Manfred Nagl

Einführung in die Programmiersprache Ada

Programmiersprachen

Formale Sprachen, von H. Becker und H. Walter

Einführung in ALGOL 60, von H. Feldmann

Einführung in ALGOL 68, von H. Feldmann

Einführung in die Programmiersprache PASCAL,
von K.-H. Becker und G. Lamprecht

Einführung in PASCAL, von H. Feldmann

Die Programmiersprache PASCAL, von D. Krekel und W. Trier

Einführung in die Programmiersprache Ada
von M. Nagl

Einführung in die Programmiersprache PL/1, von H. Kamp
und H. Pudlatz

Einführung in die Programmiersprache FORTRAN 77,
von G. Lamprecht

Einführung in die Programmiersprache SIMULA,
von G. Lamprecht

Einführung in die Programmiersprache BASIC,
von W.-D. Schwill und R. Weibezahn

BASIC in der medizinischen Statistik, von H. Ackermann

Einführung in die Programmiersprache COBOL,
von W.-M. Kähler

PEARL, Process and Experiment Automation Realtime
Language, von W. Werum und H. Windauer

Vieweg

Manfred Nagl

Einführung in die Programmiersprache Ada

Skriptum für
Hörer aller Fachrichtungen

Friedr. Vieweg & Sohn Braunschweig/Wiesbaden

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Nagl, Manfred:
Einführung in die Programmiersprache Ada: Skriptum
für Hörer aller Fachrichtungen / Manfred Nagl. –
Braunschweig; Wiesbaden: Vieweg, 1982.
(Uni-Text)
ISBN-13: 978-3-528-03347-7 e-ISBN-13: 978-3-322-85533-6
DOI: 10.1007/978-3-322-85533-6

1. Auflage 1982
Nachdruck 1983

Alle Rechte vorbehalten
© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1982

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für die Zwecke der Unterrichtsgestaltung gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien.

ISBN-13: 978-3-528-03347-7

VORWORT

Dieses Buch ist nicht für jemanden gedacht, der einen Ada-Übersetzer schreiben will. Es steht hier eher die methodische Verwendung der Sprache im Vordergrund und nicht die Abgrenzung erlaubter von nicht erlaubten (syntaktischen) Konstruktionen. Trotzdem sollte beim Nachschlagen wegen einer Unklarheit bezüglich Ada hier eine präzise Antwort auffindbar sein. Dieses Buch ist auch nicht als Einführung in das systematische Programmieren für Anfänger gedacht. Stattdessen wendet es sich an Personen, die bereits eine gewisse Erfahrung im Programmieren mit mindestens einer höheren Programmiersprache haben. Dies kann auch FORTRAN oder COBOL sein. Für diesen Personenkreis sollte die nachfolgende Ausarbeitung sowohl als Textbuch für Programmierkurse (auch wenn diese mangels eines verfügbaren Übersetzers vielerorts noch als Trockenkurs abgehalten werden müssen) als auch zum Selbststudium geeignet sein. Die Kenntnis einer neueren höheren Programmiersprache, wie etwa PASCAL, ist zwar nicht Voraussetzung für das Lesen, doch erleichtert sie den Einstieg.

Ada entstand aufgrund einer Initiative des Verteidigungsministeriums der USA, das seine Ausgaben für Software in den Griff zu bekommen versuchte (1975 etwa 3,5 Milliarden $). Ein großer Teil dieser Ausgaben ging in den Bereich der sogenannten eingebetteten Systeme (z.B. Bordcomputer eines Flugzeugs), der sich durch Effizienzvorgaben (Speicherplatz, Laufzeit), Hardwarenähe, z.B. durch Einsatz nichtstandardmäßiger Ein-/Ausgabe und Notwendigkeit von Konzepten für Realzeitanforderungen /Nebenläufigkeit auszeichnet. Insbesondere durch Unterstützung der Zuverlässigkeit, Portabilität und Wartungsfreundlichkeit von Software sollte eine Reduktion der Ausgaben erzielt werden. Das Ergebnis der Initiative ist eine universelle höhere Programmiersprache, die sich für einen breiten Einsatz eignet, nicht nur auf dem Gebiet der eingebetteten Systeme.

Ada bietet für Kontroll- und Datenstrukturen vielfältige Konstrukte an. Ada eignet sich wegen des Paketkonzepts, der Hilfsmittel zur getrennten Übersetzung und der generischen Programmeinheiten, insbesondere aber auch für das Programmieren "großer" Probleme, die von einer Programmierermannschaft und nicht von einem einzelnen Programmierer bearbeitet werden. Hier ist die Beachtung der Erkenntnisse des Software-Engineering, insbesondere der Programmiermethodik, unerläßlich. Wir haben deshalb in Kapitel 1 die wichtigsten Grundbegriffe aus diesem Bereich zusammengetragen. Aber auch während der restlichen Ausarbeitung wurde stets versucht, die methodische Sicht bei der Erörterung der Ada-Sprachelemente zu betonen.

Ada ist kein Forschungsergebnis, sondern das Resultat eines Entwicklungsprozesses, der öffentlich und unter großer Anteilnahme der Öffentlichkeit (aus Industrie, Verwaltung und Universitäten) ablief. Insoweit faßt Ada eher bisher Bekanntes zusammen, repräsentiert diese Sprache den Stand der Technik auf dem Programmiersprachensektor, als neue, bahnbrechende Wege zu gehen. Aus dieser Öffentlichkeit ist bereits jetzt, vor dem Großeinsatz dieser Sprache, ein Erfolg der Ada-Unternehmung erkenntlich: Mit der Diskussion über Ada wurde auch stets über Konzepte in Programmiersprachen diskutiert und damit zur Verbreitung und Erweiterung von Kenntnis über Programmiersprachen beigetragen.

Bücher über Programmierung fallen hauptsächlich in zwei Kategorien. Die eine Kategorie enthält Einführungen in die Programmierung, heute meist mit dem Zusatz methodisch oder systematisch. Um dieser Zielsetzung gerecht zu werden, sollten hierbei alle Gesichtspunkte des Programmierens angesprochen werden, insbesondere die Verifikation, die Effizienzanalyse und die Program-

miermethodik. Eine solche Darstellung ist jedoch einerseits weitgehend programmiersprachenunabhängig und andererseits nur für "kleine" Probleme machbar. Man sollte sich dabei einer einfachen aber sauberen Programmiersprache bedienen, wie etwa PASCAL oder ELAN. Ein Buch über Ada muß schon wegen des Umfangs der Sprache, die ja auch weniger für die Ausbildung als für den Einsatz in großem Maßstab erfunden wurde, in die zweite Kategorie von Büchern fallen, nämlich in die Einführungen in eine bestimmte Programmiersprache. Hier darf dann auch einige Kenntnis über die Programmierung vorausgesetzt werden. Wir haben bei der Erläuterung von Ada versucht, die sinnvolle und methodische Verwendung dieser Sprachkonstrukte in den Vordergrund zu rücken und nicht nur die Details zu sehen, wie das eine oder andere hinzuschreiben ist.
Jede Programmiersprache, insbesondere aber eine so umfangreiche wie Ada, ist stark rekursiv. Man weiß nicht, wo mit der Erläuterung begonnen werden soll, da die einzelnen Konstrukte direkt oder indirekt gegenseitig voneinander abhängen. Dadurch wird in vielen Fällen Wiederholung erzwungen. Wir haben versucht, aus dieser Not eine Tugend zu machen. So wurden die komplizierten Sprachelemente im allgemeinen stets zuerst in einfacher Form erläutert, die allgemeine Verwendung und das dahinterstehende Konzept wurde später nachgetragen. Eine Einführung in eine Programmiersprache sollte sich in ihrem Aufbau nicht am Sprachreport orientieren. Bei diesem ist es eine Zielsetzung, diese obige Rekursivität sichtbar werden zu lassen, während es hier aus didaktischen Gründen Zielsetzung ist, diese zunächst zu verbergen.
Ein einführendes Buch über eine Programmiersprache ist kein geeigneter Platz, sich kritisch mit einer Programmiersprache auseinanderzusetzen. Wir haben deshalb an einigen Stellen bewußt auf Randbemerkungen verzichtet. Die Akzeptanz einer Programmiersprache hängt nicht nur von ihren "technischen" Eigenschaften ab, sondern auch von dem Vorhandensein verständlicher Einführungen und Nachschlagewerke. Wir hoffen, zu diesem Gesichtspunkt der Pragmatik von Ada einen kleinen Beitrag geleistet zu haben.

Nun eine Übersicht: Kapitel 1 beschäftigt sich nach einer kurzen Darstellung der Entwicklungsgeschichte von Ada mit einigen Grundbegriffen des Software-Engineering. Darüber hinaus wird die Notation der Syntax erläutert und die Grundsymbole der Sprache werden aufgeführt.
Kapitel 2 ist den Elementen des Programmierens im Kleinen gewidmet und dabei hauptsächlich den Strukturen zur Ablaufkontrolle. Dazu zählen insbesondere die schon klassischen Kontrollstrukturen if-, case-, while- und for-Anweisung, die Funktionen und Prozeduren und die Sprünge. Nicht klassisch ist hier lediglich die Ausnahmebehandlung. Datenstrukturen werden hier nur andeutungsweise angesprochen, um kleine Programme oder Programmstücke formulieren zu können. Die Textein-/ausgabe wird jedoch bereits hier abgehandelt.
Kapitel 3 erläutert die Datenstrukturierung, die in neuen Programmiersprachen umfangreicher ist als die Ablaufstrukturierung. Hierzu zählen die Aufzählungstypen, die Felder und Verbunde, die in Ada in speziellen Formen als Felder mit unspezifizierten Grenzen und Verbunden mit Diskriminanten auftreten können. Erst dann folgt das Typenkonzept mit der Erklärung der Begriffe Typ, Untertyp, abgeleiteter Typ. Den numerischen Typen, nämlich den ganzzahligen und den reellen Datentypen, ist hier in Ada mehr Aufmerksamkeit geschenkt worden als in anderen Programmiersprachen, insofern, als sich der Programmierer über die Definition neuer numerischer Typen unabhängig von den vordefinierten machen kann. Ein ausführlicher Abschnitt über Zeiger und Haldenobjekte und ihre Verwendung in der Listenverarbeitung schließt das Kapitel ab.
Während Kapitel 2 und 3 eher die konventionellen Sprachelemente enthält, die hauptsächlich für das Programmieren im Kleinen, d.h. die Implementierung einzelner Moduln dienen, bietet das Kap. 4, teilweise auch 5, Elemente für das Programmieren im Großen, die während des Entwurfs bzw. während der Wartung großer Software-Systeme benötigt werden. Hierzu zählen hauptsächlich das Paketkonzept, das Konzept der Generizität, der privaten Typen und die Hilfs-

mittel zur getrennten Übersetzung, nämlich Bibliothekseinheiten und Untereinheiten. Da sich hinter letzteren mehr verbirgt als nur Hilfen für die textuelle Aufteilung des Programms, haben wir in einem sehr ausführlichen Abschnitt versucht, den Bezug zu Modulkonzepten herauszuarbeiten.
Kapitel 5 dient der Erklärung der nebenläufigen Programmierung. Nach Einführung der hierfür vorgesehenen Programmeinheit, der Task, der automatischen Aktivierung und der normalen Beendigung folgt das Rendezvous-Konzept als Konstrukt zur Synchronisation und zum gegenseitigen Ausschluß. Der nächste Abschnitt widmet sich der nichtdeterministischen Auswahl zwischen verschiedenen Interaktionswünschen auf der Seite der akzeptierenden Task. Verzögerung, Unterbrechung, Ausnahmebehandlung, normale und anomale Taskbeendigung schliessen sich an, sowie Ada-Spezielles wie Tasktypen und Entryfamilien. Ein größeres Beispiel schließt das Kapitel ab.
Das letzte Kapitel erläutert die Beziehungen zur Umwelt. Hierzu zählt die Ein-/Ausgabe mit ihrer Unterscheidung zwischen internen und externen Dateien, die neben Ein-/Ausgaberoutinen auch die Dateiverwaltung mit enthält. Schließlich gibt es noch vielfältige Möglichkeiten der Angabe von Darstellungen auf der Basismaschine.
Jedes Kapitel endet mit Übungsaufgaben, die der Leser zur Vertiefung seiner Ada-Kenntnisse benutzen kann. Ein umfangreiches Literaturverzeichnis soll Hinweise zum weiteren Studium geben. Die vordefinierten Pragmas und die vordefinierten Ausnahmen sind in Form zweier Anhänge am Ende zusammengestellt. Die vordefinierten Attribute werden in den entsprechenden Buchabschnitten erläutert. Dem leichten Nachschlagen schließlich dient ein Anhang, der die gesamte Grammatik enthält, sowie ein ausführliches Register.

Momentan ist eine Revision der Sprache Ada in Vorbereitung, die jedoch - bis auf Ein-/Ausgabe - nur geringfügige Änderungen bringen wird. Die meisten dieser Änderungen sind hier bereits eingearbeitet. Die eventuell noch ausstehenden dürften sich kaum auf das folgende Manuskript auswirken.

Ich möchte dieses Vorwort mit einer Danksagung an alle diejenigen abschließen, die zu der Gestalt dieses Buches in seiner jetzigen Form durch Kritik, Anregungen und Hilfe beigetragen haben. Hier sind Herr Dr. H. Hummel und Dr. Plödereder, München, sowie Herr Kollege J. Perl, Osnabrück, zu nennen. Zu besonderem Dank wegen sehr intensiver Mitarbeit bin ich jedoch den Herren Priv.-Doz. Dr. J. Ebert, G. Engels und W. Schäfer, alle Osnabrück, sowie Herrn R. Gall, Erlangen, verpflichtet. Schließlich gilt mein Dank auch Frau K. Guss für die große Geduld und Sorgfalt beim Schreiben dieses Manuskripts, sowie dem Vieweg-Verlag für das Anfertigen der Zeichnungen. Für Fehler und Mängel ist der Autor allein verantwortlich. Es kann auch kein Compiler dafür zur Rechenschaft gezogen werden, die (hoffentlich nicht zu zahlreichen) Syntaxfehler nicht erkannt zu haben.

Osnabrück, im Juni 1982 *Manfred Nagl*

INHALT