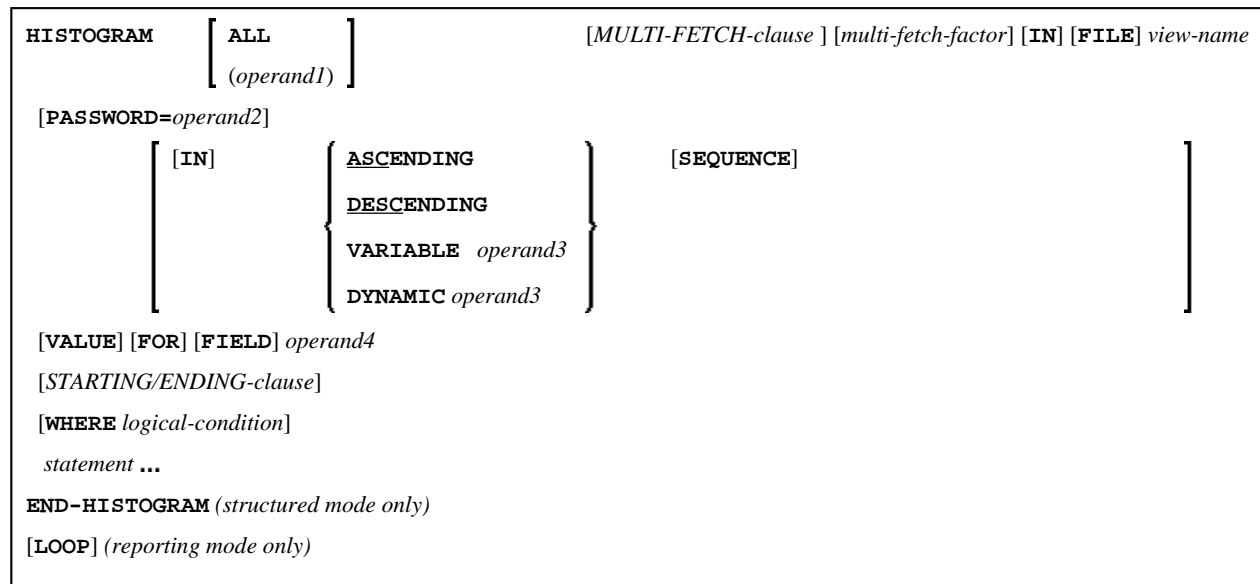


HISTOGRAM



Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkungen
- Syntax-Beschreibung
- Beispiele

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: ACCEPT/REJECT | AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

Funktion

Das Statement HISTOGRAM dient dazu, Werte eines Datenbankfeldes, das als Deskriptor, Subdeskriptor oder Superdeskriptor definiert ist, zu lesen. Die Werte werden direkt von der Adabas Invertierten Liste (bzw. dem VSAM-Index) gelesen.

Das HISTOGRAM-Statement löst zwar eine Verarbeitungsschleife aus, es kann aber auf keine anderen Datenbankfelder als auf das mit dem HISTOGRAM-Statement angegebene Feld zugegriffen werden.

Siehe auch *HISTOGRAM-Statement* im *Leitfaden zur Programmierung*.

Anmerkung:

Bei SQL-Datenbanken: Mit HISTOGRAM erhalten Sie die Anzahl der Reihen, die in einer bestimmten Spalte den gleichen Wert haben.

Einschränkungen

Dieses Statement kann nicht bei XML-Datenbanken oder mit Entire System Server verwendet werden.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition			
	C	S					N	P	I	B											
<i>operand1</i>	C	S					N	P	I	B*										ja	nein
<i>operand2</i>	C	S			A															ja	nein
<i>operand3</i>		S			A															ja	nein
<i>operand4</i>		S			A	N	P	I	F	B	D	T	L							nein	nein

* Format B von *operand1* kann nur mit einer Länge von kleiner gleich 4 benutzt werden.

Syntax-Element-Beschreibung:

<i>operand1</i> / ALL	<p>Begrenzen der Schleifendurchläufe</p> <p>Sie können die Anzahl der Deskriptorwerte, die mit dem HISTOGRAM-Statement gelesen werden sollen, auf eine bestimmte Zahl (<i>operand1</i>) begrenzen, die Sie entweder als numerische Konstante (0 bis 99999999) oder über eine Benutzervariable (die einen Ganzzahlwert enthält) angeben.</p> <p>Andernfalls werden alle Deskriptorwerte gelesen, was Sie zusätzlich durch das Schlüsselwort ALL hervorheben können.</p> <p>Das mit <i>operand1</i> angegebene Limit hat bei diesem Statement Vorrang vor einem mit einem LIMIT-Statement gesetzten Limit.</p> <p>Ist mit dem Profilparameter LT ein kleineres Limit gesetzt, so gilt das LT-Limit.</p> <p>Anmerkungen:</p> <ol style="list-style-type: none"> 1. Wenn Sie eine vierstellige Anzahl von Deskriptorwerten lesen möchten, geben Sie diese mit einer vorangestellten Null an: (0nnnn); denn Natural interpretiert jede vierstellige Zahl in Klammern als Zeilennummer-Referenzierung auf ein Statement. 2. <i>operand1</i> wird zu Beginn des ersten HISTOGRAM-Schleifendurchlaufs ausgewertet. Wird der Wert von <i>operand1</i> innerhalb der HISTOGRAM-Schleife geändert, hat dies keine Auswirkungen auf die Anzahl der gelesenen Werte.
<i>MULTI-FETCH-clause</i>	Siehe <i>MULTI-FETCH-Klausel</i> weiter unten.
<i>view-name</i>	<p>Als <i>view-name</i> geben Sie den Namen eines Views an, der entweder in einem DEFINE DATA-Block oder in einer programmexternen Global oder Local Data Area definiert ist.</p> <p>Der View darf außer dem im HISTOGRAM-Statement verwendeten Feld (<i>operand4</i>) keine anderen Felder enthalten.</p> <p>Ist das im View definierte Feld ein in einer Periodengruppe enthaltenes Feld oder ein multiples Feld, das mit einem Indexbereich definiert ist, dann wird jeweils nur die erste Ausprägung dieses Bereiches vom HISTOGRAM-Statement gefüllt; alle anderen Ausprägungen bleiben von der Ausführung des HISTOGRAM-Statements unberührt.</p> <p>Im Reporting Mode ist <i>view-name</i> der Name eines DDM, falls kein DEFINE DATA LOCAL-Statement benutzt wird.</p>
PASSWORD=<i>operand2</i>	<p>PASSWORD-Klausel:</p> <p>Die PASSWORD-Klausel dient dazu, ein Passwort (<i>operand2</i>) anzugeben, um auf Daten einer passwort-geschützten Adabas-Datei zugreifen zu können. Weitere Informationen hierzu siehe FIND-Statement und PASSW-Statement.</p>

SEQUENCE	<p>SEQUENCE-Klausel</p> <p>Die Klausel gilt nur für Adabas- und SQL-Datenbanken.</p> <p>Mit dieser Klausel können Sie bestimmen, ob die Werte in aufsteigender Reihenfolge oder in absteigender Reihenfolge gelesen werden sollen.</p> <ul style="list-style-type: none"> ● Standardmäßig werden die Werte in aufsteigender Reihenfolge gelesen (was Sie mit dem Schlüsselwort ASCENDING auch ausdrücklich angeben können, aber nicht müssen). ● Wenn die Werte in absteigender Reihenfolge gelesen werden sollen, geben Sie das Schlüsselwort DESCENDING an. ● Wenn erst zur Laufzeit bestimmt werden soll, ob die Werte in aufsteigender oder absteigender Reihenfolge gelesen werden sollen, geben Sie das Schlüsselwort VARIABLE oder DYNAMIC gefolgt von einer Variablen (<i>operand3</i>) an. Der Wert von <i>operand3</i> zu Beginn der HISTOGRAM-Verarbeitungsschleife bestimmt dann die Reihenfolge. <i>operand3</i> muss Format/Länge A1 haben und kann den Wert A (für ASCENDING = aufsteigend) oder D (für DESCENDING = absteigend) enthalten. <ul style="list-style-type: none"> ○ Wenn das Schlüsselwort VARIABLE benutzt wird, wird die Leserichtung (Wert von <i>operand3</i>) beim Start der HISTOGRAM-Verarbeitungsschleife ausgewertet, und bleibt dieselbe, bis die Schleife beendet wird, ungeachtet der Tatsache, ob das Feld für <i>operand3</i> in der HISTOGRAM-Schleife geändert wird oder nicht. ○ Wenn das Schlüsselwort DYNAMIC benutzt wird, wird die Leserichtung (Wert von <i>operand3</i>) vor jedem Einlesen eines Datensatzes in der HISTOGRAM-Verarbeitungsschleife ausgewertet und kann von Datensatz zu Datensatz geändert werden. Dies ermöglicht das Ändern der Blätter-Reihenfolge von aufsteigend in absteigend (und umgekehrt) an einer beliebigen Stelle in der HISTOGRAM-Schleife. <p>Beispiele für SEQUENCE-Klausel:</p> <ul style="list-style-type: none"> ● <i>Beispiel 2 – HISTOGRAM-Statement mit in absteigender Reihenfolge gelesenen Sätzen</i> ● <i>Beispiel 3 – HISTOGRAM-Statement mit variabler Reihenfolge</i>
----------	---

<i>operand4</i>	<p>Deskriptor:</p> <p>Als <i>operand4</i> kann ein Deskriptor, ein Subdeskriptor, ein Superdeskriptor oder ein Hyperdeskriptor angegeben werden.</p> <p>Ein Deskriptorfeld, das Teil einer Periodengruppe ist, kann entweder mit oder ohne Index angegeben werden. Wird kein Index angegeben, so wird ein Datensatz ausgewählt, wenn der Suchwert in irgendeiner Ausprägung gefunden wird. Wird ein Index angegeben, so wird ein Datensatz nur ausgewählt, wenn der Suchwert in der im Index angegebenen Ausprägung gefunden wird. Es muss ein konstanter Index angegeben werden; es darf kein Indexbereich angegeben werden.</p> <p>Ist das angegebene Deskriptorfeld ein multiples Feld, darf kein Index angegeben werden. Ein Datensatz wird ausgewählt unabhängig davon, in welcher Ausprägung des Feldes der Suchwert gefunden wird.</p>
STARTING-ENDING-clause	<p>STARTING/ENDING-Klausel:</p> <p>Mit den Schlüsselwörtern <code>STARTING</code> und <code>ENDING</code> (bzw. <code>THRU</code>) können Sie einen Startwert und einen Endwert angeben, und zwar in Form einer Konstanten oder einer Benutzervariablen. Damit legen Sie fest, ab welchem Wert und bis zu welchem Wert gelesen werden soll.</p> <p>Weitere Informationen siehe <i>Start-/Endwerte angeben</i> weiter unten.</p>

WHERE <i>logical-condition</i>	<p>WHERE-Klausel:</p> <p>Mit der WHERE-Klausel können Sie ein zusätzliches Selektionskriterium in Form einer logischen Bedingung (<i>logical-condition</i>) angeben. Dies wird ausgewertet, <i>nachdem</i> ein Wert gelesen wurde, aber bevor eine weitere Verarbeitung auf der Grundlage dieses Wertes (einschließlich AT BREAK-Verarbeitung) erfolgt.</p> <p>Der in der WHERE-Klausel angegebene Deskriptor muss mit dem im HISTOGRAM-Statement referenzierten Deskriptor identisch sein. Keine anderen Felder von der ausgewählten Datei stehen zur Verarbeitung bei einem HISTOGRAM-Statement zur Verfügung.</p> <p>Die Syntax für eine <i>logical-condition</i> ist im Abschnitt <i>Logische Bedingungen</i> im <i>Leitfaden zur Programmierung</i> beschrieben.</p> <p>Systemvariablen</p> <p>Die Natural-Systemvariablen *ISN, *NUMBER und *COUNTER können bei einem HISTOGRAM-Statement verwendet werden.</p> <p>*NUMBER und *ISN stehen erst nach Auswertung der WHERE-Klausel zur Verfügung. Sie dürfen nicht innerhalb der logischen Bedingung einer WHERE-Klausel eingesetzt werden.</p>		
	<table border="1"> <tr> <td data-bbox="561 1020 727 1192">*NUMBER</td> <td data-bbox="727 1020 1393 1192"> <p>*NUMBER enthält die Anzahl der Datensätze, die den zuletzt gelesenen Wert enthalten.</p> <p>Bezüglich SQL-Datenbanken siehe *NUMBER bei SQL Datenbanken in der <i>Systemvariablen</i>-Dokumentation.</p> </td> </tr> </table>	*NUMBER	<p>*NUMBER enthält die Anzahl der Datensätze, die den zuletzt gelesenen Wert enthalten.</p> <p>Bezüglich SQL-Datenbanken siehe *NUMBER bei SQL Datenbanken in der <i>Systemvariablen</i>-Dokumentation.</p>
*NUMBER	<p>*NUMBER enthält die Anzahl der Datensätze, die den zuletzt gelesenen Wert enthalten.</p> <p>Bezüglich SQL-Datenbanken siehe *NUMBER bei SQL Datenbanken in der <i>Systemvariablen</i>-Dokumentation.</p>		
	<table border="1"> <tr> <td data-bbox="561 1203 727 1486">*ISN</td> <td data-bbox="727 1203 1393 1486"> <p>*ISN enthält die Anzahl der Ausprägung einer Periodengruppe des aktuellen Datensatzes, die den zuletzt gelesenen Wert enthält. Ist der Deskriptor nicht in einer Periodengruppe enthalten, enthält *ISN den Wert 0.</p> <p>Bei SQL-Datenbanken kann *ISN nicht verwendet werden.</p> </td> </tr> </table>	*ISN	<p>*ISN enthält die Anzahl der Ausprägung einer Periodengruppe des aktuellen Datensatzes, die den zuletzt gelesenen Wert enthält. Ist der Deskriptor nicht in einer Periodengruppe enthalten, enthält *ISN den Wert 0.</p> <p>Bei SQL-Datenbanken kann *ISN nicht verwendet werden.</p>
*ISN	<p>*ISN enthält die Anzahl der Ausprägung einer Periodengruppe des aktuellen Datensatzes, die den zuletzt gelesenen Wert enthält. Ist der Deskriptor nicht in einer Periodengruppe enthalten, enthält *ISN den Wert 0.</p> <p>Bei SQL-Datenbanken kann *ISN nicht verwendet werden.</p>		
	<table border="1"> <tr> <td data-bbox="561 1497 727 1608">*COUNTER</td> <td data-bbox="727 1497 1393 1608"> <p>*COUNTER enthält die Gesamtzahl der bisher gelesenen Werte (nach Auswertung der WHERE-Klausel).</p> </td> </tr> </table>	*COUNTER	<p>*COUNTER enthält die Gesamtzahl der bisher gelesenen Werte (nach Auswertung der WHERE-Klausel).</p>
*COUNTER	<p>*COUNTER enthält die Gesamtzahl der bisher gelesenen Werte (nach Auswertung der WHERE-Klausel).</p>		
END-HISTOGRAM	<p>Das für Natural reservierte Schlüsselwort END-HISTOGRAM muss zum Beenden des HISTOGRAM-Statements benutzt werden.</p>		

MULTI-FETCH-Klausel

Anmerkung:

Diese Klausel kann nur bei Adabas-Datenbanken benutzt werden.

$\left[\text{MULTI-FETCH} \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{OF } \textit{multi-fetch-factor} \end{array} \right\} \right]$

Anmerkung:

[MULTI-FETCH OF *multi-fetch-factor*] wird bei den Datenbanktypen ADA und ADA2 nicht ausgewertet. Es erfolgt die Standardverarbeitung (siehe Profilparameter MFSET. Beim Datenbanktyp ADA2 wird die MULTI-FETCH-Klausel komplett ignoriert; siehe *Database Management System Assignments* in der *Configuration Utility*-Dokumentation.

Weitere Informationen siehe *Multi-Fetch-Klausel* (Adabas) im *Leitfaden zur Programmierung*.

Start-/Endwerte angeben

Mit den Schlüsselwörtern STARTING und ENDING (bzw. THRU) können Sie einen Startwert und einen Endwert angeben, und zwar in Form einer Konstanten oder einer Benutzervariablen. Damit legen Sie fest, ab welchem Wert und bis zu welchem Wert gelesen werden soll.

Wenn der angegebene Startwert nicht vorhanden ist, wird der nächsthöhere vorhandene Wert als Startwert genommen. Ist kein höherer Wert vorhanden, wird die HISTOGRAM-Schleife nicht ausgeführt.

Wenn Sie einen Endwert angeben, wird bis einschließlich des Endwertes gelesen.

Für Deskriptoren des Formats A oder B können hexadezimale Konstanten als Start- und Endwert angegeben werden.

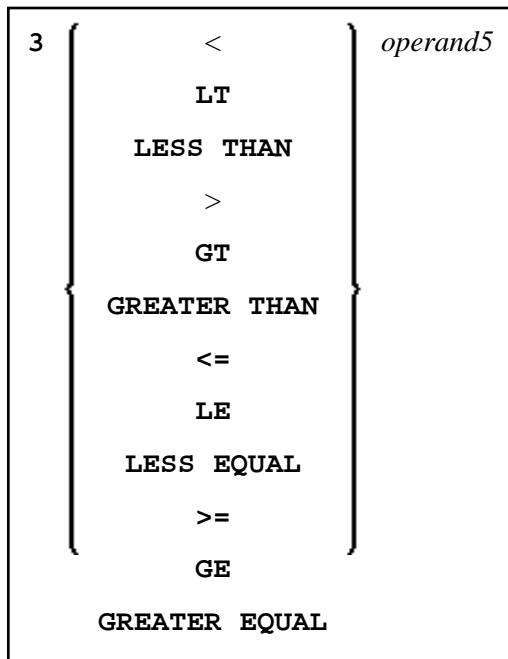
Syntax-Option 1:

$1 \left[\left[\text{STARTING} \right] \left[\begin{array}{l} \text{WITH} \\ \text{FROM} \end{array} \right] \left[\text{VALUES} \right] \textit{operand5} \right] \left[\left[\text{THRU} \right] \left[\text{ENDING AT} \right] \right] \textit{operand6} \right]$

Syntax-Option 2:

$2 \left[\text{STARTING} \right] \left[\text{WITH FROM} \right] \left[\text{VALUES} \right] \textit{operand5} \text{ TO } \textit{operand6}$

Syntax-Option 3:



Anmerkung:

Wenn die Vergleichsoperatoren von Diagramm 3 benutzt werden, dürfen die Optionen ENDING AT, THRU und TO nicht benutzt werden. Diese Vergleichsoperatoren sind auch für das READ-Statement gültig.

Operanden-Definitionstabelle:

Operand	Mögliche Struktur		Mögliche Formate												Referenzierung erlaubt	Dynam. Definition						
<i>operand5</i>	C	S					A	U	N	P	I	F	B	D	T	L					ja	nein
<i>operand6</i>	C	S					A	U	N	P	I	F	B	D	T	L					ja	nein

Syntax-Element-Beschreibung:

STARTING FROM ... ENDING AT/TO	<p>Die Klauseln <code>STARTING FROM</code> und <code>ENDING AT</code> werden benutzt, um das Lesen auf einen vom Benutzer angegebenen Bereich von Werten einzuschränken.</p> <p>Die <code>STARTING FROM</code>-Klausel (= oder <code>EQ</code> oder <code>EQUAL TO</code> oder <code>[STARTING] FROM</code>) legt den Startwert für die Lese-Operation fest. Wenn ein Startwert angegeben wird, beginnt der Lesevorgang bei dem angegebenen Wert. Wenn der Startwert nicht vorhanden ist, wird der nächsthöhere Wert (oder nächstniedrigere Wert für einen absteigenden Lesevorgang) zurückgegeben. Ist kein höherer Wert (oder niedrigerer Wert für <code>DESCENDING</code>) vorhanden, wird die <code>HISTOGRAM</code>-Schleife nicht ausgeführt.</p> <p>Um die Werte auf einen Endwert zu beschränken, können Sie eine <code>ENDING AT</code>-Klausel bei den Schlüsselwörtern <code>THRU</code>, <code>ENDING AT</code> oder <code>TO</code> angeben, die einen einschließenden Bereich implizieren. Immer wenn das Deskriptorfeld den angegebenen Endwert überschreitet, erfolgt eine automatische Beendigung der Schleife. Obwohl die Basis-Funktionalität der Schlüsselwörter <code>TO</code>, <code>THRU</code> und <code>ENDING AT</code> ähnlich ist, so unterscheiden sie sich doch intern durch ihre Funktionsweise.</p>
THRU/ENDING AT	<p>Wenn <code>THRU</code> oder <code>ENDING AT</code> benutzt wird, wird nur der Startwert der Datenbank bekannt gegeben; es wird aber vom Natural-Laufzeitsystem eine Prüfung auf den Endwert vorgenommen, nachdem der Wert von der Datenbank zurückgegeben worden ist.</p> <p>Die Klauseln <code>THRU</code> und <code>ENDING AT</code> können für alle Datenbanken benutzt werden, die die <code>HISTOGRAM</code>-Statements unterstützen.</p>
TO	<p>Wird das Schlüsselwort <code>TO</code> benutzt, werden sowohl der Startwert als auch der Endwert der Datenbank übergeben, und Natural prüft nicht auf Wertebereiche ab. Wenn der Endwert überschritten wird, reagiert die Datenbank genauso, als sei das Dateiende (End-of-File) erreicht worden, und die Datenbank-Schleife wird verlassen. Da die ganzen Bereichsprüfungen von der Datenbank durchgeführt werden, wird der niedrigere Wert (des Bereichs) immer beim Startwert und der höhere Wert beim Endwert angegeben, ganz gleich ob Sie einen Lesevorgang in aufsteigender (<code>ASCENDING</code>) oder absteigender (<code>DESCENDING</code>) Reihenfolge durchführen.</p>

Beispiele

- Beispiel 1 — `HISTOGRAM`-Statement
- Beispiel 2 — `HISTOGRAM`-Statement mit in absteigender Reihenfolge gelesenen Sätzen
- Beispiel 3 — `HISTOGRAM`-Statement mit variabler Reihenfolge

Beispiel 1 — `HISTOGRAM`-Statement

```
** Example 'HSTEX1S': HISTOGRAM (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
2 CITY
```

```

END-DEFINE
*
LIMIT 8
HISTOGRAM EMPLOY-VIEW CITY STARTING FROM 'M'
  DISPLAY NOTITLE
    CITY 'NUMBER OF/PERSONS' *NUMBER *COUNTER
END-HISTOGRAM
*
END
    
```

Ausgabe des Programms HSTEX1S:

CITY	NUMBER OF PERSONS	CNT
MADISON	3	1
MADRID	41	2
MAILLY LE CAMP	1	3
MAMERS	1	4
MANSFIELD	4	5
MARSEILLE	2	6
MATLOCK	1	7
MELBOURNE	2	8

Äquivalentes Reporting Mode-Beispiel: HSTEX1R.

Beispiel 2 — HISTOGRAM-Statement mit in absteigender Reihenfolge gelesenen Sätzen

```

** Example 'HSTDSCND': HISTOGRAM (with DESCENDING)
*****
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 NAME
END-DEFINE
*
HISTOGRAM (10) EMPL IN DESCENDING SEQUENCE FOR NAME FROM 'ZZZ'
  DISPLAY NAME *NUMBER
END-HISTOGRAM
END
    
```

Ausgabe des Programms HSTDSCND:

Page 1 05-01-13 13:41:03

NAME	NMBR
ZINN	1
YOT	1
YNCLAN	1
YATES	1
YALCIN	1
YACKX-COLTEAU	1
XOLIN	1
WYLLIS	2
WULFRING	1
WRIGHT	1

Beispiel 3 — HISTOGRAM-Statement mit variabler Reihenfolge

```

** Example 'HSTVSEQ': HISTOGRAM (with VARIABLE SEQUENCE)
*****
DEFINE DATA LOCAL
1 EMPL VIEW OF EMPLOYEES
  2 NAME
*
1 #DIR      (A1)
1 #STARTVAL (A20)
END-DEFINE
*
SET KEY PF3 PF7 PF8
*
MOVE 'ADKINSON' TO #STARTVAL
*
HISTOGRAM (9) EMPL FOR NAME FROM #STARTVAL
  WRITE NAME *NUMBER
  IF *COUNTER = 5
    MOVE NAME TO #STARTVAL
  END-IF
END-HISTOGRAM
*
#DIR := 'A'
*
REPEAT
  HISTOGRAM EMPL IN VARIABLE #DIR SEQUENCE
    FOR NAME FROM #STARTVAL
    MOVE NAME TO #STARTVAL
    INPUT NO ERASE (IP=OFF AD=0)
      15/01 NAME *NUMBER
      // 'Direction:' #DIR
      // 'Press PF3 to stop'
      /  '      PF7 to go step back'
      /  '      PF8 to go step forward'
      /  '      ENTER to continue in that direction'
    /*
    IF *PF-KEY = 'PF7' AND #DIR = 'A'
      MOVE 'D' TO #DIR
      ESCAPE BOTTOM
    END-IF
    IF *PF-KEY = 'PF8' AND #DIR = 'D'
      MOVE 'A' TO #DIR
      ESCAPE BOTTOM
    END-IF
    IF *PF-KEY = 'PF3'
      STOP
    END-IF
  END-HISTOGRAM
  /*
  IF *COUNTER(0250) = 0
    STOP
  END-IF
END-REPEAT
END

```

Ausgabe des Programms HSTVSEQ:

Page 1

05-01-13 13:50:31

ADKINSON 8
AECKERLE 1

AFANASSIEV	2
AHL	1
AKROYD	1
ALEMAN	1
ALESTIA	1
ALEXANDER	5
ALLEGRE	1

MORE

Nach Drücken von EINGABE:

Page 1

05-01-13 13:50:31

ADKINSON	8
AECKERLE	1
AFANASSIEV	2
AHL	1
AKROYD	1
ALEMAN	1
ALESTIA	1
ALEXANDER	5
ALLEGRE	1

AKROYD	1
--------	---

Direction: A

Press PF3 to stop
PF7 to go step back
PF8 to go step forward
ENTER to continue in that direction