AKTUELL ZU C++23

Ulrich **BREYMANN**

7. Auflage

C++ programmieren

// C++ LERNEN
// PROFESSIONELL ANWENDEN
// LÖSUNGEN NUTZEN

Im Internet: alle Beispiele und mehr auf www.cppbuch.de

HANSER

Breymann

C++ programmieren

Ulrich Breymann

C++ programmieren

C++ lernen – professionell anwenden – Lösungen nutzen

7., überarbeitete Auflage

HANSER

Prof. Dr. Ulrich Breymann lehrte Informatik an der Fakultät Elektrotechnik und Informatik der Hochschule Bremen.

Kontakt: breymann@hs-bremen.de

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Lektorat: Brigitte Bauer-Schiewek
Copy editing: Petra Kienle, Fürstenfeldbruck
Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München
Umschlagrealisation: Max Kostopoulos
Titelmotiv: © stock.adobe.com/michels
Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Gottingen
Printed in Germany

Print-ISBN: 978-3-446-47689-9

E-Book-ISBN: 978-3-446-47846-6

E-Pub-ISBN: 978-3-446-47964-7

Inhalt

Vorwort

Diese Auflage unterscheidet sich von der vorherigen durch eine gründliche Überarbeitung und die Umstellung auf den 2023 von der zuständigen ISO/IEC-Arbeitsgruppe verabschiedeten C++-Standard. Abschnitt A.7 bietet eine Übersicht der in diesem Buch berücksichtigten Änderungen. Das Buch ist konform zum C++23-Standard, ohne den Anspruch auf Vollständigkeit zu erheben – das Standarddokument [ISOC++] umfasst mehr als 2100 Seiten. Sie finden in diesem Buch eine verständliche und mit vielen Beispielen angereicherte Einführung in die Sprache, unabhängig vom Betriebssystem.

Für wen ist dieses Buch geschrieben?

Es ist für alle geschrieben, die einen kompakten und gleichzeitig in die Tiefe gehenden Einstieg in die Programmierung mit C++ suchen. Es ist für Interessierte ohne Programmiererfahrung gedacht und für andere, die diese Programmiersprache kennenlernen möchten. Beiden Gruppen dient das Buch als Lehrbuch und Nachschlagewerk.

Ein umfassendes Handbuch

Die ersten zehn Kapitel führen in die Sprache ein, die folgenden behandeln fortgeschrittene Themen. Die sofortige praktische Umsetzung des Gelernten anhand von leicht nachvollziehbaren Beispielen steht im Vordergrund. Klassen und Objekte, Templates und Exceptions sind Ihnen bald keine Fremdworte mehr. Es gibt 99 Übungsaufgaben – mit Musterlösungen im Anhang und zum Download. Durch das Studium dieser Kapitel werden aus Neulingen bald Fortgeschrittene – und mithilfe der weiteren Kapitel Experten.

C++ in praktischen Anwendungen

Sie finden kurze Einführungen in die Themen Programmierung paralleler Abläufe, Netzwerk-Programmierung einschließlich eines kleinen Webservers, Datenbankanbindung, grafische Benutzungsoberflächen und Zugriff auf die KIs ChatGPT und DALL·E 2. Durch den Einsatz der Boost-Library und des Qt-Frameworks wird größtmögliche Portabilität erreicht.

Softwareentwicklung ist nicht nur Programmierung

Sie lernen die Automatisierung der Programmerzeugung mit Make kennen. Das Programmdesign wird durch konkrete Umsetzungen von Design-Mustern nach C++ unterstützt. Das Kapitel über Unit-Tests zeigt, wie Programme getestet werden können. Das integrierte »C++-Rezeptbuch« mit mehr als 150 praktischen Lösungen, der Teil über die C++-Standardbibliothek, das umfangreiche Register und das detaillierte Inhaltsverzeichnis machen das Buch zu einem praktischen Nachschlagewerk für alle, die sich mit der Softwareentwicklung in C++ beschäftigen.

Moderne Programmiermethodik

Sie möchten Programme schreiben, die hohen Qualitätsansprüchen gerecht werden. Dazu gehört das Know-how, C++ richtig einzusetzen. Dass ein Programm läuft, reicht nicht. Es soll auch gut entworfen sein, möglichst wenige Fehler enthalten, selbst mit Fehlern in Daten umgehen können, verständlich geschrieben und schnell in der Ausführung sein. Deshalb liegt ein Schwerpunkt des Buchs auf guter Codierpraxis entsprechend den »C++ Core Guidelines«. Die Umsetzung wird an vielen Beispielen gezeigt.

Wie benutzen Sie dieses Buch am besten?

Es eignet sich zum Selbststudium oder als Begleitbuch zu einem Kurs oder einer Vorlesung. Man lernt am besten durch eigenes Tun! Dabei hilft es, die Beispiele herunterzuladen, sie zu studieren und zu modifizieren (*http://www.cppbuch.de/*). Auch wird empfohlen, die Übungsaufgaben zu lösen. Um sowohl Anfängern als auch Fortgeschrittenen gerecht zu werden, gibt es einfache, aber auch schwerere Aufgaben. Wenn Ihnen eine Lösung nicht gelingt – einfach bei den Lösungen im Anhang nachsehen bzw. im Verzeichnis *cppbuch/loesungen* der downloadbaren Beispiele. Und dann versuchen, die Lösungen nachzuvollziehen.

Wo finden Sie was?

Bei der Programmentwicklung wird häufig das Problem auftauchen, etwas nachschlagen zu müssen. Es gibt die folgenden Hilfen: Erklärungen zu Begriffen sind im *Glossar* aufgeführt. Es gibt ein umfangreiches *Stichwortverzeichnis* und ein detailliertes *Inhaltsverzeichnis*. Der Anhang enthält unter anderem verschiedene hilfreiche Tabellen und die Lösungen der Übungsaufgaben. Auf der Webseite *http://www.cppbuch.de/* finden Sie die Software zu diesem Buch. Sie enthält alle Programmbeispiele und die Lösungen zu den Aufgaben. Sie finden dort auch weitere Hinweise, Errata und nützliche Internet-Links.

Zu guter Letzt

Allen Menschen, die dieses Buch durch Hinweise und Anregungen verbessern halfen, sei an dieser Stelle herzlich gedankt. Insbesondere Prof.Dr .Ulrich Eisenecker danke ich für seine hilfreichen Kommentare. Frau Irene Weilhart vom Hanser Verlag und dem Lektorat danke ich für die gute Zusammenarbeit.

Bremen, im Juni 2023

Ulrich Breymann

Teil I:

Einführung in C++

1

Es geht los!

Dieses Kapitel behandelt die folgenden Themen:

- Entstehung und Entwicklung der Programmiersprache C++
- Objektorientierte Programmierung – erste Grundlagen
- Wie schreibe ich ein Programm und bringe es zum Laufen?
- Einfache Datentypen und Operationen
- Ablauf innerhalb eines Programms steuern
- Erste Definition eigener Datentypen
- Standarddatentypen vector und string
- Einfache Ein- und Ausgabe
- Guter Programmierstil

1.1 Historisches

C++ wurde etwa ab 1980 von Bjarne Stroustrup als die Programmiersprache »C with classes« (englisch *C mit Klassen*), die Objektorientierung stark unterstützt, auf der Basis der Programmiersprache C entwickelt. Später wurde die neue Sprache in C++ umbenannt. ++ ist ein Operator der Programmiersprache C, der den Wert einer Größe um 1 erhöht. Insofern spiegelt der Name die Eigenschaft »Nachfolger von C«. 1998 wurde C++ erstmals von der ISO (International Organization for Standardization) und der IEC (International Electrotechnical Commission) standardisiert. Diesem Standard haben sich nationale Standardisierungsgremien wie ANSI (USA) und DIN (Deutschland) angeschlossen. Die Anforderungen an C++ sind gewachsen, auch zeigte sich, dass manches fehlte und anderes überflüssig oder fehlerhaft war. Das C++-Standardkomitee hat kontinuierlich an der Verbesserung von C++ gearbeitet. Seit 2011 werden im Abstand von drei

Jahren neue Versionen des Standards herausgegeben. Die Kurznamen sind entsprechend den Jahreszahlen C++11, C++14, C++17, C++20 und C++23. C++23 wurde von der zuständigen ISO/IEC-Arbeitsgruppe JTC1/SC22/WG21 verabschiedet und bei der ISO zur Veröffentlichung eingereicht.

1.2 Arten der Programmierung

Es gibt viele verschiedene Arten der Programmierung. C++ unterstützt im Wesentlichen die folgenden:

Imperative Programmierung

Dabei werden die im Programmcode festgelegten Schritte der Reihe nach ausgeführt. Es geht also darum, einem Rechner mitzuteilen, *was* er tun soll und *wie* es zu tun ist. Ein Programm ist ein in einer Programmiersprache formulierter Algorithmus oder anders ausgedrückt eine Folge von Anweisungen, die der Reihe nach auszuführen sind, ähnlich einem Kochrezept. Der Algorithmus wird in einer besonderen Sprache, die der Rechner »versteht«, geschrieben. Der Schwerpunkt dieser Betrachtungsweise liegt auf den einzelnen Schritten oder Anweisungen an den Rechner, die zur Lösung einer Aufgabe abzuarbeiten sind. Mit sogenannten Kontrollstrukturen wird der Programmablauf gesteuert. So können Teile wiederholt ausgeführt oder übersprungen werden.

Objektorientierte Programmierung

Bei der imperativen Programmierung wird ein Aspekt eher stiefmütterlich behandelt: Der Rechner muss »wissen«, *womit* er etwas tun soll. Zum Beispiel soll er

- eine bestimmte Summe Geld von einem Konto auf ein anderes transferieren;
- eine Ampelanlage steuern;
- ein Rechteck auf dem Bildschirm zeichnen.

Häufig, wie in den ersten beiden Fällen, werden Objekte der realen Welt (Konten, Ampelanlage ...) *simuliert*, das heißt im Rechner abgebildet. Die abgebildeten Objekte haben eine *Identität*. Das *Was* und das *Womit* gehören stets zusammen. Beide sind also Eigenschaften eines Objekts und sollen daher nicht getrennt werden. Ein Konto kann schließlich nicht auf Gelb geschaltet werden und eine Überweisung an eine Ampel ist nicht vorstellbar. Ein *objektorientiertes Programm* kann man sich als Abbildung von Objekten der realen Welt in Software vorstellen. Die Abbildungen werden in C++ selbst wieder Objekte genannt. *Klassen* sind Beschreibungen von *Objekten*.

Generische Programmierung

Die generische Programmierung ermöglicht es, Klassen und Algorithmen für verschiedene Datentypen mit nur einem Schema zu modellieren. Das gilt für die imperative Programmierung ebenso wie für die objektorientierte Programmierung. Andere Arten der Programmierung sind für C++ von geringerer Bedeutung.