materialpakete	3
2.3	Programmierungsumgebung und Werkzeuge	4
2.4	Begleitprogramme	5
3	Material zum aktiven Lernen	5
3.1	Aufgabe: Grundgerüst	5
3.2	Aufgabe: Modifikationen	5
3.3	Verständnisfragen	6
4	Nützliche Links	6
5	Literatur	6

1 Kompetenzen und Lernergebnisse

Durch das Bearbeiten dieses Materialpaketes erwerben Sie diese Kompetenzen (Wissen, Fähigkeiten, Fertigkeiten zur Problemlösung):

Sie sind in der Lage, die Struktur der nachfolgenden Materialpakete zu erfassen und letztere zu bearbeiten.

Die oben genannten Kompetenzen erwerben Sie, indem Sie Lernziele erreichen, welche sich prüfen lassen. Lernergebnisse: Sie können nachweislich¹:

- das zugrundeliegende didaktische Konzept erläutern
- die verwendete Programmierungsumgebung benennen

¹Sie können das Erzielen der einzelnen Lernergebnisse beispielsweise bei einem Testat im Praktikum oder einer Aufgabe in der Modulprüfung nachweisen

- die verwendeten Werkzeuge benennen
- die Begleitprogramme installieren
- erläutern, wie Materialpakete aufgebaut sind und welchen Zweck die einzelnen Kapitel haben
- erläutern, wie Materialpakete im Semester verwendet werden, um Aufgaben zu lösen
- erläutern, wie Materialpakete vor der Prüfung verwendet werden, um sich vorzubereiten

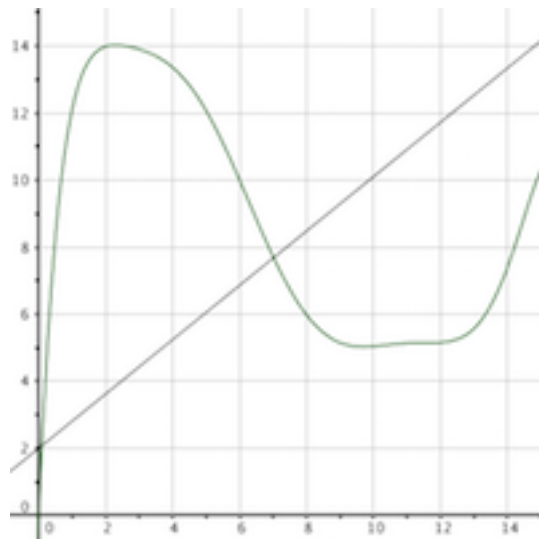
2 Konzepte

In den nachfolgenden Unterkapiteln wird das dem Kurs zugrundeliegende didaktische Konzept erläutert. Weiterhin werden die einzelnen Teile des Kurses kurz vorgestellt.

2.1 Didaktisches Konzept

Die Grundsätze des didaktischen Konzeptes sind:

- *Begleitetes Selbststudium*
 - Lernen während des Semesters - nicht erst vor der Prüfung
 - allein oder in Gruppen bearbeiten
 - Feedback (Fragen/Erkenntnisgewinn) abgeben
- Geringer Komplexität durch isolierte Pakete: Die verschiedenen Sprachmittel bzw. Programmierstile werden einzeln/isoliert vorgestellt
- Geringer Umfang durch Stoffreduktion
 - solider Ausgangspunkt für zukünftige C++-Programmierer
 - wesentlich für die Informatiker, die in Zukunft nicht in C++ programmieren werden
- Kompetenzerwerb mit ständigen Lernkontrollen (so genannte *Modifikationen*)



Der studentische Aufwand für dieses Modul (Polynom höheren Grades) im Vergleich zu der Höhe/Menge der erworbenen Kompetenzen (gerade Linie).

Wie in der Abbildung zu sehen ist, ist der Einstieg in dieses Modul womöglich steil. Das steigende Wissen und die erlangten Fähigkeiten lassen den Aufwand jedoch beträchtlich sinken, so dass der anfängliche Eindruck schnell verschwindet. Zu Prüfungsvorbereitung sollte der Aufwand nicht wesentlich steigen – schließlich wird semesterbegleitend gelernt.

2.2 Materialpakete

Das Modul ist in verschiedene Kapitel untergliedert, die Materialpakete genannt werden. Diese sind in sich geschlossen und haben ausschließlich vorangegangene Materialpakete als Voraussetzung. Die Struktur aller Materialpakete ist gleich (siehe Inhaltsverzeichnis dieses Materialpakets):

- Kompetenzen:
- Konzepte:
- Material zum aktiven Lernen
 - Aufgabe Grundgerüst: Programmieraufgabe, welche im Vorfeld allein oder in Gruppen bearbeitet wird
 - Aufgabe Modifikationen: Kleine Änderungen am Grundgerüst, mittels derer Studierende den Kompetenzerwerb nachweisen können (Testat)
 - Verständnisfragen: Fragen, die Studierende nach Erwerb der Kompetenzen beantworten können
- Nützliche Links
- Literatur

Die Materialpakete im einzelnen:

- **00_README:** dieses Materialpaket; Überblick über den Kurs
- **01_ENV:** C/C++-Toolchain, Kommandozeile, Übersetzungsvorgang (mit Zwischenschritten)
- **02_MENT:** Darstellung von Zahlen und Zeichenketten als Bitmuster im Speicher
- **03_FLOW:** a) grundlegender Kontrollfluss, b) Interrupt Handling, c) rekursive Berechnungen
- **04_UDEF:** Benutzerdefinierte Datentypen (Werttypen), operator overloading
- **05_OO:** Objektorientierte Programmierung: Vererbung, Konstruktionsreihenfolge, Beziehungen, Wert- und Heapobjekte, Speicher auf embedded-Systemen
- **06_POLY:** Subtyping Polymorphism
- **07_STD:** generischer Polymorphismus; Algorithmen und Datenstrukturen der C++-Standardbibliothek
- **08_PTRN:** C++-Idiome und Design Patterns
- **09_DSGN:** Programmierstil und Design komplexerer Projekte
- **10_PITF:** Stolperfallen
- **11_PUTT** Putting it all together; abschließendes Zusammenbringen aller Themen mit weiteren Übungsaufgaben zur Prüfungsvorbereitung

Dem Dozenten steht es frei, einige Materialpakete entfallen zu lassen.

Die folgenden großen C++-Themen werden von diesem Kurs nicht abgedeckt:

- Exceptions
- `const`
- Stream I/O
- Template (Meta-) Programmierung
- Lambdas

2.3 Programmierungsumgebung und Werkzeuge

Es steht ein Image² bereit, welches ein Linux mit allen verwendeten Werkzeugen enthält. Die Wichtigsten sind:

- `bash`: Text-basierte Benutzerschnittstelle, hilft beim Aufruf des Compilers
- `pcc`: Ein sehr einfacher C-Compiler, der übersichtlichen Assembler-Code erzeugt
- `clang/clang++`: C- bzw. C++-Compiler. Clang ist sehr modern, liefert gute Fehlermeldungen und ist aktuell, was die Unterstützung neuerer C++-Standards angeht (z. T. wird C++11 oder C++14 verwendet)
- `diff`, `Diffuse`: visualisiert Änderungen zwischen Dateien
- `CodeLite`: eine einfache IDE
- `pandoc`: wurde verwendet, um die Materialpakete zu erstellen

²<http://www.technik-empden.de/~clink/FedoraLXDE-2016-01-05.7z>

2.4 Begleitprogramme

- virtual LCD³
- CppRobots⁴



Figure 1: Virtual LCD

3 Material zum aktiven Lernen

3.1 Aufgabe: Grundgerüst

– entfällt –

3.2 Aufgabe: Modifikationen

– entfällt –

³http://www.technik-empden.de/~clink/projects/2016w-ProjGrp/05_Website_ohne_Quellcodes/11_Dokumente/13_Doxygen/doc/html/index.html

⁴<https://github.com/braak/CppRobots/wiki>

3.3 Verständnisfragen

1. Warum ist das Script des Moduls in verschiedene Materialpakete untergliedert?
2. Welche Voraussetzungen hat ein Materialpaket?
3. Reichen die Kompetenzen, die erlangt werden (wenn alle Materialpakete durchgearbeitet wurden) aus, um ein guter C++-Programmierer zu sein? Begründung!

4 Nützliche Links

- Stack Overflow: <http://stackoverflow.com>
- C++ reference: <http://en.cppreference.com/w/>
- C++ Referenz: <http://de.cppreference.com/w/>

5 Literatur

- [PPP] Stroustrup, Bjarne: Programming - Principles and Practice using C++
- [TCPL] Stroustrup, Bjarne: The C++ Programming Language, Fourth Edition
- [CUEB] U. Kirch, P. Prinz: C++ das Übungsbuch, Testfragen und Aufgaben mit Lösungen