

# Einfache Lehrn-Hilfen

## Inhaltsverzeichnis

---

Lizenz - Bitte lesen !!! .....	4
Überblick .....	5
Inhalt .....	6
Liste der Tabellen .....	11
Über dieses Handbuch .....	11
Bezeichnungen .....	11
Syntax Diagramme .....	11
Über die Sprache Pascal .....	11
1. Pascal Zeichen und Symbole .....	11
2. Konstanten .....	12
3. Typen .....	12
4. Variablen .....	12
5. Objekte .....	12
6. Klassen .....	12
7. Schnittstellen (Interface's) .....	12
8. Generics .....	12
9. Erweiterte Record's .....	12
10. Klassen, Record's, und Typen-Helfer .....	12
11. Objektorientierte Pascal - Klassen .....	12
12. Ausdrücke .....	12
13. Anweisungen .....	12
14. Benutzung von Funktionen und Prozeduren .....	12
15. Operatoren Überladung .....	12
16. Programme, Module und Blöcke .....	12
17. Ausnahmen .....	12
18. Assembler .....	12
18.1. Anweisungen .....	12
18.2. Prozeduren und Funktionen .....	12



# Pascal - FPC Referenz Handbuch



FPC v.3.2.2

Free Pascal Compiler



[<< Zurück](#)[Inhalt](#)[Weiter >](#)

[\[::Template::\]](#)

[www.freepascal.org](http://www.freepascal.org)

(C) [::Template\_Date::] by Jens Kallup - paule32  
Übersetzung lizenziert unter der Creative Common Lizenz NC





## Pascal - Einfache Lernhilfen



FPC v.3.2.2

Free Pascal Compiler



### Lizenz

Bitte lesen Sie diese Lizenz gründlich durch, bevor Sie Lazarus die Dokumentation oder Teile dieser Software verwenden wollen. Sollten Sie nicht mit den hier aufgeführten Bedingungen einverstanden sein, ist die Nutzung oder Verwendung des Quellcodes zu diesen Produkt (oder auch Teile davon) nicht gestattet.

**[::TemplateMDnewline::]**

Bei der Verwendung darf kein kommerzieller Zweck der Gewinnerzielung entstehen !

**[::TemplateMDnewline::]**

Freeware und Shareware-Programme zeichnen sich vor allem dadurch aus, dass sie kostenlos beziehungsweise gegen einen relativ geringen Preis dem Nutzer die Verwendung teilweise hochwertiger Computerprogramme ermöglichen, die sich durchaus mit kommerziellen Produkten messen lassen können. Daher sind diese Softwaretypen gerade in Zeiten begrenzter finanzieller Mittel - auch im schulischen Umfeld sehr beliebt und kommen in vielfältigen Gebieten zum Einsatz (zum Beispiel bei der Netzwerkadministration, bei der Einrichtung von Servern, bei der Installation von Internet-Software auf den Nutzer-PCs oder bei Office-Anwendungen).

**[::TemplateMDnewline::]**

Da es sich auch bei Free- und Sharewareprogrammen um Computerprogramme handelt, die gemäß § 69a Urheberrechtsgesetz (UrhG) urheberrechtlich geschützt sind, liegt es bei diesen Softwareprodukten weitgehend in den Händen des Rechteinhabers zum Beispiel des Programmierers - zu bestimmen, in welchem Umfang diese durch dritte Personen genutzt werden dürfen.

**[::TemplateMDnewline::]**

Dabei können Freeware- und Sharewareprogramme nach den Lizenzbedingungen in der Regel beliebig kopiert und weitergegeben werden, dagegen ist vor allem eine Veränderung der Programme üblicherweise nicht gestattet. Es ist deshalb eigentlich auch nicht korrekt, wenn im Zusammenhang mit Free- und Shareware immer wieder der Begriff der "Public-Domain-Software" verwendet wird, der frei übersetzt "Software, die im öffentlichen Eigentum steht" bedeutet. Denn dies impliziert, dass die Software von der Öffentlichkeit beliebig genutzt und damit auch verändert werden darf (also gemeinfrei ist); letzteres ist bei Free- und Shareware aber gerade nicht der Fall.

**[::TemplateMDnewline::]**

### **Welche Arten von Shareware gibt es ?**

**[::TemplateMDnewline::]**

Shareware kann man in zahlreiche Unterkategorien einteilen. So kann Shareware beispielsweise werbegestützte Software oder "Adware" sein, welche dem User Werbung anzeigen soll. Dies hat den Zweck Einnahmen für den Eigentümer zu generieren. Eine weitere Art bezeichnet man als Demoware, wobei es sich hier nur um eine Demoversion der Software handelt.

Dabei ist es oft eine Testversion mit einer festgelegten Zeitspanne ("Trialware"). In anderen Fällen kann die gesamte Funktionalität der Anwendung deaktiviert sein, so dass man zwar alle Funktionen sehen kannst, aber dafür bezahlen muss ("Crippleware").

**[::TemplateMDnewline::]**

Weiterhin gibt es auch "Freemium"-Shareware, bei welcher einige Funktionen in der kostenlosen Version verfügbar sind, Sie diese jedoch bezahlen müssen, um die volle Funktionalität freischalten zu können.

Obwohl Shareware eine gute Option für jeden ist, der Software testen möchte, bevor man sich zu einem Kauf verpflichtet, sollten Sie dennoch vorsichtig sein. Denn Cyberkriminelle sind nur allzu bereit, den Eifer der Menschen auszunutzen, etwas gratis zu bekommen.

**[::TemplateMDnewline::]**

Dieses Versprechen freier Software ist eine gängige Social-Engineering-Taktik, mit der sie Internetnutzer dazu bringen, bösartige Software herunterzuladen.

Bei Shareware gibt es ebenso wie bei anderen Softwareprodukten keine Garantie dafür, dass die Software virenfrei ist. Aus diesem Grund ist es wichtig, dass der Benutzer sein Antivirenprogramm auf dem aktuellen Stand hält, bevor er Shareware herunterlädt.

**[::TemplateMDnewline::]**

Prinzipiell kann jeder Shareware herunterladen, solange er einen Computer hat, der die Systemanforderungen des Programms erfüllt. Es gibt jedoch einige Seiten, die nur Benutzern in bestimmten Ländern das Herunterladen von Shareware erlauben.

**[::TemplateMDnewLine::]**

Nicht alle Shareware-Programme werden überall zum Kauf angeboten. Der beste Weg, um zu sehen, wo man ein bestimmtes Programm kaufen kann, ist, die Website des Entwicklers zu besuchen und herauszufinden, auf welchen Plattformen das Programm erhältlich ist.

**[::TemplateMDnewLine::]**

Freeware ist Software, die kostenlos heruntergeladen werden kann, aber keine Lizenzgebühren mehr verlangt. Im Gegensatz dazu erfordert Shareware normalerweise eine Lizenzgebühr für den vollen Zugang zu allen Funktionen. Beide Modelle ermöglichen es Benutzern, Software kostenlos auszuprobieren, aber Shareware ist normalerweise etwas umfangreicher und bietet mehr Funktionen.

**[::TemplateMDnewLine::]**

Dieses Software-Produkt besitzt eine Hybrid-Lizenz zwischen Free- und Share-Ware. Sie können es kostenlos einsetzen und an andere Personen weitergeben, solange Sie keine Teile der Software ändern oder kopieren. Es bedarf einer schriftlichen Einwilligung des Entwicklers (Jens Kallup), sollten Teile geändert oder übernommen werden.

**[::TemplateMDnewLine::]**

Sie können dieses Produkt auf mehreren lokalen Computer gleichzeitig nutzen. Eine öffentlich zugängliche Nutzung ist nicht bestrebt mit der aktuellen Version.

Obwohl bei der Entwicklung dieser Software viel Sorgfalt gepflegt wurde, können Fehler nicht ausgeschlossen werden.

**[::TemplateMDnewLine::]**

Es werden daher keine Garantien auf entstanden Schäden oder Kosten übernommen, die während der Verwendung dieser Software entstehen. Alles erfolgt auf Eigenes Risiko !!!

**[::TemplateMDnewLine::]**

Für zukünftige Entwicklungen können Sie jedoch einen kleinen Obolus in Form von Spenden hinterlegen.

## **Überblick**

Free Pascal ist ein ausgereifter, vielseitiger, quelloffener Pascal-Compiler. Er kann für viele Prozessorarchitekturen eingesetzt werden: Intel x86 (16 und 32 bit), AMD64/x86-64, PowerPC, PowerPC64, SPARC, SPARC64, ARM, AArch64, MIPS, Motorola 68k, AVR, und die JVM. Zu den unterstützten Betriebssystemen gehören Windows (16/32/64 Bit, CE und natives NT), Linux, Mac OS X/iOS/iPhoneSimulator/Darwin, FreeBSD und andere BSD-Varianten, DOS (16 Bit oder 32 Bit DPML), OS/2, AIX, Android, Haiku, Nintendo GBA/DS/Wii, AmigaOS, MorphOS, AROS, Atari TOS und verschiedene Embedded-Plattformen. Zusätzlich ist in der Entwicklungsversion Unterstützung für RISC-V (32/64), Xtensa und Z80 Architekturen und für die LLVM Compiler Infrastruktur verfügbar. Zusätzlich betreut das Free Pascal Team einen Transpiler für Pascal nach Javascript namens pas2js.

**[::TemplateMDnewLine::]**

### **Aktuelle Version**

Version 3.2.2 ist die letzte stabile Version von Free Pascal. Klicken Sie auf den **Download-Link** und wählen Sie einen Spiegel in Ihrer Nähe, um Ihre Kopie herunterzuladen. Die Entwicklungsversionen haben die Versionsnummern 3.3.x. Auf der Entwicklungsseite erfahren Sie, wie Sie die neuesten Quellen erhalten und die Entwicklung unterstützen können.

**[::TemplateMDnewLine::]**

### **Funktionen**

Die Sprachsyntax ist sehr gut kompatibel mit TP 7.0 sowie mit den meisten Versionen von Delphi (Klassen, rtti, Exceptions, AnsiString's, WideString's, Interfaces). Es gibt auch einen Mac Pascal Modus, der weitgehend kompatibel mit Think Pascal und MetroWerks Pascal ist. Darüber hinaus unterstützt Free Pascal Funktionsüberladung, Operatorüberladung, globale Eigenschaften und einige andere zusätzliche Funktionen.

**[::TemplateMDnewLine::]**

### **Erforderlich**

**[::TemplateMDnewLine::]**

x86-Architektur:

Für die 80x86-Version ist mindestens ein 386er Prozessor erforderlich, ein 486er wird jedoch empfohlen.

Die Mac OS X-Version erfordert Mac OS X 10.4 oder höher, wobei die Entwicklerwerkzeuge installiert

sein müssen.

**[::TemplateMDnewLine::]**

PowerPC-Architektur:

Jeder PowerPC-Prozessor ist geeignet. 16 MB RAM sind erforderlich. Es wird erwartet, dass die Mac OS classic Version mit System 7.5.3 und höher funktioniert. Die Mac OS X Version benötigt Mac OS X 10.3 oder später (kann für 10.2.8 oder später kompiliert werden), mit den installierten Entwicklerwerkzeugen. Auf anderen Betriebssystemen läuft Free Pascal auf jedem System, dass das Betriebssystem ausführen kann.

**[::TemplateMDnewLine::]**

ARM-Architektur:

16 MB RAM sind erforderlich. Läuft auf jeder ARM-Linux-Installation.

**[::TemplateMDnewLine::]**

Sparc-Architektur:

16 MB RAM sind erforderlich. Läuft auf jeder Sparc-Linux-Installation (Solaris ist experimentell).

**[::TemplateMDnewLine::]**

**Lizenz**

Die Pakete und die Laufzeitbibliothek stehen unter einer modifizierten Library GNU Public License, um die Verwendung statischer Bibliotheken bei der Erstellung von Anwendungen zu ermöglichen. Der Quellcode des Compilers selbst steht unter der GNU General Public License. Die Quellen sowohl für den Compiler als auch für die Laufzeitbibliothek sind verfügbar; der komplette Compiler ist in Pascal geschrieben.

## Free Pascal Compiler v.3.2.2

**Referenz Handbuch  
Dokument v.3.2.2  
März 2025**

**Übersetzung von Jens Kallup**

**Liste der Tabellen**

**Über diesen Leitfaden**

**Bezeichnungen**

**Syntax Diagramme**

**Über die Sprache Pascal**

### **1. 1. Pascal Zeichen und Symbole**

**1.1. 1.1. Symbole**

**1.2. 1.2. Kommentare**

**1.3. 1.3. Reservierte Schlüsselwörter**

**1.3.1. 1.3.1. Turbo Pascal**

**1.3.2. 1.3.2. Object Pascal**

**1.3.3. 1.3.3. Modifikationen**

**1.4. 1.4. Kennzeichnungen**

**1.5. 1.5. Hinweise für Direktiven**

**1.6. 1.6. Zahlen**

**1.7. 1.7. Bezeichner**

**1.8. 1.8. Zeichenketten**

### **2. 2. Konstanten**

**2.1. 2.1. Gewöhnliche Konstanten**

**2.2. 2.2. Typisierte**

**2.3. 2.3. Ressourcen Zeichenketten**

### **3. 3. Typen**

- 3.1. 3.1. Basis-Typen
  - 3.1.1. 3.1.1. Ordinale Typen
  - 3.1.2. 3.1.2. Ganze Zahlen (Integer)
  - 3.1.3. 3.1.3. Boolesche Typen
  - 3.1.4. 3.1.4. Aufzählungen
  - 3.1.5. 3.1.5. Untermengen
  - 3.1.6. 3.1.6. Zeichen
- 3.2. 3.2. Zeichen-Typen
  - 3.2.1. 3.2.1. Char oder AnsiChar
  - 3.2.2. 3.2.2. WideChar
  - 3.2.3. 3.2.3. Sonstige
  - 3.2.4. 3.2.4. Einzel-Byte Zeichenketten
    - 3.2.4.1. 3.2.4.1. ShortString
    - 3.2.4.2. 3.2.4.2. AnsiString
    - 3.2.4.3. 3.2.4.3. Zeichen-Code Umwandlung
    - 3.2.4.4. 3.2.4.4. RawByteString
    - 3.2.4.5. 3.2.4.5. UTF8String
  - 3.2.5. 3.2.5. Multi-Byte Zeichenketten
    - 3.2.5.1. 3.2.5.1. UnicodeString
    - 3.2.5.2. 3.2.5.2. WideString
  - 3.2.6. 3.2.6. Konstante Zeichenketten
  - 3.2.7. 3.2.7. Nullterminierende Zeichenketten (PChar)
  - 3.2.8. 3.2.8. Zeichenketten-Größen
- 3.3. Strukturierte Typen
  - 3.3.1. Gepackte Struktur-Typen
  - 3.3.2. Array's
    - 3.3.2.1. Statische Array's
    - 3.3.2.2. Dynamische Array's
    - 3.3.2.3. Typen-Kompatibilität dynamischer Array's
    - 3.3.2.4. Constuctor dynamischer Array's
    - 3.3.2.5. Feldkonstanten-Ausdrücke dynamischer Array's
    - 3.3.2.6. Packen und Entpacken eines Array's
  - 3.3.3. Record's
    - 3.3.3.1. Layout und Größe
    - 3.3.3.2. Bemerkungen und Beispiele
  - 3.3.4. Mengen-Typen
  - 3.3.5. Datei-Typen
- 3.4. Zeiger
- 3.5. Forward-Deklarationen
- 3.6. Prozedur-Typen
- 3.7. Variant's
  - 3.7.1. Definition
  - 3.7.2. Variant's in Zuweisungen und Ausdrücken
  - 3.7.3. Variant's im Interface-Teil
- 3.8. Alias-Typen
- 3.9. Verwaltete Typen
- 4. Variablen
  - 4.1. Definition
  - 4.2. Erklärung
  - 4.3. Geltungsbereich
  - 4.4. Initialisierte Variablen
  - 4.5. Initialisierte Variablen mit Standard-Wert

- 4.6. Thread-Variablen
- 4.7. Eigenschaften
- 5. Objekte
  - 5.1. Deklaration
  - 5.2. Abstrakte und Sealed Objekte
  - 5.3. Felder
  - 5.4. Klassen oder statische Felder
  - 5.5. Constructor und Destructor
  - 5.6. Methoden
    - 5.6.1. Erklärung
    - 5.6.2. Methoden-Aufruf
      - 5.6.2.1. Statische Methoden
      - 5.6.2.2. Virtuelle Methoden
      - 5.6.2.3. Abstrakthierte Methoden
      - 5.6.2.4. Klassen-Methoden oder statische Methoden
  - 5.7. Sichtbarkeit
- 6. Klassen
  - 6.1. Klassen-Definitionen
  - 6.2. Abstrakte und Sealed Klassen
  - 6.3. Normale und statische Felder
    - 6.3.1. Normalisierte Felder / Variablen
    - 6.3.2. Klassen-Felder / Variablen
  - 6.4. Klassen - CTOR
  - 6.5. Klassen - DTOR
  - 6.6. Methoden
    - 6.6.1. Erklärung
    - 6.6.2. Aufrufen
    - 6.6.3. Virtuelle Methoden
    - 6.6.4. Klassen - Methoden
    - 6.6.5. Klassen CTOR und DTOR
    - 6.6.6. Statische Klassen - Methoden
    - 6.6.7. Nachrichten - Methoden
    - 6.6.8. Vererbung
  - 6.7. Eigenschaften
    - 6.7.1. Definition
    - 6.7.2. Indezierte Eigenschaften
    - 6.7.3. Array basierte Eigenschaften
    - 6.7.4. Standard - Eigenschaften (public)
    - 6.7.5. Veröffentliche - Eigenschaften (published)
    - 6.7.6. Speicherinformationen
    - 6.7.7. Eigenschaften überschreiben und neu deklarieren
  - 6.8. Klassen - Eigenschaften
  - 6.9. Verschachtelte Typen, Konstanten, und Variablen
- 7. Schnittstellen (Interface's)
  - 7.1. Definition
  - 7.2. Identifikation
  - 7.3. Implementierung
  - 7.4. Vererbung
  - 7.5. Delegation
  - 7.6. COM
  - 7.7. CORBA und andere Schnittstellen
  - 7.8. Referenzzählung



## 8. Generics

- 8.1. Einführung
- 8.2. Get'ter Typ - Definition
- 8.3. Typen - Spezialisierung
- 8.4. Einschränkungen
- 8.5. Kompatibilität zu Delphi
  - 8.5.1. Syntax - Elemente
  - 8.5.2. Einschränken für Record's
  - 8.5.3. Typen - Überladung(en)
  - 8.5.4. Überlegungen für Namensbereiche
- 8.6. Typen-Kompatibilität
- 8.7. Verwendung der eingebauten Funktionen
- 8.8. Überlegungen zum Geltungsbereich
- 8.9. Operator-Überladung und Generics

## 9. Erweiterte Record's

- 9.1. Definition
- 9.2. Erweiterte Record-Aufzähler
- 9.3. Record-Operationen

## 10. Klassen, Record's, und Typen-Helfer

- 10.1. Definition
- 10.2. Einschränkungen bei Klassen Helfer
- 10.3. Einschränkungen bei Record Helfer
- 10.4. Überlegungen zu einfachen Helfer
- 10.5. Anmerkungen zu Umfang und Lebensdauer von Record-Helfer
- 10.6. Vererbung
- 10.7. Verwendung

## 11. Objektorientierte Pascal - Klassen

- 11.1. Einführung
- 11.2. Klassendeklarationen
- 11.3. Formele Deklaration
- 11.4. Instanzen zuteilen und zuordnen
- 11.5. Protokolldefinitionen
- 11.6. Kategorien
- 11.7. Namensumfang und Bezeichner
- 11.8. Selektoren
- 11.9. Der ID Type
- 11.10. Aufzählungen in Objective-C Klassen

## 12. Ausdrücke

- 12.1. Ausdrucks - Syntax
- 12.2. Funktionsaufrufe
- 12.3. Mengen - CTOR
- 12.4. Typ-Casting von Werten
- 12.5. Typ-Casting von Variablen
- 12.6. Sonstige Typ-Casting's
- 12.7. Der @ - Operator
- 12.8. Operatoren
  - 12.8.1. Arithmetische Operatoren
  - 12.8.2. Logische Operatoren
  - 12.8.3. Boolesche Operatoren
  - 12.8.4. Zeichenketten Operatoren
  - 12.8.5. Operatoren bei dynamischen Array's
  - 12.8.6. Mengen - Operatoren

- 12.8.7. Relationale Operatoren
- 12.8.8. Klassen - Operatoren
- 13. Anweisungen
  - 13.1. Einfache Anweisungen
    - 13.1.1. Zuweisungen
    - 13.1.2. Prozeduren - PROCEDURE
    - 13.1.3. Sprungs - Anweisung GOTO
  - 13.2. Strukturierte Anweisungen
    - 13.2.1. Zusammengesetzte Anweisungen
    - 13.2.2. CASE
    - 13.2.3. IF ... THEN
    - 13.2.4. FOR ... TO / DOWNT0 ... DO
    - 13.2.5. FOR .. IN .. DO
    - 13.2.6. REPEAT ... UNTIL
    - 13.2.7. WHILE ... DO
    - 13.2.8. WITH
    - 13.2.9. Ausnahmen (EXCEPT)
  - 13.3. Assembler Anweisungen
- 14. Benutzung von Funktionen und Prozeduren
  - 14.1. FUNCTION Deklarationen
  - 14.2. PROCEDURE Deklarationen
  - 14.3. Funktion Rückgabewert mittels RESULT
  - 14.4. Parameter Listen
    - 14.4.1. Werte
    - 14.4.2. Variablen
    - 14.4.3. Ausgabe Parameter
    - 14.4.4. Konstante Parameter
    - 14.4.5. Offene Array's
    - 14.4.6. Array of Const
    - 14.4.7. Untypisierte Parameter
    - 14.4.8. Verwaltete Typen und Referenzzähler
  - 14.5. Überladung von Funktionen
  - 14.6. Mit FORWARD deklarierte Funktionen
  - 14.7. Externe Funktionen
  - 14.8. Assembler Funktionen
  - 14.9. Modifikatoren
    - 14.9.1. alias
    - 14.9.2. cdecl
    - 14.9.3. cppdecl
    - 14.9.4. export
    - 14.9.5. hardfloat
    - 14.9.6. inline
    - 14.9.7. interrupt
    - 14.9.8. iocheck
    - 14.9.9. local
    - 14.9.10. MS\_ABI\_Default
    - 14.9.11. MS\_ABI\_CDecl
    - 14.9.12. MWPascal
    - 14.9.13. noreturn
    - 14.9.14. nostackframe
    - 14.9.15. overload
    - 14.9.16. pascal

- 14.9.17.public
- 14.9.18.register
- 14.9.19.safecall
- 14.9.20.saveregisters
- 14.9.21.softfloat
- 14.9.22.stdcall
- 14.9.23.SYSV\_ABI\_Default
- 14.9.24.SYSV\_ABI\_CDecl
- 14.9.25.VectorCall
- 14.9.26.varargs
- 14.9.27.winapi
- 14.10.Nicht unterstützte Modifikatoren
- 15.Operatoren Überladung
  - 15.1. Einleitung
  - 15.2. Operatoren - Deklarationen
  - 15.3. Operator - Zuweisung
  - 15.4. Arithmetische Operatoren
  - 15.5. Vergleichende Operatoren
  - 15.6. In Operator
  - 15.7. Logik Operatoren
  - 15.8. Auf- und Ab-Zählungs Operatoren
  - 15.9. Aufzählungs - Operator
- 16.Programme, Module und Blöcke
  - 16.1. Programme
  - 16.2. Module (Unit's)
  - 16.3. Namensräume
  - 16.4. Abhängigkeiten von Modulen
  - 16.5. Blöcke
  - 16.6. Anwendungsbereiche (Scope)
    - 16.6.1. Blöcke
    - 16.6.2. Record's
    - 16.6.3. Klassen
    - 16.6.4. Unit's
  - 16.7. Bibliotheken
- 17.Ausnahmen
  - 17.1. Die RAISE Anweisung
  - 17.2. Ausnahme-Behandlung und Verschachtelung
- 18.Assembler
  - 18.1. Anweisungen
  - 18.2. Prozeduren und Funktionen

---

### Liste der Tabellen

---

### Über dieses Handbuch

---

### Bezeichnungen

---

### Syntax Diagramme

---

### Über die Sprache Pascal

1. Pascal Zeichen und Symbole

2. Konstanten

3. Typen

4. Variablen

5. Objekte

6. Klassen

7. Schnittstellen (Interface's)

8. Generics

9. Erweiterte Record's

10. Klassen, Records, und Typen-Helfer

11. Objektorientierte Pascal - Klassen

12. Ausdrücke

13. Anweisungen

14. Benutzung von Funktionen und Prozeduren

15. Operatoren - Überladung

16. Programme, Module, und Blöcke

17. Ausnahmen

18. Assembler

18.1. Anweisungen

18.2. Prozedure und Funktionen