Intrinsic-Funktionen

Wachsende Anforderungen an die Software-Entwicklung und die immer komplexer werdenden Aufgaben in Wirtschaft und Administration waren Anlass dafür, die Programmiersprache COBOL um einige Funktionen zu erweitern. Daher wurde der COBOL-Standard ANSI85 um ein weiteres Modul `Intrinsic Functions` erweitert. Es handelt sich dabei um Funktionen für verschiedene Aufgaben, z.B. um mathematische Werte zu ermitteln, um das Datum aufzubereiten, und einige Funktionen zur Zeichenkettenverarbeitung. Die folgende Tabelle zeigt in alphabetischer Reihenfolge diese Funktionen im Überblick.

Liste aller Funktionen

ABS	ACOS	ANNUITY	ASIN
ATAN	CHAR	CHAR-NATIONAL	COS
CURENT-DATE	DATE-OF-INTEGER	DATE-TO-YYYYMMDD	DAY-OF-INTEGER
DAY-TO-YYYYDDD	DISPLAY-OF	E	EXP
EXP10	FACTORIAL	FRACTION-PART	INTEGER
INTEGER-OF-DATE	INTEGER-OF-DAY	INTEGER-PART	LENGTH
LOG	LOG10	LOWER-CASE	MAX
MEAN	MEDIAN	MIDRANGE	MIN
MOD	NATIONAL-OF	NUMVAL	NUMVAL-C
ORD	ORD-MAX	ORD-MIN	PI
PRESENT-VALUE	RANDOM	RANGE	REM
REVERSE	SIGN	SIN	SQRT
STANDARD-DEVIATION	SUM	TAN	UPPER-CASE
VARIANCE	WHEN-COMPILED	YEAR-TO-YYYY	

Tabelle 1: Liste aller `Intrinsic Functions`

Zeichen- und Zeichenkettenfunktionen

CHAR-Funktion

Wirkung

Liefert ein alphanumerisches Zeichen, dessen Positionswert in der verwendeten Codetabelle (ASCII/EBCDI) dem Wert des angegebenen Arguments entspricht.

```
FUNCTION CHAR ( Argument-1 )
```

Abbildung 1: CHAR-Funktion

Erläuterung

Das Argument muss numerisch, größer als 0 und kleiner bzw. gleich der Anzahl der Stellen im verwendeten Code sein.

Durch die Verwendung der ALPHABET-Klausel im SPECIAL-NAMES-Paragraphen kann es sein, dass mehrere Zeichen der gleichen Position in der Codetabelle zugeordnet wurden. In diesem Fall liefert die CHAR-Funktion das zuerst zugeordnete Zeichen laut ALPHABET-Klausel.

Es ist dabei zu beachten, dass die als Argument anzugebende Position das Zählen mit null beginnt.

Beispiel

```
WORKING-STORAGE SECTION.

01 ZEICHEN              PIC X.

PROCEDURE DIVISION.

    MOVE FUNCTION CHAR(50) TO ZEICHEN
```

Listing 1: Beispiel zur CHAR-Funktion

Unter Verwendung des ASCII-Codes liefert dieser Aufruf das Zeichen »1«.

LENGTH-Funktion

Wirkung

Liefert die Länge eines Feldes bzw. einer Datengruppe.

```
FUNCTION LENGTH ( Argument-1 )
```

Abbildung 2: LENGTH-Funktion

Erläuterung

Ist das Argument bzw. eines seiner untergeordneten Datenfelder eine mit OCCURS DEPENDING ON beschriebene Tabelle, wird die aktuelle Anzahl der Elemente aus dem DEPENDING ON-Feld zur Ermittlung der gesamten Argumentlänge herangezogen.

Beispiel

```
WORKING-STORAGE SECTION.

01  TAB.
    05 ANZ                  PIC 9(3) VALUE 50.
    05 ELEMENT              PIC X(5) OCCURS 1 TO 100
                            DEPENDING ON ANZ.
01  LAENGE                  PIC 999.

PROCEDURE DIVISION.

    COMPUTE LAENGE = FUNCTION LENGTH (TAB)
```

Listing 2: Beispiel zur LENGTH-Funktion

Die LENGTH-Funktion liefert in diesem Beispiel die Länge 253.

LOWER-CASE-Funktion

Wirkung

Liefert als Ergebnis das angegebene Argument nach seiner Umwandlung von Groß- in Kleinbuchstaben.

```
FUNCTION LOWER-CASE ( Argument-1 )
```

Abbildung 3: LOWER-CASE-Funktion

Erläuterung

Diese Funktion darf nur alphabetische oder alphanumerische Argumente benutzen.

Beispiel

```
WORKING-STORAGE SECTION.

01 FELD          PIC X(12) VALUE "AbCLMN123?!!".

PROCEDURE DIVISION.

    MOVE FUNCTION LOWER-CASE (FELD) TO FELD
```

Listing 3: Beispiel zur LOWER-CASE-Funktion

Die LOWER-CASE-Funktion wandelt den Inhalt des Feldes in die Zeichenkette abclmnl23?!! um.

NUMVAL-Funktion

Wirkung

Liefert den numerischen Wert einer Zeichenkette, die als Argument angegeben wird.

```
FUNCTION NUMVAL ( Argument-1 )
```

Abbildung 4: NUMVAL-Funktion

Erläuterung

Das Argument kann ein alphanumerisches Literal bzw. ein alphanumerisches Datenfeld sein und maximal 31 beliebige Ziffern, einen dezimalen Punkt bzw. Komma und ein Vorzeichen beinhalten.

Beispiel

Im folgenden Beispiel wird nach einer durch eine ACCEPT-Anweisung erfolgten Eingabe mit den numerischen Werten aus Kapital und Zinssatz gerechnet.

```
WORKING-STORAGE SECTION.

01  KAPITAL                 PIC +zzzzzz,zz.
01  ZINSSATZ                PIC      **.**.
01  ZINSBETRAG              PIC 9(6)V99.

PROCEDURE DIVISION.

    ACCEPT KAPITAL
    ACCEPT ZINSSATZ
    COMPUTE ZINSBETRAG = FUNCTION NUMVAL(KAPITAL) *
                         FUNCTION NUMVAL(ZINSSATZ) /
                         100
    END-COMPUTE
```

Listing 04: Beispiel zur NUMVAL-Funktion

NUMVAL-C-Funktion

Wirkung

Liefert den numerischen Wert einer Zeichenkette, die als Argument angegeben wird, und zusätzlich das Währungssymbol.

Abbildung 5: NUMVAL-C-Funktion

Erläuterung

Diese Funktion hat die gleichen Regeln wie NUMVAL, jedoch darf die Zeichenkette auch ein Währungssymbol beinhalten.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  WERT                    PIC $$$$$.$$.
01  WERT-NUM                PIC 99999.99.

PROCEDURE DIVISION.

    COMPUTE WERT-NUM = FUNCTION NUMVAL-C(WERT)
```

Listing 05: Beispiel zur NUMVAL-C-Funktion

ORD-Funktion

Wirkung

Liefert die Position des angegebenen Arguments aus der COLLATING SEQUENCE des Programms.

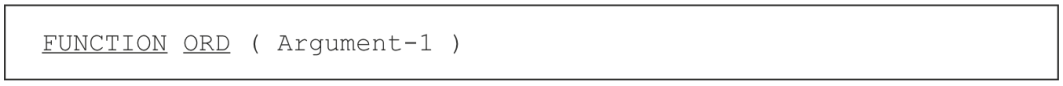

Abbildung 6: ORD-Funktion

Erläuterung

In Anlehnung an die CHAR-Funktion stellt die ORD-Funktion die Umkehrung der CHAR-Funktion dar.

Das Argument darf nur aus einem einzigen Zeichen bestehen. Die kleinste Position, die hierfür geliefert wird, ist 1.

Beispiel

```
WORKING-STORAGE SECTION.

01  ASCII-POSITION       PIC 999.

PROCEDURE DIVISION.

    MOVE FUNCTION ORD("?") TO ASCII-POSITION
```

Listing 6: Beispiel zur ORD-Funktion

Unter Verwendung des ASCII-Codes als `COLLATING SEQUENCE` für dieses Programm liefert dieser Aufruf den Wert 64 als Position für das Zeichen »?«.

REVERSE-Funktion

Wirkung

Die `REVERSE`-Funktion liefert eine invertierte Zeichenkette aus dem angegebenen Argument.

```
FUNCTION REVERSE ( Argument-1 )
```

Abbildung 7: REVERSE-Funktion

Erläuterung

Beim Invertieren einer Zeichenkette wird das letzte Zeichen eines Datenfelds zum ersten und das erste zum letzten Zeichen etc.

Diese Funktion gilt nur für alphabetische oder alphanumerische Datenfelder.

Beispiel

```
WORKING-STORAGE SECTION.

01  NACHNAME               PIC X(10).
01  ZAEHLER                PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

    MOVE FUNCTION REVERSE (NACHNAME) TO NACHNAME
    INSPECT NACHNAME TALLYING ZAEHLER FOR LEADING SPACE
```

Listing 7: Beispiel zur REVERSE-Funktion

In diesem Beispiel wird die Anzahl der Leerzeichen am Anfang des Datenfelds gezählt. Listing 08 zeigt den Ablauf.

```
Inhalt von Nachname (vor MOVE)    : "BERGER    "
Inhalt von Nachname (nach MOVE)   : "    REGREB"
Inhalt von Zaehler  (nach INSPECT): 004
```

Listing 08: Ablauf des Beispiels

Als Alternative für die vorherigen Anweisungen kann auch die INSPECT-Anweisung codiert werden.

```
    INSPECT FUNCTION REVERSE(NACHNAME)
        TALLYING ZAEHLER FOR LEADING SPACE
```

Listing 9: Alternative mit INSPECT-Anweisung

UPPER-CASE-Funktion

Wirkung

Liefert als Ergebnis das angegebene Argument nach seiner Umwandlung von Klein- in Großbuchstaben.

```
FUNCTION UPPER-CASE ( Argument-1 )
```

Abbildung 8: UPPER-CASE-Funktion

Erläuterung

Diese Funktion darf nur alphabetische oder alphanumerische Argumente benutzen.

Beispiel

```
WORKING-STORAGE SECTION.

01  MB        PIC X(30) VALUE "cobol/2 Workbench".

PROCEDURE DIVISION.

    MOVE FUNCTION UPPER-CASE (MB) TO MB
```

Listing 10: Beispiel zur UPPER-CASE-Funktion

Die UPPER-CASE-Funktion wandelt den Inhalt des Feldes MB in die Zeichenkette COBOL/2 WORKBENCH um.

Arithmetische Funktionen

FACTORIAL-Funktion

Wirkung

Die FACTORIAL-Funktion liefert die Fakultät einer Zahl.

```
FUNCTION FACTORIAL ( Argument-1 )
```

Abbildung 9: FACTORIAL-Funktion

Erläuterung

Als Argument-1 muss eine Ganzzahl >= 0 verwendet werden. Ist das Argument = 0, ist die zugehörige Fakultät = 1.

Beispiel

```
WORKING-STORAGE SECTION.

01  WERT                PIC 999 VALUE 5.

PROCEDURE DIVISION.

    MOVE FUNCTION FACTORIAL(WERT) TO RECHENFELD
```

Listing 11: Beispiel für die FACTORIAL-Funktion

Die Funktion liefert 120 im RECHENFELD als Fakultät von 5.

INTEGER-Funktion

Wirkung

Die INTEGER-Funktion liefert die größte Ganzzahl, die kleiner als oder gleich Argument-1 ist.

```
FUNCTION INTEGER ( Argument-1 )
```

Abbildung 10: INTEGER-Funktion

Erläuterung

Diese Funktion wird hauptsächlich zum Abrunden einer Variablen auf eine Ganzzahl verwendet; dabei dürfen nur numerische Variablen benutzt werden.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  BETRAG                   PIC -99.99.

PROCEDURE DIVISION.

    COMPUTE BETRAG = FUNCTION INTEGER (94.12)
    *> Inhalt von Betrag: +9400
```

Listing 12: Beispiel 1 für die INTEGER-Funktion

Beispiel 2

```
WORKING-STORAGE SECTION.

01  BETRAG                   PIC -99.99.

PROCEDURE DIVISION.

    COMPUTE BETRAG = FUNCTION INTEGER (-94.12)
    *> Inhalt von Betrag: -9500
```

Listing 13: Beispiel 2 für die INTEGER-Funktion

INTEGER-PART-Funktion

Wirkung

Die INTEGER-PART-Funktion liefert den ganzzahligen Teil aus einem numerischen Datenfeld.

```
FUNCTION INTEGER-PART ( Argument-1 )
```

Abbildung 11: INTEGER-PART-Funktion

Erläuterung

Der ganzzahlige Teil eines Feldes besteht lediglich aus dessen Vorkommastellen ohne jegliches Auf- bzw. Abrunden und aus dem zur Zahl gehörenden Vorzeichen.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  A                  PIC 9(3)V9(2) VALUE 93.49.

PROCEDURE DIVISION.
```

```
    COMPUTE A = FUNCTION INTEGER-PART(A)
    *> Inhalt von A: 09300
```

Listing 14: Beispiel 1 zur INTEGER-PART-Funktion

Beispiel 2

```
WORKING-STORAGE SECTION.

01  B                   PIC S9(3)V9(2) VALUE -12,33.

PROCEDURE DIVISION.

    COMPUTE B = FUNCTION INTEGER-PART(B)
    *> Inhalt von B: 01200-
```

Listing 15: Beispiel 2 zur INTEGER-PART-Funktion

MAX-Funktion

Wirkung

Die MAX-Funktion liefert den größten Wert aus einer Reihe angegebener Argumente.

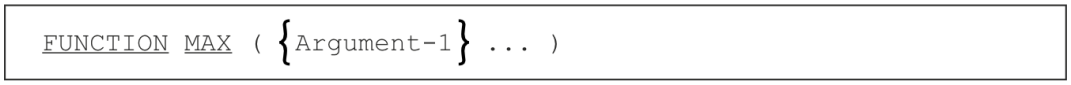

Abbildung 12: MAX-Funktion

Erläuterung

Diese Funktion kann mit Argumenten verschiedener Datenklassen benutzt werden. In einem Aufruf müssen jedoch alle Argumente die gleiche Datenklasse besitzen.

Enthalten zwei oder mehrere Argumente den größten Wert, wird der Inhalt des am weitesten links angegebenen Arguments mit diesem Wert zurückgeliefert.

Beispiel

```
WORKING-STORAGE SECTION.

01  MAXIMUM                PIC -9(03),9(2).
01  A1                      PIC 9(03) VALUE 100.
01  A2                      PIC 9(03) VALUE 500.

PROCEDURE DIVISION.
```

```
    COMPUTE MAXIMUM = FUNCTION MAX (90 A1 -50 A2)
    DISPLAY MAXIMUM
```

Listing 16: Beispiel zur MAX-Funktion

Die Funktion liefert den Wert 500 vom Feld A2.

MIN-Funktion

Wirkung

Die MIN-Funktion liefert den kleinsten Wert aus einer Reihe angegebener Argumente.

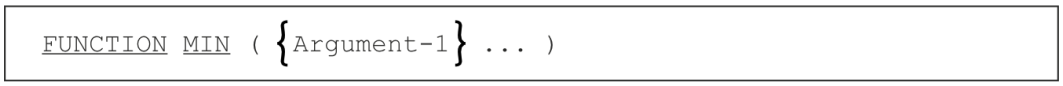

Abbildung 13: MIN-Funktion

Erläuterung

Diese Funktion kann mit Argumenten verschiedener Datenklassen benutzt werden. In einem Aufruf müssen jedoch alle Argumente die gleiche Datenklasse besitzen.

Enthalten zwei oder mehrere Argumente den kleinsten Wert, wird der Inhalt des am weitesten links angegebenen Arguments mit diesem Wert zurückgeliefert.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  MINIMUM                PIC -9(03),9(2).
01  A1                     PIC 9(03) VALUE 100.
01  A2                     PIC 9(03) VALUE 500.

PROCEDURE DIVISION.

    COMPUTE MINIMUM = FUNCTION MIN (90 a1 -50 a2)
    DISPLAY MINIMUM
```

Listing 17: Beispiel 1 zur MIN-Funktion

Die Funktion liefert den Wert -50 aus dem Literal.

Beispiel 2

```
PROCEDURE DIVISION.

    COMPUTE MINIMUM = FUNCTION MIN
```

```
                (FUNCTION INTEGER(-30.7) 90 a1 -70 a2)
    DISPLAY MINIMUM
```

Listing 18: Beispiel 2 zur MIN-Funktion

Die Funktion liefert den Wert -70 aus dem Literal.

MOD-Funktion

Wirkung

Die MOD-Funktion liefert den Divisionsrest, der bei der Division von Argument-1 durch Argument-2 entsteht.

```
FUNCTION MOD ( Argument-1 Argument-2 )
```

Abbildung 14: MOD-Funktion

Erläuterung

Beide Argumente müssen numerische ganzzahlige Datenfelder sein. Argument-2 muss einen Wert ungleich null aufweisen. Es ist ferner zu beachten, dass der Divisionsrest, der hierbei entsteht, nicht vergleichbar ist mit dem Divisionsrest, der durch eine DIVIDE-Anweisung mit dem REMAINDER-Zusatz entsteht. Ursache dafür ist die Arbeitsweise der MOD-Funktion nach folgender Formel:

```
Divisionsrest-Arg-1
    - (Arg-2 * FUNCTION INTEGER(Arg-1 / Arg-2))
```

Beispiel 1

```
    COMPUTE REST = FUNCTION MOD(1987 -4) + 0
    *> ergibt REST = -1
```

Listing 19: Beispiel zur MOD-Funktion

Das lässt sich wie folgt erklären:

```
REST=Arg-1 -(Arg-2 * FUNCTION INTEGER(Arg-1 / Arg-2))
REST= 1987 -(-4    * FUNCTION INTEGER(1987 / -4    ))
REST= 1987 -(-4    * FUNCTION INTEGER( -496,75     ))
REST= 1987 -(-4    *         -497                   )
REST= 1987 -(1988                                   )
REST= -1
```

Listing 20: Ablauf der MOD-Funktion

Bei einer DIVIDE-REMAINDER-Anweisung wird jedoch der Wert +3 geliefert.

```
    DIVIDE 1987 BY -4 GIVING QUOTIENT REMAINDER REST
```

ORD-MAX-Funktion

Wirkung

Liefert die Nummer des größten Arguments aus einer Reihe angegebener Argumente.

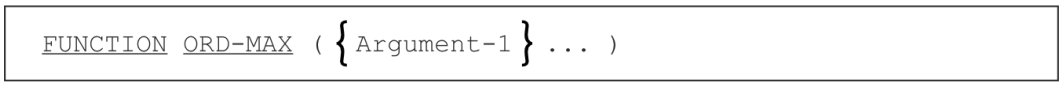

Abbildung 15: ORD-MAX-Funktion

Erläuterung

Intern vergleicht die ORD-MAX-Funktion die angegebenen Argumente nach den Regeln der IF-Anweisung und stellt damit das größte Argument fest.

Nach ANSI85 dürfen entweder nur numerische oder nur nicht-numerische Argumente verwendet werden. MicroFocus erlaubt jedoch das Mischen beider Datenkategorien.

Ist der größte Wert in mehreren Argumenten vorhanden, wird die Nummer des beim Lesen von links zuerst zutreffenden Arguments geliefert.

Beispiel

```
WORKING-STORAGE SECTION.

01  A               PIC 999 VALUE 100.
01  B               PIC 999 VALUE 40.
01  C               PIC 999V99 VALUE 99.99.
01  FELDNR          PIC 9.

PROCEDURE DIVISION.

    COMPUTE FELDNR = FUNCTION ORD-MAX (A B C)
```

Listing 21: Beispiel 1 zur ORD-MAX-Funktion

In der vorliegenden COMPUTE-Anweisung liefert die ORD-MAX-Funktion den Wert 1 als Nummer des größten Arguments (Feld A).

Die Verwendung von ALL

Beispiel

```
WORKING-STORAGE SECTION.

01  TAB.
    05  UMSATZ     PIC 9(5)V99 PACKED-DECIMAL OCCURS 12.
01  ELEMENTNR      PIC 9.
```

```
PROCEDURE DIVISION.

    COMPUTE ELEMENTNR = FUNCTION ORD-MAX (UMSATZ(ALL))
    DISPLAY UMSATZ(ELEMENTNR)
```

Listing 22: Beispiel 2 zur ORD-MAX-Funktion

In diesem Beispiel wird das größte UMSATZ-Element ausgegeben. ALL ersetzt in diesem Fall die Angabe aller 12 Indizes für die Umsatz-Tabelle.

ORD-MIN-Funktion

Wirkung

Liefert die Nummer des kleinsten Arguments aus einer Reihe angegebener Argumente.

```
FUNCTION ORD-MIN ( { Argument-1 } ... )
```

Abbildung 16: ORD-MIN-Funktion

Erläuterung

Intern vergleicht die ORD-MIN-Funktion die angegebenen Argumente nach den Regeln der IF-Anweisung und stellt damit das kleinste Argument fest.

Ist der kleinste Wert in mehreren Argumenten vorhanden, wird die Nummer des zuerst von links gelesenen zutreffenden Arguments geliefert.

Beispiel

```
WORKING-STORAGE SECTION.

01  A            PIC 999 VALUE 100.
01  B            PIC 999 VALUE 40.
01  C            PIC 999V99 VALUE 99.99.
01  FELDNR       PIC 9.

PROCEDURE DIVISION.

    COMPUTE FELDNR =   FUNCTION ORD-MIN (A B C)
```

Listing 23: Beispiel zur ORD-MIN-Funktion

In der vorliegenden COMPUTE-Anweisung liefert die ORD-MIN-Funktion den Wert 2 als Nummer des kleinsten Arguments (Feld B).

RANDOM-Funktion

Wirkung

Die RANDOM-Funktion liefert eine Zufallszahl.

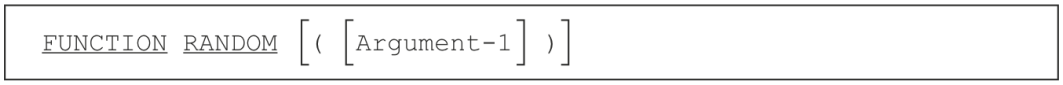

Abbildung 17: RANDOM-Funktion

Erläuterung

RANDOM liefert als Zufallszahl einen Wert größer oder gleich null und kleiner als 1. Wird ein Argument verwendet, muss dieses größer oder gleich null sein.

Beispiel 1:

```
WORKING-STORAGE SECTION.

01  ZUFALLSZAHL             PIC 9(02).9(5).

PROCEDURE DIVISION.

    COMPUTE ZUFALLSZAHL = FUNCTION RANDOM(20)
    DISPLAY ZUFALLSZAHL
```

Listing 24: Beispiel 1 zur RANDOM-Funktion

Liefert beispielsweise 00,58934.

Beispiel 2:

```
PROCEDURE DIVISION.

    COMPUTE ZUFALLSZAHL = FUNCTION RANDOM
    DISPLAY ZUFALLSZAHL
```

Listing 25: Beispiel 2 zur RANDOM-Funktion

Liefert z.B. 00,66856.

RANGE-Funktion

Wirkung

Die RANGE-Funktion liefert die Differenz zwischen dem größten und kleinsten Wert aus einer Reihe von Argumenten (Spannweite).

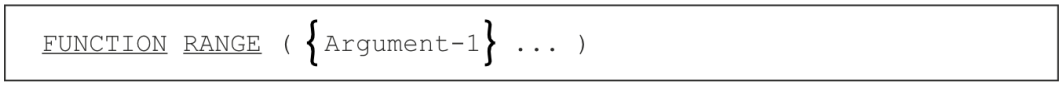

Abbildung 18: RANGE-Funktion

Erläuterung

Alle Argumente müssen numerisch definiert sein.

Beispiel

```
WORKING-STORAGE SECTION.

01  SPANNWEITE     PIC 9(03),9(2).
01  A1             PIC 9(03) VALUE 100.
01  A2             PIC 9(03) VALUE 500.

PROCEDURE DIVISION.

    COMPUTE SPANNWEITE = FUNCTION RANGE(90 A1 -50 A2)
    DISPLAY SPANNWEITE
```

Listing 26: Beispiel zur RANGE-Funktion

Die Funktion liefert den Wert `550,00` als Differenz von 500 und -50.

REM-Funktion

Wirkung

Die REM-Funktion liefert den Divisionsrest im Sinne der DIVIDE REMAINDER-Anweisung.

Abbildung 19: REM-Funktion

Erläuterung

Alle Argumente müssen numerisch definiert sein.

Beispiel

```
    COMPUTE REST - FUNCTION REM(1987 -4) + 0
```

Listing 27: Beispiel zur REM-Funktion

Ergibt `REST = +3`.

Das erklärt sich wie folgt:

```
REST =
  Arg-1 - (Arg-2 * FUNCTION INTEGER-PART(Arg-1 / Arg-2))
REST =
  1987  - (  -4  * FUNCTION INTEGER-PART(1987  / -4   ))
REST =
  1987  - (  -4  * FUNCTION INTEGER-PART(  -496,75    ))
REST =
  1987  - (  -4  *         -496                        )
REST =
  1987  - (   1984                                     )
REST =  +3
```

Listing 28: Ablauf der REM-Funktion

SQRT-Funktion

Wirkung

Die SQRT-Funktion liefert die quadratische Wurzel einer Variablen.

```
FUNCTION SQRT ( Argument-1 )
```

Abbildung 20: SQRT-Funktion

Erläuterung

Als Argument darf nur ein numerisches Datenfeld verwendet werden, dessen Inhalt gleich oder größer null sein muss.

Beispiel

```
WORKING-STORAGE SECTION.

01  QWURZEL           PIC 9(2).

PROCEDURE DIVISION.

    MOVE FUNCTION SQRT(81) TO QWURZEL
    *> Inhalt von QWURZEL: 09
```

Listing 29: Beispiel zur SQRT-Funktion

SUM-Funktion

Wirkung

Die SUM-Funktion liefert die Summe aller angegebenen Argumente.

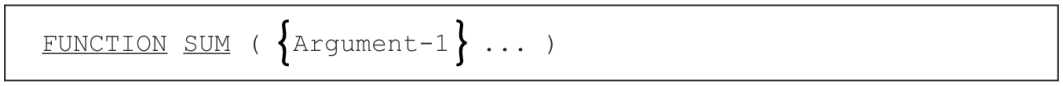

Abbildung 21: SUM-Funktion

Erläuterung

Als Argumente können beliebige numerische Datenfelder benutzt werden; auch die Summe aller Elemente einer Tabelle kann mit einem Aufruf ermittelt werden.

Beispiel

```
WORKING-STORAGE SECTION.

01  GESAMTUMSATZ            PIC 9(03),9(2).
01  TAB                     VALUE '102030405060'.
    05  UMSATZ              PIC 9(2) OCCURS 6.

PROCEDURE DIVISION.

    COMPUTE GESAMTUMSATZ = FUNCTION SUM(UMSATZ(ALL))
    DISPLAY GESAMTUMSATZ
```

Tabelle 2: Beispiel zur SUM-Funktion

Die Funktion liefert den Wert `210` als Summe der Tabelle `TAB`.

Datumsfunktionen

Einige der Intrinsic-Funktionen befassen sich mit Datumsaufbereitung und Datumsumrechnung in verschiedenen Formaten. Grundlegend dafür sind zwei verschiedene Schreibweisen für das Datum: der julianische Kalender (Format: `JJJJTTT`) und der gregorianische Kalender (Format: `JJMMTT`). Für bestimmte Aufgaben benötigt man die Anzahl der zwischen zwei Daten vergangenen Tage; hierzu wurden verschiedene Funktionen vorgesehen, die die Anzahl der Tage ausgehend von einem bestimmten Datum bis zum angegebenen Datum berechnen. Als Ausgangsdatum hat man dabei den 1. Januar 1601 festgelegt. Bei der Berechnung werden selbstverständlich Schaltjahre berücksichtigt.

CURRENT-DATE-Funktion

Wirkung

Die `CURRENT-DATE`-Funktion liefert Systeminformationen wie Tagesdatum, Uhrzeit usw. in einer Zeichenkette mit einer Länge von 21 Byte.

```
FUNCTION CURRENT-DATE
```

Abbildung 22: CURRENT-DATE-Funktion

Erläuterung

Die zu liefernden Informationen sind zum Teil systemabhängig. Wird die Feststellung der Abweichung von der mitteleuropäischen Zeit nicht vom System unterstützt, liefert diese Funktion in den letzten drei Feldern den Wert `00000` zurück. Die Struktur der Zeichenkette wird im nachfolgenden Beispiel gezeigt.

Beispiel

```
WORKING-STORAGE SECTION.

01  DATUM.
    05  JAHR                PIC 9(4).
    05  MONAT               PIC 9(2).
    05  TAG                 PIC 9(2).
    05  STUNDE              PIC 9(2).
    05  MINUTE              PIC 9(2).
    05  SEKUNDE             PIC 9(2).
    05  HUNDERTSTELSEKUNDE  PIC 9(2).
    05  MEZ-ABWEICHUNG-KZ   PIC X(1).
    05  MEZ-STUNDE          PIC 9(2).
    05  MEZ-MINUTE          PIC 9(2).

PROCEDURE DIVISION.

    MOVE FUNCTION CURRENT-DATE TO DATUM
```

Listing 30: Beispiel zur CURRENT-DATE-Funktion

Das Feld `MEZ-ABWEICHUNG-KZ` kann beinhalten:

- Das Zeichen »`0`«: Keine Systemunterstützung für die Abweichung von MEZ.
- Das Zeichen »`-`«: Abweichung von MEZ um die in den zwei letzten Feldern angezeigte Zeit nach hinten.
- Das Zeichen »`+`«: Abweichung von MEZ um die in den zwei letzten Feldern angezeigte Zeit nach vorn.

DATE-OF-INTEGER-Funktion

Wirkung

Die `DATE-OF-INTEGER`-Funktion liefert aus einer Anzahl von Tagen (`Argument-1`) das Tagesdatum im gregorianischen Kalenderformat.

```
FUNCTION DATE-OF-INTEGER ( Argument-1 )
```

Abbildung 23: DATE-OF-INTEGER-Funktion

Erläuterung

Als Argument-1 wird eine Anzahl von Tagen ausgehend vom 1. Januar 1601 angegeben. Der zurückgelieferte Wert ist das Tagesdatum im gregorianischen Format (JJJJMMTT).

Beispiel

```
WORKING-STORAGE SECTION.

01  GRE-DATUM                  PIC 9(8).

PROCEDURE DIVISION.

    COMPUTE GRE-DATUM = FUNCTION DATE-OF-INTEGER (365)
    DISPLAY GRE-DATUM
```

Listing 31: Beispiel zur DATE-OF-INTEGER-Funktion

Liefert bei der Ausgabe: 16011231 als 31. Dezember 1601.

DAY-OF-INTEGER-Funktion

Wirkung

Die DAY-OF-INTEGER-Funktion liefert aus einer Anzahl von Tagen (Argument-1) das Tagesdatum im julianischen Kalenderformat.

```
FUNCTION DAY-OF-INTEGER ( Argument-1 )
```

Abbildung 24: DAY-OF-INTEGER-Funktion

Erläuterung

Als Argument-1 wird eine Anzahl von Tagen ausgehend vom 1. Januar 1601 angegeben. Der zurückgelieferte Wert ist das Tagesdatum im julianischen Format (JJJJTTT).

Beispiel 1:

```
WORKING-STORAGE SECTION.

01  JUL-DATUM                  PIC 9(7).
```

```
PROCEDURE DIVISION.

    COMPUTE JUL-DATUM = FUNCTION DAY-OF-INTEGER (730)
    DISPLAY JUL-DATUM
```

Listing 32: Beispiel zur DAY-OF-INTEGER-Funktion

Liefert bei der Ausgabe: 1602365 als 365. Tag im Jahre 1602.

INTEGER-OF-DATE-Funktion

Wirkung

Die INTEGER-OF-DATE-Funktion liefert die Anzahl der Tage, die zwischen einem im gregorianischen Kalenderformat angegebenen Datum und dem 1.1.1601 liegen.

```
FUNCTION INTEGER-OF-DATE ( Argument-1 )
```

Abbildung 25: INTEGER-OF-DATE-Funktion

Erläuterung

Argument-1 muss im Format (JJJJMMTT) angegeben werden; die Anzahl der Tage wird zum 1. Januar 1601 zurückgerechnet.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  TAGE                        PIC 9(6).

PROCEDURE DIVISION.

    COMPUTE TAGE = FUNCTION INTEGER-OF-DATE(16021231)
    DISPLAY TAGE
```

Listing 33: Beispiel 1 für die INTEGER-OF-DATE-Funktion

Liefert den Wert 730 als Anzahl der Tage zwischen dem 1. Januar 1601 und dem 31. Dezember 1602.

Beispiel 2

Das Beispiel zeigt, wie man die Anzahl der vergangenen Tage zwischen einem ALTDATUM und dem Tagesdatum ermittelt.

```
WORKING-STORAGE SECTION.

01  DATUM-A.
    05 T-JAHR                 PIC 9(2) VALUE 20.
```

```
    05 T-DATUM                PIC 9(6).
01  DATUM REDEFINES DATUM-A   PIC 9(8).
01  ALTDATUM                 PIC 9(8) VALUE 20210203.
01  TAGE                     PIC 9(6).

PROCEDURE DIVISION.

    ACCEPT T-DATUM FROM DATE
    COMPUTE TAGE = FUNCTION INTEGER-OF-DATE (DATUM)
                   FUNCTION INTEGER-OF-DATE(ALTDATUM)
    DISPLAY TAGE
```

Listing 34: Beispiel 2 für die INTEGER-OF-DATE-Funktion

Wird dieses Programm am 4. März 2022 ausgeführt, erhält man den Wert 395 als Anzahl der Tage zwischen dem 3. Februar 2021 und dem 4. März 2022.

INTEGER-OF-DAY-Funktion

Wirkung

Die INTEGER-OF-DAY-Funktion liefert die Anzahl der Tage aus einem im julianischen Kalenderformat angegebenen Datum.

```
FUNCTION INTEGER-OF-DAY ( Argument-1 )
```

Abbildung 26: INTEGER-OF-DAY-Funktion

Erläuterung

Argument-1 muss im Format (JJJJTTT) angegeben werden; die Anzahl der Tage wird zum 1. Januar 1601 zurückgerechnet.

Beispiel

```
WORKING-STORAGE SECTION.

01  TAGE                       PIC 9(6).

PROCEDURE DIVISION.

    COMPUTE TAGE = FUNCTION INTEGER-OF-DAY(1602001)
    DISPLAY TAGE
```

Listing 35: Beispiel für die INTEGER-OF-DAY-Funktion

Liefert den Wert 366 als Anzahl der Tage zwischen dem 1. Januar 1601 und dem 1. Tag im Jahre 1602.

WHEN-COMPILED-Funktion

Wirkung

Die WHEN-COMPILED-Funktion liefert die Uhrzeit und das Tagesdatum der Programmübersetzung.

```
FUNCTION WHEN-COMPILED
```

Abbildung 27: WHEN-COMPILED-Funktion

Erläuterung

Zur Aufnahme des WHEN-COMPILED-Datums muss eine Struktur verwendet werden. Die durch WHEN-COMPILED gelieferte Information liegt in der folgenden Struktur und entspricht somit dem Aufbau der CURRENT-DATE-Struktur:

```
01  CURRENT-DATE-STRUKTUR.
    05  JJJJ PIC 9(4). *> Jahr
    05  MM   PIC 9(2). *> Monat   (01-12)
    05  TT   PIC 9(2). *> Tag     (01-31)
    05  HH   PIC 9(2). *> Stunde  (00-23)
    05  MM   PIC 9(2). *> Minute  (00-59)
    05  SS   PIC 9(2). *> Sekunde (00-59)
    05  HS   PIC 9(2). *> Hundertstel Sekunde (00-99)
    05  MEZ  PIC X(1). *> Kennzeichen für Mez. (+/-)
    05  HH   PIC 9(2). *> Abweichung von Mez. in Stunden
    05  MM   PIC 9(2). *> Abweichung von Mez. in Minuten
```

Listing 36: Aufbau der CURRENT-DATE-Struktur

Beispiel

```
WORKING-STORAGE SECTION.

01  DATUM-ZEIT.
    05  JAHR                PIC 9(4).
    05  MONAT               PIC 9(2).
    05  TAG                 PIC 9(2).
    05  STUNDE              PIC 9(2).
    05  MINUTE              PIC 9(2).
    05  SEKUNDE             PIC 9(2).
    05  HUNDERTSTELSEKUNDE  PIC 9(2).
    05  MITTEL-EURO-ZEIT    PIC X(1).
    05  MEZ-STUNDE          PIC 9(2).
    05  MEZ-MINUTE          PIC 9(2).

PROCEDURE DIVISION.
```

```
    MOVE FUNCTION WHEN-COMPILED TO DATUM-ZEIT
    DISPLAY DATUM-ZEIT
```

Listing 37: Beispiel zur WHEN-COMPILED-Funktion

Statistische Funktionen

MEAN-Funktion

Wirkung

Die MEAN-Funktion liefert den arithmetischen Durchschnitt einer Reihe von Argumenten.

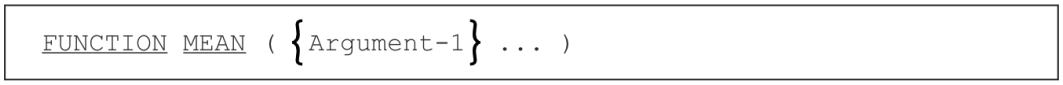

Abbildung 28: MEAN-Funktion

Erläuterung

Als Durchschnitt bezeichnet man das Ergebnis, das bei der Division der Summe aller Argumente durch ihre Anzahl entsteht. Dabei dürfen nur numerische Argumente verwendet werden.

Beispiel

```
WORKING-STORAGE SECTION.

01  SCHNITT                PIC +9(03),9(2).
01  TAB VALUE              '102030405060'.
    05  UMSATZ             PIC 9(2) OCCURS 6.

PROCEDURE DIVISION.

    COMPUTE SCHNITT = FUNCTION MEAN (UMSATZ(ALL))
    DISPLAY SCHNITT
```

Listing 38: Beispiel für die MEAN-Funktion

Liefert den Wert 35 als Durchschnitt aller Elemente der Tabelle.

MEDIAN-Funktion

Wirkung

Die MEDIAN-Funktion liefert den Zentralwert aus einer Reihe von Argumenten.

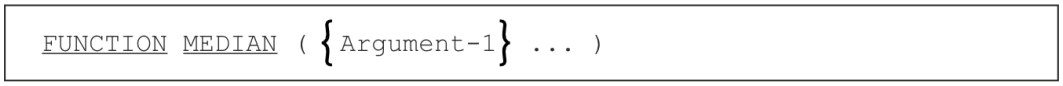

Abbildung 29: MEDIAN-Funktion

Erläuterung

Diese Funktion gilt nur für numerische Datenfelder. Bei einer ungeraden Anzahl von Argumenten ist der Zentralwert der Wert des mittleren Arguments, wenn alle intern sortiert sind.

Ist die Anzahl der Argumente gerade, wird der Durchschnitt der mittleren zwei Argumente als Zentralwert geliefert.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  A1                      PIC 9(03) VALUE 100.
01  A2                      PIC 9(03) VALUE 500.
01  ZENTRALWERT             PIC 9(03),9(2).

PROCEDURE DIVISION.

    COMPUTE ZENTRALWERT =
            FUNCTION MEDIAN(30 A1 A2 70,30 5)
    DISPLAY ZENTRALWERT
```

Listing 39: Beispiel 1 für die MEDIAN-Funktion

Diese Funktion liefert den Wert `70,30` als Zentralwert der Argumente (`5`, `30`, `70.30`, `100`, `500`).

Beispiel 2

```
WORKING-STORAGE SECTION.

01  TAB2                    VALUE '992233884400'.
    05  ELEM2               PIC 9(2) OCCURS 6.
01  ZENTRALWERT             PIC 9(03),9(2).

PROCEDURE DIVISION.

    COMPUTE ZENTRALWERT = FUNCTION MEDIAN(ELEM2(ALL))
    DISPLAY ZENTRALWERT
```

Listing 40: Beispiel 2 für die MEDIAN-Funktion

Diese Funktion liefert den Wert `38,50` als Zentralwert der Argumente (`00`, `22`, `33`, `44`, `88`, `99`), errechnet als Durchschnitt der mittleren Argumente `33` und `44`.

MIDRANGE-Funktion

Wirkung

Die MIDRANGE-Funktion liefert als Rückgabewert den Durchschnittswert aus den minimalen und maximalen Werten einer Reihe von Argumenten (mittlere Spannweite).

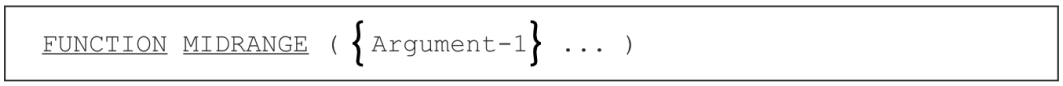

Abbildung 30: MIDRANGE-Funktion

Beispiel 1

```
WORKING-STORAGE SECTION.

01  SCHNITT               PIC +9(03),9(2).
01  WERTE                 PIC 9(12) VALUE 100200300400.
01  TAB REDEFINES WERTE.
    05  UMSATZ            PIC 9(3) OCCURS 4.
01  TAB2 REDEFINES WERTE.
    05  UMS               OCCURS 2.
        10  UMSATZ2        PIC 9(3)   OCCURS 2.

PROCEDURE DIVISION.

    COMPUTE SCHNITT = FUNCTION MIDRANGE (100 500 300)
    *> Inhalt von SCHNITT: +300,00
    *> Entspricht dem Durchschnitt von 100 + 300

    COMPUTE SCHNITT = FUNCTION MIDRANGE (UMSATZ(ALL))
    *> Inhalt von SCHNITT: +250,00
    *> Entspricht dem Durchschnitt von 100 + 400

    COMPUTE SCHNITT = FUNCTION MIDRANGE (UMSATZ2(ALL 2))
    *> Inhalt von SCHNITT: +300,00
    *> Entspricht dem Durchschnitt von 200 + 400

    COMPUTE SCHNITT =
            FUNCTION MIDRANGE (UMSATZ2(ALL ALL))
    *> Inhalt von SCHNITT: +250,00
    *> Entspricht dem Durchschnitt von 100 + 400
```

Listing 41: Beispiel für die MIDRANGE-Funktion

STANDARD-DEVIATION-Funktion

Wirkung

Die STANDARD-DEVIATION-Funktion liefert die Standardabweichung.

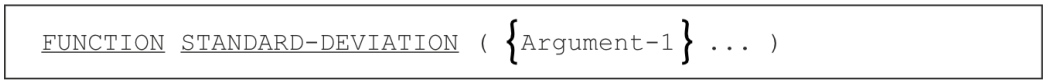

Abbildung 31: STANDARD-DEVIATION-Funktion

Erläuterung

Die Standardabweichung wird nach folgendem Algorithmus ermittelt:

Ermittlung des Durchschnitts aller Argumente. Jede Differenz aus Durchschnitt und jeweils einem Argument wird mit 2 potenziert. Die Summe aller Potenzen wird addiert und durch die Anzahl der Argumente dividiert.

Es wird nun die quadratische Wurzel aus dem vorherigen Quotienten ermittelt. Wird nur ein Argument verwendet, wird die Standardabweichung 0 geliefert.

Beispiel

```
WORKING-STORAGE SECTION.

01  ZIELE                  VALUE '25762'.
    05  ZIEL               PIC 9(1) OCCURS 5.
01  STANDARD-ABWEICHUNG    PIC 9(03),9(2).

PROCEDURE DIVISION.

    COMPUTE STANDARD-ABWEICHUNG =
            FUNCTION STANDARD-DEVIATION (ZIEL(ALL))
    DISPLAY STANDARD-ABWEICHUNG
```

Listing 42: Beispiel für die STANDARD-DEVIATION-Funktion

Diese Funktion liefert den Wert 2,05. Der Wert wurde ermittelt wie folgt:

```
2+5+7+6+2 = 22/5 = 4,4 Durchschnitt
4,4 - 2 =  2.4       ** 2 = 5,76
4,4 - 5 = -0,6 – 0,6 ** 2 = 0,36
4,4 - 7 = -2,6 - 2,6 ** 2 = 6,76
4,4 - 6 = -1,6 - 1,6 ** 2 = 2,56
4,4 - 2 =  2,4       ** 2 = 5,76
                          ------
                           21,20 / 5 = 4,24
```

Listing 43: Ablauf der STANDARD-DEVIATION-Funktion

Die Quadratwurzel aus 4,24 = 2,05.

VARIANCE-Funktion

Wirkung

Die VARIANCE-Funktion liefert die Varianz einer Reihe von Argumenten.

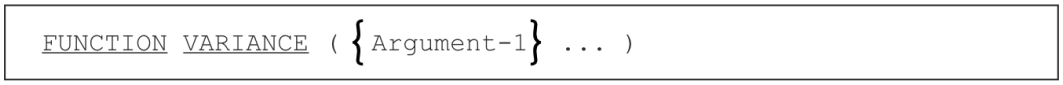

Abbildung 32: VARIANCE-Funktion

Erläuterung

Als Varianz einer Reihe von Argumenten bezeichnet man die Standardabweichung, die mit 2 potenziert worden ist (siehe auch STANDARD-DEVIATION in Abschnitt STANDARD-DEVIATION-Funktion).

Beispiel 1:

```
WORKING-STORAGE SECTION.

01  ZIELE                VALUE '25762'.
    05  ZIEL             PIC 9(1) OCCURS 5.
01  VARIANZ              PIC 9(04),9(2).

PROCEDURE DIVISION.

    COMPUTE VARIANZ = FUNCTION VARIANCE (ZIEL(ALL))
    DISPLAY VARIANZ
```

Listing 44: Beispiel für die VARIANCE-Funktion

Die Funktion liefert als Ergebnis den Wert 4,24.

Trigonometrische Funktionen

In COBOL wurden innerhalb der Intrinsic-Funktionen einige arithmetische Funktionen zur Abwicklung spezieller Aufgaben aufgenommen. Hierbei handelt es sich um trigonometrische und Arcus-Funktionen (Umkehrfunktionen der trigonometrischen Funktionen). Bei all diesen Funktionen müssen die verwendeten Argumente numerische Datenfelder sein.

ACOS-Funktion

Wirkung

Die ACOS-Funktion liefert den Arcuskosinus aus Argument-1.

```
FUNCTION ACOS ( Argument-1 )
```

Abbildung 33: ACOS-Funktion

Erläuterung

Argument-1 muss größer gleich -1 und kleiner gleich +1 sein.

Beispiel

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION ACOS(-1)
    DISPLAY WERT
    *> Inhalt von WERT:  003,141592
```

Listing 45: Beispiel für die ACOS-Funktion

ASIN-Funktion

Wirkung

Die ASIN-Funktion liefert den Arcussinus aus Argument-1.

```
FUNCTION ASIN ( Argument-1 )
```

Abbildung 34: ASIN-Funktion

Erläuterung

Argument-1 muss größer gleich -1 und kleiner gleich +1 sein.

Beispiel

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION ASIN(-1)
    DISPLAY WERT
    *> Inhalt von WERT:  001,570796
```

Listing 46: Beispiel für die ASIN-Funktion

ATAN-Funktion

Wirkung

Die ATAN-Funktion liefert den Arcustangens aus Argument-1.

```
FUNCTION ATAN ( Argument-1 )
```

Abbildung 35: ATAN-Funktion

Erläuterung

Der Rückgabewert ist immer größer als -n/2 und kleiner als +π/2.

Beispiel

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION ATAN(1)
    DISPLAY WERT
    *> Inhalt von WERT:  000,785398
```

Listing 47: Beispiel für die ATAN-Funktion

COS-Funktion

Wirkung

Die COS-Funktion liefert den Kosinus aus Argument-1.

```
FUNCTION COS ( Argument-1 )
```

Abbildung 36: COS-Funktion

Erläuterung

Der Rückgabewert ist größer gleich -1 und kleiner gleich +1.

Beispiel

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.
```

```
    COMPUTE WERT = FUNCTION COS(1)
    DISPLAY WERT
    *> Inhalt von WERT:  000,540302
```

Listing 48: Beispiel für die COS-Funktion

LOG-Funktion

Wirkung

Die LOG-Funktion liefert den natürlichen Logarithmus (zu Basis e) aus Argument-1.

```
FUNCTION LOG ( Argument-1 )
```

Abbildung 37: LOG-Funktion

Erläuterung

Argument-1 muss größer als null sein.

Beispiel

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION LOG(10)
    DISPLAY WERT
    *> Inhalt von WERT:  002,302585
```

Listing 49: Beispiel für die LOG-Funktion

LOG10-Funktion

Wirkung

Die LOG10-Funktion liefert den Logarithmus zur Basis 10 aus Argument-1.

```
FUNCTION LOG10 ( Argument-1 )
```

Abbildung 38: LOG10-Funktion

Erläuterung

Argument-1 muss größer als null sein.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION LOG10(1000)
    DISPLAY WERT
    *> Inhalt von WERT:  003,000000
```

Listing 50: Beispiel für die LOG10-Funktion

SIN-Funktion

Wirkung

Die SIN-Funktion liefert den Sinus aus Argument-1.

```
FUNCTION SIN ( Argument-1 )
```

Abbildung 39: SIN-Funktion

Erläuterung

Argument-1 muss im Bogenmaß angegeben werden. Der Rückgabewert ist immer größer gleich -1 und kleiner gleich +1.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION SIN(1)
    DISPLAY WERT
    *> Inhalt von WERT:  000,841470
```

Listing 51: Beispiel für die SIN-Funktion

TAN-Funktion

Wirkung

Die TAN-Funktion liefert den Tangens aus Argument-1.

```
FUNCTION TAN ( Argument-1 )
```

Abbildung 40: TAN-Funktion

Erläuterung

Argument-1 muss im Bogenmaß angegeben werden.

Beispiel 1

```
WORKING-STORAGE SECTION.

01  Wert             PIC -9(03),9(6).

PROCEDURE DIVISION.

    COMPUTE WERT = FUNCTION TAN(1)
    DISPLAY WERT
    *> Inhalt von WERT:  001,557407
```

Listing 52: Beispiel für die TAN-Funktion