M20

ISAM

Indexsequentieller Dateizugriff

Programmier-Handbuch

olivetti

Das Handbuch dient der Information, sein Inhalt ist ohne ausdrückliche schriftliche Vereinbarung nicht Vertragsgegenstand. Technische Änderungen behalten wir uns vor. Die angegebenen Daten sind lediglich Nominalwerte.

© Copyright 1983 by Deutsche Olivetti DTS GmbH.

GLIEDERUNG SEITE

1. Allgemeines 1

1.1 Arbeitsweise von ISAM 1

1.2 Möglichkeiten mit ISAM 2

2. Verwendung von ISAM am M20 4

2.1 Benötigte Files 4

2.2 Erstellen der Systemumgebung 6

2.3 Erstellen eines BASIC-Programms mit ISAM-Aufrufen 7

3. Übersicht über die ISAM-Funktionen 10

4. ISAM-Variablen 12

4.1 mögliche Eingangsparameter 12

4.2 mögliche Ausgangsparameter 15

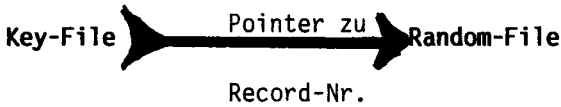
5. Return-Codes und ihre Bedeutung 17

6. Einzelbeschreibung der ISAM-Funktionen, alphabetisch 20

1. Allgemeines

1.1 Arbeitsweise von ISAM

ISAM ist ein Paket, das die Verarbeitung indexsequentieller Dateistrukturen ermöglicht. Die indexsequentielle Dateistruktur von ISAM setzt sich aus mindestens zwei Datenfiles zusammen und sieht schematisch folgendermaßen aus:



mit Keys (verwaltet von ISAM)

mit Datensätzen (verwaltet im BASIC Programm selbst)

Das Assemblerprogramm **isam.sav** verwaltet selbständig alle alphanumerischen Keys eines Key-Files auf Diskette. Numerische Keys können mit Hilfe der BASIC-Funktionen **STR$** bzw. **MKI$**, **MKS$**, **MKD$** in Alpha-Keys konvertiert werden. Im Key-File wird ein Pointer mitverwaltet, der angibt, unter welcher Record-Nr. der jeweilige Datensatz (mit oder ohne Key als Bestandteil) im Random-File zu verarbeiten ist. Das Key-File wird von ISAM selbst ständig sortiert gehalten (Verwendung einer binären Baum-Struktur; B^+-tree). Das Lesen bzw. Schreiben des Datensatzes, der einem bestimmten Key zugeordnet ist, muß per BASIC-Programm erfolgen. ISAM gibt nur die Record-Nr. des Datensatzes bekannt.

Diese Logik setzt feste Recordlänge im dem Key-File zugeordneten Random-File (im weiteren Data-File genannt) voraus. Für Organisation, Anlegen und Verwalten des gesamten Random-Files hat der Programmersteller selber Sorge zu tragen.

Der neu zu erfassende, zu lesende oder geänderte Datensatz muß in das zugeordnete Random-File per **PUT**-Anweisung geschrieben bzw. per **GET**-Anweisung daraus gelesen werden.

1.2 Möglichkeiten mit ISAM

ISAM bietet folgende Möglichkeiten der indexsequentiellen Dateiverarbeitung:

- Es kann beim Öffnen des Key-Files festgelegt werden, ob dieser anglegt werden soll, wenn nicht vorhanden, oder nicht.
- Die Keys im selben Key-File dürfen unterschiedlich lang sein. Sie müssen allerdings alphanumerisch und maximal 110 Zeichen lang sein.
- Es können für das gleiche Data-File mehrere Key-Files geführt werden, was auf jeden Datensatz den Zugriff über mehrere Keys ermöglicht ("Sekundäre Keys"). Für jeden sekundären Key muß ein eigener Key-File eingerichtet werden. Maximal 4 sekundäre Keys pro primäres Key-File sind möglich, falls die Key-Files nicht die Keys als Bestandteil in jedem Datensatz enthalten.

- Es kann vereinbart werden, daß im selben Key-File einem Key mehrere Datensätze zugeordnet werden können ("doppelte Keys").
- Es können Keys so gelöscht werden, daß dabei die zugeordnete Record-Nr. nicht freigegeben wird, d.h., der Datensatz ist für spätere Neuerfassungen nie zum Überschreiben freigegeben ("logisches Löschen").
- Es können Keys so gelöscht werden, daß dabei die zugeordnete Record-Nr. freigegeben wird, d.h., bei späterer Neuerfassung wird die Record-Nr. des gelöschten Keys wieder verwendet ("endgültiges Löschen").
- Es können linksbündig partielle Keys gesucht werden; es wird dann gesucht, ob ein Key vorhanden ist, der genau dem gesuchten Key entspricht. Falls ein solcher Key nicht im File ist, wird der nächste Key gelesen ("generisches Lesen"). Dies ist insbesondere nützlich, um schnell den Anfangswert einer Gruppe von Datensätzen zu suchen, wenn nicht bekannt ist, ob ein ganz bestimmter Key als Anfangswert vorhanden ist (z.B. Listen aller Namen, die mit "B" anfangen).
- Es kann der nächste Key in der sortierten Folge bzw. der vorherige Key abgefragt werden ("Lesen nächster Key" bzw. "Lesen vorheriger Key"). Dies ist insbesondere nützlich, um die Ermittlung des ersten Keys einer Gruppe zu unterstützen (nach generischem Lesen), bzw. um bei doppelten Keys nach einem bestimmten Data-Record zu suchen.

- Es kann jederzeit der Zustand der indexsequentiellen Dateistruktur abgefragt werden: Anzahl aktiver Datensätze, doppelte Keys zulässig J/N, Anzahl durch Löschen freigegebener Record-Nrn. ("Statusinformation").

  Achtung: ISAM vergibt die Record-Nrn. im Data-File nicht in der Reihenfolge der Erfassung der Records!

- Es ist grundsätzlich möglich, alle sekundären Keys wie primäre Keys zu lesen (Funktionen **RK, RP, RN, RG**). Bei entsprechendem Aufbau der indexsequentiellen Struktur können sekundäre Keys wie primäre Keys behandelt werden.

2. Verwendung von ISAM am M20

2.1 Benötigte Files

Folgende Files sind zur Arbeit mit ISAM nötig:

- isam.sav (assembler-Routine; im ISAM-Paket enthalten) isam.sav liegt in mehreren Versionen vor. Die unter PCOS-Release 1.3 verwendbare ist nicht kompatibel mit der unter PCOS 2.Ø ff. einzusetzenden. (Beim Versuch des Ladens mit **pl is** wird Error 99 gemeldet.) Die unter Rel. 1.3 erstellten Key-Files sind kompatibel mit denen von Rel. 2.Ø.
- isam.bas (BASIC-Programm-Modul; im ISAM-Paket enthalten)
- das BASIC-Programm, das ISAM aufruft
- das Random-File, in dem die eigentlichen Datensätze gespeichert sind

- das Key-File bzw. die Key-Files, in dem bzw. in denen die primären und eventuellen sekundären Keys und die zugeordneten Record-Nrn. von ISAM verwaltet werden. Das Anlegen geschieht mit einer der beiden ISAM-Funktionen **OC** oder **OF**. Eine evtl. notwendige Vergrößerung des Key-Files nimmt PCOS automatisch vor (wie bei allen anderen Files auch), sofern noch ausreichend Sektoren auf der Diskette frei sind (vgl. zu dieser Problematik PCOS-Befehl **ss**, Parameter 'anzulegende Sektoren'). Die Größe des Key-Files kann vom Bediener durch die Reihenfolge der Aufnahme der Keys beeinflußt werden. Es gilt die Faustregel: je unsortierter die Reihenfolge der Erfassung, desto kleiner das Key-File (Grund: B^+-Binärbaum-Logik).
  Zur Bestimmung der für ein Key-File von ISAM auf Diskette benötigten Anzahl Bytes gilt im ungünstigsten Falle: Jeder Key auf Diskette benötigt dort 3 Bytes + die aktuelle Länge des Keys in Zeichen (max. Länge: 110 Zeichen). Zur anfänglichen Reservierung einer bestimmten Byteanzahl kann der PCOS-Befehl **fn** eingesetzt werden.

- Wird ein Key-File mit den ISAM-Funktionen **OC** bzw. **OF** angelegt oder muß es während des Arbeitens mit ISAM vergrößert werden (möglich nach Aufruf der ISAM-Funktionen **DR**, **WA** oder **SA**), werden die dafür benötigten Bereiche auf Diskette von ISAM selbst entsprechend vorbereitet und belegt. Dies gilt nicht für das zugeordnete Random-File!

2.2 Erstellen der Systemumgebung

a) Steuern Sie die PCOS-Ebene an (erkenntlich am **>**).

b) Laden Sie in der PCOS-Ebene das Assembler-Modul isam.sav in den Arbeitsspeicher (**pl is** oder **is** eingeben, nachdem die PCOS-Systemdiskette und die Diskette mit dem ISAM-Paket eingelegt wurde). ISAM belegt ca. 9 KB im Nicht-BASIC-Bereich des Arbeitsspeichers.

c) Definieren Sie die Anzahl nötiger File-Puffer für alle gleichzeitig offenen Files (incl. Key-Files) sowie die Größe aller dieser Puffer in Bytes. Dies ist die maximale Anzahl von Bytes, die Sie mit Schreib- oder Lesezugriff ansprechen wollen; i.a. deckt sich diese Größe mit der Recordlänge des Files mit den längsten Records. Verwenden Sie dazu den PCOS-Befehl **sb**.

Bitte beachten: **sb** ist nur in der PCOS-Ebene wertändernd einsetzbar. Eventuell kann allerdings eine PCOS-Prozedur, die in BASIC aufrufbar ist, eingesetzt werden (vgl. 'Wechsel des Ein/Ausgabemediums; Prozeduren' im PCOS-Handbuch).

Geben Sie den für BASIC zur Verfügung stehenden Speicherplatz an (160-KB-RAM und größer: 48 KB, 128-KB-RAM: 32 KB). Setzen Sie diesen Wert als Arbeitsspeichergrösse für **sb** an. Definieren Sie die maximale Anzahl in einem Programm verwendeter Windows.

Beispiel:

3 Key-Files, 2 zugeordnete Data-Files und weitere 2 Files seien irgendwann einmal gleichzeitig offen, dies ergibt eine File-Puffer-Anzahl von 7.

Die maximale Key-Länge in Ihren Programmen sei 40, die Record-Länge des Data-Files (bzw. Datenfiles) mit dem längsten Record in allen Programmen sei 250; dies ergibt eine max. Record-Länge von 250. Es liegt ein M20 mit Speichererweiterung vor. Im Programm mit den meisten Windows werden 8 Windows benötigt.
Es ergibt sich
sb 7,48000,8,250

d) Speichern Sie die so konfigurierte Systemumgebung auf einer mit **vf** formatierten leeren Diskette oder der HDU als Betriebssystem ab (**ps**). Bei jedem Laden des Betriebssystems ist nunmehr isam.sav sofort im Speicher resident und die entsprechenden Parameter für BASIC sind sofort gültig.

2.3 Erstellen eines BASIC-Programms mit ISAM-Aufrufen

a) Steuern Sie die BASIC-Ebene an (**ba**).

b) Schreiben Sie Ihr BASIC-Programm. Verwenden Sie dabei keine Zeilennumern über 60000. Dimensionieren Sie am Anfang des Programms mit Hilfe der Anweisungen
DIM ISAM%(10), ISAM$(10) zwei Arrays. Über diese Variablen kommunizieren Sie später mit ISAM.

c) Binden Sie das BASIC-Modul isam.bas in Ihr Programm ein mit:
merge "isam.bas" (geben Sie bei allen Filenamen vor dem eigentlichen Namen noch Ø: bzw. 1: an, um die Diskettenstation anzusprechen, auf der Ihr File liegt. Beispiel: **merge** "1:isam.bas")

d) Initialisieren Sie ISAM mit Hilfe der **MI**-Funktion. Die **MI**-Funktion darf nur einmal aufgerufen werden. Führen Sie deshalb den Funktionsaufruf in Ihrem BASIC-Programm zum Selbststart durch, mit dem Sie Ihr Programmpaket starten (Filename muß sein: INIT.BAS). INIT.BAS muß auf der Diskette liegen, die Sie zum Laden des BASIC-Interpreters verwenden (**ba**).

e) Öffnen Sie das entsprechende Key-File mit einer geeigneten ISAM-Funktion; sollte das Key-File noch nicht vorhanden sein, stellt ISAM für das Anlegen eigene Funktionen bereit (**OC** bzw. **OF**). Das Öffnen der Key-Files muß in allen Programmen erfolgen, die mit den entsprechenden Key-Files kommunizieren. Am Programmende müssen die Key-Files mit der Funktion **CL** wieder geschlossen werden.

f) Wann immer Sie mit ISAM kommunizieren wollen, gehen Sie folgendermaßen vor:
   - Öffnen Sie Ihr Data-File.
   - Definieren Sie die Eingangsparameter an ISAM (Wertzuweisung an bestimmte Elemente der Arrays ISAM**%()** und ISAM**$()**, je nach gewünschter Funktion).
   - Rufen Sie ISAM mit **GOSUB** 60000 auf.
   - Prüfen Sie die Ausgangsparameter von ISAM, insbesondere den Return-Code, und arbeiten Sie in Abhängigkeit von diesem Code weiter (z.B. Lesen oder Schreiben des Data-Files mit Hilfe von **LSET-**, **RSET-**, **GET-**, und **PUT**-Anweisungen, ggf. unter Verwendung der Funktionen **MKI$**, **MKS$**, **MKD$**, **CVI**, **CVS** und **CVD**).

Beispiel (ein vorhandener Satz soll verändert werden: das Key-File sei unter der Nr. 3 eröffnet, das Data-File unter der Nr. 5; der Record-Aufbau im Data-File sei mit **FIELD** festgelegt):

```
 60  REM Erfassen Key am Schirm in T$
 .
 .
 .
100  ISAM$(2)=T$    : REM  Key bekanntgeben
110  ISAM%(1)= 3    : REM  Nr. des Key-Files bekannt
                       geben
120 ISAM$(1)="RK"   : REM  Funktion "Lesen Key" be-
                       kanntgeben
130  GOSUB 60000    : REM  ISAM aufrufen
140  IF ISAM%(8)=51 THEN PRINT
     T$; "NICHT VORHANDEN": GOTO 60
150  IF ISAM%(8)<>Ø THEN PRINT "FEHLER":STOP
160    REM    Erfassen  Data-Record  am  Schirm  und
 Zuweisen auf Feldvariable(n) mit LSET bzw.
          RSET
 .
 .
 .
200  PUT 5, ISAM%(9) : REM  update zugeordneter Re-
                            cord im Data-File
```

3. Übersicht über die ISAM-Funktionen

MI Initialisierung von ISAM

MS Abfrage des Status einer indexsequentiellen Dateistruktur

OO Öffnen eines vorhandenen Key-Files; bei Nichtvorhandensein wird es nicht angelegt und somit nicht geöffnet

OC Anlegen und Öffnen eines nicht vorhandenen Key-Files, bei Vorhandensein wird es nicht geöffnet

OF Öffnen eines Key-Files, bei Nichtvorhandensein wird es vorher angelegt

CL Schließen eines Key-Files mit endgültigem Abspeichern der aktuellen Key-Struktur

RK Lesen eines genau definierten primären Keys, bei doppelten Keys des ersten davon

SK Lesen eines genau definierten primären Keys, bei doppelten Keys des ersten davon

RG Lesen des angegebenen primären Keys, bei Nichtvorhandensein des nächsten

SG Lesen des angegebenen sekundären Keys, bei Nichtvorhandensein des nächsten

RN Lesen des nächsten primären Keys

SN Lesen des nächsten sekundären Keys

RP Lesen des vorherigen primären Keys

SP Lesen des vorherigen sekundären Keys

WA Einfügen eines neuen primären Keys in das Key-File (ggf. doppelter Key)

SA Einfügen eines neuen sekundären Keys in das Key-File (ggf. doppelter Key)

KD logisches Löschen eines primären Keys (ohne Freigabe der zugeordneten Data-File-Recordnummer)

SD logisches Löschen eines sekundären Keys (ohne Freigabe der zugeordneten Data-File-Recordnummer)

DR Löschen eines primären Keys mit Freigabe der zugeordneten Data-File-Recordnummer zum späteren Überschreiben des Data-File-Records

4. ISAM-Variablen

Die Kommunikation zwischen ISAM und dem BASIC-Programm findet über die Array-Variablen ISAM**$()** und ISAM**%()** statt. Die folgenden Tabellen beschreiben diese Variablen:

4.1 mögliche Eingangsparameter

ISAM%(1) Key-File-Nummer

Dies ist die Nummer des Key-Files. Sie wird bei allen Funktionen gebraucht, um anzuzeigen, welches der geöffneten Key-Files angesprochen wird. Sein Wert muß zwischen 1 und 15 liegen (abhängig von **sb**) und muß niedriger sein als die niedrigste File-Nr., die in BASIC für Random- oder sequentielle Files verwendet wird.

ISAM%(5) Data-Record-Nummer

Diese Variable wird verwendet, um einen neu zu erfassenden oder zu löschenden sekundären Key mit einem Record zu verbinden, der bereits existiert (Funktionen **SA** und **SD**). Die Data-Record-Nummer muß auch beim Arbeiten mit doppelten Keys angegeben werden.

ISAM%(6) Zusätzliche Pufferanzahl

Diese Variable wird gebraucht bei der Funktion **MI**, um die Gesamtzahl der Puffer festzulegen, die zusätzlich zu den Puffern, die sowieso zum Arbeiten mit der indexsequentiellen Filestruktur nötig sind, eingerichtet werden sollen.

Beim Arbeiten mit einer indexsequentiellen Dateistruktur sind grundsätzlich mindestens 2 Puffer nötig:

- der Puffer für die Data-Records (das Random-File)
- der Puffer für das Key-File (wird von der Funktion **MI** automatisch aufbereitet, muß aber bei **sb** kalkuliert werden).

Die zusätzlichen Puffer müssen bei **sb** nicht kalkuliert werden. Soll mit der Möglichkeit des Löschens gearbeitet werden, muß mindestens ein zusätzlicher Puffer bei **MI** definiert werden. Eine höhere Anzahl zusätzlicher Puffer beschleunigt die Balancierung der Keys für das Key-File, die von ISAM automatisch vorgenommen wird.

ISAM%(7) Maximale Anzahl der offenen Key-Files

Diese Variable wird bei der Funktion **MI** gebraucht, um die maximale Anzahl von Key-Files festzulegen, die zugleich offen sind.
Sein Wert muß zwischen 1 und 15 sein (abhängig von **sb**) und niedriger liegen als die niedrigste File-Nr., die in BASIC für Random- oder sequentielle Files verwendet wird.

ISAM$(1) Funktions-Code

Diese Variable gibt an, welche Funktion jeweils von ISAM durchgeführt werden soll.

Gültige Werte sind:

RK = Lesen Key (Read Key)

RG = Lesen erster Key der Gattung (Read Generic)

RN = Lesen nächster Key (Read Next)

RP = Lesen vorheriger Key (Read Previous)

WA = Schreiben neuen Key (Write Add)

DR = Löschen Key und Freigabe Record-Nr. (Delete Record)

KD = logisches Löschen Key (Key Delete)

SK = sekundärer Key: Lesen Key

SG = sekundärer Key: Lesen erster Key der Gattung

SN = sekundärer Key: Lesen nächster Key

SP = sekundärer Key: Lesen vorheriger Key

SA = sekundärer Key: Schreiben neuen Key

SD = sekundärer Key: Löschen Key

OO = Öffnen Key-File ohne Anlegen

OC = Anlegen Key-File

OF = Öffnen/Anlegen Key-File

CL = Schließen Key-File

MI = Vorbereiten ISAM-Struktur

MS = Abfragen ISAM-Status

ISAM$(2) zu verarbeitender Key

Er wird bei den folgenden Funktionen benötigt:

RK/SK

RG/SG

WA/SA

DR

KD/SD

ISAM$(4) Name des Key-Files

Diese Variable spezifiziert den Namen des Key-Files. Sie wird gebraucht bei **OC**, **OF** und **OO**. Der Filename muß den Regeln zur Bildung von Filenamen unter PCOS genügen.

ISAM$(6) Schalter doppelte Keys

Diese Variable wird gebraucht, wenn ein neues Key-File angelegt wird (Funktionen **OC** und **OF**). Dabei ist festzulegen, ob doppelte Keys in diesem File erlaubt sind oder nicht.
Der Wert kann entweder **"D"** (doppelte Keys möglich) oder **"U"** (keine doppelten Keys zulässig) sein. Der Default-Wert ist **"U"**.

4.2 mögliche Ausgangsparameter

ISAM%(1) Key-File-Nummer

Dieser Wert wird von der **OO**-Funktion geliefert und gibt an, unter welcher File-Nummer das Key-File geöffnet wurde.

ISAM%(6) Anzahl aktiver Data-Records

Dieser Wert tritt auf bei der Funktion **MS** und gibt an, wieviele Keys (inklusive doppelter Keys) und damit Data-Records unter der indexsequentiellen Struktur gerade aktiv sind.

ISAM%(7) Noch nicht überschriebene gelöschte Records

Dieser Wert tritt auf bei der Funktion **MS** und gibt an, wieviele gelöschte primäre Keys noch nicht durch neuerfaßte primäre Keys überschrieben sind.

ISAM%(8) Return-Code

Dieser kann folgende Werte annehmen:

ØØ = Normale Rückkehr aus ISAM
31 = ungültiger ISAM-Funktionsaufbau
41 = Syntax-Fehler
51 = Key nicht gefunden
61 = doppelter Key
71 = Open/close-Fehler
81 = Disk-Fehler

ISAM%(9) Data-Record-Nummer

Dieser Wert gibt an, welche Record-Nr. im Data-File zu bearbeiten ist, falls ein neuer Record erfaßt bzw. ein vorhandener Record überschrieben oder gelesen werden soll. Voraussetzung ist, daß aufgrund des Return-Codes die entsprechende Operation sinnvoll ist.

ISAM$(2) Wert des Keys

Dieser Wert wird geliefert, wenn eine andere Lesefunktion als **RK** eingesetzt wurde, also **RG/SG** oder **RN/SN** oder **RP/SP**.

ISAM$(6) Schalter doppelte Keys J/N

Dieser Wert wird ausgegeben von **MS**, um den Typ des Key-Files mitzuteilen (**"U"** oder **"D"**, vgl. Eingangsparameter ISAM$(6)).

5. Return-Codes und ihre Bedeutung

Die folgende Tabelle gibt einen Überblick über die möglichen Return-Codes:

Code-Nr.	Bedeutung	mögliche Funktionen
ØØ	Normale Rückkehr Dies zeigt an, daß die gefragte Funktion erfolgreich durchgeführt und daß der gelesene Key gültig ist.	alle
31	Ungültige Funktionsanweisung Dies zeigt an, daß die Folge von Funktionsanweisungen falsch war. Dies kommt z.B. vor, wenn der erste ISAM-Aufruf kein **MI** war.	alle

Code-Nr.	Bedeutung	mögliche Funktionen
41	Syntax-Fehler Diese zeigt an, daß ISAM betreffende Werte ungültig waren.	alle
51	Key nicht gefunden Dies zeigt an, daß der gewünschte Key aus einem der nachfolgende Gründe nicht gefunden wurde:	
	1. der gefragte Key existiert nicht	**RK/SK, KD/SD, DR**
	2. der gefragte Key wäre höher als der höchste Key im Key-File	**RG/SG**
	3. Es wurde ein Versuch gemacht, über das logische Ende des Key-Files hinaus zu lesen.	**RN/SN**
	4. Es wurde versucht, vor de logischen Anfang des Key-Files zu lesen.	**RP/SP**
	In allen 4 Fällen ist die angegebene Record-Nummer ungültig!	
61	doppelter Key Dies zeigt an, daß der Key, der zusätzlich eingefügt werden soll, den gleichen Wert hat wie ein Key, der bereits im Key-File ist. Dieser Code wird nur angegeben, wenn bei **OC** bzw. **OF** für ISAM**$**(6) **"U"** gesetzt wurde.	**WA/SA**

Code-Nr.	Bedeutung	mögliche Funktionen
71	<u>Open/Close-Fehler</u>	
	Dies zeigt an, daß die spezifizierte Funktion aus einem der folgenden Gründe scheiterte:	
	1. die gefragte File-Nummer ist bereits für ISAM vergeben	**OO, OC, OF**
	2. der gefragte Filename existiert für ISAM nicht	**OO, OC, OF**
	3. die gefragte File-Nr. ist noch nicht für ISAM vergeben	alle außer **OO, OC, OF**
81	<u>Disk-Fehler</u>	
	Dies zeigt an, daß ein Diskfehler aufgetaucht ist, von dem ISAM sich nicht erholen kann:	
	1. das Inhaltsverzeichnis ist voll	**OC, OF**
	2. auf der Diskette ist kein belegbarer Platz mehr	**WA/SA, DR**

FUNKTION: **CL**
(close file)

WIRKUNG: Schließen eines Key-Files, Abschluß der Verwaltung des betreffenden Key-Files und Freigabe der Filenummer

EINGANGS-PARAMETER: ISAM%(1) Key-File-Nummer
ISAM$(1) **"CL"**

AUSGANGS-PARAMETER: ISAM%(8) Return-Code

MÖGLICHE RETURN-CODES: Ø, 31, 41, 71

BEMERKUNGEN:
- Der Return-Code 71 wird gegeben, wenn die File-Nr. nicht von ISAM verwaltet wird.
- Das Schließen von Key-Files mit der **CLOSE**-Anweisung ist nicht möglich.
- Solange kein **CL** für ein Key-File erfolgt ist, ist nicht sichergestellt, daß die Struktur des Key-Files auf Diskette gesichert ist. Erst nach **CL** ist das Key-File mit Sicherheit auf dem neuesten Stand auf Diskette gespeichert.

BEISPIEL:

```
200 ISAM%(1)=1:ISAM$(1)="CL":GOSUB 60000
210 IF ISAM%(8)<>0 THEN 600 :REM FEHLERBEDINGUNG
220 REM Key-File Nr. 1 geschlossen + alle Updates für dieses
        Key-File durchgeführt!
```

VERWEISE: **OO, OC, OF**

FUNKTION: **DR**
(delete code)

WIRKUNG: Löschen eines primären Keys aus dem Key-File unter Freigabe der zugeordneten Data-Record-Nr. zum späteren Überschreiben durch einen neuen Datensatz

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM%(5)	Nr. des Data-Records (nur bei doppelten Keys nötig)
	ISAM$(1)	**"DR"**
	ISAM$(2)	zu löschender Key
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	Nr. des freigegebenen Data-Records

MÖGLICHE RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- **DR** sollte nur für primäre Keys eingesetzt werden
- Der von **DR** betroffene Key wird aus dem Key-File entfernt, so daß für einen neuen Key Platz wird.
- Wird mit sekundären Keys gearbeitet, führt **DR** zur Zerstörung der gesamten indexsequentiellen Dateistruktur, es sei denn, für alle sekundären Keys im freigegebenen Data-Record wurde vorher die **KD**- oder die **SD**-Funktion eingesetzt. Dazu müssen die dort zu löschenden Keys bekannt sein. Dies ist nur möglich, wenn die sekundären Keys auch im Data-Record selber stehen.

- Wenn mit Löschen von Records gearbeitet wird, empfiehlt sich grundsätzlich, ein Löschkennzeichen im Data-Record zu führen, das Auskunft darüber gibt, ob der betreffende Satz vielleicht schon mit einer **DR**-Funktion bearbeitet, aber noch nicht durch einen neuen Data-Satz überschrieben wurde (vgl. Beispiel).
- Das Reaktivieren eines mit **DR** gelöschten Keys bzw. Data-Records ist nicht möglich.
- Die den gelöschten Keys zugeordneten Record-Nrn. im Data-File werden von ISAM in einem zusätzlichen Puffer verwaltet (LIFO-Prinzip). Dieser muß unbedingt bei **MI** eingerichtet worden sein, ansonsten wird ISAM fehlerhaft arbeiten (ohne Fehlermeldungen!).

BEISPIEL:

```
0   REM Anwendung von DR
30 ON ERROR GOTO 2801:REM Fehlerbehandlungsroutine mit
31 REM selbstdefinierten Fehlern Nrn. 131,141,171,181
100 OPEN "R",6,"1:DATAF.rnd",128:FIELD 5,10 AS B.SK1$,
    15 AS B.SK2$,8 AS B.SK3$,35 AS B.SK4$,59 AS B.REST$,
    1 AS B.LOESCHKZ$
101 REM B.LOESCHKZ$ = CHR$(n); Bedeutung von n:
102 REM n=0--> Data-Record mit Primärkey aktiv
103 REM n=32-->kein sek.Key vorhanden; DR erfolgt,
105 REM        Data-Record aber noch nicht überschrieben
106 REM n=1=2^0--> nur sek.Key1 vorh.; Primärkey vorh.
107 REM n=2=2^1--> nur sek.Key2 vorh.;        "
108 REM n=4=2^2--> nur sek.Key3 vorh.;        "
109 REM n=8=2^3--> nur sek.Key4 vorh.;        "
110 REM n=16=2^4-->Primärkey mit KD gelöscht
111 REM Summation möglich: Beispiel:
113 REM n=11-->vorh. sek.Keys: 1+2+4, Primärkey vorh.
114 REM max. Wert für n ist 32
115 REM öffnen alle 5 Key-Files
                    ⋮
500 REM EINGABE PRIMÄRER KEY AUF VARIABLE T$
520 ISAM%(1)=1:ISAM$(1)="RK":ISAM$(2)=T$
530 GOSUB 60000:REM keine doppelten Keys seien möglich
535 IF ISAM%(8)=51 THEN PRINT "nicht vorhanden":GOTO 500
540 IF ISAM%(8)<>0 THEN ERROR 100+ISAM%(8)
550 GET 6,ISAM%(9):REM ISAM%(9) liefert Recordnr. für SD
555 IF B.LOESCHKZ$<>CHR$(0) THEN GOSUB 22501
560 ISAM$(1)="DR":ISAM%(1)=1:ISAM$(2)=T$:GOSUB 60000
565 IF ISAM%(8)<>0 THEN ERROR 100+ISAM%(8)
566 LSET B.LOESCHKZ$=CHR$(32):PUT 6,ISAM%(9)
                    ⋮
970 CLOSE 6:ISAM$(1)="CL":FOR I%=1 TO 5:ISAM%(1)=I%
980 GOSUB 60000:IF ISAM%(8)<>0 THEN ERROR ISAM%(8)+100
985 NEXT I%:REM ENDVERARBEITUNG
                    ⋮
22500 REM SD für sekundäre Keys
22501 A%=ASC(B.LOESCHKZ$):ISAM$(1)="SD":I%=3
22503 REM A%>15 hier nicht möglich
22508 WHILE A%>0 AND I%>=0
22510 IF A%>=2^I% THEN ON I%+1 GOTO 22513,22523,22533,22543
                  ELSE 22580
22513 ISAM$(2)=B.SK1$:GOTO 22550
22523 ISAM$(2)=B.SK2$:GOTO 22550
22533 ISAM$(2)=B.SK3$:GOTO 22550
22543 ISAM$(2)=B.SK4$
22550 ISAM%(1)=I%+2:GOSUB 60000
22555 IF ISAM%(8)<>0 THEN ERROR 100+ISAM%(8)
22560 A%=A%-2^I%
22580 I%=I%-1
22590 WEND:RETURN
```

VERWEISE: **KD, SD**

FUNKTION: **KD**
(key delete)

WIRKUNG: Löschung eines primären Keys aus dem Key-File ohne Freigabe der zugeordneten Data-Record-Nr. zum späteren Überschreiben (logisches Löschen)

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM%(5)	Data-Record-Nr. (nur bei doppelten Keys nötig)
	ISAM$(1)	**"KD"**
	ISAM$(2)	zu löschender Key
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	betroffene Data-Record-Nr.

MÖGLICHE RETURN-CODES:

- Das Reaktivieren eines mit **KD** gelöschten Keys über ISAM ist nicht möglich. Der betreffende Data-Record bleibt allerdings vorhanden.
- Der Key wird aus dem Key-File gelöscht und später durch einen neuen Key überschrieben
- Für das Löschen von Keys muß bei **MI** ein zusätzlicher Puffer eingerichtet worden sein.
- Wird mit Löschen gearbeitet, empfiehlt es sich, für jeden Key im Data-Record ein Löschkennzeichen zu setzen, das Auskunft darüber gibt, ob für den betreffenden Data-Record noch ein (primärer bzw. sekundärer) Key von ISAM verwaltet wird (vgl. Beispiel zu **DR**).

BEISPIEL:

```
900 REM Data-File ist OPEN unter Nr. 5, Key-File geöffnet
        unter Nr. 2
910 ISAM%(1)=2:ISAM$(1)="KD":ISAM$(2)="A6083"
920 ISAM%(5)=DATARECNR%:REM Recordnummer bei doppeltem Key
930 GOSUB 60000:IF ISAM%(8)<>0 THEN 600:REM nicht vorhanden
    oder Fehlerbedingung
```

Verweise: **SD, DR**

FUNKTION: **MI**
(method initialise)

WIRKUNG: Festlegung, welche Bedingungen für das weitere Arbeiten mit ISAM gelten sollen ("Initialisieren vo ISAM")

EINGANGS-PARAMETER:

ISAM%(6)	Anzahl zusätzlicher Puffer
ISAM%(7)	max. Anzahl gleichzeitig offener Key-Files
ISAM%(1)	**"MI"**

AUSGANGS-PARAMETER:

ISAM%(8)	Return-Code

MÖGLICHE RETURN-CODES: Ø, 31

BEMERKUNGEN:
- Die zusätzlichen Puffer müssen nicht bei **sb** berücksichtigt werden. Es gilt: Je mehr zusätzliche Puffer, desto schneller die Verarbeitung, aber desto mehr Platzverlust im BASIC-Bereich desArbeitsspeichers. Werden Keys gelöscht (Einsatz der Funktionen **KD, SD** oder **DR**), muß mindestens ein zusätzlicher Puffer eingerichtet werden.
- Die Nrn. aller Key-Files müssen niedriger sein als alle später verwendeten File-Nrn. von Data-Files oder sonstigen Datenfiles. Die für offene Key-Files verwendeten File-Nrn. dürfen nicht für andere Files genutzt werden.

- Ob doppelte Schlüssel zulässig sein sollen, ist für jedes einzelne Key-File beim Anlegen über **OC** oder **OF** festzulegen.
- **MI** liefert Returncode 31, wenn bereits ein **MI** erfolgt ist. Es kann dennoch weitergearbeitet werden. Generell empfiehlt es sich, die **MI**-Funktion im Selbststartprogramm des gesamten Programmpaketes (INIT.BAS) das erste und einzige Mal durchzuführen.

BEISPIEL:

```
60 ISAM%(7)=3:REM MAXIMALE ANZAHL KEY-FILES -->
                  BASIC-FILES ERST AB NR. 4 !!
62 ISAM%(6)=1:REM ANZAHL ZUSATZLICHER PUFFER
64 ISAM$(1)="MI":GOSUB 60000
66 IF ISAM%(8)<>0 THEN 900 :REM FEHLERBEDINGUNG
```

VERWEISE: **MS, OF, OC**

FUNKTION: **MS**
(method status)

WIRKUNG: Ermittlung des aktuellen Zustandes einer index-sequentiellen Dateistruktur

EINGANGS-PARAMETER:

ISAM%(1)	Key-File-Nr.
ISAM$(1)	**"MS"**

AUSGANGS-PARAMETER:

ISAM%(6)	Anzahl aktiver Records im zugeordneten Data-File
ISAM%(7)	Anzahl noch nicht überschriebener, mit **DR** gelöschten Data-Records
ISAM%(8)	Return-Code
ISAM$(6)	doppelte Schlüssel J/N ISAM$(6) = **"U"**: keine doppelten Schlüssel zulässig ISAM$(6) = **"D"**: doppelte Schlüssel zulässig

MÖGLICHE RETURN-CODES: Ø, 31, 41, 71

BEMERKUNGEN: - Vor **MS** muß das Key-File geöffnet worden sein (**OO, OC** oder **OF**).

BEISPIEL:

```
155 ISAM%(1)=1:ISAM$(1)="MS":GOSUB 60000
156 IF ISAM%(8)<>0 THEN 900 :REM FEHLERBEDINGUNG
158 PRINT "gültige Records im Data-File:";ISAM%(6)
159 PRINT "Records im Data-File mit gelöschtem primären Key,
die aber noch nicht überschrieben sind:";ISAM%(7)
161 IF ISAM$(6)="D" THEN PRINT "doppelte Keys erlaubt"
                    ELSE PRINT "keine doppelten Keys erlaubt"
```

VERWEISE: **KD, SD, DR, OO, OC, OF**

FUNKTION: **OC**
(create file)

WIRKUNG: Anlegen und Öffnen eines nicht vorhandenen Key-Files

EINGANGS-PARAMETER:

ISAM%(1)	Key-File-Nummer
ISAM$(1)	**"OC"**
ISAM$(4)	Name des Key-Files
ISAM$(6)	**"U"** oder **"D"** **"U"**: keine doppelten Keys **"D"**: doppelte Keys möglich Default-Wert: **"U"**

AUSGANGS-PARAMETER:

ISAM%(8)	Return-Code

MÖGLICHE RETURN-CODES: Ø, 31, 41, 71

BEMERKUNGEN:
- Wenn das betreffende Key-File bereits auf Diskette existiert, wird Return-Code 71 gegeben.
- Die Key-File-Nr. muß niedriger liegen als die Nummern aller im Programm verwendeten Data-Files bzw. sonstiger Datenfiles.
- Ob doppelte Keys zulässig sein dürfen oder nicht, kann nach **OC** nicht mehr geändert werden. Eine Abfrage des Status kann mit **MS** erfolgen.

BEISPIEL:

```
82 ISAM%(1)=3:ISAM$(1)="OC"
84 ISAM$(4)="1:ARTIK.key":ISAM$(6)="D":GOSUB 60000
86 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
```

VERWEISE: **CL, OO, OF, MS**

FUNKTION: **OF**
(open/create file)

WIRKUNG: Öffnen eines bereits vorhandenen Key-Files, bei Nichtvorhandensein Anlegen und Öffenen

EINGANGS-PARAMETER:

ISAM%(1)	Key-File-Nummer
ISAM$(1)	**"OF"**
ISAM$(4)	Key-File-Name
ISAM$(6)	**"U"** oder **"D"** **"U"**: keine doppelten Keys **"D"**: doppelte Keys Default-Wert: **"U"**

AUSGANGS-PARAMETER: ISAM%(8) Return-Code

MÖGLICHE RETURN-CODES: Ø, 31, 41, 71, 81

BEMERKUNGEN:
- Die Key-File-Nummer muß niedriger liegen als die Nummern aller im Programm verwendeten Data-Files bzw. sonstiger Datenfiles.
- Ob doppelte Keys zulässig sein dürfen oder nicht, kann nach **OF** nicht mehr geändert werden. Eine Abfrage des Status kann mit **MS** erfolgen.

BEISPIEL:

```
62 ISAM%(1)=1:ISAM$(1)="OF":ISAM$(6)="U"
64 ISAM$(4)="1:KUNDEN.key":GOSUB 60000
66 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
```

VERWEISE: **CL, OC, OO, MS**

FUNKTION: **OO**
(file open)

WIRKUNG: Öffnen eines vorhandenen Key-Files

EINGANGS-PARAMETER:

ISAM%(1)	Key-File-Nummer
ISAM$(1)	**"OO"**
ISAM$(4)	Name des Key-Files

AUSGANGS-PARAMETER:

ISAM%(8)	Return-Code

MÖGLICHE RETURN-CODES: Ø, 31, 41, 71

BEMERKUNGEN:
- Wenn das Key-File noch nicht auf Diskette existiert oder die Filenummer schon vergeben ist, wird Return-Code 71 gegeben.
- Die Key-File-Nummer muß niedriger liegen als die Nummern aller im Programm verwendeten Data-Files bzw. sonstiger Datenfiles.
- Ob doppelte Schlüssel zulässig sind oder nicht, wurde früher mit **OC** oder **OF** festgelegt. Eine Abfrage des Status ist mit **MS** möglich.

BEISPIEL:

```
57 ISAM%(1)=2:ISAM$(1)="OO":ISAM$(4)="NAMEN.key"
58 REM ISAM$(6) wurde bei "OC" bzw. "OF" definiert
59 GOSUB 60000:IF ISAM%(8)<>0 THEN 600:REM FEHLER
```

VERWEISE: **CL, OC, OF, MS**

FUNKTION: **RG**
(read generic)

WIRKUNG: Lesen desjenigen primären Keys, der gleich dem als Suchbegriff angegebenen String ist oder - falls kein solcher vorhanden - des nächsten

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nummer
	ISAM%(5)	Data-Record-Nr. (nur bei doppelten Keys nötig)
	ISAM$(1)	**"RG"**
	ISAM$(2)	gesuchter String bei Nichtfinden des Strings wird der nächstgrößere Key in der sortierten Folge ausgegeben

AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	Wert des gelesenen Keys

MÖGLICHE RETURN-CODES: Ø, 31, 41, 51

BEMERKUNGEN:
- Nach Finden des ersten Keys einer Gruppe kann mit **RN** sofort der Folge-Key ermittelt werden, mit **RP** der vorherige.
- Sind doppelte Keys zulässig, sollte zusammen mit ISAM$(2) die zugeordnete Data-Record-Nr. über ISAM%(5) eingereicht werden. Über **RP** bzw. **RN** kann überprüft werden, ob und ggf. welche anderen Keys bzw. Record-Nrn. mit dem gleichen Key vorhanden sind. Wird ISAM%(5) = Ø gesetzt, wird derjenige Key gelesen, der als erster mit

dem Wert ISAM$(2) im Key-File vorhanden ist (ggf. der nächstgrößere).

- Wird über den höchsten vorhandenen Key hinaus gelesen, wird der Return-Code 51 gegeben.
- Der niedrigste primäre Key im Key-File kann durch Verwendung der **RN**-Funktion als erste ISAM-Funktion nach Öffnen des betreffenden Key-Files ermittelt werden.

BEISPIEL:

```
520 REM Key-File Nr. 2 ist offen, keine doppelten Keys
522 REM OPEN "R",5,"1:DATA.rnd",80  und
        FIELD 5,80 AS RECORD5$   sind erfolgt
523 LINE INPUT "Listen von,bis ";AKEY$;EKEY$
524 IF AKEY$="" AND EKEY$="" THEN 9900 :REM ENDVERARB.
525 ISAM%(1)=2:ISAM$(1)="RG"
527 ISAM$(2)=AKEY$:GOSUB 60000
529 IF ISAM%(8)=0 THEN 700:REM Beginn Liste
531 IF ISAM%(8)=51 THEN PRINT "keine Records hinter ";
AKEY$;" vorhanden":GOTO 9900:REM ENDVERARBEITUNG
533 GOTO 600:REM FEHLERBEDINGUNG
700 GET 5,ISAM%(9):LPRINT ISAM$(2);" ";RECORD5$
710 ISAM%(1)=2:ISAM$(1)="RN":GOSUB 60000
715 IF ISAM%(8)=51 THEN 730 :REM ENDE FILE
717 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
720 IF ISAM$(2)<=EKEY$ THEN 700
730 PRINT "ENDE":GOTO 9900:REM ENDVERARB.
```

VERWEISE: **SG, RN, RP**

FUNKTION: **RK**
(read key)

WIRKUNG: Lesen desjenigen primären Keys, der gleich dem als Suchbegriff angegebenen String ist; falls kein solcher Key vorhanden ist, wird Return-Code 51 gemeldet.

EINGANGS-	ISAM%(1)	Key-File-Nr.
PARAMETER:	ISAM%(5)	Data-Record-Nr.
		(nur bei doppelten Keys nötig)
	ISAM$(1)	**"RK"**
	ISAM$(2)	gesuchter Key
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.

MÖGLICHE
RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Sind doppelte Keys zulässig, sollte zusammen mit ISAM$(2) die zugeordnete Data-Record-Nr. über ISAM%(5) eingereicht werden.
- Über **RP** bzw. **RN** kann überprüft werden, ob und ggf. welche anderen Keys bzw. Record-Nrn. mit dem gleichen Key vorhanden sind. Wird ISAM%(5) = Ø gesetzt, wird derjenige Key gelesen, der als erster mit dem Wert ISAM$(2) im Key-File vorhanden ist.

BEISPIEL:

```
195 REM Doppelte Keys möglich
195 REM 1. Record mit Key T$ im Data-File #6
211 ISAM%(1)=2:ISAM%(5)=0:ISAM$(1)="RK"
214 ISAM$(2)=T$:GOSUB 60000
216 IF ISAM%(8)=51 THEN PRINT T$;" nicht vor
handen":GOTO 100:REM weiter
217 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
219 GET 6,ISAM%(9)
```

VERWEISE: **SK, RN, RP**

FUNKTION: **RN**
(read next)

WIRKUNG: Lesen desjenigen primären Keys, der dem zuletzt über **RG**, **RK**, **RP** oder **RN** gelesenen in der sortierten Folge folgt, bzw. des ersten primären Keys im Key-File.

EINGANGS-	ISAM%(1)	Key-File-Nr.
PARAMETER:	ISAM$(1)	**"RN"**
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	gelesener Key

MÖGLICHE
RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Wird **RN** als erste Funktion nach dem Öffnen des betreffenden Key-Files aufgrufen, wird der niedrigste Key im Key-File gelesen.
- Doppelte Keys werden in der Reihenfolge nach aufsteigender Data-Record-Nr. gelesen, sofern nicht ein **DR** für einen dieser Keys erfolgt ist, der später mit **WA** überschrieben wurde.
- Der erste Key einer Gruppe kann mit Hilfe von **RG** ermittelt werden.
- Wird über den höchsten vorhandenen Key hinaus gelesen, wird der Return-Code 51 gegeben.

BEISPIELE:

```
223 REM Fortsetzung Beispiel von "RK"
225 ISAM%(1)=2:ISAM$(1)="RN":GOSUB 60000
227 IF ISAM%(8)=51 THEN 9900:REM ENDE FILE
229 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
231 IF ISAM$(2)<>T$ THEN 100:REM weiter
233 GET 6,ISAM%(9):GOTO 225
```

```
71 REM Suche niedrigster Key, Key-File Nr. 3 (DATA.key)
       ist vorhanden und OFFEN ("0F", "0C" oder "00")
75 ISAM%(1)=3:ISAM$(1)="CL":GOSUB 60000
77 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
79 ISAM%(1)=3:ISAM$(1)="00":ISAM$(4)="1:DATA.key"
81 GOSUB 60000:IF ISAM%(8)<>0 THEN 600
83 ISAM%(1)=3:ISAM$(1)="RN"
85 IF ISAM%(8)=51 THEN PRINT "KEY-FILE LEER":GOTO 9900
87 IF ISAM%(8)<>0 THEN 600
89 PRINT "Niedrigster Key: ";ISAM$(2):GET 7,ISAM%(9)
```

VERWEISE: **RK, RG, RP, SN**

FUNKTION: **RP**
(read previous)

WIRKUNG: Lesen desjenigen primären Keys, der dem zuletzt über **RG, RK, RP** oder **RN** gelesenen in der sortierten Folge vorausgeht.

EINGANGS-	ISAM%(1)	Key-File-Nr.
PARAMETER:	ISAM$(1)	**"RP"**
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	gelesener Key

MÖGLICHE
RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Doppelte Keys werden im allgemeinen in der Reihenfolge nach absteigender Data-Record-Nr. gelesen. Wenn ein **DR** mit späterem **WA** für einen doppelten Key erfolgte, kann dies jedoch zu einer geänderten Reihenfolge führen.
- Der letzte Key einer Gruppe, die in absteigender Reihenfolge gelesen werden soll, kann mit Hilfe von **RG** ermittelt werden.
- **RP** nach Lesen des ersten Keys in sortierter Folge liefert den Return-Code 51.

BEISPIEL:

```
224 REM Lesen in ABSTEIGENDER Reihenfolge
225 REM Erster Key wurde mit "RG" gefunden
226 REM Es soll bis EKEY$ gelesen werden
227 ISAM%(1)=2:ISAM$(1)="RP":GOSUB 60000
228 IF ISAM%(8)=51 THEN 9900:REM ANFANG FILE
229 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
231 IF ISAM$(2)<EKEY$ THEN 9900
233 GET 6,ISAM%(9):GOTO 227
```

VERWEIS: **RK, RG, RN, SP**

FUNKTION: **SA**
(secondary add)

WIRKUNG: Schreiben (Einfügen) eines neuen sekundären Keys in das Key-File; der zugeordnete Data-Record muß bereits vorhanden sein.

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM%(5)	Data-Record-Nr.
	ISAM$(1)	**"SA"**
	ISAM$(2)	einzufügender Key
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	Data-Record-Nr.

MÖGLICHE RETURN-CODES: Ø, 31, 41, 61, 71

BEMERKUNGEN:
- In ISAM%(5) ist anzugeben, welche Data-Record-Nr. dem sekundären Key zugeordnet werden soll. Dieser Data-Record muß im Key-File des primären Keys vorhanden sein.
- Der Key wird nicht notwendigerweise sofort auf Diskette geschrieben. Erst **CL** stellt sicher, daß die indexsequentielle Dateistruktur ordnungsgemäß aus Diskette gespeichert ist.
- Vor **SA** muß der primäre Key mit **WA** erfaßt und der Data-Record per **PUT** in das Data-File geschrieben worden sein.
- ISAM vergrößert das Key-File auf Diskette automatisch, falls dies nötig werden sollte.
- Doppelte Keys können bei **SA** nur eingefügt werden, wenn dies beim Anlegen des Key-Files

über **OC** oder **OF** durch Setzen des Eingangsparameters ISAM%(6) auf **"D"** festgelegt wurde. Eine nachträgliche Änderung ist nicht mehr möglich. Ob doppelte Keys für ein Key-File zulässig sind oder nicht, kann mit **MS** abgefragt werden.

- Der Key darf maximal 110 Zeichen lang sein; eine feste Key-Länge ist nicht nötig.
- Das Löschen eines sekundären Keys darf nur mit **SD** erfolgen; **DR** würde die indexsequentielle Struktur zerstören.
- Wird mit Löschen gearbeitet, empfiehlt es sich, für jeden Key im Data-Record ein Löschkennzeichen zu setzen, das Auskunft darüber gibt, ob für den betreffenden Data-Record noch ein (primärer bzw. sekundärer) Key von ISAM verwaltet wird (vgl. Beispiel zu **DR**)

Beispiel:

```
620 REM Neuerfassen: Key-File 1=Primärkey, #2=sek.Key
622 REM OPEN und FIELD für Data-File 6 erfolgt
672 REM sek. Key und Lösch-KZ in den Record übernehmen
673 REM vgl. Problematik von DR (s. dortiges Beispiel)
628 REM doppelter Key nur bei Sekundärkey zulässig
629 LINE INPUT "prim. Key ";ISAM$(2)
631 ISAM%(1)=1:ISAM$(1)="RK":GOSUB 60000
633 IF ISAM%(8)=0 THEN 629:REM schon vorhanden
635 IF ISAM%(8)<>51 THEN 1600:REM FEHLERBEDINGUNG
636 RNR%=ISAM%(9)
652 ISAM%(1)=1:ISAM$(1)="WA":GOSUB 60000
655 REM IF ISAM%(8)=61 THEN 629 ist überflüssig wegen
    vorheriger Abfrage mit RK !
656 IF ISAM%(8)<>0 THEN 1600:REM FEHLERBEDINGUNG
658 LINE INPUT "sek. Key ";SKEY$:ISAM$(2)=SKEY$
665 ISAM%(1)=2:ISAM$(1)="SA":ISAM%(5)=RNR%:GOSUB 60000
667 REM ISAM%(8) <> 61 bei doppeltem Key!
668 IF ISAM%(8)<>0 THEN 1600:REM FEHLERBEDINGUNG
671 REM Annehmen Data-Record, Löschkennzeichen=CHR$(1)
673 REM (vgl. Logik von Beispiel zu DR) und Übergabe der
675 REM Werte in den Puffer für File #6
681 PUT 6,ISAM%(9):REM Record war bisher noch nicht erfaßt
```

VERWEISE: **WA, SD, CL, OC, OF, MS**

FUNKTION: **SD**
(secondary delete)

WIRKUNG: Löschen eines sekundären Keys aus dem Key-File ohne Freigabe der zugeordneten Data-Record-Nr. zum späteren Überschreiben (logisches Löschen).

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM%(5)	Data-Record-Nr. (nur bei doppelten Keys nötig)
	ISAM$(1)	**"SD"**
	ISAM$(2)	zu löschender Key
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	betroffene Data-Record-Nr.

BEMERKUNGEN:
- Das Reaktivieren eines mit **SD** gelöschten Keys über ISAM ist nicht möglich. Der betreffende Data-Record bleibt allerdings vorhanden.
- Der Key wird aus dem Key-File gelöscht und später durch einen neuen Key überschrieben.
- Für das Löschen von Keys muß bei **MI** ein zusätzlicher Puffer eingerichtet worden sein.
- Wird mit Löschen gearbeitet, empfiehlt es sich, für jeden Key im Data-Record ein Löschkennzeichen zu setzen, das Auskunft darüber gibt, ob für den betreffenden Data-Record noch ein (primärer bzw. sekundärer) Key von ISAM verwaltet wird (vgl. Beispiel zu **DR**).

BEISPIEL:

```
900 REM Data-File ist OPEN unter Nr. 5, Key-File geöffnet
        unter Nr. 2
910 ISAM%(1)=2:ISAM$(1)="SD":ISAM$(2)="A6083"
920 ISAM%(5)=DATARECNR%:REM Recordnummer bei doppeltem Key
930 GOSUB 60000:IF ISAM%(8)<>0 THEN 600:REM nicht vorhanden
    oder Fehlerbedingung
```

VERWEISE: **KD, DR**

FUNKTION: **SG**
(secondary generic)

WIRKUNG: Lesen desjenigen sekundären Keys, der gleich dem als Suchbegriff angegebenen String ist oder - falls kein solcher vorhanden - des nächsten

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM%(5)	Data-Record-Nr. (nur bei doppelten Keys nötig)
	ISAM$(1)	**"SG"**
	ISAM$(2)	gesuchter String bei Nichtfinden des Strings wird der nächstgrößere Key in der sortierten Folge ausgegeben
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	Wert des gelesenen Keys

MÖGLICHE RETURN-CODES: Ø, 31, 41, 51

BEMERKUNGEN:
- Nach Finden des ersten Keys einer Gruppe kann mit **SN** sofort der Folge-Key ermittelt werden, mit **SP** der vorherige.
- Sind doppelte Keys zulässig, sollte zusammen mit ISAM$(2) die zugeordnete Data-Record-Nr. über ISAM%(5) eingereicht werden. Über **SP** bzw. **SN** kann überprüft werden, ob und ggf. welche anderen Keys bzw. Record-Nrn. mit dem gleichen Key vorhanden sind. Wird ISAM%(5) = Ø gesetzt, wird derjenige Key gelesen, der als erster mit

dem Wert ISAM$(2) im Key-File vorhanden ist (ggf. der nächstgrößere).

- Wird über den höchsten vorhandenen Key hinaus gelesen, wird der Return-Code 51 gegeben.
- Der niedrigste sekundäre Key im Key-File kann durch Verwendung der **SN**-Funktion als erste ISAM-Funktion nach Öffnen des betreffenden Key-Files ermittelt werden.

BEISPIEL:

```
520 REM Key-File Nr. 2 ist offen, keine doppelten Keys
522 REM OPEN "R",5,"1:DATA.rnd",80  und
        FIELD 5,80 AS RECORD5$   sind erfolgt
523 LINE INPUT "Listen von,bis ";AKEY$;EKEY$
524 IF AKEY$="" AND EKEY$="" THEN 9900 :REM ENDVERARB.
525 ISAM%(1)=2:ISAM$(1)="SG"
527 ISAM$(2)=AKEY$:GOSUB 60000
529 IF ISAM%(8)=0 THEN 700:REM Beginn Liste
531 IF ISAM%(8)=51 THEN PRINT "keine Records hinter ";
AKEY$;" vorhanden":GOTO 9900:REM ENDVERARBEITUNG
533 GOTO 600:REM FEHLERBEDINGUNG
700 GET 5,ISAM%(9):LPRINT ISAM$(2);" ";RECORD5$
710 ISAM%(1)=2:ISAM$(1)="SN":GOSUB 60000
715 IF ISAM%(8)=51 THEN 730 :REM ENDE FILE
717 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
720 IF ISAM$(2)<=EKEY$ THEN 700
730 PRINT "ENDE":GOTO 9900:REM ENDVERARB.
```

VERWEISE: **RG, SN, SP**

FUNKTION: **SK**
(secondary key)

WIRKUNG: Lesen desjenigen sekundären Keys, der gleich dem als Suchbegriff angegebenen String ist; falls kein solcher Key vorhanden ist, wird Return-Code 51 gemeldet.

EINGANGS-	ISAM%(1)	Key-File-Nr
PARAMETER:	ISAM%(5)	Data-Record-Nr. (nur bei doppelten Keys nötig)
	ISAM$(1)	**"SK"**
	ISAM$(2)	gesuchter Key
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.

MÖGLICHE
RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Sind doppelte Keys zulässig, sollte zusammen mit ISAM$(2) die zugeordnete Data-Record-Nr. über ISAM%(5) eingereicht werden.
- Über **SP** bzw. **SN** kann überprüft werden, ob und ggf. welche anderen Keys bzw. Record-Nrn. mit dem gleichen Key vorhanden sind. Wird ISAM%(5) = Ø gesetzt, wird derjenige Key gelesen, der als erster mit dem Wert ISAM$(2) im Key-File vorhanden ist.

BEISPIEL:

```
195 REM Doppelte Keys möglich
195 REM 1. Record mit Key T$ im Data-File #6
211 ISAM%(1)=2:ISAM%(5)=0:ISAM$(1)="SK"
214 ISAM$(2)=T$:GOSUB 60000
216 IF ISAM%(8)=51 THEN PRINT T$;" nicht vor
handen":GOTO 100:REM weiter
217 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
219 GET 6,ISAM%(9)
```

VERWEISE: **RK, SN, SP**

FUNKTION: **SN**

(secondary next)

WIRKUNG: Lesen desjenigen sekundären Keys, der dem zuletzt über **SG**, **SK**, **SP** oder **SN** gelesenen in der sortierten Folge folgt bzw. des ersten sekundären Keys im Key-File.

EINGANGS-	ISAM%(1)	Key-File-Nr.
PARAMETER:	ISAM$(1)	**"SN"**
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	gelesener Key

MÖGLICHE RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Wird **SN** als erste Funktion nach dem Öffnen des betreffenden Key-Files aufgrufen, wird der niedrigste Key im Key-File gelesen.
- Doppelte Keys werden in der Reihenfolge nach aufsteigender Data-Record-Nr. gelesen, sofern nicht ein **DR** für einen zugeordneten primären Key erfolgte und dieser später mit **WA** überschrieben wurde.
- Der erste Key einer Gruppe kann mit Hilfe von **SG** ermittelt werden.
- Wird über den höchsten vorhandenen Key hinaus gelesen, wird der Return-Code 51 gegeben.

BEISPIELE:

```
223 REM Fortsetzung Beispiel von "SK"
225 ISAM%(1)=2:ISAM$(1)="SN":GOSUB 60000
227 IF ISAM%(8)=51 THEN 9900:REM ENDE FILE
229 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
231 IF ISAM$(2)<>T$ THEN 100:REM weiter
233 GET 6,ISAM%(9):GOTO 225
```

```
71 REM Suche niedrigster Key, Key-File Nr. 3 (DATA.key)
       ist vorhanden und OFFEN ("OF", "OC" oder "OO")
75 ISAM%(1)=3:ISAM$(1)="CL":GOSUB 60000
77 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
79 ISAM%(1)=3:ISAM$(1)="OO":ISAM$(4)="1:DATA.key"
81 GOSUB 60000:IF ISAM%(8)<>0 THEN 600
83 ISAM%(1)=3:ISAM$(1)="SN"
85 IF ISAM%(8)=51 THEN PRINT "KEY-FILE LEER":GOTO 9900
87 IF ISAM%(8)<>0 THEN 600
89 PRINT "Niedrigster Key: ";ISAM$(2):GET 7,ISAM%(9)
```

VERWEISE: **SK, SG, SP, RN**

FUNKTION: **SP**
(secondary previous)

WIRKUNG: Lesen desjenigen sekundären Keys, der dem zuletzt über **SG, SK, SP** oder **SN** gelesenen in der sortierten Folge vorausgeht.

EINGANGS-	ISAM%(1)	Key-File-Nr.
PARAMETER:	ISAM$(1)	**"SP"**
AUSGANGS-	ISAM%(8)	Return-Code
PARAMETER:	ISAM%(9)	Data-Record-Nr.
	ISAM$(2)	gelesener Key

MÖGLICHE
RETURN-CODES: Ø, 31, 41, 51, 71

BEMERKUNGEN:
- Doppelte Keys werden im allgemeinen in der Reihenfolge nach absteigender Data-Record-Nr. gelesen. Wenn ein **DR** mit späterem **WA** für einen zugeordneten primären Key erfolgte, kann dies jedoch zu einer geänderten Reihenfolge führen.
- **SP** nach Lesen des ersten Keys in sortierter Folge liefert den Return-Code 51.

BEISPIEL:

```
224 REM Lesen in ABSTEIGENDER Reihenfolge
225 REM Erster Key wurde mit "SG" gefunden
226 REM Es soll bis EKEY$ gelesen werden
227 ISAM%(1)=2:ISAM$(1)="SP":GOSUB 60000
228 IF ISAM%(8)=51 THEN 9900:REM ANFANG FILE
229 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
231 IF ISAM$(2)<EKEY$ THEN 9900:REM weiter
233 GET 6,ISAM%(9):GOTO 227
```

VERWEISE: **SK, SG, SN, RP**

FUNKTION: **WA**
(write add)

WIRKUNG: Schreiben (Einfügen) eines neuen primären Keys in das Key-File und Zuordnung einer freien Data-Record-Nr. zu diesem Key.

EINGANGS-PARAMETER:	ISAM%(1)	Key-File-Nr.
	ISAM$(1)	**"WA"**
	ISAM$(2)	einzufügender Key
AUSGANGS-PARAMETER:	ISAM%(8)	Return-Code
	ISAM%(9)	Data-Record-Nr.

MÖGLICHE RETURN-CODES: Ø, 31, 41, 61, 71

BEMERKUNGEN:
- Der Key wird nicht notwendigerweise sofort auf Diskette geschrieben. Erst **CL** stellt sicher, daß die indexsequentielle Dateistruktur ordnungsgemäß auf Diskette gespeichert ist.
- Nach **WA** muß der Data-Record per **PUT** unter der Record-Nr. ISAM%(9) in das Data-File geschrieben werden.
- ISAM vergrößert das Key-File auf Diskette automatisch, falls dies nötig werden wollte.
- Sind im Stack für mit **DR** gelöschte Keys noch nicht verwendete Data-Record-Nrn., wird die zuletzt dort aufgenommene für ISAM%(9) verwendet.
- Ein sekundärer Key darf nur mit **SA** aufgenommen werden, nachdem der betreffende primäre Key mit **WA** erfaßt und der Data-Record mit **PUT** geschrieben wurde.

- Doppelte Keys können bei **WA** nut eingefügt werden, wenn dies beim Anlegen des Key-Files über **OC** oder **OF** durch Setzen des Eingangsparameters ISAM%(6) auf **"D"** festgelegt wurde. Eine nachträgliche Änderung ist nicht mehr möglich. Ob doppelte Keys für ein Key-File zulässig sind oder nicht, kann mit **MS** abgefragt werden.
- Der Key darf maximal 110 Zeichen lang sein; eine feste Key-Länge ist nicht nötig.
- Wird mit Löschen gearbeitet, empfiehlt es sich, für jeden Key im data-Record ein Löschkennzeichen zu setzen, das Auskunft darüber gibt, ob für den betreffenden Data-Record noch ein (primärer bzw. sekundärer) Key von ISAM verwaltet wird (vgl. Beispiel zu **DR**).

BEISPIEL:

```
625 REM Neuerfassen; Key-File sei unter #2 offen
626 REM OPEN "R",5,"1:NAMEN.rnd",50 und
        FIELD 5,50 AS B.NAM$ sei erfolgt
627 REM keine doppelten Keys möglich
629 LINE INPUT "KUNDEN-Nr. ";NR$:ISAM$(2)=NR$
631 ISAM%(1)=2:ISAM$(1)="RK":GOSUB 60000
633 IF ISAM%(8)=0 THEN PRINT "schon vorhanden":
            GOTO 629
635 IF ISAM%(8)<>51 THEN 600:REM Fehlerbedingung
652 ISAM%(1)=2:ISAM$(1)="WA":GOSUB 60000
655 IF ISAM%(8)=61 THEN PRINT "schon vorhanden":
            GOTO 629
656 IF ISAM%(8)<>0 THEN 600:REM FEHLERBEDINGUNG
659 REM Eingabe des Data-Records und Zuweisung
    auf B.NAM$ per LSET
661 PUT 5,ISAM%(9)
```

VERWEISE: **SA, DR, CL, OC, OF, MS**