

C++

Xpert.press

Richard Kaiser

C++ mit Microsoft Visual C++ 2008



mit CD-ROM

Standard-C++ • C++/CLI • Windows .NET-Programmierung

 Springer

Richard Kaiser

C++ mit Microsoft Visual C++ 2008

Einführung in Standard-C++, C++/CLI und die
objektorientierte Windows .NET-Programmierung

Mit CD-ROM



Springer

Geleitwort

Wenn man heute nach Literatur über Programmiersprachen sucht, so findet man viel Neues über Sprachen wie C# und Java, die von einer virtuellen Maschine ausgeführt werden, aber auch über die dynamischen Sprachen wie Python, Ruby und PHP. Es könnte der Eindruck entstehen, dass Softwareentwicklung im Wesentlichen mit diesen Sprachen stattfindet. Dies ist aber nicht der Fall. Ein nicht zu unterschätzender Teil der professionellen Softwareentwicklung wird auf der Basis der Programmiersprache C++ durchgeführt. In meiner beruflichen Praxis, in der ich Unternehmen betreue, die mit den Softwareentwicklungswerkzeugen von Microsoft arbeiten, begegne ich häufig der Programmiersprache C++.

C++ ist auch heute immer noch die erste Wahl, wenn es darum geht, hocheffiziente Software für den technisch-wissenschaftlichen Bereich oder hardwarenahe Aufgaben zu entwickeln. Man darf auch nicht vergessen, dass für C++ umfangreiche, qualitativ hochwertige und plattformunabhängige Klassenbibliotheken für viele Anwendungsdomänen existieren. Ein anderer Vorteil von C++ ist meiner Ansicht nach, dass die Sprache sowohl das prozedurale als auch das objektorientierte Entwicklungsparadigma unterstützt. Für Microsoft spielt C++ weiterhin eine wichtige Rolle bei der Softwareentwicklung im eigenen Haus, aber auch in der Weiterentwicklung der Werkzeuge für die Programmierung mit C++, um diese noch produktiver und sicherer zu machen. Microsoft arbeitet aktiv im ISO Komitee zur Standardisierung von C++ mit und hat mit dem Standard ECMA-372 (C++/CLI) C++ für den Einsatz auf der .NET Laufzeitumgebung erweitert. Außerdem ist die einfache Integration von nativen C++ Klassenbibliotheken mit Programmen, die für Microsoft .NET entwickelt werden bzw. wurden, eine nicht zu unterschätzende Eigenschaft der .NET Laufzeitumgebung, einmal unter dem Gesichtspunkt des Investitionsschutzes, als auch unter dem Aspekt der Laufzeiteffizienz.

Das vorliegende Buch von Richard Kaiser, der selbst aktiv im DIN an der Standardisierung von C++ mitarbeitet, gibt eine umfassende Einführung in alle Aspekte von Standard C++, sowie in die Spezialitäten von C++ auf der Microsoft Windows Plattform und C++/CLI für Microsoft .NET. Die Kapitel 1 und 2 bieten eine sehr gute Einführung in das Arbeiten mit Microsoft Visual Studio 2008. Sehr positiv ist noch, dass die prozeduralen und objektorientierten Eigenschaften von C++ explizit dargestellt werden, und dass in Kapitel 3.7 eine Einführung in die Techniken und Möglichkeiten der Programmverifikation im Kontext von C++

gegeben wird. Diese Techniken werden an Bedeutung gewinnen, um die Qualität und Sicherheit von Software besser zu gewährleisten.

Ich wünsche dem Buch viel Erfolg, denn es bietet eine fundierte Einführung in alle Aspekte der Softwareentwicklung mit C++ auf der Microsoft Windows Plattform, einschließlich dem .NET Framework, und es trägt dazu bei, den Einstieg in diese mächtige Programmiersprache zu erleichtern.

Klaus Rohe, Platform Strategy Manager, Developer Platform & Strategy Group, Microsoft Deutschland GmbH

Vorwort

Dieses Buch entstand aus dem Wunsch, in meinen Vorlesungen über C++ nicht nur Textfensterprogramme (Konsolenanwendungen), sondern von Anfang an Windows-Programme zu entwickeln. Dafür ist Visual Studio 2008 sehr gut geeignet. Es ist so einfach zu bedienen, dass man es auch in Anfängervorlesungen einsetzen kann, ohne dabei Gefahr zu laufen, dass die Studenten nur noch mit dem Entwicklungssystem kämpfen und gar nicht mehr zum Programmieren kommen.

Als ich dann die Möglichkeit bekam, Kurse für erfahrene Praktiker aus der Industrie zu halten, wurde ich mit einer Fülle von Anregungen aus ihrer täglichen Arbeit konfrontiert. Diese gaben dem Buch viele praxisorientierte Impulse.

Dieses Buch besteht aus drei Teilen:

- Teil 1 stellt die Entwicklungsumgebung Visual Studio 2008 vor und zeigt, wie man Windows-Programme mit den wichtigsten Steuerelementen entwickelt.
- Teil 2 stellt den gesamten Sprachumfang des C++-Standards umfassend vor. Dazu gehören nicht nur die Sprachelemente von C sowie Klassen, Templates und Exception-Handling, sondern auch die C++-Standardbibliothek.
- Teil 3 behandelt den gesamten C++/CLI-Standard und gibt einen Einblick in die .NET Klassenbibliothek. Die C++/CLI-Erweiterungen von Standard-C++ sind die Grundlage für die Integration der .NET-Programmierung in C++. C++/CLI ermöglicht insbesondere, bestehende C++-Programme in Windows .NET-Anwendungen aufzunehmen.

Diese drei Teile lassen sich aber nicht streng trennen: Da alle Beispiele und Übungsaufgaben Windows .NET-Programme sind, werden in Teil 2 einige C++/CLI- und .NET Elemente verwendet.

Die Programmiersprache C++ wurde als Obermenge der Programmiersprache C entworfen. Dieser Entscheidung verdankt C++ sicher seine weite Verbreitung. Sie hat aber auch dazu geführt, dass oft weiterhin wie in C programmiert wird und lediglich ein C++-Compiler anstelle eines C-Compilers verwendet wird. Dabei werden viele Vorteile von C++ verschenkt. Um nur einige zu nennen:

- In C++ werden die fehleranfälligen Zeiger viel seltener als in C benötigt.

- Die Stringklassen lassen sich wesentlich einfacher und risikoloser als die nullterminierten Strings von C verwenden.
- Die Containerklassen der C++-Standardbibliothek haben viele Vorteile gegenüber Arrays, selbstdefinierten verketteten Listen oder Bäumen.
- Exception-Handling bietet eine einfache Möglichkeit, auf Fehler zu reagieren.
- Objektorientierte Programmierung ermöglicht übersichtlichere Programme.
- Templates sind die Basis für eine außerordentlich vielseitige Standardbibliothek.

Ich habe versucht, bei allen Konzepten nicht nur die Sprachelemente und ihre Syntax zu beschreiben, sondern auch Kriterien dafür anzugeben, wann und wie man sie sinnvoll einsetzen kann. Deshalb wurde z.B. mit der objektorientierten Programmierung eine Einführung in die objektorientierte Analyse und das objektorientierte Design verbunden. Ohne die Beachtung von Design-Regeln schreibt man leicht Klassen, die der Compiler zwar übersetzen kann, die aber kaum hilfreich sind.

Man hört immer wieder die Meinung, dass C++ viel zu schwierig ist, um es als einführende Programmiersprache einzusetzen. Dieses Buch soll ein in vielen Jahren erprobtes Gegenargument zu dieser Meinung sein. Damit will ich aber die Komplexität von C++ überhaupt nicht abstreiten.

Zahlreiche Übungsaufgaben geben dem Leser die Möglichkeit, die Inhalte praktisch anzuwenden und so zu vertiefen. Da man Programmieren nur lernt, indem man es tut, möchte ich ausdrücklich dazu ermuntern, zumindest einen Teil der Aufgaben zu lösen und sich dann selbst neue Aufgaben zu stellen. Der Schwierigkeitsgrad der Aufgaben reicht von einfachen Wiederholungen des Textes bis zu kleinen Projektchen, die ein gewisses Maß an selbständiger Arbeit erfordern. Die Lösungen der meisten Aufgaben findet man auf der beiliegenden CD und auf meiner Internetseite <http://www.rkaiser.de>.

Anregungen, Korrekturhinweise und Verbesserungsvorschläge sind willkommen. Bitte senden Sie diese an die Mail-Adresse auf meiner Internetseite.

Bei meinen Schulungskunden und Studenten bedanke ich mich für die zahlreichen Anregungen und anregenden Fragen. Herrn Engesser und seinem Team vom Springer-Verlag danke ich für die Unterstützung und Geduld.

Tübingen, im April 2009

Richard Kaiser