



Hochschule Darmstadt
- FACHBEREICH INFORMATIK -

Ada, eine Programmiersprache

Wissenschaftliches Arbeiten in der Informatik 1
Sommersemester 2016

vorgelegt von
Anna Klara Cominelli
744495
Waldemar Pastuchov
744212

Referentin: Prof. Dr. Schestag

Erklärung

Wir versichern hiermit, dass wir die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt haben.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von uns selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, den 15.06.16

Zusammenfassung

Die vorliegende Arbeit soll dem Leser einen Überblick über die Geschichte, Konzepte und Anwendungsfelder von der Systemprogrammiersprache Ada bieten.

Dabei wird auf die folgenden Aspekte genau eingegangen:

Die wissenschaftliche Arbeit fängt mit der Geschichte von der für das US Verteidigungsministerium erschaffenen Programmiersprache Ada an. Darin werden die Gründe und die Aspekte für die Entwicklung von Ada betrachtet. Die Herleitung des Namens wird ebenfalls analysiert.

Daraufhin wird die Struktur der Programmiersprache Ada untersucht. Zudem werden einige Code Beispiele vorgeführt. Diese werden mit bekannten Programmiersprachen, zur besseren Veranschaulichung, verglichen und erläutert. Es werden zusätzlich Vor- und Nachteile dargestellt.

Als Nächstes werden die Ähnlichkeiten und Unterschiede zu bereits bekannten Programmiersprachen untersucht. Insbesondere wird auf eine Gegenüberstellung von Ada und C++ eingegangen.

Zum Schluss befasst sich die Recherche mit der damaligen und heutigen militärischen und zivilen Anwendung. Darunter werden die Bereiche, in denen Ada angewendet wurde und wird, bestimmt. Zusätzlich wird dargestellt, wo diese Programmiersprache heute noch vorzufinden ist. Dabei wird auf den aktuellen Forschungsstand eingegangen.

Inhaltsverzeichnis

1	Einleitung	1
2	Geschichte	2
2.1	Geschichte des Namens	3
2.2	Anlass für die Entwicklung	3
3	Struktur von Ada	4
3.1	Vorteile	4
3.2	Nachteile	5
3.3	Analogie zu anderen Programmiersprachen	5
3.4	Ada im Vergleich zu C++	5
4	Anwendungsfelder	8
4.1	Militärische Anwendung	8
4.2	Zivile Anwendung	8
5	Forschungsstand	9
	Literaturverzeichnis	10

1 Einleitung

In der Welt der Programmierung gibt es sehr viele verschiedene Sprachen mit verschiedenen Anwendungsgebieten. Eine dieser Programmiersprachen nennt sich Ada. In der vorliegenden Arbeit wird genauer auf diese eingegangen.

Im Rahmen dieser wissenschaftlichen Arbeit wird zunächst auf die Entwicklung und die Geschichte von Ada 83 bis Ada 2012 eingegangen. Des Weiteren werden die Struktur und Analogien zu anderen Programmiersprachen behandelt. Anschließend wird auf die Anwendungsgebiete, sowohl im militärischen als auch im zivilen Bereich, eingegangen.

Das Ziel der Arbeit ist es, dem Leser einen Überblick über die Programmiersprache Ada zu geben. Der Schwerpunkt liegt beim Vergleich von Ada und C++.

2 Geschichte

Oft wurden vom amerikanischen Verteidigungsministerium Programmiersprachen entwickelt, zum Beispiel zwei ältere Programmiersprachen, wie Cobol (Common Business Oriented Language) und APT (Automatically Programmed Tools), die heute aber noch eingesetzt werden.

Ada wurde ebenfalls vom Verteidigungsministerium entwickelt und gefördert.

1974 wurde ein sechsjähriges Forschungs- und Entwicklungsprogramm gestartet, um eine "Höhere Programmiersprache für eingebettete Systeme" [1] zu schreiben. In den ersten Jahren von diesem Projekt wurden von Fachleuten des Militärs 14 verschiedene Programmiersprachen in den USA und in NATO Ländern untersucht. Keine von diesen untersuchten Sprachen war aber für alle militärischen Anwendungen geeignet (die Gründe werden in Abschnitt 2.2 *Anlass für die Entwicklung* erläutert).

Zwischen 1977 und 1979 wurden von verschiedenen Firmen, wie beispielsweise *Intermetrics* und *Honeywell*, verschiedene Programmiersprachen entwickelt. Die entwickelten Sprachen hatten ein Pseudonym, damit die Sprachen nicht zu der entwickelnden Firmen zurückverfolgt werden könnten. Zum Beispiel hatte die Sprache von *Intermetrics* das Pseudonym *Red* und die von *CII-Honeywell Bull* das Pseudonym *Green*.

Am 2. Mai 1979 fiel die Entscheidung auf die von Jean Ichbian (ein französischer Informatiker, Mitarbeiter von *Honeywell*) entwickelte Programmiersprache.

Ada war in dem Moment die erste standardisierte Hochsprache. Bis Herbst 1979 wurden Vorschläge zur Modifikation gesammelt, und Anfang 1980 wurden die ersten Compiler für Ada entwickelt. Die TU-Karlsruhe nahm ebenfalls am Projekt für die Compiler Entwicklung teil [1].

Die erste Version dieser Programmiersprache hieß Ada 83, die nachfolgende Version war Ada 95. Ada 95 wurde zwischen 1992 und 1995 von Tucker Traft, einem Mitarbeiter von *Intermetrics*, entworfen. Dabei wurden einige Verbesserungen vorgenommen wie zum Beispiel die Einführung von "tagged types" (erweiterbare Typen).

2005 wurde der Standardisierungsprozess mit Ada 05 abgeschlossen. Es wurden einige Anpassungen mit dem Umgang von Zeigern vorgenommen.

Die aktuellste Version von Ada ist Ada 2012.

Das Entwicklungsprojekt von Ada ist eine Innovation im Projekt-Management des Verteidigungsministeriums und dient als Vorbild für andere Programme und Projekte. Bei der Entwicklung von Ada, hat das Verteidigungsministerium zum ersten mal die gesamte Kontrolle über das Projekt behalten und über alle Frage selbst entschieden.

2.1 Geschichte des Namens

In Mai 1979 wurde die Programmiersprache von *Honeywell* Ada genannt, nach Augusta Ada Byron King, Countess of Lovelace allgemein als Ada Lovelace bekannt, einer Mathematikerin aus dem 19. Jahrhundert. Ada Lovelace (geborene Augusta Ada Byron 10. Dezember 1815 in London) war die erste Programmiererin. Ada zeigte schon als Kind ihre mathematischen Talente. Sie erhielt ihre Ausbildung von William Frend (Sozialreformer), William King (Familienarzt) und Mary Somerville (Astronomin und Mathematikerin).

In einem neun-monatigen Zeitraum zwischen 1842 und 1843 übersetzte Lady Lovelace einen italienischen mathematischen Artikel (*Notions sur la machine analytique de Charles Babbage*) von Luigi Menabrea über Analytical Engine. In der Übersetzung fügte sie Kommentare ein, um beim Leser ein besseres Verständnis für dieses Thema zu bewirken. Diese Kommentare über das Thema wurden dann länger als der Original-Artikel. In diesen Kommentaren ist ein komplexes Programm für den beschriebenen mechanischen Computer (Analytical Engine) zu finden, welches als das erste Computer Programm betrachtet wird. Aus diesem Grund wird Ada heute als erste Programmiererin der Welt bezeichnet.

In fast allen biographischen Werken findet man eine Hinweis darauf, dass Ada im Alter von 36 Jahre am 27. November 1852 an Gebärmutterkrebs gestorben ist[9].

2.2 Anlass für die Entwicklung

Die Programmiersprache Ada wurde (hauptsächlich) aus militärischen Gründen entwickelt wie zum Beispiel zur Raketensteuerung, aber auch zum Einsatz im nicht militärischen Bereich, wie beispielweise Schiffs- und Flugzeugsteuerung, rechnergesteuerte Maschinen und für Kommunikationssysteme. Das Verteidigungsministerium wollte eine besonders für militärische Zwecke geeignete Programmiersprache entwickeln.

Das Verteidigungsministerium stellte mit Untersuchungen fest, dass bei der Entwicklung eines neuen Systems 60% der Ressourcen bei der Wartung ausgegeben wurden und die restlichen 40% des Geldes für die Neuentwicklung. Der Grund dafür ist die Anzahl an verschiedenen und teilweise inkompatiblen Programmiersprachen (ca. 450 inklusive inkompatibler Dialekte bei 200 verschiedenen Rechnermodellen). Deswegen war es sehr schwierig, genügend Programmierer zu finden mit den notwendigen Kenntnissen. Die Sprache, die das Verteidigungsministerium gesucht hatte, sollte die folgenden Kriterien erfüllen: Allgemeinheit, Zuverlässigkeit, Wartbarkeit, Effizienz, Einfachheit, Implementierbarkeit, Maschinenunabhängigkeit. Die schon existierenden Sprachen waren für diese Zwecke nicht geeignet.

3 Struktur von Ada

Ada ist eine leicht verständliche Sprache, die Syntax basiert auf Pascal und auf der Sprachfamilie Algol (Algorithmic Language). Ada ist keine case-sensitive Sprache, zum Beispiel würden Ada, ADA oder ada laut Compiler als das gleiche Schlüsselwort interpretiert. Ada akzeptiert griechische, russische und asiatische Buchstaben. Ada ist objektorientiert und kann mit Unterprogrammen, Rekursion und abstrakten Datenstrukturen umgehen. Damit kann der Quellcode verkürzt werden und mögliche Fehler wegen langem und unübersichtlichem Code können vermieden werden.

In Ada werden wenige Symbole verwendet, wie zum Beispiel die arithmetische Operatoren (+, -, *, /, und **) und die Vergleichsoperatoren (<, <=, >, >=, =, /=).

In dieser Programmiersprache sind die folgenden 73 Schlüsselwörter reserviert:

about	abs	abstract	accept	access	aliased
all	and	array	at	begin	body
case	constant	declare	delay	delta	digits
do	else	elsif	end	entry	exception
exit	for	function	generic	goto	if
in	interface	is	limited	loop	mod
new	not	null	of	or	others
out	overriding	package	pragma	private	procedure
protected	raise	range	record	rem	renames
requeue	return	reverse	select	separate	some
subtype	synchronized	tagged	task	terminate	then
type	until	use	when	while	with
xor					

3.1 Vorteile

Die Programmiersprache Ada ist in vielen verschiedenen Bereichen anwendbar, der bekannteste davon ist der militärische Bereiche. Ada wird aber auch im zivilen Bereiche angewendet wie zum Beispiel in Schienenverkehr (dies wird in Kapitel 4. *Anwendungsfelder* besser erläutert.)

Ein großer Vorteil dieser Sprache ist, dass viel Wert auf die Lesbarkeit gelegt wurde, dadurch wird der Code übersichtlicher.

Ada ist besonders geeignet für die Entwicklung von großen und langlebigen Softwareprojekten. In der Entwicklung von Ada wurde Wert drauf gelegt, dass die Sprache portabel ist und es keine systemabhängigen Unterschiede in einer Code-Übersetzung gibt.

In der neuen Version von Ada (ab Ada 95) kann man durch den Exception Handling Mechanismus die Fehler abfangen und damit den Quellcode sicherer machen.

3.2 Nachteile

Leider ist die Fehlerbehandlung in Ada oft unklar und kompliziert. Dies hat in der Vergangenheit zu falschen Programm-Implementierungen geführt. In Ada 83 konnten einige Fehler nicht abgefangen werden, weshalb schwere Unfälle verursacht wurden. Es konnte beispielsweise nicht abgefangen werden, ob eine Berechnung einen Overflow ergeben hat. Am 4. Juni 1996 musste deshalb bei der Rakete Ariane 5 von der ESA (European Space Agency - europäischen Raumfahrtagentur), welche größtenteils in Ada programmiert wurde, die Selbstzerstörung ausgelöst werden. In diesem Fall hat die Fehlerbehandlung nicht richtig funktioniert, eine falsche Geschwindigkeitsberechnung, die nicht abgefangen wurde, hat den Kurs der Rakete in Richtung Karibik abgelenkt.

Zu den Anfängen der Programmiersprache Ada wurden wegen der Validierung noch nicht viele Compiler entwickelt. Dadurch gab es keine Konkurrenz und die Compilerlizenzen waren außerordentlich teuer. Mit der Zeit setzte sich Ada durch und es wurden immer bessere und leistungsfähigere Compiler entwickelt, wodurch der Wettbewerb stieg und die Preise der Compiler sanken. Dennoch ist, hinsichtlich der Konkurrenz, erst kürzlich der qualitative, leistungsfähiger und kostenlose Compiler *GNAT* erschienen.

3.3 Analogie zu anderen Programmiersprachen

Ada unterscheidet sich syntaktisch von anderen Programmiersprachen wie C, C++ und Java, weil in dieser Sprache besonderer Wert auf die Lesbarkeit des Quellcodes gelegt wurde. Ada hat ähnliche Conditions wie C++ und Java, die am häufigsten verwendeten Condition in Ada sind: *if*, *then*, *elsif*, *else* und *end if*. *Use* in Ada ist wie *using namespace* in C++, oder *import* in Java. Ada hat keine Konstruktoren und keine Destruktoren wie bei C++ und Java, aber eine ähnliche Funktion in default Form von *initialization* und *finalization*. Ada unterscheidet sich in der Division und Moduleberechnung von C++ und Java. Ada hat dafür reservierte Worte: *div* und *mod*. In Java und C++ wird es mit dem Symbol *"/*" dividiert und mit dem Symbol *"%"* Modul berechnet.

In Gegenteil zu C++ und Java gibt es in Ada keine *double* Variablen, um Kommazahlen zu definieren werden in Ada *float* Variablen verwendet.

3.4 Ada im Vergleich zu C++

Sowohl in Ada, als auch in C++, gibt es zwei verschiedene Dateien für die Spezifikation und für den Körper. In C++ heißen diese Header und Cpp und in Ada heißen sie *a.ads* für die Spezifikation und *a.adb* für die Implementierung. Mittels *# include <header_name>* werden in C++ andere bekannte Dateien in das Quellprogramm eingefügt, in Ada gibt es dafür den Befehl *with* (z.B: *with Ada.Text_IO*).

Ein großer Unterschied zwischen Ada und C++ ist, dass in Ada keine *main*-Methode existieren muss. *Main* ist in Ada kein Schlüsselwort, es kann als Name für eine Funktion verwendet werden, es muss aber nicht unbedingt in einem Programm die Funktion

main existieren.

Im Gegensatz zu C++ gibt es in Ada keinen Unterschied zwischen *struct* und *class*, es gibt nur *package*, welches die gleichen Eigenschaften wie *class* aufweist.

In beiden Programmiersprachen gibt es einen Aufzählungstyp, in Ada heißt er *type* und in C++ *enum*.

In Ada werden keine geschweiften Klammern verwendet, stattdessen werden Schlüsselwörter benutzt, um die verschiedene Blöcke zu trennen. Beispielweise fangen die Blöcke mit dem Schlüsselwort *begin* an und enden mit dem Schlüsselwort *end*.

In Ada gibt es keine reservierten Schlüsselwörter für Datentypen wie Integer, Float oder Character, diese Variablentypen sind im Package Standard definiert, aber um sicher zu sein, dass die Programme übertragbar sind sollte man seine eigenen Variablentypen definieren. In beiden Sprachen gibt es die Möglichkeiten Variablen vom Typ boolean bzw. String zu benutzen.

In Ada gibt es mehrere reservierte Wörter um positive Zahlen zu bezeichnen.

1. *natural*, kann man mit *unsigned* in C++ vergleichen. Es wird benutzt um positive Zahlen zu definieren.

2. *positive*, wird verwendet um positive Zahlen, zu bezeichnen, die 0 nicht inbegriffen. Die *Identifiers* müssen mit einem Buchstaben anfangen aber können, wie auch in C++, Symbole wie "_" oder "-" enthalten.

In Ada werden keine Doppel-Operatoren (==, ++ oder --) verwendet, es wird nur der einzelne Operator (=, + oder -) benutzt. In Ada kann der Operator "++" oder "--" nicht benutzt werden um zu inkrementieren oder zu dekrementieren. Das "--" wird wie "//" in C++ benutzt, um Kommentare zu schreiben, die bis zum Ende der Zeile andauern. Im Unterschied zu Ada gibt es in C++ die Möglichkeit, Blockkommentare zu schreiben mit "/*..*/".

In C++ werden die Symbolen "!", "&" und "|" benutzt, um boolesche Operationen durchzuführen, Ada hat dafür reservierte Schlüsselwörter wie *not*, *and* und *or*. Um zwei Werte auf Gleichheit zu vergleichen gibt es in C++ und Ada einen kleinen Unterschied. In Ada wird die Gleichheit mit dem Symbol "=" bezeichnet, in C++ wird das Symbol "==" verwendet. Die Ungleichheit wird ebenfalls unterschiedlich bezeichnet. In Ada wird dafür "/=" verwendet und in C++ wird "!=" eingesetzt.

for-Schleife in C++

```
for (int i =0; i< 9 ; i++) {  
    cout << i << endl;  
}
```

for-Schleife in Ada

```
for Variable in 0 .. 9 loop  
    Put_Line (Integer ' Image (Variable));  
end loop;
```

Hello World C++

```
#include <iostream>
using namespace std;
int main(){
cout << "Hello world" << endl;
return 0;
}
```

Hello World in Ada

```
with Ada.Text_IO;
use Ada.Text_IO;
procedure Main is
begin
    Put_Line ("Hello World");
end Main;
```

Switch case in C++

```
switch (Variable) {
case 0: cout << "Zero" << endl;
break;
case 1: case 2: case 3: case 4: case 5:
case 6: case 7: case 8: case 9:
cout << "Positive Digit" << endl;
break;
case 10: case 12: case 14: case 16: case 18:
cout << "Even Number between 10 and 18" << endl;
break;
default: cout << "Something else";
}
```

Switch case in Ada

```
case Variable is
when 0 => Put_Line ("Zero");
when 1 .. 9 =>
Put_Line ("Positive Digit");
when 10 | 12 | 14 | 16 | 18 =>
Put_Line ("Even Number between 10 and 18");
when others => Put_Line ("Something else");
end case;
```

4 Anwendungsfelder

Die Programmiersprache Ada wurde für eingebettete und real-time Systeme entwickelt und wird immer noch in diesen Gebieten verwendet. Ada wird im militärischen Bereich bei der Flugsicherung, für Waffensysteme oder in der Raumfahrt verwendet, aber auch in zivilen Bereichen, wie zum Beispiel in der Medizin oder für Sicherungssysteme in Eisenbahnen und Kernkraftwerken. Da Ada in mehreren internationalen Standards zur Entwicklung von sicherheitsrelevanten Systemen als sehr empfehlenswert gekennzeichnet ist, wird diese oft bei Systemen mit hohen Sicherheitsanforderungen eingesetzt, wie beispielsweise die Software der Boeing 777, welche zu 99% aus Ada besteht.

4.1 Militärische Anwendung

Aufgrund der Tatsache, dass Ada hauptsächlich zur militärischen Anwendung entwickelt wurde, ist diese dort auch am meisten vertreten. Ada ist in zahlreichen militärischen Systemen und Maschinen der Länder Australien, Kanada, Indien, Schweden und USA enthalten. In Australien kommt Ada in der Luftwaffe bei dem Flugzeug *P3 Surveillane* zum Einsatz. Ebenso wird Ada bei der Marine in Fregatten bei Steuerungssystemen und bei Radar- und Warnsystemen eingesetzt. In Kanada wird Ada in Hubschraubern der Luftwaffe verwendet. Der schwedische und indische Militär nutzen Ada ebenfalls in der Luftwaffe. Im Militär der USA findet Ada die meisten Anwendungen. In vielen Flugmaschinen der US Air Force, wie beispielsweise der *F-16 Fighter*, und in zahlreichen Maschinen der US Army und US Navy kommt Ada ebenfalls zum Einsatz. In der Raumfahrttechnik kommt Ada beispielsweise in der Internationalen Raumstation (ISS-International Space Station) bei der Flugsteuerungs-Software, zum Einsatz. Der Mars Express der ESA enthält ebenfalls Ada Code. In einigen weiteren Projekten der Raumfahrt und Satelliten kommt ebenfalls die Programmiersprache Ada zum Einsatz.

4.2 Zivile Anwendung

Ada hat sich nicht nur in der Luftwaffe bewährt, sondern auch im zivilen Luftverkehr. In einigen Ländern, unter anderem China, Deutschland und den USA kommt Ada im Flugverkehrsmanagement zum Einsatz. Namhafte Passagierflugzeug, wie beispielsweise der Airbus 380 und weitere Airbus und Boeing Modelle, enthalten Module, die mit Ada programmiert wurden. Im Schienenverkehr spielt Ada ebenfalls eine wichtige Rolle. Die U-bahn-Systeme einiger Großstädte, unter anderem Hong Kong, London, Madrid und Paris, sind teilweise in Ada programmiert. Das European Train Control System (ETCS) und die New York City U-Bahn enthalten ebenfalls Module, die mit Ada programmiert wurden. Einige Kommunikations- und Navigationsatelliten sind auch in der Programmiersprache Ada geschrieben. Sogar bei Desktop Anwendungen trifft man auf Ada. Beispielsweise in dem Programm TeXCAD, welches dazu dient in LaTeX Bilder zu zeichnen und zu retuschieren. Desweiteren kommt Ada in Bank- und Finanz-Systemen, Informationssystemen, TV/Unterhaltungsindustrie, Medizintechnik und allgemeiner Industrie zur Anwendung.

5 Forschungsstand

Wie man dem vorangegangenen Kapitel entnehmen kann, wird Ada aktuell noch in vielen Branchen angewendet. Vor allem im militärischen und im Raumfahrt-Bereich wird diese Programmiersprache in naher Zukunft nicht untergehen, da vor allem raumfahrttechnische Projekte oft Jahrzehnte andauern. Dies hat zur Folge, dass die Programmiersprache Ada immer weiter entwickelt wird, so lange sie angewendet wird. Mit der letzten Aktualisierung im Jahr 2012 wurde die vorherige Version Ada 2005 überarbeitet. Die aktuellste Version von Ada ist Ada 2012.

“It’s interesting that people think that Ada has gone away. In this industry, there is a technology du jour. And people assume things disappear. But especially in the Defense Department, nothing ever disappears”, Robert Dewar, president of AdaCore und Professor der New York University. [www.adacore.com]

Ada Lovelace ist eine Pionierin auf dem Gebiet der Informatik. Sie gilt nicht nur für die Programmiersprache Ada als Vorbild, sondern auch allgemein für Frauen in Technik und Naturwissenschaften. Es werden Projekte und Initiative mit Lovelace als Leitbild ins Leben gerufen. Beispielsweise die in Australien gegründete Ada Initiative, setzt sich für vermehrte weibliche Teilhabe im Open-Source-Umfeld ein. Der *Ada Lovelace Day* ist ein internationales Fest der Leistung von Frauen in der Wissenschaft, Technik, Ingenieurwissenschaften und Mathematik, welches jährlich in Oktober gefeiert wird.

Literaturverzeichnis

- [1] Bickenbach, J (Hrsg) (1985): Militarisierte informatik. Marbuch Verl., Marburg.
- [2] Barnes, J (2012): Ada 2012 Rationale. Springer
- [3] Burns, A, Wellings, AJ (2007): Concurrent and real-time programming in Ada 2005. Cambridge University Press, Cambridge, New York.
- [4] Goos, G, Persch, G, Uhl, J (1987): Programmiermethodik mit Ada. Springer, Berlin.
- [5] Kordon, F, Kermarrec, Y (Hrsg) (2009): Reliable software technologies - Ada-Europe 2009. 14th Ada-Europe International Conference on Reliable Software Technologies, Brest, France, June 8 - 12, 2009 ; proceedings. Springer, Berlin.
- [6] Ledgard, H (1983): ADA®. An Introduction. 1983. Auflage. Springer New York, New York, NY.
- [7] Michael B. Feldman, Professor Emeritus, Dept. of Computer Science: Who's Using Ada? <https://www.seas.gwu.edu/~mfeldman/ada-project-summary.html>. Abgerufen am 13.06.2016
- [8] Quentin Ochem (2013): Ada for the C++ or Java Developer. 1. Auflage. AdaCore.
- [9] Wacker, H (26.12.2015): Die Programmiersprache Ada - Linksammlung und Überblick. <http://www.wackerart.de/ada.html>. Abgerufen am 19.04.2016.