

# Kapitel 1

## Eine erste Einführung

*Was erreichen Sie mit der Programmierung in C? Was benötigen Sie, um anzufangen? Wie nutzen Sie dieses Buch möglichst gut?*

Die Programmiersprache C gibt es bereits seit Anfang der 1970er-Jahre. Dennoch ist diese Sprache nach wie vor aktuell, schnell und universell einsetzbar. Ihr Einsatzgebiet erweitert sich ständig und erstreckt sich vom normalen Anwender-PC über das eingebettete System innerhalb eines technischen Geräts bis hin zum vernetzten Rechnernetz.

**Universell**

### 1.1 Welche Vorteile bietet C?

Die Sprache C besitzt einen breiten Anwendungsbereich und erfreut sich großer Bekanntheit und Popularität. C-Programme sind zwischen den verschiedenen Rechnersystemen leicht portierbar. Sie benötigen weder eine spezielle Laufzeitumgebung noch einen Interpreter.

**Portierbar**

Sie können mit C schnelle, ressourcensparende Programme erstellen. Dies ist unter anderem dem direkten Zugriff auf den Speicherplatz und die Hardware zu verdanken.

**Schnell**

C ist weitverbreitet und bindet Sie nicht an einen bestimmten Hersteller. Der C-Compiler aus der *GNU Compiler Collection* (kurz: GCC-C-Compiler) steht für viele Betriebssysteme zur Verfügung und stellt das zentrale Element vieler Entwicklungsumgebungen dar. Einige davon werden auch in diesem Buch eingesetzt.

**GCC**

Durch seinen einfachen Aufbau und seine geringe Anzahl an Schlüsselwörtern ist C leicht erlernbar. Seine Effizienz hat C zur Basis vieler bekannter Sprachen gemacht.

**Einfach**

Sowohl zur Programmierung von Anwendungen für den Endbenutzer als auch zur Systemprogrammierung für Bestandteile und Erweiterungen von Betriebssystemen findet C seinen Einsatz.

## 1.2 Was benötige ich zum Programmieren?

**IDE** Für die verbreiteten Betriebssysteme (*Windows, Ubuntu Linux, macOS*) gibt es eine Reihe IDEs (kurz für: *Integrated Development Environment*, zu Deutsch: *Integrierte Entwicklungsumgebung*), die Ihnen alles bieten, was Sie als angehender C-Programmierer benötigen:

- ▶ einen Editor, mit dessen Hilfe Sie den Programmcode schreiben,
- ▶ einen Compiler, mit dessen Hilfe Sie den Programmcode in die Sprache übersetzen, die der jeweilige Rechner versteht, und
- ▶ einen Debugger, der Ihnen bei der Fehlersuche hilft.

Viele Editoren bieten, um Ihnen die Arbeit zu erleichtern, ein sogenanntes *Syntax-Highlighting*, also ein farbliches Hervorheben der Elemente von C.

**Code::Blocks** Sie können unter anderem mit der übersichtlichen, frei verfügbaren Entwicklungsumgebung *Code::Blocks* arbeiten, die unter *Windows, Ubuntu Linux* und *macOS* zur Verfügung steht. Sie ist einfach zu installieren und leicht zu nutzen.

**Weitere IDEs** Die Installation und die Nutzung dieser und anderer Entwicklungsumgebungen wird in Anhang A beschrieben. Sie können auch mit einem C-Compiler auf Kommandozeilenebene arbeiten. Dies vereinfacht teilweise die Nutzung externer Bibliotheken, die Ihnen weitere Funktionen bieten.

## 1.3 Wie ist dieses Buch aufgebaut?

In Kapitel 2 werde ich Ihnen anhand des klassischen Hallo-Welt-Programms die Bedienung der Entwicklungsumgebung erläutern.

**Grundkurs** In Kapitel 3 bis Kapitel 11 lernen Sie anhand eines anschaulichen Beispiels aus dem Alltag die wichtigen Elemente der Programmierung mit C kennen: Variablen, Operatoren, Kontrollstrukturen, Funktionen, Felder, Zeichenketten, Zeiger, Strukturen und Dateispeicherung.

Dies geschieht Schritt für Schritt, mit ausführlicher Erläuterung der Begriffe. Dabei werden wir die Programme Stück für Stück durch das neu erworbene Wissen erweitern. Allerdings werden wir uns dabei nicht in sämtlichen Details der Programmierung verlieren.

Anschließend sind wir in der Lage, die wichtigen Begriffe richtig einzuordnen. Kapitel 12 wird die Kenntnis der Programmier-elemente vertiefen. Wir schauen auch einmal nach links und rechts: Welche Möglichkeiten bieten diese Elemente noch? Auf welche Weise können sie miteinander verbunden werden?

Kapitel 13 bis Kapitel 18 erläutern zahlreiche nützliche Funktionen aus verschiedenen Themengebieten: dynamische Felder, Zahlen und Mathematik, Präprozessor, Dateien und Verzeichnisse, Datum und Zeit sowie die Kommunikation mit dem Betriebssystem.

In Kapitel 19 und Kapitel 21 werde ich Ihnen die Vielfalt der Programmierung mit C anhand von zwei Beispielen zeigen. Sie lernen:

- ▶ wie Sie auf SQLite-Datenbanken zugreifen
- ▶ wie Sie grafische Benutzeroberflächen mithilfe von *GTK+* erstellen

Kapitel 20 bietet einige größere Beispielprojekte. Für mehrere Spiele und andere Programme verbinden wir viele Elemente der bisherigen Kapitel jeweils zu einem umfangreichen Programm.

Der Anhang bietet die notwendigen Installationsanleitungen, wichtige Befehle unter Unix, die Schlüsselwörter der Sprache C, eine Funktionsreferenz und die Lösungen der Übungsaufgaben.

Für die Hilfe bei der Erstellung dieses Buchs bedanke ich mich bei Anne Scheibe und dem ganzen Team des Rheinwerk-Verlags.

## 1.4 Wie sollten Sie mit diesem Buch arbeiten?

Lernen Sie Schritt für Schritt die einzelnen Programmier-elemente, ihre Besonderheiten und ihre Vorteile kennen. Die Beispielprogramme und die Erläuterungen helfen Ihnen dabei.

Verändern Sie die Programme, und prüfen Sie, ob die Ergebnisse so sind, wie Sie sie erwartet haben. Lösen Sie die Übungsaufgaben, aber erstellen Sie auch Programme nach eigenen Aufgabenstellungen, die die neu erlernten Programmier-elemente enthalten.

Das Buch gibt Ihnen Hinweise auf viele typische Einsteigerfehler. Sollten Sie gerade überlegen, warum Ihr Programm nicht das ausführt, was Sie erwarten, prüfen Sie es einmal dahingehend.

Vertiefung

Datenbanken, GUIs

Beispiele

Programmieren

Fehler

**Webseite zum Buch** In Anhang A werden die Links zum Herunterladen der notwendigen Software genannt. Auf der Webseite zum Buch (<http://www.rheinwerk-verlag.de/5087>) finden Sie unter »Materialien zum Buch« alle Beispielprogramme des Buchs in fertiger Form. Der Dateiname steht hier im Buch jeweils in der Listing-Unterschrift.

**E-Book-Ausgabe** Leser der E-Book-Ausgabe finden die Beispielprogramme auch im beigefügten Downloadpaket.

## 1.5 C-Standards

Seit dem ersten Auftritt der Sprache C Anfang der 1970er-Jahre gab es einige Änderungen und Erweiterungen bei der Programmierung mit C. Einige davon sind auch für Einsteiger interessant.

**C95, C99, C11** Mit dem Begriff *Standard-C* wird normalerweise die Programmierung nach der im Jahre 1990 festgelegten Norm *ISO/IEC 9899:1990* bezeichnet, die auch kurz C90-Standard genannt wird. Seitdem gab es drei weitere Standards: C95, C99 und C11. Der C18-Standard enthält gegenüber dem C11-Standard nur Fehlerkorrekturen, keine sprachlichen Änderungen. Die Änderungen gegenüber Standard-C werden an der jeweiligen Stelle im Buch hervorgehoben.

**Schalter** In den verschiedenen Entwicklungsumgebungen können Schalter gesetzt werden, damit die Standards C99 und C11 beachtet werden, falls dies nicht standardmäßig der Fall ist. Eine Anleitung finden Sie jeweils im zugehörigen Abschnitt in Anhang A.

## 1.6 Ubuntu Linux unter Windows

**Aktuelle Version** Die Beispielprogramme für dieses Buch habe ich mithilfe der im Januar 2020 aktuellen Version 9.2.1 des GCC-C-Compilers unter *Ubuntu Linux* (64-Bit-Version) erstellt, wobei ich den Sprachstandard C11 berücksichtigt habe.

**VirtualBox** Sie müssen keine Bedenken haben, falls Sie mit den Programmen dieses Buchs unter einem anderen Betriebssystem arbeiten möchten. Sollten Sie aber auf Ihrem *Windows*-System noch ca. 20 GB Platz haben, so können Sie zum Ausprobieren *Ubuntu Linux* in einer virtuellen Maschine unter *Oracle VirtualBox* installieren, ohne Ihr *Windows*-System wesentlich zu beeinflus-

sen. Die Installation wird ausführlich in Anhang A beschrieben. Nach der Installation können Sie auf Ihrem Rechner parallel unter *Windows* und unter *Ubuntu Linux* arbeiten.

Sowohl *Ubuntu Linux* als auch das *Windows*-Programm *Oracle VirtualBox* sind frei verfügbar und können später auch wieder von Ihrem *Windows*-System deinstalliert werden.