

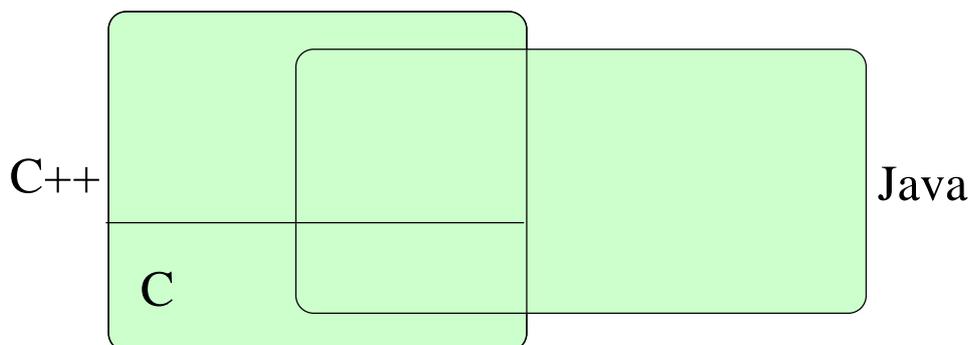
# Kapitel 1

## Einleitung

### 1.1 Vorwort

Dieser Text entstand am Lehrstuhl Wagner im Rahmen der Praktika, die Studierende im Hauptstudium machen müssen. Sie lernen im Grundstudium objektorientiertes Programmieren anhand der Programmiersprache *Java*. Es hat sich allerdings gezeigt, dass für Algorithmen und Datenstrukturen Java schlechter geeignet ist als C++. Deshalb sind wir dazu übergegangen, im ersten Teil des Praktikums C++ zu vermitteln. Dabei beschränken wir uns auf einige Teilaspekte von C++, die wir für notwendig und sinnvoll für ihre Aufgabe erachten. Insbesondere werden hardwarenahe Teile der Programmiersprache auf ein Minimum reduziert.

Es soll aber nicht gesagt werden, dass Java eine schlechte Programmiersprache wäre oder gar gänzlich unnützlich. Es ist vielmehr so, dass wir Dinge, die Java bereitstellt, nicht aber dafür andere Dinge, die es nicht bereitstellt, sehr wohl benötigen. Aus unserer Sicht der Dinge stellt sich das Verhältnis von Java, C und C++ folgendermaßen dar:



C++ kann seine Herkunft von C schon allein durch den Namen sicherlich nicht leugnen. Die meisten Teile von C sind in C++ enthalten. Es bietet aber durch die Mittel zum objektorientierten und generischen Programmieren weit mehr. In Java findet man (derzeit) nur die Hilfsmittel zur Objektorientierung wieder. Es setzt sich aber insbesondere bei der Speicherverwaltung stark von C und C++ ab. Auf der anderen Seite gehört eine wesentlich umfangreichere Bibliothek zu Java dazu, die z.B. die Programmierung von GUIs oder Netzwerkfähigkeit ermöglicht. Diese Dinge spielen jedoch bei Algorithmen und Datenstrukturen eine untergeordnete Rolle im Gegensatz zur Speicherverwaltung. Letztere ist dem Programmierer in Java jedoch gänzlich aus den Händen genommen. Summa summarum erscheint uns daher die Verwendung von C++ für unseren Zweck sinnvoll.

## 1.2 Voraussetzungen

Voraussetzung für diesen Kurs sind wie gesagt Kenntnisse der Programmiersprache Java. Insbesondere sollten allgemeine Programmierkonzepte wie

- Variablen
- Schleifen und Verzweigungen
- Unterprogramme
- Objektorientierung
- Syntax
- Fehlermeldungen interpretieren

bekannt sein.

## 1.3 Was C++ gegenüber Java fehlt

Der geübte Java-Programmierer wird bei einem Wechsel zu C++ sicherlich einige Dinge vermissen:

- Grafik (incl. GUI)
- Threads, Prozesssteuerung
- Netz
- Hilfen zur Dokumentation
- Reflection
- Garbage Collection ( $\Rightarrow$  Speicher selbst verwalten)

Selbstverständlich ist es möglich, diese Defizite durch zusätzliche Programme oder Programm-bibliotheken auszugleichen. In diesem Text wird darauf allerdings nicht eingegangen werden, da sie in unseren Praktika nicht nötig waren.

## 1.4 Danksagung

An dieser Stelle möchten wir Gabi Dorfmueller und Thomas Puppe danken, die als Versuchskaninchen für diese Einführung hergehalten haben und wertvolle Hinweise auf Lücken und Unzulänglichkeiten aufgedeckt haben. Außerdem danke ich Sabine Cornelsen für das Korrekturlesen.

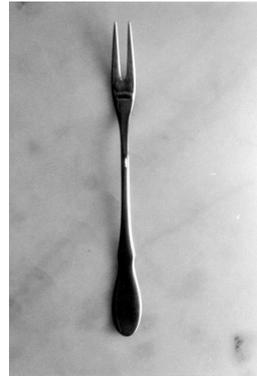
## 1.5 Literatur

Dieser Text soll beim Erlernen von C++ nur als Leitfaden dienen. Daher ist es wahrscheinlich notwendig, aber sicherlich sinnvoll, einige alternative Informationsquellen zu benutzen. Exemplarisch möchte ich folgende nennen:

- *ISO/IEC 14882 - Programming languages - C++*
- Stroustrup, Bjarne:  
*The C++ programming language*  
Addison-Wesley
- *The C++ report/Journal of Object-Oriented Programming.*  
SIGS Publ.
- Newsgroup *comp.lang.c++.moderated*
- John Lakos:  
*Large scale C++ software design*  
Addison-Wesley
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides:  
*Design Patterns, Elements of Reusable Object-Oriented Software*  
Addison-Wesley
- John J. Barton; Lee R. Nackman:  
*Scientific and engineering C++*  
Addison-Wesley
- Krzysztof Czarnecki; Ulrich Eisenecker:  
*Generative programming : methods, tools, and applications*  
Addison-Wesley
- Andrei Alexandrescu:  
*Modern C++ Design : generic programming and design patterns applied*  
Addison-Wesley
- Scott Meyers:  
*Effective C++ und More effective C++*  
Addison-Wesley-Longman
- Herb Sutter:  
*Exceptional C++ und More Exceptional C++*  
Addison-Wesley

## 1.6 Warnung

Da es ja in diesem Text darum geht, einige Teile von C++ vorzustellen, die in Java „weggelassen“ wurden, möchten wir eine Warnung vorweg schicken. Es ist nämlich nicht so, dass die Entwickler von Java keinen Grund gehabt hätten, dass sie nicht alle Konstruktionen von C++ übernommen haben. Zumindest in einigen Fällen war ihre Motivation, den Programmierer vor sich selbst zu schützen. Dies soll der bekannte Kindervers verdeutlichen:



*Messer, Gabel, Schere, Licht, ist für kleine Kinder nicht!*

Es gibt also durchaus Gefahren, die durch die neuen Möglichkeiten entstehen. Wie bei den genannten Werkzeugen können sie aber ziemlich hilfreich sein. (Oder haben Sie schon einmal versucht, ein Steak ohne Messer und Gabel zu essen?) Letztenendes ist unsere Hoffnung, dass sich der Leser als erwachsen genug erweisen wird, mit den „gefährlichen“ Seiten von C++ fertig zu werden.