

C++ Kurs
TU Dresden
Fakultät für Informatik

Maximilian Starke
Student der TU Dresden

30. März 2017

Inhaltsverzeichnis

1	Einrichtung	2
1.1	ISO C++	2
1.1.1	Allgemeines	2
1.1.2	Versionen	2
1.2	Dateien in einem C++ Projekt	3
1.3	Compiler	3
1.4	IDEs	4
1.4.1	JA oder NEIN	4
1.4.2	IDEs im Überblick	4
1.5	Referenzen	4
1.6	The Hello World	4
2	Datentypen in C++	6

Kapitel 1

Einrichtung

1.1 ISO C++

1.1.1 Allgemeines

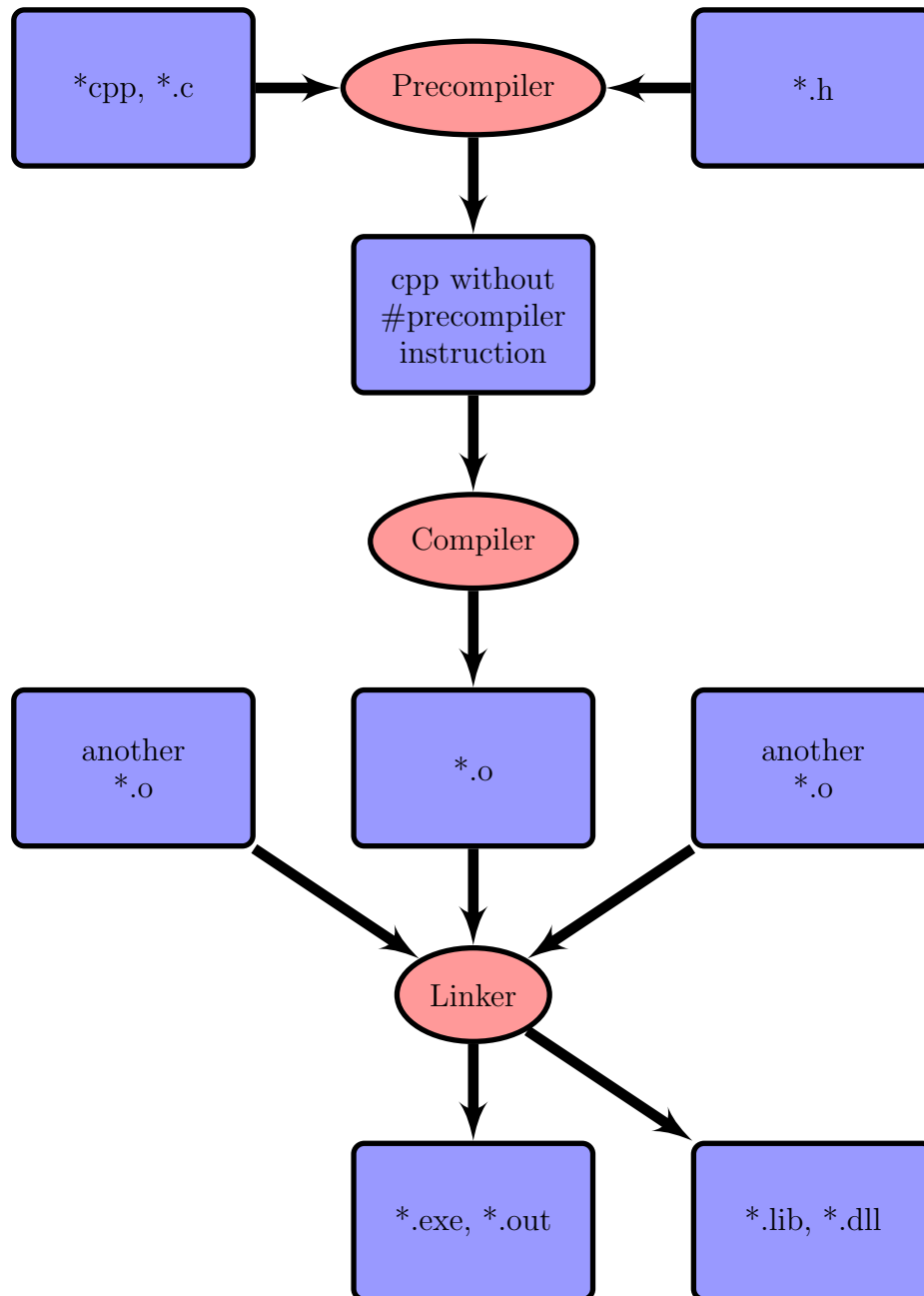
- ab 1979 von Bjarne Stroustrup bei AT&T entwickelt als Erweiterung der Programmiersprache C
- später von ISO genormt
- effizient und schnell - Schnelligkeit eines der wichtigsten Designprinzipien von C++
- hohes Abstraktionsniveau u.a. durch Unterstützung von OOP
- ISO Standard beschreibt auch eine Standardbibliothek
- C++ ist **kein** echtes Superset von C (siehe stackoverflow.com, ...)
- C++ ist (wie C) **case sensitive**
- Paradigmen:
 - **generisch** (durch Benutzung von Templates, automatische Erstellung multipler Funktionen für verschiedene Datentypen)
 - **imperativ** (Programm als Folge von Anweisungen, Gegenteil von deklarativ siehe Haskell und Logikprogrammierung)
 - **objektorientiert** (Klassen, Objekte, Vererbung, Polymorphie, Idee: Anlehnung an Realität)
 - **prozedural** (Begriff mit verschiedenen Bedeutungsauffassungen, Unterteilung des Programms in logische Teilstücke (Prozeduren), die bestimmte Aufgaben / Funktionen übernehmen)
 - **strukturiert** (prozedural und Teilung in Sequenz, Verzweigung, Wiederholung, ...)
 - **funktional** (ab C++11, Definitionsskleinkram, siehe Wikipedia, Programm als verschachtelter [...] Funktionsaufruf organisierbar)

1.1.2 Versionen

- C++98
- C++03
- C++11
 - wesentliche Neuerungen. Einführung von `constexpr`, Elementinitialisierer, ... Neue Bedeutung des Schlüsselworts `auto` `#` Referenzen ergänzen
- C++14
 - aufweichung der `constexpr` Bedingungen.
- C++17
 - soll 2017 vollendet werden.

1.2 Dateien in einem C++ Projekt

Dateiendung	Bezeichnung	Inhalt
(* .cpp) (* .cc)	Quelldatei	Funktionsimplementation, Klassenimplementation, Berechnungen bzw. eigentliche Arbeit erledigen
(* .h)	Headerdatei	Funktionsdeklaration, Klassendefinition, Bezeichner öffentlich bekannt machen
(* .o)	Objektdatei	Objektcode (Maschinencode) einer Übersetzungseinheit



1.3 Compiler

GCC
gcc
g++

1.4 IDEs

1.4.1 JA oder NEIN

ohne IDE	mit IDE
Compiler, Linker über Shell bedienen Texteditor evtl. make + makefile Dokumentationen	Projekteinstellungen & Buttons in IDE integriert automatisch generiertes makefile geordneter Menübaum
Einarbeitungszeit(??) für kleine und mittelgroße Projekte	Einarbeitungszeit (??) kleine, mittlere und große Projekte

1.4.2 IDEs im Überblick

IDE	Plattform	Anmerkungen
Eclipse, Netbeans Qt SDK Code::Blocks	Java (JVM) WIN, Linux, Mac WIN, Linux, Mac	in und für Java geschrieben, unterstützt auch C++ bringt umfangreiches Qt-Framework mit für GUIs u.v.m.
Visual Studio	Windows	kostenfreie BVersion für den Hausgebrauch: VS Community 2016 /2017RC, sehr umfangreich (Refactoring Tools, Debugger, Laufzeitanalyse, Frameworks wie MFC, ATL, WTL) und damit auch speicherintensiv, zu installierende Features wählbar, benutzt eigenen MS VC++ Compiler
Orwell DEV-C++	Windows	
Geany	Linux, WIN	schlichter Texteditor mit Syntaxhighlighting und diversen Compile Buttons
KDevelop	Linux, WIN	#
Anjuta	Linux	#
XCode	MacOS	„hauseigene“ IDE von Apple

1.5 Referenzen

- Buch:
 - Wolf, Jürgen: C++ - Das umfassende Handbuch. Rheinwerk Computing
- Websites:
 - <http://en.cppreference.com/w/>
 - <http://www.cplusplus.com/reference/>

Anmerkung ergänzen

1.6 The Hello World

```
#include <iostream>
```

```
int main(int argc, char* argv[])  
// main-Funktion: Einstiegspunkt der Anwendung  
// count: Anzahl der uebergebenen Parameter  
// arg: Pointer auf ein Array von Pointern auf C-Style-Strings (die Parameter)  
// Parameter der main-Funktion duerfen in der Signatur auch weggelassen werden.  
  
// Parameter der main-Funktion  
{ // Beginn vom Anweisungsblock der main-Funktion  
  
    std::cout << "Hello World" << std::endl;  
    /*  
    * implizite Klammerung:
```

```

* ((std::cout) << "Hello World") << (std::endl);
* std          ... ein Namensraum
* ::          ... scope-Operator (Bereichsoperator)
* cout:       ... gepufferter Standardausgabestream
* <<         ... Ausgabeoperator (auch bitshift-Operator)
* "Hello World" ... C-Style-String Literal
* endl        ... Objekt aus dem std Namensraum,
                das einen Zeilenumbruch ('\n') erzeugt.
* ;          ... Abschluss einer einzelnen Anweisung
*/

for(int i = 0; i < argc; ++i ){
    std::cout << i << ". Parameter: " << argv[i] << '\n';
} // Beispiel fuer die (Verarbeitung &) Ausgabe der Komandozeilenargumente

return 0; // Rueckgabewert 0 "erfolgreich (ohne Fehler) beendet"
}

```

Kapitel 2

Datentypen in C++