

Vorwort

Dieser korrigierte Nachdruck der 4. Auflage enthält Korrekturen und kleinere Ergänzungen. Die 4. Auflage unterscheidet sich von der 3. Auflage durch die vollständige Umstellung auf den 2014 verabschiedeten ISO C++-Standard. Um die Seitenzahl beizubehalten, wurden einige Kapitel neu strukturiert und zusammengefasst, und die Lösungen zu den Programmierübungsaufgaben sind nicht mehr abgedruckt, sondern auf der Internetseite zum Buch zu finden. Die Veränderungen gegenüber dem C++-Standard von 2011 sind durch Markierungen am Seitenrand (wie hier) gekennzeichnet. Das Buch ist konform zum neuen C++-Standard, ohne den Anspruch auf Vollständigkeit zu erheben – das Standard-Dokument [ISOC++] umfasst allein schon mehr als 1300 Seiten. Sie finden in diesem Buch eine verständliche und mit vielen Beispielen angereicherte Einführung in die Sprache. Im Teil »Das C++ Rezeptbuch« gibt es zahlreiche Tipps und Lösungen für typische Aufgaben, die in der täglichen Praxis anfallen. Es gibt konkrete, sofort umsetzbare Lösungsvorschläge. Zahlreiche Algorithmen für praxisnahe Problemstellungen helfen bei der täglichen Arbeit. Auf größtmögliche Portabilität wird geachtet: Die Beispiele funktionieren unter Linux genau so wie unter Windows. Die problembezogene Orientierung lässt die in die Sprache einführenden Teile kürzer werden. Damit wird das Lernen erleichtert, und die Qualität des Buchs, auch als Nachschlagewerk dienen zu können, bleibt erhalten. C++14

Für wen ist dieses Buch geschrieben?

Dieses Buch ist für alle geschrieben, die einen kompakten und gleichzeitig gründlichen Einstieg in die Konzepte und Programmierung mit C++ suchen. Es ist für Anfänger¹ gedacht, die noch keine Programmiererfahrung haben, aber auch für andere, die diese Programmiersprache kennen lernen möchten. Beiden Gruppen und auch C++-Erfahrenen dient das Buch als ausführliches Nachschlagewerk.

Die ersten 10 Kapitel führen in die Sprache ein. Es wird sehr schnell ein Verständnis des objektorientierten Ansatzes entwickelt. Die sofortige praktische Umsetzung des Gelernten steht im Vordergrund. C++ wird als Programmiersprache unabhängig von speziellen

¹ Geschlechtsbezogene Formen meinen hier und im Folgenden stets Männer, Frauen und andere.

Produkten beschrieben; C-Kenntnisse werden nicht vorausgesetzt. Das Buch eignet sich zum Selbststudium und als Begleitbuch zu einer Vorlesung oder zu Kursen. Die vielen Beispiele sind leicht nachzuvollziehen und praxisnah umsetzbar. Klassen und Objekte, Templates und Exceptions sind Ihnen bald keine Fremdworte mehr. Es gibt mehr als 90 Übungsaufgaben – mit Musterlösungen im Anhang. Durch das Studium dieser Kapitel werden aus Anfängern bald Fortgeschrittene.

Diesen und anderen Fortgeschrittenen und Profis bietet das Buch kurze Einführungen in die Themen Thread-Programmierung, Netzwerk-Programmierung mit Sockets einschließlich eines kleinen Webservers, Datenbank-Anbindung, grafische Benutzeroberflächen und mehr. Dabei wird durch Einsatz der Boost-Library und des Qt-Frameworks größtmögliche Portabilität erreicht.

Softwareentwicklung ist nicht nur Programmierung: Einführend werden anhand von Beispielen unter anderem die Automatisierung der Programmierzeugung mit Make, die Dokumentationserstellung mit Doxygen und die Versionskontrolle behandelt. Das Programmdesign wird durch konkrete Umsetzungen von UML²-Mustern nach C++ unterstützt. Das integrierte »C++-Rezeptbuch« mit mehr als 150 praktischen Lösungen, das sehr 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

Beim Programmieren geht es nicht nur darum, eine Programmiersprache zu lernen. Es geht auch darum, Programme zu schreiben, die hohen Qualitätsansprüchen gerecht werden. Dazu gehört das Knowhow, die Mittel der Sprache 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. In diesem Sinn liegt ein Schwerpunkt des Buchs auf der Methodik des Programmierens, unterstützt durch Darstellung der informatischen Grundlagen, viele Hinweise im Text und demonstriert an vielen Beispielen.

Übersicht

Schwerpunkt von Teil I ist die Einführung in die Programmiersprache. Die anschließenden Teile gehen darüber hinaus und konzentrieren sich auf die verschiedenen Probleme der täglichen Praxis. Die in dieser Übersicht verwendeten Begriffe mögen Ihnen zum Teil noch unbekannt sein – in den betreffenden Kapiteln werden sie ausführlich erläutert.

Teil I - Einführung in C++

Das *Kapitel 1* vermittelt zunächst die Grundlagen, wie ein Programm geschrieben und zum Laufen gebracht wird. Es folgen einfache Datentypen und Anweisungen zur Kontrolle des Programmablaufs. Die Einführung der C++-Datentypen `vector` und `string` und die einfachen Ein- und Ausgabe beenden das Kapitel.

Das *Kapitel 2* zeigt Ihnen, wie Sie Funktionen schreiben. Makros, Templates für Funktionen und die modulare Gestaltung von Programmen folgen.

² UML ist eine grafische Beschreibungssprache für die objektorientierte Programmierung.

Objektorientierung ist der Schwerpunkt von *Kapitel 3*. Dabei geht es nicht nur um die Konstruktion von Objekten, sondern auch um den Weg von der Problemstellung zu Klassen und Objekten. Zeiger sowie die Erzeugung von Objekten zur Laufzeit sind Inhalt von *Kapitel 4*. *Kapitel 5* führt das Thema Objektorientierung fort. Dabei lernen Sie kennen, wie eine String-Klasse funktioniert, und wie Sie Klassen-Templates und Templates mit einer variablen Anzahl von Parametern konstruieren. Das *Kapitel 6* zeigt Ihnen das Mittel objektorientierter Sprachen, um Generalisierungs- und Spezialisierungsbeziehungen auszudrücken: die Vererbung mit ihren Möglichkeiten. Strategien zur Fehlerbehandlung mit Exceptions finden Sie in *Kapitel 7*. Das *Kapitel 8* zeigt, wie Sie Operatorsymbolen wie + und - eigene Bedeutungen zuweisen können und in welchem Zusammenhang das sinnvoll ist. Sie lernen, »intelligente« Zeiger (Smart Pointer) zu konstruieren und Objekte als Funktionen einzusetzen. *Kapitel 9* beschreibt ausführlich die Ein- und Ausgabemöglichkeiten, die vorher nur einführend gestreift werden, einschließlich der Fehlerbehandlung und der Formatierung der Ausgabe. Eine Einführung in die Standard Template Library (STL) bietet *Kapitel 10*. Die STL und ihre Wirkungsweise bilden die Grundlage eines sehr großen Teils der C++-Standardbibliothek.

Teil II - Fortgeschrittene Themen

In diesem Teil folgen fortgeschrittene Themen wie Lambda-Funktionen (*Kapitel 11*), Wert- und Referenzsemantik (*Kapitel 12*), reguläre Ausdrücke (*Kapitel 13*) und die Programmierung paralleler Abläufe mit Threads (*Kapitel 14*). Das *Kapitel 15* zeigt, wie grafische Benutzungsschnittstellen konstruiert werden. Wie ein Programm die Verbindung mit dem Internet aufnehmen kann, dokumentiert das *Kapitel 16*. Und wohin mit den ganzen Daten, die bei Programmende nicht verloren gehen sollen? In *Kapitel 17* lernen Sie, wie ein Programm an eine Datenbank angebunden wird. Manche der genannten Themen sind so umfangreich, dass sie selbst Bücher füllen. Für diese Themen wird daher nur ein Einstieg geboten.

Teil III - Praktische Methoden und Werkzeuge der Softwareentwicklung

Die Entwicklung von Programmen besteht nicht nur im Schreiben von Code. Die Compilation eines Projekts mit vielen Programmdateien und Abhängigkeiten kann schnell ein komplexer Vorgang werden. Die Automatisierung dieses Prozesses mit dem Tool *make* ist Thema von *Kapitel 18*. Programme sind nicht auf Anhieb fehlerfrei. Sie müssen getestet werden: *Kapitel 19* stellt ein Werkzeug für den Unit-Test vor und zeigt den praktischen Einsatz. *Kapitel 20* demonstriert ein Werkzeug zur automatischen Dokumentationserstellung und geht kurz auf die Versionsverwaltung und die Projektverwaltung ein.

Teil IV - Das C++-Rezeptbuch: Tipps und Lösungen für typische Aufgaben

Sichere Programmentwicklung ist die Überschrift des *Kapitels 21*. Sie finden dort Regeln zum Design von Methoden und Tipps zur defensiven Programmierung, die die Wahrscheinlichkeit von Fehlern vermindern. *Kapitel 22* zeigt Rezepte, wie Sie bestimmte UML-Muster in C++-Konstruktionen umwandeln können. Algorithmen für viele verschiedene Aufgaben finden Sie in *Kapitel 23*. Wegen der Vielzahl empfiehlt sich ein Blick in das Inhaltsverzeichnis, um einen Überblick zu gewinnen. Der C++-Standard bietet für viele

Datei- und Verzeichnisoperationen keine Unterstützung an. *Kapitel 24* enthält deshalb fertige Rezepte zum Anlegen, Löschen und Lesen von Verzeichnissen und mehr.

Teil V - Die C++-Standardbibliothek

Hier wird die C++-Standardbibliothek in Kürze beschrieben. Die Inhalte dieses Teils sind: Hilfsfunktionen und -klassen, Container, Iteratoren, Algorithmen, Einstellung nationaler Besonderheiten, String, Speicherverwaltung, Funktionen der Programmiersprache C.

Anhang

Der Anhang enthält unter anderem verschiedene hilfreiche Tabellen und die Lösungen der Übungsaufgaben.

■ 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* ab Seite 955 aufgeführt.

Es gibt ein recht umfangreiches *Stichwortverzeichnis* ab Seite 969 und ein sehr detailliertes *Inhaltsverzeichnis*.



Software zum Buch

Auf der Webseite <http://www.cppbuch.de/> finden Sie die Software zu diesem Buch. Sie enthält unter anderem einen Compiler, eine Integrierte Entwicklungsumgebung sowie alle Programmbeispiele und die Lösungen zu den Aufgaben. Sie finden dort auch weitere Hinweise, Errata und nützliche 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 von der Universität Leipzig. Frau Brigitte Bauer-Schiewek und Frau Irene Weilhart vom Hanser Verlag danke ich für die sehr gute Zusammenarbeit.

Bremen, im August 2016

Ulrich Breymann