



Fortran

by Carlotta Williams Lopez



Gliederung

- Allgemeine Information
- Geschichte + Versionen
- Bibliotheken
- Compiler
- Fortran
 - Aufbau
 - Datentypen, Variablen
 - IF - Abfrage, Do- Schleife
 - Funktion, Subroutine
 - Module
- Beispiel
- Zusammenfassung
 - Vor- und Nachteile
- Quellenverzeichnis



Fortran Logo

[Home - Fortran Programming Language \(fortran-lang.org\)](http://fortran-lang.org)

Allgemeine Information

- Höhere Programmiersprache
- Fortran verwendet das Konzept der starken Typisierung
 - kennt (explizite und implizite Typisierung)
- Für numerische Berechnungen vorgesehen und optimiert
- Potenz-Operator ** von Anfang an
- Vektor- und Matrix-Operationen standardisiert
- Umfangreiche Bibliotheken (Wissenschaftlich und numerische Berechnung)

Bibliotheken

Quelltext Bibliotheken

- Werte Definitionen, Deklarationen
Funktionen, Klassen, generische Bestandteile

Statische Bibliotheken

- Linker/ Binder verbindet
- Nur einmal im Speicher gehalten werden
- sucht Unterprogramme heraus
- Dateien extrahiert und an das Programm gebunden

Dynamische Bibliotheken

- Erst bei Bedarf geladen
- Lader verbindet
- nur einmal im Speicher gehalten
- Multitasking- System vorteilhaft
- Laufzeitbinder aktiviert
- Sucht Unterprogramm
- führt Unterprogramm erstmalig aus
- Adresse vorhanden
- Speicherbedarf kleiner
- virtuellen Speicher
- nur benötigte Teile der Bibliothek

Geschichte / History

1953

Vorschlag
von John W.
Backus.



20. September 1954

Erstes
Fortran-Programm
ausgeführt.



1954

Beginn des Projekts
Fortran, unter Leitung von
John W. Backus.



1957

Compiler als Marktreif
empfunden.
Mit jedem IBM 704 System
ausgeliefert



Versionen

1957/58	FORTRAN II	Inliner-Assembler, Kommentare
1961/62	FORTRAN III	Verbessert + Erweitert
1966	FORTRAN 66	1. Standardisierte Fortran Version
1978	FORTRAN 77	Do- Schleife, IF THEN -ELSE, Character Datentyp
1991	FORTRAN 90	Free Form style, Module, Zeiger Datenverbund
2003	FORTRAN 2003	OOP, C.Bindung
2010	FORTRAN 2008	Co-Array & kleine Änderungen
2018	FORTRAN 2018	Kleine Änderung

Proprietäre Software

- Anbieter sind Computerhersteller oder sp. Softwarehersteller
- FORTRAN 77 Compiler nicht mehr hergestellt
- Privatanwender bzw. nicht kommerzielle Nutzung unentgeltlich

Compiler

Freie Software

- Ab Version 4.0 enthält die praktisch für alle Plattform verfügbare GNU Compiler Collection

Transcompiler

- Transcompiler automatische Übersetzung von FORTRAN 77 in C
- Kaum für Menschen lesbar

Compiler

Fortran

Aufbau:

- Anfang: program-, subroutine- oder funktion-statement
- Deklarationen: Vereinbarung von Variablen u.a.
- Ausführbarer Teil: alle ausführbaren Anweisungen
- Ende: end-statement
- Direkt lesbaren Text, Buchstaben Zahlen und Sonderzeichen
- Leerzeichen keinerlei Bedeutung
- Oben nach Unten abgearbeitet

Datentypen

	Datentyp	Kommentar	
Arithmetische Datentypen	Integer	Ganzzahlen	15. -6500, 200.000.000
	real	Gleitkommazahlen einfacher Genauigkeit	3.1415, -5.5, 7e3,
	(double precision)	Gleitkommazahlen doppelte Genauigkeit	3.1415D0, -5.5D0
	complex	Komplexe Zahlen (Zwei real-Zahlen)	(3.1415, -5.5), (1.4, 7.1E4)
Logischer Ausdruck	logical	Logischer Datentyp (Wahr oder Falsch)	.TRUE., .FALSE.

Schreibweise 1	Schreibweise 2	kommentar
B"zahl"	b´zahl´	B inärzahl
O"zahl"	O´zahl´	O ktalzahl
Z"zahl"	Z´zahl´	Hexadezimalzahl

DO-SCHLEIFE

- Einfache DO-SCHLEIFE

```
DO nr zählvariable = startwert, endwert [, schrittweite]
  anweisungsblock
nr CONTINUE
```

Quelle: [Fortran: FORTRAN 77: Verzweigungen und Schleifen – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)

IF- Abfrage

- Unterschiedliche IF-Abfragen:
 - IF Einzeiler
 - IF-THEN
 - IF-THEN-ELSE
 - ELSE-IF

Variablen

- Beim Programmstart keinen Wert

Charakterisiert durch:

- Symbolischen Namen
- Datentyp
- Wert
- Speicherplatz

Funktion Unterprogramm

- Gibt einen Wert zurück
- Datentyp durch Zuweisung an den Funktionsnamen erreicht
- Sonst im Vereinbarungsteil

```
datentyp function funktionsname ( [formale parameter] )  
  Vereinbarungsteil  
  Anweisungsteil  
  funktionsname = wert  
  [return]  
end function funktionsname
```

Allgemeine Funktionsdeklaration

Quelle [Fortran: Fortran 95: Unterprogramme – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)

Subroutine Unterprogramm

- Mehr als ein Wert zurückgeben
- Keinen Datentyp und Rückgabewert

```
subroutine subroutinenname ([formale parameter])  
  Vereinbarungsteil  
  Anweisungsteil  
  [return]  
end subroutine subroutinenname
```

Die allgemeine Deklaration einer Subroutine

Quelle: [Fortran: Fortran 95: Unterprogramme – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)

Module

- Ersetzen FORTRAN 77-
COMMON Blöcke
- Variablen, Konstanten und
Unterprogramme abgelegt
werden
- Von mehreren
Programmeinheiten genutzt
werden

```
use modulname [, only liste]
```

```
module modulname  
  [implicit none]  
  [save]
```

Deklaration von Variablen, Konstanten

```
[contains  
  Unterprogramme  
]  
end module modulname
```

Quelle: [Fortran: Fortran 95: Unterprogramme – Wikibooks](#),
[Sammlung freier Lehr-, Sach- und Fachbücher](#)

Zusammenfassung

Vorteile:

- Felder(Vektoren, Matrizen, ...) teil der Sprache
- sehr schneller Code
- Fließkommazahlen
- Optimierung Zahlreiche Bibliotheken
- Herstellerunabhängige Standardisierung
- Höchstleistungsrechnen
- Heutzutage noch eingesetzt
- Flache Lernkurve

Nachteile:

- hardwarenahe Programme, Web Applikationen, Datenbanken, ego shooter games

Quellenverzeichnis

- [Fortran: Einleitung – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)
- [Fortran: Fortran 95: Unterprogramme – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)
- [fortran.pdf \(uni-kiel.de\)](#)
- [Fortran: Einleitung – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)
- [Fortran gestern, heute und morgen - entwickler.de](#)
- [Wissenschaftliches Programmieren in Fortran \(uni-leipzig.de\)](#)
- [Typisierung \(Informatik\) – Wikipedia](#)
- [Deutsche Fortran WebSite](#)
- [Fortran: FORTRAN 77: Verzweigungen und Schleifen – Wikibooks, Sammlung freier Lehr-, Sach- und Fachbücher](#)