

# Einleitung

- Geschichte von C
- Eigenschaften von C
- Geschichte von C++
- Eigenschaften von C++

# Einleitung

## Geschichte von C

eng verknüpft mit der Entwicklung des Betriebssystems UNIX in den AT&T Bell Labs

C wurde 1971/72 von Dennis Ritchie für UNIX entworfen und unter UNIX implementiert auf einer DEC PDP-11.

Betriebssystem, C-Compiler und nahezu alle UNIX-Kommandos und -Bibliotheken sind in C geschrieben.

Entwurfsziele:

- Unterstützung der **Strukturierten Programmierung**
- Möglichkeiten einer **hardwarenahen Programmierung** vergleichbar mit Assembler
- **Performance** des Objektcodes vergleichbar mit Assembler

C wurde von **Kernighan** und **Ritchie** weiterentwickelt. 1978 erscheint ihr Buch „The C Programming Language“.

ab 1983 Standardisierung von C  
1989 als ANSI Standard X3.159-1989 verabschiedet, der 1990 durch den ISO/IEC Standard 9899 *Programming Languages – C* ersetzt wurde.

# Einleitung

## Eigenschaften von C

relativ „maschinennah“:

arbeitet mit den gleichen Objekten wie der Prozessor

- Zeichen,
- Zahlen und
- Speicheradressen

keine Operationen, die eine direkte Manipulation von zusammengesetzten Objekten wie Zeichenketten, Mengen, Listen oder Feldern ermöglichen

keine Sprachelemente, die parallele Prozesse unterstützen, und keine Ein-/Ausgabe-Anweisungen

Der C-Standard definiert auch die sogenannte **C-Standardbibliothek**, die Funktionen zur Ein-/Ausgabe, String-Manipulation, Speicherallokation etc. enthält.

# Einleitung

## Eigenschaften von C

### Einführendes Beispiel (traditionell)

```
/* hello.c */  
  
#include <stdio.h>  
main()  
{  
    printf("Hello, World\n");  
}
```

- Einschluss vordefinierter Programmteile
- Identifikation des Hauptprogramms
- Anweisung: Ausgabe der Zeichenkette *Hello, World*
- Kommentar

# Einleitung

## Einführendes Beispiel (real)

```
/* getopt.c -- Turbo C
Copyright (c) 1986,87,88 by Borland Int'l Inc. */

int getopt(int argc, char *argv[], char *options)
{
    unsigned char ch;
    char *optP;
    if (argc > optind) {
        if (letP == NULL) {
            if ((letP = argv[optind]) == NULL ||
                *(letP++) != SW) goto gopEOF;
            if (*letP == SW) {
                optind++; goto gopEOF;
            }
        }
        if (0 == (ch = *(letP++))) {
            optind++; goto gopEOF;
        }
        if (':'==ch ||
            (optP=strchr(options, ch))==NULL)
            goto gopError;
        if (':' == *(++optP)) {
            optind++;
            if (0 == *letP) {
                if (argc <= optind)
                    goto gopError;
                letP = argv[optind++];
            }
            optarg = letP;
            letP = NULL;
        } else {
            if (0 == *letP) {
                optind++;
                letP = NULL;
            }
            optarg = NULL;
        }
    }
    return ch;
}
:
} /* getopt */
```

# Einleitung

## Geschichte von C++

ab 1980 von Bjarne Stroustrup in den AT&T Bell Labs als objektorientierte Sprache entwickelt

zunächst “C mit Klassen” genannt

Bezeichnung **C++** existiert seit 1983

einige Ideen von C++ sind in den ANSI-C-Standard eingeflossen

z.B. Funktionsdeklarationen durch Prototypen

ab 1987 Standardisierung von C++

im Januar 1997 zweiter ISO-C++-Standardentwurf

Im Sommer 1998 wird der ISO/IEC Standard

14882-1998 *Information Technology – Programming Languages – C++* verabschiedet.

# Einleitung

## Eigenschaften von C++

objektorientierte Erweiterung von C + verbessertes C

### objektorientierte Erweiterung von C:

- modulare Programmierung, Datenkapselung
- Datenabstraktion, abstrakte Datentypen, Klassen
- Vererbung, Mehrfachvererbung, Polymorphismus, spätes Binden

### verbessertes C:

- // -Kommentare
- Konstanten
- enum und struct definieren Typnamen
- Deklaration ist eine Anweisung => kann an beliebiger Stelle im Programm stehen
- Überladen von Funktionsnamen
- Überladen von Operatoren
- vorgegebene Parameterwerte
- Referenzparameter
- inline-Spezifikation
- Operatoren new und delete
- objektorientierte Ein-/Ausgabe-Bibliothek
- Schablonen (templates)
- Ausnahmebehandlung

# Einleitung

## Was ist C++ außerdem?

eine schwierige Programmiersprache

- viele Sprachelemente
- viele Regeln (z.B. automatische Typkonversion)

Folgerungen daraus:

- vollständige Behandlung von C++ nicht möglich
- Konzentration auf das Wesentliche