C/C++ Materialpaket (Level AB)
00_README – Überblick

Prof. Dr. Carsten Link

Zusammenfassung

Abbildung 1: Ein Wegweiser durch die Materialpakete

Inhaltsverzeichnis

1 Kompetenzen und Lernegebnisse **2**

2 Konzepte **2**
2.1 Didaktisches Konzept 2
2.2 Materialpakete 3
2.3 Programmierumgebung und Werkzeuge 5

3 Material zum aktiven Lernen **5**
3.1 Aufgabe: Grundgerüst 5
3.2 Aufgabe: Modifikationen 5
3.3 Verständnisfragen 5

4 Nützliche Links **5**

5 Literatur **6**

1 Kompetenzen und Lernegebnisse

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. Lernegebnisse: Sie können nachweislich[1]:

- das zugrundeliegende didaktische Konzept erläutern
- 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*)
- Verschieden Kompetenzstufen: Die Level A (Basic), B (Intermediate) und C (Proficient) orientieren sich am Common European Framework of Reference

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

for Languages[2]. Hierdurch können Studierende Inhalte ausschließen und sich auf die von Ihnen angestrebte Stufe konzentrieren. Falls Sie in der Prüfung ein *sehr gut* anstreben, müssen Sie auch Aufgaben der Stufe C beherrschen und im Praktikum bearbeiten

Wie in Abbildung 2 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.

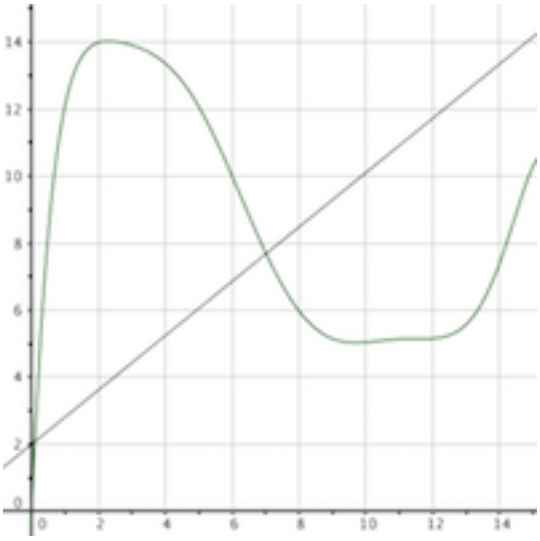

Abbildung 2: Andeutung des studentischen Aufwandes für dieses Modul (Polynom höheren Grades) im Vergleich zu der Höhe/Menge der erworbenen Kompetenzen (gerade Linie)

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: Wissen, Fähigkeiten und Fertigkeiten zur Problemlösung
- Konzepte: Erläuterung des jeweiligen Inhalts
- Material zum aktiven Lernen
  - hierzu finden Sie **Quellcode-Vorlagen** in `src-cpp-student/`

[2]https://en.wikipedia.org/wiki/Common_European_Framework_of_Reference_for_Languages

  - 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_TOOL:** C/C++-Toolchain, Kommandozeile, Übersetzungsvorgang (mit Zwischenschritten)
- **02_FLOW:** a) grundlegender Kontrollfluss, b) Interrupt Handling, c) rekursive Berechnungen, d) weiterführende Kontrollstrukturen
- **03_DATA:** Darstellung von Zahlen und Zeichenketten als Bitmuster im Speicher
- **04_UDEF:** Benutzerdefinierte Datentypen (Werttypen), operator overloading
- **05_OO:** Objektorientierte Programmierung: Vererbung, Konstruktionsreihenfolge, Beziehungen, Wert- und Entitätsobjekte, Speicher auf embedded-Systemen
- **06_BIND:** Subtyping Polymorphism, frühe und späte Bindung von Funktionsaufrufen
- **07_STD:** generischer Polymorphismus; Algorithmen und Datenstrukturen der C++-Standardbibliothek
- **08_PTRN:** C++-Idiome und Design Patterns
- **09_PRACT:** Best and worst practices, Guidelines, 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 Materialpakete werden in zwei Bündeln ausgeliefert: einmal für die Kompetenzstufen A und B; einmal für die Kompetenzstufe C. Nicht jedes Bündel enthält alle Materialpakete.

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

- IDEs (Integrated Development Envoironments, z. B. VisualStudio, XCode, etc.)
- Streams
- I/O
- Exceptions

- `const` correctness
- Character encodings und Unicode
- Template (Meta-) Programmierung

2.3 Programmierumgebung und Werkzeuge

Es steht ein Image 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 bis C++17 verwendet)
- meld, diff, Diffuse: visualisiert Änderungen zwischen Dateien
- pandoc: wurde verwendet, um die Materialpakete zu erstellen

3 Material zum aktiven Lernen

3.1 Aufgabe: Grundgerüst

– entfällt –

3.2 Aufgabe: Modifikationen

– entfällt –

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. Freiwillig: Zeigen Sie auf, inwiefern das Piktogram auf der Titelseite dieses Materialpaketes den Inhalt zusammengefasst darstellt.

4 Nützliche Links

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

5 Literatur

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