COBOL Datentypen

Typ	Format	Beispiel	Beschreibung
Alphanumerisch	PIC X(n)	01 NAME PIC X(30).	Beliebige Zeichen
Alphabetisch	PIC A(n)	01 LAND PIC A(20).	Nur Buchstaben
Numerisch	PIC 9(n)	01 ALTER PIC 9(3).	Nur Ziffern
Dezimal	PIC 9(n)V9(m)	01 PREIS PIC 9(5)V99.	Virtueller Dezimalpunkt
Signiert	PIC S9(n)	01 SALDO PIC S9(7)V99.	Mit Vorzeichen +/-

Variablen

```
    01 CUSTOMER-DATA.
        05 CUSTOMER-NAME PIC X(30) VALUE SPACES.
        05 CUSTOMER-AGE PIC 9(3) VALUE 0.
```

01 ist die Hauptebene. 05, 10, 15 die Unterebenen.

```
    MOVE "Emma" TO CUSTOMER-NAME. oder MOVE 25 TO CUSTOMER-AGE.
```

Zuweisung von Werten zu Variablen.

Ablaufsteuerung

Statement	Bedeutung	Beispiel
DISPLAY	Ausgabe	DISPLAY "Eingabe (JA/NEIN):".
ACCEPT	Eingabe	ACCEPT ANTWORT.
MOVE	Zuweisung	MOVE "AKTIV" TO STATUS.
IF	Bedingung	IF ANTWORT = "JA" ... ELSE ... END-IF.
PERFORM	Paragraph aufrufen	PERFORM CALC-ABSCHNITT.
STOP RUN	Programm beenden	STOP RUN.

Methoden

```
    PROCEDURE DIVISION.
        BEGIN.
            DISPLAY "Programm gestartet.".
            PERFORM CALC-SUM.
        STOP RUN.

        CALC-SUM.
            MOVE 100 TO SUMME.
            DISPLAY "Summe: " SUMME.
```

01 ist die Hauptebene. 05, 10, 15 die Unterebenen.

```
    MOVE "Emma" TO CUSTOMER-NAME. oder MOVE 25 TO CUSTOMER-AGE.
```

Zuweisung von Werten zu Variablen.

Cobol Functionen

Funktion	Beschreibung	Beispiel
CURRENT-DATE	Aktuelles Datum/Uhrzeit	DISPLAY FUNCTION CURRENT-DATE.
LENGTH(var)	Länge eines Feldes	MOVE FUNCTION LENGTH(NAME) TO LEN.
NUMVAL(str)	String zu Zahl	MOVE FUNCTION NUMVAL("123") TO NUM.
UPPER-CASE(str)	GroSSbuchstaben	MOVE FUNCTION UPPER-CASE(TEXT) TO UPPER.
LOWER-CASE(str)	Kleinbuchstaben	MOVE FUNCTION LOWER-CASE(TEXT) TO LOWER.

Vollständiges Beispiel

```
1     IDENTIFICATION DIVISION.
2     PROGRAM-ID. WORKSHOP-DEMO.
3     ENVIRONMENT DIVISION.
4     CONFIGURATION SECTION.
5     SOURCE-COMPUTER. TECHNISCHE-HOCHSCHULE-MANNHEIM.
6     OBJECT-COMPUTER. TECHNISCHE-HOCHSCHULE-MANNHEIM.
7     DATA DIVISION.
8     WORKING-STORAGE SECTION.
9         01 FRAGE PIC X(30) VALUE "Ist der Kurs vorbei? (JA/NEIN)".
10        01 ANTWORT PIC X(10).
11        01 KURSSTATUS PIC X(15) VALUE "AKTIV".
12    PROCEDURE DIVISION.
13    BEGIN.
14        DISPLAY FRAGE.
15        ACCEPT ANTWORT.
16        IF ANTWORT = "JA"
17            MOVE "FEIERABEND" TO KURSSTATUS
18        DISPLAY "Status: " KURSSTATUS "."
19        ELSE
20            DISPLAY "Der Kurs geht weiter."
21        END-IF.
22        STOP RUN
```

COBOL Cheat Sheet

Hilfestellungen für Aufgabe 1

Datei definieren

```
FILE-CONTROL.
  SELECT VARIABLENNAME
    ASSIGN TO "DATEINAME"
    ORGANIZATION IS EINLESEART.

  FILE SECTION.
    FD VARIABLENNAME.
      01 BUCHUNG-LINE PIC X(80).
```

Datei öffnen und schließen

```
OPEN INPUT VARIABLENNAME.
CLOSE BUCHUNGEN.
```

Eine Zeile einlesen

```
READ BUCHUNGEN
END-READ
```

Berechnungen

```
COMPUTE VARIABLENNAME = X + Y * Z.
```

Mathematische Functionen

```
ADD X TO Y GIVING SUMXY.
SUBTRACT Y FROM X GIVING SUBTRACTION.
MULTIPLY X BY Y GIVING PRODUCT.
DIVIDE X BY Y GIVING QUOTIENT.
```

COBOL Cheat Sheet

Hilfestellungen für Aufgabe 2 (a und b)

Leerzeilen behandeln und Tabs entfernen

```
MOVE FUNCTION TRIM(BUCHUNG-LINE) TO LINE-TRIM.
INSPECT LINE-TRIM REPLACING ALL X"09" BY " ".   *> Tabs -> Spaces

IF LINE-TRIM = SPACES
   OR LENGTH OF FUNCTION TRIM(LINE-TRIM) = 0
   CONTINUE   *> Leere Zeile überspringen
END-IF.
```

Strings per UNSTRING zerlegen

```
MOVE SPACES TO WS-VORNAME WS-NACHNAME
               WS-STUNDEN WS-GEHALT WS-GEBURT.

UNSTRING LINE-TRIM
    DELIMITED BY ALL SPACE
    INTO WS-VORNAME
         WS-NACHNAME
         WS-STUNDEN
         WS-GEHALT
         WS-GEBURT
END-UNSTRING.
```

Strings zusammenbauen (STRING)

```
MOVE FUNCTION TRIM(WS-VORNAME) TO WS-NAME-ZUS.
MOVE 1 TO WS-J.

STRING
    " " DELIMITED BY SIZE
    FUNCTION TRIM(WS-NACHNAME) DELIMITED BY SIZE
    INTO WS-NAME-ZUS
    WITH POINTER WS-J
END-STRING.
```

Längenprüfung eines Strings

```
MOVE LENGTH OF FUNCTION TRIM(WS-GEBURT) TO WS-LEN.

IF WS-LEN NOT = 10
   *> Fehlerfall behandeln
END-IF.
```

Elektronische Zeichenprüfung (INSPECT TALLYING)

```
MOVE 0 TO WS-CNT.

INSPECT ALLOWED-NAME-CHARS
   TALLYING WS-CNT
   FOR ALL WS-NAME-ZUS(WS-I:1).

IF WS-CNT = 0
   *> Sonderzeichen gefunden
END-IF.
```

Numerische Prüfung

```
IF WS-STUNDEN(1:1) = "-"
   *> Negative Werte nicht zulässig
END-IF.

IF WS-WERT NUMERIC
   *> gültige Zahl
ELSE
   *> Fehler: nicht numerisch
END-IF.
```

Mehrfach-Trenner prüfen

```
MOVE 0 TO WS-CNT.
INSPECT WS-STUNDEN TALLYING WS-CNT FOR ALL ".".

IF WS-CNT > 1
   *> zu viele Dezimalpunkte
END-IF.
```

Datum zerlegen und prüfen

```
MOVE WS-GEBURT(1:2) TO WS-TAG.
MOVE WS-GEBURT(4:2) TO WS-MONAT.
MOVE WS-GEBURT(7:4) TO WS-JAHR.

IF WS-MONAT < 1 OR WS-MONAT > 12
   *> Ungültiger Monat
END-IF.

*> Max. Tageszahl je Monat (inkl. Schaltjahrregel)
EVALUATE WS-MONAT
  WHEN 2
     IF (FUNCTION MOD(WS-JAHR 400) = 0)
     OR (FUNCTION MOD(WS-JAHR 4) = 0
         AND FUNCTION MOD(WS-JAHR 100) NOT = 0)
        MOVE 29 TO WS-TAG-MAX
     ELSE
        MOVE 28 TO WS-TAG-MAX
     END-IF
  WHEN 4  MOVE 30 TO WS-TAG-MAX
  WHEN 6  MOVE 30 TO WS-TAG-MAX
  WHEN 9  MOVE 30 TO WS-TAG-MAX
  WHEN 11 MOVE 30 TO WS-TAG-MAX
  WHEN OTHER MOVE 31 TO WS-TAG-MAX
END-EVALUATE.

IF WS-TAG < 1 OR WS-TAG > WS-TAG-MAX
    *> Unmoegliches Datum
END-IF.
```

Fehlerlog schreiben (STRING WRITE)

```
MOVE SPACES TO FEHLER-RECORD.

STRING
    FUNCTION TRIM(LINE-TRIM)
    " ; Fehler: " FUNCTION TRIM(ARG-FELD)
    " - "         FUNCTION TRIM(ARG-MSG)
    INTO FEHLER-RECORD
END-STRING.

WRITE FEHLER-RECORD.
```

Fehlerzähler erhöhen

```
ADD 1 TO CNT-ERROR.
```

COBOL Cheat Sheet

Hilfestellungen für Aufgabe 3

Arrays in Cobol

```
01 FirstArray
  05 ITEM OCCURS 10 TIMES PIC x VALUE "0"
```

So wird ein Array der Länge 10 mit String Variablen definiert. Jedes Element hat als Defaultwert "0".

```
ITEM(3)
```

So wird auf das dritte Element eines Array zugegriffen.

Consolen Eingaben in Cobol

```
ACCEPT EINGABE.
```

Mit *ACCEPT* wird die Eingabe des Users in die Variable EINGABE geschrieben.

Systemcalls in Cobol

```
CALL "SYSTEM" USING "cat 'hello world'"
```

Mit *CALL "SYSTEM"* können direkt im Programm Komandozeilen Befehle aufgerufen werden

Floating Points in Cobol

```
01 FLOATING_POINT PIC 9V9(4)
```

Definiert eine Kommzahl mit vier Nachkommastellen

Zufallszahlen in Cobol

```
COMPUTE RANDOM_NUMBER = FUNCTION RANDOM
```

Weist der Variable *RANDOM_NUMBER* eine zufällige Zahl zwischen 0.0 und 1.0 zu.