

1 Zielsetzung, Einordnung, Literatur



Zielsetzung

- Konstrukte heutiger Programmiersprachen für
Ablaufkontrolle (Konstrollstrukturen)
Datenstrukturierung (Datentypkonstruktoren)
Programmgrobstrukturierung
- Syntaxnotation
- Systematisches Programmieren im Kleinen
Problem ▶ Algorithmus ▶ Programm
▶ Betrachtung (Test, Effizienz)
- Standardkenntnisse: Suchen und Sortieren
- Modularisierung:
Übergang zum Programmieren im Großen:
Bausteine, Zusammensetzung, Wiederverwendung
- Datenabstraktion und Formulierung durch Module
- Standardkenntnisse Datenstrukturen:
Listen, Bäume, Graphen: Wie realisiert man sie?
Wie verwendet man sie?
- Grundzüge der Objektorientierung
- Nutzung der weitverbreiteten Programmiersprache C++
- Ausblick Softwaretechnik

dabei: Arbeiten mit Arbeitsplatzrechnern (CIP-Pool), Arbeiten mit mod. Werkzeugen



Veranstaltung legt Grundlage für

- Programmierpraktikum (anschließend)
- vergl. Darstellung höherer Programmiersprachen
- Effiziente Algorithmen: genauere Effizienzbetrachtung
- theor. Konzepte für Programmierung/Programmiersprachen
- andersartige Programmiersprachen
(Lisp, Smalltalk, Prolog)
- Softwaretechnik
- maschinennahe Programmierung
- nebenläufige Programmierung
- Compilerbau
- Datenbanksysteme
- Betriebssysteme

und beliebige Anwendungsprogrammierung.

Wichtig in erster Linie für Vertiefung in Technischer Informatik



Warum C++?

C++

- Objektorientierung und andere Konstruktionsprinzipien
- weite Verbreitung für technische Anwendungen
- C hat Features für systemnahe Programmierung (z.B. Adreßrechnung)
- Auf vielen Plattformen verfügbar

C++ und Methodik

- auch in C++ läßt sich sauber programmieren bzw. ein Programm strukturieren
- Erklärung ist nicht C++, sondern methodische Verwendung (z.B. Syntax, z.B. Sprachgebrauch):
Einschränkung, Disziplin
- Transliterationstabelle?



Literatur

1. Standardliteratur zur Sprache C++

Bjarne Stroustrup: Die C++ Programmiersprache, Addison–Wesley, 1998

Margaret A. Ellis / Bjarne Stroustrup: The annotated C++ reference manual, Addison–Wesley, 1995

Stanley B. Lippman: C++ primer, Addison–Wesley, 1991

Martin Schader / Stefan Kuhlins: Programmieren in C++, Springer, 1998

2. OO Sicht auf C++

H. M. Deitel / P. J. Deitel: C++ How to Program, Prentice–Hall Inc., 1997

Nicolai Josuttis: Objektorientiertes Programmieren in C++: von der Klasse zur Klassenbibliothek, Addison–Wesley, 1994

3. Beschreibung der Standardbibliothek

Nicolai Josuttis: Die C++–Standardbibliothek, Addison–Wesley–Longman, 1996

Martin Schader / Stefan Kuhlins: Die C++–Standardbibliothek, Springer, 1999

David R. Musser / Atul Saini: STL tutorial and reference guide: C++ programming with the standard template library, Addison–Wesley, 1996



4. Vergleich zu anderen Programmiersprachen

László Böszörményi / Carsten Weich: Programmieren mit Modula-3, Springer 1995

Greg Nelson: Systems programming with Modula-3, Prentice Hall, 1991

Manfred Nagl: Softwaretechnik und Ada 95, Vieweg, 1999

5. Architektursicht

Grady Booch: Object-oriented analysis and design with applications, Benjamin/Cummings, 1994

James Rumbaugh / Michael Blaha / William Premerlani / Frederick Eddy / William Lorenzen: Objektorientiertes Modellieren und Entwerfen, Prentice-Hall, 1993

Erich Gamma / Richard Helm / Ralph Johnson / John Vlissides: Design Patterns: Elements of Object-Oriented Software, Addison-Wesley, 1994

Frank Buschmann / Regine Meunier / Hans Rohnert / Peter Sommerlad / Michael Stal: Pattern-orientierte Software-Architektur, Addison-Wesley-Longman, 1999

Manfred Nagl: Softwaretechnik: Methodisches Programmieren im Großen, Springer, Berlin, 1990

6. Programmierstil, Empfehlungen

James O. Coplien: Advanced C++ programming styles and idioms, AT&T Bell Telephone Laboratories, 1992

Mats Henricson / Eric Nyquist : Industrial Strength C++, Prentice Hall, 1997

