Übersicht

Lehrinhalt: Programmierung in C

- **Überblick über Programmiersprachen, Allgemeines**
- C-Basisdatentypen, Zahlendarstellung, Variablen, Konstanten
- Operatoren und Ausdrücke
- Anweisungen
- Kontrollstrukturen
- Funktionen
- Zeiger und Felder
- Zeichenketten (Strings)
- Benutzerdefinierte Datentypen
- Dynamischer Speicher
- Dateiarbeit
- Funktionspointer, Rekursion
- Preprozessor

Allgemeines

- Verschiedene Sprachkonzepte
- C-Sprachfamilie
- C-ähnliche Programmiersprachen
- Allgemeines zu C

```
#include <stdio.h>

int main()
{
   printf("hello world\n");
   return 0;
}
```

Verschiedene Sprachkonzepte

Programmiersprachen unterteilt man in s.g.:

- **Imperative Programmiersprachen** –
  Beschreibung einer Berechnungsvorschrift durch einzelne Schritte und einen Steuerfluss

  Beispiele: PASCAL, C, C++, Java

- **Deklarative Programmiersprachen** –
  beschreiben das Problem oder das Ziel des Programms

  Beispiele: Datenbankabfragesprache SQL,
  Logische Programmiersprache PROLOG

Standardisierung für C-Sprachen

C

- K&R – C 1972, Kerningham-/Ritchie
- ANSI/ISO-C89 und C90
- ANSI/ISO C95
- ANSI/ISO-C99
- ISO/IEC C 11 (2011)
- ISO/IEC C18 (2018)

C++

- C++ 1985, Bjarne Stroustrup
- ANSI/ISO-C++ 98 erster Standard, basierend auf C90 mit C++ Erweiterungen
- ANSI/ISO-C++ 03 (2000)
- ISO/IEC C++11 (2011)
- ISO/IEC C++14 (2014)
- ISO/IEC C++17 (2017)
- ISO/IEC C++20 (…)

Die Großfamilie der C-Sprachen

C/C++ - ähnliche Sprachen:

- **Java** – C++-ähnliche objektorientierte Sprache, SUN 1995
- **C#** – C++-ähnlich, objektorientiert, Microsoft 2001
- **Objective-C** - objektorientiertes C, nicht kompatibel zu C++
- **C++.NET** mit Microsoft-Erweiterungen für .NET,
  s.g. „managed C++“
- **JavaScript** und **PHP** – Sprachen zur Verarbeitung innerhalb von Web-Anwendungen, prozedural mit objektorientierten Erweiterungen
- **C-Shell** – Skriptsprache zur UNIX-Shell-Programmierung

… und viele andere, die von C beeinflusst wurden

Programmiersprache C

Imperative Programmiersprache:

C-Anweisungen werden in der im Programm angegebenen Reihenfolge ausgeführt:

- zeilenweise (von oben nach unten)
- innerhalb Zeile möglicherweise mehrere Anweisungen, dann von links nach rechts
- Steuerfluss-Konstrukte (*if, else, for, while, do, break, continue, return*) zur Beeinflussung der Abarbeitungsreihenfolge

Zum Vergleich: manch andere Programmiersprachen (logische und funktionale, beispielsweise PROLOG) arbeiten die Ausdrücke nicht notwendigerweise in der im Programm angegebenen Reihenfolge ab.

Programmiersprache C

Ein Programm besteht aus:

Variablenvereinbarungen: hier werden Bezeichner für die Verarbeitungselemente festgelegt. Es wird ein Typ für jede Variable angegeben, z.B. Ganzzahl (int) oder Zeichen (char)

Anweisungen:

- zur Verarbeitung der Variablen
- zur Beeinflussung des Steuerflusses

Mehrfach auftretende Anweisungsfolgen werden oft in **Funktionen** gekapselt, die Unterprogrammen in C darstellen.

Eine Hauptfunktion (**main-Funktion**) ist der Punkt, an dem mit der Ausführung eines C-Programms begonnen wird.

Programmiersprache C

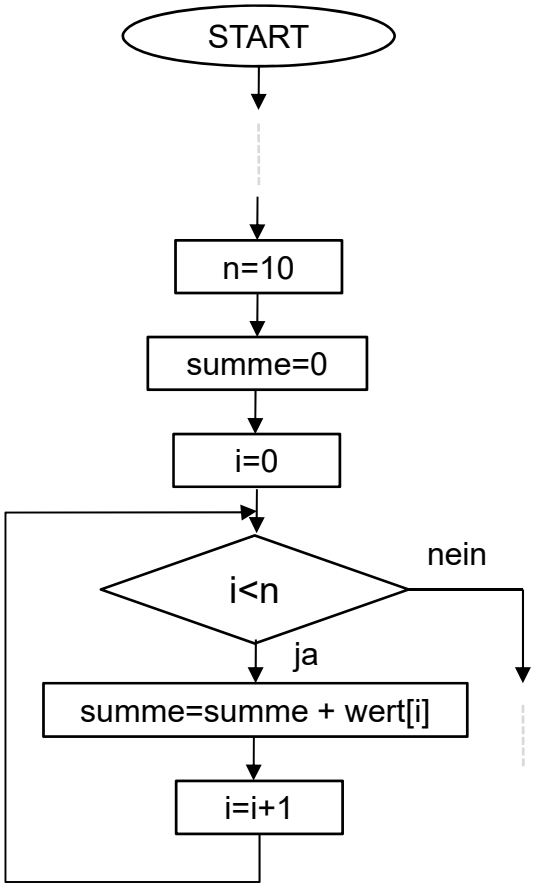
C-Programme werden in ein Binärprogramm übersetzt
Das Übersetzerprogramm ist der s.g. Compiler.

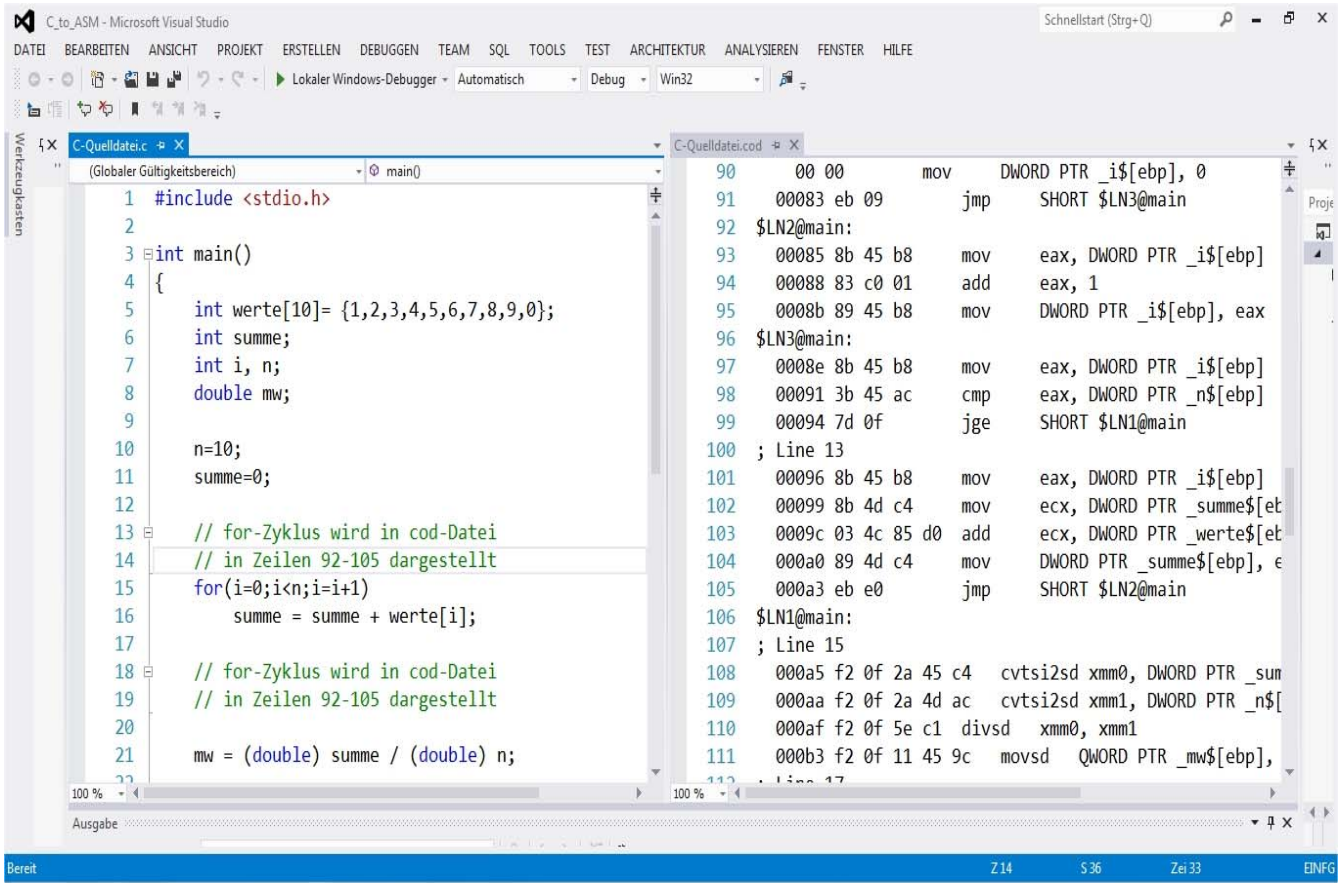
Die Übersetzung verlangt ein syntaktisch richtiges C-Programm,
d. h. das Programm muss sprachlichen Regeln genügen.

Andere Programmiersprachen werden zum Teil interpretiert,
d.h. Anweisungen werden schrittweise übersetzt und ausgeführt.

Erzeugung eines ausführbaren Programms

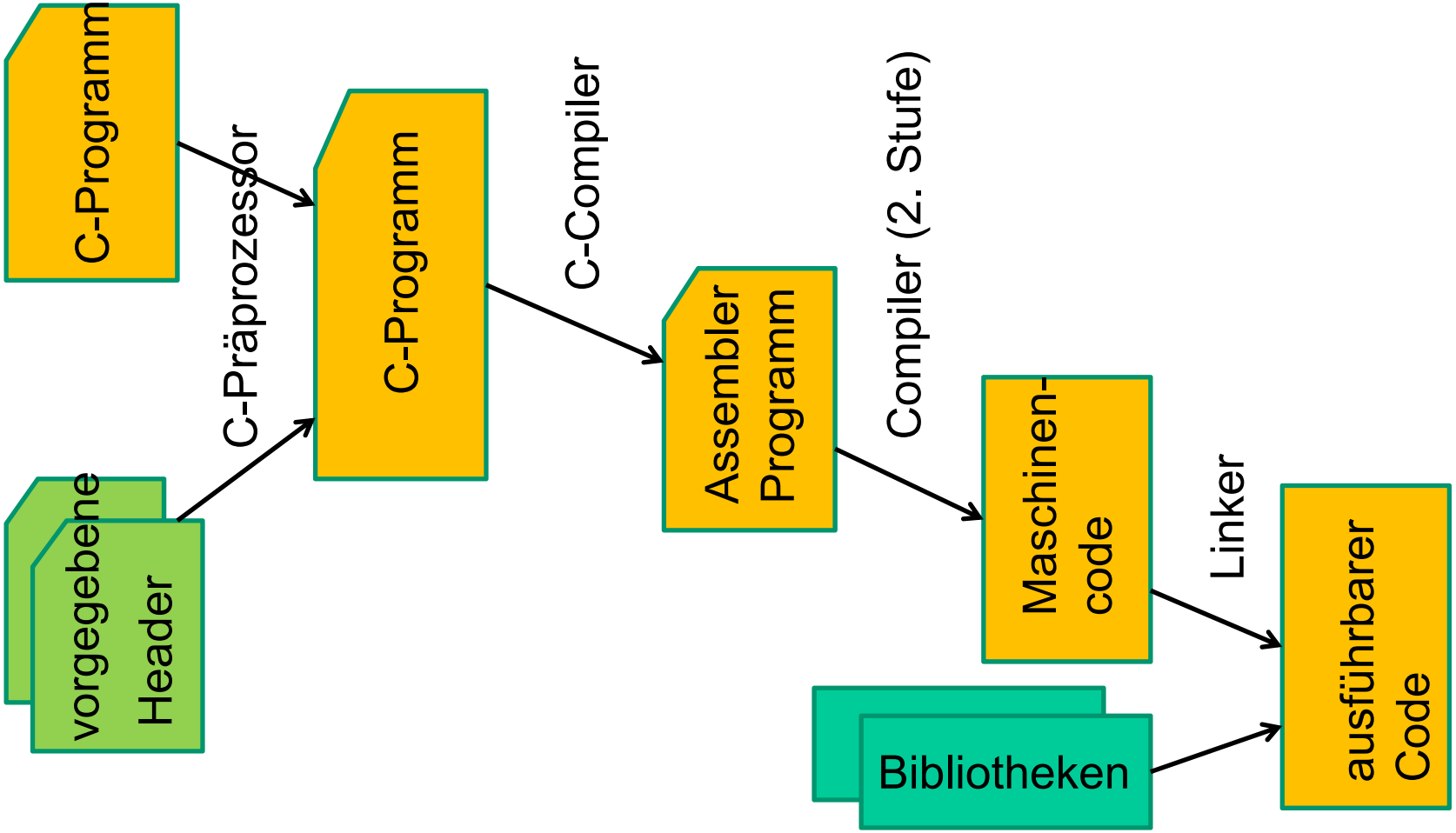
Algorithmus →→→ C-Programm →→→ x86-Maschinencode



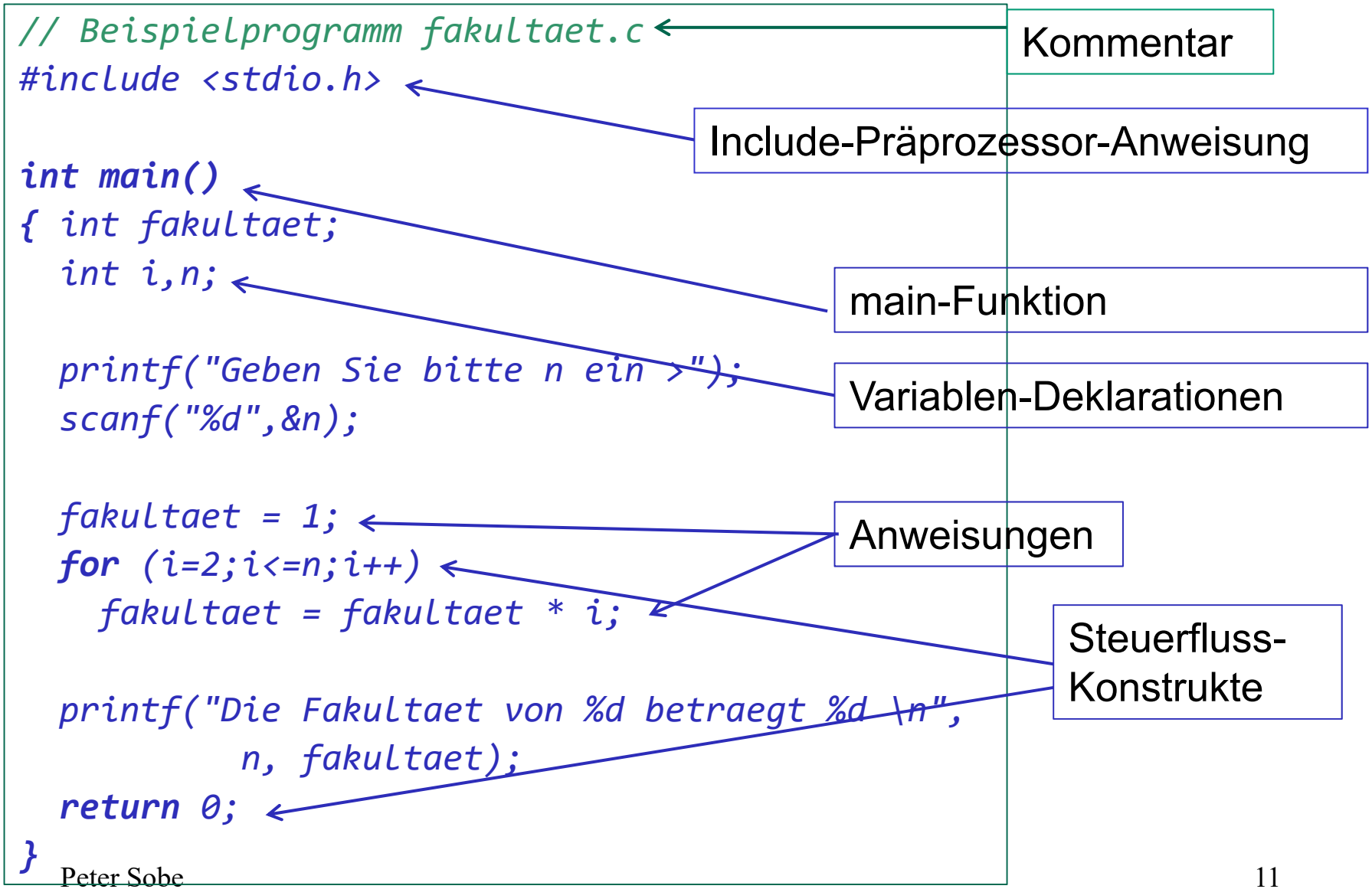


Programmiersprache C

Vor dem Ausführen des Programms:



Aufbau eines C-Programms



C-Programm als Text

Ein C-Programm wird in einer (oder mehreren) Quelldatei(en) als Text niedergeschrieben.

Die Sprache C nutzt ein Alphabet

- aus Buchstaben a-z, A-Z,
- aus Ziffern 0-9
- und aus Sonderzeichen, z.B. !,?, (,), _

Aus dem Alphabet werden Bezeichner, Schlüsselworte, Konstanten, Zeichenketten, Operatoren, Trennzeichen und Kommentare gebildet.

Groß und Kleinschreibung wird unterschieden.

Kommentare

Kommentare in einem C-Programm werden beim Übersetzen übergangen. Sie haben keine Auswirkung auf das Programm.

Kommentare sind aber nützlich, um Erklärungen im Programm zu notieren.

Oft dienen Kommentare der Dokumentation von Programmteilen.

Beginn und Ende durch /* und */

```
/* Das ist ein Kommentar */

/* Das ist ein mehr-
   zeiliger Kommentar */

// Das ist ein einzeiliger Kommentar
```

Kommentare

```
/* ---------------------------------------- */
/* Programm zur Berechnung eines Zylindervolumens */
/* d: Durchmesser Grundkreis, h: Hoehe      */
/* ---------------------------------------- */
#include <stdio.h>
int main()
{       double d, h, gf, v;
        printf("d:");
        scanf("%lf",&d);
        printf("h:");
        scanf("%lf",&h);

        // Grundflaeche
        gf = 3.14159 * d*d / 4;
        // Volumen
        v= gf*h;
        printf("Grundflaeche: %lf, Volumen: %lf \n", gf, v);
        return 0;
}
```

Formatfreiheit

C ist eine formatfreie Sprache, d.h. das Aussehen, die Zeilenumbrüche oder Einrückungen haben keinen Einfluss auf die Funktion.

Beispiele:

```
int main()
{
  int a;
  scanf("%d",&a);
  printf(" quadrat = %d\n", a*a);
  return 0;
}
```

```
int main() { int a; scanf("%d",&a);
printf(" quadrat = %d\n", a*a);
return 0; }
```

Trotzdem sollte das Programm wie links dargestellt aussehen, um bessere Lesbarkeit zu gewährleisten.