

# COBOL - COmmon Business Oriented Language

Thorsten Töpper - Hochschule Mannheim Proseminar

9. Juni 2010

## Entstehung

- ▶ Zur Entwicklung kaufmännischer Systeme entwickelt
- ▶ 1959 von Grace Hopper spezifiziert
- ▶ Entwicklung wurde vom *US Department of Defense* (US-Verteidigungsministerium) gesponsort
- ▶ 1960 zum Standard erklärt

## Geschichte

- ▶ 1968: ANS COBOL 68
- ▶ 1974: COBOL 74
- ▶ 1985: COBOL 85
- ▶ 2002: COBOL 2002 - ISO/IEC 1989:2002

## Pro

- ▶ Standard
- ▶ Code für damalige Verhältnisse portabel
- ▶ Auf kaufmännische Applikationen ausgelegt
- ▶ Noch weit verbreitet ⇒ erhöhte Jobsicherheit

## Contra

- ▶ Code nicht wirklich portabel
- ▶ viele zueinander inkompatible Implementierungen
- ▶ viele Dialekte
- ▶ nur globale Variablen
- ▶ Für naturwissenschaftliche Applikationen nicht geeignet

## Syntax

- ▶ Syntax basiert auf Lochkarten

```

Zeilenschablone für Eingabezeilen:

|----|||---||-----||-----|
1      6 7 8    11 12                               72 73    80
Spalten                                Inhalt
1 - 6      wahlfreie Zeilenfolgennummer
            (wird vom Compiler u.U. benutzt)
7          Kommentarzeile (* oder /)
            Fortsetzungszeile (-)
8 - 11     Feld A: Überschriften von
            Hauptteilen
            Abschnitten
            Paragraphen
            Stufennummern (FD,SD,01,77)
12-72      Feld B: Namen in Datenbeschreibungen
            Anweisungen
73-80      wahlfreie Zeilenkennung
            (wird vom Compiler nicht beachtet)

```

Quelle: Einführung in die Programmiersprache COBOL

## Programmaufbau

- ▶ 4 Hauptteile:
  - ▷ Erkennungsteil - IDENTIFICATION DIVISION
  - ▷ Maschinenteil - ENVIRONMENT DIVISION
  - ▷ Datenteil - DATA DIVISION
  - ▷ Verarbeitungsteil - PROCEDURE DIVISION

## Erkennungsteil

- ▶ Angabe von folgenden Informationen:
  - ▷ PROGRAM-ID - Programmname
    - Angabe an erster Stelle Pflicht
    - Format des Namens divergiert bei Compilern
  - ▷ Nicht zwingend nötig:
    - AUTHOR - Name des Programmautors
    - INSTALLATION - Name der Einrichtung
    - DATE-WRITTEN - Datum der Programmerstellung
    - SECURITY - Angabe von Sicherheitsvermerken

## Maschinenteil

- ▶ Nicht zwingend notwendig
- ▶ CONFIGURATION SECTION - Konfigurations-Kapitel:
  - ▷ SOURCE-COMPUTER - Bezeichnung des Computers, auf dem kompiliert wird
  - ▷ OBJECT-COMPUTER - Bezeichnung des Computers, auf dem ausgeführt wird
  - ▷ SPECIAL-NAMES - verschiedene Anpassungen bspw.:
    - DECIMAL-POINT IS COMMA.
- ▶ INPUT-OUTPUT SECTION - Regelung der Ein- und Ausgabe:
  - ▷ FILE-CONTROL - Zuweisung von Geräten zu Dateien
    - SELECT dateiname ASSIGN TO systemname.

## Datenteil

- ▶ Nicht zwingend notwendig
- ▶ Unterteilt in drei Kapitel:
  - ▷ FILE SECTION - Deklaration interner Dateien
  - ▷ WORKING-STORAGE SECTION - Verwaltung des Arbeitsspeichers
    - sn var PICTURE IS datentyp VALUE IS wert.
  - ▷ LINKAGE SECTION - Verbindungskapitel

## Verarbeitungsteil

- ▶ Notwendig
- ▶ Enthält die Prozeduren

## Quellen

- ▶ <http://de.wikipedia.org/wiki/COBOL>
- ▶ <http://en.wikipedia.org/wiki/COBOL>
- ▶ <http://www.opencobol.org/>
- ▶ <http://www.cobolstandards.com/>
- ▶ Alexander Graf, Peter Sandner, Peter Stede - *Einführung in die Programmiersprache COBOL*  
ISBN: 3-411-76481-3

## Code-Beispiel: Fakultätsberechnung

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.  
    FACULTYPGM.  
  
DATA DIVISION.  
FILE SECTION.  
  
WORKING-STORAGE SECTION.  
01  EINGABE.  
    02 eingabewert PIC 99.  
01  FAKULTAET.  
    02 ergebniswert PIC 9(18) VALUE 1.  
  
PROCEDURE DIVISION.  
    main SECTION.  
        DISPLAY "Wert zur Fakultaetsberechnung:"  
        ACCEPT eingabewert.  
        DISPLAY "Wert: " eingabewert.  
  
        PERFORM faculty.  
  
        DISPLAY "Fakultaet: " ergebniswert.  
        STOP RUN.  
  
    faculty SECTION.  
        MULTIPLY eingabewert by ergebniswert.  
        SUBTRACT 1 FROM eingabewert.  
        IF eingabewert > 1 THEN  
            PERFORM faculty.
```