Stefan Kuhlins · Martin Schader

Die C++-Standardbibliothek

Einführung und Nachschlagewerk

Vierte, durchgesehene Auflage

Mit 77 Abbildungen und 37 Tabellen

Springer

Vorwort

Warum wird eine Standardbibliothek für C++ gebraucht? Programmentwickler benötigen immer wieder die gleichen Datenstrukturen wie z. B. dynamische Felder. Aufgrund verschiedener konkreter Anforderungen werden einzelne Implementierungen aber in der Regel sehr unterschiedlich ausfallen. Dadurch werden Programme anderer Entwickler schwer verständlich, und beim Wechsel der eingesetzten Bibliotheken entsteht erheblicher Lernaufwand. Durch eine Standardbibliothek lassen sich diese Probleme in den Griff bekommen. Um sich an die verschiedenen Anforderungen, unter denen eine Bibliothek zum Einsatz kommt, anpassen zu können, ist eine wichtige Voraussetzung für den Erfolg einer Standardbibliothek, dass sie flexibel und erweiterbar ist. Mit der *Standard Template Library* (STL), dem Kernstück der aktuellen C++-Standardbibliothek, wird dieses Ziel erreicht.

Darüber hinaus ist die STL äußerst effizient, so dass sich eine hervorragende Performance erreichen lässt. Allerdings setzt ein Gewinn bringender Einsatz der STL ein tiefes Verständnis für das Design der STL voraus. Um die STL zu verstehen, genügt es nicht, lediglich eine Funktionsreferenz zu lesen. Deshalb haben wir in diesem Buch besonderen Wert darauf gelegt, die Konzepte und Funktionsweisen der STL zu erläutern sowie auf typische Stolpersteine hinzuweisen. In diesem Zusammenhang ist vor allem Kapitel 2 hervorzuheben, das sich mit der Konzeption und Entwicklung der STL beschäftigt.

Bei der Beschreibung der einzelnen Komponenten der STL – Container, Iteratoren, Algorithmen, Funktionsobjekte usw. – gehen wir auch auf Implementierungsdetails ein, die vom Standard zwar nicht festgeschrieben sind, sich in der Regel aber aufgrund von Vorgaben für die Komplexität automatisch ergeben. Das Wissen um Implementierungsdetails unterstützt eine effiziente Benutzung der STL. Ferner demonstrieren zahlreiche kleine Anwendungsbeispiele den Gebrauch.

Im Schatten der STL existieren viele weitere interessante Komponenten der C++-Standardbibliothek, zu denen unter anderem *Streams*, *Strings*, *Auto-Pointer*, *Bitsets* und komplexe Zahlen gehören. Auch für diese Klassen liefern wir die für den praktischen Einsatz notwendigen Informationen.

Unsere Ausführungen basieren auf dem aktuellen C++-Standard, der unter der Bezeichnung ISO/IEC 14882:2003, *International Standard for the C++ Programming Language* (*Second Edition*), erschienen ist. Das Dokument kann beim *American National Standards Institute* (ANSI) bezogen werden. Der Download einer elektronischen Version ist derzeit für $18 von http://www.ansi.org/ möglich. Eine ältere Version vom Dezember 1996 ist unter http://www.dkuug.dk/jtc1/sc22/open/n2356/ frei zugänglich. Vom C++-Standardkomitee unter http://www.open-

std.org/jtc1/sc22/wg21/ diskutierte Korrekturen und Verbesserungsvorschläge haben wir weitgehend berücksichtigt.

Gegenüber der Urfassung der STL, die von *Alexander Stepanov* und *Meng Lee* bei *Hewlett Packard* als Technischer Report HPL-94-34 mit dem Titel „*The Standard Template Library*“ im April 1994 publiziert wurde, sind während der Standardisierung unzählige Änderungen vorgenommen worden. An Stellen, an denen es uns nützlich erscheint, weisen wir auf frühere Besonderheiten hin, damit sich unsere Programmbeispiele an ältere Versionen der STL anpassen lassen.

Da wir hier den C++-Standard beschreiben, sollten sich alle unsere Beispielprogramme mit jedem standardkonformen C++-Compiler erfolgreich übersetzen lassen. Zum Testen unserer Programmbeispiele haben wir die folgenden C++-Compiler eingesetzt (in alphabetischer Reihenfolge):

- Borland C++ unter Windows
- GNU g++ unter Linux und Solaris sowie Windows (cygwin)
- Kuck & Associates, Inc. KAI C++ unter Linux
- Microsoft Visual C++ unter Windows

Jedes Beispielprogramm lässt sich mit mindestens einem dieser Compiler erfolgreich übersetzen. Darüber hinaus decken die im Text angegebenen Definitionen für Klassen, Funktionen usw. einen Großteil der STL ab. Insbesondere älteren Implementierungen kann man damit auf die „Sprünge“ helfen, so dass man seine Programme gemäß dem aktuellen Standard schreiben kann und späterer Umstellungsaufwand entfällt.

Im Internet existieren frei verfügbare Implementierungen der STL und interessante Erweiterungen. Derzeit erscheinen uns insbesondere *STLport* und *Boost* (http://www.stlport.org/ sowie http://www.boost.org/) erwähnenswert.

Die C++-Standardbibliothek reizt die Programmiersprache C++ voll aus. Das geht sogar so weit, dass die Sprache für die Bibliothek erweitert wurde; *Member-Templates* sind dafür ein Beispiel (siehe Seite 70). Um die in diesem Buch dargestellten Themen verstehen zu können, muss man über fundierte C++-Kenntnisse verfügen. Zum Erlernen der Sprache C++ und als Nachschlagewerk empfehlen wir gerne unser Buch „*Programmieren in C++*“, das ebenfalls im Springer-Verlag erschienen ist.

Wir gehen davon aus, dass das Buch in der vorgegebenen Reihenfolge durchgearbeitet wird. Um langweilende Wiederholungen zu vermeiden, werden z. B. Elementfunktionen von Containerklassen nur beim ersten Auftreten im Text ausführlich und mit Beispielen erläutert.

Jeweils am Kapitelende haben wir einige Aufgaben zusammengestellt. Lösungsvorschläge befinden sich in Kapitel 15. Die meisten Aufgaben beschäftigen sich mit Themen, die über das im Text Gesagte hinausgehen. Dementsprechend

ist der Schwierigkeitsgrad recht hoch, und ein Blick auf die Lösungen lohnt sich.

Für Informationen rund um das Buch haben wir im *World Wide Web* eine Homepage unter http://www.wifo.uni-mannheim.de/veroeff/stl/ eingerichtet. Dort finden Sie:

- den Sourcecode aller mit dem Diskettensymbol 🖫 gekennzeichneten Programme und Lösungen;
- sämtliche Programmfragmente – Sie brauchen also nichts abzutippen;
- aktuelle Ergänzungen und Korrekturen.

Außerdem können Sie sich dort in unsere *Mailing-Liste* eintragen, damit wir Sie mit den jeweils neuesten Informationen zum Buch auf dem Laufenden halten können.

Über Anregungen unserer Leserinnen und Leser an unsere Postanschrift oder als E-Mail an stlbuch@wifo.uni-mannheim.de würden wir uns freuen. Wer uns einen Fehler (gleichgültig welcher Art) zuerst mitteilt, den erwähnen wir namentlich im *World Wide Web* auf unseren Seiten zum Buch.

Wir danken unseren Lesern – und hier insbesondere Rüdiger Dreier, Frank Fasse, Christian Hoffmann, Wolfgang Kaisers, Arnold Ludwig, Thomas Mehring, Olaf Raeke und Peter Schatte – für hilfreiche Hinweise zu den ersten drei Auflagen. Außerdem gilt unser Dank dem Springer-Verlag für die, wie immer, sehr gute Zusammenarbeit.

Stefan Kuhlins, Martin Schader

Inhaltsverzeichnis

Index

Falls sich von mehreren Seitenzahlen eine durch **Fettschrift** hervorhebt, ist dort eine ausführliche Erklärung zu finden. Einträge, die *kursiv* gesetzt sind, gehören nicht zur Standardbibliothek, sondern sind Beispiele.

– C –

– D –

– E –

– F –

– G –

– H –

– N –

– O –

– P –

– Q –

– R –

– S –

– T –

– U –

– V –

– W –

– X –

– Z –

Buchrückentext

Die C++-Standardbibliothek, deren interessantester Teil die Standard Template Library (STL) ist, revolutioniert die C++-Programmierung. Der korrekte und effiziente Einsatz stellt eine große Herausforderung dar, die nur mit fundiertem Wissen zu meistern ist. In diesem Buch wird die Funktionsweise und Implementierung der einzelnen Komponenten (Container, Iteratoren, Algorithmen, Funktionsobjekte, Strings, Streams usw.) verständlich und detailliert erklärt, wobei typische Stolpersteine aufgezeigt werden. Zahlreiche, mit gängigen C++-Compilern getestete Programmbeispiele demonstrieren den praktischen Einsatz. Antworten auf Standardfragen sowie unzählige Tipps und Tricks machen das Buch zu einem nützlichen Ratgeber für den alltäglichen Gebrauch. Anhand von anspruchsvollen Aufgaben und deren Lösungen kann das erworbene Wissen überprüft und vertieft werden. Der komplette Programmcode ist im Internet verfügbar. Ein ausführlicher Index ermöglicht gezieltes Nachschlagen.