

Vorbemerkungen

Dozent	Spann (Rechenzentrum des Mathematischen Instituts) wissenschaftl. Arbeitsgebiet: Numerische Mathematik
Veranstaltung	1. Teil eines einführenden C++-Kurses auch fortgeschrittene Aspekte jedoch keine Informatikveranstaltung Schwerpunkt auf imperativer Programmierung C++ als verbessertes C Objektorientierung jetzt nur eingeschränkt, ausführlich dann im 2. Teil (Programmieren II) Beispiele und Aufgaben überwiegend aus der Numerischen Mathematik und Diskreten Mathematik
Inhalt	imperative Sprachelemente von C++ einfache Klassen (u.a. keine Vererbung) Überladen von Operatoren (vereinfacht) Verwandtschaft von C++ und Java (grundleg. Aspekte)

Vorbemerkungen II

Hörerkreis	Bachelorstudierende aus der Mathematik und Wirtschaftsmathematik (ab 4. Semester) offen auch für Studierende anderer Fachrichtungen
Übungen	regelmäßige Bearbeitung von Übungsaufgaben - in diesem Semester keine Korrektur
Prüfung	voraussichtlich in der 1. oder 2. Semesterferienwoche wahrscheinlich Klausur im Hörsaal
Organisation	2 Std. Vorlesung (Folien, Infoblätter, Bsp.) 2 Std. Tutorium (Erläuterung der Aufgaben, Unterstützung bei der Prog., Verbess.vorschläge) <i>evtl. Vorlesung im Hybrid-Modus</i>
Allgemeines	Zeitaufwand nicht unterschätzen: C/C++-Syntax elegant, aber gewöhnungsbedürftig Erlernen ungewohnter Konzepte erforderlich Genaueres Arbeiten notwendig (Entwurf und Umsetzung) Fehlersuche - aktives Vorgehen und Frustrationstoleranz!

Historischer Abriss

1956	<i>Fortran</i>	<i>Backus (IBM)</i>	<i>Formula Translator</i> erste Programmiersprache
1960	<i>Algol 60</i>	<i>Naur et al.</i>	<i>Algorithmic Language</i> formal definierte Syntax (BNF)
1963	<i>CPL</i>		erste Systemprogrammiersprache, geringer Verbreitungsgrad
1967	<i>BCPL</i>	<i>Richards</i>	<i>Basic Combined Programming Language</i> typenlos, Kontrollstrukturen wie in C
1969	<i>B</i>	<i>Thompson</i>	Teilmenge von BCPL, interpretiert
1972	<i>C</i>	<i>Ritchie</i>	Datentypen, zusätzliche Operatoren, kompilierbar

Historischer Abriss II

- 1978 Kernighan/Ritchie: *The C Programming Language* (1st Ed.)
[K&R-C]
- 1980 Stroustrup: *C with Classes*
- 1986 Stroustrup: *The C++ Programming Language* (1st Ed.)
- 1988 Kernighan/Ritchie: *The C Programming Language* (2nd Ed.)
Funktionsprototypen (Anpassung an C++)
- 1989 ANSI C amerikanische Norm (1990 internationale Norm)
- 1990 Ellis/Stroustrup: *The Annotated C++ Reference Manual*
Mehrfachv., abstr. Klassen, protected, templates, exceptions
- 1991 Stroustrup: *The C++ Programming Language* (2nd Ed.)
- 1994 Stepanov: *Standard Template Library*
- 1997 Stroustrup: *The C++ Programming Language* (3rd Ed.)
name spaces, C++ Standard Library, neue Header, STL
- 1998 C++98 internationale Norm
- 2011 C++11 [C++0x] internationale Norm
- 2013 Stroustrup: *The C++ Programming Language* (4th Ed.)
- 2014 C++14 internationale Norm

[*] Vergleich mit anderen Programmiersprachen

- ▶ mehr arithmetische Datentypen und Zeichentypen
- ▶ viel mehr explizite und implizite Typumwandlungen
- ▶ Zeigerarithmetik (Adressarithmetik)
- ▶ Inkrement/Dekrementoperatoren, Bitoperatoren
- ▶ getrenntes Übersetzen (Funktionsbibliotheken!)
- ▶ Betriebssystemschnittstelle, Assemblereinbindung
- ▶ Ein/Ausgabe, Stringverarbeitung, Container über Bibliothek
- ▶ Überladen von Operatoren
- ▶ Laufzeitüberwachung von Feldgrenzen nicht voreingestellt
„der Programmierer muss wissen, was er tut“
- ▶ normalerweise Wertparameter, Variablenparameter über Referenzen oder Zeiger
- ▶ Effizienz als Entwurfsziel (im Unterschied zu Java)
[Enthält nur effizient implementierbare Sprachelemente]
- ▶ Unterstützung verschiedener Programmierstile (imperativ, objektorientiert, generisch, funktional)

[*] Weitere Aspekte

Mathematische Programmierung

- ▶ C++ nicht ideal (kein standardisierter numerisch effizienter Matrixtyp, aber Ausweichen auf C oder Fremdbibliotheken möglich)
- ▶ starke Verbreitung von Fortran im Numerikbereich, jedoch zunehmende Verfügbarkeit von numerischen C- und C++-Programmbibliotheken
- ▶ C++ und C gegenüber Fortran im Vorteil bei Integration nicht numerischer Programmteile

Sonstiges

- ▶ Starker Einfluss von C/C++ auf andere Programmier- und Skriptsprachen, z.B. Java, Javascript, Perl, PHP
- ▶ Dominanz von C/C++ im Open Source Bereich (trotz starker Maschinenabhängigkeit ist mittels geeigneter Werkzeuge portable Programmierung möglich!)