

Inhaltsverzeichnis

| | |
|---|----|
| Verzweigen mit Hilfe der Ablaufsteuerung | 1 |
| Verzweigen mit Hilfe der Ablaufsteuerung: Einführung..... | 1 |
| Verwenden von Steuerpaaren | 2 |
| If / End If..... | 2 |
| Else If / End Else If..... | 4 |
| Else / End Else..... | 6 |
| While / End While..... | 8 |
| Do / Until | 10 |
| Select / End Select..... | 12 |
| Case / End Case | 15 |
| Default Case / End Default Case | 16 |
| Allgemeine Schleifen verwenden..... | 17 |
| Verwendungszwecke für Schleifen | 20 |
| So erstellen Sie eine Schleife | 21 |
| Verwenden des Druckbefehls in einer Schleife | 21 |
| Sprungmarken verwenden..... | 24 |
| Springen zu einer Sprungmarke mit GOTO | 25 |
| Schleife zurücksetzen..... | 27 |
| Verzweigen bei einem Fehler | 28 |
| Unterstützte Schnittstellen BEI FEHLER..... | 31 |
| Verzweigung bei Unterprogrammen | 32 |
| Erstellen eines neuen Unterprogramms..... | 33 |

| | |
|---|----|
| Bearbeiten eines vorhandenen Unterprogramms | 38 |
| Hinweise zum Dialogfeld "Argument bearbeiten" | 39 |
| Aufrufen eines Unterprogramms | 40 |
| Beispiele für Unterprogramme | 48 |
| Messroutine beenden | 55 |

Verzweigen mit Hilfe der Ablaufsteuerung

Verzweigen mit Hilfe der Ablaufsteuerung: Einführung

Angenommen, Sie haben ein Werkstück mit vielen Elementen, möchten jedoch nur einige der Elemente wiederholt messen, um umfangreiche statistische Daten für diese Elemente zu sammeln. Außerdem möchten Sie abhängig von Benutzereingaben zu einem bestimmten Werkstück in der Messroutine springen. Diese und viele ähnliche Aufgaben können Sie mit Hilfe der Ablaufregelungsbefehle ausführen. Durch das Festlegen von Bedingungen für bestimmte Befehle können Sie den Ablauf der Messroutine steuern.

In diesem Abschnitt erhalten Sie die erforderlichen Informationen für die Ausführung solcher Aufgaben. Es werden syntaxbedingte Anweisungen, Schleifen und Unterprogramme erklärt. Außerdem enthält dieser Abschnitt viele Codebeispiele.

In diesem Abschnitt behandelte Hauptthemen:

- Verwenden von Steuerpaaren
- Allgemeine Schleifen verwenden
- Sprungmarken verwenden
- Verzweigen bei einem Fehler
- Verzweigung bei Unterprogrammen
- Messroutine beenden

Befehlsmodus-Befehle nach Kommentaren

Da viele der Codebeispiele in diesem Kapitel getippte **KOMMENTAR**-Befehle verwenden, sollten Sie Folgendes beachten:



Nachdem Sie einen PC-DMIS-Kommentar eingefügt haben, müssen Sie, um weitere PC-DMIS-Befehle im Befehlsmodus einzugeben, zunächst *zweimal* die Eingabetaste nach dem Befehl `KOMMENTAR` drücken. Dadurch wird PC-DMIS mitgeteilt, dass Sie dem Kommentar keinen Text mehr hinzufügen möchten, sondern bereit sind, einen neuen Befehl hinzuzufügen.

Verwenden von Steuerpaaren

Das Untermenü **Einfügen | Programmablaufsteuerung | Steuerpaare** enthält verschiedene Befehlspaare, die im Bearbeitungsfenster zur Regelung bzw. Steuerung des ordnungsgemäßen Ablaufs der Messroutine dienen. Um im Bearbeitungsfenster einen Steuerpaarbefehl einzufügen, geben Sie einfach den Befehl ein oder wählen Sie einen Befehl in diesem Untermenü aus.



Wird eine bedingte Verzweigungsanweisung zur Wertigkeitsprüfung eines JA/NEIN-Kommentars verwendet, beachten Sie bitte, dass bei dieser Prüfung ein "JA"- oder "NEIN"-Wert in Großbuchstaben gesucht werden sollte. Ein "Ja" oder "Nein" in Kleinbuchstaben wird nicht funktionieren. Weitere Informationen hierzu finden Sie unter "Einfügen von Programmiererkommentaren" im Abschnitt "Einfügen von Protokollbefehlen".

If / End If

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | If / End If** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen IF und END IF stehenden Elemente werden nur dann ausgeführt, wenn der Ausdruck für den Befehl IF als wahr (nicht Null) ausgewertet wird. Andernfalls springt der Ausführungsablauf zum ersten Befehl nach dem `END/IF`-Befehl.

Die im Bearbeitungsfenster für eine IF / END IF-Anweisung angezeigte Befehlszeile lautet:

`IF/Ausdruck`

`END_IF/`


So fügen Sie die Befehle If / End If ein:

Verwenden von Steuerpaaren

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster.
2. Wählen Sie die Option **If / End If** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung IF / END_IF .

Codebeispiel für If / End If

Sehen Sie sich das folgende Beispiel an, in dem der Benutzer gefragt wird, ob er ein Punktelement messen möchte.



```
C1= KOMMENTAR/JANEIN,Möchten Sie das Punktelement PKT1
messen?

IF/C1.INPUT=="JA"

    PKT1=ELEM/PUNKT,KART
    ...
    ...

ENDEMESS/

END_IF/
```

Erläuterung des Beispielcodes

C1=KOMMENTAR/JANEIN

Diese Zeile nimmt und speichert die Eingabe JA oder NEIN des Benutzers.

IF/C1.EINGABE=="JA"

Diese Zeile ist der Ausdruck. Er testet, ob die Eingabe von Kommentar 1 ein JA ist. Lautet die Eingabe JA, dann ist die IF-Anweisung WAHR, und es werden die auf die IF-Anweisung folgenden Anweisungen ausgeführt, in diesem Fall wird das Element **PKT1** gemessen. Wenn NEIN bewegt es sich zum Ausdruck END_IF.

END_IF

Diese Zeile beendet die Ausführung von Befehlen innerhalb des IF / END IF-Codeblocks. Wenn der Benutzer auf den Kommentar mit **Nein** antwortet, springt PC-DMIS zum nächsten Befehl.

Else If / End Else If

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | Else If / End Else If** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen ELSE IF und END ELSE IF stehenden Elemente werden nur dann ausgeführt, wenn der Ausdruck für den Befehl ELSE IF als wahr (nicht Null) ausgewertet wird. Der Block ELSE IF / END ELSE IF muss direkt *nach* einem IF / END IF-Block oder einem anderen ELSE IF / END ELSE IF-Block eingefügt werden. Wenn alle IF / ELSE IF-Ausdrücke oberhalb des aktuellen Blocks als falsch ausgewertet werden, dann wird der Ausdruck ausgewertet. Wird der Ausdruck als falsch (Null) ausgewertet, springt die Ausführung zum nächsten Befehl nach dem END ELSE IF-Befehl. Wenn einer der IF /ELSE IF-Ausdrücke oberhalb des aktuellen Blocks als wahr ausgewertet wird, werden alle nachfolgenden ELSE IF / END ELSE IF-Blöcke dieser Folge übersprungen.

Die im Bearbeitungsfenster für eine ELSE IF / END ELSE IF-Anweisung angezeigte Befehlszeile lautet:

```
ELSE_IF/Ausdruck
```

```
END_ELSE_IF/
```

So fügen Sie die Befehle ELSE IF / END ELSE IF ein:


1. Platzieren Sie den Cursor an der gewünschten Stelle im Bearbeitungsfenster nach einer vorhandenen IF/END IF-Anweisung oder einer ELSE IF/END ELSE IF-Anweisung.
2. Wählen Sie die Option **Else If / End Else If** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung ELSE_IF / END ELSE_IF .



Diese Art von Block ist nur gültig, wenn sie *nach* einem IF / END IF- oder ELSE IF / END ELSE IF-Block positioniert wird. Falsch positionierte Steuerpaare werden im Bearbeitungsfenster rot angezeigt.

Codebeispiel für Else If / End Else If

Sehen Sie sich das folgende Beispiel an. Darin wird der Benutzer durch eine Meldung informiert, wenn einer der X-, Y-, oder Z-Werte des Messpunkts außerhalb der definierten Toleranzbereiche liegt:



```
PNT2=FEAT/POINT,RECT
...
...
ENDMEAS/
IF/PNT2.X<6.9 OR PNT2.X>7.1
    COMMENT/OPER,"The measured X value of PNT2: " + PNT2.X
    + " is out of tolerance."
END_IF/
ELSE_IF/PNT2.Y<3.3 OR PNT2.Y>3.5
    COMMENT/OPER,"The measured Y value for PNT2: " +
    PNT2.Y + " is out of tolerance."
END_ELSEIF/
ELSE_IF/PNT2.Z<.9 OR PNT2.Z>1.1
    COMMENT/OPER,"The measured Z value for PNT2: " +
    PNT2.Z + " is out of tolerance."
END_ELSEIF/
```

Erläuterung des Beispielcodes

Dieser Code testet zuerst den X-Wert des Punkts. Wenn die Bedingung nicht erfüllt (falsch) ist, testet der Code den Y-Wert. Wenn die Bedingung für den Y-Wert nicht erfüllt (falsch) ist, wird der Z-Wert getestet.

Wenn eine der Bedingungen erfüllt (wahr) ist, zeigt PC-DMIS den entsprechenden Kommentar an und überspringt die restlichen bedingten Anweisungen.

IF/PNT2.X7.1

Diese Zeile ist der Ausdruck. Er testet, ob der gemessene X-Wert kleiner als 6,9 oder größer als 7,1 ist. Wenn der Wert außerhalb dieses Bereichs liegt, wird der erste Kommentar ausgeführt.

END_IF

Diese Zeile beendet die Ausführung von Befehlen innerhalb des IF / END IF-Codeblocks. Wenn die IF THEN-Bedingung nicht erfüllt (falsch) ist, springt PC-DMIS zum nächsten Befehl.

ELSE_IF/PNT2.Y3.5

Diese Zeile ist der Ausdruck für den ersten ELSE_IF-Befehl. Er wird nur dann ausgeführt, wenn der vorherige IF / END IF-Block als falsch ausgewertet wurde. Er testet, ob der gemessene Y-Wert kleiner als 3.3 oder größer als 3.5 ist. Wenn der Wert außerhalb dieses Bereichs liegt, wird der zweite Kommentar ausgeführt.

END_ELSEIF/

Diese Zeile beendet die Ausführung von Befehlen innerhalb des ELSE IF/ END ELSE IF-Codeblocks.

ELSE_IF/PNT2.Z1.1

Diese Zeile ist der Ausdruck für den zweite ELSE_IF-Befehl. Er wird nur dann ausgeführt, wenn der vorherige ELSE IF / END ELSE IF-Block als falsch ausgewertet wurde. Er testet, ob der gemessene Z-Wert kleiner als 0,9 oder größer als 1,1 ist. Wenn der Wert außerhalb dieses Bereichs liegt, wird der dritte Kommentar ausgeführt.

END_ELSEIF/

Diese Zeile beendet die Ausführung von Befehlen innerhalb des zweiten ELSE IF/ END ELSE IF-Codeblocks.

Else / End Else

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | Else / End Else** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen ELSE und END ELSE stehenden Elemente werden nur ausgeführt, wenn alle anderen If / End If- und Else If / End Else If-Blöcke oberhalb des Else-Blockes fehlgeschlagen sind (alle mit Null ausgewertet wurden). Die Blöcke ELSE / END ELSE sind nur dann gültig, wenn sie sich am Ende eines Satzes von IF / END IF- oder ELSE IF / END ELSE IF-Blöcken befinden.

Die im Bearbeitungsfenster für eine ELSE / END ELSE-Anweisung angezeigte Befehlszeile lautet:

ELSE/


END_ELSE/

So fügen Sie die Befehle Else / End Else ein:

1. Setzen Sie den Cursor an die gewünschte Stelle im *Bearbeitungsfenster*. Beachten Sie, dass ELSE / END ELSE-Blöcke nach einem IF / END IF- oder ELSE IF / END ELSE IF-Block positioniert werden müssen.
2. Wählen Sie die Option **Else / End Else** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung ELSE / END ELSE.

Codebeispiel für Else / End Else

Sehen Sie sich das folgende Beispiel an, in dem der Benutzer gefragt wird, ob er ein Punktelement messen möchte.



```
C1= KOMMENTAR/JANEIN,Möchten Sie das Punktelement PKT1
messen? Durch Klicken auf 'Nein' wird das nächste Element
gemessen.
IF/C1.INPUT=="YES"
    PNT1=FEAT/POINT,RECT
    ...
    ...
    ENDMEAS/
END_IF/
ELSE
    PNT2=FEAT/POINT,RECT
    ...
    ...
    ENDMEAS/
END_ELSE
```

Erläuterung des Beispielcodes

C1=KOMMENTAR/JANEIN

Diese Zeile nimmt und speichert die Eingabe JA oder NEIN des Benutzers.

IF/C1.EINGABE=="JA"

Diese Zeile ist der Ausdruck. Er testet, ob die Eingabe von Kommentar 1 ein JA ist. Lautet die Eingabe JA, dann ist die IF-Anweisung WAHR, und es werden die auf die IF-Anweisung folgenden Anweisungen ausgeführt, in diesem Fall wird das Element **PKT1** gemessen. Wenn NEIN bewegt es sich zum Ausdruck END_IF.

END_IF

Diese Zeile beendet die Ausführung von Befehlen innerhalb des IF / END IF-Codeblocks. Wenn der Benutzer auf den Kommentar mit **Nein** antwortet, springt PC-DMIS zum nächsten Befehl.

ELSE

Wenn der obige IF / END IF-Block als falsch ausgewertet wird, werden die Befehlszeilen, die zwischen dieser Zeile und der END_ELSE-Zeile liegen, ausgeführt. In diesem Fall wird PKT2 ausgeführt.

END_ELSE

Diese Zeile beendet die Ausführung von Befehlen innerhalb des ELSE / END_ELSE-Codeblocks.

While / End While

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | While / End While** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen WHILE und END WHILE stehenden Elemente werden solange in einem Schleifendurchlauf ausgeführt, bis die Bedingung (oder der Ausdruck), die (der) die Schleife aktiviert hält, nicht mehr erfüllt wird; d. h., der Ausdruck für die While-Schleife wird als FALSCH ausgewertet (d. h. Null). Der Befehl WHILE kann an einer beliebigen Stelle in die Messroutine eingefügt werden. Der Ausdruck wird zu Beginn jeder Schleife geprüft.

Die im Bearbeitungsfenster für eine WHILE / END WHILE-Anweisung angezeigte Befehlszeile lautet:

```
WHILE/Ausdruck
```

```
END_WHILE/
```

So fügen Sie eine While / End While-Option ein:

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster.
2. Wählen Sie die Option **While / End While** aus der Menüleiste aus. Die Anweisung WHILE / END WHILE wird im Bearbeitungsfenster angezeigt.

Codebeispiel für While / End While

Sehen Sie sich das folgende Beispiel an. Darin wird ein Element so oft gemessen, wie vom Benutzer der Messroutine vorgegeben.

Verwenden von Steuerpaaren

```
C1=KOMMENTAR/EINGABE,Wie oft soll PKT1 gemessen werden?  
Geben Sie nur eine Ganzzahl ein.
```

```
ASSIGN/COUNT=0
```

```
WHILE/COUNT<C1.INPUT
```

```
    PKT2=ELEM/PUNKT,KART
```

```
    ...
```

```
    ...
```

```
    ...
```

```
ENDEMESS/
```

```
ASSIGN/COUNT=COUNT+1
```

```
COMMENT/OPER,"Gemessen "+COUNT+" von "+C1.INPUT+"  
Mal."
```

```
END_WHILE/
```

Erläuterung des Beispielcodes

C1=KOMMENTAR/EINGABE

Diese Zeile nimmt und speichert die vom Benutzer eingegebene Ganzzahl in der Variablen `C1.INPUT`.

ASSIGN/COUNT=0

Diese Zeile initialisiert `ZÄHLER`, eine benutzerdefinierte Variable, und weist ihr den Ausgangswert 0 zu. Der Code verwendet diese Variable, um zu zählen, wie oft PC-DMIS das Element innerhalb der Schleife misst.

WHILE/COUNT<C1.INPUT

Diese Zeile ist der Ausdruck. Er testet, ob der Wert von `ZÄHLER` (hatte den Ausgangswert 0) kleiner als die vom Benutzer eingegebene Ganzzahl ist. Wird der Test als wahr ausgewertet, werden die Anweisungen zwischen `WHILE/` und `END_WHILE/` ausgeführt.

ASSIGN/COUNT=COUNT+1

Diese Zeile erhöht den Wert der Variablen `ZÄHLER` um 1, wenn die Bedingung nicht mehr erfüllt ist, wird die Schleife schließlich beendet.

```
COMMENT/OPER, "Gemessen "+COUNT+" von "+C1.INPUT+" Mal."
```

Diese Zeile zeigt eine Nachricht an, in der angegeben wird, wie oft die Schleife bisher ausgeführt wurde.

```
END_WHILE
```

Diese Zeile beendet die Ausführung von Befehlen innerhalb des `WHILE` / `END WHILE`-Codeblocks, wenn die Bedingung nicht erfüllt (falsch) ist. Andernfalls springt PC-DMIS bei Antreffen dieses Befehls zur `WHILE`-Anweisung zurück.

Do / Until

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | Do / Until** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen DO und UNTIL stehenden Elemente werden solange in einem Schleifendurchlauf ausgeführt, bis der Ausdruck des Befehls UNTIL als WAHR (nicht Null) ausgewertet wird. Der Befehl DO / UNTIL kann an einer beliebigen Stelle in die Messroutine eingefügt werden. Der Ausdruck wird am Ende jeder Schleife geprüft.

Die im Bearbeitungsfenster für eine DO / UNTIL-Anweisung angezeigte Befehlszeile lautet:

```
DO/
```

```
UNTIL/ Ausdruck
```

So fügen Sie DO / UNTIL-Befehle ein:

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster.
2. Wählen Sie die Option **Do / Until** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung DO / UNTIL.

Codebeispiel für Do / Until

Sehen Sie sich das folgende Beispiel an. Darin wird ein Element so oft gemessen, wie vom Benutzer der Messroutine vorgegeben. Dieses Beispiel ist jenem für While / End While ähnlich. Der einzige Unterschied liegt darin, dass PC-DMIS die Bedingung hier am Ende der Schleife testet anstatt am Anfang.

Verwenden von Steuerpaaren



```
C1=      COMMENT/INPUT,Type the number of times PC-DMIS
should measure the PNT1 feature:(type an integer only)
ASSIGN/COUNT=0 DO/
    PNT1=FEAT/POINT,RECT
    ...
    ...
ENDMEAS/
ASSIGN/COUNT=COUNT+1
COMMENT/OPER,"Measured "+COUNT+" out of "+C1.INPUT+"
times."
UNTIL/COUNT==C1.INPUT
```

Erläuterung des Beispielcodes

C1=KOMMENTAR/EINGABE

Diese Zeile nimmt und speichert die vom Benutzer eingegebene Ganzzahl in der Variablen `C1.INPUT`.

ASSIGN/COUNT=0

Diese Zeile initialisiert `ZÄHLER`, eine benutzerdefinierte Variable, und weist ihr den Ausgangswert 0 zu. Der Code verwendet diese Variable, um zu zählen, wie oft PC-DMIS das Element innerhalb der Schleife misst.

DO/

Beginnt die `DO / UNTIL`-Schleife. Alle Anweisungen werden mindestens einmal ausgeführt und die Schleife wird erst beendet, wenn der Ausdruck als falsch ausgewertet wird.

ASSIGN/COUNT=COUNT + 1

Diese Zeile erhöht den Wert der Variablen `ZÄHLER` um 1, wenn die Bedingung nicht mehr erfüllt ist, wird die Schleife schließlich beendet.

KOMMENTAR/BEDIENER,"Messwert " + ZÄHLER + " von " + C1.INPUT + " Male."

Diese Zeile zeigt eine Nachricht an, in der angegeben wird, wie oft die Schleife bisher ausgeführt wurde.

UNTIL/ZÄHLER == C1.INPUT

Diese Zeile beendet die Ausführung des Befehls innerhalb der **DO / UNTIL**-Schleife, wenn die Bedingung nicht erfüllt (falsch) ist. Andernfalls springt PC-DMIS bei Antreffen dieses Befehls zur **DO**-Anweisung zurück.

Select / End Select

Mit der Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | Select / End Select** können Sie einen Bedingungsblock hinzufügen, der in Verbindung mit den Steuerpaaren **Case / End Case** und **Default Case / End Default Case** verwendet wird. Der Ausdruck für den Befehl "Auswählen" enthält Daten, die mit dem Ausdruck in den Case-Anweisungen verglichen werden. Wenn die beiden Ausdrücke dasselbe Ergebnis haben, werden die Anweisungen im **Case / End Case**-Block ausgeführt. Der **SELECT / END SELECT**-Block umklammert die Sätze mit den **CASE / END CASE**- und **DEFAULT CASE / END DEFAULT CASE**-Blöcken.

Die im Bearbeitungsfenster für eine **SELECT / END SELECT**-Anweisung angezeigte Befehlszeile lautet:

```
SELECT/Ausdruck
```

```
END_SELECT/
```

So fügen Sie die Befehle **Select / End Select** ein:

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster.
2. Wählen Sie die Option **Select / End Select** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung **SELECT / END_SELECT**.

Codebeispiel für Select / End Select

Die Steuerpaare **SELECT / END_SELECT**, **CASE / END_CASE** und **DEFAULT_CASE / END_DEFAULTCASE** können gemeinsam eingesetzt werden, um mehrere Bedingungen auszuwerten. Dadurch ergibt sich eine ganze Reihe von Alternativen.

Angenommen, Sie haben fünf Kreise (**KREIS1** bis **KREIS5**) und möchten, dass der Bediener mit einem einfachen Tastendruck auf der Tastatur einen Kreis messen kann. Verwenden Sie hierfür diesen oder einen ähnlichen Code:

Vollständiger Code

```
DO/  
  
    C1=KOMMENTAR/EINGABE,Geben Sie eine Zahl für den zu  
    messenden Kreis ein:  
  
    ,Für KREIS1 - Drücken Sie 1.  
  
    ,Für KREIS2 - Drücken Sie 2.  
  
    ,Für KREIS3 - Drücken Sie 3.  
  
    ,Für KREIS4 - Drücken Sie 4.  
  
    ,Für KREIS5 - Drücken Sie 5.  
  
    ,Jede andere Taste beendet die Schleife.  
  
SELECT/C1.INPUT  
  
    CASE/1  
  
        KREIS1=ELEM/KREIS  
  
        ...  
  
        ...  
  
    ENDEMESS/  
  
END_CASE  
  
CASE/2  
  
    KREIS2=ELEM/KREIS  
  
    ...  
  
    ...  
  
    ENDEMESS/  
  
END_CASE
```

```
CASE/3

    KREIS3=ELEM/KREIS

    ...

    ...

    ENDEMESS/

END_CASE

CASE/4

    KREIS4=ELEM/KREIS

    ...

    ...

    ENDEMESS/

END_CASE

CASE/5

    KREIS5=ELEM/KREIS

    ...

    ...

    ENDEMESS/

END_CASE

DEFAULT CASE

    KOMMENTAR/BEDIENER,Schleife wird beendet.

END_DEFAULT CASE

END_SELECT

UNTIL C1.INPUT < 1 ODER C1.INPUT > 5
```


Erläuterung des Beispielcodes

`SELECT/C1.INPUT`

Diese Codezeile nimmt eine Zahl oder einen Wert an (in diesem Fall eine Zahl), die der Benutzer eingibt, und bestimmt anhand der Eingabe, welcher `CASE/END_CASE`-Block ausgeführt wird. Beachten Sie, dass das Steuerpaar `SELECT/END_SELECT` die gesamte Codeliste umschließt. Alle `CASE/END_CASE`- und `DEFAULT_CASE/END_DEFAULTCASE`-Steuerpaare müssen innerhalb dieser beiden Zeilen liegen.

`END_SELECT`

Diese Zeile kennzeichnet das Ende des Codes innerhalb des `SELECT/END_SELECT`-Steuerpaars.

`CASE/1 bis CASE/5`

Je nach Wert von `C1.INPUT` wird einer der CASE-Codeblöcke ausgeführt. Wenn `C1.INPUT` beispielsweise als 1 ausgewertet wird, wird der Codeblock CASE 1 ausgeführt und KREIS1 gemessen. Wenn die Auswertung 2 ergibt, wird der Codeblock CASE 2 ausgeführt und KREIS2 gemessen usw.

`END_CASE`

Diese Zeilen beenden die einzelnen CASE-Codeblocks.

`DEFAULT CASE`

Wenn der Wert von `C1.INPUT` mit keiner der definierten CASE-Anweisungen übereinstimmt (wenn der Wert keine Zahl zwischen 1 und 5 ist), wird der Codeblock `DEFAULT CASE` ausgeführt. In diesem Fall werden Sie in einer Meldung darüber informiert, dass die Schleife beendet wird.

Beachten Sie, dass die `DO / UNTIL`-Schleife den gesamten Beispielcode umschließt. Dadurch kann der Benutzer im Menü, das durch die Zeile `KOMMENTAR/EINGABE` erstellt wird, immer wieder eine Auswahl vornehmen, bis er ein Zeichen eingibt, das von den CASE-Anweisungen nicht erkannt wird.

Case / End Case

Über die Menüoption **Einfügen | Programmablaufsteuerung | Steuerpaare | Case / End Case** können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen CASE und END CASE stehenden Elemente werden ausgeführt, wenn die

Auswertung des Ausdrucks für die CASE-Anweisung einen Wert ergibt, der dem Ausdruck des entsprechenden SELECT- (AUSWAHL-) Befehls entspricht. Andernfalls werden die im Block enthaltenen Anweisungen übersprungen. Der Block mit den Anweisungen CASE / END CASE muss direkt nach einem SELECT- oder einem END CASE-Befehl eines vorhergehenden CASE / END CASE-Blocks stehen. Außerdem kann PC-DMIS nicht mehrere Ausdrücke in einer einzigen CASE-Anweisung vergleichen.

Die im Bearbeitungsfenster für eine CASE / END CASE-Anweisung angezeigte Befehlszeile lautet:

CASE/Ausdruck

END_CASE/

So fügen Sie die Option Case / End Case ein:

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster. Beachten Sie die oben beschriebenen Positionierungsvoraussetzungen.
2. Wählen Sie die Option **Case / End Case** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung CASE / End CASE.

Default Case / End Default Case

Über die Menüoption Einfügen | Programmablaufsteuerung | Steuerpaare | Default Case / End Default Case können Sie die Messroutine einen bedingten Block hinzufügen. Die zwischen den Befehlen DEFAULT CASE und END DEFAULT CASE stehenden Elemente werden ausgeführt, wenn alle anderen Ausdrücke in vorhergehenden CASE / END CASE-Blöcken innerhalb des entsprechenden SELECT / END SELECT-Blocks als falsch ausgewertet werden. Innerhalb eines SELECT/ END SELECT-Blocks ist nur ein DEFAULT CASE / END DEFAULT CASE-Block zulässig. Der Block DEFAULT CASE / END DEFAULT CASE muss nach sämtlichen CASE / END CASE-Blöcken innerhalb des Blocks SELECT / END SELECT stehen.

Die im Bearbeitungsfenster für eine DEFAULT CASE / END DEFAULT CASE-Anweisung angezeigte Befehlszeile lautet:

DEFAULT CASE/

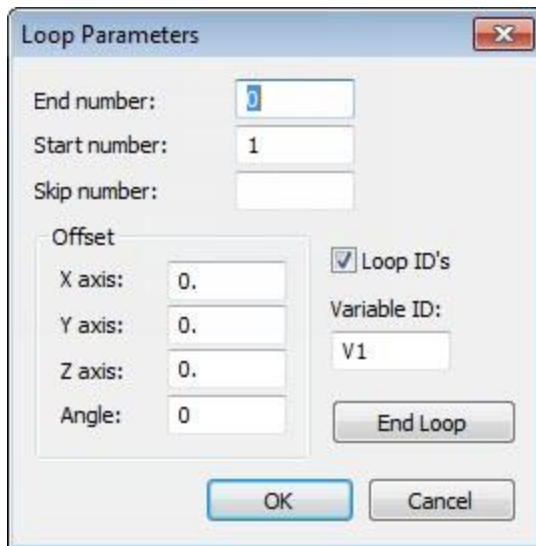
END_DEFAULT_CASE/

So fügen Sie DEFAULT CASE/END DEFAULT CASE-Befehle ein:

Allgemeine Schleifen verwenden

1. Setzen Sie den Cursor an die gewünschte Stelle im Bearbeitungsfenster, und beachten Sie dabei die oben beschriebenen Einschränkungen in Bezug auf die Positionierung.
2. Wählen Sie die Option **Default Case / End Default Case** aus der Menüleiste aus. Im Bearbeitungsfenster erscheint die Anweisung DEFAULT CASE / END_DEFAULT CASE.

Allgemeine Schleifen verwenden



Dialogfeld Schleifen

Die Menüoption **Einfügen | Programmablaufsteuerung | Schleifen** öffnet das Dialogfeld **Schleifen**. Sie können mittels dieses Dialogfeldes einen Befehl **SCHLEIFE** erstellen, der die Messroutine (oder Teile davon) mit oder ohne Versatz wiederholt. Der Befehl **SCHLEIFE** kann an jeder beliebigen Stelle in der Messroutine eingefügt werden, obwohl sich diese Funktion jedoch am Anfang und Ende der Messroutine am hilfreichsten erweist.

Startnummer - Dieses Feld teilt PC-DMIS die Startnummer innerhalb einer Reihe von Werkstücken mit.



Angenommen es liegen 10 Werkstücke in X- (Y- oder Z-) Richtung vor und Sie möchten mit der Position Nummer 5 beginnen. Dann müssen Sie 10 für die Gesamtanzahl der Werkstücke und 5 für die Startposition eingeben.

Endnummer - Dieses Feld definiert wie oft PC-DMIS die Messroutine ausführen soll. Diese Zahl entspricht in der Regel der Anzahl der von der Spannvorrichtung gehaltenen Werkstücke (oder Werkstückmuster) in X-, Y- oder Z-Richtung. PC-DMIS fordert außerdem zur Angabe der Startnummer des Werkstücks (Musters) auf.



Angenommen es liegen 10 Werkstücke in X-Richtung vor und Sie möchten mit der Position Nummer 5 beginnen. Geben Sie im Feld **Endnummer** 10 ein. Für das Feld **Startnummer** wählen Sie 5.

Nummer überspringen - PC-DMIS wiederholt ein Messroutine für die definierte Anzahl. Mit dem Feld **Nummer überspringen** kann ein vorgegebenes Inkrement übersprungen werden. Beachten Sie, dass Sie das *erste* Inkrement nicht mit Hilfe dieses Feldes *überspringen* können. Sie haben jedoch die Möglichkeit, die **Startnummer** auf 2 zu setzen und somit die Startnummer auf diese Weise zu überspringen.



Angenommen Sie wollen die Parameter so einstellen, dass jedes dritte Inkrement der Schleife übersprungen wird. Wird der Wert "3" eingegeben, misst PC-DMIS das erste und zweite Werkstück und überspringt das dritte, so dass es beim vierten angelangt.

Versatz - Der Bereich Versatz enthält folgende Felder:

X-Achse, Y-Achse, Z-Achse - In diesen Feldern wird der X-,Y- oder Z-Versatz zwischen Werkstücken oder Mustern auf demselben Werkstück festgelegt. Diese Werte versetzen das Werkstück um die definierten Versatzabstände, sobald die Schleife ausgeführt wird.

Winkel - Dieses Feld definiert den Winkelversatz zwischen Werkstücken oder Werkstückmustern. Der erste Versatz basiert auf dem Nullpunkt des Werkstücks. PC-DMIS versetzt das Werkstück um den Winkelwert, sobald die Ausführungsschleife durchlaufen wird.

Die Software basiert die Versatzabstände auf dem Ursprung des Werkstücks. Beim ersten Durchlauf durch die Schleife verwendet PC-DMIS den Ursprung des

Allgemeine Schleifen verwenden

Werkstücks. Beim zweiten Durchgang durch die Schleife bewegt er sich um die Versatzabstände. Beim dritten Mal durch die Schleife bewegt er sich um die mit zwei multiplizierten Versatzabstand, und so weiter.



Wenn ein Ausrichtungsbefehl innerhalb einer Schleife vorliegt und in der Schleife mit Versätzen gearbeitet wird, *müssen* alle Achsen für diese Ausrichtung durch Sie definiert werden. Außerdem muss die Ausrichtung innerhalb der Schleife die in der Schleife gemessenen Elemente verwenden.

Schleifen-IDs - Dieses Kontrollkästchen bestimmt, ob PC-DMIS den aktuellen Schleifenwert in eckigen Klammern in der Element-ID anzeigt.



Angenommen, Sie haben ein Kreis-Element von KREIS1. Es wird beim ersten Mal durch die Schleife zu KREIS1[1], beim zweiten Mal durch die Schleife zu KREIS1[2], und so weiter.

Element-IDs in Statistik-Datenbanken

Wenn Sie das Kontrollkästchen **Schleifen-IDs** aktivieren und Statistikdaten an eine Datenbank übertragen, löscht PC-DMIS diese Schleifen-IDs in einigen Fällen aus der Datenbank.

Beachten Sie folgendes:

Wenn Sie einen `STAT/EIN`-Befehl und einen `STAT/AKTUALISIEREN`-Befehl *innerhalb eines Schleifenblocks* besitzen, werden keine Schleifen-IDs in der Datenbank angezeigt.

Wenn Sie einen `STAT/EIN`-Befehl *außerhalb eines Schleifenblocks* und einen `STAT/AKTUALISIEREN`-Befehl *innerhalb eines Schleifenblocks* besitzen, werden keine Schleifen-IDs in der Datenbank angezeigt.

Wenn Sie einen `STAT/EIN`-Befehl *außerhalb eines Schleifenblocks* und einen `STAT/AKTUALISIEREN`-Befehl *außerhalb eines Schleifenblocks* besitzen, werden keine Schleifen-IDs in der Datenbank angezeigt.

Wenn Sie einen `STAT/EIN`-Befehl *innerhalb eines Schleifenblocks* und einen `STAT/AKTUALISIEREN`-Befehl *außerhalb eines Schleifenblocks* besitzen, werden keine Schleifen-IDs in der Datenbank angezeigt.

Variablen-ID - In diesem Feld können Sie den Variablennamen festlegen, unter dem die aktuelle Schleifenwiederholung (bzw. die aktuelle Schleife innerhalb der angegebenen Schleifenanzahl) verfolgt wird. Während der Ausführung der Messroutine entspricht diese Variable der aktuellen Iterationsnummer der Schleife.

Schleife beenden - Mit dieser Schaltfläche wird der Schleifendurchlauf beendet. Auf den Befehl [SCHLEIFE/ANFANG](#) muss im Bearbeitungsfenster der Befehl [SCHLEIFE/ENDE](#) folgen.

Verwendungszwecke für Schleifen

Es gibt drei Hauptverwendungszwecke für die Option Schleifen:

- Sie besitzen eine aus mehreren Teilen bestehende Spannvorrichtung, die ein Raster von Werkstücken hält. Die Spannvorrichtung sollte einen gleichmäßigen Abstand zwischen den Reihen aufweisen. Mit Hilfe der Verschiebungs-/Rotationsversätze können Sie im Werkstückraster von einem Werkstück zum nächsten voranrasten.
- Sie besitzen eine Spannvorrichtung, die ein Werkstück hält, und Sie möchten vor jeder Schleife der Messroutine zu einem neuen Werkstück wechseln. In diesem Fall können Sie einen Befehl [KOMMENTAR](#) dazu verwenden, das KMG beim Wechsel zu einem neuen Werkstück anzuhalten. Der Befehl kann sowohl am Anfang als auch am Ende der Schleife stehen.
- Sie möchten die Schleifenfunktionalität nutzen, um die Messroutine zu drehen, um einen anderen Teil desselben Werkstücks zu messen. Sie könnten beispielsweise eine Messroutine zur Messung eines komplizierten Lochmusters, das sich zehn Mal auf dem Werkstück wiederholt, erstellen. Die Messroutine muss die Lochmuster dann nur einmal messen. Sie könnten dann die Schleifenfunktionalität nutzen, um die Messroutine zu versetzen, um die anderen neun Vorkommnisse des Musters zu messen.



Wenn Sie eine Ausrichtung innerhalb einer Schleife verwenden, können Sie mit PC-DMIS eine aktive Ausrichtung in der Befehlszeile [AUSRICHTUNG/START](#) verwenden, anstatt stets eine zuvor gespeicherte Ausrichtung aufzurufen. Siehe das Thema „Verwenden einer Ausrichtung innerhalb von Schleifen“ im Abschnitt „Erstellen und Verwenden von Ausrichtungen“.

So erstellen Sie eine Schleife

1. Wählen Sie in der Menüleiste die Option **Einfügen | Programmablaufsteuerung | Schleifen** aus. Daraufhin erscheint das Dialogfeld Schleifen.
2. Definieren Sie die Werte in dem Dialogfenster.
3. Geben Sie die jeweils erforderlichen Parameter an (z. B. **Werkstückzahl**, **Startnummer**, **Überspringe Nummer**, **Versatzwinkel**).
4. Setzen Sie den Cursor im Bearbeitungsfenster an die Stelle, wo der Schleifendurchlauf beginnen soll.
5. Klicken Sie auf die Schaltfläche **OK**, um den Befehlsblock `SCHLEIFE/START` einzufügen.

Die im Bearbeitungsfenster für den Schleifendurchlauf angezeigte Befehlszeile lautet:



```
VARNAME = LOOP/START, ID = Y/N, NUMBER = 0, START  
        = 1, SKIP = , OFFSET: XAXIS = 0, YAXIS = 0, ZAXIS =  
0, ANGLE = 0
```



Um den Schleifendurchlauf abzuschließen, benötigen Sie innerhalb des Bearbeitungsfensters den Befehl `SCHLEIFE/ENDE`. PC-DMIS wiederholt jeden Befehl im Bearbeitungsfenster, der durch die Befehle `SCHLEIFE/START` und `SCHLEIFE/ENDE` eingeschlossen ist.

Sie können den Befehl `SCHLEIFE/ENDE` auf eine der folgenden Arten einfügen:

- Geben Sie im Bearbeitungsfenster `SCHLEIFE/ENDE` ein.
- Wählen Sie die Menüoption **Einfügen | Programmablaufsteuerung | Schleife beenden** aus.
- Klicken Sie im Dialogfeld **Schleifen** auf die Schaltfläche **Schleife beenden**.

Verwenden des Druckbefehls in einer Schleife

Um den Befehl `DRUCKEN` in einer Schleife in PC-DMIS zu verwenden, tun Sie dies:

1. Fügen Sie die Befehle **SCHLEIFE/ANFANG** und **SCHLEIFE/ENDE** in Ihre Messroutine ein.
2. Fügen Sie den Befehl **DRUCKEN** zwischen den Schleifenbefehlen dort ein, wo Sie ihn wünschen.
3. Klicken Sie im Bearbeitungsfenster auf den Befehl **DRUCKEN**. Drücken Sie dann die Taste F9, um das Dialogfeld **Ausgabekonfiguration** für diesen Befehl **DRUCKEN** zu öffnen. Einzelheiten zur Einstellung der Ausgabe- und Druckeroptionen für das Report-Fenster finden Sie unter "Ausgabe- und Druckereinstellungen für das Bearbeitungsfenster" im Abschnitt "Verwenden von grundlegenden Dateioptionen".
4. Wählen Sie eine der Registerkarten für das Ausgabeformat (**Protokoll**, **DMIS** oder **Excel**), um das Format für Ihren Befehl **DRUCKEN** zu definieren.
5. Aktivieren Sie das Kontrollkästchen oben links auf der Registerkarte, um die Optionen für die Ausgabe zu aktivieren.
6. Legen Sie fest, wohin das generierte Protokoll gesendet werden soll:
 - a. Klicken Sie auf die Schaltfläche "Durchsuchen" (...) rechts neben dem Feld für Pfad- und Dateinamen, um das Dialogfeld **Speichern unter** anzuzeigen.
 - b. Legen Sie im Dialogfeld **Speichern unter** den Dateinamen fest und klicken Sie auf **Speichern**.
7. Nehmen Sie alle weiteren Änderungen vor, die Sie benötigen. Wenn Sie z. B. jedes Protokoll nach der Generierung durch PC-DMIS anzeigen möchten, aktivieren Sie das Kontrollkästchen **Protokoll anzeigen**.
8. Wenn Sie Ausgaben in mehreren Formaten erzeugen möchten, drücken Sie die nächste Registerkarte und wiederholen Sie die Schritte 4 bis 7.
9. Klicken Sie auf **OK**, um Ihre Änderungen im Befehl **DRUCKEN** zu speichern.

Der Befehl **DRUCKEN** funktioniert innerhalb von Schleifen fast genauso wie außerhalb von Schleifen. Der Unterschied besteht darin, dass PC-DMIS intern verfolgt, was ausgegeben wurde, und nur Daten ausgibt, die nach der Ausführung des letzten Befehls **DRUCKEN** ausgeführt wurden.



Betrachten Sie zum Beispiel den folgenden Codeausschnitt. Wenn PC-DMIS den Befehl **DRUCKEN** zum ersten Mal ausführt, erstellt die Software die erste Ausgabedatei mit *allen* bis dahin erfassten Daten. Das bedeutet, dass die erste Ausgabedatei auch Daten enthält, die vor dem Befehl **SCHLEIFE/ANFANG** gesammelt wurden. Bei der nächsten Ausführung des Befehls **DRUCKEN** führt PC-DMIS nur noch den Inhalt innerhalb der Schleife aus. Das bedeutet, dass die zweite Datendatei nur die Daten innerhalb der Schleife enthält. Dies wird so lange wiederholt, bis die Messroutine beendet ist.

.
. .
.

Schleifenanfang

```
KREIS1 = KREIS (KONTAKT)

KREIS1 = KREIS (KONTAKT)

Bewegungspunkt

Bewegungspunkt

Bewegungspunkt

Bewegungspunkt

KREIS3 = KREIS (KONTAKT)

Bewegungspunkt

Bewegungspunkt

Bewegungspunkt

KREIS4 = KREIS (KONTAKT)

KREIS5 = KREIS (KONTAKT)

KREIS6 = KREIS (KONTAKT)
```

```
KREIS7 = KREIS (KONTAKT)
```

```
KREIS8 = KREIS (KONTAKT)
```

```
LOC1 Passed: KREIS1
```

```
Protokoll drucken
```

```
Schleifenende
```

```
.  
.  
.
```

Sprungmarken verwenden

Sie können einen **SPRUNGMARKEN**-Befehl mit einem **GOTO**- oder einem **IF GOTO**-Befehl verwenden, um zu steuern, wo der Programmablauf bei bestimmten Bedingungen ansetzt. Die Bezeichnung der Sprungmarke darf keine Leerzeichen enthalten und nicht länger als 230 Zeichen sein. PC-DMIS stellt die Bezeichnung der Sprungmarke in Großbuchstaben an.

Ein **SPRUNGMARKEN**-Befehl kann auf folgende Weisen erstellt werden:

Eingabe des Befehls.

1. Setzen Sie den Cursor auf eine leere Zeile im Bearbeitungsfenster.
2. Geben Sie **SPRUNGMARKE** ein und drücken Sie die Tabulatortaste.
3. Bestimmen Sie im hervorgehobenen Feld eine neue Bezeichnung für die Sprungmarke.

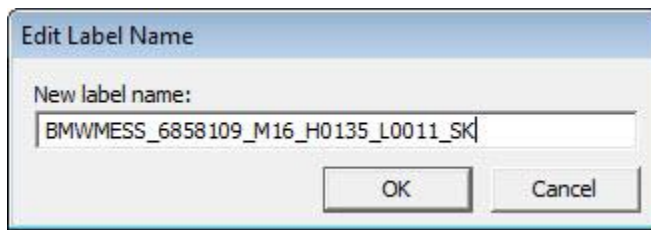
oder

Wählen Sie den Befehl

1. Setzen Sie den Cursor ins Bearbeitungsfenster.
2. Wählen Sie aus der Menüleiste **Einfügen | Programmablaufsteuerungs-Befehl | Sprungmarke**, um das Dialogfeld **Sprungmarkennamen bearbeiten** zu öffnen.

Sprungmarken verwenden

3. Geben Sie den Namen der Sprungmarke im Feld **Neuer Sprungmarkenname** ein.



Dialogfeld Sprungmarkennamen bearbeiten

4. Klicken Sie auf **OK**, um den Sprungmarkennamen an der nächstmöglichen Position im Bearbeitungsfenster einzufügen.

Der SPRUNGMARKEN-Befehl im Befehlsmodus des Bearbeitungsfenster lautet:

`ID = SPRUNGMARKE/`

Dabei steht ID für den entsprechenden Sprungmarkennamen.

Informationen zum Einsatz des `GOTO`-Befehls mit Sprungmarken finden Sie im Abschnitt „Springen zu einer Sprungmarke mit GOTO“.

Informationen zum Einsatz des `IF_GOTO`-Befehls mit Sprungmarken finden Sie im Abschnitt „Springen zu einer Sprungmarke auf Basis bestimmter Bedingungen“.

Springen zu einer Sprungmarke mit GOTO

Sie können Ihrer Messroutine mit einem GOTO-Befehl mitteilen sich zu einer bestimmten Sprungmarke zu bewegen.

Ein GOTO-Befehl kann auf folgende Weisen erstellt werden:

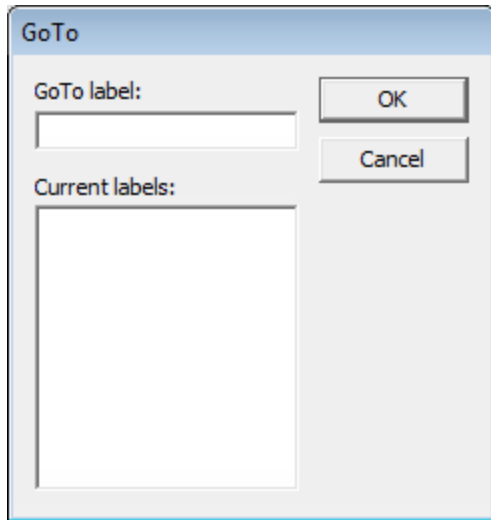
Eingabe des Befehls.

1. Setzen Sie den Cursor auf eine leere Zeile im Bearbeitungsfenster.
2. Geben Sie GOTO ein und drücken Sie die Tabulatortaste.
3. Bestimmen Sie den Namen der entsprechenden Sprungmarke.

oder

Wählen Sie den Befehl

1. Setzen Sie den Cursor ins Bearbeitungsfenster.
2. Wählen Sie die Menüoption **Einfügen | Programmablaufsteuerungs-Befehl | Goto**, um das Dialogfeld **Goto** zu öffnen.



Dialogfeld GoTo

3. Wenn in der Messroutine bereits Sprungmarken existieren, werden diese im Feld **Vorhandene Sprungmarken** angezeigt.
4. Wählen Sie die gewünschte Sprungmarke aus dem Feld **Vorhandenen Sprungmarken**, oder geben Sie den Namen direkt im Feld **GoTo Sprungmarke** ein.
5. Klicken Sie auf **OK**, um den GOTO-Befehl an der nächstmöglichen Position im Bearbeitungsfenster einzufügen.

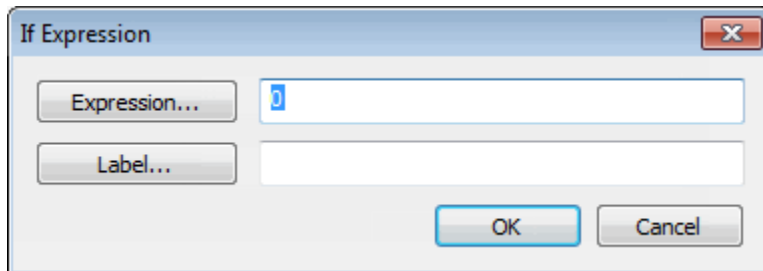
Der GOTO-Befehl im Befehlsmodus des Bearbeitungsfenster lautet:

`GOTO/SPRUNGMARKE_ID`

Dabei steht Sprungmarke_ID für die entsprechenden Sprungmarke. Die Bezeichnung der Sprungmarke darf keine Leerzeichen enthalten und nicht länger als 230 Zeichen sein.



Wenn die Sprungmarke noch nicht erstellt wurde, wird der Sprungmarkenname im Befehlsmodus des Bearbeitungsfensters in **Rot** angezeigt und der GOTO-Befehl wird bei der Ausführung ignoriert.



Dialogfeld If-Ausdruck

Mit der Option **Einfügen | Programmablaufsteuerungs-Befehl | If Goto** können Sie mit dem Dialogfeld **If-Ausdruck** IF GOTO-Anweisungen in Ihrer Messroutine erstellen. Trifft PC-DMIS bei Ausführung der Messroutine auf eine IF GOTO-Anweisung, rückt PC-DMIS zur Position der Sprungmarken-ID vor, wenn die Auswertung des speziellen Ausdrucks einen Wert ungleich Null ergibt.

Die im Bearbeitungsfenster für eine IF_GOTO-Anweisung angezeigte Befehlszeile lautet:

```
IF_GOTO/Ausdruck, GOTO=Sprungmarke
```

Ausdruck - Wenn Sie auf **Ausdruck** klicken, öffnet sich der Ausdrucks-Generator. Mit dem Ausdrucks-Generator können Sie eine Vielzahl verschiedener Ausdrücke erstellen und diese bei Bedarf in der Messroutine verwenden. Sobald Sie den Ausdruck erstellt haben, wird dieser im Feld **Ausdruck** angezeigt. Informationen zum Erstellen von Ausdrücken finden Sie unter „Verwenden von Ausdrücken und Variablen“.

Sprungmarke - Wenn Sie auf **Sprungmarke** klicken, öffnet sich das Dialogfeld **Goto**. Im Dialogfeld **Goto** können Sie die Sprungmarke wählen, an die die Messroutine springen soll, wenn der definierte Ausdruck erfüllt ist. Die ausgewählte Sprungmarke erscheint im Feld **Sprungmarke**. Wenn Sie den Namen der Sprungmarke kennen, können Sie diesen auch direkt im Feld eingeben. Weitere Informationen zu Sprungmarken finden Sie unter „Springen zu einer Sprungmarke“.

Schleife zurücksetzen

Der Befehl Schleife zurücksetzen setzt den Ausführungsbaum in einer Schleife zurück. Sie können den Befehl Schleife zurücksetzen hinzufügen, um mehrfache Werkstückmessungen mit diesen Befehlspaaren zu unterstützen:

- While/End While
- Do/Until
- Loop/End Loop

Sie müssen den Befehl Schleife zurücksetzen innerhalb einer dieser Schleifen verwenden. Der Befehl hat keine Wirkung, wenn Sie ihn außerhalb einer Schleife platzieren. Um einen Befehl Schleife zurücksetzen einzufügen, wählen Sie die Menüoption **Einfügen | Programmablaufsteuerung | Schleife zurücksetzen** aus. Die Befehlszeile im Bearbeitungsfenster für einen Befehl Schleife zurücksetzen würde lauten: `RESET LOOP/`

Bei Messungen mehrerer Werkstücke können Sie einen Bericht über das zuletzt gemessene Werkstück in der Schleife erhalten. Fügen Sie dazu den Befehl Schleife zurücksetzen am Anfang der Schleife ein, wie in dieser Abbildung gezeigt:

```

WHILE/COUNTER<=3
  RESET LOOP/
  CURRENT/OPER,NO,FULL_SCREEN=YES,AUTO_CONTINUE=YES,TIME_DELAY=1,OVC=NO,
  WAIT
  F1
  =GENERIC/CIRCLE,DEPENDENT,CARTESIAN,OUT,$
  NOM/XYZ,<0,0,0>,$
  MEAS/XYZ,<0,0,0>,$
  NOM/LPR,<0,0,1>,$
  MEAS/LPR,<0,0,1>,$
  DIAMETER/3,COUNTER
  DIM LOC1= LOCATION OF CIRCLE F1 UNITS=MM,$
  GRAPH=OFF TEXT=OFF MULT=10.00 OUTPUT=BOTH HALF ANGLE=NO
  AX NOMINAL +TOL -TOL MEAS DEV OUTTOL
  D 3.000 0.050 0.050 3.000 0.000 0.000 ----f----
  END OF DIMENSION LOC1
  PRINT/REPORT,EXEC MODE=END,$
  TO_FILE=ON,AUTO=4, OUTPUT FORMAT/TEXT,$
  RESET REPORT=YES,AUTO_OPEN=OFF,$
  TO_PRINTER=OFF,COPIES=1,$
  TO_EMAIL_REPORT=ON,FILE_OPTION=INDEX,FILENAME=C:\Users\Sudhansu.Trive
  REPORT THEORETICALS=NO,REPORT FEATURE WITH DIMENSIONS=NO,$
  TO_EXCEL=ON,AUTO=9,FILENAME=C:\Users\Sudhansu.Trivedi\Downloads\Whil
  PREVIOUS BOMS=DELETE,INSTANCES
  EXCEL_FORM1 =EXCELFORMREPORT/TEMPLATE=Form Row Wise, TYPE=EXCEL
  RESULT FOLDER=C:\Users\Public\Documents\Hexagon\PC-DMIS\2024.2\ExcelFor
  ASSIGN/COUNTER=COUNTER+1
  END_WHILE/
  END OF MEASUREMENT FOR

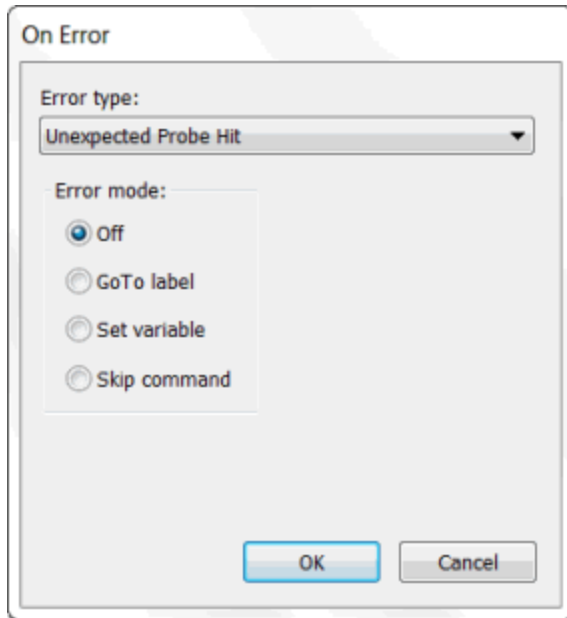
```

Der Befehl Schleife zurücksetzen entfernt alle Objekte aus dem Ausführungsbaum.

Verzweigen bei einem Fehler

Mit der Option **Einfügen | Programmablaufsteuerungs-Befehl | Bei Fehler** öffnen Sie das Dialogfeld **Bei Fehler**:

Verzweigen bei einem Fehler



Dialogfeld Bei Fehler

In diesem Dialogfeld können Sie die Maßnahmen definieren, die PC-DMIS bei Auftreten eines Maschinenfehlers ausführen soll.

Fehlertyp - PC-DMIS verfolgt diese Fehlerbedingungen:

- Unerwarteter Messpunkt
- Verfehlter Messpunkt
- Reflektor nicht gefunden - Wird mit dem Tracker in 'PC-DMIS Portable' benutzt.
- Laserfehler - Wenn Sie eine Laserkonfiguration verwenden, lesen Sie das Thema "Umgang mit Lasertasterfehlern unter Einsatz der Funktion BEI_FEHLER" in der Dokumentation zu PC-DMIS Laser.
- Temperatur überschreitet Grenzwert - Der Befehl zur Temperaturkompensation in der Messroutine löst diesen Fehler aus, wenn eine oder mehrere Temperaturen für die X-Achsen-, Y-Achsen- oder Z-Achsen-Skala oder Werkstück den Oberen Grenzwert oder den Unteren Grenzwert, die durch den Befehl zur Temperaturkompensation definiert sind, überschreiten bzw. unterschreiten.



Der Befehl Bei Fehler muss in der Messroutine vor dem Befehl zur Temperaturkompensation platziert werden.

- **Kalibrierung außerhalb der Grenzen** - Der Befehl Kalibriergrenzen prüfen in der Messroutine löst diesen Fehler aus, wenn eine oder mehrere Kalibrierfehlerprüfungen außerhalb der Grenzen liegen. Weitere Informationen zum Befehl Kalibriergrenzen prüfen finden Sie unter "Über Kalibriergrenzen prüfen" in der Dokumentation der PC-DMIS Toolkit-Module.



Der Befehl Bei Fehler muss in der Messroutine vor dem Befehl Kalibriergrenzen prüfen platziert werden.

Fehlermodus - PC-DMIS kann abhängig vom Fehlertyp die folgenden Maßnahmen ergreifen:

- **Aus** - PC-DMIS ohne Aktion.
- **Zu Sprungm. gehen** - Die Ausführung der Messroutine bewegt sich zu einer definierten Sprungmarke (siehe "Sprungmarken verwenden"). Die folgenden Optionen werden verfügbar:
 - **Sprungmarken-ID** - Damit können Sie eine Referenz für eine Sprungmarke definieren, die noch nicht vorhanden ist.
 - **Aktuelle Sprungmarken** - Damit erhalten Sie eine Übersicht über alle Sprungmarken in der Messroutine.
- **Variable festlegen** - Setzt den Wert einer Variable auf eins.
- **Befehl überspringen** - Die Messroutine überspringt den aktuellen Befehl und setzt mit dem nächsten markierten Befehl der Messroutine fort.



Details dazu, wie PC-DMIS Merkmale meldet, die während der Ausführung übersprungene Elemente verwenden, finden Sie unter dem Thema "Melden eines Merkmals, das ein während der Ausführung übersprungenes Element verwendet" in dieser Dokumentation.

Standardmäßig sind die Maßnahmen für beider Fehlertypen bei allen Messroutinen am Anfang **ausgeschaltet** (keine Reaktion). Der Aktionsmodus für jeden Fehlertyp kann während der Messroutine geändert werden.



Trifft PC-DMIS bei Ausführung des Werkstückprogramms auf einen Befehl **BEI FEHLER/ UNERWARTETER MESSPUNKT / ZU SPRUNGMARKE SPRINGEN**, so haben alle nach dieser Stelle in der Messroutine auftretenden unerwarteten Messpunkte zur Folge, dass die Ausführung der Messroutine zur angegebenen Sprungmarke springt. Die Aktion, mit welcher der "Wert einer Variablen auf Eins gesetzt wird", hat zur Folge, dass die Variable sofort nach Auftreten des angegebenen Fehlertyps auf diesen Wert gesetzt wird. Dieser Variablenwert kann anhand einer IF-Anweisung geprüft werden, um bei der Ausführung einen Sprung zu einer neuen Stelle in der Messroutine zu bewirken.

So verwenden Sie den Befehl **BEI_FEHLER**

1. Wählen Sie **Einfügen | Programmablaufsteuerungs-Befehl | Bei Fehler**, um das Dialogfeld **Bei Fehler** zu öffnen.
2. Wählen Sie von der Liste **Fehlertyp** den Fehlertyp zur Verzweigung.
3. Wählen Sie im Bereich **Fehlermodus** eine der Optionen, um die jeweils gewünschte Aktion festzulegen.
4. Klicken Sie zum Hinzufügen des Befehls Bei Fehler auf **OK**. Klicken Sie auf **Abbrechen**, um das Dialogfeld zu schließen, ohne dass Änderungen übernommen werden.

Unterstützte Schnittstellen **BEI FEHLER**

Einige Schnittstellen unterstützen den Befehl On Error (Bei Fehler). Ob Ihre Schnittstelle den Befehl unterstützt, entnehmen Sie bitte der folgenden Tabelle.

- Wenn Ihre Schnittstelle aufgelistet ist, wird durch ein kleines, schwarzes Feld angegeben, welcher Fehlertyp von dieser Schnittstelle unterstützt wird.
- Ist Ihre Schnittstelle nicht aufgeführt, können Sie den Befehl **BEI_FEHLER** nicht verwenden.

| Unterstützte Schnittstellen | Unerwarteter Messpunkt | Verfehlter Messpunkt | Reflektor nicht gefunden |
|--|------------------------|----------------------|--------------------------|
| DEA | - | ■ | - |
| FDC | ■ | ■ | - |
| I++ DME Client | ■ | ■ | - |
| Johansson | ■ | ■ | - |
| Leica-Tracker | - | - | ■ |
| Leitz | ■ | ■ | - |
| LK Direct (auch bekannt unter LKRS232) | ■ | ■ | - |
| LK-Treiber | ■ | ■ | - |
| Mora | ■ | ■ | - |
| Sharpe | ■ | ■ | - |
| Sheffield | ■ | ■ | - |
| Wenzel | ■ | ■ | - |
| Zeiss | ■ | ■ | - |

Verzweigung bei Unterprogrammen

Unterprogramme sind Codeblöcke in Ihrer Messroutine oder in einer externen Messroutine, auf die normalerweise wiederholt verwiesen wird, wodurch eine kürzere Programmierung möglich wird. In PC-DMIS können Sie Informationen an „Argumente“ (oder an lokale Variablen) im Unterprogramm übergeben. Zu den Argumenttypen, die an ein Unterprogramm übergeben werden können, gehören numerische Werte, Variablen, Zeichenfolgen und Elementnamen.



Die Befehlsblöcke der Unterprogramme stehen innerhalb der Befehle **UNTERPROGRAMM** und **ENDE_UNTERPROG**.

Nachdem Sie das Unterprogramm in der Messroutine erstellt haben, können Sie es in der aktuellen Messroutine oder von einer anderen Messroutine aus „aufrufen“. Dadurch wendet sich der Ablauf der Messroutine dem angegebenen Unterprogramm zu und die im Befehlsblock des Unterprogramms enthaltenen Befehle werden ausgeführt. Die Messroutine kehrt anschließend in seinem Ablauf zu der Anweisung zurück, die direkt auf den Aufruf folgt.



Unterprogramme werden mit Hilfe des Befehls **AUFR_UNTERPROG** aufgerufen.

Externe Unterprogramme

Externe Unterprogramme oder Unterprogramme, die sich in einer Messroutine außerhalb der aufrufenden Messroutine befinden, haben keinen Zugriff auf Elemente, Variablen oder Ausrichtungen aus der aufrufenden Messroutine. Das Unterprogramm hat jedoch Zugriff auf die Objekte innerhalb der eigenen Messroutine. Die externe Messroutine und die aufrufende Messroutine müssen dieselben Maßeinheiten verwenden.

Verschachteln von Unterprogrammen

Sie können Unterprogramme in anderen Unterprogrammen verschachteln. Die einzige Einschränkung in Bezug auf die Anzahl von Argumenten und die verschachtelten Unterprogramme ist der jeweils verfügbare Speicher.

Erstellen eines neuen Unterprogramms

*Erstellen eines Unterprogramms durch Eingabe von **UNTERPROGRAMM***

Sie können diesen Befehl eingeben, indem Sie **UNTERPROG** in den Befehlsmodus des Bearbeitungsfensters eingeben und dann die **TABULATORASTE** drücken. Nachdem der Befehl eingefügt ist, müssen Sie den Namen des Unterprogramms sowie alle Argumente angeben. Diese Angaben finden Sie in der Syntax des Unterprogramms und im Beispiel weiter unten.

Geben Sie den **ENDE_UNTERPROG**-Befehl ein und drücken Sie die **TABULATORASTE**, um den Befehlsblock zu beenden. Alle im Bearbeitungsfenster in

diesen Befehlsblock eingefügte Befehle werden als Teil des Unterprogramms betrachtet und beim Aufrufen des Unterprogramms ausgeführt.

Erstellen eines Unterprogramms über den Menüeintrag "Unterprogramm"

1. Wählen Sie im Untermenü die Option **Einfügen | Programmablaufsteuerung | Unterprogramm erstellen** aus. Dadurch wird das Dialogfeld "Unterprogramm erstellen" eingeblendet. Weitere Informationen zu diesem Dialogfeld finden Sie im Thema "Dialogfeld 'Unterprogramm erstellen'".



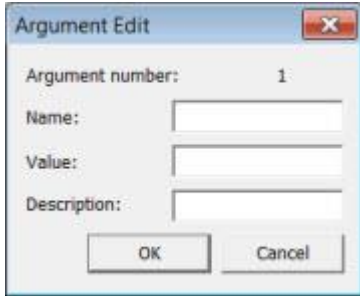
Dialogfeld Unterprogramm erstellen

2. Benennen Sie das Unterprogramm durch Eingabe des Namens in das Feld **Name**. Der Name darf maximal 180 Zeichen lang sein. Wenn Sie einen Wert ab 181 eingeben, erscheint die Meldung "Zeile zu lang".

Wenn Sie die Schaltfläche **OK** auswählen, wird der Name automatisch auf 180 Zeichen gekürzt und das Dialogfeld **Unterprogramm-Erstellung** geschlossen.

3. Wenn das Unterprogramm Argumente (Platzhalter für Informationen, die an das Unterprogramm weitergegeben werden) verwendet, können Sie diese einzeln durch Klicken auf die Schaltfläche **Argument hinzufügen** hinzufügen. Es erscheint das Dialogfeld **Argument bearbeiten**. Weitere Informationen zu diesem Dialogfeld finden Sie im Thema „Dialogfeld 'Argument bearbeiten'“.

Verzweigung bei Unterprogrammen



Dialogfeld Argument bearbeiten

4. Geben Sie im Feld **Name** einen Namen für das Argument ein.
5. Geben Sie im Feld **Wert** einen Standardwert für das Argument ein. Das Unterprogramm verwendet den Standardwert, wenn von der `AUFR_UNTERPROG`-Anweisung keine Werte an das Unterprogramm übergeben werden. Gültige Argumentwerte sind numerische Werte, Variablen, Textzeichenfolgen oder Elementnamen.
6. Eine Beschreibung des Arguments können Sie je nach Bedarf im Feld **Beschreibung** eingeben.
7. Klicken Sie im Dialogfeld **Argument bearbeiten** auf **OK**, um das Argument zu erstellen.
8. Wiederholen Sie Schritt 3 bis 7 für jedes Argument, das im Unterprogramm aufgenommen werden soll.
9. Klicken Sie im Dialogfeld **Unterprogramm erstellen** auf die Schaltfläche **OK**, um die Erstellung des Unterprogramms abzuschließen. Dieses Unterprogramm erscheint im Bearbeitungsfenster mit allen definierten Argumenten.
10. Beenden Sie das Unterprogramm durch Auswahl der Menüoption **Einfügen | Programmablaufsteuerungs-Befehl | Unter. beenden**. Dadurch wird der Befehl „`ENDEUNTERPROG/`“ in das Bearbeitungsfenster platziert, womit der Befehlsblock des Unterprogrammes vollständig ist. Alle weiteren Befehle der Messroutine, die in das Unterprogramm aufgenommen werden sollen, müssen vor den Befehl `ENDE_UNTERPROG` innerhalb des Befehlsblocks des Unterprogramms hinzugefügt werden.

Syntax für einen Unterprogramm-Befehlsblock

Die im Bearbeitungsfenster für ein Beispiel-Unterprogramm angezeigte Syntax für einen Befehlsblock lautet:



```
SUBROUTINE/<Name>,  
<A1> = <Arg1> : <Description>,  
<A2> = <Arg2> : <Description>,
```

```
=
<Commands>
ENDSUB/
```

UNTERPROGRAMM/ ist der Befehl, der zum Starten des Befehlsblocks für das Unterprogramm verwendet wird.

<Name>= der Name des Unterprogramms. Der Name darf maximal 256 Zeichen lang sein. Wenn Sie 257 oder mehr Zeichen eingeben, wird der Name automatisch auf 256 Zeichen gekürzt.

<A1>= Das erste, im Unterprogramm verwendete Argument (oder lokale Variable). Auf diese Variable kann normalerweise außerhalb des Unterprogramms nicht zugegriffen werden.

<A2>= Zweites, im Unterprogramm verwendete Argument. Auf diese Variable kann normalerweise außerhalb des Unterprogramms nicht zugegriffen werden. Je nach Bedarf können weitere Argumente hinzugefügt werden.

<Arg1> = Der Standardwert für das erste Argument.

<Arg2>= Der Standardwert für das zweite Argument.

<Beschreibung> = Die Beschreibung des Arguments.

<Befehle> = Weitere Befehle des Bearbeitungsfensters können, falls gewünscht, hinter den Argumenten und vor dem ENDE_UNTERPROG-Befehl eingefügt werden.

ENDE_UNTERPROG/ ist der Befehl, der zum Beenden des Befehlsblocks für das Unterprogramm verwendet wird.

Beispiel-Befehlsblock für ein Unterprogramm

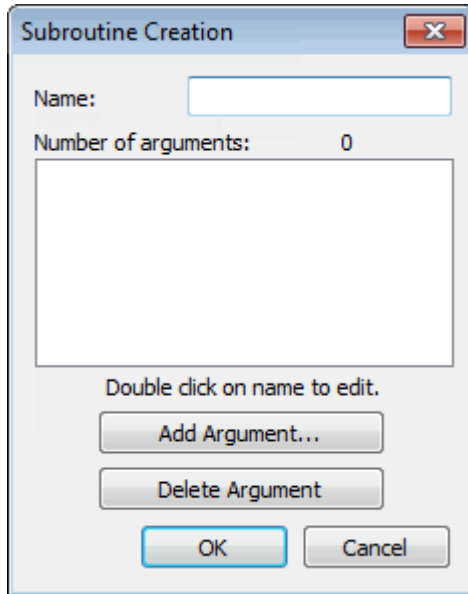
Ein abgeschlossenes Unterprogramm mit Bedienerdaten, die im Protokoll angezeigt werden, sieht beispielsweise so aus:



```
SUBROUTINE/GET_OPERATOR_INFO,
  OPNAME = <Operator> : OPERATOR NAME,
  SHIFT = <Shift> : SHIFT TIME,
  =
  COMMENT/REPT,OPNAME
  COMMENT/REPT,SHIFT
ENDSUB/
```

Dialogfeld "Unterprogramm erstellen"

Wählen Sie aus der Menüleiste **Einfügen | Programmablaufsteuerungs-Befehl | Unterprogramm**, um das Dialogfeld **Unterprogramm erstellen** zu öffnen.



Dialogfeld Unterprogramm erstellen

Nachfolgend werden die im Dialogfeld **Unterprogramm erstellen** verfügbaren Optionen beschrieben.

Name - Dieses Feld definiert den Namen des Unterprogrammes. Dies ist der Name, den Sie verwenden, wenn Sie das Unterprogramm später aufrufen. Wenn Sie also über mehrere Unterprogramme in einer einzigen Messroutine verfügen, muss jeder Name einmalig sein.

Der Name darf maximal 180 Zeichen lang sein. Wenn Sie einen Wert ab 181 in das Feld **Name** eingeben, erscheint die Meldung "Zeile zu lang".

Wenn Sie die Schaltfläche **OK** auswählen, wird der Name automatisch auf 180 Zeichen gekürzt und das Dialogfeld **Unterprogramm-Erstellung** geschlossen.

Anzahl der Argumente - Diese Liste enthält die Argumente für das Unterprogramm, das gerade erstellt wird. Argumente erscheinen in diesem Bereich in folgender Form:



Wurde beispielsweise eines der Argumente mit dem Namen "Durchmesser" und dem Standardwert 3 versehen, dann erscheint dieses Argument in dieser Liste so:



DURCHMESSER = 3 : Der Durchmesser des Lochs

PC-DMIS verwendet den Standardwert immer dann, wenn kein anderer Wert vom Befehl `AUFR_UNTERPROG` übergeben wurde.

Doppelklicken Sie zur Bearbeitung eines Arguments auf das Argument, das geändert werden soll. Das Dialogfeld **Argument bearbeiten** wird geöffnet und Sie haben die Möglichkeit, die Änderungen vorzunehmen. Weitere Informationen zu diesem Dialogfeld finden Sie im Thema „Dialogfeld 'Argument bearbeiten'“.

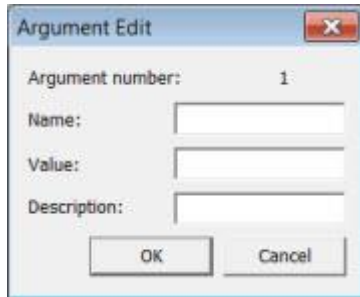
Argument hinzufügen - Mit dieser Schaltfläche werden Ihrem Unterprogramm neue Argumente hinzugefügt. Klicken Sie auf diese Schaltfläche, woraufhin das Dialogfeld **Argument bearbeiten** erscheint. Weitere Informationen zu diesem Dialogfeld finden Sie im Thema „Dialogfeld 'Argument bearbeiten'“.

Argument löschen - Mit dieser Schaltfläche können Sie Argumente aus dem Unterprogramm löschen. Markieren Sie einfach das Argument in der Liste und klicken Sie dann auf die Schaltfläche **Argument löschen**.

Bearbeiten eines vorhandenen Unterprogramms

Um ein vorhandenes Unterprogramm zu bearbeiten, können Sie jederzeit den Befehlsmodus des Bearbeitungsfensters verwenden und das Unterprogramm direkt bearbeiten. Alternativ können Sie das Dialogfeld **Unterprogramm erstellen** aufrufen, indem Sie den Mauszeiger auf dem Unterprogramm im Bearbeitungsfenster platzieren und die Taste F9 drücken. Dadurch wird das Dialogfeld **Unterprogramm erstellen** geöffnet. Informationen zu diesem Dialogfeld finden Sie im Thema "Dialogfeld 'Unterprogramm erstellen' ".

Hinweise zum Dialogfeld "Argument bearbeiten"



Dialogfeld Argument bearbeiten

Das Dialogfeld **Argument bearbeiten** erscheint, wenn Sie ein Argument entweder im Dialogfeld **Unterprogramm erstellen (Einfügen | Programmablaufsteuerung | Unterprogramm)** oder im Dialogfeld **Unterprogramm aufrufen (Einfügen | Programmablaufsteuerung | Unterprogramm aufrufen)** erstellen bzw. bearbeiten möchten.

Das Dialogfeld **Argument bearbeiten** kann in den beiden nachfolgenden Kontexten eingesetzt werden:

- Zum Definieren der Argumente eines Unterprogramms und deren Standardwerte in einem `UNTERPROGRAMM`-Befehlsblock.
- Zum Definieren von Werten, die von einem `AUFR_UNTERPROG`-Befehl an das Unterprogramm übergeben werden.

Im Dialogfeld **Argument bearbeiten** sind die folgenden Optionen verfügbar:

Name - In diesem Feld definieren Sie den Namen des zu erstellenden bzw. zu bearbeitenden Arguments.

Wert - In diesem Feld geben Sie den Wert des Arguments ein.

Wenn Sie den Befehl `UNTERPROGRAMM` erstellen oder bearbeiten, ist dies der Standardwert, der dann benutzt wird, wenn kein Wert von der `AUFR_UNTERPROG`-Anweisung an das Unterprogramm übergeben wird.

Wenn Sie den Befehl `AUFR_UNTERPROG` erstellen oder bearbeiten, ist dies der Wert, der an das Unterprogramm übergeben wird.

Gültige mögliche Werte:

- Numerische Zeichen
- Variablen

- Textzeichenfolge - Textzeichenfolgen müssen in doppelten Anführungszeichen stehen.
- Elementname - Der Elementname muss in geschwungenen Klammern stehen (zum Beispiel: {F1}).

Beschreibung - Definiert die Beschreibung des Arguments im Unterprogramm. Diese Beschreibung wird im Bearbeitungsfenster im Befehlsblock `UNTERPROGRAMM` neben dem Argument eingeblendet.

Aufrufen eines Unterprogramms

Um ein Unterprogramm aufzurufen, müssen Sie einen `AUFR_UNTERPROG`-Befehl in die Messroutine einfügen. Damit rufen Sie ein vorhandenes Unterprogramm in der aktuellen Messroutine oder ein Unterprogramm aus einer externen Messroutine auf.



Wenn ein Befehl `AUFR_UNTERPROG` ausgeführt wird, führt PC-DMIS alle Befehle des Unterprogramms aus. Dabei werden sogar unmarkierte Befehle ausgeführt.

Das aufgerufene Unterprogramm muss in einer gültigen PC-DMIS-Version (Versionen 2020 R2 bis 2026.1) erstellt worden sein.

Aufrufen eines Unterprogramms durch Eingabe von `AUFR_UNTERPROG`

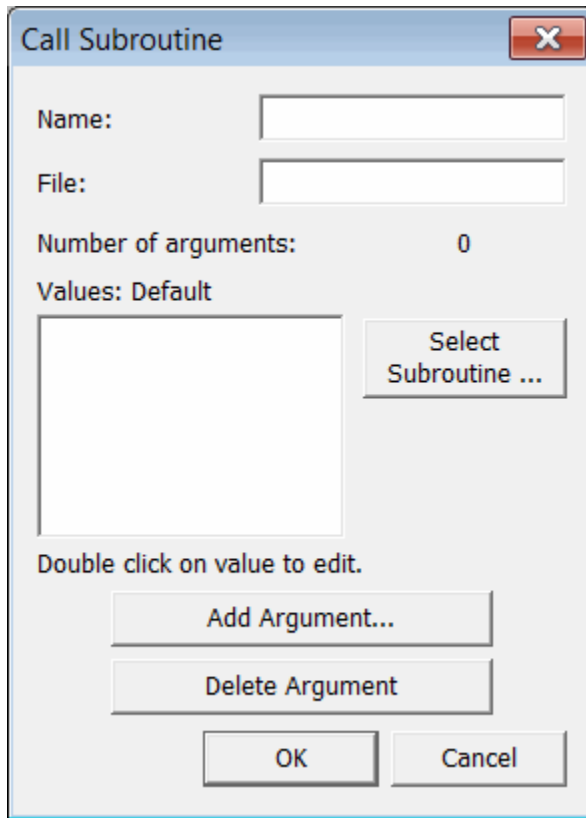
Sie können diesen Befehl einfügen, indem Sie im Bearbeitungsfenster `AUFR_UNTERPROG` eingeben und anschließend die TABulatortaste an der Stelle drücken, an der der Befehl im Bearbeitungsfenster erscheinen soll.

Nachdem der Befehl eingefügt ist, müssen Sie den Namen des Unterprogramms, seine Position - wenn es sich um eine externe Messroutine handelt - sowie alle Werte, die an verfügbare Argumente übergeben werden sollen, angeben. Beispiele zum Übergeben/Übernehmen von Argumenten finden Sie unter „Übergeben von Variablen an ein Unterprogramm“.

Aufrufen eines Unterprogramms mit Hilfe des Menüeintrags "Unterprogramm aufrufen"

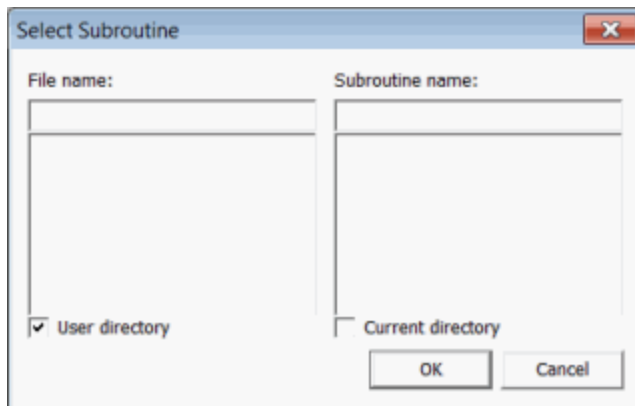
1. Wählen Sie aus dem Untermenü die Option **Einfügen | Programmablaufsteuerung | Unterprogr. aufrufen** aus. Das Dialogfeld *Unterprogramm aufrufen* wird geöffnet. Weitere Informationen zu diesem Dialogfeld finden Sie unter "Dialogfeld 'Unterprogramm aufrufen'".

Verzweigung bei Unterprogrammen



Unterprogramm aufrufen (Dialogfeld)

2. Klicken Sie auf die Schaltfläche **Unterprogramm auswählen**. Das Dialogfeld *Unterprogramm auswählen* wird geöffnet.



Dialogfeld Unterprogramm auswählen

3. Markieren Sie entweder das Kontrollkästchen **Benutzerverzeichnis**, das Kontrollkästchen **Aktuelles Verzeichnis** oder beide. Wenn die Messroutine, aus dem das Unterprogramm stammt, in dem nach Unterprogrammen zu

durchsuchenden Verzeichnis installiert ist, markieren Sie das Kontrollkästchen **Benutzerverzeichnis**. Wenn es aus dem aktuellen Verzeichnis stammt, markieren Sie das Kontrollkästchen **Aktuelles Verzeichnis**. PC-DMIS führt alle zur Auswahl stehenden Messroutinen auf.

4. Wählen Sie die Messroutine, in dem das gewünschte Unterprogramm enthalten ist. Im Feld **Unterprogrammname** werden alle mit der gewählten Messroutine verknüpften Unterprogramme angezeigt.
5. Klicken Sie auf das Unterprogramm, das Sie aufrufen möchten.
6. Klicken Sie auf **OK**. Die Angaben zum Unterprogramm, die Sie aufrufen möchten, erscheinen in den Feldern **Name** und **Datei** des Dialogfeldes **Unterprogramm aufrufen**.
7. Wenn Sie Informationen an das Unterprogramm übergeben möchten, klicken Sie auf die Schaltfläche **Arg. hinzufügen** und definieren im Dialogfeld **Argument bearbeiten** die entsprechenden Argumente und Werte. Weitere Informationen zu diesem Dialogfeld finden Sie unter „Dialogfeld 'Argument bearbeiten'“. Beispiele zum Übergeben/Übernehmen von Argumenten finden Sie unter „Übergeben von Variablen an ein Unterprogramm“.
8. Klicken Sie auf die Schaltfläche **OK**. PC-DMIS fügt den Befehl `AUFR_UNTERPROG` an der ausgewählten Stelle des Bearbeitungsfensters ein.

Syntax für den Befehl `AUFR_UNTERPROG`

Die Befehlssyntax im Bearbeitungsfenster zum Aufrufen eines Unterprogramms lautet:



```
CS1          =CALLSUB/<Name>, <File>:<Arg1>,<Arg2>,
```

CS1 = die dem Befehl `AUFR_UNTERPROG` zugeordnete Sprungmarken-ID.

<Name> = der Name des Unterprogramms, das aufgerufen werden soll.

<Datei> = der vollständige Verzeichnispfad zur Messroutine, welches das Unterprogramm enthält. Wenn dieses Feld leer ist, sucht PC-DMIS in der aktuellen Messroutine nach dem Unterprogramm.

<Arg1> = der Wert für das erste Argument im Unterprogramm. Wenn dieses Feld leer ist, verwendet PC-DMIS stattdessen den Standardwert, der für das erste Argument in der Unterprogramm definiert wurde.

<Arg2> = der Wert für das zweite Argument im Unterprogramm. Wenn dieses Feld leer ist, verwendet PC-DMIS stattdessen den Standardwert, der für das zweite Argument in der Unterprogramm definiert wurde. Dieses Syntax-Beispiel zeigt nur zwei Argumente. Dem Unterprogramm können je nach Bedarf weitere Argumente übergeben werden.



In Ihrem Befehl `CALLSUB` sollten Sie einen Satz mit den Zeigern zu allen Objekten ablegen, die für das Unterprogramm erstellt wurden, so dass Sie mit Hilfe der Unterprogramm-ID später wieder darauf verweisen können. Zusätzliche Informationen zu Zeigern finden Sie unter "Zeiger" im Abschnitt "Verwenden von Ausdrücken und Variablen".

Beispiel eines `AUFR_UNTERPROG`-Befehls

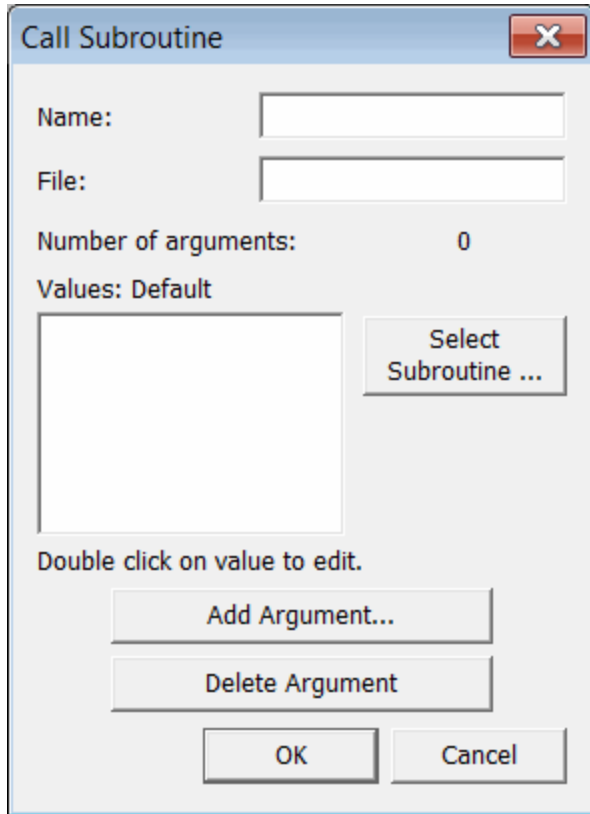


```
CS1          =CALLSUB/GET_OPERATOR_INFO,D:\MEASROUTINES\V42SUBROUT  
INETEST.PRG:V1,V2,,
```

Dieser Beispiel-Befehl `AUFR_UNTERPROG`, CS1, ruft ein Unterprogramm die Messroutine „V42UNTERPROGTEST.PRG“ mit Namen `BEDIENER_INFO` auf, das im Verzeichnis „D:\MESSROUTINEN\“ gespeichert ist.

Er übergibt zwei Werte - in diesem Fall die Variablen V1 und V2 - an das Unterprogramm.

Dialogfeld "Unterprogramm aufrufen"



Unterprogramm aufrufen (Dialogfeld)

In der nachfolgenden Tabelle werden die im Dialogfeld **Unterprogramm aufrufen** verfügbaren Optionen beschrieben.

Feld **Name**



Das Feld **Name** enthält den Namen des Unterprogramms, das Sie nach Anwendung der Schaltfläche **Unterprogramm auswählen...** ausgewählt haben.

Der Name darf maximal 180 Zeichen lang sein. Wenn Sie einen Wert ab 181 in das Feld **Name** eingeben, erscheint die Meldung "Zeile zu lang".

Wenn Sie die Schaltfläche **OK** auswählen, wird der Name automatisch auf 180 Zeichen gekürzt und das Dialogfeld **Unterprogramm aufrufen** geschlossen.

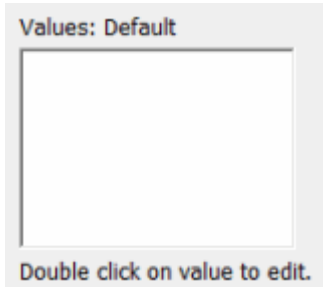
Verzweigung bei Unterprogrammen

Feld **Datei**



Das Feld **Datei** enthält den Verzeichnispfad zur aufgerufenen Unterprogramm-Datei.

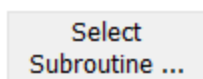
Feld **Werte**



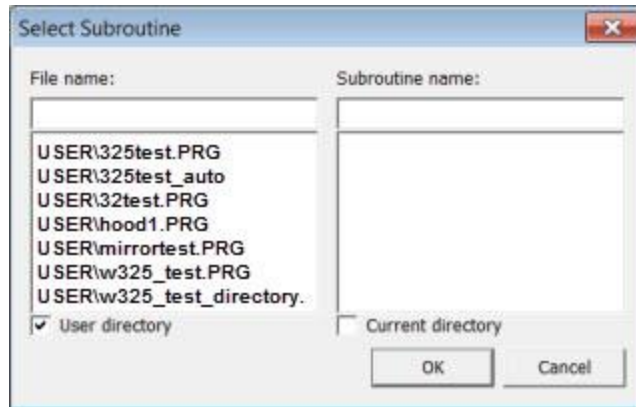
Das Feld **Werte** enthält eine Liste der Werte jedes Arguments, das mit dem Unterprogramm verknüpft ist. Diese Werte werden bei Ausführung an das Unterprogramm übergeben.

Um diese Werte zu ändern, doppelklicken Sie auf den Wert, der geändert werden soll. Es erscheint das Dialogfeld **Argument bearbeiten**. Weitere Informationen zu diesem Dialogfeld finden Sie im Thema „Dialogfeld 'Argument bearbeiten'“.

Schaltfläche **Unterprogramm auswählen**



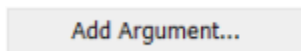
Mit der Schaltfläche **Unterprogramm auswählen** wird das Dialogfeld **Unterprogramm auswählen** angezeigt.



Dialogfeld Unterprogramm auswählen

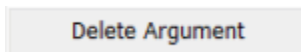
Mit diesem Dialogfeld können Sie zuvor erstellte Unterprogramme aufrufen, indem Sie das Benutzerverzeichnis oder das aktuelle Verzeichnis durchsuchen. Siehe das Thema "Aufrufen eines Unterprogramms", in dem weitere Informationen zum Dialogfeld **Unterprogramm auswählen** verfügbar sind.

Schaltfläche **Argument hinzufügen**



Diese Schaltfläche definiert einen Wert für die Argumente, die an das Unterprogramm übergeben werden sollen.

Schaltfläche **Argument löschen**



Mit dieser Schaltfläche können Sie Argumente aus dem Feld **Werte** löschen. Wählen Sie den angezeigten Wert aus und klicken Sie dann auf die Schaltfläche **Argument löschen**. Das mit diesem Wert verknüpfte Argument wird daraufhin gelöscht.

So fügen Sie ein neues Argument mit Hilfe des Dialogfelds "Unterprogramm aufrufen" hinzu:

So fügen Sie dem Befehl `AUFR_UNTERPROG` ein neues Argument zur Übernahme in das Unterprogramm hinzu:

Verzweigung bei Unterprogrammen

1. Klicken Sie im Dialogfeld **Unterprogramm aufrufen** auf die Schaltfläche **Arg. hinzufügen**. Es erscheint das Dialogfeld **Argument bearbeiten**.
2. Klicken Sie in das Feld **Wert**.
3. Geben Sie im Feld **Wert** den Wert des Arguments ein.
4. Klicken Sie auf **OK**.

So bearbeiten Sie vorhandene Argumente mit Hilfe des Dialogfelds "Unterprogramm aufrufen":

So bearbeiten Sie ein vorhandenes Argument innerhalb des [AUFR_UNTERPROG](#)-Befehls:

1. Doppelklicken Sie im Dialogfeld **Unterprogramm aufrufen** auf den Wert des Arguments, den Sie ändern möchten. Es erscheint ein Feld mit dem Standardwert für das aufgerufene Unterprogramm.
2. Geben Sie einen neuen Wert ein.
3. Klicken Sie auf **OK**.

Weitere Hinweise zum Bearbeiten oder Erstellen neuer Argumente für ein Unterprogramm finden Sie unter „Erstellen eines neuen Unterprogramms“ und „Bearbeiten eines vorhandenen Unterprogramms“.

So löschen Sie Argumente aus einem UNTERPROGAUFR-Befehl:

1. Setzen Sie den Cursor auf den [AUFR-UNTERPROG](#)-Befehl.
2. Drücken Sie F9, um das Dialogfeld **Unterprogramm aufrufen** zu öffnen.
3. Wählen Sie ein oder mehrere Argumente aus der Liste der Argumente.
4. Klicken Sie auf die Schaltfläche **Arg. löschen**.
5. Klicken Sie auf **OK**.

Sie können Argumente auch direkt im Bearbeitungsfenster-Text löschen. Um dies zu tun,

1. Versetzen Sie PC-DMIS in den Befehlsmodus.
2. Setzen Sie den Cursor auf den [AUFR_UNTERPROG](#)-Befehl und drücken Sie die Tabulatortaste, bis das gewünschte Argument markiert ist.

3. Geben Sie die Buchstaben „del“ ein, um das Argument zu löschen. Beachten Sie bitte, dass die Tasten Löschen oder Rücktaste das Argument nicht komplett löschen, sondern es nur leeren.

Verwenden von AUFR_UNTERPROG-Anweisungen im Mehrarm-Betrieb

Wenn Sie eine AUFR_UNTERPROG-Anweisung Arm1 zuweisen, werden alle Befehle im Unterprogramm beim Aufrufen des Unterprogramms Arm1 zugewiesen.

Wenn Sie eine AUFR_UNTERPROG-Anweisung Arm2 zuweisen, werden alle Befehle im Unterprogramm beim Aufrufen des Unterprogramms Arm2 zugewiesen.

Wenn Sie eine AUFR_UNTERPROG-Anweisung für beide Arme markieren, lässt PC-DMIS die Markierungen des Unterprogramms so, wie sie ursprünglich eingerichtet wurden.

Wenn ein Unterprogramm einen BEWEGEN/SYNC-Befehl enthält und Sie die AUFR_UNTERPROG-Anweisung "Arm1" oder "Arm2" zuweisen, zeigt PC-DMIS bei der Ausführung eine Fehlermeldung an, die besagt, dass dies unzulässig ist, und das Unterprogramm wird nicht aufgerufen.

Weitere Informationen über das Zuweisen eines Befehls zu einem bestimmten Messarm für die Ausführung finden Sie im Thema "Zuweisen eines Befehls zu einem Messarm" im Abschnitt "Arbeiten im Mehrarmbetrieb".

Beispiele für Unterprogramme

Beachten Sie die Informationen in den folgenden Themen für einige Beispiele zum Übergeben von Argumenten und zu Unterprogrammen im Allgemeinen.

- Übergeben von Variablen an ein Unterprogramm
- Codebeispiel für ein Unterprogramm

Übergeben von Variablen an ein Unterprogramm

Zu den Argumenttypen, die an ein Unterprogramm übergeben werden können, gehören numerische Werte, Variablen, Zeichenfolgen und Elementnamen. Zum Übergeben von Werten an Argumente geben Sie den Wert entweder im Feld **Wert** des Dialogfelds **Argument bearbeiten** oder direkt im Befehlsmodus in einen eingefügten AUFR_UNTERPROG-Befehl ein.

Verzweigung bei Unterprogrammen

Weitere Informationen zur Vorgehensweise finden Sie unter „Erstellen eines neuen Unterprogramms“ und „Dialogfeld 'Argument bearbeiten'“.


Übergeben von Variablen an ein Unterprogramm

Argumente, die Daten wieder zurückgeben können, werden als Variablen bezeichnet. Wird eine Variable als Argument zu einem Unterprogramm verwendet, werden sämtliche Änderungen, die an der entsprechenden Variablen des Unterprogramms auftreten, zurückgegeben und nehmen den Wert der an das Unterprogramm übergebenen Variablen an.



Beispiel zum Übergeben von Variablen:

Das folgende Beispiel zeigt, wie der Wert einer Variablen, die als Argument an ein Unterprogramm übergeben wird, vom Unterprogramm modifiziert und zurückgegeben wird:



```
CS4      ASSIGN/V1=6
        =CALLSUB/MYSUB, :V1,,
        SUBROUTINE/MYSUB,
            A1 = 0 : FIRST ARGUMENT,
            =
        ASSIGN/A1=A1+1
        COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
        The value of A1 is:
        A1
        ENDSUB/
```

Erläuterung des Codes mit Ausführungsablauf

Der Variablen V1 wird der Wert 6 zugewiesen:

Die Anweisung `CALLSUB` ruft dann die Unteroutine auf und übergibt die Variable V1 als erstes Argument.

Die Anweisung `SUBROUTINE/MYSUB` definiert MYSUB als Unterprogramm.

A1 ist der Name des ersten Arguments. Das heißt, wenn der Aufruf stattfindet, hat A1 denselben Wert wie V1 zum Zeitpunkt des Aufrufs (den Wert 6).

Das Unterprogramm führt dann seine beiden Anweisungen aus:

- Die erste Anweisung ist ein Befehl `ASSIGN`, der den Wert von A1 auf 7 erhöht.
- Die zweite Anweisung ist ein Befehl `COMMENT/OPER`, der den aktualisierten Wert A1 anzeigt.

Das Unterprogramm endet daraufhin mit dem Befehl `ENDE_UNTERPROG/`.

Der Ausführungsablauf kehrt zu der Anweisung zurück, die direkt auf den Befehl `CALLSUB` folgt. Dies ist ein weiterer Befehl `COMMENT/OPER`, der den endgültigen Wert von V1 anzeigt.



Beim Zurückspringen der Ausführung werden alle Variablen, die als Argumente verwendet wurden – in diesem Fall V1 – auf den Wert der entsprechenden Variablen im Unterprogramm aktualisiert – in diesem Fall A1. Nach der Rückkehr aus dem Unterprogramm hat V1 jetzt also den Wert 7. Dieser Wert wurde vom Unterprogramm zurückgegeben.

Übergeben von Zahlenwerten an ein Unterprogramm

Argumente können auch numerische Zeichen enthalten.



Beispiel zum Übergeben von Zahlenwerten:

Dieses Beispiel veranschaulicht, wie Zahlenwerte an ein Unterprogramm übergeben werden. Es werden bis zu zwei Zahlen übergeben, die dann addiert werden.



```
CS1      =CALLSUB/SUM_NUMBERS, :,
CS2      =CALLSUB/SUM_NUMBERS, :5,10,,
          SUBROUTINE/SUM_NUMBERS,
            NUM1 = 1 : FIRST NUMBER,
            NUM2 = 1 : SECOND NUMBER,
            =
          ASSIGN/TOTAL=NUM1+NUM2
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          NUM1 + "+" + NUM2 + "=" + TOTAL
          ENDSUB/
```

Verzweigung bei Unterprogrammen

Im ersten Befehl `CALLSUB` (CS1) werden keine Zahlenwerte an das Unterprogramm übergeben. Stattdessen werden die Standardwerte, 1 für NUM1 und 1 für NUM2, verwendet. Die erzeugte Summe wäre 2.

Im zweiten Befehl `CALLSUB` (CS2) werden die beiden Zahlenwerte 5 und 10 übergeben. NUM1 ist also 5 und NUM2 ist 10, und die Gesamtsumme ist 15.


Übergeben von Textzeichenfolgen an ein Unterprogramm

Argumente können außerdem eine Textzeichenfolge enthalten. Stellen Sie zum Übergeben einer Textzeichenfolge sicher, dass die alphanumerischen Zeichen in doppelten Anführungszeichen platziert werden.



Beispiel zum Übergeben von Textzeichenfolgen:

Im folgenden Beispiel wird veranschaulicht, wie Zeichenwerte an ein Unterprogramm übergeben werden. Es werden zwei Zeichenwerte an die beiden Parameter übergeben und anschließend im Protokoll angezeigt:



```
CS1          =CALLSUB/GET_OPERATOR_INFO,:"BOB
JONES", "MORNING",,,
              SUBROUTINE/GET_OPERATOR_INFO,
                OPNAME = "<NO PASSED OPERATOR>" :
OPERATOR NAME,
                SHIFT = "<NO PASSED SHIFT>" : SHIFT TIME,
                =
                COMMENT/REPT,
                OPNAME
                COMMENT/REPT,
                SHIFT
                ENDSUB/
```

Das erste Argument, `BEDIENERNAME`, erhält den übergebenen Wert "RAINER ZUFALL" und das zweite Argument, `SCHICHT`, erhält "FRÜHSCHICHT". Die beiden Befehle `COMMENT/REPT` senden dann die übergebenen Zeichenfolgen an den Prüfbericht.

Übergeben von Elementnamen an ein Unterprogramm

Sie können die Namen der Elemente in geschweiften Klammern übergeben. Wenn Sie z. B. im Befehl `CALLSUB` {PNT1} als Parameter verwenden, übergibt die Software PNT1 an das Unterprogramm. Beim Übergeben des Elementnamens erhält das Unterprogramm außerdem uneingeschränkten Zugriff auf dieses Element.

**Beispiel zum Übergeben von Elementnamen:**

Dieses Beispiel übergibt den Elementnamen PKT1 an das Unterprogramm, wodurch das Unterprogramm uneingeschränkten Zugriff auf das Element erhält. Wird kein Wert übergeben, dann wird der standardmäßige Elementname "F1" verwendet. Dieses Unterprogramm fragt dann nach einem neuen Wert und ändert den theoretischen X-Wert für das Element.



```

CS1      =CALLSUB/CHANGE_THEO_X,:{PNT1},,
C1       =COMMENT/INPUT,NO,FULL SCREEN=NO,
          Passed feature:
          FEAT1
          The current theo X is:
          FEAT1.TX
          Type a new theo X value:
          ASSIGN/FEAT1.TX=C1.INPUT
          COMMENT/OPER,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,OVC=NO,
          "Feature updated to " + FEAT1.TX

```

Da das Unterprogramm Zugriff auf das übergebene Element erhält, modifiziert die Anweisung `ASSIGN/FEAT1.TX=C1.INPUT` den theoretischen X-Wert des tatsächlichen Elements PNT1. PKT1 ändert jetzt dauerhaft seinen theoretischen X-Wert.

Codebeispiel für ein Unterprogramm

Mit Hilfe des folgenden Codebeispiels kann der Bediener die theoretischen X-, Y- und Z-Werte eines Elements nach dessen Messung ändern. Alle nachfolgenden Messungen verwenden dann diese aktualisierten theoretischen Werte.



```

PNT1     =GENERIC/POINT,DEPENDENT,CARTESIAN,$
          NOM/XYZ,<5,10,15>,$
          MEAS/XYZ,<7,12,17>,$
          NOM/IJK,<0,0,1>,$
          MEAS/IJK,<0,0,1>
C1       =COMMENT/YESNO,NO,FULL SCREEN=NO,AUTO-
CONTINUE=NO,
          Do you want to change the theoretical values
for PNT1?
          IF/C1.INPUT=="YES"

```

Verzweigung bei Unterprogrammen

```
CS1      =CALLSUB/CHANGETHEO, :,  
          END_IF/  
          COMMENT/OPER, NO, FULL SCREEN=NO, AUTO-  
CONTINUE=NO,  
          The XYZ theoretical and actual values for  
PNT1 are:  
          "Theo X= "+PNT1.TX  
          "Theo Y= "+PNT1.TY  
          "Theo Z= "+PNT1.TZ  
          -----  
          "Act1 X= "+PNT1.X  
          "Act1 Y= "+PNT1.Y  
          "Act1 Z= "+PNT1.Z  
          ROUTINE/END  
          SUBROUTINE/CHANGETHEO,  
              POINT1={PNT1}:,  
              =  
              DIMINFO/PNT1;ICON,DIMID,FEATID,VERT,HORIZ,, $  
              HEADINGS,,MEAS,,,,,,,,,  
C2      =COMMENT/INPUT,NO,FULL SCREEN=NO,  
          Type the new X theo value for PNT1.  
          "Its current value is "+PNT1.TX  
          ASSIGN/PNT1.TX=C2.INPUT  
C3      =COMMENT/INPUT,NO,FULL SCREEN=NO,  
          Type the new Y theo value for PNT1.  
          "Its current value is "+PNT1.TY  
          ASSIGN/PNT1.TY=C3.INPUT  
C4      =COMMENT/INPUT,NO,FULL SCREEN=NO,  
          Type the new Z theo value for PNT1.  
          "Its current value is "+PNT1.TZ  
          ASSIGN/PNT1.TZ=C4.INPUT  
          ENDSUB/
```

Erläuterung des Beispielcodes

C1=KOMMENTAR/JANEIN

Diese Zeile nimmt und speichert die Eingabe JA oder NEIN des Benutzers.

IF/C1.EINGABE=="JA"

Diese Zeile ist der Ausdruck. Er testet, ob die Eingabe von Kommentar 1 ein JA ist. Lautet die Eingabe JA, dann ist die IF-Anweisung WAHR, und es werden die auf die IF-Anweisung folgenden Anweisungen ausgeführt, in diesem Fall wird das Element **PNT1** gemessen. Wenn NEIN bewegt es sich zum Ausdruck END_IF.

CS1=AUF_{RE}_UNTERPROG/NENNÄNDERN, : ,

Diese Zeile ruft das Unterprogramm NENNÄNDERN auf. Die Messroutine springt nun zur Zeile **UNTERPROGRAMM/NENNÄNDERN**.

UNTERPROGRAMM/NENNÄNDERN

Diese Zeile initialisiert das Unterprogramm NENNÄNDERN. Die Messroutine fährt mit der Ausführung des Codes zwischen dieser Zeile und der **ENDE_UNTERPROG/-**Zeile fort.

PUNKT1={PKT1} : ,

Dies ist das einzige Argument des Unterprogramms. Es ermöglicht dem Unterprogramm den Zugriff auf die Daten des Elements PKT1.

C2=KOMMENTAR/EINGABE, C3=KOMMENTAR/EINGABE, C4=KOMMENTAR/EINGABE

Diese Eingabekommentare nehmen die neuen durch den Benutzer eingegebenen theoretischen X-, Y- und Z-Werte und speichern sie unter C2.INPUT, C3.INPUT bzw. C4.INPUT.

ZUWEISEN/PKT1.TX = C2.INPUT

ASSIGN/PNT1.TX=C2.INPUT Diese Zeile übernimmt den theoretischen X-Wert aus C2.INPUT und weist ihn der Variablen **PNT1.TX** zu. **PNT1.TX** ist eine PC-DMIS-Variable, die den theoretischen X-Wert (gekennzeichnet durch TX) für den Punkt mit der ID-Sprungmarke PNT1.

ZUWEISEN/PKT1.TY=C3.INPUT

Diese Zeile übernimmt den theoretischen Y-Wert aus C3.INPUT und weist ihn der Variablen **PNT1.TY** zu. **PNT1.TY** ist eine PC-DMIS-Variable, die den theoretischen Y-Wert (gekennzeichnet durch TY) für den Punkt mit der ID-Sprungmarke PNT1 enthält.

ZUWEISEN/PKT1.TZ=C4.INPUT

Diese Zeile übernimmt den theoretischen Z-Wert aus C4.INPUT und weist ihn der Variablen **PNT1.TZ** zu. **PNT1.TZ** ist eine PC-DMI-Variable, die den theoretischen Z-Wert (gekennzeichnet durch TZ) für den Punkt mit dem ID-Sprungmarke PKT1 enthält.

ENDE_UNTERPROG/

Diese Zeile beendet das Unterprogramm, und die Messroutine kehrt zu jener Zeile zurück, die direkt auf den Unterprogrammaufruf folgt. In diesem Fall ist es die Anweisung **END_IF/**. Die Messroutine fährt dann mit dem nächsten

Messroutine beenden

Bedienerkommentar fort, der die theoretischen X-, Y- und Z-Werte sowie die X-, Y- und Z-Istwerte anzeigt. Anschließend wird das Werkstückprogramm mit dem Befehl `ROUTINE/ENDE` beendet.

Befehlsmodus-Befehle nach Kommentaren



Nachdem Sie einen PC-DMIS-Kommentar eingefügt haben, müssen Sie, um weitere PC-DMIS-Befehle im Befehlsmodus einzugeben, zunächst *zweimal* die Eingabetaste nach dem Befehl `KOMMENTAR` drücken. Dadurch wird PC-DMIS mitgeteilt, dass Sie dem Kommentar keinen Text mehr hinzufügen möchten, sondern bereit sind, einen neuen Befehl hinzuzufügen.

Messroutine beenden

Über den Menüeintrag **Einfügen | Programmablaufsteuerung | Routine beenden** wird ein Befehl zum Beenden der Routine `ROUTINE/ENDE` in das Bearbeitungsfenster eingefügt. Immer dann, wenn PC-DMIS während des Programmablaufs auf diesen Befehl trifft, wird die Ausführung der Messroutine sofort unterbrochen.

Dieser Befehl ist dann hilfreich, wenn Sie eine Messroutine auf Basis bestimmter Bedingungen früher als üblich beenden wollen.