

Vorwort zur dritten Auflage

Programmierbücher haben meist nur eine begrenzte »Haltbarkeit«. Sie beschreiben eine Technologie oder ein Werkzeug, die schnell wieder durch andere ersetzt werden. In diesem Fall trifft das (zum Glück) nur zum Teil zu. Die Programmiersprache C++ blieb seit der letzten Auflage weitgehend gleich und selbst der GCC-Compiler änderte sich nur in Teilaspekten, die bereits über die in einem solchen Einsteiger-Buch sinnvolle Detailtiefe hinausgehen.

Andere Werkzeuge haben dagegen in der Zwischenzeit eine stärkere Veränderung durchgemacht. Daher war im hinteren Teil des Buches mancherlei Anpassung nötig. Zudem wurde das Buch deutlich schlanker und ist damit nun hoffentlich auch als Lehr- und Lernbuch besser geeignet. Dazu soll auch die deutlich größere Zahl an Übungsaufgaben beitragen.

Einige zu umfangreiche Beispiele wurden gestrafft, andere ergänzt oder aktualisiert. Der Abschnitt über den *DDD*-Debugger entfiel ebenso wie der zu *RCS* – beide Werkzeuge sind zwar nach wie vor erhältlich und werden eingesetzt, haben aber nicht mehr die einstmalige Bedeutung. Das Kapitel über Editoren ist ebenfalls deutlich kürzer geworden, da die neuen Versionen der Editoren so selbsterklärend sind, dass Sie dafür wohl nicht extra ein Buch wie dieses benötigen.

Dafür wurde der Abschnitt zu Eclipse um einiges größer. Mittlerweile hat Eclipse in der Version 3 einen Grad der Reife erreicht, bei dem ich Ihnen dieses Werkzeug guten Gewissens empfehlen kann – auch für die C++-Entwicklung. Im Gegenzug ist der Abschnitt zu *SNiFF+* ganz herausgefallen, da es nun mit *KDevelop* und Eclipse zwei hervorragende freie Entwicklungsgebungen gibt, die es nicht mehr nötig machen, auch eine weitere kommerzielle ausführlich zu behandeln.

Über diese Ergänzungen hinaus habe ich das ganze Manuskript einer kritischen Durchsicht unterzogen und an vielen Stellen die Erläuterungen leicht geändert. Ich hoffe, ich konnte damit die Verständlichkeit des zugegebenermaßen nicht ganz einfachen Inhalts noch etwas verbessern.

Und noch eine Neuerung: Dieser Auflage liegt als erster keine CD-ROM mehr bei. Sie können alle Beispiele und auch die Lösungen zu den Übungsaufgaben unter der bekannten Adresse

<http://www.cpp-entwicklung.de>

herunterladen. Dort gebe ich Ihnen auch Links zu der Software an, die hier besprochen wird und die bislang teilweise auf der CD-ROM enthalten war. Außerdem finden Sie dort weiterhin den gesamten Text des Buches in HTML-Form zum Nachschlagen und als Online-Referenz.

Ich danke allen Lesern, die mir mit ihren Zuschriften viel positives Feedback gegeben und mich auf die eine oder andere Ungereimtheit hingewiesen haben. Ich würde mich freuen, auch weiterhin Anmerkungen und Eindrücke von Ihnen zu erfahren, am besten per E-Mail unter *info@cpp-entwicklung.de*.

Vorwort zur ersten Auflage

Die Programmiersprache C++ erfreut sich immer größerer Beliebtheit und Verbreitung. Warum ist das so? Mittlerweile hat sich die Erkenntnis durchgesetzt, dass Wartbarkeit und Wiederverwendbarkeit von Software – zwei Kriterien von wesentlicher Bedeutung – am ehesten durch eine objektorientierte Programmierung erreicht werden können. Und da die Programmiersprache C eine der im technischen Bereich am weitesten verbreiteten ist und C++ deren objektorientierte Erweiterung darstellt, wird die Beliebtheit langsam verständlich.

Kaum ein Entwickler, der technische Software erstellen soll, kommt ohne Kenntnisse in C++ aus. An sehr vielen Arbeitsplätzen dominieren für die Programmierung aber noch die Entwicklungsumgebungen unter Windows, die mit einer mächtigen grafischen Oberfläche ausgestattet sind. Doch werden in jüngster Zeit viele Firmen angesichts der Marktmacht des Windows-Herstellers Microsoft immer skeptischer, ob die Abhängigkeit von einem Anbieter auf Dauer sinnvoll ist, und suchen nach Alternativen. So wird das allgemeine Interesse an Linux immer größer. Handelt es sich dabei doch um ein Betriebssystem, das völlig offen und so gut wie kostenlos erhältlich ist – und sich zudem noch als äußerst stabil und schnell erweist. Außerdem läuft es auf derselben preisgünstigen PC-Hardware wie Windows auch.

Viele Neulinge, aber auch eine Reihe von erfahreneren Entwicklern aus der Windows-Welt, sind angesichts dieses Trends etwas verunsichert. Bei ihnen gilt das Programmieren unter Linux (wie allgemein unter Unix) als unbequem und »steinzeitlich«. In diesem Buch will ich Ihnen zeigen, dass diese Vorurteile heute nicht mehr berechtigt sind. Es lässt sich nämlich in der Tat unter Linux ähnlich bequem entwickeln wie unter Windows – wenn auch ein wenig anders, als Sie es möglicherweise bisher kennen. Dafür bekommen Sie aber fast alles, was Sie brauchen, so gut wie umsonst.

Dieses Buch ist für alle,

- ❑ die mit C++ programmieren wollen (oder müssen ...) und dafür eine kompakte und praxisnahe Einführung suchen,
- ❑ die schon immer einmal wissen wollten, wie man eigentlich mit C++ Programme schreibt, die sich aber dafür keine teuren Entwicklungspakete anschaffen wollten,
- ❑ die schon unter Windows oder Macintosh programmiert haben und nun die Programmentwicklung unter Linux kennen lernen möchten,
- ❑ die schon mit anderen Unix-Varianten Erfahrung haben und nun mehr über die modernen Werkzeuge, die unter Linux zur Verfügung stehen, erfahren wollen.

Ziele und Besonderheiten des Buches

*Programmiersprache +
Werkzeuge*

Das Besondere an diesem Buch ist, dass Sie hier nicht nur etwas über die Programmiersprache C++, sondern auch viel über die Werkzeuge lernen, die für die Entwicklung unter Linux zur Verfügung stehen. Der Schwerpunkt der vorgestellten Werkzeuge liegt auf frei erhältlicher Software. Diese ist fast immer nicht nur unter Linux, sondern auch unter den meisten anderen Unix-Plattformen, ja sogar oft unter Windows erhältlich, so dass das Buch für die Benutzer dieser Betriebssysteme ebenso interessant sein kann.

*Konzentration auf die
objektorientierte
Sichtweise*

Die Einführung in die C++-Programmiersprache konzentriert sich in diesem Buch auf die *objektorientierte* Sichtweise. Im Gegensatz zum traditionellen Ansatz vieler Lehrbücher wird daher nicht mit den C-Altlasten begonnen (außer einigen grundlegenden Typen und Operatoren), sondern mit dem Klassenkonzept, d.h. mit Attributen und Methoden, Ableitung und Polymorphismus. Das verlangt von Ihnen sicher einiges an Durchhaltevermögen – wird Sie aber letztlich sehr viel weiter bringen! Denn Objektorientierung ist vor allem eine *andere Art des problemlösenden Denkens*, als es etwa von den prozeduralen Sprachen verlangt wird.

*Integrierte Entwick-
lungsumgebungen*

Zunächst aber erfahren Sie konkret, wie Sie unter Linux entwickeln und welche Werkzeuge Sie dabei tatkräftig unterstützen. In diesem Zusammenhang werden Sie nicht nur die grundlegenden Programme wie Editor und Compiler (einschließlich deren wichtigsten Optionen) kennen lernen, sondern auch moderneren integrierten Entwicklungsumgebungen begegnen, durch die Sie bei der Arbeit produktiver werden kön-

nen und die Ihnen den Einstieg in die Programmierung (bzw. den Umstieg von Windows) erleichtern sollen.

C++ ist eine sehr mächtige Programmiersprache, die auch den erfahrenen Programmierer immer wieder unsicher machen kann. Alle Details und Probleme können leider im Rahmen dieses Buches nicht behandelt werden. Wenn Sie sich also in der Tiefe mit C++ auseinandersetzen wollen, sollten Sie im Anschluss zu einem der vielen guten Fortgeschrittenen-Bücher zu diesem Thema greifen. Die Grundbegriffe und die Entwicklungswerkzeuge kennen Sie ja dann bereits.

Voraussetzungen

Auch wenn Sie noch keine Erfahrung im Programmieren haben, können Sie dieses Buch verwenden. Es ist allerdings hilfreich, wenn Sie bereits Programme geschrieben haben – am besten natürlich in C, da C++ ja darauf aufbaut. Aber auch Kenntnisse in Pascal, Java oder Fortran helfen Ihnen sicher über einige Verständnisprobleme hinweg.

Programmiererfahrung?

Obschon sich dieses Buch auf die Entwicklungsarbeit unter Linux bezieht, so kann es doch keine Einführung in Linux selbst sein. Ich setze also voraus, dass Sie Linux bereits erfolgreich auf Ihrem Rechner installiert haben und wissen, was eine Shell ist, wie man Verzeichnisse anlegt, Dateien kopiert, verschiebt und löscht und wie Sie sich in einer grafischen Benutzeroberfläche zurechtfinden, die auf dem X-Window-System basiert.

Grundkenntnisse über den Umgang mit Linux

Die Abbildungen von Programm- und Dialogfenstern in diesem Buch sind unter der Oberfläche KDE erstellt worden. Das heißt allerdings nicht, dass Sie unbedingt KDE auf Ihrem Rechner benötigen. Das meiste wird auch mit anderen Window-Managern funktionieren.

Organisation des Buchs

Der erste Teil des Buchs ist der Programmiersprache C++ gewidmet. Dabei starten wir mit einem allgemeinen Überblick, was das Programmieren unter Unix von dem unter anderen Plattformen unterscheidet.

Programmieren unter Unix

Dann geht es im 2. Kapitel direkt in die Tiefen der objektorientierten Denkweise. An dieser Stelle mag Ihnen der Text zuweilen ein wenig trocken vorkommen; das liegt daran, dass Ihnen zu diesem Zeitpunkt noch zu viele Sprachmerkmale fehlen, um sinnvolle Beispiele erstellen zu können. Andererseits führt Sie das Kapitel aber durch alle wichtigen OO-Aspekte, so dass Sie anschließend Ihre Programme gut danach ent-

Objektorientierung

werfen können. Außerdem sehen wir uns in diesem Kapitel noch den GNU-C++-Compiler GCC und seine Bedienung an.

*Programmieren mit
C++*

Kapitel 3 ist dem praktischen Programmieren mit C++ gewidmet. Nachdem Sie die theoretischen Konzepte nun schon beherrschen, können wir uns an einige richtige Aufgaben machen. Als Beispiele erstellen wir Programme, die mit Linux-Systembefehlen arbeiten, z.B. die Uhrzeitbestimmung oder die Signalbehandlung,

*Fortgeschrittene
Sprachmerkmale*

Anschließend befassen wir uns in Kapitel 4 mit den fortgeschrittenen Sprachmerkmalen wie Templates, der Standard Template Library STL, Operatorüberladung und Ausnahmebehandlung. Hier werden Sie einige Funktionsmerkmale aus dem neuesten ANSI/ISO-Standard kennen lernen, die Sie vielleicht anfangs noch nicht so oft in Ihren eigenen Programmen brauchen werden, die aber zum Verständnis anderer Programme notwendig sind.

Editoren

Im zweiten Teil geht es um die konkrete Arbeit mit Linux. Als Erstes stelle ich Ihnen in Kapitel 5 einige Editoren vor, mit deren Hilfe Sie Ihre Programme schreiben können. Ob Sie schließlich *vi*, *XEmacs*, *KEdit*, *NEdit*, *XCoral* oder etwas ganz anderes einsetzen, bleibt Ihrem Geschmack überlassen.

Programmierwerkzeuge

Danach geht es im 6. Kapitel um verschiedene Werkzeuge, die Sie zum Programmieren neben dem Compiler auch benötigen werden. *make* dient zur Verwaltung von Quelltexten in mehreren Dateien, der Debugger *gdb* zur Fehlersuche und *CVS* zur Versionsverwaltung.

*Integrierte Entwick-
lungsumgebungen*

Abschließend lernen Sie in Kapitel 7 verschiedene integrierte Entwicklungsumgebungen kennen, vor allem das sehr komfortable *KDevelop*, das als Open-Source-Anwendung frei erhältlich ist.

Am Ende des Buches finden Sie noch die zitierten Literaturstellen sowie einige zusätzliche lesenswerte und weiterführende Werke.

Typografische Konventionen

In diesem Buch werden außer dem Fließtext folgende Aspekte durch Formatierungen hervorgehoben:

- *Kursive Schreibweise* dient zur Hervorhebung von Begriffen.
- Serifenlose Schrift steht für Namen von Dateien und Pfaden.
- *Kursive, serifenlose Schrift* weist auf ausführbare Programme hin.
- Programmelemente wie Klassen, Variablen, Schlüsselwörter etc. sind in Schreibmaschinenschrift gesetzt.
- Eingaben, die Sie beispielsweise in der Shell vornehmen müssen, sind ebenfalls in Schreibmaschinenschrift, aber durch Fettdruck hervorgehoben. Das Zeichen »%« steht für die Eingabeaufforde-

rung. Falls Sie einen Befehl als Benutzer *root* ausführen müssen, bezeichnet »#« die Eingabeaufforderung.

- Wenn Sie in Anwendungsprogrammen für einen Befehl eine oder mehrere Tasten drücken müssen, so sind diese in den Erläuterungen durch abgerundete Rahmen gekennzeichnet. Ein Pluszeichen zwischen zwei dieser Rahmen steht dabei für gleichzeitig zu drückende Tasten (z.B. Strg+v), ein Komma für nacheinander zu drückende (beispielsweise :, q, !). Der Pfeil nach oben ↑ ist die Umschalttaste.
- In Kleindruck sind Zusatz- und Hintergrundinformationen gesetzt, die Sie beim ersten Durcharbeiten nicht benötigen, die jedoch hilfreich für das tiefere Verständnis sind.

Begriffe wie »der Programmierer« sind nur der Einfachheit halber in männlicher Form gehalten. Gemeint ist damit immer eine männliche oder weibliche Person.

Aktuelle Ergänzungen

Für dieses Buch gibt es eine Seite im World Wide Web, auf der Sie den gesamten Text sowie Korrekturen, Ergänzungen und Verweise finden. Schlagen Sie nach unter:

<http://www.cpp-entwicklung.de>

Danksagungen

Wenn man ein Buch wie dieses schreibt, stellt man die Geduld vieler Menschen auf eine harte Probe. Das Feilen und Überarbeiten, um zu einem bestmöglichen Ergebnis für Sie zu kommen, nimmt geraume Zeit in Anspruch. Ich möchte an dieser Stelle dem dpunkt.verlag, insbesondere Herrn Schönfeldt, für die ausgesprochen gute Betreuung in dieser Zeit und darüber hinaus herzlich danken. Viele kompetente Verbesserungsvorschläge kamen von ihm.

Natürlich geht mein Dank auch an meine Frau Andrea, die nun hoffentlich wieder mehr Abende und Wochenenden mit mir wird verbringen können.